

## Alex Murray

Small description: The program is an educational tool, that will be used to teach/help to teach others about computers and robotics. It will include small tests on topics, as well as worked examples of how computers 'see' and work through things. Teachers will be able to review the data by students to further help them. The project also includes a physical bot; this will automatically move around collecting data to be sorted by the main program as an example of what happens to data as it processed.

### Bot:

- Movement: The bot will be able to manoeuvre around a room, turning when it hits an obstacle in its way (hence avoiding it)
- Logging data: The bot will log data based on when it hits something, the time and sensor hit. This will be recorded and then used as data to be processed.
- Memory: The Bot will save all its logged data onto a micro SD card fitted onto it. The data will be saved onto simple text files to be read, copied, and then deleted from the SD card by the main program (VB)
- Bot Processing: The processing power of the bot will be from an Arduino board (Arduino Uno) This will control the wheels and therefore the direction on the bot, as well as saving data about the bot's journey to the SD card.

### Main Program:

- Login: Each user will be able to login to the system, or set up a new account with the system. The system will be based on having different levels of access (e.g. a teacher could access the pupil's data and change scores, but only the admin could add a new teacher.) The login will also allow: Saving of mini tests to each user – saving of mini games – identification of message on any question sent or received – admin privileges – etc.
- Bot Data input: Data collected from the bot will be saved off the SD card and onto a database in the main program. The data will then be processed by my main program, sorted, searched and checked, before it's translated into graphs/graphics.
- Search function: a function that allows users to search through data and information given by the programs (e.g. names of users, date of bot journey etc.) to find the original information related to the data, or just the information itself.
- Edit: Data entered will be able to be manipulated and edited, but after being edited will still have a way of obtaining original data (from a notification to alert the user of the change and a file will the original data to be kept.)
- Interactive Learning: The main program will have a page on it for interactive learning, it will help to educate people (both with previous knowledge and without) about robotics in the modern world, with different levels of knowledge available.
- Teacher's uses: a teacher will be able to monitor and track a student's score via a search function, they will also be able to set extra tasks.

Technical limitations:

- The main limitation on the bot will be its lack of hardware, the wheels of the bot (and the chassis) are small, and made of plastic, this will limit what surfaces the bot can travel on (and therefore the variation of data it can obtain.) The sensors on the bot will be another limitation, as the Arduino board has a limited amount of inputs and outputs.
- The main program's limitation will be the systems that it runs on – the program will be made and designed for windows, it will not be compatible with other systems, it will also not be available on a mobile device.

Other limitations:

- The program could have also been developed into a website, this way data could be more easily saved. It would also mean the content is accessible without the need to install the software.
- Another element that could have been added would be a form of “route guidance” that lets the user control where the bot ‘should’ move to. I decided not to add this as it would have required either an additional Arduino board, or educating the user how to upload a program onto the Arduino board which would cause too many problems.

### **Feed Back**

After I had finish my description and aims, I showed it to a different selection of people and asked for feedback. I decided to ask three different groups: Teachers -as they would know what features an educational program would need, Students with a higher than average understanding of computers (They took it for GCSE/are taking it for A level – as they could produce feedback with more ideas/technical input, and they would have a better understanding on the reality of my goals. Students with an average understanding of computers – As they can give me a better understanding of it the project would be useful to them or if I have missed anything important.

**Teachers:** I asked two teachers for feedback, to make sure I still have a variety of information in my feedback one taught computing, the other English. Both teachers commented on how the program would be giving back information to them and the students regarding the ‘mini’ test scores. With one questioning if ‘students can track and evaluate their own progress’ rather than have only allow the teachers access. The other teacher commented on how they would track scores and monitor progress.

This indicates that if I am to make the educational side of the program effective I will have to put more consideration into how both teachers and students will track progress, as this appears to be a key area for teachers. The value of tracking progress and seeing where improvement is needed, is greater for teachers than I had thought. To improve this I will need to research some current methods of tracking progress.

**Students taking computing:** The students who had taken computing had a more technical input than the other students. A reoccurring theme given by these students, involved the bot, with a request for more information, as I had not made it clear enough on how the bot will be used. I will need to further explain how the bot will be used to log data and how the data given can be used to educate others. Also the question of 'what the bot would do if trapped' I had not yet thought about this kind of scenario, as a result I can now implement 'safety' features as well as a form of validation for if the bot is stuck, and reads false data.

On the questionnaire I gave out I asked if users would be willing to give personal information (such as an email address) to the program. Most responses were negative (despite them leaving a contact email with me.) One of the students suggested that users would be willing to leave person information with the system if it had some sort of encryption 'such as XOR' to protect the information given. I believe the statement to be true as most people did leave an email to me, (probably as I am 'trusted' and safe.) So I will now look further into encryption, and the different ways I could make the users feel safe enough to leave data.

**Overall:** As I had range of feedback, from different points of views, I was able to pick out some overlaps. Some of the feedback suggested I uses a more visual approach, with 'interactive videos' and 'YouTube' being mentioned multiple times, although useful, this would involve a lot of time to make the videos. I will look into if it's feasible to add them into the project.

Most of the feedback was useful as I have been given insight on my project, I can now focus on making the aims of my project clearer to others, looking into ways of encrypting the data so users feel more inclined to give it to the system, and rethink the methods of showing data on how well a student's progress is, as well as who is allowed access to how well the student is doing.

**Do you feel like you understand the idea my project, and the goals I hope to achieve? If not why?(Rating of understanding out of 10 – 10 being very good)**

7, because I am not exactly sure what the users will be learning by using the bot... Perhaps a bit more detail on these facts.

**Have I missed anything important that I might need in my project?**

Validation on the bot for what to do in situations where is trapped etc.

**Are there any ideas that I should not include/that are too complex to implement?**

Im not sure how you will save deleted data and keep as an original but good luck.

**Do you feel like I should re-evaluate any areas of the project? If so which areas?**

I think the base idea for the project is both interesting and fun for users. Giving a hands on learning tool for students to understand about robots. I would maybe recommend having the bot have sensors to detect when it is getting too close to an object to then turn and avoid. Rather than a pressure sensor. I feel this will add some more complexity to the program.

**Should there be more focus be put onto the teaching/testing/bot/data manipulation slide? If so why?**

Yes, because that way it will have a real world use if it had some real world teaching abilities.

**Would you use this project to teach/learn?** Y/~~N~~

**Do you like the 'real' (bot) side of the project?** Y/~~N~~

**Is it useful to have the physical example of the bot?** Y/~~N~~

**Would you be willing to give personal details to the system? (e.g. email)** ~~Y~~/N

**Are there any questions or notes you would like to ask or add?**

I would not give **personal** details to the system as you haven't discussed any form of encryption, such as XOR. Maybe attempt to integrate that and more people would be willing to give their details to the system.

Thank you for your feedback. - Please leave your name and T-number/(email) - Will Scott - T0045288

Alex Murray - Computing Project

Do you feel like you understand the idea my project, and the goals I hope to achieve?  
If not why?  
(Rating of understanding out of 10 - 10 being very good)

9/10

Have I missed anything important that I might need in my project?

No.

Are there any ideas that I should not include/that are too complex to implement?

No.

Do you feel like I should re-evaluate any areas of the project? If so which areas?

Not really.

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?

If so why?

~~Teaching~~ - to expand the ways in which the bot can help people learn

Would you use this project to teach/learn?

Do you like the "real" (bot) side of the project?

Is it useful to have the physical example of the bot?

Would you be willing to give personal details to the system? (e.g. email)

Are there any questions or notes you would like to ask or add?

YN

YN

YN

YN

Thank you for your feedback.

Please leave your name and T-number/(email)

T0044742@cardinalnewman.ac.uk.

Alex Murray - Computing Project

Do you feel like you understand the idea <sup>of</sup> my project, and the goals I hope to achieve?  
If not why?  
(Rating of understanding out of 10 - 10 being very good)

6

Have I missed anything important that I might need in my project?

I'm not sure

Are there any ideas that I should not include/that are too complex to implement?

The logging data.

Do you feel like I should re-evaluate any areas of the project? If so which areas?

NaN

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?

If so why?

Testing to ensure everything works properly in the bot before making the final bot.

Would you use this project to teach/learn?

Do you like the "real" (bot) side of the project?

Is it useful to have the physical example of the bot?

Would you be willing to give personal details to the system? (e.g. email)

Are there any questions or notes you would like to ask or add?

YN

YN

YN

YN

Thank you for your feedback.

Please leave your name and T-number/(email)

Jenny  
Wilton

T0044330@cardinal  
newman.ac.uk

Alex Murray - Computing Project

Do you feel like you understand the idea my project, and the goals I hope to achieve?  
if not why?  
(Rating of understanding out of 10 - 10 being very good)

9

Have I missed anything important that I might need in my project?

I don't think so

Are there any ideas that I should not include/that are too complex to implement?

different colour

Do you feel like I should re-evaluate any areas of the project? If so which areas?

nope

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?  
if so why?

nope

Would you use this project to teach/learn?

Y/N

Do you like the 'real' (bot) side of the project?

Y/N

Is it useful to have the physical example of the bot?

Y/N

Would you be willing to give personal details to the system? (e.g. email)

Y/N

Are there any questions or notes you would like to ask or add?

Thank you for your feedback.  
Please leave your name and T-number/(email)

70047260@cardinalnewman.ac.uk

Alex Murray - Computing Project

Do you feel like you understand the idea of my project, and the goals I hope to achieve?  
if not why?  
(Rating of understanding out of 10 - 10 being very good)

7/10

Have I missed anything important that I might need in my project?

Help files and/or interactive videos

Are there any ideas that I should not include/that are too complex to implement?

no! see about 'edit' functionality.

Do you feel like I should re-evaluate any areas of the project? If so which areas?

web portal!!

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?  
if so why?

teaching - as main aim of project

Would you use this project to teach/learn?

Y/N

Do you like the 'real' (bot) side of the project?

Y/N

Is it useful to have the physical example of the bot?

Y/N

Would you be willing to give personal details to the system? (e.g. email)

Y/N

Are there any questions or notes you would like to ask or add?

Thank you for your feedback.  
Please leave your name and T-number/(email)

Dave Murray

Alex Murray - Computing Project

Do you feel like you understand the idea my project, and the goals I hope to achieve?  
If not why?  
(Rating of understanding out of 10 - 10 being very good)

10

Have I missed anything important that I might need in my protect?

No

Are there any ideas that I should not include/that are too complex to implement?

No, everything seems easy to understand

Do you feel like I should re-evaluate any areas of the project? If so which areas?

No areas seem to be more difficult

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?  
If so why?

No

Would you use this project to teach/learn? Y/N  
Do you like the 'real' (bot) side of the project? Y/N  
Is it useful to have the physical example of the bot? Y/N  
Would you be willing to give personal details to the system? (e.g. email) Y/N

Are there any questions or notes you would like to ask or add?  
No

Thank you for your feedback.  
Please leave your name and T-number/(email)

Joe Rigby T0043344@reading.ac.uk

Alex Murray - Computing Project

Do you feel like you understand the idea my project, and the goals I hope to achieve?  
If not why?  
(Rating of understanding out of 10 - 10 being very good)

7 I can't tell when you're talking about data on the screen or data from the robot.

Have I missed anything important that I might need in my protect?

No

Are there any ideas that I should not include/that are too complex to implement?

Multiple difficulty levels may not be needed if target is A-Level.

Do you feel like I should re-evaluate any areas of the project? If so which areas?

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?  
If so why?  
Low focus on testing as the rest is more educational.

Would you use this project to teach/learn? Y/N  
Do you like the 'real' (bot) side of the project? Y/N  
Is it useful to have the physical example of the bot? Y/N  
Would you be willing to give personal details to the system? (e.g. email) Y/N

Are there any questions or notes you would like to ask or add?  
Consider how the bot will be controlled, as it seems to not have much info on this in documentation.

Thank you for your feedback.  
Please leave your name and T-number/(email)

Carson Hayes T0043344@reading.ac.uk  
T0048820

Alex Murray - Computing Project

Do you feel like you understand the idea of my project, and the goals I hope to achieve? **Yes**  
If not why?  
(Rating of understanding out of 10 - 10 being very good)

7 (I'm not good at computing)

Have I missed anything important that I might need in my protect? **project**  
Using an app I think to update phone technology eg "Robot" advertised at times at Mopins

Are there any ideas that I should not include/that are too complex to implement?

Possibly the above

Do you feel like I should re-evaluate any areas of the project? If so which areas?

Can't say - not a specialist

Should there be more focus be put onto the teaching/testing/bot/data manipulation slide?  
If so why?

Depends - what are aims + objectives?

Would you use this project to teach/learn? **No - I'm not into computers** Y/N  
Do you like the 'real' (bot) side of the project? Y/N  
Is it useful to have the physical example of the bot? Y/N  
Would you be willing to give personal details to the system? (e.g. email) Y/N

Are there any questions or notes you would like to ask or add?

What is the cost  
timescale  
use of data - can students track + evaluate their own progress?

Thank you for your feedback.  
Please leave your name and T-number/(email)

- what about Youtube tips/tutorials?  
- projected costs?

# Computing Coursework - Alex Murray



Carberry, Chris

Fri 06/05

Murray, Alexander Thomas (T0043344)

Reply all

Action Items

I assume the bot will be running automatically - can the tests be related to the code that makes the bot run perhaps - with questions on what changes in the code would change the behaviour of the bot etc

Could the user be allowed to write their own code for an 'on screen' bot which would represent a real life system and allow the user to test if their code works correctly?

Would the teacher be able to track scores, set new tests/ monitor progress etc?

### **Investigation into current systems**

Current system – Most methods of learning/information about computers/how computers work are limited to being hidden away in classrooms. The most common “system” of learning about computers are in the classroom, mainly text based, lessons with a lecture (accompanied by a PowerPoint) talk and discuss about how computers work, and how systems and algorithms are used. I will be focusing my current system on classrooms teachings.

Lessons/lectures about computers mainly involve a teacher/lecturer providing information on specific topics on computers (e.g. topic on algorithms, ethical issues, etc.) These topics are given by the lecturer or by an exam board to later test its students on. The information is provided by a mixture of PowerPoints, Text books, videos, lectures, and some diagrams. Different methods of teaching allow a variety of different knowledge to be passed on, no two lessons will be exactly the same. For example, a student can ask the teacher a question, and then get instant feedback, but the question could differ per student or not be asked at all. Because each lesson is different, and because of the verity of different methods of teaching I will have to carry out different methods of investigating how well this system works. I will conduct a questionnaire to gain a brief understanding from lots of different students, as well as observations of lessons as this will give me my own depth of how the lessons are to the student and how effective they are, but as each lesson is different I will also conduct a few interviews to see what others think and feel about the lessons. The documents currently used will give me a much bigger understanding of how the ‘system’ works, as textbooks will hold layouts and ordering of information, whereas tests and mark schemes will show me how the goal is achieved.

### **Questionnaires:**

Name: Amy Banks

Tick all you have/are studying:  
GCSE  AS Level  A Level  Degree  Other (e.g. Online tutorial)   
If other please specify: Khan Academy

How often do you have lessons in a week?:  
Less than once per week  1-2 times  3-4 times  5+ times

How are your topics chosen?:  
By you  By the teacher  By exam board   
By you from a list of available topics given

Why do you lean the information?:  
To pass the test/exam  To further your own knowledge  Both

What resources are you given?:  
Textbooks  PowerPoints  Educational programs  Tests   
Educational websites  Simulations  Booklets  Human Q+A   
Other  (please specify)

If you don't have them, which would you like to be given?:  
Textbooks  PowerPoints  Educational programs  Tests   
Educational websites  Simulations  Booklets  Human Q+A   
Other  (please specify)

How much do you like being taught in the way you currently are?:  
Not at all  Its okay but not great  A lot

Do you have to be registered in some form? E.g. login online, or teacher registers with names:  
Yes  No  Its optional

How often are you tested in lesson?:  
Once per topic  More than once per topic  Less than once per topic

Do you wish you were tested more or less?:  
More  Less  Neither, I have the right amount

How much of the information learnt based on theory? E.g. talked about, but no practical work can be done on it in lessons:  
25% or less  About 50%  75% or more

Do you have any comments?:  
Have coding part of lesson u "practical!"

Current system investigation

Name: Jacob Aplin

Tick all you have/are studying:  
GCSE  AS Level  A Level  Degree  Other (e.g. Online tutorial)   
If other please specify: \_\_\_\_\_

How often do you have lessons in a week?:  
Less than once per week  1-2 times  3-4 times  5+ times

How are your topics chosen?:  
By you  By the teacher  By exam board   
By you from a list of available topics given

Why do you lean the information?:  
To pass the test/exam  To further your own knowledge  Both

What resources are you given?:  
Textbooks  PowerPoints  Educational programs  Tests   
Educational websites  Simulations  Booklets  Human Q+A   
Other  (please specify)

If you don't have them, which would you like to be given?:  
Textbooks  PowerPoints  Educational programs  Tests   
Educational websites  Simulations  Booklets  Human Q+A   
Other  (please specify)

How much do you like being taught in the way you currently are?:  
Not at all  Its okay but not great  A lot

Do you have to be registered in some form? E.g. login online, or teacher registers with names:  
Yes  No  Its optional

How often are you tested in lesson?:  
Once per topic  More than once per topic  Less than once per topic

Do you wish you were tested more or less?:  
More  Less  Neither, I have the right amount

How much of the information learnt based on theory? E.g. talked about, but no practical work can be done on it in lessons:  
25% or less  About 50%  75% or more

Do you have any comments?:

Current system investigation

Name: Joe Conway

Tick all you have/are studying:  
 GCSE  AS Level  A Level  Degree  Other (e.g. Online tutorial)   
 If other please specify: Computer Science

How often do you have lessons in a week?  
 Less than once per week  1-2 times  3-4 times  5+ times

How are your topics chosen?  
 By you  By the teacher  By exam board   
 By you from a list of available topics given

Why do you learn the information?  
 To pass the test/exam  To further your own knowledge  Both

What resources are you given?  
 Textbooks  PowerPoints  Educational programs  Tests   
 Educational websites  Simulations  Booklets  Human Q+A   
 Other  (please specify) \_\_\_\_\_

If you don't have them, which would you like to be given?  
 Textbooks  PowerPoints  Educational programs  Tests   
 Educational websites  Simulations  Booklets  Human Q+A   
 Other  (please specify) \_\_\_\_\_

How much do you like being taught in the way you currently are?:  
 Not at all  Its okay but not great  A lot

Do you have to be registered in some form? E.g. login online, or teacher registers with names:  
 Yes  No  Its optional

How often are you tested in lesson?:  
 Once per topic  More than once per topic  Less than once per topic

Do you wish you were tested more or less?:  
 More  Less  Neither, I have the right amount

How much of the information learnt based on theory? E.g. talked about, but no practical work can be done on it in lessons:  
 25% or less  About 50%  75% or more

Do you have any comments?:  
No

Current system investigation

Name: TULLY GAFFNEY

Tick all you have/are studying:  
 GCSE  AS Level  A Level  Degree  Other (e.g. Online tutorial)   
 If other please specify: \_\_\_\_\_

How often do you have lessons in a week?  
 Less than once per week  1-2 times  3-4 times  5+ times

How are your topics chosen?  
 By you  By the teacher  By exam board   
 By you from a list of available topics given

Why do you learn the information?  
 To pass the test/exam  To further your own knowledge  Both

What resources are you given?  
 Textbooks  PowerPoints  Educational programs  Tests   
 Educational websites  Simulations  Booklets  Human Q+A   
 Other  (please specify) \_\_\_\_\_

If you don't have them, which would you like to be given?  
 Textbooks  PowerPoints  Educational programs  Tests   
 Educational websites  Simulations  Booklets  Human Q+A   
 Other  (please specify) AS/A LEVEL

How much do you like being taught in the way you currently are?:  
 Not at all  Its okay but not great  A lot

Do you have to be registered in some form? E.g. login online, or teacher registers with names:  
 Yes  No  Its optional

How often are you tested in lesson?:  
 Once per topic  More than once per topic  Less than once per topic

Do you wish you were tested more or less?:  
 More  Less  Neither, I have the right amount

How much of the information learnt based on theory? E.g. talked about, but no practical work can be done on it in lessons:  
 25% or less  About 50%  75% or more

Do you have any comments?:  
WOULD LIKE MORE PRACTICAL APPLICATION

Current system investigation

The feedback from the questionnaires gave me a brief insight on how people were taught about computers. The questionnaires can only give me a brief understanding as follow up questions can't be asked, and people may lie on them or not pay full attention. However, from the questioners I can make some observations that will be useful to me, like how almost all the students that filled in the form and ticked online course also had another form of education. This shows that some students who look computing as an option also looked for more information on the same subject, this tells me that there must be something missing from the current system, that they're getting from online to fill the gap.

In the questionnaire I asked what recourses were available, after shifting through the results I've discovered that there is no 'golden' recourse, sure some are more common than others, but the recourses are different for each class/method of learning. Text books were the most common, but when someone didn't have one, they usually had a teacher made booklet. Once more, if a student was missing a recourse they seemed to want it, almost every time the students picked recourses they didn't have, as ones they wanted, no student left the 'what would you like to be given' boxes blank.

The average lesson time per week was 3-4, with over three quarters of the students having the lessons 3 or 4 times in the week, this may have been affected by who had taken it in higher education, but still the average was 3-4 times. If this is how often lessons take place, it would be how often my program would be used per week by each student.

Something that was made very clear to me was the lack of choice over topic, the students have to follow topics set by exam boards, some had a bit of choice over what they wanted to do from a set list, but non got to dwell into a topic fully from their own individual choice. However, it is clear why they would have

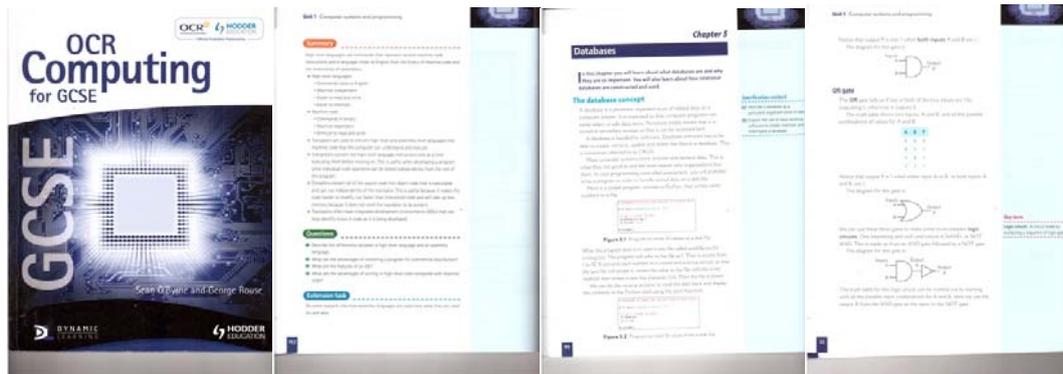
to follow the exam boards topics. When the students took an online course they seemed to have a lot more freedom over topics, some even chose their own, others from a short ('but growing' as per one comment) list.

### **Observations:**

To ensure I had some variety I observed an AS Level class and a GCSE class (both multiple times to ensure the topics being taught didn't change how the system works). At the beginning of each class a register was taken, mainly to ensure attendance was kept (but also for health and safety reasons) this was the case for both AS and GCSE, both registers were electronic, on a spreadsheet system that linked into other lessons.

The AS level class split its lesson onto two parts, one half programming and the other half theory. The theory is taught by the teacher, using a PowerPoint to assist them with the main notes they explain the topic to the students, informing them verbally while the students made their own notes next to the provided PowerPoint printed in front of them. Regularly the teacher questioned the students to ensure they understand the information along the way (the teacher appeared to be asking the struggling students more and then rereading over the info with them). Sometimes diagrams and videos are used to help explain more complex topics, for example the teacher used a video of people performing a bubble sort in a dance as this helped the student to visualize what happened. On occasion (but not every lesson or topic) a small test would take place on whiteboards just to recap knowledge, the test results are not recorded. The second half of the lesson was left for programming, practical skills on coding for the practical part of the exam. The students mainly worked on their own trying to complete tasks given by the teacher, they use resources given on the collage documents sharing website to understand how the code works, this part of the lesson is a lot more independent but with the teacher's help still available.

The GCSE class also had a programming side, but they did this once every two lessons, rather than splitting lesson time. They used booklets containing information about the language and completed small guided tasks with additional help at request from the teacher. The theory side was also similar to the AS level class. The teacher would present information on a topic with use from a PowerPoint verbally talking to the students. The students did have textbooks however, (this was different to the AS class,) after following along with the text book they would reach questions in the book to complete or to make them consider possible answers for discussion with the class. Another noticeable difference was the paper handouts, they contained some (but not all information) on real life events and examples to with computer systems that had to be filled in, these were then used as notes afterwards.



## Interviews:

Jack Scurfield-

What level were you taught computing for, gcse?, A Level? ***I took it for my GCSE class.*** Did you enjoy the lessons? ***Yes, it was enjoyable I guess.*** What kind of topics did you get taught? ***There were lots of different ones, I cant really remember them all but stuff to do with different memory, cost of them, and how they're used. Just things that would be on the test.*** Did you do lots of tests? ***Not really, we did some every now and again but I wouldn't say lots... maybe one every two months?*** So how were you prepared for the tests? Did you get lectures on the topics? ***The teacher would split the lesson into two bits, the first half would always be him talking about the topic from a PowerPoint, we had a copy of it on our screens and would follow him.*** I'm assuming you could ask him about things you didn't understand? ***If you didn't get it you could ask and he would try and explain it, but that was about it.*** ***One time when the whole class didn't get it he showed us a video on YouTube that cleared it up.*** Did you get shown videos often? ***No, it was a one off, I think it was because everyone was stuck.*** But the video helped and cleared all of it up? ***Yup, it has some extra info and the class got it after.*** So the main way of learning was presentations, and reading? ***More or less yeh.*** Did you like this method? ***It was okay, I liked having it explained by voice, but when I did revision it was a lot better to read about it from booklets and notes.*** Did the teacher keep track of when you attended lesson and what your test scores were? ***Well we had to be marked into the lesson each time, they kept track of it yeh. As for tests the scores were taken in and reviewed to make sure we were meeting out predicted grades.*** How did they take them in? was there a mark book of some kind with all the scores on it? ***He used a spread sheet to keep track of them and take them in, every test had its score taken in and compared, I was only allowed to see mine but it had my name, attendance, and test scores on it as well as my 'average mark' and last question, Is there any particular reason you decided not to take computing for As level?*** ***No real reason, I had just kinda got bored of it, it was okay, I just liked the other subjects I chose better.*** Okay thank you I think that's about it are there any other comments you want to add? ***Nope that's it.***

Noah Carr –

So are you enjoying computing for your A level? ***Its fun, but a lot of work.*** Okay, can you run me through your average lesson. ***Well, it usually starts with the teacher explain what were gonna be doing in the lesson, like what were were gonna be learning*** So the topic of the lesson? ***Exactly, we get told the topic and then the teacher talks about it, lecturing us. It not as boring as it sounds, its very inclusive, as we***

**also get asked questions by the teacher and we ask some ourselves.** So how important is the human aspect to you? Is being able to ask questions to the teachers important to you? ***I would have to say its one of the most important parts of the lesson, you can read off the PowerPoint if you really wanted to, but the way the information is explained has a huge impact on how well its understood... for me personally anyway.*** You didn't mention any registration at the beginning of the lesson, did you not have one? ***Oh sorry I missed that out, uhm, they called out of names and put us into the system, just like 'y' for in or a line for no.*** 'system'? so on a spreadsheet? ***They used a spreadsheet that was connected to the other lessons. I think it was mainly just to check we were in collage rather to do with the actual lesson.*** Did you ever get given any resources? Textbooks? Booklets? ***We weren't given them, but we have free access to the library there are some books there we can use like ones on software engineering, that sorta thing. But we do get recourses, mainly from the collages shared area, where we have loads of documents and PowerPoints etc. that have info on them I personally use them all the time for both revision and making my own notes.*** Revision for a test? ***Mainly for tests, but also just to keep it in my head.*** Do you have lots of tests? ***I would say we have few, maybe two for each section we work on, like one test would be small and teacher made, the other is bigger and for the whole topic.*** What happens if you do badly on a test? ***On the small one, nothing much, you have to make corrections and re hand it in. But on the big one you get into more trouble and have to go back on a free to slowly and carefully go over mistakes.*** I'm guessing the bigger test scores are taken in? ***I guess so, the teacher marks them onto another spreadsheet and uses them as examples in our reports.*** Reports? ***Like how good or bad we've been and how well were coping with the subject. They see if we get good grades still and if not where we've been dropping and why.*** Thankyou, and last bit of questioning, how much practical/ coding do you do? ***Roughly around one lessons worth for every two of theory, we have long lessons and about 1/3 of them is coding, but it changes from lesson to lesson, sometimes more, sometimes less. As for practical we don't really do anything, erm once in a while like maybe twice a year we get shown something, like we got shown a parrot drone, and had to discuss what it had and uses to make it fly. Does that count?*** That's what I was after, were you allowed a go with it? or was it more show and tell? ***More show and tell, it was flown a little bit by the teacher, but only to demon straight what it can do, we asked, but weren't allowed a go.*** Would you have wanted a go? Or to be able to see how they work more? ***Of course I would, something like that would have been really interesting and fun too. I considered getting one my self after seeing it.***

### **Document Inspection:**

The first document I'm going to inspect is a marked test form a student, I will be focusing on how the test asked questions, how the response is given and how marks and feedback are given. I am not interested with the topic of the test, so I will not need all the document, a few pages will suffice.

Name: Alex Murray

23/30 +10 (extra)  
33/40

1) Explain these terms:

- 1a) A constant [1] value  
A constant does not change throughout the running of a program (e.g. int value). While the program is running, it will not change.
- 1b) Assignment [1]  
Assignment is when a value is changed during a program, for a value could be assigned to a number.
- 1c) Verification [1]  
Check or a check that data has been captured correctly, method would be double check.
- 1d) Validation [1]  
Check that correct data is being entered (e.g. 0 on age) and a range check could tell if the data entered is incorrect.

2a) Give one reason for creating Pseudo code [1]

It's easy for humans/programs to understand and read.

2b) Explain how insertion sort works [2]

The program the unsorted list is sorted checked for the length of iterations needed. It then begins to insert unsorted elements into a sorted list, if the next number unsorted list is smaller than the latter ( $a < b$ ) then the number will move down until it gets sorted.

3) A client's data will have to be input into a new computerised system. Validation checks will be carried out on some of the data input.

(i) Describe a suitable validation check that could be carried out on the client's postcode, giving an example of invalid data that would be detected by this check. [2]

A format check could be used, as it would ensure that the correct format is used e.g. EX2 4LZ. It could check there are letters and numbers used, and no gaps in between.

(ii) Describe a different suitable validation check that could be carried out on the date of a client's appointment giving an example of invalid data that would be detected by this check. [2]

Range checks could be used, to ensure that the months and days are valid.  
 Day: 1-31      Month: 1-12  
 (1 to 12 for months)  
 (holds days in that month)

You had to give an example of invalid data that your checker would detect.

The test starts off with one mark questions, as shown by first page, it asks the student to explain the terms. Already it is clear to me that the human aspect of the teacher being able to interpret different answers is very important as different students could explain differently, this would/could lead to hundreds of variables all of which would be needed to taken onto account (an easy task for a human, not for a computer) I'll need to rethink my way of testing the students.

As the test continues the questions are worth more marks, it would appear to start of easy and become progressively harder (working on the assumption the question is worth more marks as it's a harder question). The feedback given to the student is a tick if correct, if wrong a small message telling them the correct answer, or what they missed out. It would be very hard for me replicate this into my program, as reading the variants of answers alone would be too hard, but to also see where marks were lost and then make corrections would be impossible. (As with all the variation I couldn't just check for keywords, I would have to make sure they were used at the correct time in a logical order, and also factor in spelling and grammatical errors.)

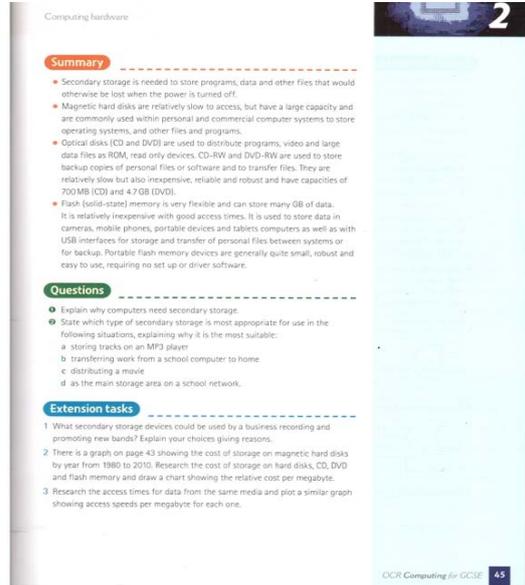
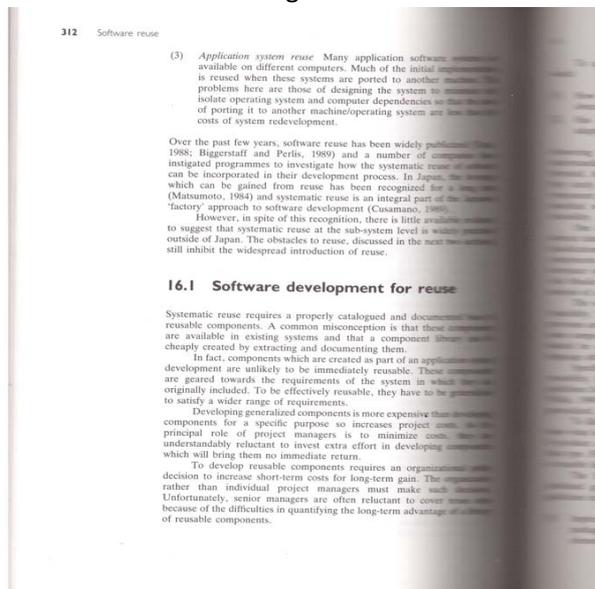
Next I'm going to look at two textbooks used to teach computing, one is for GCSE level, the other is more specialized (for software engineering.) I am looking at two text books so I can find the similarities and differences with them. The first similarity is the contents page, although almost all books have this (or something similar e.g., chapters) this will be an important part to add to my program, a sort of guide to let people know what information or resources are where in the program rather than having to search through it and find them. The text books have the topics broken down into chapters and then parts of the topics location in the chapter labeled, this helps to find the part of the topic that is needed quickly, as my program is not a book, I could use a quick menu to help its users quickly find any page they're looking for by having it setup like the contents page of the textbook.

Contents	
Preface	iv
Chapter 1 Introduction	1
1.1 Well engineered software	1
1.2 The software process	5
1.3 Management process models	17
Chapter 2 Human Factors in Software Engineering	21
2.1 Human diversity	22
2.2 Knowledge processing	24
2.3 Group working	31
2.4 Ergonomics	36
Part One Software Specification	45
Chapter 3 Software Requirements Definition	47
3.1 The software requirements document	47
3.2 System context	52
3.3 Requirements definition	56
3.4 Requirements evolution	79
Chapter 4 System Modelling	85
4.1 Validator analysis	86
4.2 Model description	71
4.3 Data modelling	79
Chapter 5 Requirements Specification	85
5.1 Requirements specification solutions	86

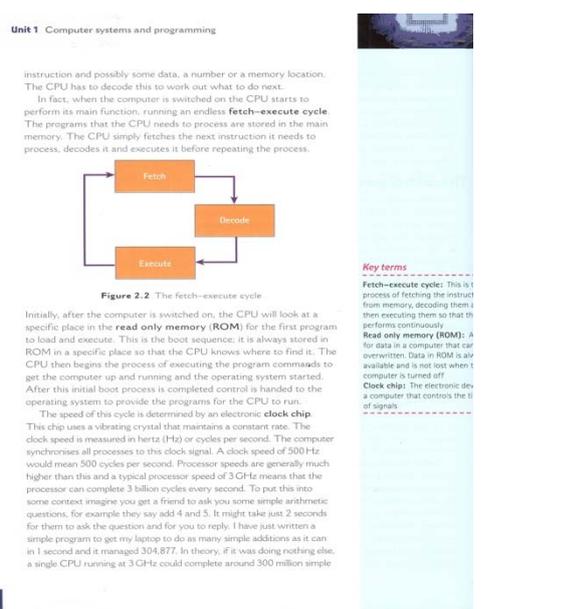
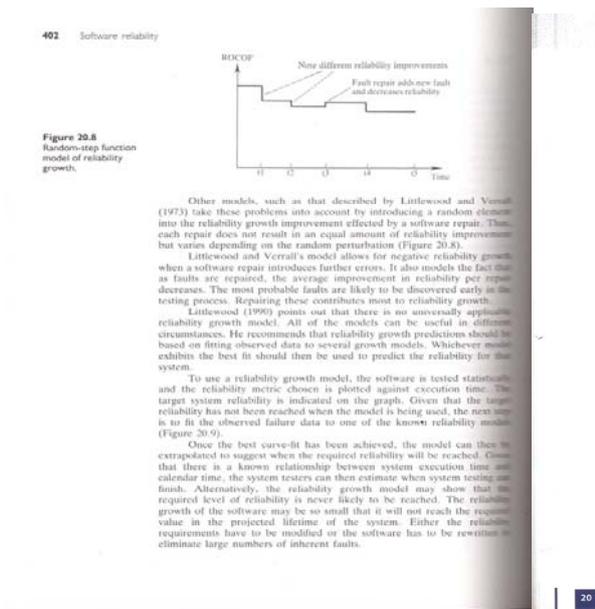
Contents	
Introduction	1
Unit 1 Computer systems and programming	
Chapter 1 Fundamentals of computer systems	7
Computer systems	7
Why are computer systems important?	16
Reliability	16
Testing	19
Standards	12
Ethical considerations	14
Environmental considerations	14
Legal considerations	18
Summary	16
Exam style questions	17
Chapter 2 Computing hardware	18
The central processing unit (CPU)	19
Memory	25
Binary logic	31
Input and output devices	36
Secondary storage	42
Summary	42
Exam style questions	44
Chapter 3 Software	47
What is software?	47
System software	48
Applications	44

The main difference in the two books is the help and guidance that the GCSE book offers, as this book is mainly used in lesson, it has questions that are for the student to think about and then ask the teacher for help if they don't understand it. The GCSE book also has a summary this is a method of 'spoon feeding' information to the student, it summaries all the the key points into bullet points for the sub topic, making it easier to remember and make notes. The book does this for every sub chapter. The software engineering textbook has a lot more detail in it, however it does not 'spoon feed' the info, most of it is given in big blocks of text and a student would have to read into all of it. There are also no questions to make the student think, only information and the odd diagram. This book is more independent learning rather than being given the information and then asking when stuck, this book forces the user to think more about what they are learning, and also prevents corners from being cut (as there are no notes to get the key point, all would have to be read through to understand it.) Out of the two text books I personally prefer the GCSE book's approach, this method of having the student read the big text, then summarize, then question seems more of a friendly and welcoming approach, if I am to

incorporate this style into my program ill have to make sure I don't 'spoon feed' the users and make them work towards the goal.



The textbooks also both used diagrams to help explain the information being given to the reader. I also suspect the diagrams help to break up the big blocks of text too as they almost always appear in the middle of a big chunk of information (not always however.) As both books have diagrams it suggests that they are a good way of learning and helping people to understand the text along side them as it's a visual guide. Both books seem to keep the diagrams as simple as possible where they can, as to not confuse its readers. After looking at these books I think I should include diagrams into my program, I will need to think of how I will make/get hold of them, and where best to place them in my program.



Bubble sort, insertion sort and quick sort exercise

Exercise 3:

Unsorted list – 25, 6, 21, 15, 98, 5, 27, 12

Bubble sort

Iteration	List following iteration
1 <sup>st</sup>	
2 <sup>nd</sup>	
3 <sup>rd</sup>	
4 <sup>th</sup>	
5 <sup>th</sup>	
6 <sup>th</sup>	5, 6, 12, 15, 21, 25, 27, 98

Insertion sort

Element to be inserted	Sorted	Unsorted
25	25	6, 21, 15, 98, 5, 27, 12
12	5, 6, 12, 15, 21, 25, 27, 98	-

Quick sort

Left	Pivot	Right	Left	Pivot	Right
			12		
			12		
			12		
			12		
			12		
			12		
			12		
			12		
			12		
			12		

Bubble sort, insertion sort and quick sort exercise

For each unsorted list, note down the steps when sorting using bubble sort, insertion sort and quick sort.

For example:

Unsorted list – 15, 3, 35, 2, 16, 7

Bubble sort

Iteration	List following iteration
1 <sup>st</sup>	3, 15, 2, 16, 7, 35
2 <sup>nd</sup>	3, 2, 15, 7, 16, 35
3 <sup>rd</sup>	2, 3, 7, 15, 16, 35
4 <sup>th</sup>	2, 3, 7, 15, 16, 35

Insertion sort

Element to be inserted	Sorted	Unsorted
15	15	3, 35, 2, 16, 7
3	3, 15	35, 2, 16, 7
35	3, 15, 35	2, 16, 7
2	2, 3, 15, 35	16, 7
16	2, 3, 15, 16, 35	7
7	2, 3, 7, 15, 16, 35	-

Quick sort

Left	Pivot	Right	Left	Pivot	Right
3, 2			7	15, 35, 16	
Left	Pivot	Right	Left	Pivot	Right
-	2	3	7	15	16, 35

Above are two pages from a handout, the sheets were examples of sorting algorithms. The handout shows a worked example of what is supposed to be done on the following pages after, and how it wishes the tables to be set out. The pages after have the tables and some information filled in, but not all (this seems to be just for the student to check they have it right pat way along.) This handout was designed to go along with what the lesson had been taught as there are no methods of how to achieve the sorting algorithms on them. This way of guiding the students to complete the tasks in the layout the teacher wanted is an amazing way of making all the students have the same correct answer, or the wrong different answer. It eliminates having correct answers in different ways. I could use a similar method with my programs tests.

Lastly I have a spread sheet from a test, the teacher marked down the grade from each topic (as one topic was on one page) and the grade from the practical coding side. They then calculated the total and gave a grade, as well as comparing if the student had met their target grade. Some marks have been colored in red as they are much lower than expected for that particular student, the teacher changed the color but the grade and total were automatically added in using a formula. The use of the spreadsheet to help mark the grades and allow the teacher to easily see how the student has done is very efficient for a small class, but this only accounts for one test and only one class, I think a new system would be needed for my program as the number of student on the program could be much greater than the size of one class, and there will be more than just one test.

Full Name	Page 1	Page 2	Page 3	Code	CA	TOTAL	MTG	
Baker, Charles	F3	C	F3	C	F7	C	B5	C (B)
Baker, Charal	F1	E	N0	D-	F2	C-	B9	C (B)
Devey, Caran	F3	J	N1	D-	F3	C	B2	B
Deveraux, Oliver	F	E	N0	D-	F3	C	B2	C (B)
Ditanev, Chise	F1	D	F3	B+	F1	D	B5	B
Evelson, Jessica	F1	D	F3	C	F	E	B5	C (B)
Franklin, Casey	F3	E	F2	C	F1	E	B5	C (B)
Fisher, Brandon	F3	C	F3	D	F3	B	B4	C (B)
Herbert, Lauren	F2	C	F2	D	F3	C	B5	C (B)
Hepner, Danielle	F0	E	F1	D	F5	E	B3	C (B)
Hooge, William	F3	C	F2	G-	F	E	B2	B
Hoodjackson, Steven	F2	D	F3	C	F0	D	B4	B
Holt, Adam	F3	C	N0	D	F4	D	B8	B
Hughes, Ben	F3	E	F3	E	F3	E	B4	C (B)
Humphreys, Liam	F	E	F2	D	F3		B7	C (B)
Matthew, John	F4		F4	C	F3	B	B1	A
McWilliams, Joseph	F3	D	F3	G	F3	D	B0	C (B)
Morales, Kham	F2	D	N0	D	F5	E	B5	C (B)
Ortiz-Rojas, Lyssa	F5		N0	D	F	E	B9	B (A)
Patten, Lucy	F0	A	F7	A	F6	A	B4	A
Reid, Alan	F1	D	N0	C	F5	E	B2	B
Reidway, Jesse	F	E	N0	D	F	E	B3	C (B)
Roper, Chase	F4	B	F1	D	F3	C	B0	C (B)
Rosary, Emily	F3	D	F1	D	F3	E	B4	C (B)
Shane, Zaira	F6	A	F5	A	F7	A	B0	C (B)
Smith, William	F1	D	N	E	F2	C	B5	C (B)
Tomas, Raine	F5	C	F6	C	F4	B	B8	C (B)
Valentine, Patricia	F2	C	F2	C	F5	D	B0	C (B)
Verich, Mollie	F2	C	F3	C	F4	B	B8	C (B)
Wash, Corina	F4	C	F4	B	F3	C	B8	B (A)
Wilson, Erin	F2	D	F2	C	F5	D	B5	C (B)

Males: 13 Females: 18

### Overview of current system:

The current classroom teaching system has a lot of benefits and drawbacks, the human aspect of the teachers is very important cannot be matched, because of this realization my system will be designed to work along with teachers, as well as independently, it will not replace them, but be a tool they can use to help them teach.

It was very insightful seeing how lessons take place and how a variety of different methods are used to help achieve the same goal, teaching the students. After using a range of different investigation techniques I believe I have been able to successfully identify where I can improve on the current system of teaching students about computer science.

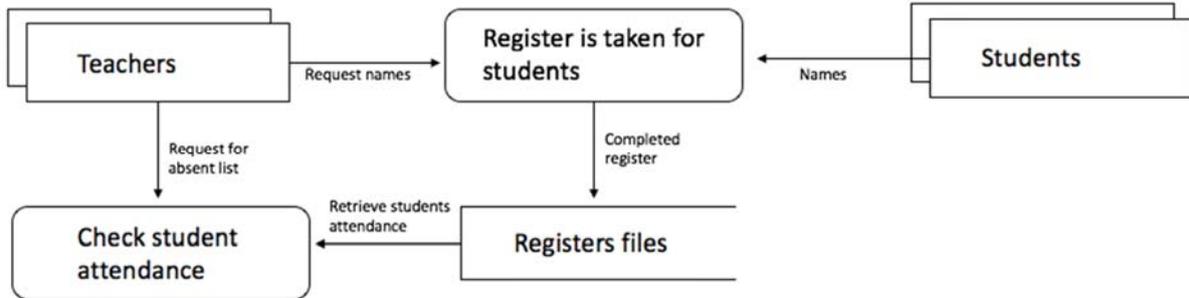
Some of the main problems of the current system generate from having a lack of resources, this was made clear from the questioners and interviews. The students were always left wanting more, if they could have it. To address this problem my programs function has now changed slightly and will aid in teaching (be used alongside teachers) rather than be used independently. I can also link my program to other resources, making it useful to students as if the content on my program is insufficient for them, they can still easily find the information they need without having to leave my program entirely.

Another drawback to the current system was the limited topics, although this was more to do with the exam boards and being taught to pass tests, I still feel like a wider range of topics would be useful for the student, I could incorporate this into linking my program into other resources, so if I don't have topic, I could still have a connection to a topic via another free and available resource.

The teachers register/mark book was another area I believe would need improvement for my system as it only works for a small class, (as my system is designed to run with multiple classes I will have to add a method of searching for the student's name, or for a suitable way of finding different students and checking if their results are up to date, rather than having to manually search through them.

**Data flow in existing system:**

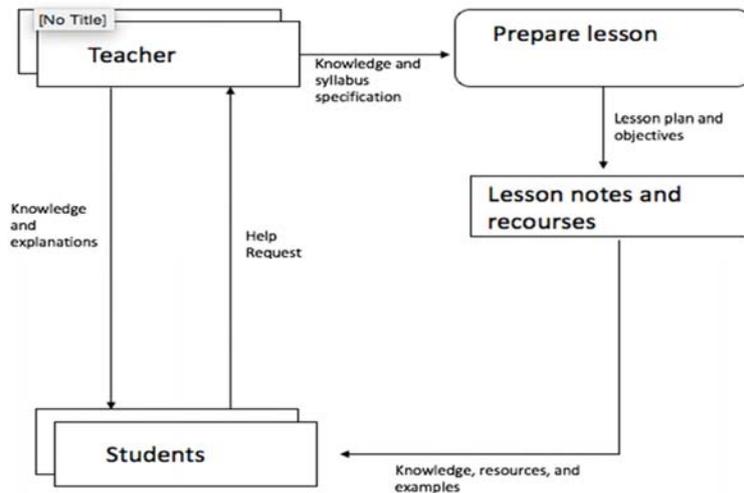
Registration:



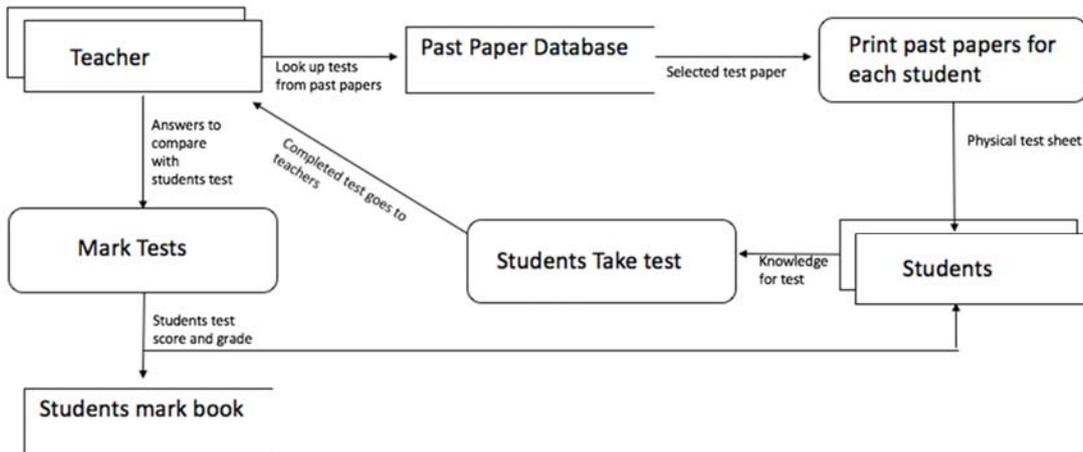
This is the data flow for the registering of the students. As the teacher calls out the students names they respond to show they are there. The teacher then adds the attendance to a file. The teacher can then search through the file and pull up a students data to see how often they have been off.

Lessons:

For preparation of lessons teachers create lesson plans and objectives, they use a database of information to obtain recourses (worksheets etc.) Theses are then passed onto the student. The students can request help of they need it.



Tests:



To create a test for the students, teachers must first look up past papers from database or filing system, they then select the paper they want and print off copies for the students. Who then sit the test. The completed test then has to go back to the teacher to be marked (this can be very time consuming as its normally around 28 students per teacher) The test scores are then added to the student's mark book, and a copy of the results are given to the student.

### Similar Systems:

In order to make my program user-friendly and to ensure it meets the previous requirements set by the current system, I will investigate the strengths and weakness of current systems in use. I will be focusing on four main systems in use: Cedar, Kahn Academy, [one more] and Code academy. I'm choosing these sites as they share similarities with the requirements of my program. They are used for educational purposes, as well as showing results, having a login for students, etc.

Brief look:

Cedar – Cedar is an education portal. Using a webpage 'portal' it allows both students and teachers to access data about their results/progress ( e.g. AS level results on results day, progress from in class tests, what lessons they have for the day)

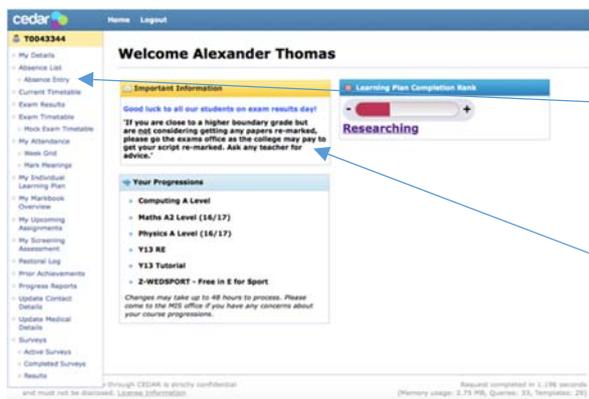
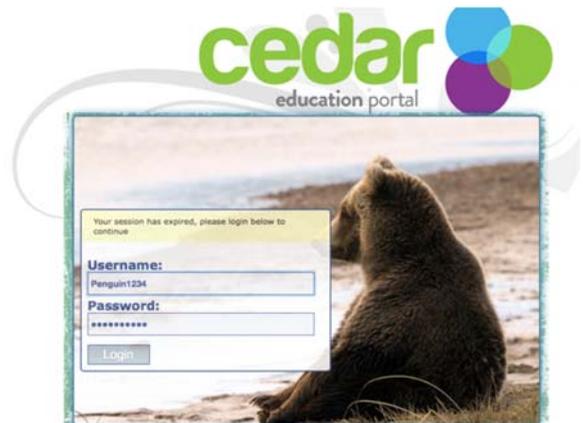
Code Academy - This website teaches its members how to code, starting with simple tasks, it adds more knowledge when a more complex task is given to the member. Small achievements are given to encourage members to do more coding and learn more.

Kahn Academy – An education based website, that was made to help others to both further their education and help students to pass exams. The website is free to use and uses a combination of text, diagrams, and videos to educate anyone who wishes to further their knowledge.

BotLogic – A small educational game that teaches its users how to use logic to moves a small (virtual) bot around to get to its home. The bot has a limited number of moves and needs to be “charged” in some levels, leading the user to decide the best/most efficient route. The game is free to play, all that’s required to start it, is an email if the user is at the age to have one.

### Cedar -

One of the first things I noticed about cedars login, is the lack of options to do anything else, there is no signup button and no ‘login later’ kind of feature, in order to use any of cedars features the user has to login, this makes since as cedar provides information on the student, such as their test results and exam timetables, it wouldn’t be worth using if no results could be shown as the system didn’t know who was using it. The way of forcing the user to login before using the system is a very good way of knowing who’s online and ensuring only the intended user sees their correct data and information.



After logging in the user is shown their homepage, from their homepage they can navigate the entire cedar system by clicking on the links on the left side of the page. Another feature that works really well is the message system, it shows the students any important information as soon as they login, meaning any notifications can easily be distributed to all students. The homepage also has some other impotent features (e.g. logout button) but most of these are common practice

The links have many different pages, from results to leaving a sick note if the student is ill, the system helps both students and teachers to keep track/monitor different aspects of their education. The systems layout is centered around the links at the side, and the design is based off ‘blocks’ as the system has a rectangular layout, everything is centered around these blocks and the information they hold. This design could be as the system is web bases, but I think its more to do with the need

- My Details
- Absence List
- Absence Entry
- Current Timetable
- Exam Results
- Exam Timetable
- Mock Exam Timetable
- Week Grid
- Mark Hearings
- My Individual Learning Plan
- My Markbook Overview
- My Learning Assignments
- My Screening Assessment
- Pastoral Log
- Prior Achievements
- Progress Reports
- Update Contact Details
- Surveys
- Completed Surveys
- Results

### My Attendance Summary

15/16

Reference	Name	Status	Last Present	YTD	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
CH2-2A-A	Chemistry A Level Block A Set 2	Present	22 Jun 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
CHP-1G-A	Computing A Block G	Present	24 Jun 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
MAT-2B-A5	Maths AS Block B Set 2	Present	21 Jun 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
PHF-1CA	Physics A Level Block C	Present	20 Jun 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Y12D-HPD-3	Y12 HE* Tutor Group 3	Present	10 May 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Y12D-TUE3-RE	Year 12 RE Set 3 Tuesday 3 Block	Present	10 May 2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Overall			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

**Results**

Above Target	0
On Target	1
Below Target	1

**Number of Assignments**  
3 (2 marked)

**Average Percentage Mark**  
84%

**Missing Marks**  
1

**Minimum Target Grade**  
A

**Overview**

Module Filter:  
All Marks  
Computing A Level (15/16)

**Computing A Level (15/16)**

Assignment Name	Date	Mark	Maximum	%	Grade	Minimum Target Grade	Info
2.1.5 Topic Test	03-Nov-2015				A	A	
Organisation of data topic test	26-May-2016	15	22	68%	C	A	

being simplistic, there for the design is simplistic too. I think the design works as its simple but effective, I'm considering using it in my program.

### Code academy-

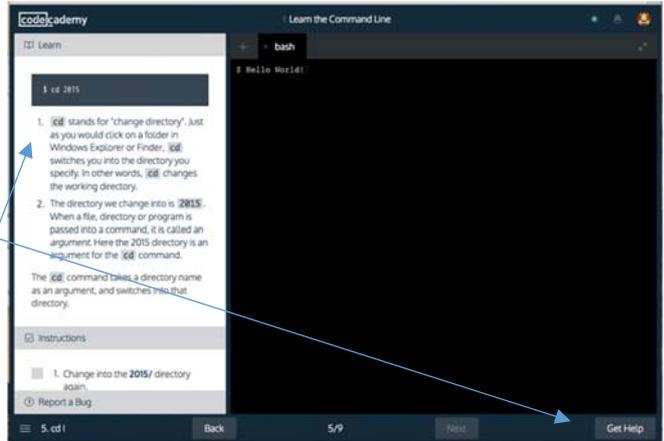
Unlike cedar this website allows users to sign up, although this was more to do with security reasons with cedar. This way of being able to sign up your self (rather than an admin doing it for you) is easier, as the admin does not have to fill in each users details one by one (a huge task for a school) but it also means the student can quickly enter all the details and be on the website faster. To make the signing up progress even easier, they have added a sign up with Facebook or Google, this would link the accounts instantly giving them email and passwords if wanted, as well as age, name, dob, etc. I think this way of quickly transferring information to start an account is a very good idea, however as my program is small and does not have the sophisticated networking or safety measures I'm unlikely to be able to use this method, however I can still make signing up as quick and as simple as filling in only a few details.



After logging in the user is reminded of their progress with the last lesson, they are shown a progress bar with the percentage of how far through the topic they are, as well as what percentage of the lesson they achieved. The student is prompted to continue the lesson via the big 'continue' button in the middle of the page. The website does prompt the student to carry on with the current lesson however by having this page almost force you to continue is discourages learning the other topics, especially is the current one has lost the students interest, they would be less inclined to go back onto the website

knowing the big 'continue' button was awaiting them. That being said, the way of keeping track of the lessons progress is very useful as it can tell the student what they have done and how well they've done it, because of this its worth considering having some form of progress bar in my program.

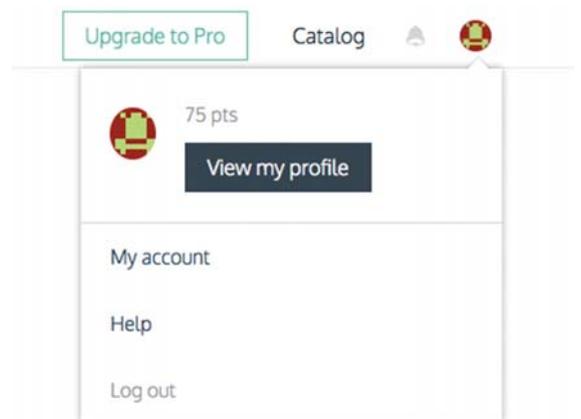
The way Code Academy teaches its 'lessons' are in a very different way to how a class would have it for exam topics, as it teaches coding languages its more piece by piece mixed in with some trial and error. However, there are still a few points that the website does well. The use of a help button ensures students needing a bit more info, or a better clue to help them complete the task is a wonderful idea, that I will use in my project as it gives guidance and support without bothering the teacher. This will save time for both the students and teachers. Another feature that took my attention was the way the information being given to the student was a small amount at a time, that then scrolled down to reveal more when the student was ready, this could be the solution to the problem the current method of downing the student with information, as it simply and easily delivers the knowledge in small easy to handle chunks, not an hour lecture and not a wall of text.



I will also be taking note of the design and layout this website, its different languages (topics) are set out as more of a visual display rather than a list. This layout makes the list seem less overwhelming and makes it easier to navigate the different topics, the user just has to click in the one they want to learn about making it easier for them to select what they want, as if it was in list format, they would have to read all the entries (normally small and close together) this design is a lot easier for the eyes. I will use this design with my

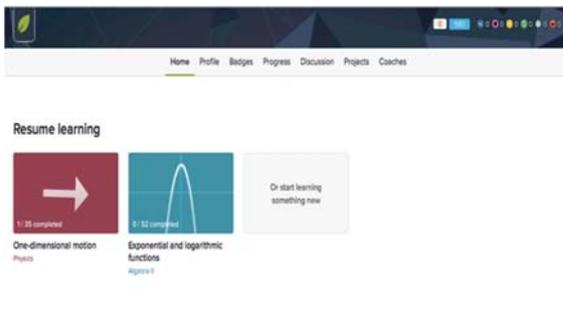
program, as this will make it easier to find a topic, as well as make the topics seem less of a task as its not just text.

The last feature that could be useful was the points system, as a sort of reward for completing lessons points are given to the user. The more points the student has, the harder they have worked. This is a good way of showing the progress of the students, it could also be shown over time allowing more detail onto the students progress. The feature is more of an added bonus and would be nice to implement, however it doesn't add any bonus to the student or their education at the moment in time, bar a higher score than someone else.



## Kahn academy –

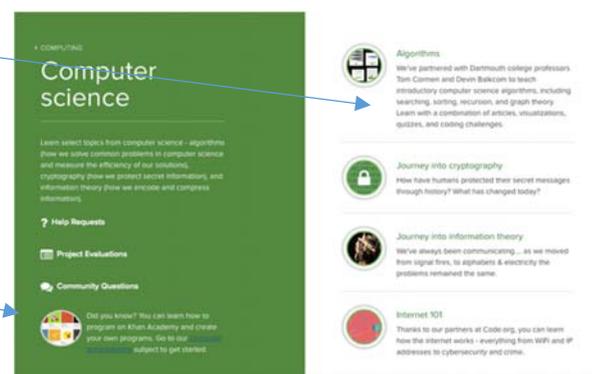
Again this system prefers the users to login, Kahn academy makes it look like the user needs to log in to be able to access the information given on it, however this isn't the case, the website only makes it look like you have to. If the user clicks on the subjects bar they can access the website normally. Although I can see why people prefer not to have to login as it does take time to sign up however I think the benefit on a website like this that can keep track of the progress you make and even recommend topics that might be useful, its worth taking the two mints to sign up. For this reason, my program will only allow people who are logged in to use it.



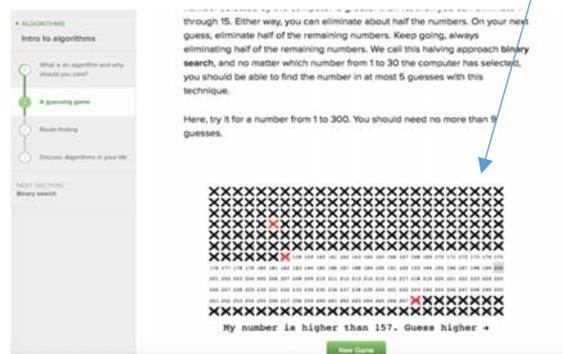
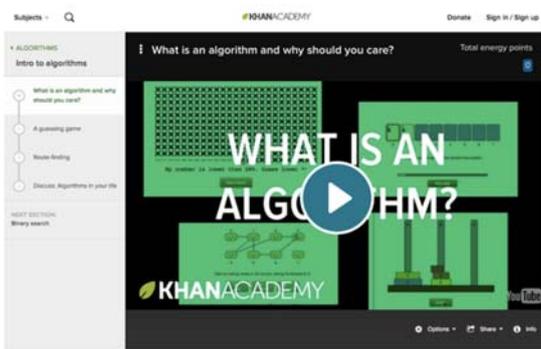
After logging in the user is given the option to resume what they are/were currently learning. It also saves what part of the topic the user was up to making it easier for the student to pick up where they left off. The layout is mainly focused on carrying on topics that haven't yet been completed. This might be worth considering adding to my project as it encourages users to finish what they start.

Once a topic has been selected the user is shown a list of sub-topics with information about them. Although I was planning to stay away from lists the use of having a brief description on the links does seem like a good idea, as it makes the student have a better idea of what there about to go on. I should reconsider how I'm going to lay out the different topics.

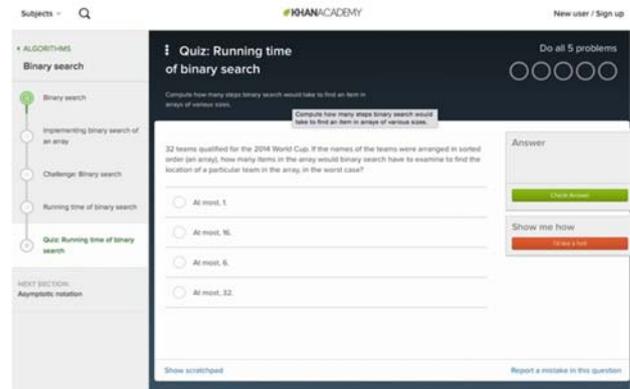
A sort of recommendation is also given to the user, the website was tiring to get me to look at the coding section, as I was already looking into computers. I could also link topics together, however I feel this could lead to having too much for the user to deal with all at once.



During the 'lessons' the website uses a mixture of videos, mini games and text to teach the students the information. I was already planning on using text and videos, but the mini games really caught my attention and I loved them, they give you a working example of why the information you are learning is important and how its used. I feel they would be extremely useful at both keeping the student interested and teaching them more than just the information about it.



The website also quizzed the students via a multiple choice test, it allowed hints and to check the answer, from what I can tell the test was more to just to refresh memory of the students, but it worked well. If I am to do a test in my program It will be multiple choice. This website has set my mind now. However, I will need to consider if I am to have the features this has, yes I will need a way of making it, but do I want to mark it all at the end? Or just one question at a time and explain why.



Questions Tips & Thanks Guidelines Report a mistake

Top Recent

So then, a 600-number array should need, at most, 10 guesses?

Edit: While using the above method.

149 Votes 19 Comments Flag 2 years ago by Deadbood

For binary search, the total iterations required to find a number would be almost  $\log_2(\text{total\_array\_size})$ .

So for an array of size 600 (assuming the array to be sorted) the easiest way to find is, calculate the total number of times 2 needs to be multiplied to get 600.

Multiplying 2, 9 times ( $2^9$ ) gives 512.

But  $2^9 < 600$ .

$2^{10} = 1024$ ,  $1024 > 600$ .

$2^9 < 600 < 2^{10}$ .

if 2 is multiplied approximately 9.xx times 600 will be achieved. Since decimal counting is not appropriate in... (more)

298 Votes 16 Comments Flag 2 years ago by vemonalachatarus

Show all 37 answers - Answer this question

I dont know how to give a mathematical argument {

39 Votes 6 Comments Flag 2 years ago by musifa

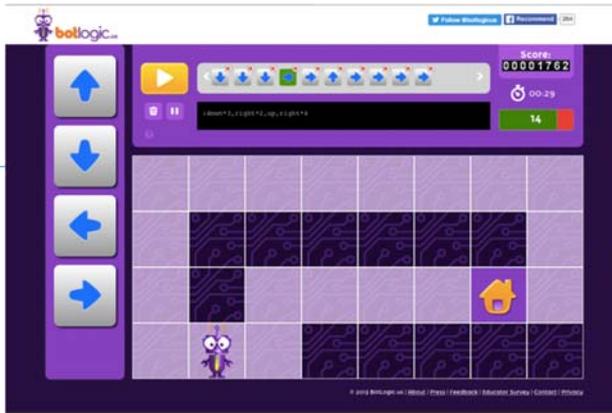
Here's one: You can't keep halving 300 more than 9 times without reaching the correct answer. These are the number of remaining values left after each guess:  $150 > 75 > 38 >$

Kahn Academy also has a comment section on each part of the sub-topic. This allows people to talk to each other, asking questions and helping others with answers. Although comments can be very useful and boost the knowledge already on the website, they can also be bad/ have bad thing put on them, yes you can ban accounts, but the comments has to be found first and manually monitoring the comments from hundreds of people can be impossible. I will need to consider both the feasibility of a comments system as well as decide if its worth adding, as it has its pros and cons.

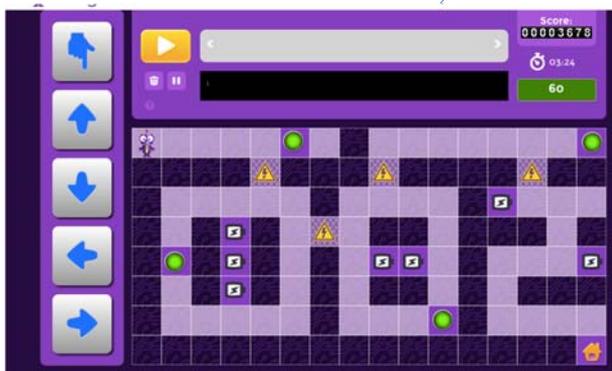
## BotLogic –

Biologic is a small educational website, its designed to help teach users about logical programming, it uses a game to teach both children and adults about logic and programming. The on screen bot is programmed using buttons (also on screen.) The user has to program the bot home to pass the level. The catch is the bot only has so much “power” and can run out if a mistake or a longer, unnecessary rout is planned. The different levels add different challenges to the user, making it harder to complete the level each time. The websites is primarily developed for children, with the characters made to look friendly and loveable over realistic, and with the sounds of the character comedic and funny, however it still appeals to adults too as the mini game is addictive and genuinely fun to play. The site uses very simplistic methods to teach the user, any new information is given by a clear text box (in the form of a sort of speech bubble) and brought on step by step.





The different levels of this logic game bring in new challenges, they use must decided how to best use the limited number of charges they have, some levels require picking up more power, or even stopping to charge the little bot on route to its home. The use of simplistic Graphics and design has shown me how valuable a strong core concept is, this small website has taken what appears to be a small game on the surface into an educational tool that can help teach beginner programmers about logic.



The game has 20 levels, they can be accessed from the start by any user, this allowed me to skip to some of the lower levels to find some more challenging levels suited to my skill level. This was a nice feature to add on as it allowed users to select the skill level they wanted, and replay levels they like.

Botlogic is a small startup educational game, it teaches its users problem solving (Via making them think of creative ways to pass the level with limited charges.) The game also teaches some basic programming skills, similar to scratch, it generators the actual code behind the scenes (for the novice users) when the buttons are pressed.

### **Users of the system:**

All the users of the systems I have looked at are mainly teachers, or students. There are other users for the systems (admins etc.) but the vast majority of users are students. This means I will have to make my project as student friendly as possible.

Students – I have looked at a range of similar systems, the users age in the systems varies. The students ages ranged from 3+ (as there isn't really an age limit on the sites.) However, I believe I can increase the age limit from 3 to 10+ for my system, as some of the other systems I looked at (BotLogic) were made for the younger age group I mine. I decided to increase my minimum age to 10 to ensure any students using it can easily read and use the system without the need of help. The difficulty of the actual content is decided by the teachers, so I haven't included this factor in my age rating.

Teachers – My system will rely on teachers using my system to input content for students. The teachers will have to input their knowledge to produce tests and lessons on topics they wish to teach the students. The project is aimed at teacher in a classroom (where I presume they would be technically able.) However, my project is also aimed at other educational places outside a classroom (such as an interactive museum) therefore I'll be making my project easy to use when on the teacher side of it, as to allow for both experienced and novice users.

Admins – My project will have three levels of access, for security purposes. The idea is such that only the admins (a very small percentage of users) will be able to see all the features and data regarding other users. The admins of any system should already be capable to use simplistic systems, but as I have no control over who the systems users will give admin privileges to. I decided to make the admin side and options really easy to use. As it was better to oversimplify than overcomplicate.

Overview – As I have no way of controlling who will use my system after I release it, I will have to ensure that anyone with basic skills can use my system, I shall allow some lenience with the admin side as the admins should be slightly more capable than the average user. However, the system should be easy to use for any user.

### **Requirements of users:**

- Admin – The admins will need to have complete control over all the system, they should be able to check what the student is up to, as well as change anything is something is incorrect. The admin needs to be able to view all the data in the system, everything including all the user's data and have the ability to amend it. So the admin will have all the following features as well. Only Admins should to be able to:
  - View and edit user's data
  - Use off of the program to its full capability
  - Access all the data inputted and outputted from the program.

- Teachers – The teachers of my system will need to be able to type text into a pre given space, in order to create the lessons for the students. The teachers will also need to be able to create tests, I will make this as easy as possible for the teachers, but they will still have to think of the questions themselves and type in the correct answers. Teachers will need to
  - Create test for students
  - Fill in “lessons” containing topic information for students to use.
  - Review the students’ progress
  - Access all the same features as the student
  - Change/add to other teacher’s lessons
- Students – The students are the main user of my program, I will have to ensure they can easily and quickly carry out their needs. The students need a system that they can use to help teach them content and information regarding their tests and personal knowledge. They need a variety of methods to help them to this, as well as having a way to be recognised by the system to see how they personally are doing.
  - Display data and information to the students to help them learn
  - Give simulations to help explain topics in more detail and more easily
  - Allow the student to log in / sign up (create and identity to be used by the system)
  - Take tests on different topics
  - Have the tests marked and saves (with answers cleared)

### **Similar systems overview:**

After looking at some similar systems, I have rethought some aspects of my programs design. For example, I have decided I’m going to force the users of my program to login before using it, after seeing how cedar and code academy enforce logging in to make sure the program knows who’s using it. There are other features I’m going to add bases on my findings of the similar systems:

- A quick menu system – Cedar made me realise how useful jumping from page to page is, because it a lot easier to navigate.
- Scroll bar to break up text – After seeing how code academy, and Kahn academy split the text into smaller parts, to make it easier to read and easier to understand. I will use a similar method to split the text containing the information I have.
- Topic buttons – On the main menu the buttons for the different pages will be more similar to icons and contain a small amount of information on the topic.
- Mini game simulations – after seeing Kahn academy’s mini game that educated the users on how the code can be put into use in real life. After playing BotLogic’s game, that made me realise how valuable interaction to the user is, keeping users entertained is a key part to teaching them. decided to add at least on into my program.

### **Users and clients:**

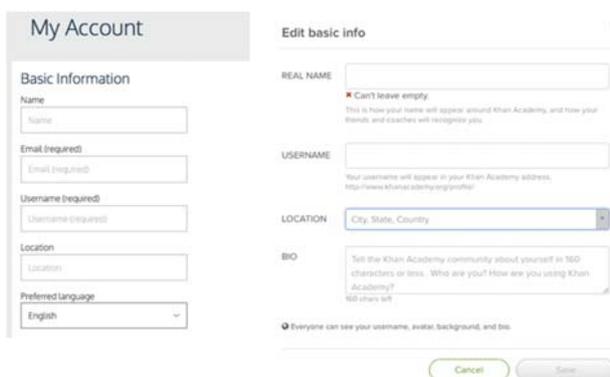
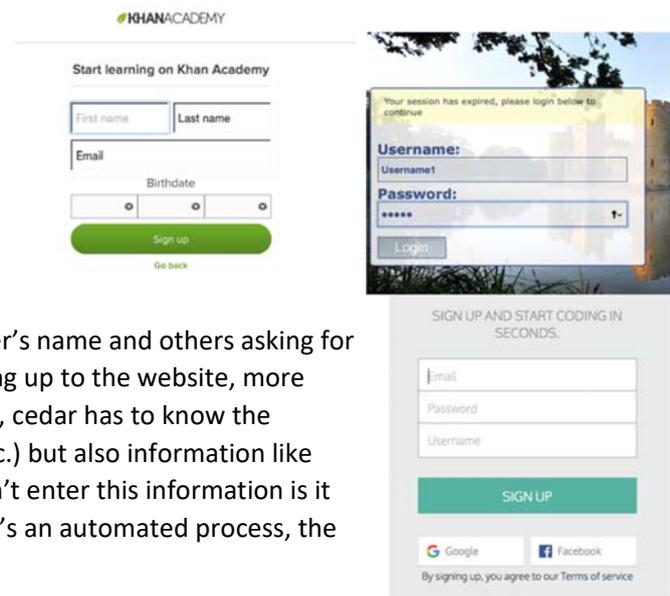
Client – My program is likely to be bought or downloaded for by schools and colleges, it also has a potential market place at interactive museums (or similar places) but my main focus are schools and

colleges. Its aimed at the teachers and students as an educational assistance tool that would help the teachers with their job, and be more interesting for the student.

Users/Costumers – The users of the program are students and teachers, both would provide information to the system and then use it. The students sign up with their information and are then provided a login, after they can access the information on the program to learn from. Teachers have to sign up the same as a student, however they require and admin (admin pass would be given to the programs user upon setup) to give them teacher level access.

### Current systems data input:

All the current systems (apart from BotLogic which still can require an email) require the users to login, or sign up. The login for all the systems are the same, they only require either an email or username, and then a password. Signing up is a little different, the websites ask for different information, they all require an email and password. With some asking for the user’s name and others asking for the date of birth. However, this is only upon signing up to the website, more details can be/ will be asked later on. For example, cedar has to know the students name, contact details (phone number etc.) but also information like their age and address, although the student doesn’t enter this information is it still entered at some point by the collage even if it’s an automated process, the information is inputted.



More forms from the websites about changing details, most of them request the users name, email, username (if needed), and home address or location. Other info was asked for but it wasn’t necessary, such as links to other websites and information about the person (profile info.)

The form is divided into several sections:

- About Me:** Includes a text area for 'About Me', a 'Website' field with 'http://' pre-filled, and social media links for GitHub, Twitter, and LinkedIn (all with their respective names pre-filled).
- My Contact Details:** Includes fields for 'Mobile Telephone Number', 'Email Address', 'Home Telephone Number', and a large text area for 'Home Address'.
- Emergency Contact:** Includes a section for 'Emergency Contact details, if different from parent/guardian's contact numbers above.' with fields for 'Name' and 'Telephone Number'.
- Additional Information:** A section for further user details.

The main information that has to go into the program seems to be:

- Name of user
- Email
- Password

Other common information tends to be:

- Date of birth
- Location
- Other form of contact

The rest of the input from the system differs a bit, in a class room the teacher takes a register marking people either in or absent. They enter this information into a spreadsheet (normally, but can be done on paper.)

Full Name	Mon	Tue	Wed	Thu	Fri
Baker, Charlie	/	/	0	/	/
Blakey, Chantal	/	/	0	/	/
Davey, Cavan	/	/	0	/	/
Dewhurst, Oliver	/	/	0	/	a
Dillaway, Chloe	/	/	0	/	/
Evedon, Jessica	/	/	0	/	/
Fearnley, Casey	/	/	0	/	/
Fisher, Brandon	/	/	0	/	/
Herbert, Lauren	/	/	0	/	/
Hester, Danielle	/	/	0	/	/
Hodge, William	a	a	0	a	/
Hodgkinson, Steven	/	/	0	/	/
Holt, Adam	/	/	0	/	/
Hughes, Ben	/	/	0	/	a
Humphreys, Liam	/	/	0	/	/
Mathew, Joan	/	/	0	/	/
McWilliams, Joseph	/	/	0	/	/
Moreau, Kham	a	/	0	/	/
Parriss-Riley, Lydia	/	/	0	/	/
Patten, Lucy	/	/	0	/	/
Perrett, Kiera	/	/	0	/	/
Rodaway, Jade	/	/	0	a	/
Roper, Chloe	/	/	0	/	/
Rowley, Emily	/	/	0	/	a
Sheta, Zeina	/	a	0	/	/
Smith, William	/	/	0	/	/
Towers, Rachel	/	/	0	/	/
Valentine, Patrick	/	/	0	/	/
Veitch, Mollie	/	/	0	/	/
Welsh, Connie	a	/	0	a	/
Wilson, Erin	a	/	0	/	/

Here is a register for a week of a computing lesson, the teacher has marked students who have shown up for the lesson with "/" and students who were absent with "a" on days when the lessons went held the register is marked with an "0" this is done automatically.

Although taking a register is part of the class room teaching system, my program will run independently of it. My program does not need a register to be taken as it only works if the user of logged in. However, one of the reasons a register is taken is to ensure the student is doing required amount of work, I could have a part of my program check for if the student has last been online (but as its designed to help teach, not replace it, there's no guarantee it would be used every lesson.)

Another input the teacher currently does is adding the students grades into a spreadsheet, this is done automatically on the websites, but it still a form of input. The mark from each part of the test is added and the according grade. The need for inputting the information after is removed if the test is in my program and marked in it as well.

The spreadsheet displays columns for student names and various test scores (Part A, Part B, Total, etc.) and a final column for 'Grade/Bonus'. The data is organized into rows for each student, with some rows highlighted in yellow.

The last main part of input comes from the tests of the websites, the multiple choice tests allows the users to pick one of four options, the radio button choices allows one answer to be selected, and makes marking it easier, once marked the score is uploaded to the students account. The the only input from the user is clicking on the label/radio button containing the correct piece of information.

Q Search for subjects, skills, and videos

The above search bar is also used in the websites to allow the user to easily find what they need to, this is a text base input that looks for keywords and only shows the user the matching words.

Data	Use, format, and field
Name	Used to identify the student, would be a string.
Email	Could be the login used for the student, saved as string
Password	Used to login and for security reasons, saved as encrypted sting.
Date of birth	Could be saved as string or on DoB format (dd/mm/yyyy)
Register input	Used to keep track of attendance, could be saved as symbols in a string or as Boolean.
Score/test results	Numeric value that is then converted into a grade, saves as numeric.
Answers to test	Radio Buttons to be clicked, saved as Boolean (correct or wrong)
Search Bar keyword	Word for phrase entered then compared with a list for similarities, saved as string.

All the current systems were based with a GUI, all the systems used graphics, buttons, and text to communicate with the user. None of the systems were purely text base, they all involved graphics of some shape or form. Most of the graphics used were there to make the site/program look appealing, with the intention of keeping the interest of the users.

This approach leads the sites to a more “button” based pages, with almost everything being controlled by buttons rather than text (some obvious exceptions e.g. Search bars) But most of the tests were check boxes rather than having the user type in their answer, the pages navigated via linked buttons, opposed to hyperlinks (although hyperlinks are still similar to buttons.)

## System Output:

One of the outputs given by the system was the score from tests, here's cedars test review, it had a grid view of the tests and what mark/grade the user got.

For this to happen the system has to look up the students score from different record and then enter the information into the grid, this method involves opening different documents and extracting data from their contents.

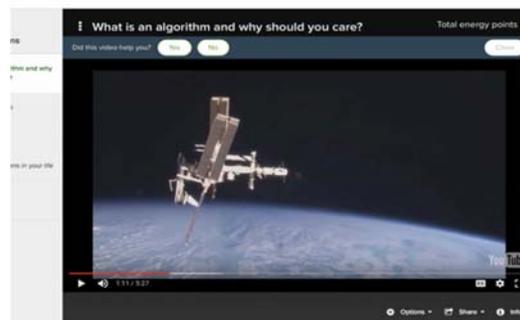
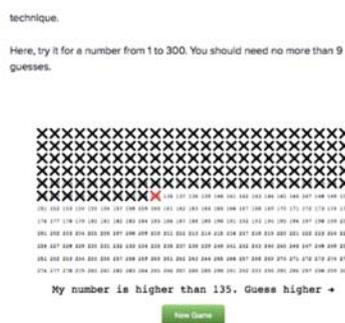
## Overall Grades

Name	Board	Season	Level	UMS	Max Marks	Grade
MATHEMATICS	Edexcel - GCE	Summer 2016	AS	261	300	A
CHEMISTRY A	OCR	Summer 2016	AS		0	B
PHYSICS AS	AQA	Summer 2016	AS			B

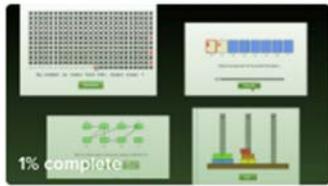
## Summer 2016

Examination Name	Board	Level	UMS	Max Marks	Grade	Resit
DECISION MATHEMATICS 1	Edexcel - GCE	AS	86	100	A	
CORE MATHEMATICS 2	Edexcel - GCE	AS	82	100	A	
CORE MATHEMATICS 1	Edexcel - GCE	AS	93	100	A	

The output of the knowledge was given in different forms, sometimes as a video, sometimes as text, sometimes as a mini game. In order to play the video, the website was linked to YouTube, and played the video off it (the video was not from the website.) The process of having the mini algorithm game was simple once the algorithm was made, it appeared to have just been a visual representation of how the algorithm worked, just shortening the potential list after the human user takes a guess. In short a mixture of an algorithm being displayed via some simplistic graphics. The text was displayed via a scroll board, the process seemed to be just to load up the text into the scroll bar.



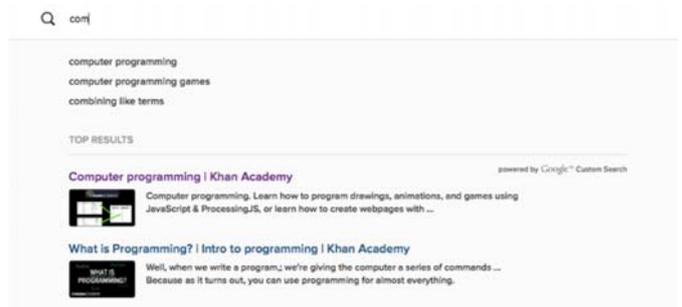
Another output is the completion bar, it tells the user what percentage of the topic has been done, and also gives them an indication of how much more work they need to do. The process of this involves the system checking what pages have been read and what tasks have been done, and then calculating a percentage from the total number of tasks/pages in the topic.



**Algorithms**  
Computer science



The output from the search engine is given as a list, the search engine is powered by google, so there is a link on the systems website that connects itself and then conducts a refined search, using google.



**My Attendance Summary**

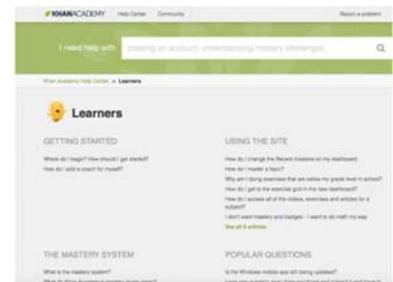
Reference	Name	Status	Last Present	YTD	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
CHE-2A-A	Chemistry A Level Block A Set 2	📄	23-Jun-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
COMP-1G-A	Computing A Block G	📄	24-Jun-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
MAT-2B-AS	Maths AS Block B Set 2	📄	21-Jun-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
PHY-1C-A	Physics A Level Block C	📄	20-Jun-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Y12D-HH1-3	HY12 HE* Tutor Group	📄	05-May-2016	100%	100%	100%							100%		
Y12D-TUE3-RE	Year 12 RE Set 3 Tuesday D Block	📄	10-May-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Overall		24-Jun-2016	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

For the classroom, attendance is kept, after being marked down by the teacher the attendance is entered to the systems database, it then added to the student's previous attendance. A percentage, of the lessons attended is then calculated and displayed for each month.



appears to be as the users can access it easily for reference. With a video they might need to listen to it, or have to watch half of an unrelated topic to get to the part they need, but with text, they can easily find the part they need and read it. Some users highlighted text when copying down information and even copied and pasted the text into a document to use later on. For the systems I have looked at, the websites seem to hold the text/information in a database, they call upon the data to be opened when the corresponding page is opened.

- Search Results – The sites that did have a search function included, had to send out the results of the search to the user. Some of the search engines, (Kahn academy) were powered by google. They used an outside source to search for the result. Google uses a special algorithm for its search results, with Kahn academy the algorithm only includes the necessary results. Paper based systems, schools etc. use filing cabinets, the documents and recourses are placed in alphabet order and then a human performs a search in the right area. In both situations an output is given, the search engines give out the possible results the student could be looking for. The paper bases system gives out a possible document in the correct range. For googles search engine, if the name of the requested page was entered incorrectly (spelling error/wrong name) googles advanced algorithm can likely still find the correct result, whereas in the paper based system is a lot less likely to have the correct end result if the search was wrong.
- Help – The sites all had some form of a help button. Although some were a lot better than others, they all had information to help the user in using the system. The help output was always given in a easy to understand format. The most common was to read a question and then click on it for an answer (the idea being the user clicks on the question they have.) Other systems had a tutorial, which went through the system step by step showing how to use it, with annotations on screen, this was a much more graphical output, but it was also more interactive.



### Limitations of current system:

Although the current system of teaching works well, as students get grades to go onto further education, there are some limitations with the system. One of the points I picked up on from both the interviews and observations, was there is very little to none, actual interaction with anything physical. Despite computers being used in almost all devices used on a daily basic. I think having something physical and actually seeing with a student's own eyes how computer can be used to interact with day to day devices would be a better learning experience for the student.

The texts books had a limitation of their own, even with an index and a contents page, actually finding one bit of information from the sea of text was still had, as you had to find the page, and then read the big chunk of text in front of you till you found it. This added effort deterred students from using the books as a common reference, and instead mainly used it to follow the teacher while the teacher talked about a topic in it.

Even some of the websites had limitations, for example Kahn academy's multiple choice tests were left very easy to cheat on, once you stated the test you could take all the time you want researching the

answer, (it was easy to google it) this didn't really effect Kahn's system as the tests were more for the student's own sake, (the score is not seen by anyone else) but for a teacher seeing the test results this could be very misleading as to the student's progress.

Although most students are happy with their current way of being taught almost all of them wanted more knowledge, more resources. This tells me they want more as they don't have it, a sort of 'can't complain' situation with what they currently have, part of it is access to the programs and recuses provided by schools and collages as most 'official' exam board content costs, whereas the free information online isn't always relatable to the exam.

Keeping the students interested was also a problem that came up, as interesting as ethics of computing is, some students don't always like the topic, losing interest and therefore not working as hard. This is a more subjective limitation though as not every student would do this. Different methods of teaching the student could keep their interests for longer.

The biggest limitation on the websites have the be the lack of ability mark, more complex answers. As there can be a ridiculous number of variations, to the same answer for a question, the making process has to either cut corners and look for keywords, or make the answer pre-selected. When a solution like searching for keywords is used, it doesn't always correctly mark the answer as spelling errors can mess up the program, in short, it's very hard for a computer to mark a human answer more than a few words long.

One of the smaller limitations on the websites was creating a user name that wasn't already taken, the user could type in one, but if it was taken the website would then make the user think of another, without letting them what variations of the user name was also already taken. Although the username had no effect on the use of the site, this was an annoyance when signing up and could potentially deter users from following through with creating an account.

#### **Specification and justification:**

- Way of logging into program – The program need to know who's on it, this is easily done if users are forced to login before use, it makes it easier to track who's done what on the program.
- Way of signing up to program easily – In order to not be an annoyance signing up should be as easy as possible. The quicker the student can sign up mean the quicker they can start working.
- Way of changing data by admin – in case data is added incorrectly or in case the data of some part of the program changes the admin need to be able to edit it and change where appropriate, this ensure the program is up to date and the old incorrect data isn't wasting memory.
- Only get certain people access the correct info - it's very important that the data is restricted so that only the people who need it can access it. This stops email addresses and other important information from being taken.
- Secure way of encrypting user data – The need to encrypt the data is also important, personal details like emails, names, etc. need to be stored securely, or I would be vitiating the data protection act. So I will make sure the users data (as well as other data) is encrypted.

- Page with topics to choose from – The students will want to see what they can go on when using the program, a main page with lots of topics on it is the main way I have seen other systems do this, it's also a lot easier on the eyes than a list (it looks less intimidating.)
- Way of quickly navigating the program – The program will have a lot of different pages and forms, a quick way of navigating to the one wanted is key to making the system easy to use, it also means people are less likely to get annoyed or frustrated with the system as they can't find the page they want. It would save time and effort if I added in a quick menu for navigation.
- Way for teachers to check on student's progress - A form that the teacher can look up and check on the students score from tests, this would mean the teacher can tell if the student has done well, or need more support.
- Tests for students with multiple choice – a way of testing the students to make sure they are leaning the topics. The reason it will be multiple choice is to make marking them easier. As I have explained it can be hard to mark other forms of input. It also leaves the possibility of teachers creating the tests themselves available.
- Information for students – Information for the students to learn about and read about will be needed, this information will be the main reason why the program is used.
- Range of format for the info – in order to keep the program from getting boring and losing the students interest, different ways of proving the information must be given, this can include linked videos, minigames, etc.
- Easy to use layout – The simpler the design in the better, if I make the design simple and intuitive the users will be able to pick up how to use it easily, but I made It too complex people would be less likely to want to use it, never mind leaning how to.
- Easy to use for admins and teachers – it's important that the teacher and admins can also easily use the system, as they might not be used to using educational programs( as from the survey most schools and collages didn't) The easier I make it for them to use the more likely they will use it.
- Include 'physical' component – Most places didn't have any practical/physical item to work with, they only seemed to talk about the potential application of it. It would be more interesting and a better experience to have something they can use and see how it works.
- Must be able to be used alongside the teacher – As my design has changed slightly I know understand its vital to keep the human aspect of teaching, the program need to be a tool that can be used with the teacher, rather than replace them.
- Must prevent cheating in the tests – As the students will be learning the information given, it would be pointless to have a test they can cheat on, as cheating isn't allowed in the classroom tests, I will need a way of blocking it in mine.

#### **Technical Specification and justification:**

- Programming language – As I have decided my project will have a physical component (the bot.) I'm going to use two different programing languages. The main program will be written in Visual Basics (VB.) VB is an event orientated language, (when a button X is pressed it does Y) this would allow me to easily connect the actions of the program together, making an event system that works by the user clicking on buttons, or selecting options, rather than typing in commands. Another reason for choosing visual basics was the IDE it comes with. Using visual studio, I can

easily generate buttons into my program, I can quickly drag and drop text boxes and other useful objects into my program without having to manually write the code for them and create graphics for every different objects. There are other benefits to visual studio, I'll be able to use breakpoints in my code, as well as being able to compile and run it without having to leave the IDE.

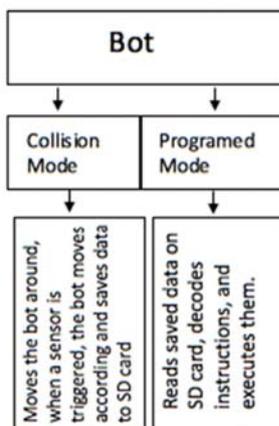
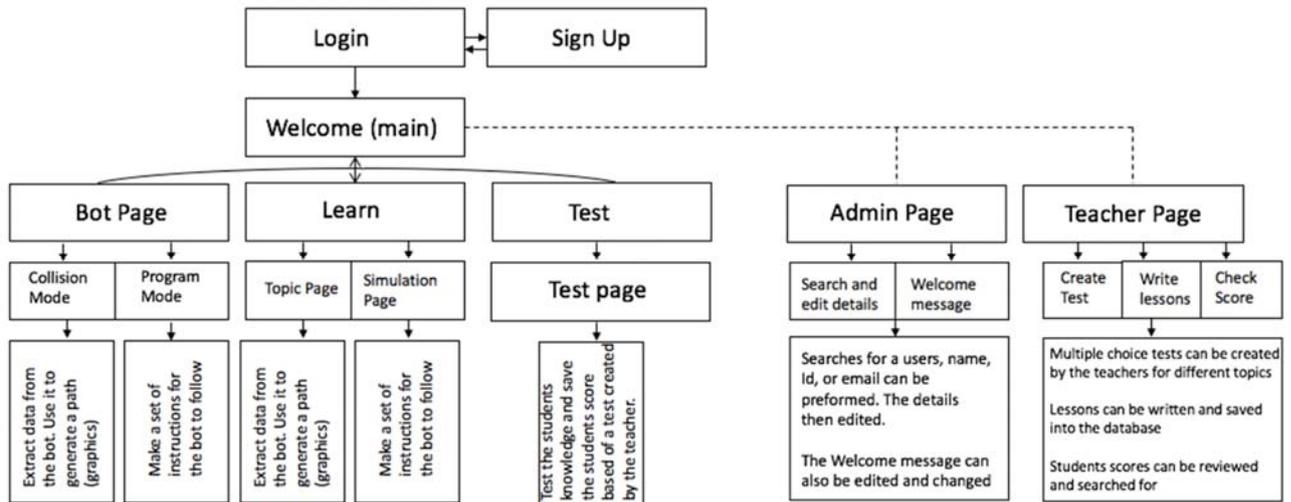
- Bot – For the physical side of my project I will make and program a small bot. The bot will require separate hardware as well as software from the main program.
  - Microcontroller – The bot will need a microcontroller to act as a brain and run the code for it. I have decided to use an Arduino Uno for my bot's microcontroller. I chose the Arduino Uno as it's a relatively small but powerful controller, It can also give out and receive analogue inputs and outputs which allows it to interact more with real world items/objects. (Unlike my backup option the Raspberry PI)
  - Language - As I will be using an Arduino I will use the language the Arduino sketches are written in (C/C++ but minor changes are made.) The language allows me to program the hardware and set the pins of the Arduino to input and output signals based off the code. This allows me to collect feedback from a switch, or tell motors to move. The language is relatively easy to learn, but a bit more complex than VB, however with it being used by a wide range of people, there is power hidden in this language.
  - IDE - Arduino is open source; it offers its own IDE. Admittedly this IDE does lack some of the features that VB offers, however the language is more suited to for the needs of the bot.
  - Memory – One of the limitations of the Arduino will be its memory. It has a limited memory available that will be used up mostly by my code for it. As when saving logged data, I will be taking up more memory, I'll need to add an external storage. For this I will need a micro SD card, in this way, I can switch out the SD card when I want to. I can also use whichever size of memory I want, (so long as I can acquire the SD card of said size, but you can easily get a relatively huge size memory easily)
  - Chassis – The bot will need a body to hold all of the components and attach the wheels. For this I will be using a very cheap body from online (a basic chassis for bots and other projects.) Made from plastic, it's both lightweight and relatively durable. This will allow me to hold the bot together. If I have the time and can acquire the resources, after the bot and program are complete, I could make a cover for the bot, a sort of lid to hide/protect its insides.
  - Motor control – To move the motors I will need to supply them with more power than the Arduino normally gives out. So in order to power the motors, I'll need to use a motor controller powered by a separate power source. The motor controller will allow me to quickly and easily, with the use of the motor controller I would be able to program the bot more easily (as I would have to otherwise redefine the pins each time after using the motors)
- Search Algorithm – For my program I will be storing data that holds educational information (written by the teachers.) With the teachers having no limit to the number of lessons they could write (as I didn't see the need to limit knowledge when it takes up so little space, stored as text), I decided to use a binary search to locate correct lesson from the sea of records. This will drastically reduce the number of searches the program will have to do in order to locate the

correct one, as binary search essentially cuts down the possible selection in half each search, making it quicker than other search algorithms for a big number of records. I will have to make sure that when each record is stored, that its stored in alphabetic order (as the binary search requires it to be in order)

Objective	Success Criteria	Performance
<p><b>The program should be able to hold information for a range of different topics, with the teacher being able to add new topics as and when they see fit. The data should be accessible by the students, and not take a long time to load.</b></p>	<p>A questionnaire will be sent out the teachers and students (a different one for each) this will allow feedback for what should be a clear “worded / didn’t work” situation for the different parts. I can read from the data files to test if the entered data was saved.</p>	<p>The system must be able to save multiple and different forms of data (text) written by the teachers. The students must be able to access this data quickly and easily.</p>
<p><b>The program should be interactive. (such as simulations the user can try) Too much of the current system and old system was ‘boring’ The new system will need to grab the attention of its users and keep it making sure its interesting and fun to use.(not a pain/drag)</b></p>	<p>As this is more subjective and differs for the individual, I will need both a larger range of responses and more detail, therefore a questionnaire will be sent off primarily with follow up interviews on a selected group. I can test the simulations to ensure they output the correct result.</p>	<p>Users should find the system fun and intractable, with features that react to them, instead of pure text base lessons. Simulations or demonstration should be useable and provide a correct output.</p>
<p><b>The system must be easy to use. As some of the users may not be experienced with computers, or this type of system. It should be easy to pick up and navigate the system, as well as intuitive.</b></p>	<p>For this a yes/no question can be asked in the questionnaire, (students, admin, teachers.) But I would also benefit from an interview, or watching a user use the program without instructions so see how they use and react to it.</p>	<p>All users should have little to no problems when using the system and all of its available features. The system will need an appropriate UI that’s easy to grasp.</p>
<p><b>A way of testing the student’s knowledge will be multiple choice test, theses tests need to be able to be created by the teachers and taken by the students, with the results of the test saved. The location of the answers to the test should also be randomized each time to prevent cheating.</b></p>	<p>Data can be pulled from the system to ensure it’s been saved correctly. The randomization will have to change the location each time, this will have to be checked manually. Ill send out a questionnaire for the tests, to ensure there are no problems/ glitches for the students. But ill interview a teacher to see it the system works for them.</p>	<p>The test will load up and give possible answers in a randomized layout this will stop cheating. The test will load up completely and correct in regards to the questions and answers provided by the teachers before. The Tests should also output a mark which</p>

		will be saved, and answers color coded.
<b>Topic lessons should load up relatively quickly when requested. The number of topics should not drastically increase the time to load up one of the topics.</b>	Timed tests on a file filled with lots of 'sample' topics will be taken. The test will compare with the real life system (file cabinet/handing out sheets/opening a power point.)	The new system should be quicker than the old system when loading up lessons. It must be faster than the old system, and not be inefficient.
<b>Admins and teachers will have special access that the students don't. They will be able to check and change data stored by the program as and when necessary.</b>	All data regarding students will not be shown to other students no matter what. Only teacher and admins will be able to see and edit the data. This will be checked for by looking over all the forms after and during use.	Only teachers and admins will be able to change information regarding the students. Students wont be able to asses other students data (to read or write).
<b>Sensitive/private data stored in the program must be kept safe and secure, so no unauthorized personal can access it.</b>	The secured data stored must be unreadable to humans when accessed without permission. This will be checked with sample data.	When the program is running or not, all the secured data will be encrypted and unreadable to everyone.
<b>New users should be able to easily join without hassle. Already existing users should be able to login easily, using a secure password and username.</b>	Users will provide feedback in the questionnaire and rate the login section based of ease to use. I will then be able to evaluate if its effective enough.	The system must be able to save the information entered, and not edit it unless instructed to. The data must also be readable to check the user has logged in.
<b>Test will be able taken by the students, and then marked automatically. The teachers should be able to view the results from the individual tests.</b>	Teachers and students will take questioners with questions on the testing side of the system. The test will also be given 'sample data' that should be marked correctly. Testing regarding the data is saved correctly will also need to be performed.	The student should be able to load up a test, take it, then have it marked and the score saved to the database. The teachers should be able to view this score and compare it to others.

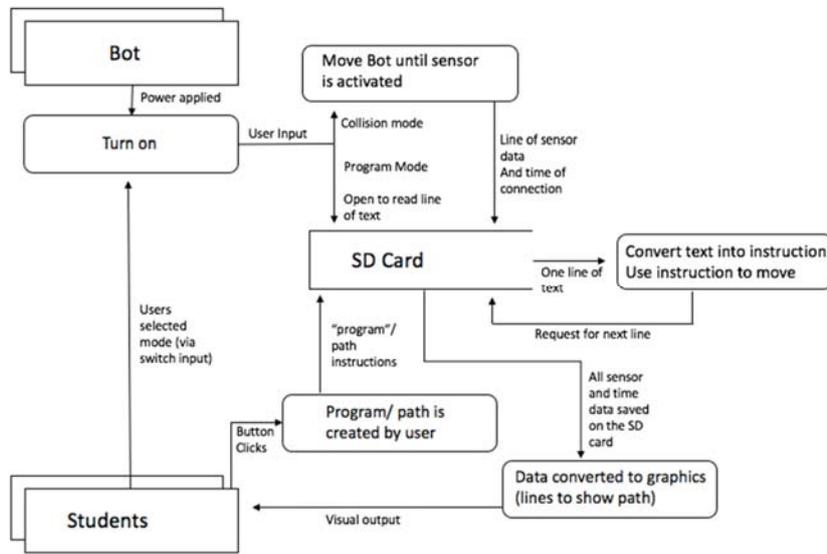
**Main program Overview:**



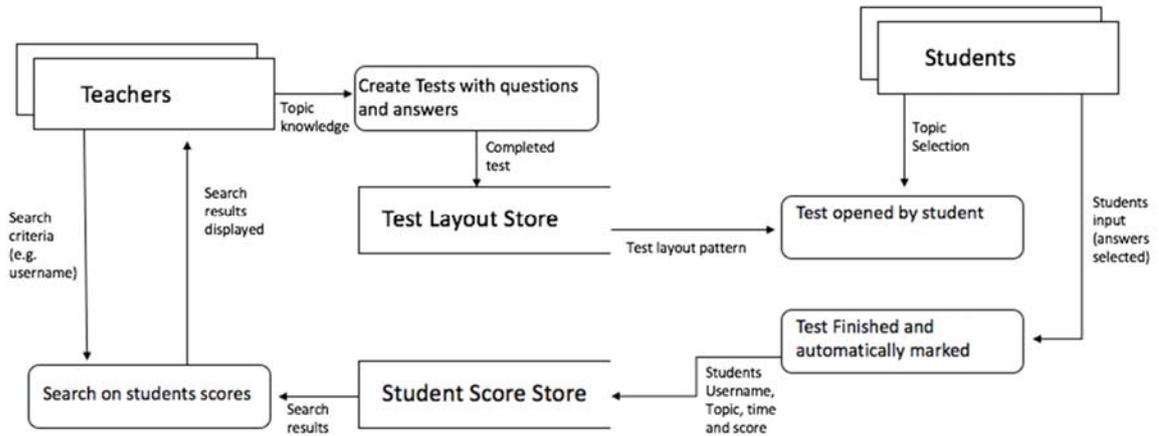
This is the way my system will fit together. The main program will hold all the key features, such as the tests, lessons, and bot data manipulation. The Admin and teacher page can only be accessed of the user logged in has the correct level of access.

The bot is separate to the main program, however it's still part of my system, the bot doesn't have forms, but it does have modes that can be changed during setup.

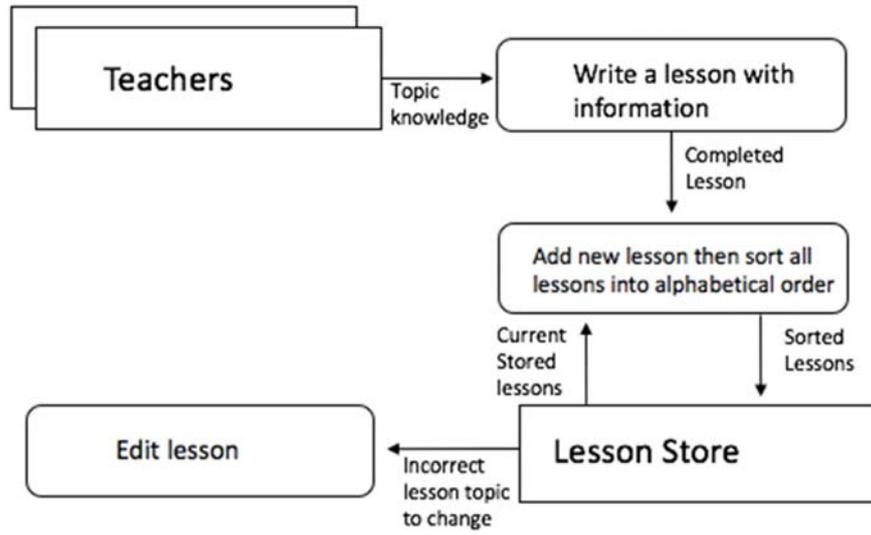
### Bot DFD



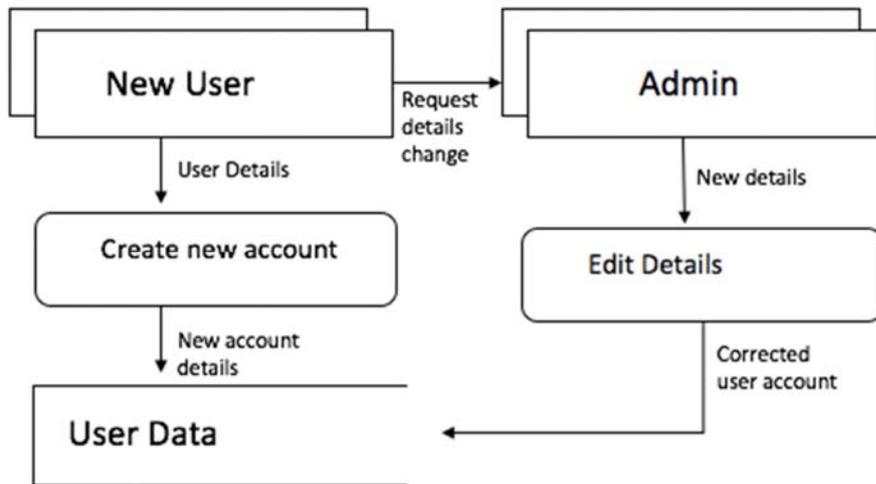
### Test and score DF



Lesson DFD



User Data DF



### Sign In Form:

A picture box will be used to make the program more appealing to the users.

The picture box will contain images and photos related to the program, the images will change periodically.

Text Boxes will allow the user to enter in their login details. The password text box will not be readable, to prevent another user overseeing the password.

A button that when pressed, takes the data entered form the text boxes and then compares them with the records form the User Data file, searching for a matching password and username.

If the user doesn't yet have an account, they press this button and are taken to the sign-up form.

This is the first form the users will see; the forms purpose is to allow the user to login to the program using their account. The design was chosen to be simple and easy to use, there's only three options upon opening it: Login, Close the program, or create an account. When the user enters their login details (username and password) the program searches for a matching username and password in the User data file. If found, access is granted and a 'note' of the user's record number is made before the main welcome page is shown. If no passwords match the users is denied access and is kept on this form with a message box telling them that something is wrong with their details.

### Form Output:

This form only gives one of two small outputs, if the user has entered incorrect details, then the system outputs an error message telling the user the details are incorrect, this allows them to think and ensure they re enter the details more carefully. If the correct login details are given the output message is a welcome message. The welcome message it just to reassure the user they are the person logged In.

## Sign Up Form:

Sign Up

First Name

Second Name

Email

Password

Repeat Password

Text boxes to hold the user's data. The different text boxes will hold information regarding the users details, these will then be used later on.

Button, starts the proses of validating the data entered starts before saving the file if correct.

Takes the user back to the Login Page when pressed.

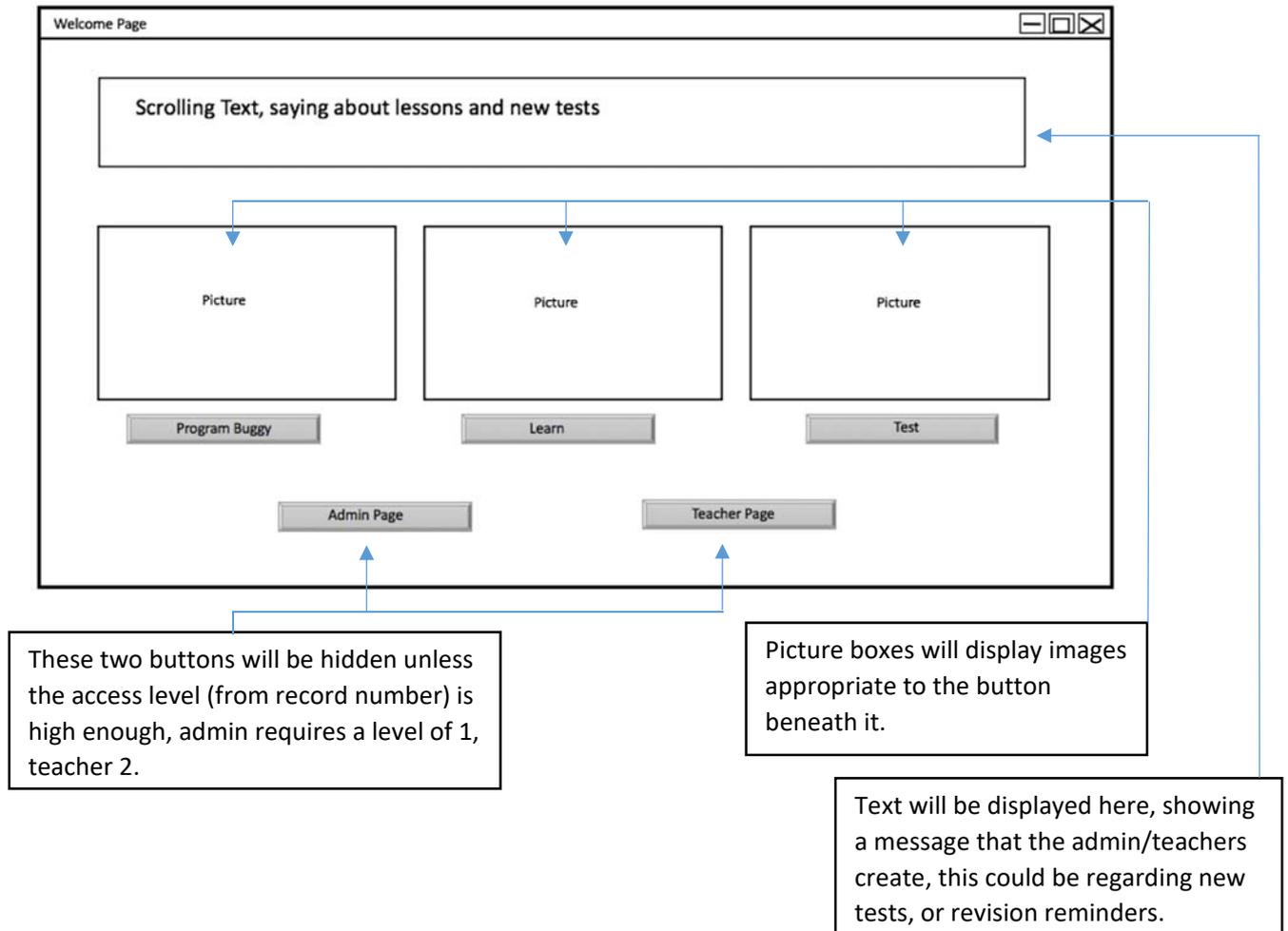
This form allows the user to sign up into the user database, when the sign up button is clicked the program reads the data in the text boxes and validates them. Using regular expressions to ensure the email has the correct format (a valid email.) The password will have to be typed in twice to ensure that it entered correctly (the program will compare them.) If the data entered is valid the program will allow then generate a username based off the Second Name, First Name, and Record number e.g. MurrayA18. The data is then saved into the text file; the record is added with the data.

## Sign Up Output:

The output from this form are either an error message telling the user to check their details, or a textbox with the user's new username on it. The error message is there to allow users to understand where they might have entered invalid details (such as passwords that don't match or an invalid email) this allows them to then correct the details before trying again.

The username message box is an important output as it gives the user the user their unique login details which are needed to enter the program, this username is generated by the program, therefore it has to be given out to the user.

## Welcome Page Form



This Form is the first shown once the user has logged in, it contains the links for the main pages, as well as having a way for the Admin/Teachers to leave messages for the students. The page reads the record number that was passed over from the logged in form, and finds the access level for the user. If the user has the correct access level the buttons "Admin Page" and/or "Teacher Page" can be accessed, if not, the buttons are invisible and unusable to the user.

The page's main use is to provide the user with a common homepage to make them comfortable with the software, the form does also have a more practical use of providing links to the other pages, as well as restricting access so only the admin and teachers can access the other features.

## Learn Form:

This combo box will hold all the different titles of the records it has in its learn data file. They can be selected by the user.

As a visual aid can be more helpful than just reading, I have included a way to watch videos from YouTube. The video's link can be copied in from the lesson.

Once pressed the program will perform a binary search for the Title in the combo box. With the Learn records. When a match is found, it loads up the text into the multiline text box.

The student can email the teacher if they require extra help. The combo box holds the teachers name, which is linked to the email. Text written in the text box is put into the email, when the user hits send the Email is created then sent.

A button that takes the user to the algorithms page, where they can use the algorithm simulations to help them understand how they work.

The learn page form is the main way that the the information. After the information has been the teacher page, the student can open and read the learn page form. The page also ensures that a valid link is entered, using a regular expressions check. The page quickly loads the data from the text file (as it finds it using a binary search) Once the data is loaded the student can copy parts that they find important, or even write notes in the text for their own benefits, however the edited text will not be saved.

students learn uploaded from it. The combo

## Learn page Outputs:

- Learn data –The data from the learn file is sent out the user as text, the student selects which topic they wish to learn and the program outputs it into a text box. The data is given so the students can learn and further their understanding from the text the teacher has left for them.
- Video output – the students can load up YouTube videos of the link is given by the teachers, this output gives them a visual and audio learning method rather than just reading text, this was added to keep the students entertained while learning and keep the students eager to learn
- Error message (video link) – the only videos that can be displayed are the ones linked on YouTube, if an incorrect link is given an error message is used to tell the user why the video did not load and prompt them to only use YouTube links.

### Test Page Form:

The screenshot shows a web browser window titled 'Test Page'. At the top left, it displays 'Time: 10s'. To the right is a dropdown menu labeled 'Test:' with 'Data Structures' selected, and a 'Start' button. Below this are five questions, each with a label (e.g., 'Question 1: Example Text?') and four vertically stacked checkboxes, each followed by an option (A) Answer, (B) Answer, (C) Answer, and (D) Answer. At the bottom center is a 'Mark and Save' button. Three callout boxes on the left point to the 'Time: 10s' label, the checkboxes, and the 'Mark and Save' button. Three callout boxes on the right point to the 'Start' button, the dropdown menu, and the 'Mark and Save' button.

A text label, that's connected to a timer, every second the label is updated with the time progressed.

Check Boxes and labels are the cosmetics used to hold and answer the questions.

The combo box holds all the test available (they are updated when the page loads)

Once the start button has been pressed, the timer starts, and the test is loaded.

When pressed, the timer is stopped, the current user's data (Username, etc) is saved with their score and time.

A check is made to ensure the correct number of boxes has been checked.

The Test page was designed to help the students revise and test their knowledge. Any test created by a teacher can be opened and taken by any student.

When the start button is clicked test is loaded, the Question file is opened, and the program loads the different questions from the same test. Ensuring the answer is in a different location each time, using a random number for the location it, within the four checkboxes.

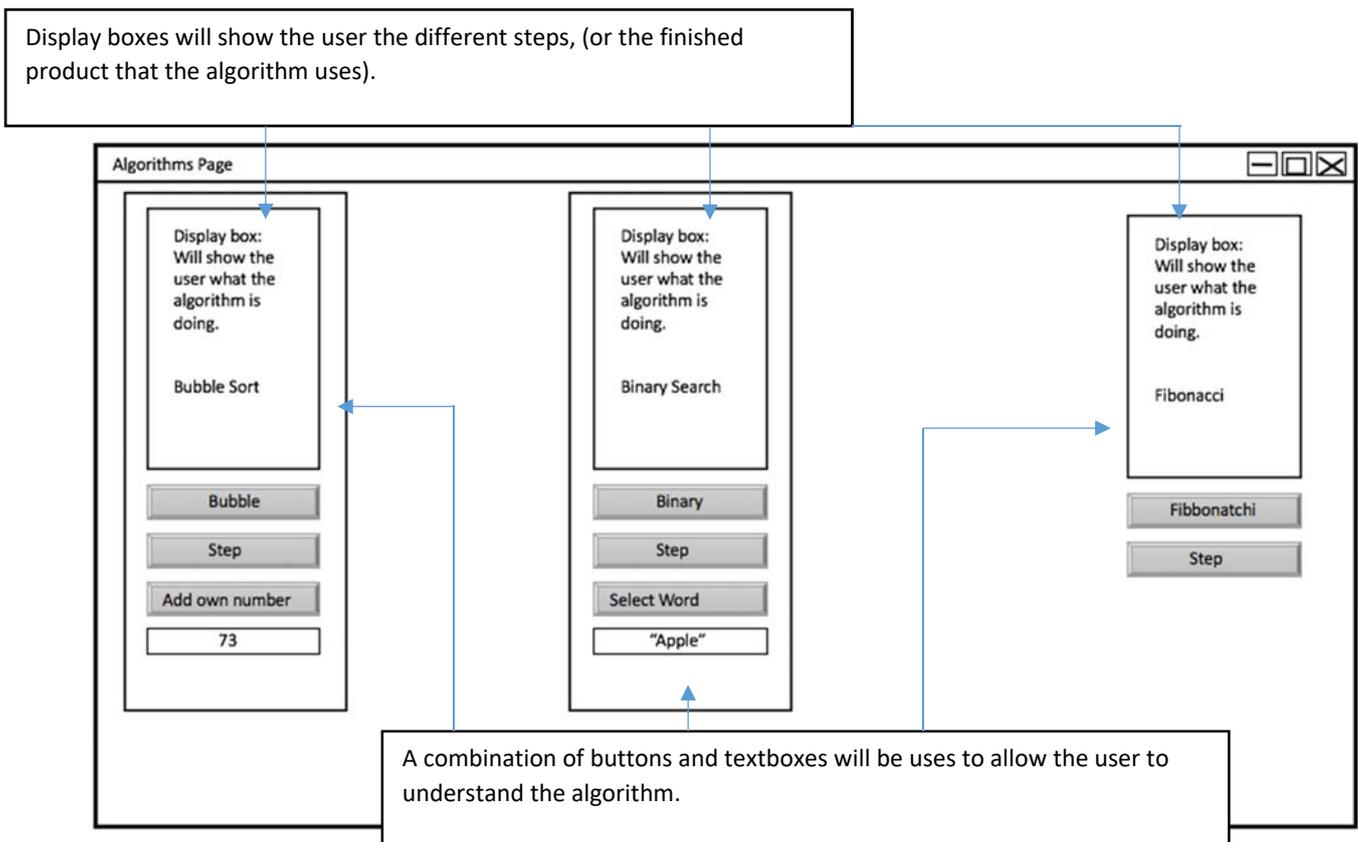
When the student has answered all the questions they click "mark and save" this stops the timer, and then marks the student's work, highlighting answers red or green if they got them correct or incorrect. The program will then add the score of the student (after checking that only the right amount of answers were selected) to the Test saved data file, which will hold the student's username, test taken name, time, and score.

### Test page outputs:

- Score and time – the student's score and time from the test are given out as a message just after the results have been saved. This is given to the student to let them know where they are up to in that topic, with the idea of self-improvement that they can retake the test if they are unhappy with the result

- Over Ticked score – if the student has selected over five answers (which is more answers than there are questions, then the program sends a message saying the score has been set to 0 and that the student needs to retake the test without ticking over five answers.
- Answers – after the score has been saved the student can see where they got answers right or wrong. The program marks the students selected answerer green if it was correct, and red if it was wrong, this visual output helps the student see where they went wrong and can allow them to go back and revise that topic.

Algorithm page:



Here the students can learn some of the common algorithms in more detail. With the help of the page, they can look at how the algorithm works, either step by step, or look at the whole solution for them to look back on. The students will be able to add their own arrays to look over and use (for sorting or similar algorithms) so they can better understand how the logic works (rather than just have a singular set array that wont change.) This will help grant the student a better understanding as it allows a lot more variation with the answers the student can see.

The outputs of this form are just the responses from the algorithms themselves (e.g. the sorted number from the bubble sort) This data is needed to be given as an output to show how the algorithms work and allow the users to understand and see the end results

### Collision mode bot page:

This textbox will be used to hold the location of the file the user wants to read from. They will copy the address into the textbox for the program to use.

This list box will act as a display, showing the contents of the file (data the bot has collated.)

The screenshot shows a window titled "Bot Page - Collision Mode". At the top left, there is a text box labeled "Data Location:" containing the text "E:\\BotSDcard.FileName". Below this is a "Display:" section with the text: "Will show the data extracted from the bot. Allowing the user to view how the bot 'sees' the world." To the right of the display is a large empty rectangular area. At the bottom of the window, there are four buttons: "Extract", "Erase", "Show Path", and "Program Mode". A blue arrow points from the "Data Location" text box to the top of the window. Another blue arrow points from the "Display" text to the left side of the window. A third blue arrow points from the "Show Path" button to the bottom of the window. A fourth blue arrow points from the "Program Mode" button to the right side of the window. A fifth blue arrow points from the "Extract" and "Erase" buttons to the bottom of the window. A sixth blue arrow points from the "Show Path" button to the large empty area.

When the buttons are pressed the program will either read (extract) or Delete (Erase) the text file, from the location given before.

When this button is pressed the program draws the path of the bot onto a page, that can then be seen by the user in the designated area.

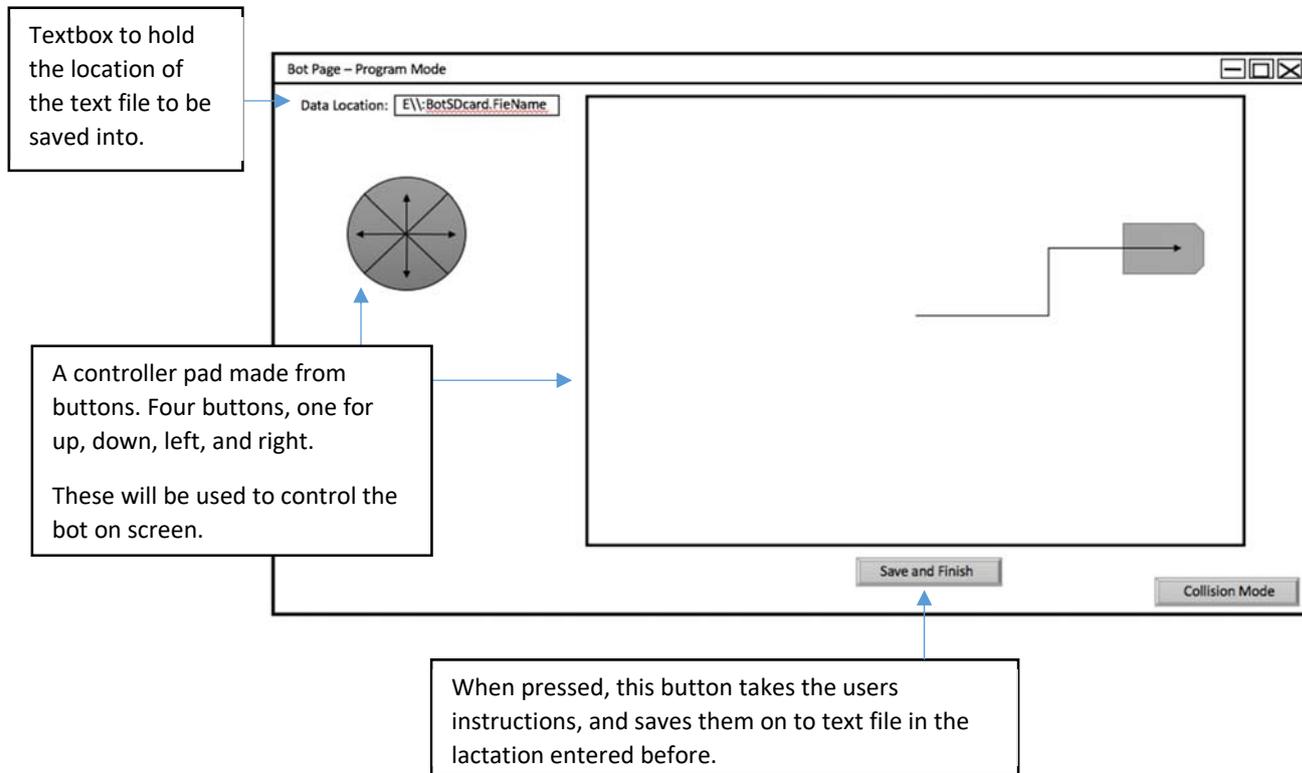
Takes the user to the other bot page, (program mode.)

After the bot has collated data, the student can use this form to see what the bot collided with, the different turns it took, and the relative distance (calculated via the time between turns.) The student gives the form the location of the text file created by the bot, and then extracts the data, where is manipulated, and then shown to the user via length of lines. From this form the user can access the other bot page, and can also erase the text in the text file, leaving the SD card ready for use again.

### Form outputs:

This form has two outputs. The first output is the data collected from the bot that was saved onto the sd card. This data is saved onto the program for the user to see and understand before the next stage occurs, this makes it easier for the user to "see" what the bot is doing and how it does it. Next the output is a visual output, given to the user, the lines on the screen need to be shown, otherwise the bots path will remain in text form.

### Program bot form:



This page is used to program the bot, using the arrow buttons, and on screen bot, the user can maneuver the bot around the area on the page. Each time a direction is added to the virtual bot, a text file has an instruction written on it, e.g (Up) this is then saved once the student has finished.

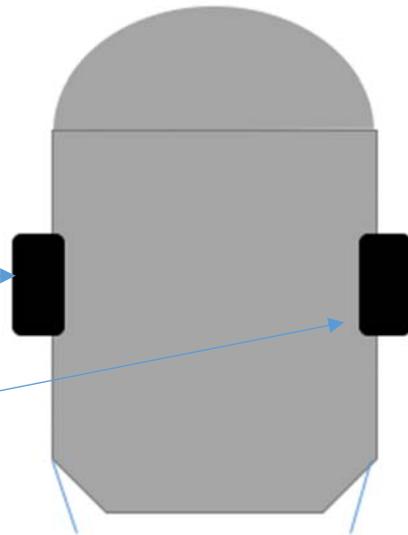
After the bot's text file has been programmed, it can then be added to the SD card, and then used in the bot, which will read the file line by line, and decode the instructions left by this page.

This forms main output is the set of instructions given by the program that's then saved onto the sd card, this 'program' is generated by the main program when the user presses the buttons on the screen. It allows the users program to be transferred onto the real bot afterwards, not just having the on screen bot move about.

The secondary output for the page is the virtual bot's movement, this allows the user to see 'about' where the bot would move if programmed, for example when the user selects up to have the bot move up on screen. This gives the user a better understanding of what they are actually programming the bot to do, and how they get the the bot to move.

Bot:

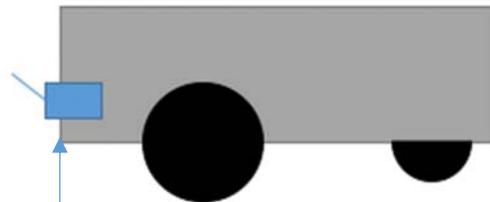
Top View



The bot will have two powered wheels and one 'free' wheel at the back for stability.

To move the bot the powered wheels, have to be turned on, with different combinations for different actions. E.g. (Wheel A + B on for the bot to move forward)

Side View



Micro Switches, small push to make/ break switches that can be triggered easily are used at the front of the bot as a way of sensing when the bot collides with an object, or when a mode is selected during setup.

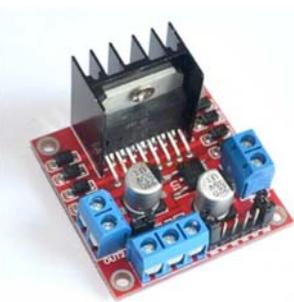
Although the bot isn't a form, it still has form like properties, such as in order to start the bot off a mode need to be selected by a user. The Two micro switches at the front on the bot are essentially buttons, they will allow the user to select the mode they wish by pressing one of them in upon startup. This will tell the bot what mode to start up, and use the correct code.

The bot will have two main modes, Compass mode, and Collision mode, both will require an SD card to feed/receive data to and from the bot, as well as move the motors and decide which way to move. Compass mode uses data created by the Visual basic main program, whereas Collision mode generates data for the Main program.

Collision mode uses the bots sensors to navigate around, after it bums into an object (e.g. a wall, chair, foot, etc. ) It stops, reverses a tiny bit, then turn and continues to move forward until something activates the switch again. All the time this is happening the time of the bot will be saved on the SD card, along with the switch status. The data will be used in the main program later on.

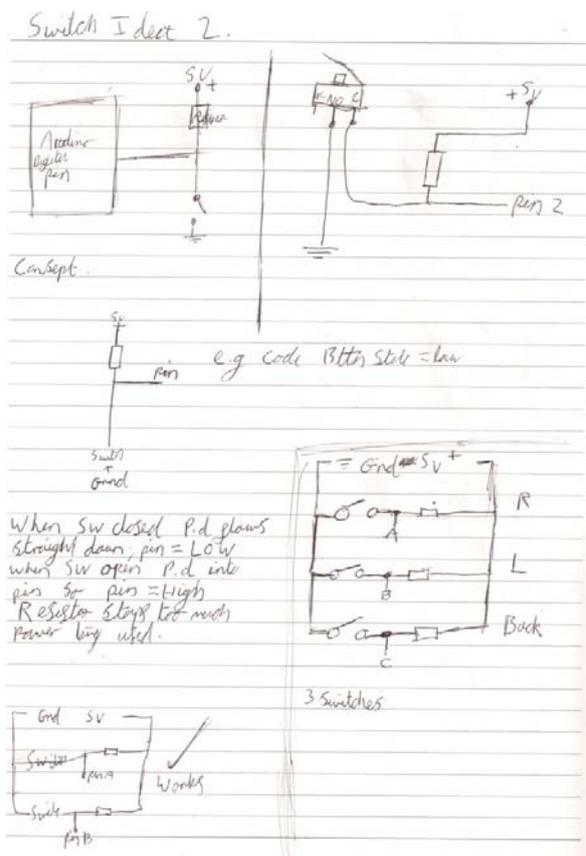
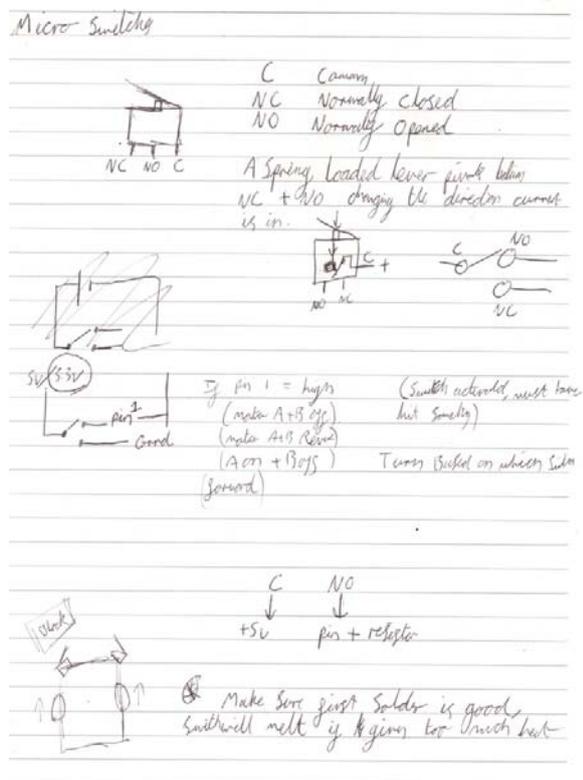
Compass mode used the data created by the main program, the bot will read the SD card line by line seeing the instructions as text and decoding them. The bot will preform each instruction one by one (e.g "forward" the bot will activate its wheels and move forward.)

As the bot is not a form the design and construction stage of it was very different to what I did for all the other forms. As the bot is a physical item I had to look into what parts I will need before I made it, I needed a chassis, wheels, a brain (micro controller), a power supply, an SD card adapter and sensors. I already had a few of the list but I later realized I would need more specialized items, such as a motor driver and a secondary power supply (one for the motors and one for the microcontroller.) I ended up designing my bot to work with a L298N Dual Motor Controller as this was a relatively cheap one I could buy and replace if I shorted it. I already had planned to use an Arduino Uno and power it using a battery pack. This will all sit upon a a multi purpose use plastic chassis I have. The wheels would be connected up to the motor drier and the motor driver to the Arduino.

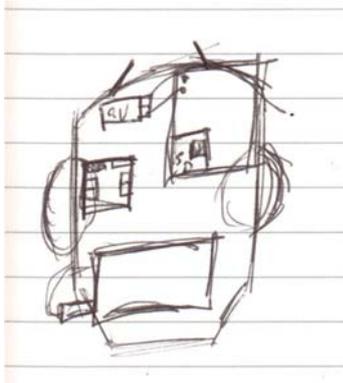


Because the wheels will need much more power I have to use two power supplies, therefor the motor driver will have its own 9V battery connected up to it, meaning it can drive the wheels and itself without interrupting the Arduinos power supply.

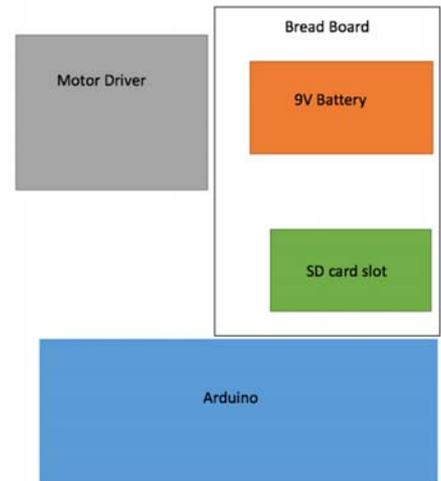
The sensors I decided to use were micro switches as they were cheap and had a simple output. However, I still needed to know how to make the switches connect with the Arduino and a small amount of research was needed. After some different designs I settled on using the using micro switch in the following way.



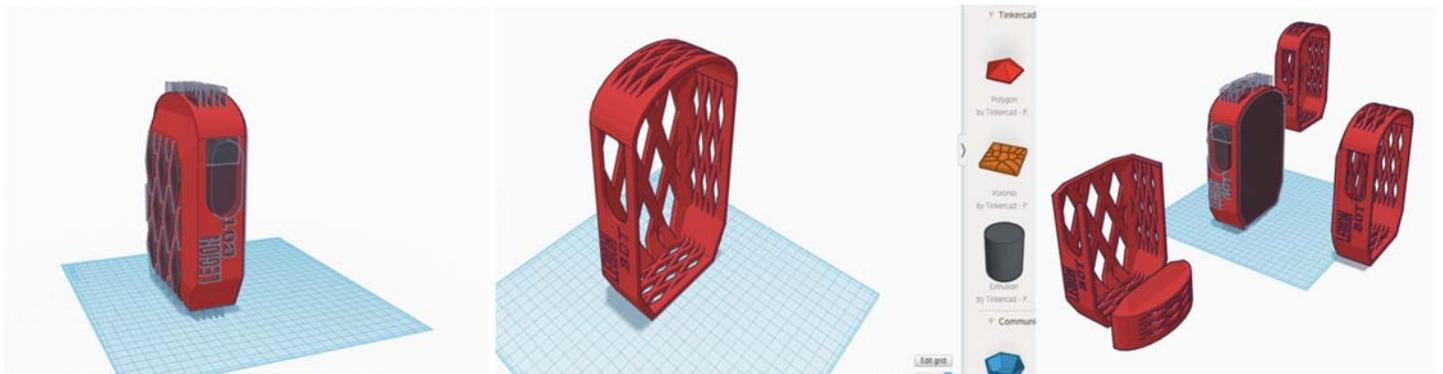
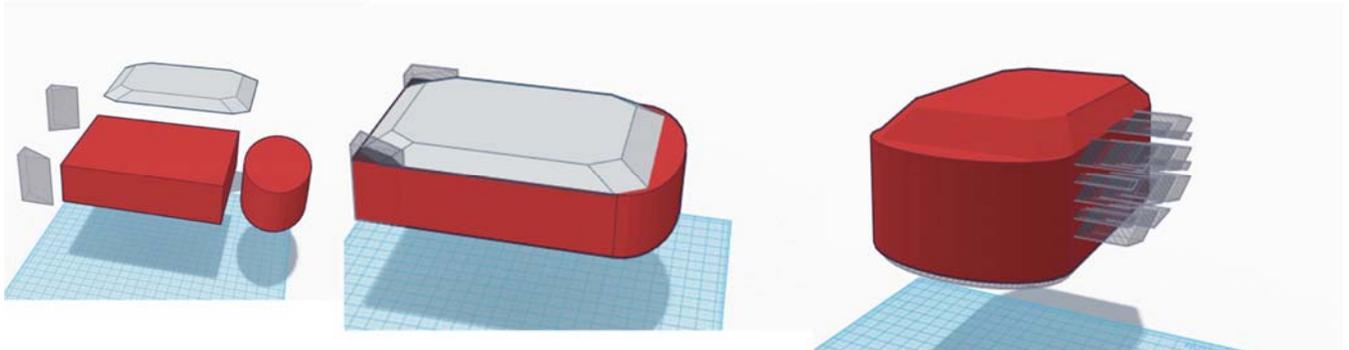
The design used the switch to act as a break for the circuit, therefore the pins on the board will receive a continuous signal until the switch has been hit, then the power from the board will be sent to the ground pin which stops the power going to the board. (a resistor will reduce the amount of current the switch can draw, thus increasing my battery life) After seeing a rough circuit diagram, and realizing that all my wiring might need additional connections I decided to add a breadboard to my bot's design.



One of the basic sketches for the bot I made showing some of the components I will use on it. The bot will hold all the items it needs on it and be self contained, this will allow it to move around freely and not have to have cables arched to it. On the right is the layout of the items I intend to add to the board.



In my design I also had the idea to make a case for the bot, to protect the wiring as it would be delicate and could be damage if a pin was taken out. This case will be printed by a 3D printer, the designs for it have to be made on 3D creation software, I used tinkercad to make the case using different shapes and sizes and then joining them together and ensuring they were the correct size for the bot.



This Text box will be locked, to stop a user from altering it.

Text boxes will hold the selected records data.  
Allowing the Admin to change them if needed. The boxes will be validated to stop incorrect data being entered.

The screenshot shows a window titled "Admin Page" with standard window controls (minimize, maximize, close) in the top right corner. On the left side, there are six text input fields, each with a label and a blue arrow pointing to it: "Record Number" (containing "7"), "User Name" (containing "SurnameN7"), "First Name" (containing "FirstName"), "Second Name" (containing "Surname"), "Email" (containing "example@email.com"), and "Password" (containing "Pa\$\$0rd1"). Below these fields are two buttons: "Save" and "Search". In the center of the window is a large rectangular "Display Box" containing the text "Display Box: Will show all the text from the admins searches". At the bottom center of the window is a "Display All" button. On the right side of the window, there is a "Multi-Line text box" containing the text "Multi-Line text box: Will be the scrolling text for the welcome page." Below this text box is a "Save" button. Blue arrows point from the text boxes on the left to the corresponding input fields in the form, and from the "Save" button in the bottom right to the "Multi-Line text box".

The text that is entered here will go on the welcome page for all the other users to see, that can be changed as any admin sees fit.  
Once written the admin just clicks "save" to make changes to the text file its connected to.

The Admin page is only accessible to Users with the correct clearance, as it holds sensitive information regarding the users, the form is for use of admins only. The Admin can load up the records of the users, (or they can search for a particular one if they know the information using the text boxes and search button.) The records are loaded up into the display box in the same format as the textboxes are layout. If the admin wishes to change the data they'll click on the record number of the displayed student data, which will be sent to the "record number" text bot, and all the data will be automatically entered into the other textboxes. From there the admin can edit and save when done. The admin will also be able to edit the message displayed to all users, to tell them about new lessons and events. This message is saved in a text file.

#### Admins page output:

The page's output is contained within the page, no outside sources (bar the user) can get the output. The big output for the page is the user's information, on this page is can be seen normally and isn't encrypted. Therefore, this will be on of the only forms that can give the user data as an output. The trust is given to only admins as they have the correct authority to view the data.

This combo box holds the names of the tests, they are loaded in from the file when the form loads up.

The combo box holds the names of the different lessons, they will be loaded in when the page starts up.

The screenshot shows a 'Teacher Page' window with three main sections. The left section is for adding tests, featuring a 'Test Name' dropdown menu (currently showing 'Algorithms'), a 'Q Number' spinner (set to 1), a 'Question.....?' text box, and four radio button options labeled 'A)', 'B)', 'C)', and 'D)'. Below these is an 'Add' button. The middle section is for lessons, with a 'Lesson Name' dropdown menu (showing 'CPU Clock Speed') and a large multi-line text box for content. Below the text box are 'Load' and 'Save' buttons. The right section displays student information: 'User Name' (NameU6), 'Test Name' (Bubble Sort), 'Score' (99%), and 'Time Taken' (60s). At the bottom right is a 'Display All' button. A 'Display Box' on the far right is labeled 'Will show students test scores'. Callout boxes with arrows point to various elements: the 'Test Name' dropdown, the 'Lesson Name' dropdown, the radio buttons, the 'Load' and 'Save' buttons, the 'Time Taken' field, and the 'Display All' button.

The answers to the the questions are written in these check boxes.

These buttons will load up an existing entry, or save a new one, depending on if the "lesson name" has already been used before.

When the record is loaded in, the details are displayed in the text boxes.

When pressed, all the records in the test score file, will be loaded up.

This form is vital for keeping the program up to date, the teachers need this page to load in new data for the student's education. The page allows teachers to add in tests, and test questions, and add in lessons, containing information in a text form. The teachers can also see how a student is doing by checking their score on the tests.

Using the left section, they can add new tests by typing in a new test name into the combo box, as this combo box already has all the names of previous tests in it, any new ones will be found, and then a new test will be added to the record. The questions of the new test can be added by changing the question number, then adding the answers and question into the provided text box and clicking the "Add" button.

The middle section is where the content of the lessons can be added, the idea is to have short lessons, with links that can be used added in for the students to visit. The lessons will be written by the teachers, they can edit old lessons by loading them up then saving them, or just by saving a new one. The name of the lesson is the record in the file, the combo box holds the names for the users to easily select one to load, or they can write a new one in themselves.

The right section allows then teachers to keep track of a student's score, they can display all the scores from all the tests at once, or search by test and then look down the list of usernames. The students

score and test details are all created upon the student taking a test. They cannot be edited by the teacher as there is no need for the teacher to edit them.

Outputs:

- Lesson data – this output is given by a text box, the lessons the teachers write are both inputs and outputs as it allows the teachers to load up the data they have made into the textbox for them to check over or change.
- Students scores– the teachers get a visual output of the students tests and scores as they can load this output into the combo box used for viewing the score. This is needed so the teachers can track the students progress and keep up to date with the work they have set the students.
- Tests: questions and answers – the teacher have the test questions they have made outputted onto a list box so they can see what tests and questions have already been made/used. This was added to help prevent teachers repeating tests.

**Data Structures:**

Text Files:

File Name	Function	Method of access
AdminMes.txt	This text file as a very small job, its function is to hold a written text message that the admin has written for others to view upon stating the program. It is completely re written each time its changed, as the message is only supposed to be small.	Only an admin can access the page required to edit this file. After entering the desired text to save the admin presses save changes button, then the file is completely re written with the new entered text. This file rewires it self whenever the new data is saved.
LearnData.txt	The file holds all the lessons created by the teachers, the names and contents of the lesson are stored in alphabetical order, as to be quickly and easily accessed. The lessons held are text based, but can contain links to other sites for the students, they can hold a more than suitable amount of text.	This file can be accessed by anyone on the program, but only admins and teachers can edit the contents of it. The top of the file contains the number of records it holds, after this is read, the file read all the remaining lines putting them into a structured array (LeanArray)This structured array is then used to search or edit the contents. All the data in the file is stored in alphabetic order, this allows for binary searches. This file will be sequential as the topics will be arrange in alphabetical order when saved into the file as to reduce the number of searchers the program will have to do to find one particular topic (as there will be lots of topic data.)
TestLayout.txt	The multiple choice questions created by the teachers are stored in this file. The questions and possible answers to all the tests created are stored in this one file. This prevents having separate files for every different	This file will be a serial file, all the test questions added will be sent to the end of the file. The questions will still be linked by the record and question number. The files first line contains the number of records in it, each record is one question from a set of five. After reading the first line the main program reads the file into a structured array (TestLayout) This structured array can be manipulated and

	test. The file can be edited by teachers (creating new test/changing them.) The file will be read by students, where the correct test will be selected, and brought up for the student to complete.	then written back into the file (so long as the record number is also changed accordingly.) The teachers can manipulate and change the data in the array to then change the file, but the students can only read it and search for particular texts. The structured array is read and put into label boxes / check boxes as a visual output to the students to read.
TestScore.txt	After the students have completed a test the data about the results are saved in this file, this allows the teachers to review the students progress, they can even use the data to compare the students result to another. This data is an important output as it allows the teachers to understand the weak points of the students and lets them know they need to do more to help them (or push the individual to help themselves.)	After the student has marked a test the program begins to open the file, it reads the first line, to get the number of records, before reading the rest of the data and saving it into the structured array (TestScore)The data created from the students test is then added to the array and saved back into the file with a changed number of records. The teachers can access the file to review data, they can search and look for particular usernames. The new records added will be saved one after another in a serial file.
UserData.txt	This file holds the most important data in the program. All the students, teachers, and admins details are contained in this file. The unique username for each user is stored in the file, with the user's email, name, personal data etc. The file is used by the program to see who's logged in and using the system, as well as who should have access to certain features.	This file is encrypted when not in use, so the file has to be decrypted before it can be accessed. After the file has been decrypted, the number of records is read of from the file. The rest of the file is then read line by line into a structured array (UserData) this array can then be manipulated by the admins, or read by the program to check data such as assess levels. Once the file is finished with, it must be re encrypted to prevent it being accessed by an outside source. This will be a serial file, as the records will be added one by one with the oldest being the first and the newest being the last.
BotFile.txt	This is the text file that will be shared with the bot the and main program, (held on the SD card.) This files job is to receive data from the bot (in the form of letter and number) or to provide data for the bot to read. The file is the only method the user will have with communicating with the bot. It more or less just holds the data created by/for the main program to be used by/for the bot	The file can be accessed in different ways due to it being used by two programming languages. In VB the program reads the file line by line and sets the data in to an array, when writing to the file the program reads the created array into the file line by line. The Arduino side, writes to the file line by line adding to what is already in the file (it don't write over, it just makes a new line.) When reading data the program looks for a new line, then checks the lettering for the information its looking for. Due to the way the bot saves data, this file is a serial file, the new data is added to the old line by line.

### AdminMes.txt

Field	Description	Type	Example Data
Display Message	The message from the admin is stored here. This will hold any text the user wrote and saved. The message is shown on the Welcome page to all users.	String	A new test has been added at the test page, go now to test your knowledge on "algorithms"

### LearnData.txt

Field	Description	Type	Example Data
Number Of Records	Only appears once in the file (at the very start) used to tell the array how many records are in the file.	Integer	3
Name	The 'name' of the lesson is held, this is used for sorting the data, and to locate the correct lesson (info)	String	Documentation
Info	Holds the text based lessons written by the teachers. The information held is displayed to the students for them to learn from. The lessons are also prompted to have a YouTube link in it.	String	An important part of writing a program or system is to document it according, for example if you were designing an educational system, you would have to write up how you would store lessons and the information on the lesson..... etc.

### TestLayout.txt

Field	Description	Type	Example Data
Record number	A number that tells the array the records identity, this is unique for each entry can cannot be edited by a human.	Integer	2
Question Number	The question number that the test has (out of five)	Integer	4
Test Name	Name of the test	String	Binary Search
Question	A question written by the teacher for the student to answer	String	What is the maximum number of searches than can be preformed with an array of size 16?
Answer A	Answers are written by the teacher, all answers bar D should be similar to the correct answer, but only D should be the real answer.	String	16
Answer B		String	2
Answer C		String	99
Answer D		String	4

#### TestScore.txt

Field	Description	Type	Example Data
Record Number	A number that tells the array the records identity, this is unique for each entry can cannot be edited by a human.	Integer	2
Test Name	Holds the name of the test taken by the student, this is read from the the drop box the user selected.	String	Bubble Sort
User Name	The name of the user that took the test. Its read from the user data file.	String	MurrayA1

Mark	The students raw mark from the test, based form the correct answers they got correct.	Integer	5
Time	The time taken for the user to complete the test and press mark, saved to the nearest second.	Integer	60
Score	The students mark and time is used to make a score, this rewards the student for a shorter time, and a high mark	Integer	30

#### **UserData.txt**

Field	Description	Type	Example Data
Record Number	Number that represents the records place.	Integer	5
Username	A unique name that is generated by the users Second name, first name and record number. Used to login and identify user	String	MurrayA1
Email	Users email address, for contact purposes.	String	example@eg.com
First Name	Users First Name	String	Alex
Second Name	Users Second Name	String	Murray
Date of Birth	Users Date of Birth	Date	1 <sup>st</sup> June 1998
Password	A secure word/number commination than only the user should know (acts as a key)	String	Pa55W0rd3

Access Level	A number that represents the authority of a user. Restricting or allowing the usage of the programs features.	Integer	2
--------------	---	---------	---

Structures:

Field	Description	Type	Example Data	Validation Used
Record Number	Number that represents the records place.	Integer	5	-None- (not needed as set by program)
Username	A unique name that is generated by the users Second name, first name and record number. Used to login and identify user	String	MurrayA1	None- not required as its generated by the program (from two other validated parts)

Email	Users email address, for contact purposes.	String	example@eg.com	Regular Expression check (format and presence check)
First Name	Users First Name	String	Alex	Length check (Not over 15 characters)
Second Name	Users Second Name	String	Murray	Length check (Not over 15 characters)
Date of Birth	Users Date of Birth	Date	1 <sup>st</sup> June 1998	None – selected from windows form.
Password	A secure word/number combination that only the user should know (acts as a key)	String	Pa55W0rd3	Regular Expression ([0-9][A-Z][a-z]) Presence check for a number, lowercase and uppercase letter.
Access Level	A number that represents the authority of a user. Restricting or allowing the usage of the programs features.	Integer	2	None- (not needed as pre set of 3 is given to the user)

Field	Description	Type	Example Data	Validation
Number Of Records	Only appears once in the file (at the very start) used to tell the array how many records are in the file.	Integer	3	-Preset-
Name	The 'name' of the lesson is held, this is used for sorting the data, and to locate the correct lesson (info)	String	Documentation	
Info	Holds the text based lessons written by the teachers. The information held is	String	An important part of writing a program or system is to document it according, for example if you were	

	displayed to the students for them to learn from. The lessons are also prompted to have a YouTube link in it.		designing an educational system, you would have to write up how you would store lessons and the information on the lesson..... etc.	
--	---	--	---	--

Field	Description	Type	Example Data	Validation
Record number	A number that tells the array the records identity, this is unique for each entry can cannot be edited by a human.	Integer	2	none
Question Number	The question number that the test has (out of five)	Integer	4	Range check
Test Name	Name of the test	String	Binary Search	Presence check
Question	A question written by the teacher for the student to answer	String	What is the maximum number of searches than can be preformed with an array of size 16?	Presence check
Answer A	Answers are written by the teacher, all answers bar D should be similar to the correct answer, but only D should be the real answer.	String	16	Presence check
Answer B		String	2	Presence check
Answer C		String	99	Presence check
Answer D		String	4	Presence check

Field	Description	Type	Example Data	Validation
Record Number	A number that tells the array the records identity, this is unique for each entry can cannot be edited by a human.	Integer	2	None – Given by program
Test Name	Holds the name of the test taken by the student, this is read from the the drop box the user selected.	String	Bubble Sort	None – as taken from program (user cant change)
User Name	The name of the user that took the test. Its read from the user data file.	String	MurrayA1	None – as taken from program (user cant change)
Mark	The students raw mark from the test, based form the correct answers they got correct.	Integer	5	None – as taken from program (user cant change)
Time	The time taken for the user to complete the test and press mark, saved to the nearest second.	Integer	60	None – as taken from program (user cant change)
Score	The students mark and time is used to make a score, this rewards the student for a shorter time, and a high mark	Integer	30	None – as taken from program (user cant change)

### Arrays:

#### Bubble Array-

Field	Description	Type	Example Data	Validation Used
0	The Array will hold numbers, that will then be sorted using a sorting algorithm, the value of the numbers themselves doesn't matter.	Integer	34	None required as this array is built in to the program.
1		Integer	23	
2		Integer	23	
3		Integer	1	
4		Integer	27	
5		Integer	99	
6		Integer	12	

New array-

Field	Description	Type	Example Data	Validation Used
0	The Array will hold a given amount of numbers chosen by the user. The user enters the numbers, these numbers are then sorted using assorting algorithm	Integer	34	The length of the array must be under 30 and greater than 0 (range check)
1		Integer	23	
2		Integer	23	
3		Integer	1	The data entered into the textbox, must be numeric.
4		Integer	27	
5		Integer	99	
...etc.		Integer	12	

SetStringArr-

Field	Description	Type	Example Data	Validation Used
0	This array will hold 6 words to be binary searched, they will be stored in alphabetic order.	String	Alpha	None as its pre set into the core code of the program.
1		String	Delta	
2		String	Hotel	
3		String	Indigo	
4		String	Lima	
5		String	Zulu	

Program mode array-

Field	Description	Type	Example Data	Validation Used
0	This array holds the instructions to be saved onto the SD card. Button clicks add to the array.	String	Up	None required as the strings are added when buttons are pressed, from a preset option.
1		String	Up	
2		String	Left	
3		String	Up	
4		String	Right	
Etc..		String	Up	

## Validation

### Range check:

A range check can help prevent data of a certain size or value from being entered when another should be in its place. For example, the length of a password should be between 5-10 characters, so its not too long or too short. Range checks ensure data is between two values, less than one greater than another. Sometimes a range check is needed to stop the user entering too much/too high a value, which could cause the human to make a mistake further on (e.g. a password too long to remember.) Or to stop the human user from starting something they wont complete, (such as entering numbers into an array)

Human input	Output/Operation
12	*proceed with code
9999999	Too Large Please select a lower value (<30)
0	Please select the number of numbers
10	*proceed with code
3	*proceed with code
31	Too Large Please select a lower value (<30)
92	Too Large Please select a lower value (<30)

### Numeric check:

Some data used by the program has to be a certain type, for example you can't divide a string by a number, it isn't logically possible and would crash a computer if it was attempted. If a human user is given the ability to add data into a system they must be stopped from entering the wrong type of data. a numeric check ensures the data in a location is an integer, not text or another.

Human Input	Output/Operation
3	*proceed with code
Two	"Please enter an integer"
56	*proceed with code
999999	*proceed with code
kdw	"Please enter an integer"
eleven	"Please enter an integer"
12.98	*proceed with code (round to nearest integer)
3.14	*proceed with code (round to nearest integer)

**Format check:**

Format checks are used to ensure data is save in a certain form/way. They can be used to make sure the data entered, is entered in the correct box/input area, especially if the data being entered is similar. Format checks are used to ensure the data entered is of a certain layout (must be dd/mm/yy) if the data if the same but in the incorrect format the data would be rejected (e.g. 12<sup>th</sup> October 1998.) For some data a format check will be just to ensure all the data is in the same layout for ease of use when reading (human.) The other reason for format checks is so the data entered wont break the system do to the data type being different. This can be very important when using complex searches or website links.

Use example	Human Input	Output
Link to a YouTube video (Regular expression)	<a href="https://www.youtube.com/watch?v=qDXhTPjFMrY">https://www.youtube.com/watch?v=qDXhTPjFMrY</a>	*proceed
Link to a YouTube video (Regular expression)	<a href="http://www.youtube.com/qDXhTPjFMrY">www.youtube.com/qDXhTPjFMrY</a>	*don't allow
Date of birth (dd/mm/yy)	12/10/19	*proceed
Date of birth (dd/mm/yy)	12 <sup>th</sup> /10/19	*don't allow
Date of birth (dd/mm/yy)	12 <sup>th</sup> October 1998	*don't allow
Date of birth (dd/mm/yy)	09/12/12	*proceed
Date of birth (dd/mm/yy)	13/04/1992	*don't allow

**Presence check:**

Humans are normally lazy and/or forgetful. When entering data humans normally skip or miss important data entry points. As some systems could be broken if they are missing data a presence check is normally used to ensure data has been entered, the data its self doesn't matter so long as there is data in the field. This is a very common form of validation as data is normally entered if its important (e.g. emergency contact number)

Use example	Human Input	Output
Name entry	-Null-	*message box asking for a valid input
Name entry	Alex	*proceed with code
Name entry	asdfgh	*proceed with code
Name entry	2554	*proceed with code
Password (sign up)	Password1	*proceed with code

Password (sign up)	-Null-	*message box asking for a valid input
Password (sign up)	123	*proceed with code
Password (sign up)	\$£%%\$^	*proceed with code
Sorting array entry	-+=	*proceed with code
Sorting array entry	q	*proceed with code
Sorting array entry	21	*proceed with code
Sorting array entry	-Null-	*message box asking for a valid input

### Length check:

When data is entered into system, the length of the information could be the only way to tell if something is valid, for example a first name being entered into a system can be any different combination of letters given in an order (or numbers with some VERY peculiar name changes) therefore the only way to ensure the data entered into the field is correct is length check to ensure someone hadn't just accidentally lead on key for a few seconds. Length checks ensure that the length of the data entered isn't over or under a certain number of characters (e.g. a password isn't over 50 characters but is more than 3) Below I checked that a name isn't over 10 characters

Use example	Human Input	Output
First Name entry	Alex	* proceed with code
First Name entry	Alexander Murray	* message box asking for a valid input
Second Name entry	Frank	*proceed with code
Second Name entry	Frankkkkkkkkkkk	* message box asking for a valid input

### Pseudo Code:

#### Bot-

#### Turn on/Setup

- Set pins (Declare them)
- Check Switches
- Check for SD card connection
- Setup Serial port
- If Right Switch has been pressed set mode to 1
- If Left, set mode to 2

### Mode 1

- Set "CurrentTime" to the accumulated time
- Read R+L Switches for digital input
- If the "R/L SW state" is not the "last R/L SW state"
  - If the R/L switch state is low
    - Add 1 to the R/L counter
    - Serial print "Switch R/L on"
    - Serial Print "Switch R/L counter"
    - Set previous time as current time
    - Set "save value" as previous time
    - Call "save to file"
    - Reverse bot
    - Delay program by 1s
    - Turn bot left /Right
    - Delay program for 50 milli seconds
    - End if statement
  - Delay for 30 Milli Seconds
  - End if statement
- Set the "last SW state" to the "current SW State"
- Move the bot forward
- Loop

### Save to file

- Set File to write
- If file is found
  - Serial print "writing to file"
  - Write to file "R" and "SW R Counter"
  - Write to file "L" and "SW L Counter"
  - Write to file "At the time of" and "Save Value"
  - Serial print "done"
- Else

- Serial print "error"
- End if statement

### Mode 2

- Set file to read
- if the file is found
  - Start to Read file if not fully read - loop
  - Set read text as "Line"
  - Stop when a new line in the file is found
  - Serial print "line"
  - If "line" contains Instruction
    - Run related instruction
    - Delay program for 1s
  - Loop if the file hasn't been fully read
  - Close file
- Else
  - Serial print "Error"
- End if statement

### Movement

- Set digital pins of Right motor ( + and - )
- Set digital pints of Left Motor ( + and - )
- Set the PWM pins to 180 for Left
- Set the PWM pins to 160 for Right

(change polarity of pins for a motor to change direction, use both motors in the same polarity for forward and reverse, use different polarities for turning.)

(Pulse Width Modulation is used to control the motors speeds, needed as the wheel (hardware) wobbles and pulls to the side, increasing the speed of the left motor compensates for this.)

Main Program-

**Sign in Form –**

### Read User File

- Decrypt the file
- Read the first line in the file
- Set the first line as the number of records in the file

- Set the structured array to hold the number of records
- Set the counter to 0
- Loop until the file has been read fully
  - Increase counter by 1
  - Read line - Set the structured array to hold the ID number for position of counter
  - Read line - Set the structured array to hold the Username for position of counter
  - Redline -Set the structured array to hold the Email for position of counter
  - Etc. (filling in all the details of the structure for the array position 'counter')
  - Redline -Set the structured array to hold the Access level for position counter
  - Return to top of loop
- Close the file
- Encrypt the file

#### Upon 'Sign in' Button Press

- Set logged in as false
- Read User File ()
- Set counter to 0
- Loop until logged in is true
  - Increase counter by 1
  - If counter is greater than the number of records
    - Show a message box with an error and text
    - Clear password and username textbox in form
    - Exit loop
    - End if statement
  - If the username and password text box holds text, the same as the User File structured array username and password for position of the counter.
  - Change logged in value as true
  - Sent a message box saying welcome with username
  - Open the welcome form
  - Hide the current form
  - If the access level of the user is 3
    - On welcome page – hide teacher and admin buttons
  - Or if the level is 2
    - On welcome page show teacher and hide admin buttons
  - Or if the level is 1
    - On welcome page show teacher and admin buttons
  - End embedded If statement
- End if statement
- Loop to top of loop

#### **Sign Up form –**

#### Save Users into file

- Decrypt user file
- Read the first line, set value as record number
- Encrypt the file (this stop its being decrypted twice further on)
- Call read user file ()
- Decrypt user file
- Add one to the record number
- Generate user name
  - Add the fist letter of first name to the surname
  - Add the record number to the Surname with the letter
- Set as the username
- Preserve and increase the length of the structured array holding the user data by 1
- Set the structured array positon of the (New) record number to hold the details in the text box. (e.g. set structured array 'password' positon 'record number' to textbox password)
- Set the structured array access level to 3
- Set the structured array Username to the generated username
- Close the file
- Encrypt File
- Sent a message to the user, saying there account has been saved, and telling them their username.

### Validation

- Set Proceed value to True
- If there is no text in any of the boxes
  - Set proceed to false
  - Show message box saying to fill all the fields
- If the email validation is incorrect
  - Set proceed to false
  - Show message box saying to enter a valid email
- If the password doesn't have a capital letter, lowercase letter or number
  - Set proceed to false
  - Show message box saying to Enter a strong password
- If the two passwords entered don't match
  - Set proceed to false
  - Show message box saying to ensure passwords match
- End

#### When Sign Up button is pressed

- Call validation()
- If proceed is true
  - Call save users into file()
  - Clear all text boxes regarding user info on sign up form
  - Send a message box saying welcome with username
  - Open the login page
  - Hide this page
- Else if proceed is false
  - Message box the details you entered were not valid
- End if statement

Welcome Page-

#### When Photo/button is pressed:

- Load up the corresponding page

#### Upon Page loading up

- Open File "Admin Message"
- read all of the file
- Set a textbox to hold the text read from file
- Close the file
- Start ticker timer

#### Upon timer ticking

- Set picture number to picture number + 1
- If the picture number is over 3
  - Set Picture number to 1
- End if statement
- For all three picture boxes
  - Set the image name to 'picture box name' & 'picture number'
  - Set the image in the correct box to 'image name'

Admin Page –

### Edit Save

- Reads user file
- Re - dims the structured users array to the same number of records as before (loads information in)
- Sets 'user to change' as the record number in the text box for ID number
- Set structured array at the position 'user to change'
- Change all the parts of the structure at this position to the corresponding text boxes (e.g. set username in the structured array to the text in TBUsername.)
- Open User data file
- Write line with the number of records on it
- Start a loop, with an incrementing counter
  - Each time the loop runs the counter increases
  - Write to the file the structured array at position counter
  - (do this for all elements of the structure)
- Close the file
- Encrypt the file

### Email search

- Read the user file
- Clear any text in the display box
- Loop with incrementing counter
  - If the Users structured array email at 'counter' is the same as the email in the search text box
    - Add the all the structured parts of the array at the position counter to the display box
  - End if statement
- Loop back until all of array has been read

### User Name search

- Read the user file
- Clear any text in the display box
- Loop with incrementing counter
  - If the Users structured array username at 'counter' is the same as the given username in the search text box
    - Add the all the structured parts of the array at the position counter to the display box
  - End if statement
- Loop back until all of array has been read

### Surname search

- Read the user file
- Clear any text in the display box
- Loop with incrementing counter
  - If the Users structured arrays surname at 'counter' is the same as the surname entered to the text box
    - Add the all the structured parts of the array at the position counter to the display box
  - End if statement
- Loop back until all of array has been read

### Display User

- Clear Current User display box
- Add items from the users structured array at the position equal to the 'logged in user id number'
- Change the password to "password not shown"

### Display User

- Clear Current User display box
- Add items from the structured array from the first position the the last in the file

### When the selected item of the main display box is changed

- Check if the selected item of the display box is numeric
- If its numeric
  - Change the ID Textbox to the selected number
  - Set all the remaining text boxes to the corresponding data located in the users structured array (at the same position as the ID number in the textbox)
- If its not a number
  - Send a message box saying to "select a record number from the selected list"

### Email Before File is Changed

- Get Id number from ID text box
- Make a copy of all the ID's details (Username, password, email, etc)
- Setup an email using Outlook
- Set the Email address to be sent to
- Set the emails body to hold the copy of the ID's data, and the details of the user who changed it

- Send the email
- Send a message box to user saying the email has been sent off

### **Teacher Page-**

#### Upon Page load up

- Load all the combo boxes with data by:
- Reading the corresponding file
- Using a counter until the end of the file
  - Look at the current data in the combo box
  - If the information to be entered into the combo box is already entered
    - Read the next piece of information
  - If not already in the box
    - Enter the information

#### Read Test file

- Read the first line in the file
- Set the first line as the number of records in the file
- Set the structured array to hold the number of records
- Set the counter to 0
- Loop until the file has been read fully
  - Read the lines of the file that match to the structure of the array
- Close the file

#### Layout (search "all" to file)

- Read the Test Layout Data file
- Clear the display box
- Using a loop with incrementing counter
  - Add the structured array in order for the position of the counter
  - Displays the rest of the structured array
- Loops back until array has been finished

#### Save Lessons

- Check to see if the name of the reduced has already been created
- If it already exists
  - Send an option to conform if they want to overwrite or not
  - If they do want to overwrite

- Re write the file as normal, but save over the record with the new information (deleting the old, but keeping the name)
    - If not, end
  - If a new name has been entered
    - Add the new file
    - Perform a bubble sort to ensure its in alphabetical order
  - End if
  - Reload the combo box with the lesson names

### Bubble sort lessons

- Find the length of the lesson array
- Loop to here
  - Set has swapped to false
  - For loop, length of the array
    - Check the name of the lesson at postposition n in the array, with the position n+1
    - If the name is "bigger" (further along in the alphabet) than the next
      - Save the record at position n in a temporary location
      - At position n, save the record of n+1
      - At position n+1 save the record stored in the temporary location
      - Set swapped equal to true
    - End If
  - End for loop
- Loop to the top until swapped is false
- Once the array has be sorted, resave the array into the text file

### Delete Test Button is Pressed

- Read the current file with the test layout information
- Set found to false
- Set an integer (add val) to 0
- Resize the temporary array created to hold one less structure than the original
- Loop here for an increasing counter, do until counter is the same as the size of the original file)
  - if the name of the test at position 'counter' is the same as the name to be deleted
    - change found to true
    - set 'add val' to one
  - End if

- Set the temp structured array holding the test data at position 'counter' equal to the data held in the original structured array at position 'counter + add val'
- Do this for all the fields in the structured array apart from the ID number, set the ID number as the same as the original.
- Loop here and increase the counter
- Clear the display box
- Loop to here until 'counter' is the same as the number of records – 1
  - Add in the data in the temporary array for all the structure at position 'counter' to the display box
- Loop back until counter is the same as the number of records – 1
- Open the 'Test layout' text file
- Write the number of records (minus one) into the file
- Loop to here until counter is equal to the number of records for the temp array
  - On each line write one field of the structured array with the record number
  - Repeat for all the structures
- Return to top of loop
- Close the file

If the delete lesson button is pressed

- Read the learn file
- Set 'add value' to 0
- Set found value to false
- Resize the temporary learn array to the size of the current learn array minus two
- Loop back here until the counter is the same as the learn array size minus one
  - set 'counter' to itself add one
  - if the learn arrays name at position 'counter' is equal to the selected name to be deleted
    - set found to true
    - set add value to one
  - end if statement
  - make the learn structure at position 'counter' equal to learn array structure at position 'counter + add value'
- loop back
- clear items in the display box
- loop to here with an incrementing counter
  - add item to display box – learn array at position 'counter'
- loop
- open text file holding learn data
- write the number of records
- save each record into the file on a new line
- close the file

## Test Page –

### Save test data

- open the file to be read
- read the first line to get the number of records in the file
- close the file
- call read test data ()
- add one to the number of records
- increase the array size by one, keeping the old data in it
- in the structured array, at the position created, add the new data (score, mark, username, test)
- open the file
- save the new record number
- save all the positions of the array into the file
- close the file when complete

### Read test data file

- open the file
- read the first line, save this as the number of records in the file
- set the counter to zero
- loop to the end of the file
  - add one to the counter
  - at the position 'counter' in the structured array, input the data from the line read in the file
  - do this for all the fields in the structure
- loop back
- close the file

### Mark score

- create an integer to hold the number of boxes checked by the user
- loop five times with an incrementing counter
  - if check box A has been checked
    - add one to amount of boxes checked
    - change the colour of the box to red
  - if check box B has been checked
    - add one to amount of boxes checked
    - change the colour of the box to red
  - if check box C has been checked
    - add one to amount of boxes checked
    - change the colour of the box to red
  - if check box D has been checked
    - add one to amount of boxes checked
    - change the colour of the box to green
    - add one to the students score
  - end if statement
- loop back
- if the amount of boxes checked is over 5
  - message the user to tell them they have selected too many boxes and need to re take the test
  - set the score to zero
- end if

### Upon the 'start button' being clicked

- read the test layout file()
- layout the selected questions
- start the timer
- make the 'save and finish' button visible

### Upon timer tick

- Add one to the 'timer counter'
- Make a text label show the 'timer counter'

### When the 'save and finish' button is pressed

- Stop the timer
- Mark the test ()

- Send a message box saying the users score
- Save the data into the text file ()
- Reset the timer and timer counter to zero
- Make the save button invisible

#### When the page loads up

- Clear all the items in the combo box used to select the test
- Read the test layout file ()
- Loop for the number of records in the file
  - add one to a counter
  - if the name in the structured array at position 'counter' is not the same as any item in the combo box
    - add the name of the test to the combo box
  - end if
- loop back to top

#### **Algorithms page –**

##### Bubble sort with set array

- find the set arrays length
- loop back to here
  - set swapped to false
  - Loop for every section in the array (with incrementing counter)
    - If Bubble array 'counter' is greater than Bubble array at the position after it (counter + 1)
    - Then hold the data in the array at 'counter' in a temporary store
    - Set the array at position counter to the one after it
    - Set the array at position of counter + 1 to the data in the temporary store
    - Set swapped to true
    - End if statement
  - Loop back to array loop
- Loop back to bigger loop while swapped is true
- Add all contents of the array to the display box

##### When 'Add' button is pressed

- If the 'Array Counter' is smaller than or equal to the chosen array size

- if the textbox the user has entered the information to holds a number
  - the New Array at position 'array counter' holds the number in the textbox
  - clear the textbox for the net number
  - increase the 'array counter' by one
- if it does not hold a number
  - send a message box telling the user so enter a number
- end if
- if the counter is bigger than the array size
  - send a message box to tell the user the array is full
- End if

#### When the 'up down' numeric counter is changed

- If the value of the UpDown picker is greater than 30
  - Send a message to the user saying this is too big an array to have
  - Change the value of the number picker to 0
- If not
  - Create length of new array by taking one from the users number (save this number as 'len')
  - Set the new array size to 'len'
  - Set the 'array counter' to zero
- End if

#### If check box is checked

- Make the 'add number' button visible
- Make input box visible
- Make the Numeric up down visible

#### If the button 'Bubble' has been pressed

- Clear all the items in the display box
- if the check box has been checked
  - run 'bubble with made array'
- if not
  - run 'bubble with set array'
- end if

#### If button 'step' is pressed

- Clear all the items in the display box
- if the check box has been checked
  - run 'step with created array'
- if not
  - run 'step with set array'
- end if

#### Step with set array

- find the length of the array save value as 'array length'
- loop with 'counter' increasing by one each time until counter reaches array length
- If Bubble array 'counter' is greater than Bubble array at the position after it (counter + 1)
  - Then hold the data in the array at 'counter' in a temporary store
  - Set the array at position counter to the one after it
  - Set the array at position of counter + 1 to the data in the temporary store
- End if statement
- End loop
- Clear the display box
- Go through the array position by position and add it to the display box

#### Fibonacci sequence

- Set number Z as 0
- Set number B as 1
- Set number A as 0
- Loop for X times
  - Set Z to the sum of A and B
  - Set A equal to B
  - Set B equal to Z
  - Add Z to the display box
- loop back

#### Fibonacci step

- Set Z to the sum of A and B
- Set A equal to B
- Set B equal to Z

- Add Z to the display box

#### Binary search setup

- Set search for as the text entered by the user
- Set the 'starting number' as 1
- Set the 'end number' as the length of the array
- Set 'over counter' to 0
- Display all the data in the set array into the display box

#### Binary search

- Loop while found is false
- set 'looking at' to the (sum of the 'start number' and 'end number') divided by two
- if the array at position 'looking at' holds the text being searched for
  - show the user a message box saying it has been found
  - set found to true
- if the array at position 'looking at' is smaller/lower than what is being looked for
  - set the 'start number' as 'looking at' + 1
  - Add one to the 'over counter'
- If the array at position 'looking at' is bigger/greater than what is being looked for
  - Set the 'end number' as 'looking at -1
  - Add one to the 'over counter'
- End if statement
- Calculate the maximum number of 'passes' that could be done to find an item in the list (log base 2 of array length)
- If the 'over counter' is greater or equal to the number of passes
  - Set found to true
  - Send a message box to the user telling them it could not be found
- End if
- Loop back to top

#### **Bot page collision mode –**

##### If 'extract' button is pressed

- Open filename in textbox
- If filename does not open / if not a tex file / not entered
  - Send a message box to user saying to enter a correct file path and name
  - Make the 'show path' button invisible
- If the filename is accepted
  - Make the 'show path' button visible
  - Clear the display box
  - Read the file to get the number of lines in the file
  - Add the total number of lines into the display box

- Turn the contents of each line of the file into a part of an array
- Add all the contents of the array line by line into the display box
- End if statement

#### If 'show path' button is pressed

- Loop for the length of the created array with an increasing 'counter'
- If the counter is not a multiple of three
  - Open up the graphics resource
  - Draw a black line in the picture box from the coordinates 'a' to 'b'
  - set the 'startX' coordinate to the 'finishX' coordinate
  - set the 'startY' coordinate to the 'finishY' coordinate
  - Run direction check()
- Else if line is a multiple of three
  - Set the previous time to the 'old time'
  - Find the new time in the array set it to 'hold time'
  - Run change points()
  - Set 'a' as coordinate, with the X = to 'startX' and the Y= to 'startY'
  - Set 'b' as coordinate, with the X = to 'FinishX' and the Y= to 'FinishY'
  - Draw a blue line from 'A' to 'B'
- End if
- Loop back until all the array has been completed

#### Direction check

- Create an integer called 'hold'
- read the array at position 'counter' set the number found to 'hold'
- if hold is bigger than 'right turns'
  - Add one to 'NESW'
  - Set 'right turns' to 'hold'
- If hold is bigger than 'left turn'
  - Take one from 'NESW'
  - Set 'left turns' to 'hold'
- End if statement
- Check if NESW has gone over 4 or below 1
  - Loop it back round so it always 1/2/3/4 (cycle to other side dependant on situation)
- End

#### Change points

- If 'NESW' is 1
  - Subtract the difference in the 'hold' and 'old' time from the last 'Finish Y'

- Set this as the new 'Finish Y'
- If 'NESW' is 2
  - Add the difference in the 'hold' and 'old' time from the last 'Finish X'
  - Set this as the new 'Finish X'
- If 'NESW' is 3
  - Add the difference in the 'hold' and 'old' time from the last 'Finish Y'
  - Set this as the new 'Finish Y'
- If 'NESW' is 4
  - Subtract the difference in the 'hold' and 'old' time from the last 'Finish X'
  - Set this as the new 'Finish X'
- End if statement

### **Bot program page –**

#### When the 'Up' Button is pressed

- Add one to the array size
- At the new position write "Up"
- Check the 'NESW' direction of the front of the bot
- Then move the "bot" button +20 in the correct direction

#### When the 'Down' Button is pressed

- Add one to the array size
- At the new position write "Down"
- Check the 'NESW' direction of the front of the bot
- Then move the "bot" button -20 in the correct direction

#### When the 'Left' Button is pressed

- Add one to the array size
- At the new position write "Left"
- Change the 'NESW' value by subtracting one
- Loop 'NESW' back if its out of range (1-4)

#### When the 'Right' Button is pressed

- Add one to the array size
- At the new position write "Right"
- Change the 'NESW' value by adding one

- Loop 'NESW' back if its out of its range (1-4)

#### If the button 'save and finish' is pressed

- Check the filename/path given is appropriate
- If the filename leads to a text file
  - Open the file
  - Read in line by line each part of the created array
  - Close the file
  - Send a message notifying the user the data has been saved
- If the filename does not work
  - Send a message asking for a working filename/path
- End if statement

#### Prototype:

The prototype of my system contains most of the key features that will be in the finished system. I've ensured that I was able to complete the core components of the system, and ensure that the other features are feasible. The ground work has been done ready for the final parts to be added and refined.

With this demo, all the possible users of the system can access and use the core parts of the system, this will allow them to see the systems capability and use for help as an educational tool, as well as allow them to give feedback that can be used later to refine and improve the program before its final release.

#### Student Users Can:

- Sign Up by creating an account
- Sign into the program
- Open up lessons written by teachers
- Open videos (links given by teachers) from indie the lessons
- Email a teacher to request help
- Load up and take a test set by a teacher
- Use the bot
- Use data collected by the bot to show its path on screen
- Program the bot and set it off to move
- Access a few simulations/examples of code (bubble sort etc.)

#### Teachers Can:

- Create a new test, with answers
- Create and enter new lessons for the students

- Load up and edit the lessons
- Look at the student's test results
- Search through the student's results

Admin's can:

- Access all the data in the system
- Search students from Name, Email, Username
- Edit student's data if needed
- Change the message displayed to all the users on the welcome page
- Access all content student and teachers can

The bot can:

- Move around a room, bumping into obstacles and recording the data if which sensor and what time it was hit
- Have a set of instructions generated by a user on the main program for the bot to decode and follow.

The main program:

- Encrypts the 'Users data' when not in use
- Saves the students scores on tests
- Binary searches the Lessons (to speed up the process with many lessons)
- Sorts the lessons upon a new one being added/created
- Keeps track of which user is 'logged in' to correspond with saving test scores and keeping features restricted (access level)
- Marks the tests the students have finished, colour code the errors/answers.

I decided to include these features as I believe these to be the core components of my project. The login system allows for a hierarchy of authority, separating the students and admins, but it allows allows the program to keep a reference of which user just took a test, and in a later version of the program I can allow it to keep track of which admin changed what data. But for now its just there as a sort of prof of concept and use for the main part.

The other features I have added are also equally important, the students would have nothing to lean if the teachers could not make and save lessons. The test could not be taken if the students didn't learn anything to take a test on. But I still have more work to do with the tests and scores, as I still need to randomize the answers on the test, into a non predictable pattern, but I have not done this yet, as I determined its not a core requirement as this stage of the project. All the features I have currently added provide either a core function, or prof of concept the the rest of the program to be built upon.

The demo does contain most of the core features however it is still missing parts, I decided to not include some features from finished main program, as I wanted a working prototype to demo to potential users first and obtain feedback to improve and fix at this stage of development, rather than later on, when it would be harder to change a feature without completely changing the design and functions.

I left out some features of the final system for a few different reasons. For some, the time to add in a core feature, was taking longer than expected, therefore I felt it best to push the other requirements back and focus on the core to produce the working demo. For other features, I had already done something very similar in my program, so there was little need to repeat it for my demo, as I know the code already works, and I have to just implement it later on, such as the feature of keeping track of what and who changes data on the 'user data' file, in the finished system I will have an email sent with the data that's been changed, but this is not on the demo as I already have an email feature for students to get help from teachers, which performs a very similar task. This email, as well as the logged in data would both be used together to make the changed data email. In order for a working demo to be produced on time (in order to leave time for feedback and refinement) I did have to leave out some features for different reasons.

On the bot I decided not to include a mode and a feature I was going to add in the final system. I decided to leave out both, a form of physical output (visual or audible) and a mode of the bot to calculate the size of room from using an algorithm. I decided to leave these features out of the demo, as I already have a two modes for the bot react for the demo, I have already shown the bot can move around and store data, so making the mode change isn't a huge difference from what I already have. With the output of the bot, I decided to leave this as I might need to change the hardware on the bot after the demo (if the users find fault with it,) therefore I'll only add the output as a sort of bonus if there is still room, as it won't break/ruin the project if it's not added, but not adding it for now give me a little 'wiggle room' with my current setup of the Arduino board.

### **Screen shots:**

Areas include in the prototyping system:

- Login Form
- Sign up form
- Welcome page
- Admin page
- Teacher page
- Bot mode selection page
- Collision mode data extraction
- Program mode
- Learn Page
- Test page
- Algorithms/simulation page

## Sign in form

To login:

- 1) Login – The user will type in their Username and password into the corresponding box. The click sign in.
- 2) If the user does not have an account, they can create a new one by pressing 'sign up' to open the correct form.

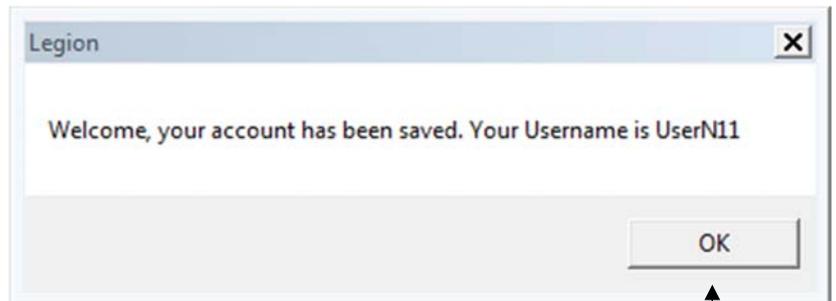
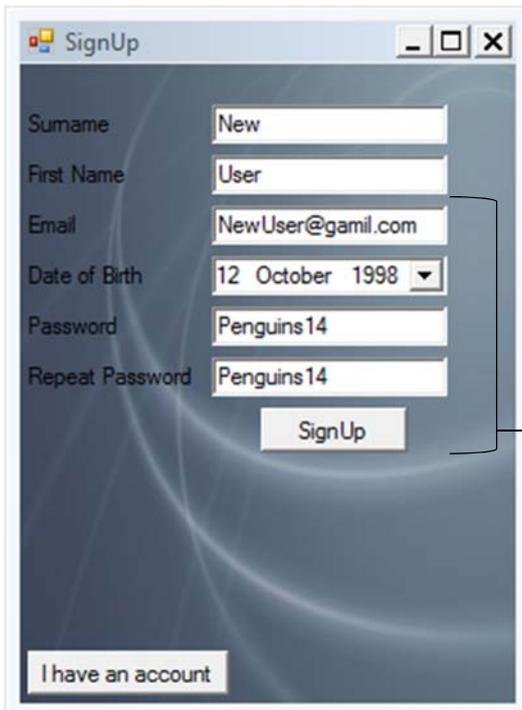
Enter the Username and password here.



## Sign Up form

To make a new Account:

- 1) The user enter their datils, (name, emial, etc) The program will alreat them if what they have entered is not valid (e.g more secure password)
- 2) The user clicks sign up, their details (if valid) are saved, and their username will show in a pop-up message box.



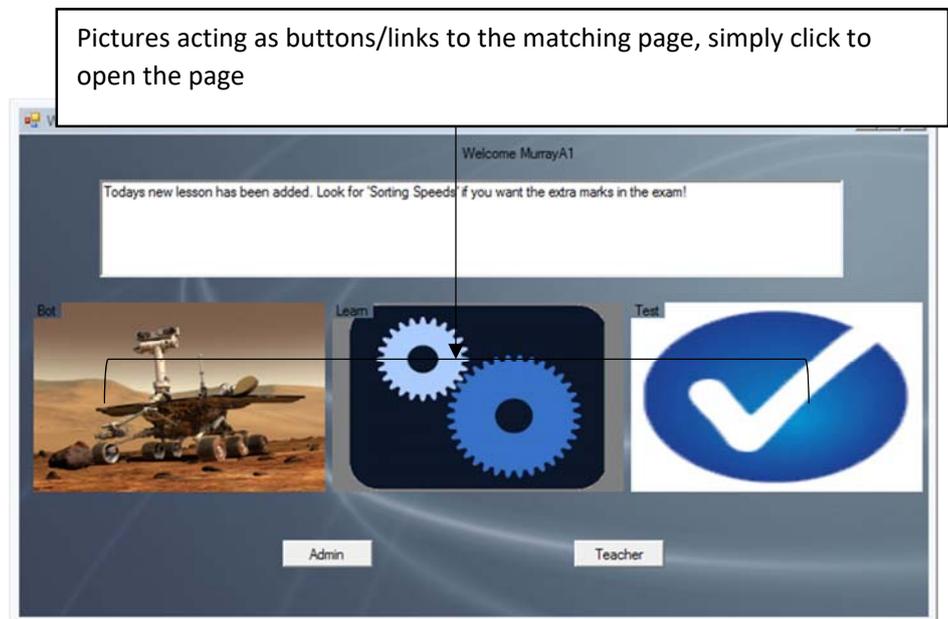
User enters data in corresponding text boxes

A message box showing the users new username will appear.

## Welcome page

On this page the user can select which page/service they want by clicking on the pictures of the page they wish to access.

Admins and teachers also have the option of the extra features of the admin and teacher page that can be accessed via buttons only useable by the admin and teachers.



## Admin Page

Locating/searching for a user:

- 1) If the users Email, Surname, or username is known, enter it into the search box
- 2) The user selects which field they are searching for, and then presses the 'search' button
- 3) The search result data is displayed in the display area, in order of oldest to newest.
- 4) If a general search is needed the admin can display all the users at once using the 'display all' button

Changing the message on the welcome page:

- 1) The user enters the desired message into the text box in the message section
- 2) If the message is correct the admin clicks the 'change button' to save and display it.

Changing a Users details:

- 1) The Admin must first have searched the user; (any search method will do.)
- 2) The Admin can then click on the record number of the user they wish to change. The record number will be shown in the display area. This will load up the users details into the correct fields in the textboxes next to the display area.

- 3) The Admin can edit and change all the data in the boxes apart from the record number.
- 4) Once finished the admin presses 'Save changes' to save the new data (this will re-write the old record and enter the edited/new one)

The screenshot shows an Admin interface with several key components:

- Search Section:** Located at the top left, it includes radio buttons for 'Email', 'Surname', and 'Username', a search input field, and a 'Search' button. A callout box states: "Known details can be entered here and the search mode selected with the radio buttons".
- Message Section:** Below the search section is a large text area labeled 'Message' with a 'Change' button below it. A callout box explains: "Textbox where the admin can enter/change the message displayed to all the users at the welcome page."
- User Selection and Editing Section:** On the right, there is a list of users. A callout box notes: "Once a record has been selected, these fields will be automatically filled, ready for editing." Below the list are fields for date (12 October 1998), a numeric field (1), and a 'Save Changes' button.
- Display Section:** To the right of the editing fields is a large empty box. A callout box indicates: "Display are: User details and record numbers will be shown here." Below this box is a 'Display All' button.

### Teacher Page

Checking a student's score:

- 1) The Teacher selects the search mode they wish to use, (via radio buttons)
- 2) They then fill in the required field (if necessary) and press the 'search button'
- 3) The search results are shown in the list box next to the radio buttons.

Making a new lesson:

- 1) The Teacher enters a lesson name into the drop box.
- 2) They then enter information/text/video links about the lesson
- 3) Once the lesson had been finished they press the 'save' button

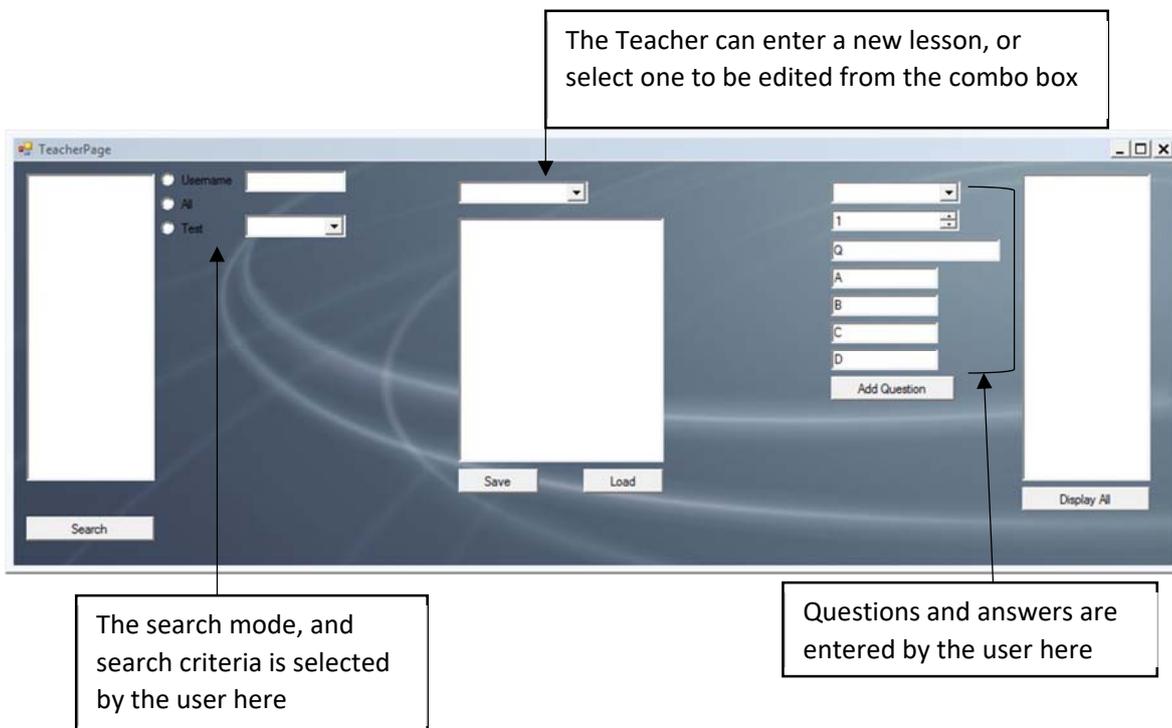
Loading/editing a lesson:

- 1) The teacher uses the combo box to select the lesson they wish to open (they will see a list upon clicking the dropdown arrow)

- 2) They can then load up the saved data by clicking the 'load' button, this will load all the data for the lesson into the multiline textbox below the combo box
- 3) The teacher can edit the lessons information as they see fit
- 4) Once finished they can press the 'save' button to save the changes they made
- 5) A message box will appear asking the user if they wish to continue with overwriting the data, the teacher can select yes to continue, or no to cancel.

#### Making a test:

- 1) The Teacher can select which test they wish to add a question to, or they can enter a new name to make a new test, at the combo box at the right of the page.
- 2) After selecting/entering a name, the teacher enters the question into the larger textbox (the question number is done automatically but can also be changed if required)
- 3) The teacher then enters the 'fake' or wrong answers into the next three textboxes below, these are dummy answers that are supposed to trick/make students think about the correct answers.
- 4) The correct answer is placed in the final textbox, and this is the one that counts as being marked correctly.
- 5) The teacher presses the 'add question' button to save it.

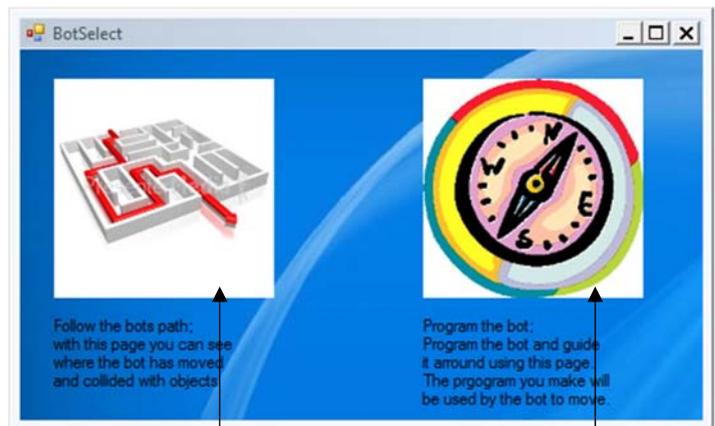




The teacher has to press yes to continue saving, when this message appears.

### Bot Page Selection:

The user selects which page they wish to open from this form, they are given two options in the demo, similar to the welcome page, the photo acts as a button, and the user clicks/presses it to open the next from.

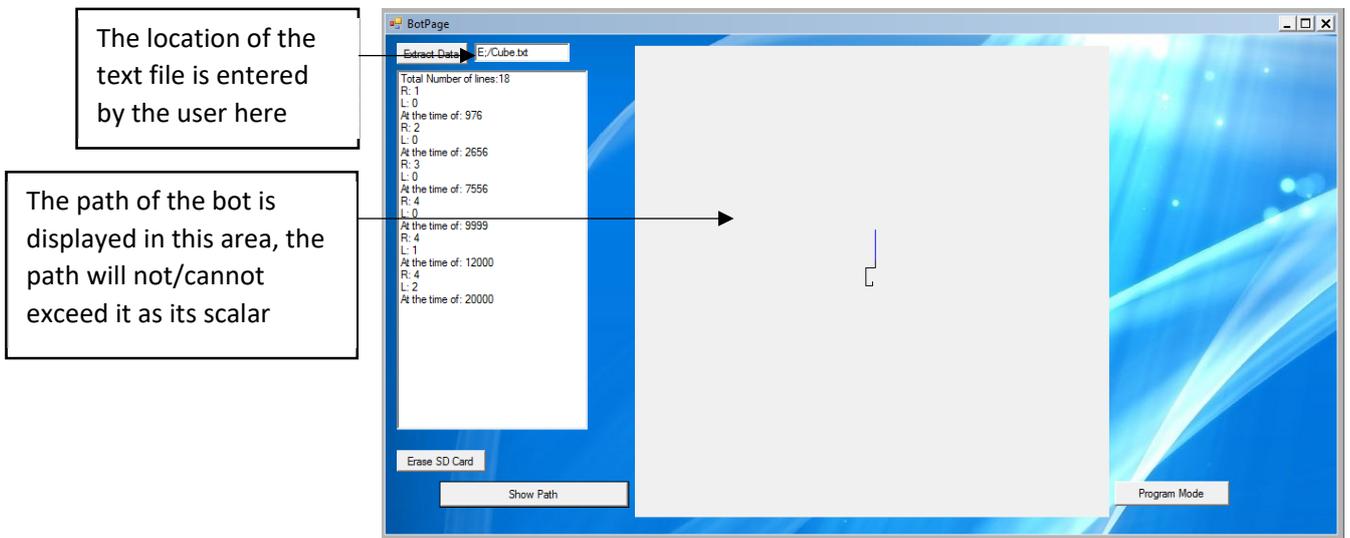


The user clicks these photos to enter the page/mode they want. A small description helps guide them.

### Collision Mode

Showing the bots path:

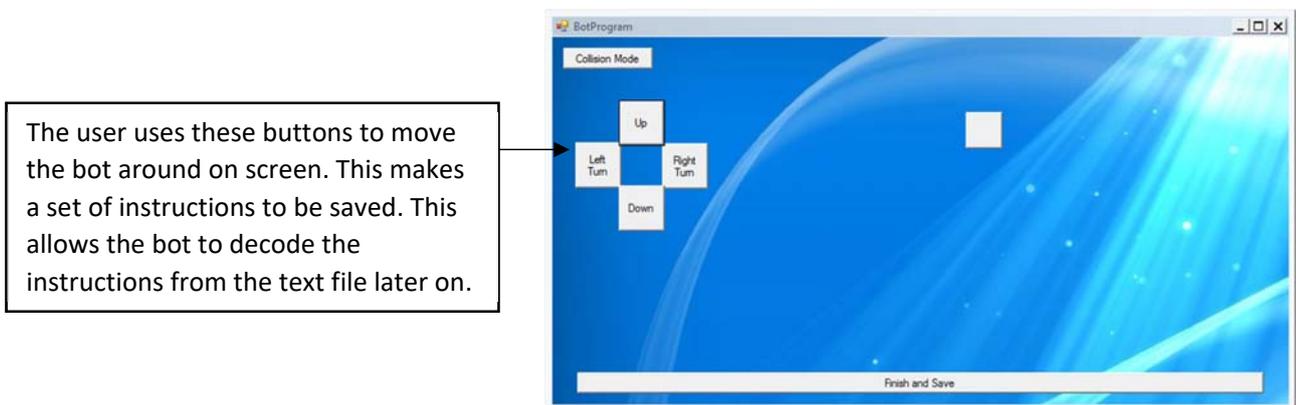
- 1) The user enters the location of the text file that contains the data stored from the bot
- 2) After the correct location has been entered the user presses 'extract data'
- 3) Data from the bot is then shown at the side, the user can erase the data (if they are finished with it, or displeased with the data they have collected)
- 4) If the users are happy with the data they can press the 'show path' button, this will convert the data provided into vectors that are displayed on screen.



### Program Mode

Make a program for the bot to follow:

- 1) The user opens up the page and sees a small a square in the screen
- 2) They can then use the buttons on the left of the page to move the square around the screen (imitating the bots path)
- 3) Once they are happy with the path they have made, they can press finish and save
- 4) This will prompt the user to enter the location of the text file they wish to save to.



### Learn Page

Open up a lesson:

- 1) The user uses the drop box at the top left of the page, to reveal the list of lessons

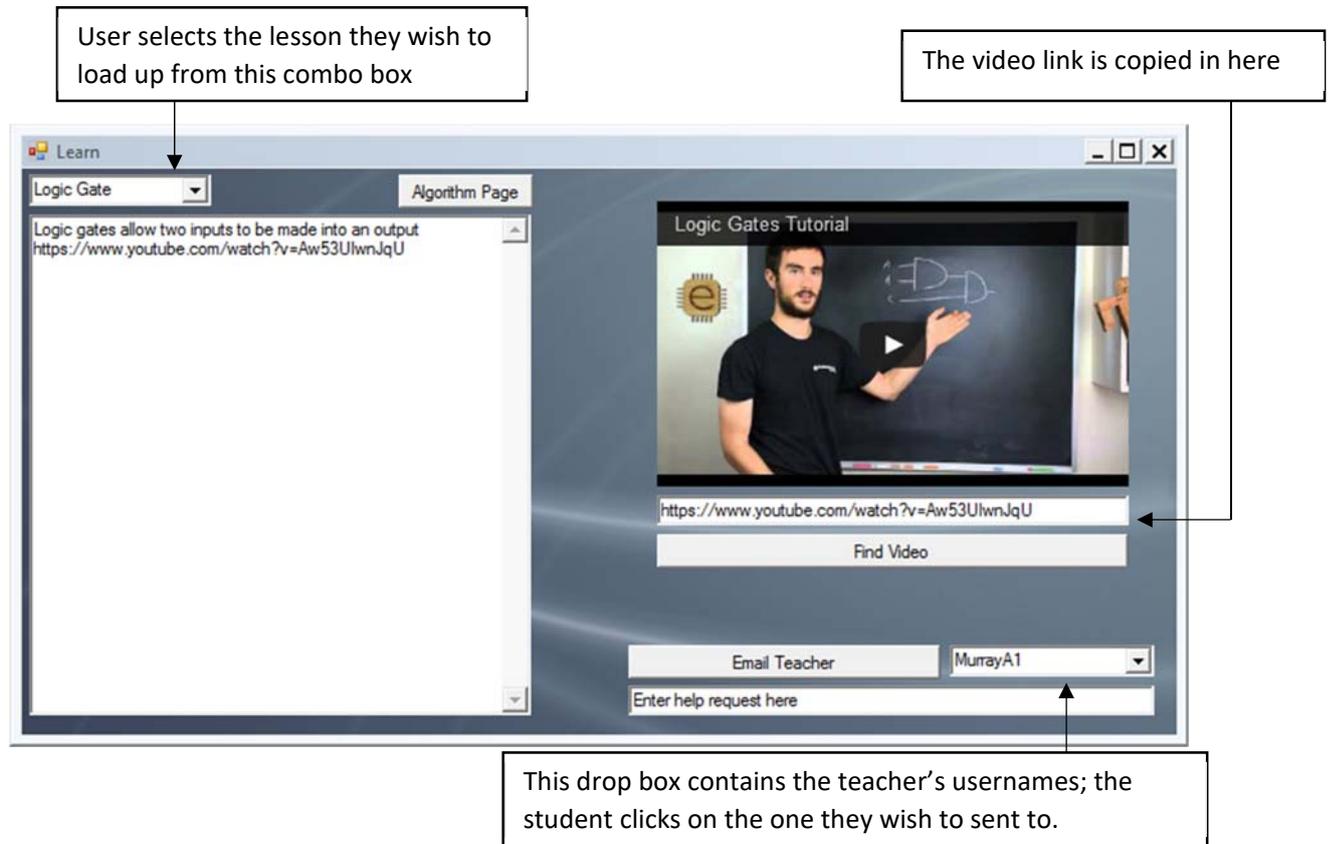
- 2) If a lesson is selected from the list by a user, it will automatically load up into the multiline textbox below.

View a video link:

- 1) If a link was left in the lesson by a teacher, the student can copy and past in the link to the bar below the video screen.
- 2) Next the user pressed the 'find video' button, and the program loads up the video from YouTube
- 3) If the video link is correct the video loads and the user can press the play button to view it normally while still on the learn form.

Email for help:

- 1) The user selects the teacher they want to ask for help from by selecting the teacher's username from the combo box
- 2) Then they enter in their message to the teacher in the textbox below.
- 3) Once finished they press the button 'Email teacher' to send it.



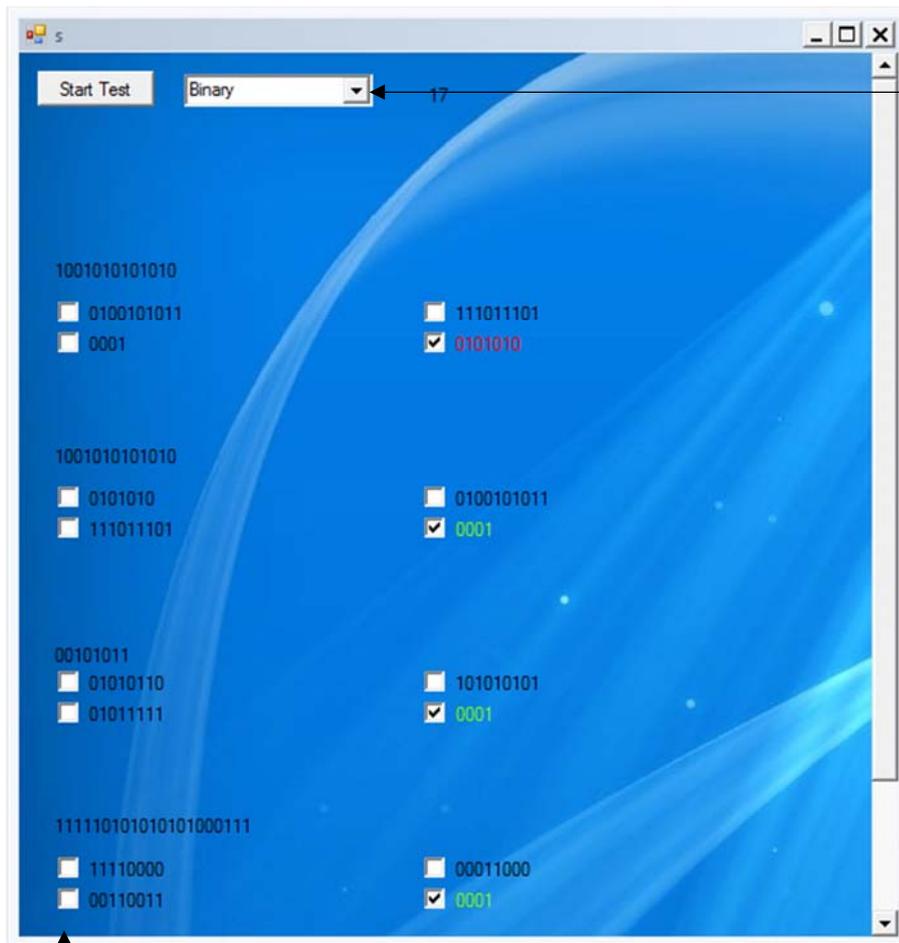
## Test Page

To load up a test:

- 1) The student chooses a test from the drop box
- 2) When one had been selected the data loads in automatically

To take a test:

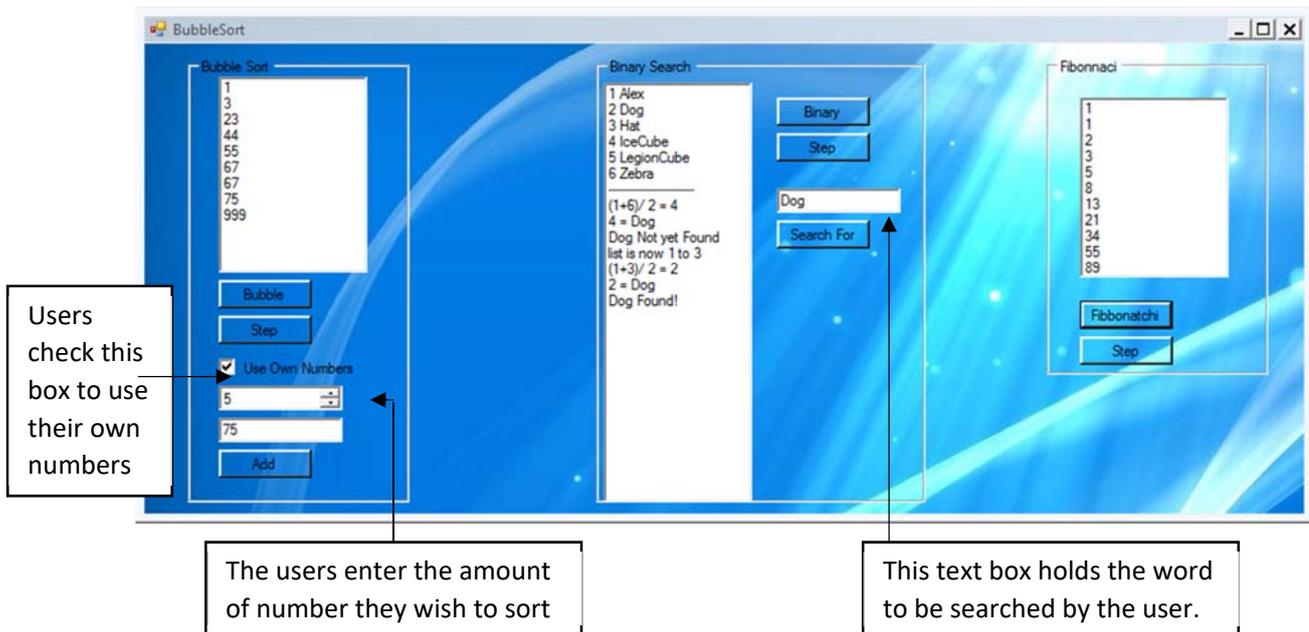
- 1) The user has to load up a test first
- 2) Once the student is ready they press the 'start' button
- 3) Using the check boxes the student click on the one they believe to be the answer
- 4) At the end of the test (once the student has finished) they press the 'mark and save' button.



The drop box provides a list of all the available tests for the user to take.

The 'mark and save' button is at the bottom of the page, the user, must have seen all of the questions before they can access it.

Check boxes can be selected by the user for each question.



## Algorithms

Using the bubble sort:

- 1) The user can perform a bubble sort on pre-set numbers by pressing 'bubble'
- 2) They can watch the sort step by step on the pre-set number by pressing 'step'
- 3) The user can add in their own numbers by checking the checkbox
- 4) Then setting the numeric counter to the array length they desire
- 5) And entering in the numbers they wish into the text box below until the array is full
- 6) Then they can use the sorting buttons above as before .

Binary search:

- 1) The user enters the word they wish to search for into the textbox in the Binary section
- 2) They then press 'search for' button to enter the word into the search
- 3) They can then either see the whole process at once (by clicking 'binary') or see the search step by step by pressing the 'step' button

Fibonacci:

- 1) To see the sequence, take place, the user can press 'Fibonacci' to have the sequence shown at once
- 2) The user can use 'step' button to see the numbers being added bit by bit.

### Evaluating the prototype system

Now that I have finished my prototype, I will need to obtain feedback in order to see if my demo features are up to the standards they should be at. In order to do this, I will need to have a range of different users (for each different type of users in the system e.g. admin) and gather feedback from them all in questionnaires.

The users who tested my prototype were all given the same questionnaire, but the students stopped once they saw the teacher questions, and the teachers stopped the admin section. In the questionnaire I asked the users to rate different aspects of my prototype from one to five. With one being the best/agree/positive outcome and five being the bad/negative. I decided to use a number marking system over a 'agree/disagree' as this made the marker less 'guilty' if they were going to mark the system down and therefore will be more honest, I also allowed for a 'middle' value as this will show me if my work has been on the right track, but is missing the extra something.

Question	1	2	3	4	5
As a new user to the system how easy was it to make a new account?	36%	42%	15%	7%	0%
How easy was it to log into the system once an account had been made?	73%	24%	3%	0%	0%
Did the home/welcome page make navigating the program for the first time easy?	10%	28%	40%	22%	0%
Was the program easy to navigate after the first time?	2%	6%	22%	43%	27%
When using the bot, did you find the practical side of having a physical bot entertaining?	44%	48%	5%	3%	0%
How easy was the bot to use?	12%	68%	10%	7%	3%
Would you want to have more modes/features with the bot?	40%	35%	15%	10%	0%

Did you understand what the bot was doing and how it worked?	9%	34%	30%	24%	3%
Would you have enjoyed different types (or more) bots?	33%	42%	24%	1%	0%
Was it easy to open up a given lesson and read the contents?	70%	24%	5%	1%	0%
When loading up a lesson, did the lesson load up within a suitable time?	77%	19%	3%	1%	0%
Did you find the video feature helpful/useful when a video was linked?	43%	48%	6%	3%	0%
Did you enjoy the use of the simulated algorithms?	65%	10%	20%	2%	3%
Do you feel they helped with education?	63%	12%	20%	2%	3%
Should there be more use of the algorithms and simulations?	50%	15%	26%	5%	4%
Was it clear on a test where the answers were correct/incorrect?	81%	19%	0%	0%	0%
Did you find the tests were the correct length/size?	40%	32%	23%	5%	0%
On the whole, how fun was the system to use?	39%	36%	25%	0%	0%
Would you use this system again?	42%	33%	23%	2%	0%
<b>Teachers and admin only</b>	-----	-----	-----	-----	-----
When making a test, was it easy to add a new one?	22%	46%	29%	3%	0
Was having all the teachers 'tools' on the same page useful/important?	25%	23%	19%	24%	9%
Was the system easy to use once you had been given a short demonstration?	52%	34%	10%	4%	0%
When adding a lesson did you feel the text area given was suitable?	63%	71%	6%	0%	0%
Did you find it easy to check a students test results?	18%	20%	25%	20%	17%
Would you use this system to help with education?	48%	49%	2%	1%	0%
<b>Admin only</b>	-----	-----	-----	-----	-----
Did you feel you had suitable control over the system?	39%	51%	9%	2%	0%
Was it easy to enter in a message to the main page?	86%	10%	4%	0%	0%
Was the method of populating the fields when editing a users data useful or helpful?	72%	25%	3%	0%	0%
Was it easy to change any data/information about the user (e.g. change password)?	72%	27%	1%	0%	0%
Did you find the ability to use the 'teachers' side of the project necessary and or useful?	67%	32%	1%	0%	0%

### Questionnaire analysis:

With the results of the questionnaire, I was able to go through and look at the positive and negative results for the questions I asked. When I sent my questionnaire I used a numbering system where one was positive result, and 5 a negative, with 3 being a neutral point. This allows me to better understand the results of the questionnaire, as instead of having a hard 'Yes' or 'No' I get to see patterns with how the users of the system scored it.

After looking at the results I've used a color code to mark where the feedback indicated a positive or negative point in the demo of the system. The green results show positive response, therefore these questions show where the system performed as intended. Yellow shows where improvement is needed, and red shows where the system did not fulfil its needs/requirements.

Question	1	2	3	4	5
As a new user to the system how easy was it to make a new account?	36%	42%	15%	7%	0%
How easy was it to log into the system once an account had been made?	73%	24%	3%	0%	0%
Did the home/welcome page make navigating the program for the first time easy?	10%	28%	40%	22%	0%
Was the program east to navigate after the first time?	2%	6%	22%	43%	27%
When using the bot, did you find the practical side of having a physical bot entertaining?	44%	48%	5%	3%	0%
How easy was the bot to use?	12%	58%	10%	17%	3%
Would you want to have more modes/features with the bot?	40%	35%	15%	10%	0%
Did you understand what the bot was doing and how it worked?	19%	34%	30%	14%	3%
Would you have enjoyed different types (or more) bots?	33%	42%	24%	1%	0%
Was it easy to open up a given lesson and read the contents?	70%	24%	5%	1%	0%
When loading up a lesson, did the lesson load up within a suitable time?	77%	19%	3%	1%	0%
Did you find the video feature helpful/useful when a video was linked?	43%	48%	6%	3%	0%
Did you enjoy the use of the simulated algorithms?	65%	10%	20%	2%	3%
Do you feel they helped with education?	63%	12%	20%	2%	3%
Should there be more use of the algorithms and simulations?	50%	15%	26%	5%	4%
Was it clear on a test where the answers were correct/incorrect?	81%	19%	0%	0%	0%

Did you find the tests were the correct length/size?	40%	32%	23%	5%	0%
On the whole, how fun was the system to use?	39%	36%	25%	0%	0%
Would you use this system again?	42%	33%	23%	2%	0%
<b>Teachers and admin only</b>	-----	-----	-----	-----	-----
When making a test, was it easy to add a new one?	22%	46%	29%	3%	0%
Was having all the teachers 'tools' on the same page useful/important?	25%	23%	19%	24%	9%
Was the system easy to use once you had been given a short demonstration?	52%	34%	10%	4%	0%
When adding a lesson did you feel the text area given was suitable?	63%	71%	6%	0%	0%
Did you find it easy to check a students test results?	8%	20%	25%	26%	21%
Would you use this system to help with education?	48%	49%	2%	1%	0%
<b>Admin only</b>	-----	-----	-----	-----	-----
Did you feel you had suitable control over the system?	39%	51%	9%	2%	0%
Was it easy to enter in a message to the main page?	86%	10%	4%	0%	0%
Was the method of populating the fields when editing a users data useful or helpful?	72%	25%	3%	0%	0%
Was it easy to change any data/information about the user (e.g. change password)?	72%	27%	1%	0%	0%
Did you find the ability to use the 'teachers' side of the project necessary and or useful?	67%	32%	1%	0%	0%

The questionnaire shows where the system struggled to, or just did not fulfil its objectives. Different questions revealed different weak points in the system. For example, I realize that navigation seemed to be a key issue with users, especially for when they used the system for the first time, this is something I can start planning to fix straight away. However, for some of the other negative parts of the system (like how the teachers seemed to dislike/find a fault in the way the students scores were presented) further inspection will be needed. When I interview the users, I'll ensure to ask about the systems negative side, where the questionnaire has pointed out. These areas include:

- navigation of the system
- Ease of use for first time users
- Bot usability
- More explanation on bot and simulation
- Teacher page layout
- Students results layout

There were also positive areas of my system. Some areas of it worked extremely well (according to the questionnaire) and it leaves room for further expansion on the systems strong areas. For example, the algorithms and simulations seemed to be very useful and enjoyable for the students, with over half the

students enjoying them enough to want more. After I have preformed interviews with the users Ill be able to see where the improvement and corrections can be made.

## Interviews

After I had analyzed the questionnaire I was able to see where the project didn't meet the user's requirements, and where the system has lacked behind. However, the questionnaire didn't allow me to find out why it failed, only where. Therefore, in order to gain a better understanding of the system, ill need to be able to talk to the users, and ask them follow up questions to find out in more detail, about the areas my system needs improving on. When I preform the interviews, ill make sure to question at least one of each type of user from my system.

Daryl (Admin) -

So with the prototype you had admin privileges, how was having access of the whole system? "Well it was nice to be in control of everything, as in I can go and do something for a student or teacher if they are stuck and show them, without having to logout from my account and into theirs." So you were able to help other users more easily by allowing you the extra access? "oh definitely, it would have been a pain not to have the access" I made sure all the admins changed at lest one part of another users record, what did you end up changing? "I changed the access level for a teacher" How was it? Was loading the teachers details easy for you? "There wasn't too much too it really, I was told the username by the teacher, then I just searched it up. I guess it was nice being able to just click on the teachers record and have it load into the boxes for me, saved me form entering it my self, and the rest was as I expected, edit and hit save. Was the layout of the page useful? Did it hinder you? "I mean it wasn't some fabulous page that could solve any problem with one magic button, but its what I expected, it had what it needed to get any job I was given done." So it had all you needed? "I guess so yes, it had the key search and edit function of the users, and if I needed something else, I just used the teacher side of the program." Well that's it question wise, are there any comments you wish to add about it? "not really, it did all I needed to do with it, from the admin side all you really needed was control over the system in case something goes wrong or needs changing"

Joe (student) –

So Joe, what were your first impressions with the system? How did you fist find it when you jumped in? "Erm.. it was okay, I had to make an account first before I could use it tho" Did it bother you having to make one? "well no not really I mean, it didn't take ages only a min or two" And im assuming you went bothered with giving the information it requested, like your name and email? "well not really... like I already give my email out to Facebook and stuff when I signed up to them its wasn't that different really" So after you signed up what did you do? "I was on the page with the changing photos, and I.. I think I went onto the bot thingy first." And how was it? "it was fun, I got to see the bot move around and have it drawn out for me after on the page, I could see where it kept hitting the wall" So you must have taken out the memory card with the bot and used it with the main program, was it easy to do? Was it a bit fiddly? "It was okay, there was a hole in the case thing, so I could just take out the card and then place it into the computer, it was a tiny bit of effort having to type in the whole path but it wasn't hard, then I used the 'bot page' to get it to draw the lines. But I did have a little trouble when turning the bot

on and off, taking out the battery pack each time was a little annoying, as it was hard to tell if the bot was in the correct mode" Did you get to try the other mode, where you programmed it? "Oh yeh, I really liked that one, It was a bit hard to see which way it was facing when using it, we had to keep track of which way it was facing. I ended up setting a mini obstacle course with my class to move it round and try and get it to the end" So you made your own track for it? "yeh, we got bags and chairs, and we had to drive it around them all, it was really fun" So would you want to see more of the bot? I take it you liked it a lot? "yeh it was really fun, I enjoyed being able to move it, and track it" So the bot was good then, but how did you enjoy the rest of the program? Did you find the lessons were useful? "Yeh, I read a lesson about logic gates, it was interesting. At the end of the lesson there was a video link, and I got to watch it in the screen alongside the lesson." Did the video help with the topic? "it was nice to have as a bonus, but it did repeat some of the points made in the lesson, then again it did have new points the teacher didn't add to the lesson" So the videos were good to have as a sort of extra then? "yeh I would say that, it was kinda disappointing when there wasn't one to be honest." What about the simulation page? "the ones with the sorts and search thingy? It was enough to use..." ....But... something needs improving? "kind of yeh, like the ones were all good for the lessons I did with sorts and searches, but they didn't help for logic gates and other subjects my teachers added. The ones on it were fun, but they dint have all the ones I needed" Oh okay, so if you could, you'd have one added for each lesson, where it would help with the topic? "yeh it was a lot better having it to help visualize how it all worked" Okay thanks, and what about the tests? How did you find them? "they were okay, I mean I don't like tests normally (hahaha), but it was what I expected. I did like how it showed where I got it right and wrong with colors, that was kind of useful." Okay, that good to hear, thanks for all your help. That's it for questions, unless you have anything you want to add? "just one thing, I was gonna say it before but I didn't want to seem rude. Something a few of u in my class didn't like having to go into one thing to open up a page behind it, like the simulation page thing, I couldn't find it on the main page and I got confused the first time." So you think some sort of quick menu could help benefit other users? "Yeh, just something to get around the pages quicker"

Chloe (teacher) –

Tanks for taking the time to talk to me, so you were a teacher on the system, how did find it? "it was alright, useful actually, I particularly liked how I could add lessons, and then update them later on, my students really seemed to like it when I added a video for them to watch along with the lesson" Did you think that having the video and the lesson on the same page helped? "I would say they did, as the students could only go on videos given, they seemed to read the text then watch the video, pausing the video if they needed to re read the lessons again. So I think it was nice having it all in the same space" You had three main tools as a teacher, making lessons, tests, and checking scores. Were these tools enough to help you teach? "They were the minimum of needed tools, normally I am used to a bit more, as in on my school's register system I can see who's in and who's off ill or sick, that being said I guess as the students are already in class its not really needed to have the same tools, just in different places. Still having a way of comparing how a student does compare to others should be good." So you would have more accessibility with the students scores and results? "yes I would like to see that, a better way to compare two students than just reading them off one by one." Did your student seem to enjoy the system, was there anything they seemed to like in particular? "the little buggy was there favorite mode, they really liked being able to drive it around and liked tying to get it to follow paths" As a teacher, what did you think of it, was the bot a good way to get them to learn? "it definitely got their attention and

*time, I actually used it as a reward in my GCSE class to get them to complete all the tests” And I think you had another class you used with it? What level were they at? “They were my A2 class” Did they enjoy it the same amount? “they seemed to, they wanted to drive the bot just as much as the GCSE class. Have you got anything you want to add? Any last notes? There is one small thing, once I had made a lesson or test, I couldn’t get rid of it, I could only edit it and change what it had in it, the name stayed the same. I would’ve liked a way to removed old ones or ones with mistakes in, rather than just having to change its insides. So a remove/ delete function on the teacher page for lessons and tests? Yes, I think that would be very beneficial to for teachers.*

### Interview analysis

After having the chance to talk to the users I was able to gain a better understanding of the problems the system had, but I was also able to see where the system worked well and even gave me ideas of how to improve the system that I hadn’t thought of before.

The admin side of the system seemed to contain all that was needed for the user, with the features built in being expected from a ‘normal’ system, as this keeps the system normal and familiar to the users I believe this means the admin section was made to the correct standards. The admins were pleased with the tools and features they had, this reassures me that the system works for the admin side, and doesn’t need any corrections.

However, the other side of the system is in need of some corrections, from the questionnaire I already had some pointers of where to start, but with the added benefit of the interview I was able to understand properly where the system was inadequate. Some of the problems brought to my attention where new, others were the problems shown in the questionnaire but with more detail.

- navigation of the system – users found the navigation difficult as they had to go though the different pages to access the one they want. – this could be fixed with a quick menu.
- Ease of use for first time users – First time users found the system confusing, this however was negated after being showed how to use the system. – To correct this I could add a help guide.
- Bot usability – Some of the users found the bot hard to power up and select the correct mod. To correct powering up the bot, I could add a physical switch, instead oh having the user plug and unplug the power supply. As for the bot mode selection a physical output could help to already the user of it function/ operation
- Teacher page layout – after an interview with a teacher, and looking at the questionaries’ scorings I see that the spread was mixed of the layout being bad and good, with the interview I now understand its more down to preference, and aesthetics, as this doesn’t effect the systems performance (and because it was a even enough split) I wont be changing this part.
- Students results layout – after the interviews, I now understand the teachers wanted a different way to review the students scoring, not as a text base list, but a more visual representation, such as a graphical system.
- Simulation page - The users enjoyed the simulations and algorithms, but they wanted more, and found that the lessons would quickly make the algorithms become obsolete – to fix this ill allow the users too add their own programs to be saved and added, this way new ones can be added or removed to keep them in date.

- Bot mode – As the bot was a popular part of the system, users wanted more of it, I will make another mode for the bot to use so the system has more to offer the user.
- File selection – The users had some improvement areas, one of the ‘nagging’ factors as having to manually type in files/file locations of outside data, to amend this I will add a different way of selecting a files location.
- Bot program page – The users had some trouble seeing which way up the virtual bot faced –to fix ill add a way of the user seeing direction.
- Teacher page – removal and deletion of obsolete records, the interview showed that the teachers would want the ability to remove records, not just edit them, therefore I will allow teachers to remove records from the learn data and test data.

### Areas for improvement on the system

#### Navigation of the system –

1. Have a menu button on the main page once the user signs in
2. When pressed opens up small side page with buttons linking all the student’s pages available
3. The buttons will open the correct page
4. Admins and teachers will have access too the admin and teacher page from this, but students will not be able to see or use the buttons.

#### Bot Page File Loading Improvement –

1. To select the new location, the user clicks the ‘extract data’ button
2. A windows file selection form is then opened, with a filter for the correct type of file
3. Fine the correct file that the user wants to load/save
4. Press continue to confirm file selection.

#### Simulation Page improvement –

1. Upon loading the page, all the names of saved programs are loaded into the combo box
2. The admin (and only the admin, due to safety issues as the selected program to be added needs to be safe and ensures its not malicious)
3. The admin can add or remove a program from the list (saved by the program before)
4. This can be done from the ‘Add’ and ‘Remove’ buttons in the admin section.
5. When adding a file
6. The admin selects the location of the file they wish to add via selecting the file through a normal file selection form.
7. They choose the name of the program that they wish students to see and write it into the combo box
8. A short description about the program being added is given onto the textbox

9. The Admin then presses the add button
10. When removing a file, the admin selects the name of the program they wish to remove from the combo box
11. They then press the remove button.

To show the admins an example/give them an idea of the kind of program the system will run, I will include a small encryption program to be sent of with my current system, as this will be fun and entertaining to the students, but helpful to the admins and teachers as it will show how a small homemade program can be added to the system. For this I will use a programming language that is easy to use and understand.

Tracking students process –

1. On the teacher page the teacher will be able to press a button to open up a 'graph page'
2. Upon the graph page loading all the test that have been taken will be added to a selection box
3. The user will choose the correct test that they wish to see the results for.
4. The tests result for each student will be loaded into a graph at the side of the page
5. The user will also be able to select an option to compare two tests with each other at the same time.
6. Both test would be loaded into the graphs, with different colours.

Bot Improvement –

1. The bot will be adapted, and given new physical hardware.
2. A switch will be added, so the user can turn on and off the bot more easily
3. and an audio output will be added, as to give the user indication of the mode the bot it currently in.

Bot program mode –

1. The user will now be able to 'see' which way the virtual bot was facing
2. When the user presses Left or Right turn, a compass on the top of the page will change colour indicating the direction the bot is facing.
3. This will change when the L or R button is pressed changing the direction of the bot, and showing the user the way its facing.

New bot mode –

1. A new mode for the bot that allows the bot to drive around the room
2. The bot will hit the wall and turn right
3. But turn left if the wall has not been hit within a given time.

#### Teacher record removal-

1. The teacher will have two new buttons one on the test creation area, the other in the lesson creation area
2. When clicked they will remove the record currently selected by the teacher.

#### Third party feedback

In order to develop my system as best as I can, I sent off my demo to a third party. I had already got feedback from the interviews and questioners, but I wanted to ensure that the system has everything it needs. The feedback from the third party will allow me to ensure my project meets the standards it should be at, if I haven't already met them, the feedback I receive will help to me fulfil them.

#### re Feedback



Carberry, Chris

Fri 13/01, 12:43

Murray, Alexander Thomas (T0043344) ✕



Reply all | ▾

Flag for follow up. Start by 13 January 2017. Due by 13 January 2017.

Great System.

Can we have some additional refinements to the Teacher Side of the system.

Could we have some more formal tracking of student results for tests.

Ideally this would need to be provided by charts or graphs which could allow us to see if students are improving when they take additional tests.

If they are graded out of 100% then it would be easy to see the track results.

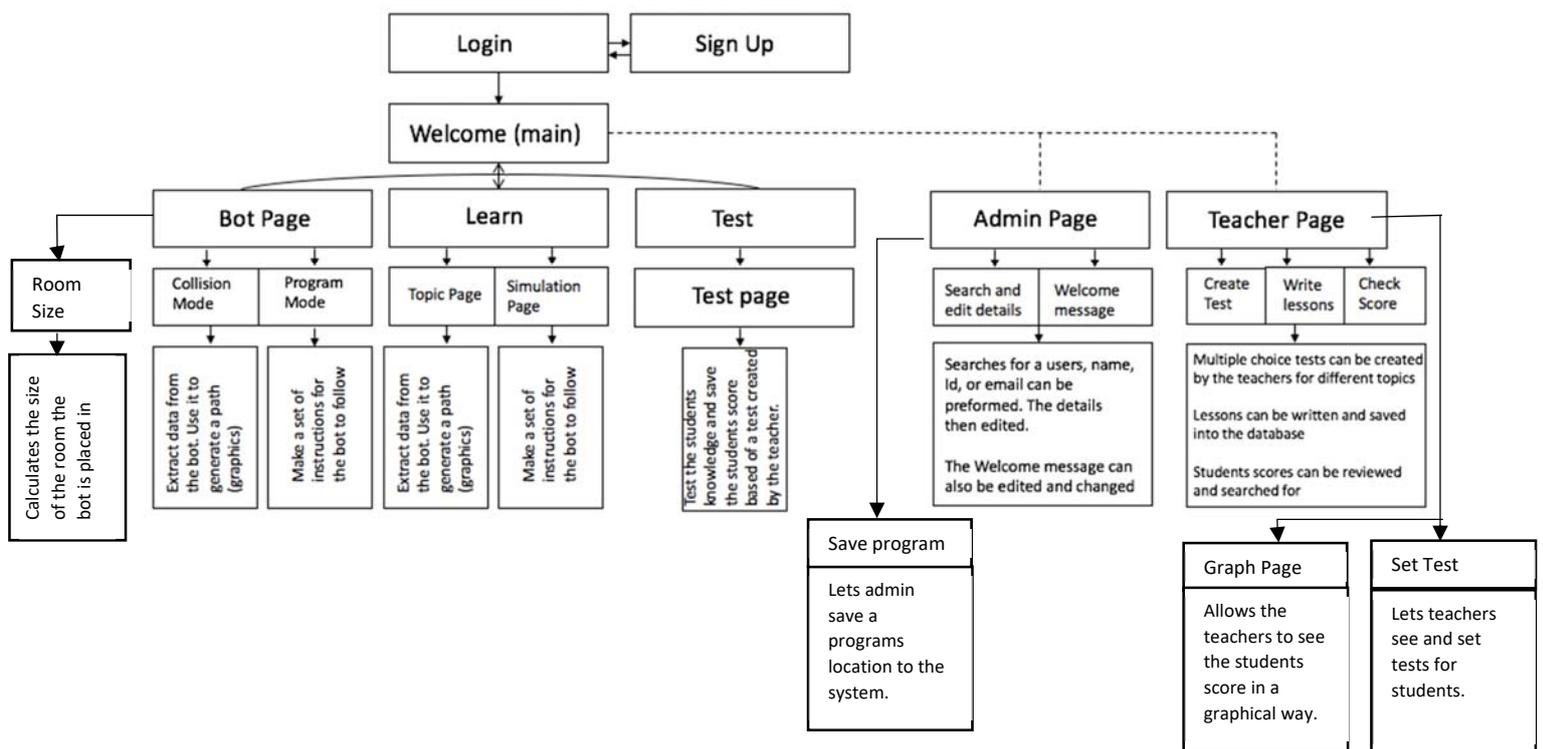
Perhaps teachers could also have the ability to book in students to take tests on a specific date – or instead request that students take specific tests by specific dates.

The feedback received from the third party allows me to confirm some of the downfalls in my system, for example the teacher system needs improvement when it comes to viewing the students scores. This is something that's been picked up multiple times from the demo, so I'm going to ensure I add a form to my system so that the teachers can view the scores properly. The feedback also talks about having teachers book students for the tests to be taken on particular dates, instead of just having them take

them randomly. This would mean adding a completely new element to my system. I can see why adding this to my system would be beneficial, as it would allow the teachers to give more guidance with the tests, for who takes what tests when.

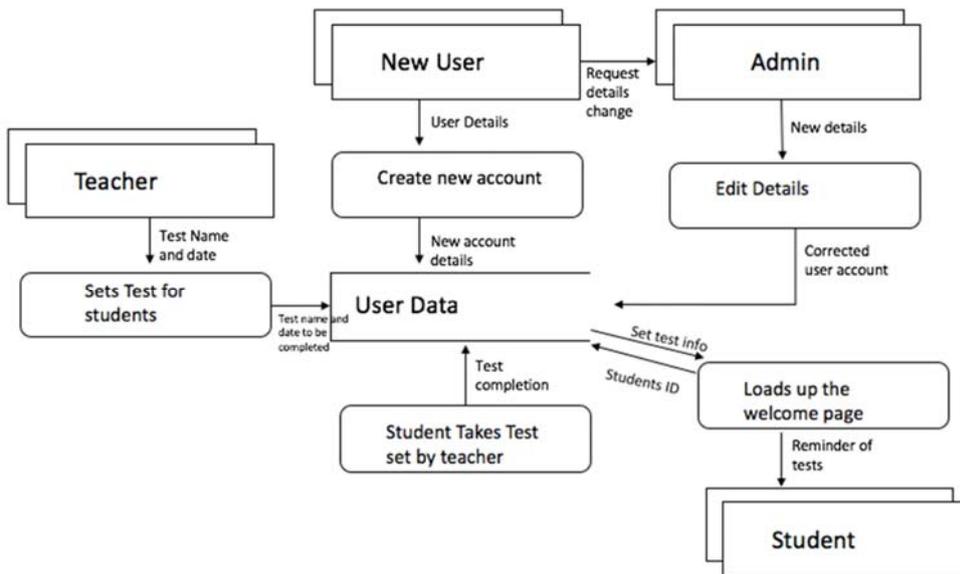
From my own feedback I have already realized that I will need some form of a chart/graphical display to show the students progress with the test. But as this was commented on by both the third party and my own investigation I understand that is vital I develop this new section to the correct standards.

New System Layout:



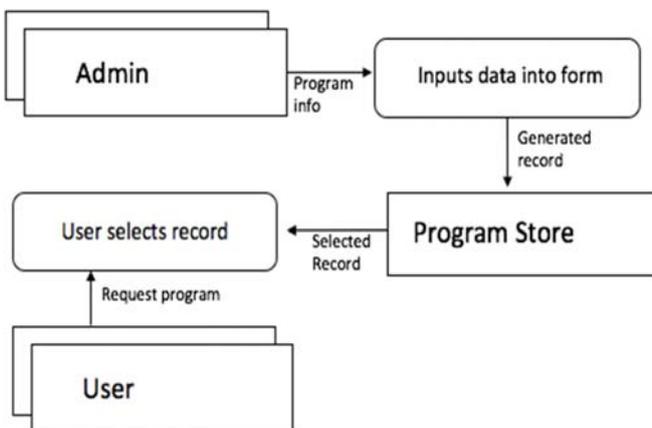
New DFD:

**User Data:**



The teacher can now set tests, this saves a test with a completion date and a completed flag into the student's own record, hence accessing the User data. When a student completes the test the flag is changed. The student also loads up the file (behind the scenes) when opening up the welcome page. The program checks for if they have a test that's still in need of completion.

**Program Store:**



The admin enters the programs details (Name, location, and description) into the from, they then save it.

The student can then see a list of available programs. They can access the file and load up a selected one.

### Altered forms and array

#### **UserData.txt (Altered file) –**

The file is still used for the same purpose as before, the only new features are the addition of the three end fields. The new fields allow the teachers to keep track of the student's progress with that particular test. As the file has changed the corresponding structured array has also changed in the same way.

<b>Field</b>	<b>Description</b>	<b>Type</b>	<b>Example Data</b>
Record Number	Number that represents the records place.	Integer	5
Username	A unique name that is generated by the users Second name, first name and record number. Used to login and identify user	String	MurrayA1
Email	Users email address, for contact purposes.	String	example@eg.com
First Name	Users First Name	String	Alex
Second Name	Users Second Name	String	Murray
Date of Birth	Users Date of Birth	Date	1 <sup>st</sup> June 1998
Password	A secure word/number combination that only the user should know (acts as a key)	String	Pa55W0rd3
Access Level	A number that represents the authority of a user. Restricting or allowing the usage of the programs features.	Integer	2
Test Name	Saves the test name that the teacher has asked the user to complete.	String	Bubble Sort
Date to be done by	The date which the teacher set the test to be completed by.	Date	12/10/98
Finished status	A Boolean flag to indicate if the user has completed the test	Boolean	True

### ProgramLocation.txt (New file)

This File will allow the users to open and use small programs/simulations developed by teachers and admins, the file holds records, each record hold a record number, name, short description and the location of the program. The file is read line by line and converted to a structured array when read, and saved from the array line by line when saved.

Field	Description	Type	Example Data
Record Number	Number that represents the records place.	Integer	5
Program name	Name given to the program by an admin	String	Encrypter
Description	A short description of what the program does	String	Makes data unreadable
Location	The file path /location of he program	String	//D:Folder1/new/program

### BotDataFile.txt (Alternate file use)

The bot only has one File on the SD card, however the purpose changes with the different modes. As the new mode had to use similar code (in order to hold the program as it has limited memory) the data sorted will be the same but the use of the data will be different. The data is read into with the newest data being added to the end of the file

Field	Description	Type	Example Data
Right turn	The number of times the bot hit a wall and turned right	Integer	1
Left Turn	Holds the number of times the bot had to re find the wall	Integer	1
Time	Holds the Time (mill seconds)at which the values are saved	Long Integer	5829

### Changed Structures:

Field	Description	Type	Example Data	Validation Used
Record Number	Number that represents the records place.	Integer	5	-None- (not needed as set by program)

Username	A unique name that is generated by the users Second name, first name and record number. Used to login and identify user	String	MurrayA1	None- not required as its generated by the program (from two other validated parts)
Email	Users email address, for contact purposes.	String	example@eg.com	Regular Expression check (format and presence check)
First Name	Users First Name	String	Alex	Length check
Second Name	Users Second Name	String	Murray	Length check
Date of Birth	Users Date of Birth	Date	1 <sup>st</sup> June 1998	None- not required as its
Password	A secure word/number combination than only the user should know (acts as a key)	String	Pa55W0rd3	Regular Expression ([0-9][A-Z][a-z]) Presence check for a number, lowercase and uppercase letter.
Access Level	A number that represents the authority of a user. Restricting or allowing the usage of the programs features.	Integer	2	None- (not needed as pre set of 3 is given to the user)
Test Name	Saves the test name that the teacher has asked the user to complete.	String	Bubble Sort	Name is selected from a set list
Date to be done by	The date which the teacher set the test to be completed by.	Date	12/10/98	None- not required as taken from the program
Finished status	A Boolean flag to indicate if the user has completed the test	Boolean	True	Is set to false upon file changing (not required to be validated)

New structure

Field	Description	Type	Example Data	Validation
Record Number	Number that represents the records place.	Integer	5	None (as added in by code)
Program name	Name given to the program by an admin	String	Encrypter	Presence check and check It hasn't already been used
Description	A short description of what the program does	String	Makes data unreadable	Presence check
Location	The file path /location of he program	String	//D:Folder1/new/program	Presence chekc

Re designed Admin form

The screenshot shows an 'Admin Page' window with the following components:

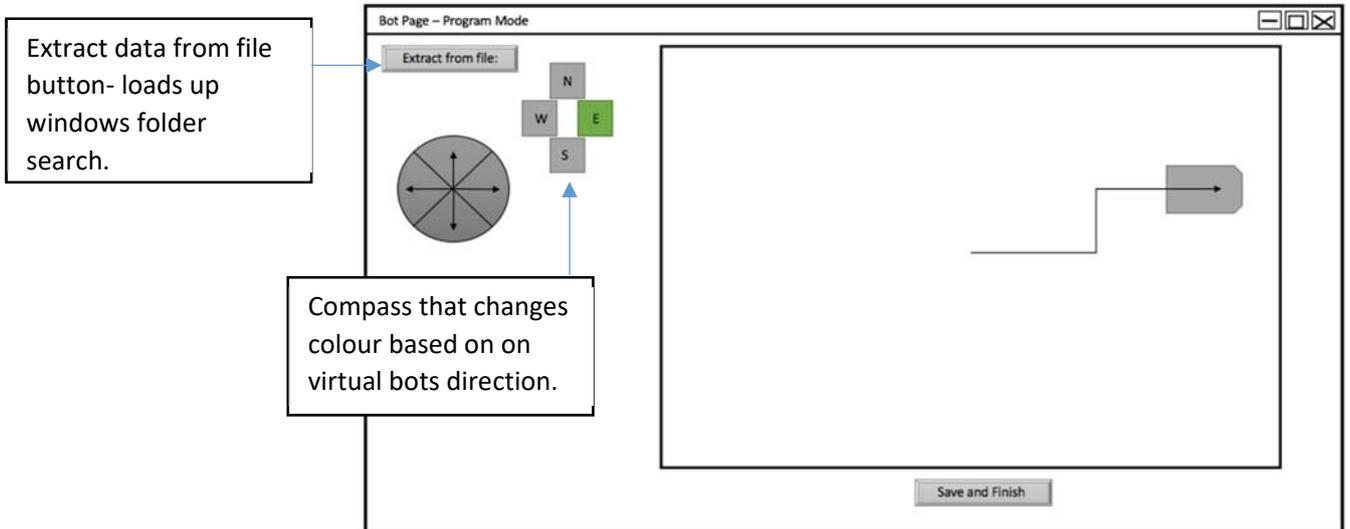
- Search Section:** Radio buttons for 'Email', 'Surname', and 'User Name', followed by a search input field and a 'Search' button.
- File Management Section:** A 'Locate File' button, a dropdown menu, and 'Remove' and 'Add' buttons. A text box below is annotated as 'Text box – for a description'.
- Message Section:** A 'Save Message' button and a 'Message box: for the admin to enter a message.' text area.
- User Profile Section:** Input fields for 'Record Number', 'User Name', 'First Name', 'Second Name', 'Email', and 'Password', with a 'Save' button.
- Display Section:** A 'Display Box: Will show all the text from the admins searches' and a 'Display All' button.

Callouts provide additional context:

- 'Add and Remove buttons for adding and removing the program'
- 'Combo box that allows the admin to select a name for the program, or make a new one.'

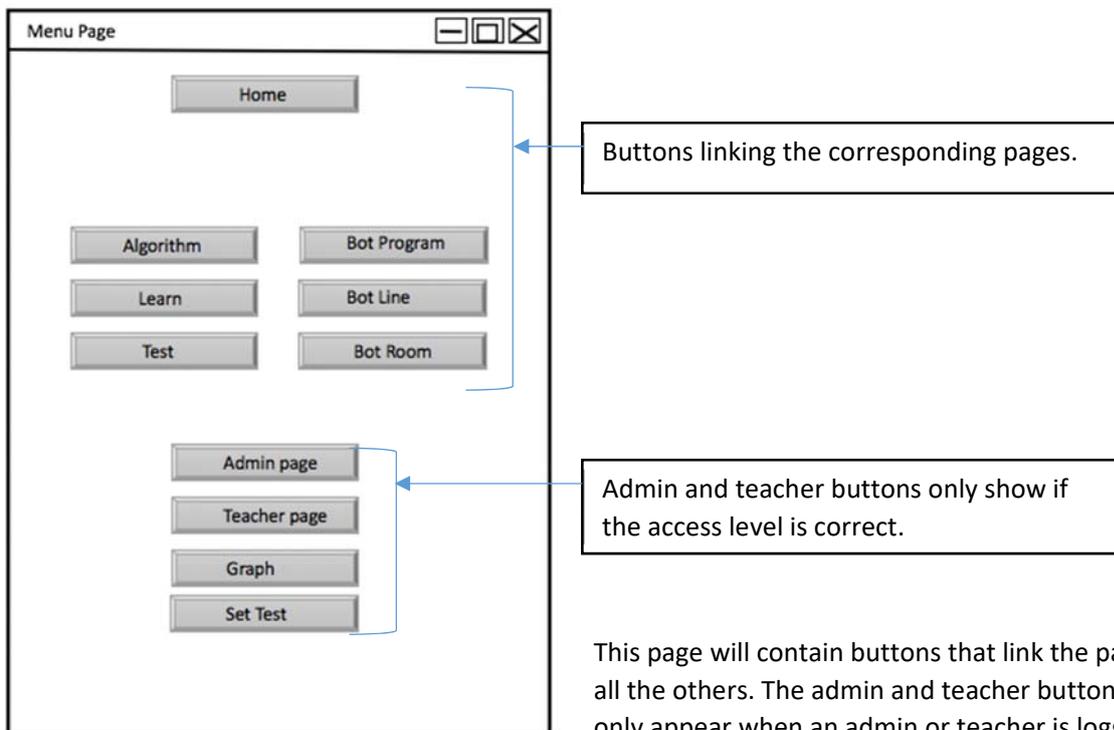
The Admins will be able to add programs by entering in a name into the combo box, along with a short description of the program into the textbox. They can then find the program in a familiar 'file search' selection mode, by pressing 'locate file' button. Once all the data has been entered the user can save the program by pressing 'add'. To remove a program, the admin selects the name of the program they wish to remove from the combo box and then presses the 'remove' button.

New Bot Program mode:



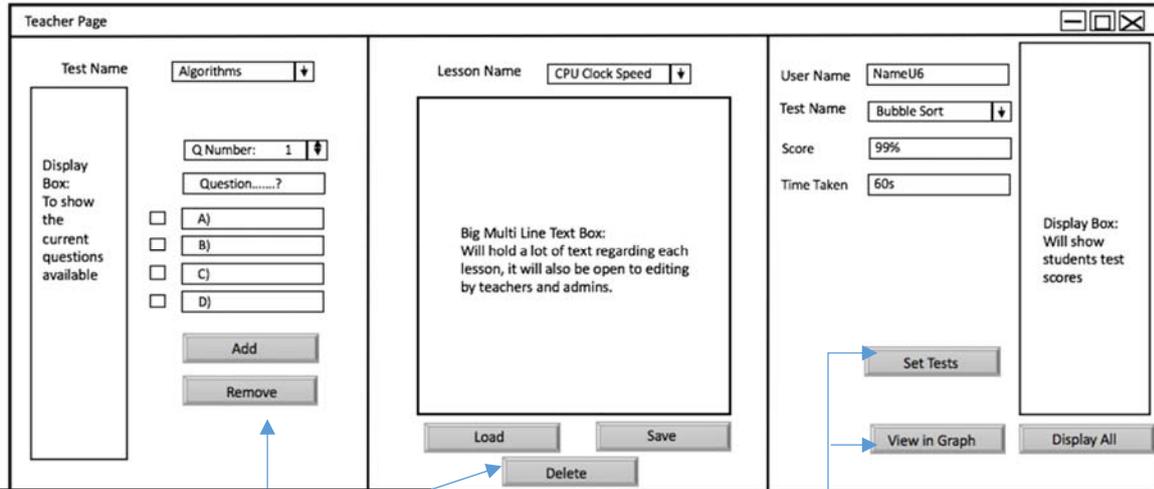
The page now has a new way of selecting the file the 'program' will be saved in, this can be used by pressing the 'extract from file button' to extract the location of the file using the windows file selection system. The compass changes color based on the direction that the on screen bot is facing.

New menu:



This page will contain buttons that link the page to all the others. The admin and teacher buttons will only appear when an admin or teacher is logged into the system.

Edited Teacher page:



New buttons allow removal of the record selected

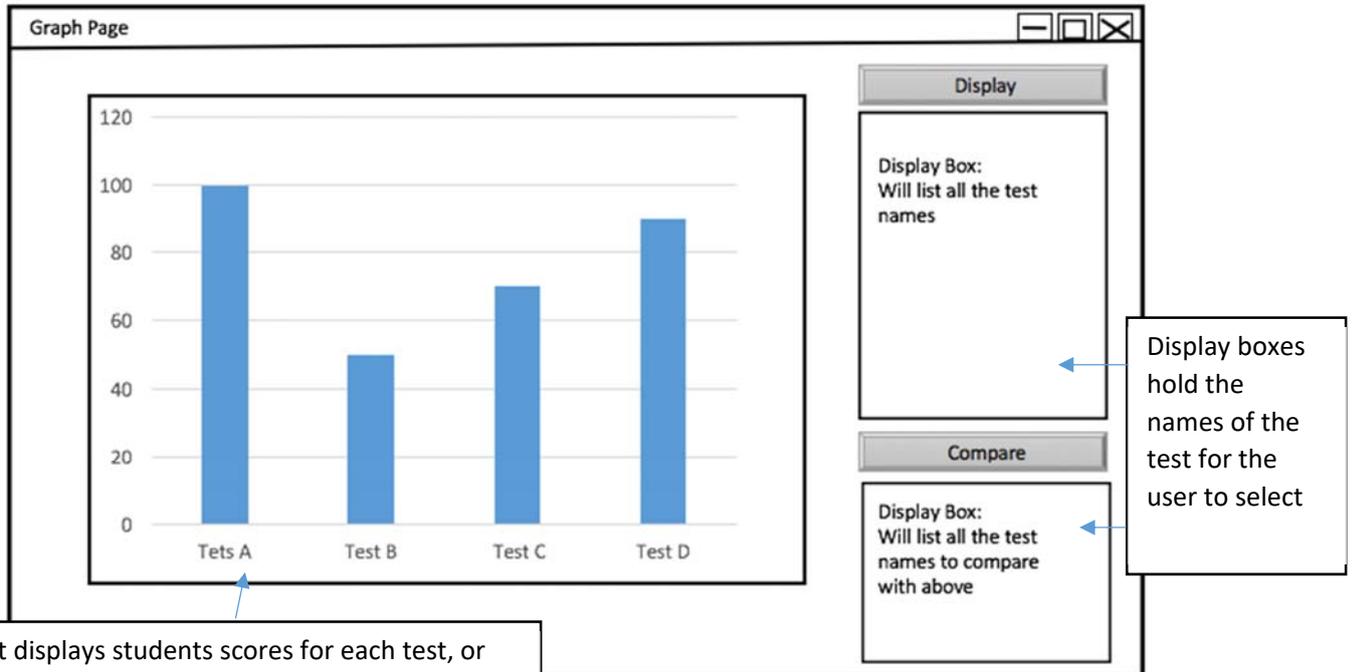
Theses new buttons allow the teacher to set tests or view the data in a graph, and links this page to another.

Feedback from both my enquiries and the third party has allowed me to update the teacher from, teachers can now delete record in the learn and tests section, this allows them to remove any unwanted ones. The teachers can also the the 'set test' and 'view in graph' buttons so open corresponding new forms.

**New output:**

The page has had some new changes, one of which is the delete/remove function for the lessons and the test creation side. Use of both these buttons provide a new output to the user of a message with a yes or no response to see if the user is sure they with to permanently remove the record from the file. This output is important as it gives the user a second chance to decide if they want to remove it or not as they may have selected the wrong file to delete.

## New Graph Page:



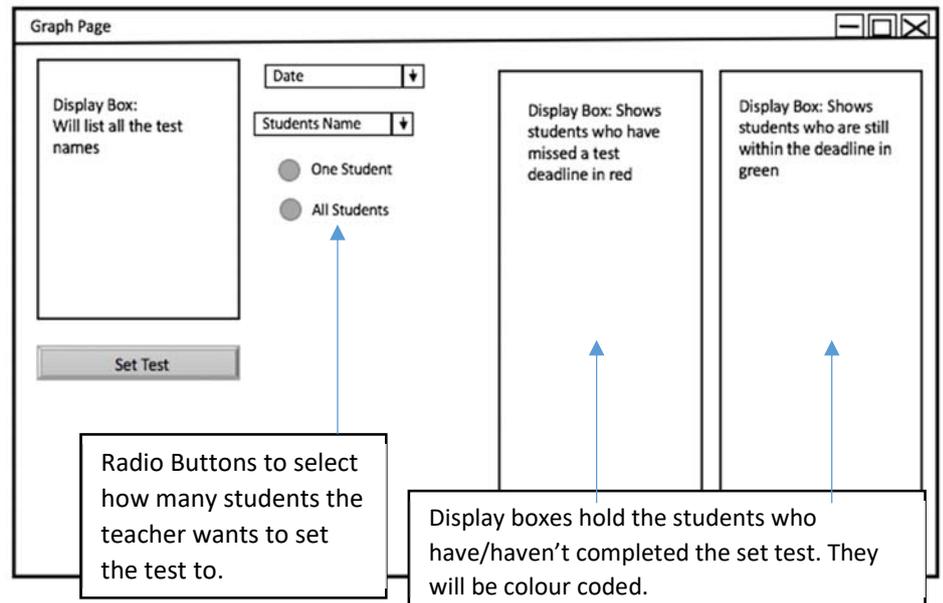
Upon opening the page loads the tests names into the display boxes, the user then selects a test name and has all the scores displayed on screen for that test, this gives a greater inside to how well a student is doing for the teacher. The teacher also compares two tests by selecting two test names and pressing compare.

### Outputs:

- Graph – the main output for this page is the graph, this bar chart makes it a lot easier for teacher to see how the students are doing on the tests, by comparing the data in a visual from. This allows the teachers to compare multiple students at once and can even compare two different tests at the same time if they wish.
- Test Names – the names of the tests are given to the teachers. This output is taken from the programs student scores file any tests in the file have to have been completed, so they are the possible tests the teachers can compare.

### New – Set test page:

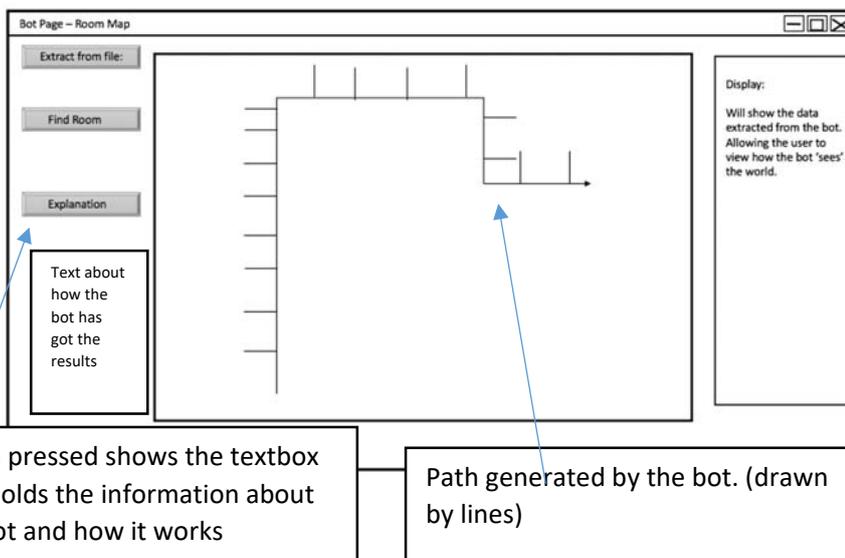
Open the page opening the test names and student names are loaded into the correct display boxes. The teacher can add a test to be completed, by selecting the test, and a date, then either selecting the one student, or all the students and press 'set test' This will open up the correct student's record and change the correct data in it.



### Outputs:

- Test Names – the names of all the tests that have been taken are given out to the user via a list box, the teacher gets to see all the test name to make it easier for them when selecting one.
- colour coded student's names – the students names who have and have-not completed the tests in the set time are shown to the teacher. Green for completed in time, Red f not. This colour code make it even easier for the teacher to see who has completed and not completed the test.

### Bot Room Mapping Mode



This page will act in a similar way to the collision mode page, but using a different proses it will map out the room that bot was sent to map out. Users extract the data as before (via button click and widows file selection form.) They can then map out the room by pressing the "Find room" button.

The user can also press the explanation button to show a textbox with information on the bot

## **New Pseudo Code: (developed due to my own investigations)**

Menu:

- When a button is pressed on the menu
- Load up corresponding page

Bot page – compass direction:

- When L or R button is pressed, change the NESW counter to -1 or +1
- Check if NESW counter is lower than 1 or greater than four, if so loop back over
- Check the value of NESW, if 1 then set the north button to green and all the rest grey
- If 2 set east to green and the rest grey, etc.

Teacher page – Delete Lesson:

- Read the file ()
- Set 'Addition' to 0
- Set found to false
- Set a counter to 0
- Loop back to here for length of the file -1
  - If the selected name to be deleted is = to the name in the array at counter
    - Set found = true
    - Set addition to 1
  - End if
  - Save into the array at position ('counter' + 'Addition') the fields of the records
  - Set counter to counter + 1
- Loop back
- If 'found' is true
  - Enter all the data in the changed array into a display box
  - Save the array into the text file
- Else
  - Send a message to the user "Please select a record to be deleted"
- End if
- Change the contents of the combo box showing the lessons.

Chart page – upon loading:

- Read the test score data file
- if a test name has not been added to the display boxes
  - Add the names of the test to the display boxes

- end If

Chart page – User pressing display button:

- Make the buttons on the page invisible
- loops to here for number of records
  - Adds the students name and score to the graph, if the test name matches the selected, at the position 'counter'
  - Add one to the counter
- loops back

Chart page – User pressing compare button:

- Make the buttons on the page invisible
- loops to here for number of records
  - Adds the students name and score to the graph, if the test name matches the selected, at the position 'counter'
  - Adds the other test at the same position where applicable.
  - Add one to the counter
- loops back

Admin Page – Addition of a new Program:

- Reads the program file
- Check to see if the correct fields have been entered
- if not – sends a message asking user to fill in all fields
- if they are filled in code process
- checks if the name entered has already been added before
- if its been added before
  - ask the user to change the name and check its not already been uploaded
- else
  - add the new record to the program file
- end if

Admin Page –Removal of a program:

- reads the program file
- Collets name of program to be deleted from user
- Starts saving the array into the file
- If the array finds the name

- Skip the record and move onto the next
- End if

Algorithm/simulation page – loading a program:

- Checks the combo box for a changed name
- Find the record with the same name
- Use the records description- loading it into the textbox
- If the load button is pressed
  - Find the program using its path location
  - Run the program
- End if

Small Example Encryption Code:

- Get the user to input a message
- Get the user to input a key
- Convert the message and key into ASCII
- Add the keys ASCII code to the message
- Ensure the ASCII code is still within the alphabet bounds
- Convert back into letters
- Output the Encrypted message

Small Example Decryption Code:

- Get the user to input an already encrypted message
- Get the user to input the key used
- Convert the message and key into ASCII
- Subtract the keys ASCII code from the message
- Ensure the ASCII code is still within the alphabet bounds
- Convert back into letters
- Output the Decrypted message

Bot Buzzer output:

- Set Buzzer Output pin to HIGH
- Wait for a pre-set amount of time (1 second)
- Change the output back to LOW

Bot mode- Room Calculation:

- The bot starts to move forward
- If the bot hits something
  - Stop
  - reverse a little
  - turn right
  - Add to right counter
  - save data
  - start to move forward again
- Else if a second has past
  - Turn left
  - More forward until something is hit
  - Reverse a little
  - Add to left counter
  - Turn right
  - Save data
- End if

Amendment developed due to third party feedback:

Set Tests page – When user sets a test:

- Read the user file
- Check which radio button has been selected
- If the radio button for all the students is checked
  - Make the student selection combo box invisible
  - Send out as Yes or No feedback message box asking if the user is sure
  - If they are sure
    - Edit all the students record and set the new test
  - If not
    - Do nothing
- If just for one student is checked
  - In the user structured array, set the test name at the position of the selected student to the Test name selected in the display box
  - Set 'completedby' date to the date in the date and time picker
  - Set the Boolean flag for completion to false
  - Save the records into the text file.
  - Encrypt the file()
  - Send a message to the user saying it has been done
- End if

When the page loads –

- Read the user file
- If the students test is not done by the completion date
  - Add the name to the uncompleted text box in red
- Else
  - Add them to the completed text box in green
- End if
- check if the test have been added to the test selection display box
- if a test has not yet been added
  - add it the test to the box
- end if
- Add all the users of the system to the student selection combo box

## Software development

### Login form-

```
Imports System.IO
```

```
Imports System.Security.Cryptography
```

```
Imports System.Text
```

```
Public Class SignIn
```

```
    'All encryption and decryption
    Dim Key As String = "IlpHatKdw" 'Essencailly the key
    Sub EncryptFile() 'Encypst the UserData file making it impossible to read
        Dim DataFromFile As String = File.ReadAllText("UserData.txt")
        Dim EncKey As Byte() = encrypt(DataFromFile, Key)
        Dim Output As String = Convert.ToBase64String(EncKey)
        File.WriteAllText("UserData.txt", Output)
    End Sub
    Sub DecryptFile() 'Decyrps the file, making it readabe after
        Dim DataFromFile As String = File.ReadAllText("UserData.txt")
        Dim EncKey As String = decrypt(DataFromFile, Key)
        File.WriteAllText("UserData.txt", EncKey)
    End Sub
    Function method(ByVal key As String) As TripleDES 'Part of the
tripple DES encryption/decryption function
        Dim DES As TripleDES = New TripleDESCryptoServiceProvider
        Dim MD5Cry As MD5 = New MD5CryptoServiceProvider
        DES.Key = MD5Cry.ComputeHash(Encoding.Unicode.GetBytes(key))
        DES.IV = New Byte(((DES.BlockSize / 8)) - 1) {}
        Return DES
    End Function
    Function encrypt(ByVal value As String, ByVal key As String)
        Dim DES As TripleDES = method(key)
        Dim IC As ICryptoTransform = DES.CreateEncryptor
        Dim Input() As Byte = Encoding.Unicode.GetBytes(value)
        Return IC.TransformFinalBlock(Input, 0, Input.Length)
    End Function
    Function decrypt(ByVal value As String, ByVal key As String)
        Dim Base As Byte() = Convert.FromBase64String(value)
        Dim DES As TripleDES = method(key)
        Dim IC As ICryptoTransform = DES.CreateDecryptor
        Dim output() As Byte = IC.TransformFinalBlock(Base, 0, Base.Length)
        Return Encoding.Unicode.GetString(output)
    End Function
```

DES encryption/decryption function:

The file holding the user information is read, and the data saved to a string. The data is changed into chunks of bytes to begin encryption/decryption stage. This is then called, and the data from the file and the encryption key are passed along (by value.)

The inbuilt functions are then used to change the data and generate a strong encryption key.

The now encrypted data is then changed to or back from base 64 and saved back into the file.

```
'.....
.....
```

```
    'All User Data file info
    Dim SR As StreamReader
    Public RecordNumber As Integer
    Dim Counter As Integer = 0
    Public UserIdNo As Integer
    'The structure that the data is held in my text file
    Structure Users
        Dim IdNumberInt As Integer
        Dim UserNameStr As String
        Dim EmialStr As String
```

```

Dim FNameStr As String
Dim SNameStr As String
Dim DoBStr As String
Dim PasswordStr As String
Dim AcessLvInt As Integer
Dim TestComName As String
Dim CompBy As Date
Dim FinishedSta As Boolean
End Structure
Public SystemUsers() As Users 'Sets structre to array

Public Sub ReadUserFile() 'Opens and reads the user file
    DecryptFile() 'Makes the file readable
    SR = New StreamReader("UserData.txt") 'Sets the
file to open
    RecordNumber = SR.ReadLine 'Gets Num of
records from first line of text
    ReDim Preserve SystemUsers(RecordNumber) 'Makes the
array the same size as the text file
    Counter = 0
    Do Until SR.EndOfStream = True
        Counter = Counter + 1
        SystemUsers(Counter).IdNumberInt = SR.ReadLine() 'Goes
through each line of the text
        SystemUsers(Counter).UserNameStr = SR.ReadLine()
'file, each time it saves the line
        SystemUsers(Counter).EmialStr = SR.ReadLine() 'into
the preserved array
        SystemUsers(Counter).FNameStr = SR.ReadLine()
        SystemUsers(Counter).SNameStr = SR.ReadLine()
        SystemUsers(Counter).DoBStr = SR.ReadLine()
        SystemUsers(Counter).PasswordStr = SR.ReadLine()
        SystemUsers(Counter).AcessLvInt = SR.ReadLine()
        SystemUsers(Counter).TestComName = SR.ReadLine()
        SystemUsers(Counter).CompBy = SR.ReadLine()
        SystemUsers(Counter).FinishedSta = SR.ReadLine()
    Loop
    SR.Close()
    EncryptFile() 'Encrypts the file making it unreadbale
End Sub 'Opens and reads the data file

Private Sub BTNSignIn_Click(sender As Object, e As EventArgs) Handles BTNSignIn.Click
    Dim loggedIn As Boolean = False 'Flag to stop the loop
    ReadUserFile() 'Reads the user file to input contents into the
structured array
    Counter = 0
    Do Until loggedIn = True 'Does while the user has not been found
        Counter = Counter + 1
        If Counter > RecordNumber Then 'Gives an out for if there is no file found
in the array
            MsgBox("An Incorrect Password Or User Name Has Been Entered")
            Clear()
            Exit Do
        End If
        If TBUsername.Text = SystemUsers(Counter).UserNameStr And TBPASSWORD.Text =
SystemUsers(Counter).PasswordStr Then
            UserIdNo = Counter
        End If
    End Do
End Sub

```

The file has to be decrypted before the file can be read.

The first line of the text file is read, this holds the number of records in the file so the program knows how far to read into the file and make the arrays to hold it.

Line by line the program reads the data in the file assigning it to the correct field in the sorted array.

Once finished it closes the file and encrypts it again to make the data secure.

```

        loggedIn = True 'When name and password match lets value to true, stopps
checking for more than needed.
        MsgBox("Welcome " & TBUsername.Text)
        Welcome.LABUsername.Text = ("Welcome " & TBUsername.Text)
        Welcome.Show()
        Me.Hide()
        If SystemUsers(Counter).AccessLvInt = 3 Then      'hides or shows the right
buttons for admin or teacher
            Welcome.Student()                            'allows for securiy for
the correct users
        ElseIf SystemUsers(Counter).AccessLvInt = 2 Then
            Welcome.Teacher()
            MenuPage.BtnTeacher.Visible = True
        ElseIf SystemUsers(Counter).AccessLvInt = 1 Or 0 Then
            Welcome.Admin()
            MenuPage.BtnTeacher.Visible = True
            MenuPage.BtnAdmin.Visible = True
        End If
    Else
    End If

    Loop
End Sub      'When login is clicked - Searches for matching password and username

'.....
.....

'Display: clears text boxes
Sub Clear()
    TBPASSWORD.Text = ""
    TBUsername.Text = ""
    UserIdNo = 0
End Sub

'Button: Links pages
Private Sub BTNSignUp_Click(sender As Object, e As EventArgs) Handles BTNSignUp.Click
    Me.Hide()
    SignUp.Show()
End Sub
End Class

```

### Signup –

```

Imports System.IO
Imports System.Text.RegularExpressions

```

```

Public Class SignUp

```

```

    'All this saves the new file
    Dim SR As StreamReader
    Dim SW As StreamWriter
    Dim RecordNumber As Integer
    Dim Proceed As Boolean
    Sub UsersIntoFile()

```

```

SignIn.DecryptFile() 'Makes it readable
SR = New StreamReader("UserData.txt")
RecordNumber = SR.ReadLine
SR.Close()
SignIn.EncryptFile() 'need to re-encrypt it as readfile will try to decrypt it,
cant decrypt it as it already decrypted
Call SignIn.ReadUserFile()
SignIn.DecryptFile()
RecordNumber = RecordNumber + 1 'Adds one to the number of records
Dim TempNameStr As String = TBSname.Text + Mid(TBFname.Text, 1, 1) & RecordNumber
'generates a unique username
ReDim Preserve SignIn.SystemUsers(RecordNumber) 'makes the new
list bigger
SignIn.SystemUsers(RecordNumber).IdNumberInt = RecordNumber 'Bit by bit adds
all new data
SignIn.SystemUsers(RecordNumber).UserNameStr = TempNameStr
SignIn.SystemUsers(RecordNumber).EmailStr = TBEmail.Text
SignIn.SystemUsers(RecordNumber).FNameStr = TBFname.Text
SignIn.SystemUsers(RecordNumber).SNameStr = TBSname.Text
SignIn.SystemUsers(RecordNumber).DoBStr = DTPforDOB.Text
SignIn.SystemUsers(RecordNumber).PasswordStr = TBPASSWORD.Text
SignIn.SystemUsers(RecordNumber).AccessLvInt = 3
SignIn.SystemUsers(RecordNumber).TestComName = "Null"
SignIn.SystemUsers(RecordNumber).CompBy = Date.FromOADate(9 / 9 / 99)
SignIn.SystemUsers(RecordNumber).FinishedSta = True

SW = New StreamWriter("UserData.txt") 'Part where the
records are actually saved into the file
SW.WriteLine(RecordNumber)
For Counter = 1 To RecordNumber
    SW.WriteLine(SignIn.SystemUsers(Counter).IdNumberInt)
    SW.WriteLine(SignIn.SystemUsers(Counter).UserNameStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).EmailStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).FNameStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).SNameStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).DoBStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).PasswordStr)
    SW.WriteLine(SignIn.SystemUsers(Counter).AccessLvInt)
    SW.WriteLine(SignIn.SystemUsers(Counter).TestComName)
    SW.WriteLine(SignIn.SystemUsers(Counter).CompBy)
    SW.WriteLine(SignIn.SystemUsers(Counter).FinishedSta)
Next
SW.Close()
SignIn.EncryptFile()
End Sub 'Saves the new data into the file
Private Sub BTNSignUp_Click(sender As Object, e As EventArgs) Handles BTNSignUp.Click
    Validate()
    If Proceed = True Then
        UsersIntoFile()
        Clear()
        MsgBox("Welcome, your account has been saved. Your Username is " &
SignIn.SystemUsers(RecordNumber).UserNameStr)
        SignIn.Show()
        Me.Hide()
    End If
End Sub 'Button: If validation is correct, saves

```

A temporary space is used to hold a generated username, this is made from the users second name and the first letter of the first name, along with the record number to make it unique.

The file is read and has it previous data stored into the array.

The data is saved into a file, using the structured array on the sign in form to set the array to be written into the file. The data once written in is then encrypted.

```

'.....
'All validation for this sub here

Sub Validate()
    Proceed = True
    If (TBSname.Text = "") Or (TBFname.Text = "") Or (TBEmail.Text = "") Or
(TBPassword.Text = "") Or (TBPassword2.Text = "") Then
        MsgBox("Please fill All of the fields before proceeding")
        Proceed = False
    ElseIf (validateEmail(TBEmail.Text) = False) Then
        MsgBox("Email not Valid")
        Proceed = False
    ElseIf (ValidtePassWord(TBPassword.Text) = False) Then
        MsgBox("Passwords MUST have a lowercase and a capital letter as well as a
number. Please enter another")
        Proceed = False
    ElseIf (TBPassword.Text = TBPassword2.Text) = False Then
        MsgBox("The passwords you entered do not match, please try again")
        Proceed = False
    End If
End Sub 'Validation: Password check, and error messages

Public Function ValidtePassWord(PassWord) As Boolean
    Dim Validte As New Regex("(?=.*[0-9])(?=.*[A-Z])(?=.*[a-z])")
    If Validte.IsMatch(PassWord) Then
        Return True
    Else
        Return False
    End If
End Function 'checks password hads number, capital, and lowercase

Public Function validateEmail(UserEmail) As Boolean
    Dim EmailVali As New Regex("([\w-+]+(?:\.[\w-+]+)*)@(?:[\w-
]+\.)+[a-zA-Z]{2,7}")
    If EmailVali.IsMatch(UserEmail) Then 'Presense
check using regualr expressions
        Return True
    Else
        Return False
    End If
End Function 'Validation: Short way of validating the emial

'.....
'
'Button links and display things

Private Sub BTNLogin_Click(sender As Object, e As EventArgs) Handles BTNLogin.Click
    Me.Hide()
    SignIn.Show()
End Sub 'Button: Links pages
Sub Clear()
    TBEmail.Clear()
    TBFname.Clear()
    TBPassword.Clear()

```

Here regular expressions are used to ensure the correct data is allowed.

[0-9] checks for a numeric digit

[A-Z] ensures a capital letter given

[a-z] checks a lowercase is given

\*@ ensures the emails has the at symbol

In all, the regular expression ensures that the data entered is valid, this then send a boolean flag back to show the program the data is valid.

```

        TBPASSWORD2.Clear()
        TBSname.Clear()
        DTPforDOB.ResetText()
    End Sub 'Display: Takes care of info left over

    Private Sub TBSname_TextChanged(sender As Object, e As EventArgs) Handles
TBSname.TextChanged
        If TBSname.Text.Length > 15 Then
            MsgBox("Please enter a shortened second name")
            TBSname.Text = ""
        End If
    End Sub

    Private Sub TBfname_TextChanged(sender As Object, e As EventArgs) Handles
TBfname.TextChanged
        If TBfname.Text.Length > 15 Then
            MsgBox("Please enter a shortened first name")
            TBfname.Text = ""
        End If
    End Sub
End Class

```

## Welcome-

```

Imports System.IO
Public Class Welcome
    Dim Learn, Bot, Test As PictureBox
    'Display: Button access to the correct access leve
    Sub Admin()
        BTNAdmin.Visible = True
        BTNTeacher.Visible = True
    End Sub
    Sub Teacher()
        BTNAdmin.Visible = False 'will be called in form1(login) if needed
        BTNTeacher.Visible = True
    End Sub
    Sub Student()
        BTNAdmin.Visible = False
        BTNTeacher.Visible = False
    End Sub

    'Buttons
    Private Sub BTNLogout_Click(sender As Object, e As EventArgs) Handles BTNLogout.Click
        Me.Hide()
        SignIn.Clear()
        SignIn.Show()
    End Sub 'Loggs out the user, clears all data from them
    Private Sub BTNAdmin_Click(sender As Object, e As EventArgs) Handles BTNAdmin.Click
        AdminPage.Show()
    End Sub 'Admin Page
    Private Sub BTNTeacher_Click(sender As Object, e As EventArgs) Handles
BTNTeacher.Click
        TeacherPage.Show()
    End Sub 'Teacher page
    Private Sub PictureBox3_Click(sender As Object, e As EventArgs) Handles
TestPicture.Click

```

```

        TestPage.Show()
    End Sub 'Test Page
    Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles
BotPicture.Click
        BotSelect.Show()
    End Sub 'Bot Page
    Private Sub BtnMenu_Click(sender As Object, e As EventArgs) Handles BtnMenu.Click
        MenuPage.Show()
    End Sub 'Menu Page
    Private Sub PictureBox2_Click(sender As Object, e As EventArgs) Handles
LearnPicture.Click
        LearnPage.Show()
    End Sub 'Learn Page

    Private Sub Welcome_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim SR As StreamReader
        SR = New StreamReader("AdminMes.txt")
        TestCompletion()
        TbAdminMessage.Text = (SR.ReadToEnd) & extra
        SR.Close()
    End Sub

    Dim extra As String = ""
    Sub TestCompletion()
        If SignIn.SystemUsers(SignIn.UserIdNo).FinishedSta = False Then
            extra = ("You have still to complete " &
(SignIn.SystemUsers(SignIn.UserIdNo).TestComName) & ". You have untill " &
SignIn.SystemUsers(SignIn.UserIdNo).CompBy)
            MsgBox(extra)
        End If
    End Sub

    Dim PicNum As Integer = 0
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        PicNum = PicNum + 1
        If PicNum > 3 Then 'loops back counter if out of bounds
            PicNum = 1
        End If

        Learn = Me.Controls("LearnPicture")
        Learn.BackgroundImage = Image.FromFile("Learn" & PicNum & ".jpg") 'Changes the
photos in the box
        Bot = Me.Controls("BotPicture") 'everytime
the timer changes value
        Bot.BackgroundImage = Image.FromFile("Bot" & PicNum & ".jpg")
        Test = Me.Controls("TestPicture")
        Test.BackgroundImage = Image.FromFile("Test" & PicNum & ".jpg")
    End Sub
End Class

```

### Admin Page –

```

Imports System.IO
Public Class AdminPage
    Dim SW As StreamWriter

```

```

Dim SR As StreamReader

'Upon Loading
Private Sub AdminPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Call Form1.ReadUserFile() 'Calls file to read
    UserDisplay()           'Displays users data in a listbox
    LoadCbProgramNames()   'Load up names into file
End Sub 'UponLoading PageLoadUP

Sub LoadCbProgramNames()
    ReadProgramFile() 'Upon Loading the program reads the file
    For Counter = 1 To RecNo
        If CbProgramN.Items.Contains(ProgramLocation(Counter).Name) = False Then
'Adds name to the combo box if
            CbProgramN.Items.Add(ProgramLocation(Counter).Name)
'It isnot already in
        End If
    Next
End Sub

Dim Counter As Integer = 0
Public RecNo As Integer

Structure ProgramData 'data structure for the program data
    Dim Reccord As Integer
    Dim Name As String
    Dim Description As String
    Dim Location As String
End Structure
Public ProgramLocation() As ProgramData

Public Sub ReadProgramFile() 'Reads through the program file
    SR = New StreamReader("ProgramLocation.txt") 'saves the file into a structured
array
    RecNo = SR.ReadLine
    ReDim Preserve ProgramLocation(RecNo)
    Counter = 0
    Do Until SR.EndOfStream = True
        Counter = Counter + 1
        ProgramLocation(Counter).Reccord = SR.ReadLine()
        ProgramLocation(Counter).Name = SR.ReadLine()
        ProgramLocation(Counter).Description = SR.ReadLine()
        ProgramLocation(Counter).Location = SR.ReadLine()
    Loop
    SR.Close()
End Sub

Sub AddProgram()
    SR = New StreamReader("ProgramLocation.txt")
    RecNo = SR.ReadLine
    SR.Close()
    ReadProgramFile()
    RecNo = RecNo + 1 'Adds one to the number of records
    ReDim Preserve ProgramLocation(RecNo) 'makes the new list bigger
    ProgramLocation(RecNo).Reccord = RecNo

```

```

ProgramLocation(RecNo).Name = CbProgramN.Text
ProgramLocation(RecNo).Description = TbDes.Text
ProgramLocation(RecNo).Location = TbPath.Text
SW = New StreamWriter("ProgramLocation.txt")
SW.WriteLine(RecNo)
For Counter = 1 To RecNo
    SW.WriteLine(ProgramLocation(Counter).Reccord)
    SW.WriteLine(ProgramLocation(Counter).Name)
    SW.WriteLine(ProgramLocation(Counter).Description)
    SW.WriteLine(ProgramLocation(Counter).Location)
Next
SW.Close()
End Sub

'.....
'.....
'File Edit and save parts and functions

Sub EditSave()
    Call Form1.ReadUserFile()
    ReDim Preserve Form1.SystemUsers(Form1.RecordNumber)
    Form1.SystemUsers(TBId.Text).IdNumberInt = TBId.Text      'resaves the array at
this postion
    Form1.SystemUsers(TBId.Text).UserNameStr = TBUsername.Text 'to hold the new data
    Form1.SystemUsers(TBId.Text).EmialStr = TBEmial.Text
    Form1.SystemUsers(TBId.Text).FNameStr = TBFName.Text
    Form1.SystemUsers(TBId.Text).SNameStr = TBSName.Text
    Form1.SystemUsers(TBId.Text).DoBStr = DTPforDOB.Text
    Form1.SystemUsers(TBId.Text).PasswordStr = TBPASSWORD.Text
    Form1.SystemUsers(TBId.Text).AcessLvInt = NumUDAccess.Text
    SW = New StreamWriter("UserData.txt")
    SW.WriteLine(Form1.RecordNumber)      'Writes in the number of records (to be
read for next time)
    For Counter = 1 To Form1.RecordNumber
        SW.WriteLine(Form1.SystemUsers(Counter).IdNumberInt)      'saves into file
        SW.WriteLine(Form1.SystemUsers(Counter).UserNameStr)
        SW.WriteLine(Form1.SystemUsers(Counter).EmialStr)
        SW.WriteLine(Form1.SystemUsers(Counter).FNameStr)
        SW.WriteLine(Form1.SystemUsers(Counter).SNameStr)
        SW.WriteLine(Form1.SystemUsers(Counter).DoBStr)
        SW.WriteLine(Form1.SystemUsers(Counter).PasswordStr)
        SW.WriteLine(Form1.SystemUsers(Counter).AcessLvInt)
        SW.WriteLine(Form1.SystemUsers(Counter).TestComName)
        SW.WriteLine(Form1.SystemUsers(Counter).CompBy)
        SW.WriteLine(Form1.SystemUsers(Counter).FinishedSta)
    Next
    SW.Close()
    Form1.EncryptFile()
End Sub 'Saves into file, rewriting the data for one user only

'Search Functions
Sub SurnameSearch()
    Call Form1.ReadUserFile() 'Search: looks for surnames in Text
file
    DBMain.Items.Clear()
    For Counter = 1 To Form1.RecordNumber
        If Form1.SystemUsers(Counter).SNameStr = TBSearchBox.Text
Then

```

Calls upon the array used to save data into the file to allow editing to the file, and the reading of data from the file on the different forms.

```

        DBMain.Items.Add(Form1.SystemUsers(Counter).IdNumberInt)
        DBMain.Items.Add(Form1.SystemUsers(Counter).UserNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).EmialStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).FNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).SNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).DoBStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).PasswordStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).AcessLvInt)
        DBMain.Items.Add(Form1.SystemUsers(Counter).TestComName)
        DBMain.Items.Add(Form1.SystemUsers(Counter).CompBy)
        DBMain.Items.Add(Form1.SystemUsers(Counter).FinishedSta)
    End If
Next
End Sub
Sub EmialSearch()
    Call Form1.ReadUserFile() 'Calls the ReadUserFile function to store the data
    DBMain.Items.Clear() 'into the array - then clears the Display Box
    For Counter = 1 To Form1.RecordNumber
        If Form1.SystemUsers(Counter).EmialStr = TBSearchBox.Text Then 'If the email
of the user if the
            DBMain.Items.Add(Form1.SystemUsers(Counter).IdNumberInt) 'same as the
emial being searched for
            DBMain.Items.Add(Form1.SystemUsers(Counter).UserNameStr) 'Enter the
data into the Display box
            DBMain.Items.Add(Form1.SystemUsers(Counter).EmialStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).FNameStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).SNameStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).DoBStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).PasswordStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).AcessLvInt)
            DBMain.Items.Add(Form1.SystemUsers(Counter).TestComName)
            DBMain.Items.Add(Form1.SystemUsers(Counter).CompBy)
            DBMain.Items.Add(Form1.SystemUsers(Counter).FinishedSta)
        End If
    Next
End Sub 'Search: Checks for same email
Sub UsernameSearch()
    Call Form1.ReadUserFile()
    DBMain.Items.Clear()
    For Counter = 1 To Form1.RecordNumber
        If Form1.SystemUsers(Counter).UserNameStr = TBSearchBox.Text Then 'If the
Username of the user if the
            DBMain.Items.Add(Form1.SystemUsers(Counter).IdNumberInt) 'same as the
Username being searched for
            DBMain.Items.Add(Form1.SystemUsers(Counter).UserNameStr) 'enters the
users data into the display box
            DBMain.Items.Add(Form1.SystemUsers(Counter).EmialStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).FNameStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).SNameStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).DoBStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).PasswordStr)
            DBMain.Items.Add(Form1.SystemUsers(Counter).AcessLvInt)
            DBMain.Items.Add(Form1.SystemUsers(Counter).TestComName)
            DBMain.Items.Add(Form1.SystemUsers(Counter).CompBy)
            DBMain.Items.Add(Form1.SystemUsers(Counter).FinishedSta)
        End If
    Next
End Sub 'Search: looks for Usernames

```

```

'.....
.....
'Display Functions
Sub UserDisplay()
    DBUserInfo.Items.Clear() 'Adds the admins details into a display box
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).IdNumberInt)
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).UserNameStr)
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).EmialStr)
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).FNameStr)
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).SNameStr)
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).DoBStr)
    DBUserInfo.Items.Add("Password not shown")
    DBUserInfo.Items.Add(Form1.SystemUsers(Form1.UserIdNo).AcessLvInt)
End Sub 'Display: Shows logged in details
Sub DisplayBttn()
    DBMain.Items.Clear()
    For Counter = 1 To Form1.RecordNumber
        DBMain.Items.Add(Form1.SystemUsers(Counter).IdNumberInt)
        DBMain.Items.Add(Form1.SystemUsers(Counter).UserNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).EmialStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).FNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).SNameStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).DoBStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).PasswordStr)
        DBMain.Items.Add(Form1.SystemUsers(Counter).AcessLvInt)
        DBMain.Items.Add(Form1.SystemUsers(Counter).TestComName)
        DBMain.Items.Add(Form1.SystemUsers(Counter).CompBy)
        DBMain.Items.Add(Form1.SystemUsers(Counter).FinishedSta)
    Next
End Sub 'Displays user file onto list box
Private Sub DBMain_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
DBMain.SelectedIndexChanged
    If IsNumeric(DBMain.SelectedItem) Then
        TBId.Text = DBMain.SelectedItem
        TBUsername.Text = (Form1.SystemUsers(TBId.Text).UserNameStr)
        TBEmial.Text = (Form1.SystemUsers(TBId.Text).EmialStr)
        TBFName.Text = (Form1.SystemUsers(TBId.Text).FNameStr)
        TBSName.Text = (Form1.SystemUsers(TBId.Text).SNameStr)
        DTPforDOB.Text = (Form1.SystemUsers(TBId.Text).DoBStr)
        TBPASSWORD.Text = (Form1.SystemUsers(TBId.Text).PasswordStr)
        NumUDAccess.Text = (Form1.SystemUsers(TBId.Text).AcessLvInt)
    Else
        MsgBox("Please select the record number of the account you want")
    End If
End Sub 'Part of the program function: Loads info into text boxes and checks correct
correcte data is stord

Sub VPresenceCheck()
    If TBId.Text = "" Then 'Checks step by step to ensure the fields have been
entered
        MsgBox("Please enter the ID") 'Outputs an error message to the user of not
valid.
    ElseIf TBUsername.Text = "" Then
        MsgBox("Please enter the UserName")
    ElseIf TBEmial.Text = "" Then

```

```

        MsgBox("Please enter the Email")
    ElseIf TBFName.Text = "" Then
        MsgBox("Please enter the Users Name")
    ElseIf TBSName.Text = "" Then
        MsgBox("Please enter the Users Second name")
    ElseIf DTPforDOB.Text = "" Then
        MsgBox("Please enter the DOB")
    ElseIf TBPASSWORD.Text = "" Then
        MsgBox("Please enter the Password")
    ElseIf NumUDAccess.Text = "" Then
        MsgBox("Please enter the access level ")
    End If

End Sub 'Validation, precense check

'Buttons
Private Sub BtnDisAll_Click(sender As Object, e As EventArgs) Handles BtnDisAll.Click
    DisplayBttn()
End Sub 'Shows all user in file when clicked

Sub EmialBeforeChange()
    Dim RCText As String = " Record Changed: " &
(Form1.SystemUsers(TBId.Text).IdNumberInt)
    Dim HolderID As String = " ID=" & (Form1.SystemUsers(TBId.Text).UserNameStr)
    Dim HolderEmial As String = " Emial=" & (Form1.SystemUsers(TBId.Text).EmialStr)
    Dim HolderFName As String = " FirstName=" &
(Form1.SystemUsers(TBId.Text).FNameStr)
    Dim HolderSName As String = " SecondName=" &
(Form1.SystemUsers(TBId.Text).SNameStr)
    Dim HolderDob As String = " DateOfBirth=" & (Form1.SystemUsers(TBId.Text).DoBStr)
    Dim HolderPass As String = " Password=" &
(Form1.SystemUsers(TBId.Text).PasswordStr) 'Gathers all the data from the users record
whos to be
    Dim HolderLv As String = " AccesLevel=" &
(Form1.SystemUsers(TBId.Text).AcessLvInt) 'changes, and emails it to a secure email.

    Dim objOutlook As Object
    Dim objOutlookMsg As Object
    objOutlook = CreateObject("Outlook.Application")
    objOutlookMsg = objOutlook.CreateItem(0)
    With objOutlookMsg
        .To = "T0043344@cardinalnewman.ac.uk"
        .Subject = "Record Befoer Change"
        .Body = RCText & ". This was changed by: Record details before change: " &
HolderID & HolderEmial & HolderFName & HolderSName & HolderDob & HolderPass & HolderLv
        .Send()
    End With
    objOutlookMsg = Nothing
    objOutlook = Nothing
    MsgBox("Your Message Has Been Sent")
End Sub

Private Sub BtnSaveChange_Click(sender As Object, e As EventArgs) Handles
BtnSaveChange.Click
    CheckFiledEntered()

```

```

        If Proceed = True Then
            EmailBeforeChange()
            EditSave() 'Be nice to add "are you sure box"
            MsgBox("The data has been changed")
            DisplayBttn()
        End If
    End Sub 'Saves changes, then display them

    Private Sub BTNSearch_Click(sender As Object, e As EventArgs) Handles BTNSearch.Click
        If RadSurname.Checked Then
            SurnameSearch()
        ElseIf RADEmail.Checked Then
            EmailSearch()
        ElseIf RADUserName.Checked Then
            UsernameSearch()
        End If
        MsgBox("Search Completed")
    End Sub 'Button: loads up correct search function (via radio buttons), Searches

    Private Sub BtnChangeMes_Click(sender As Object, e As EventArgs) Handles
    BtnChangeMes.Click
        SW = New StreamWriter("AdminMes.txt") 'Saves the message into the admin
    file
        SW.Write(TbInputAdminMes.Text)
        SW.Close()
        MsgBox("The Message Has Been Saved And Can Now Be Viewd By All Users.")
    End Sub

    Dim Proceed As Boolean
    Sub CheckFiledEntered()
        Proceed = False
        If TBUsername.Text = "" Or TBFName.Text = "" Or TBSName.Text = "" Or TBEmail.Text
    = "" Or TBPASSWORD.Text = "" Or TBID.Text = "" Then
            MsgBox("Please Fill all the fields") 'Presence check to ensure all the
    fields are filled
        Else
            Proceed = True
        End If
    End Sub

    Private Sub CbProgramN_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
    CbProgramN.SelectedIndexChanged
        For Counter = 1 To RecNo
            If CbProgramN.SelectedItem = (ProgramLocation(Counter).Name) Then
                TbPath.Text = ProgramLocation(Counter).Location
            'Loads up the description and location on screen
                TbDes.Text = ProgramLocation(Counter).Description
            End If
        Next
    End Sub

    Private Sub BtnFileLoc_Click(sender As Object, e As EventArgs) Handles
    BtnFileLoc.Click
        Dim counter As Integer = 0
        Dim ofd As New OpenFileDialog 'This part opens a file view and filers
    for text files
        ofd.FileName = ""

```

```

        ofd.Title = "Select Program"
        ofd.ShowDialog()
        If ofd.FileName = "" Then                                'Needed to stop
crashing if cancelled
            MsgBox("Please select a file!", MsgBoxStyle.Critical)
        Else
            TbPath.Text = ofd.FileName
        End If
    End Sub

    Private Sub BtnAddPro_Click(sender As Object, e As EventArgs) Handles BtnAddPro.Click
        If (TbPath.Text = "") Or (TbDes.Text = "") Or (CbProgramN.Text = "") Then
'ensures no fields are left empty
            MsgBox("Please ensure all the fields are filled")          'shows messagebox
if the field is empty
        Else
            For Counter = 1 To RecNo
                If CbProgramN.Text = (ProgramLocation(Counter).Name) = True Then
                    MsgBox("Please Enter a different name")
                Else
                    AddProgram()                                     'Saves
the new data
                    MsgBox("The program has been added")           'Tells
user
                    LoadCbProgramNames() 'resets the combo box
                End If
            Next

        End If
    End Sub

    Private Sub BtnSubPro_Click(sender As Object, e As EventArgs) Handles BtnSubPro.Click
        ReadProgramFile()                                       'read the file to save to the array
        Dim Add As Integer = 0
        Dim found As Boolean = False
        Counter = 1
        Do Until Counter = RecNo
            If ProgramLocation(Counter).Name = CbProgramN.Text Then
                found = True
                Add = 1
            End If
            ProgramLocation(Counter).Reccord = ProgramLocation(Counter).Reccord
            ProgramLocation(Counter).Name = ProgramLocation(Counter + Add).Name '(Counter
+ Add) allows skipping of a record
            ProgramLocation(Counter).Description = ProgramLocation(Counter +
Add).Description ' for deletion
            ProgramLocation(Counter).Location = ProgramLocation(Counter + Add).Location
            Counter = Counter + 1
        Loop
        If found = True Then                                     'If statemtn used to stop random
entries being lost when delete functuon is run
            SW = New StreamWriter("ProgramLocation.txt") 'Saves the data to the file
            SW.WriteLine(RecNo - 1)
            For Counter = 1 To RecNo - 1
                SW.WriteLine(ProgramLocation(Counter).Reccord)
                SW.WriteLine(ProgramLocation(Counter).Name)
                SW.WriteLine(ProgramLocation(Counter).Description)
                SW.WriteLine(ProgramLocation(Counter).Location)
            Next
        End Sub
    End Sub

```

```

        Next
        SW.Close()
        MsgBox("Deleted")
    Else
        MsgBox("Please select an item to be deleted")
    End If
    LoadCbProgramNames() 'Resets the names in the combo box
End Sub
End Class

```

## Teacher -

```

Imports System.IO
Imports KeyEventArgs.SuppressKeyPress
Public Class TeacherPage
    Dim SR As StreamReader
    Dim SW As StreamWriter
    Public TestDataID As Integer
    Public TestAmount As Integer
    Dim Counter As Integer = 0

    Private Sub TeacherPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadSearchCombobox()
        LoadComboBox()
        LoadComboBoxLearn()
    End Sub 'When Page opens Fills the ComboBox
    Sub LoadComboBox()
        ReadTestLayoutData()
        For Counter = 1 To TestDataID
            If CbTestNames.Items.Contains(TestQ(Counter).TestName) = False Then
                CbTestNames.Items.Add(TestQ(Counter).TestName)
            End If
        Next
    End Sub 'Used again
    if changes are made
        Sub LoadComboBoxLearn()
            ReadLearnFile() 'When the page opens, it loads in the Info
            CbLearnDataNames.Items.Clear()
            For Counter = 0 To NoOfInfo - 1 'To the combo box, for the user to select
                CbLearnDataNames.Items.Add(LearnArray(Counter).Name) 'data they wany to look
            Next
        End Sub 'Main use is here
    tho
    Sub LoadSearchCombobox()
        ReadTestDataFile()
        For Counter = 1 To TestAmount
            If CbSearch.Items.Contains(TestArray(Counter).TestName) = False Then
                CbSearch.Items.Add(TestArray(Counter).TestName)
            End If
        Next
    End Sub

```

```

'.....
.....
'Test Layout - Read,Write,search

Structure TestQuestions
    Dim TestIDno As Integer
    Dim TestQno As Integer
    Dim TestName As String
    Dim Q As String
    Dim A As String
    Dim B As String
    Dim C As String
    Dim D As String
End Structure 'Structure holding the TestQuestions

Public TestQ() As TestQuestions
Public TemoTestQ() As TestQuestions

Public Sub ReadTestLayoutData()
    SR = New StreamReader("TestLayout.txt") 'Reads the files, loading up the records
into the array
    TestDataID = SR.ReadLine
    ReDim Preserve TestQ(TestDataID)
    Counter = 0
    Do Until SR.EndOfStream = True
        Counter = Counter + 1
        TestQ(Counter).TestIDno = SR.ReadLine()
        TestQ(Counter).TestQno = SR.ReadLine()
        TestQ(Counter).TestName = SR.ReadLine()
        TestQ(Counter).Q = SR.ReadLine()
        TestQ(Counter).A = SR.ReadLine()
        TestQ(Counter).B = SR.ReadLine()
        TestQ(Counter).C = SR.ReadLine()
        TestQ(Counter).D = SR.ReadLine()
    Loop
    SR.Close()
End Sub 'Read Test Questions File

Sub LayoutIntoFile()
    SR = New StreamReader("TestLayout.txt") 'Saves the test layout into the file
    TestDataID = SR.ReadLine
    SR.Close()
    ReadTestLayoutData()
    QonIS()
    TestDataID = TestDataID + 1
    ReDim Preserve TestQ(TestDataID)
    TestQ(TestDataID).TestIDno = TestDataID
    TestQ(TestDataID).TestQno = NumUpDn.Value
    TestQ(TestDataID).TestName = CbTestNames.Text
    TestQ(TestDataID).Q = TBQuestion.Text
    TestQ(TestDataID).A = TBA.Text
    TestQ(TestDataID).B = TBB.Text
    TestQ(TestDataID).C = TBC.Text
    TestQ(TestDataID).D = TBD.Text

```

```

SW = New StreamWriter("TestLayout.txt")
SW.WriteLine(TestDataID)
For Counter = 1 To TestDataID
    SW.WriteLine(TestQ(Counter).TestIDno)
    SW.WriteLine(TestQ(Counter).TestQno)
    SW.WriteLine(TestQ(Counter).TestName)
    SW.WriteLine(TestQ(Counter).Q)
    SW.WriteLine(TestQ(Counter).A)
    SW.WriteLine(TestQ(Counter).B)
    SW.WriteLine(TestQ(Counter).C)
    SW.WriteLine(TestQ(Counter).D)
Next
SW.Close()
End Sub 'Saves The Test Layout into the file

Public holdQno As Integer
Sub QonIS()
    holdQno = 0
    If TestQ(TestDataID).TestName = CbTestNames.Text Then
        If TestQ(TestDataID).TestQno > holdQno Then
            holdQno = TestQ(TestDataID).TestQno
        End If
    End If
End Sub 'Makes sure the Q no is in the right order (used in LayoutFile)

Sub SearchLayoutAll()
    ReadTestLayoutData()
    DBTestMaker.Items.Clear()
    For Counter = 1 To TestDataID
        DBTestMaker.Items.Add(TestQ(Counter).TestIDno)
        DBTestMaker.Items.Add(TestQ(Counter).TestQno)
        DBTestMaker.Items.Add(TestQ(Counter).TestName)
        DBTestMaker.Items.Add(TestQ(Counter).Q)
        DBTestMaker.Items.Add(TestQ(Counter).A)
        DBTestMaker.Items.Add(TestQ(Counter).B)
        DBTestMaker.Items.Add(TestQ(Counter).C)
        DBTestMaker.Items.Add(TestQ(Counter).D)
    Next
End Sub 'Displays the Test Layout data into the ListBox

Private Sub BTNAddQuestion_Click(sender As Object, e As EventArgs) Handles
BTNAddQuestion.Click
    Dim Allow As Boolean = True
    ReadLearnFile()
    For Counter = 1 To TestDataID
        If (NumUpDn.Value > 5) Or ((TestQ(Counter).TestName = CbTestNames.Text) And
(TestQ(Counter).TestQno = NumUpDn.Value)) Then
            Allow = False
        End If
    Next
    If Allow = True Then
        LayoutIntoFile()
        OneNameTest()
    Else
        MsgBox("The test already has that test name and question number")
    End If

```

The file has the record number written into it. Then the rest of the structured array in order to save the data.

This checks that the teacher can add in a question number lower than one that has already been made, this prevents the data being repeated or incorrectly entered.

```

End Sub 'Check if the Test has already been made

Private Sub BTNDisplayLayout_Click(sender As Object, e As EventArgs) Handles
BTNDisplayLayout.Click
    SearchLayoutAll()
End Sub 'Btn - loads al Test Layout stuff into list box
Private Sub BtnSearchByTest_Click(sender As Object, e As EventArgs) Handles
BtnSearchByTest.Click
    OneNameTest()
End Sub 'Btn - Loads only test already selected
Sub OneNameTest()
    DBTestMaker.Items.Clear()
    ReadTestLayoutData()
    For Counter = 1 To TestDataID
        If TestQ(Counter).TestName = CbTestNames.Text Then
            DBTestMaker.Items.Add(TestQ(Counter).TestIDno)
            DBTestMaker.Items.Add(TestQ(Counter).TestQno)
            DBTestMaker.Items.Add(TestQ(Counter).TestName)
            DBTestMaker.Items.Add(TestQ(Counter).Q)
            DBTestMaker.Items.Add(TestQ(Counter).A)
            DBTestMaker.Items.Add(TestQ(Counter).B)
            DBTestMaker.Items.Add(TestQ(Counter).C)
            DBTestMaker.Items.Add(TestQ(Counter).D)

        End If
    Next

End Sub 'Loads only test already selected Activated by Btnn

'.....
'.....
'Layout that student test score data follows - Read,Search

Structure TestData
    Dim TestNo As Integer
    Dim TestName As String
    Dim UserName As String
    Dim Score As Integer
    Dim Time As Integer
    Dim Mark As Integer
End Structure 'Holds the students score, time, etc
Public TestArray() As TestData

Public Sub ReadTestDataFile() 'Opens and reads the user file
    SR = New StreamReader("TestScore.txt")
    TestAmount = SR.ReadLine 'Gets Num of records from first
line of text
    ReDim Preserve TestArray(TestAmount) 'Makes the array the same size as the
text file
    Counter = 0
    Do Until SR.EndOfStream = True
        Counter = Counter + 1
        TestArray(Counter).TestNo = SR.ReadLine()
        TestArray(Counter).TestName = SR.ReadLine()
        TestArray(Counter).UserName = SR.ReadLine()
        TestArray(Counter).Score = SR.ReadLine()
    
```

```

        TestArray(Counter).Time = SR.ReadLine()
        TestArray(Counter).Mark = SR.ReadLine()
    Loop
    SR.Close()
End Sub ' Reads the file

Sub SearchUsername()
    ReadTestDataFile()
    DBMain.Items.Clear()
    For Counter = 1 To TestAmount
        If TestArray(Counter).UserName = TBUsearch.Text Then
            DBMain.Items.Add(TestArray(Counter).TestNo)
            DBMain.Items.Add(TestArray(Counter).TestName)
            DBMain.Items.Add(TestArray(Counter).UserName)
            DBMain.Items.Add(TestArray(Counter).Score)
            DBMain.Items.Add(TestArray(Counter).Time)
            DBMain.Items.Add(TestArray(Counter).Mark)
        End If
    Next
End Sub ' Search By username, only displays records with the correct username
Sub SearchAll()
    ReadTestDataFile()
    DBMain.Items.Clear()
    For Counter = 1 To TestAmount
        DBMain.Items.Add(TestArray(Counter).TestNo)
        DBMain.Items.Add(TestArray(Counter).TestName)
        DBMain.Items.Add(TestArray(Counter).UserName)
        DBMain.Items.Add(TestArray(Counter).Score)
        DBMain.Items.Add(TestArray(Counter).Time)
        DBMain.Items.Add(TestArray(Counter).Mark)
    Next
End Sub 'Displays all the records
Sub SearchTest()
    ReadTestDataFile()
    DBMain.Items.Clear()
    For Counter = 1 To TestAmount
        If TestArray(Counter).TestName = CbSearch.Text Then
            DBMain.Items.Add(TestArray(Counter).TestNo)
            DBMain.Items.Add(TestArray(Counter).TestName)
            DBMain.Items.Add(TestArray(Counter).UserName)
            DBMain.Items.Add(TestArray(Counter).Score)
            DBMain.Items.Add(TestArray(Counter).Time)
            DBMain.Items.Add(TestArray(Counter).Mark)
        End If
    Next
End Sub ' Only searches for the records with the correct test name

Private Sub BtnSearch_Click(sender As Object, e As EventArgs) Handles BtnSearch.Click
    If RADUname.Checked Then 'Sees which radbtn has been selected
        SearchUsername() 'Runs correct code responding with
button
    ElseIf RADAll.Checked Then
        SearchAll()
    ElseIf RADTest.Checked = True Then
        SearchTest()
    End If
End Sub

```

```
End If
End Sub 'Search Select Option - via radio bttns
```

```
'.....
'.....
'Only Learn DataFile Stuff here
```

```
Structure Learn
    Dim Name As String
    Dim Info As String
End Structure
Public LearnArray() As Learn
Public TemoLearnArray() As Learn
Public NoOfInfo As Integer
Public Sub ReadLearnFile()
    SR = New StreamReader("LearnData.txt")
    NoOfInfo = SR.ReadLine
    ReDim Preserve LearnArray(NoOfInfo - 1)
    Counter = 0
    Do Until SR.EndOfStream = True
        LearnArray(Counter).Name = SR.ReadLine()
        LearnArray(Counter).Info = SR.ReadLine()
        Counter = Counter + 1
    Loop
    SR.Close()
End Sub 'Read Learn File
Sub SaveLearnFile()
    SR = New StreamReader("LearnData.txt")
    NoOfInfo = SR.ReadLine
    SR.Close()
    ReadLearnFile()
    ReDim Preserve LearnArray(NoOfInfo)
    LearnArray(NoOfInfo).Name = CbLearnDataNames.Text
    LearnArray(NoOfInfo).Info = TBLearnData.Text
    SW = New StreamWriter("LearnData.txt")
    SW.WriteLine(NoOfInfo + 1)
    For Counter = 0 To NoOfInfo
        SW.WriteLine(LearnArray(Counter).Name)
        SW.WriteLine(LearnArray(Counter).Info)
    Next
    SW.Close()
End Sub ' Save Lean File
```

```
Private Sub BtnLearnSave_Click(sender As Object, e As EventArgs) Handles
BtnLearnSave.Click
```

```
    If CbLearnDataNames.Items.Contains(CbLearnDataNames.Text) = True Then
'Tells user if there is already an existing record
        Dim style = MsgBoxStyle.YesNo
'Gives them a choice to change it, or leave it
        Dim Result As Integer
        Result = MsgBox("There is already an entry with the same name, Do you wish to
overwrite it? ", style)
```

```

        If Result = 6 Then
            DeleteTopicSaveNew()
        End If
    Else
        SaveLearnFile()
    End If
    TBLearnData.Clear()
    Sort() 'Sorts using bubble so array is in order
    LoadComboBoxLearn() 'reloads the data in the combobox to make sure user can
reselect without chaning page.
End Sub 'Btn- Checks, sorts, and saves the lean file

Sub DeleteTopicSaveNew()
    SR = New StreamReader("LearnData.txt")
    NoOfInfo = SR.ReadLine
    SR.Close()
    ReadLearnFile()
    ReDim Preserve LearnArray(NoOfInfo - 1)
    LearnArray(index).Name = CbLearnDataNames.Text
    LearnArray(index).Info = TBLearnData.Text
    SW = New StreamWriter("LearnData.txt")
    SW.WriteLine(NoOfInfo)
    For Counter = 0 To NoOfInfo - 1
        SW.WriteLine(LearnArray(Counter).Name)
        SW.WriteLine(LearnArray(Counter).Info)
    Next
    SW.Close()
End Sub 'Allows Learn File Record to be re-written
Private Sub BtnLearnLoad_Click(sender As Object, e As EventArgs) Handles
BtnLearnLoad.Click
    ReadLearnFile()
    For Counter = 0 To NoOfInfo - 1
        If CbLearnDataNames.Text = LearnArray(Counter).Name Then 'the matching text
needed for it.
            TBLearnData.Text = LearnArray(Counter).Info
        End If
    Next
End Sub 'Btn - When Loading in data, Retrives recoded with the correct name
'^Could change to a Binary search - Check code when time

'.....
'.....
'Sorting And Binary

Sub Sort()
    ReadLearnFile()           'Read
    BubbleWithSetArray()      'Bubble
    ReSaveLearn()             'Save
End Sub ' Calls multipul other subs (easer than re writing)

Sub BubbleWithSetArray()
    Dim ArrayLength As Integer = LearnArray.Length - 1           'Has to be -1 here
otherwise out of bound
    Dim Temp As String
    Dim TempINFO As String

```

```

Dim Swapp As Boolean
Do
    Swapp = False
    For Counter = 0 To ArrayLength - 1 'Sets to the array size
        If LearnArray(Counter).Name > LearnArray(Counter + 1).Name Then 'if
the name is greater thean the next name, then it sorts it inti order
            Temp = LearnArray(Counter).Name
            TempINFO = LearnArray(Counter).Info
            LearnArray(Counter).Name = LearnArray(Counter + 1).Name
            LearnArray(Counter).Info = LearnArray(Counter + 1).Info
            LearnArray(Counter + 1).Name = Temp 'Temps holds the item for a brief
moment

            LearnArray(Counter + 1).Info = TempINFO
            Swapp = True
        End If
    Next
    DBMain.Items.Clear() 'Clears the item in the display box
Loop Until Swapp = False
For Counter = 0 To (LearnArray.Length - 1)
    DBMain.Items.Add(LearnArray(Counter).Name) 'Adds items to the displaybox
    DBMain.Items.Add(LearnArray(Counter).Info)
Next
End Sub
Sub ReSaveLearn() 'Saves the learn file
    SW = New StreamWriter("LearnData.txt")
    SW.WriteLine(NoOfInfo)
    For Counter = 0 To NoOfInfo - 1 'rReads the records in one by one
        SW.WriteLine(LearnArray(Counter).Name)
        SW.WriteLine(LearnArray(Counter).Info)
    Next
    SW.Close()
End Sub 'Resaves the sorted file

```

```

Dim index As Integer
Private Sub CbLearnDataNames_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles CbLearnDataNames.SelectedIndexChanged
    index = CbLearnDataNames.SelectedIndex 'when the combo box's value is changeg,
the holder is changed
End Sub

```

```

'.....
.....

```

```

Sub ReadToDel() 'reads the file getting it ready to delete
    SR = New StreamReader("TestLayout.txt")
    TestDataID = SR.ReadLine 'reads the first line og the file saveing the recnumber
    ReDim Preserve TestQ(TestDataID)
    Counter = 0
    Do Until Counter = TestDataID - 5 'Skips five for the record fieds
        Counter = Counter + 1 'as the 'old record takes up five slots
    Loop

```

```

    TestQ(Counter).TestIDno = SR.ReadLine()
    TestQ(Counter).TestQno = SR.ReadLine()
    TestQ(Counter).TestName = SR.ReadLine()
    TestQ(Counter).Q = SR.ReadLine()
    TestQ(Counter).A = SR.ReadLine()
    TestQ(Counter).B = SR.ReadLine()
    TestQ(Counter).C = SR.ReadLine()
    TestQ(Counter).D = SR.ReadLine()
Loop
SR.Close()
End Sub

```

```

Private Sub BtnDelTest_Click(sender As Object, e As EventArgs) Handles
BtnDelTest.Click
    ReadTestLayoutData() 'reads the data to know whats in the file
    Dim Add As Integer = 0
    Dim found As Boolean = False
    ReDim TemoTestQ(TestDataID - 1)
    Counter = 1
    Do Until Counter = TestDataID 'Loops
        If TestQ(Counter).TestIDno = DBTestMaker.SelectedItem Then
            found = True 'if the correct one is found
            Add = 1 'changes status of add
        End If
        TemoTestQ(Counter).TestIDno = TestQ(Counter).TestIDno 'Stays the same
regardless
        TemoTestQ(Counter).TestQno = TestQ(Counter + Add).TestQno 'when the
value
        TemoTestQ(Counter).TestName = TestQ(Counter + Add).TestName ' of 'add' is
changed it moves the
        TemoTestQ(Counter).Q = TestQ(Counter + Add).Q 'records up one therefor
        TemoTestQ(Counter).A = TestQ(Counter + Add).A 'it changes the reccord
        TemoTestQ(Counter).B = TestQ(Counter + Add).B
        TemoTestQ(Counter).C = TestQ(Counter + Add).C
        TemoTestQ(Counter).D = TestQ(Counter + Add).D
        Counter = Counter + 1
    Loop
    DBTestMaker.Items.Clear()
    For Counter = 1 To TestDataID - 1
        DBTestMaker.Items.Add(TemoTestQ(Counter).TestIDno) 'Clears the display box
        DBTestMaker.Items.Add(TemoTestQ(Counter).TestQno) 'then shows the data
        DBTestMaker.Items.Add(TemoTestQ(Counter).TestName)
        DBTestMaker.Items.Add(TemoTestQ(Counter).Q)
        DBTestMaker.Items.Add(TemoTestQ(Counter).A)
        DBTestMaker.Items.Add(TemoTestQ(Counter).B)
        DBTestMaker.Items.Add(TemoTestQ(Counter).C)
        DBTestMaker.Items.Add(TemoTestQ(Counter).D)
    Next
    SW = New StreamWriter("TestLayout.txt")
    SW.WriteLine(TestDataID - 1)
    For Counter = 1 To TestDataID - 1
        SW.WriteLine(TemoTestQ(Counter).TestIDno) 'saves the data as its written
        SW.WriteLine(TemoTestQ(Counter).TestQno) 'rinto a fiel
        SW.WriteLine(TemoTestQ(Counter).TestName)
        SW.WriteLine(TemoTestQ(Counter).Q)
        SW.WriteLine(TemoTestQ(Counter).A)
        SW.WriteLine(TemoTestQ(Counter).B)
    Next

```

```

        SW.WriteLine(TemoTestQ(Counter).C)
        SW.WriteLine(TemoTestQ(Counter).D)
    Next
    SW.Close()
End Sub

Private Sub BtnInfoDel_Click(sender As Object, e As EventArgs) Handles
BtnInfoDel.Click
    ReadLearnFile() 'read the file to save to the array
    Dim Add As Integer = 0
    Dim found As Boolean = False
    ' ReDim TemoLearnArray(NoOfInfo - 2) 'resize the tempart array so there is one
less space than the real array
    Counter = 0
    Do Until Counter = NoOfInfo - 1
        If LearnArray(Counter).Name = CbLearnDataNames.Text Then
            found = True
            Add = 1
        End If
        LearnArray(Counter).Name = LearnArray(Counter + Add).Name
        LearnArray(Counter).Info = LearnArray(Counter + Add).Info
        Counter = Counter + 1
    Loop
    If found = True Then 'If statemtn used to stop random
entries being lost when delete functuon is run
        DBTestMaker.Items.Clear() 'shows data in
display boz *review if needed*
        For Counter = 0 To NoOfInfo - 2
            DBTestMaker.Items.Add(LearnArray(Counter).Name)
            DBTestMaker.Items.Add(LearnArray(Counter).Info)
        Next
        SW = New StreamWriter("LearnData.txt") 'Saves the data to the file
        SW.WriteLine(NoOfInfo - 1)
        For Counter = 0 To NoOfInfo - 2
            SW.WriteLine(LearnArray(Counter).Name)
            SW.WriteLine(LearnArray(Counter).Info)
        Next
        SW.Close()
    Else
        MsgBox("Please select an item to be deleted")
    End If
    LoadComboBoxLearn() 'shows the new list availab;pe
End Sub

Private Sub TBLearnData_KeyDown(ByVal sender As System.Object, _
ByVal e As System.Windows.Forms.KeyEventArgs) _
Handles TBLearnData.KeyDown
    If e.KeyCode = Keys.Enter Then
        e.SuppressKeyPress = True
    End If
End Sub ' Validation, stops enter key being pressedf and breaking it

Private Sub BtnSetTest_Click(sender As Object, e As EventArgs) Handles
BtnSetTest.Click
    SetTests.Show()
End Sub

```

```

Private Sub BtnToGra_Click(sender As Object, e As EventArgs) Handles BtnToGra.Click
    ChartPage.Show()
End Sub
End Class

```

## Chart-

```
Public Class ChartPage
```

```

Private Sub ChartPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    TeacherPage.ReadTestLayoutData() 'Upon the page loading, populstes all the
necessary list and combo boxes

```

```

For Counter = 1 To TeacherPage.TestDataID ' with all the correct data
    If DbTestCha.Items.Contains(TeacherPage.TestQ(Counter).TestName) = False Then
        DbTestCha.Items.Add(TeacherPage.TestQ(Counter).TestName)

```

```

CbTestCharTwo.Items.Add(TeacherPage.TestQ(Counter).TestName)
    End If
Next
End Sub

```

The student's data structured array is checked for if it has the correct test, if so the students data is added to the graph.

```

Private Sub BtnPopChart_Click(sender As Object, e As EventArgs)
Handles BtnPopChart.Click
    BtnPopChart.Visible = False 'When the buttons is
pressed, loads up the correct data in the graph
    TeacherPage.ReadTestDataFile() 'checks first to see if it needs to
comape tests or just display the data
    If CompCkB.Checked = True Then
        TeacherPage.ReadTestDataFile()
        For Counter = 1 To TeacherPage.TestAmount
            If TeacherPage.TestArray(Counter).TestName = DbTestCha.SelectedItem Then

```

```

Me.ChaStuSco.Series("Compare").Points.AddXY(TeacherPage.TestArray(Counter).UserName,
TeacherPage.TestArray(Counter).Mark)
                ElseIf TeacherPage.TestArray(Counter).TestName =
CbTestCharTwo.SelectedItem Then

```

```

Me.ChaStuSco.Series("Users").Points.AddXY(TeacherPage.TestArray(Counter).UserName,
TeacherPage.TestArray(Counter).Mark)
            End If

```

```

Next
Else

```

```

For Counter = 1 To TeacherPage.TestAmount
    If TeacherPage.TestArray(Counter).TestName = DbTestCha.SelectedItem Then

```

```

Me.ChaStuSco.Series("Users").Points.AddXY(TeacherPage.TestArray(Counter).UserName,
TeacherPage.TestArray(Counter).Mark)
            End If

```

```

Next
End If
End Sub

```

```

Private Sub CompCkB_CheckedChanged(sender As Object, e As EventArgs) Handles
CompCkB.CheckedChanged

```

```

        CbTestCharTwo.Visible = True
    End Sub

End Class

```

### Set Tests-

```

Imports System.IO
Public Class SetTests
    Dim SW As StreamWriter

    Private Sub SetTests_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadTestnames() 'Loads up the test names into the list box
        LoadCompletedAndNot() 'loads up studens names in the correct colour and list
box
        For Counter = 1 To SignIn.RecordNumber
            CbOneStu.Items.Add(SignIn.SystemUsers(Counter).UserNameStr) 'Reads into
display Box
        Next
    End Sub
    Sub LoadTestnames() 'Loads up all the test names into the list box to be displayed
        TeacherPage.ReadTestLayoutData()
        For Counter = 1 To TeacherPage.TestDataID
            If DbTestName.Items.Contains(TeacherPage.TestQ(Counter).TestName) = False
Then 'If not already added, adds it
                DbTestName.Items.Add(TeacherPage.TestQ(Counter).TestName)
            End If
        Next
    End Sub
    Sub LoadCompletedAndNot()
        For Counter = 1 To SignIn.RecordNumber 'loads student names
            If (SignIn.SystemUsers(Counter).FinishedSta) = False Then 'adds them to
correct display box
                DbToDo.Items.Add(SignIn.SystemUsers(Counter).UserNameStr)
            ElseIf (SignIn.SystemUsers(Counter).CompBy > Today) And
(SignIn.SystemUsers(Counter).FinishedSta) = False Then
                DbDeadline.Items.Add(SignIn.SystemUsers(Counter).UserNameStr) 'Adds to
displaybox
            End If
        Next
    End Sub

    Private Sub BtnSetTest_Click(sender As Object, e As EventArgs) Handles
BtnSetTest.Click
        SignIn.ReadUserFile() 'Read the the file to obtain the most upt to date
version
        CheckRad() 'moves onto the next part
    End Sub

    Sub CheckRad()
        If RadAll.Checked = True Then
            CbOneStu.Visible = False
            Dim style = MsgBoxStyle.YesNo 'Sends a messaeg box to the user so make sure
they wish to proceed
            Dim Result As Integer

```

```

        Result = MsgBox("Are you sure, this wil change all student set work? ",
style) 'Messae to user
        If Result = 6 Then
            AllStudentEdit()                'If yes selected
            MsgBox("The Tests have been set") 'Edits all users
        End If

        ElseIf RadOne.Checked = True Then
            If CbOneStu.SelectedItem = "" Then
                MsgBox("Plese slect a student")
            Else
                OneStudentEdit()
                MsgBox("Done")
            End If
        End If
    End Sub

Sub OneStudentEdit()
    Call SignIn.ReadUserFile()
    ReDim Preserve SignIn.SystemUsers(SignIn.RecordNumber)
    SignIn.SystemUsers(UserRecNo).TestComName = DbTestName.SelectedItem
    SignIn.SystemUsers(UserRecNo).CompBy = DaTPick.Value
    SignIn.SystemUsers(UserRecNo).FinishedSta = False
    SW = New StreamWriter("UserData.txt")
    SW.WriteLine(SignIn.RecordNumber)           'Writes in the number of records (to be
read for next time)
    For Counter = 1 To SignIn.RecordNumber     'Does not need to be decprrted as its
written in the form
        SW.WriteLine(SignIn.SystemUsers(Counter).IdNumberInt)
        SW.WriteLine(SignIn.SystemUsers(Counter).UserNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).EmialStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).FNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).SNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).DoBStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).PasswordStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).AcessLvInt)
        SW.WriteLine(SignIn.SystemUsers(Counter).TestComName)
        SW.WriteLine(SignIn.SystemUsers(Counter).CompBy)
        SW.WriteLine(SignIn.SystemUsers(Counter).FinishedSta)
    Next
    SW.Close()
    SignIn.EncryptFile()
End Sub 'Saves into file, rewriting the data for one user only

Sub AllStudentEdit()
    Call SignIn.ReadUserFile()
    ReDim Preserve SignIn.SystemUsers(SignIn.RecordNumber)
    For Counter = 1 To SignIn.RecordNumber
        SignIn.SystemUsers(Counter).TestComName = DbTestName.SelectedItem
        SignIn.SystemUsers(Counter).CompBy = DaTPick.Value
        SignIn.SystemUsers(Counter).FinishedSta = False
    Next
    SW = New StreamWriter("UserData.txt")
    SW.WriteLine(SignIn.RecordNumber)           'Writes in the number of records (to be
read for next time)

```

```
For Counter = 1 To SignIn.RecordNumber 'Does not need to be decrypted as its  
written in the form
```

```
SW.WriteLine(SignIn.SystemUsers(Counter).IdNumberInt)  
SW.WriteLine(SignIn.SystemUsers(Counter).UserNameStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).EmailStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).FNameStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).SNameStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).DoBStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).PasswordStr)  
SW.WriteLine(SignIn.SystemUsers(Counter).AccessLvInt)  
SW.WriteLine(SignIn.SystemUsers(Counter).TestComName)  
SW.WriteLine(SignIn.SystemUsers(Counter).CompBy)  
SW.WriteLine(SignIn.SystemUsers(Counter).FinishedSta)
```

```
Next  
SW.Close()  
SignIn.EncryptFile()  
End Sub 'Saves into file, rewriting the data for one user only
```

```
Dim UserRecNo As Integer  
Private Sub CbOneStu_SelectedIndexChanged(sender As Object, e As EventArgs) Handles  
CbOneStu.SelectedIndexChanged  
UserRecNo = CbOneStu.SelectedIndex + 1 'Used for finding the record  
number of the chosen student  
End Sub
```

```
End Class
```

## Menu –

```
Public Class MenuPage  
Private Sub BtnMain_Click(sender As Object, e As EventArgs) Handles BtnMain.Click  
Welcome.Show()  
End Sub  
  
Private Sub BtnBotPage_Click(sender As Object, e As EventArgs) Handles  
BtnBotPage.Click  
BotPage.Show()  
End Sub  
  
Private Sub BtnTests_Click(sender As Object, e As EventArgs) Handles BtnTests.Click  
TestPage.Show()  
End Sub  
  
Private Sub BtnBubble_Click(sender As Object, e As EventArgs) Handles BtnBubble.Click  
Algorithm.Show()  
End Sub  
  
Private Sub BtnBot_Click(sender As Object, e As EventArgs) Handles BtnBot.Click  
BotProgram.Show()  
End Sub  
  
Private Sub BtnBotRoom_Click(sender As Object, e As EventArgs) Handles  
BtnBotRoom.Click  
BotRoom.Show()  
End Sub
```

```

Private Sub BtnShowLearn_Click(sender As Object, e As EventArgs) Handles
BtnShowLearn.Click
    LearnPage.Show()
End Sub

Private Sub BtnTeacher_Click(sender As Object, e As EventArgs) Handles
BtnTeacher.Click
    TeacherPage.Show()
End Sub

Private Sub BtnAdmin_Click(sender As Object, e As EventArgs) Handles BtnAdmin.Click
    AdminPage.Show()
End Sub
End Class

```

### Learn –

```

Imports System.Net
Imports outlook = Microsoft.Office.Interop.Outlook
Public Class LearnPage

```

```

Private Sub Learn_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    SignIn.ReadUserFile()
    CbEmailPicker.Items.Clear()
    CbHidden.Items.Clear()
    For Counter = 1 To SignIn.RecordNumber
        If
CbEmailPicker.Items.Contains(SignIn.SystemUsers(Counter).UserNameStr) = False
And (SignIn.SystemUsers(Counter).AccessLvInt) < 3 Then
CbEmailPicker.Items.Add(SignIn.SystemUsers(Counter).UserNameStr) 'loads in data
into combo boxes
        CbHidden.Items.Add(SignIn.SystemUsers(Counter).EmailStr)
    End If
Next

    TeacherPage.ReadLearnFile() 'When the page opens,
it loads in the info
    For Counter = 0 To TeacherPage.NoOfInfo - 1 'To the combo box,
for the user to select the
        CbLearnTopics.Items.Add(TeacherPage.LearnArray(Counter).Name) 'data
they way to look at
    Next
End Sub

```

Adds in the teachers and admins emails and names to both the hidden and visible text box. This ensures that the students can not see the searchers email. When the selected item in one combo box is pressed, it matches the email in the other which is at the same location in the combo box.

```

'Search Method- use binary
Private Sub CbLearnTopics_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles CbLearnTopics.SelectedIndexChanged
    TeacherPage.ReadLearnFile()
    Setup()
    BinarySearch()
    TBInfo.Text = TeacherPage.LearnArray(ArrayNum).Info
End Sub

```

```

Private Sub BtnFindVid_Click(sender As Object, e As EventArgs) Handles
BtnFindVid.Click
    'uses flashplayer to allow a youtube video to be played for the user
    Dim VideoAdd() As String
    If (TBTemp.Text.IndexOf("youtube") <> -1) And (TBTemp.Text.IndexOf("watch?v=") <>
-1) Then 'if the correct link
        VideoAdd = Split(TBTemp.Text, "watch?v=") 'Splits the link into two parts
        VideoPlayerFla.Movie = VideoAdd(0) & "v/" & VideoAdd(1) 'uses the flashplar
to find the link online
    Else
        MessageBox.Show("Please enter an allowed link") 'else shows an error message
    End If
End Sub

```

```

Dim SearchFor As String
Dim StartNo As Integer
Dim EndNo As Integer
Dim LookingAt As Integer
Dim Found As Boolean = False
Dim ArrayNum As Integer
'Works hand in hand with the binary search, sets varibales so start it.
Sub Setup()
    SearchFor = CbLearnTopics.Text
    StartNo = 1
    EndNo = TeacherPage.LearnArray.Length
    Found = False
End Sub
'Performs a binary search on the learn data names to find the loactin of the correnct
one in the array
Sub BinarySearch()
    Do While Found = False
        LookingAt = ((StartNo + EndNo) / 2) 'selects number to search
        ArrayNum = LookingAt - 1
        If TeacherPage.LearnArray(ArrayNum).Name = SearchFor Then
            MsgBox((TeacherPage.LearnArray(ArrayNum).Name) & " " & SearchFor)
'Display found status to user
            Found = True 'set found to true
        ElseIf TeacherPage.LearnArray(ArrayNum).Name < SearchFor Then
            StartNo = LookingAt + 1 'sets searhcng start point to
looking at
        EndNo = EndNo
    Else
        StartNo = StartNo
        EndNo = LookingAt - 1 'sets end number to the looking at
point
    End If
    Loop
End Sub

```

```

Private Sub BtnEmail_Click(sender As Object, e As EventArgs) Handles BtnEmail.Click
Dim Address As String 'writes and sends the emial to the
teacher
Address = CbHidden.SelectedItem
If Address = "" Then

```

```

        MsgBox("Please select a teacher you wish to send the request too") 'Generates
email to be sent to teacher
    Else
        Dim objOutlook As Object
        Dim objOutlookMsg As Object
        objOutlook = CreateObject("Outlook.Application")
        objOutlookMsg = objOutlook.CreateItem(0)
        With objOutlookMsg
            .To = Address
            .Subject = "Student Question"
            .Body = "Test: " & CbLearnTopics.Text & " : Message: " &
TBEmailHelp.Text & " Sent From : " & (SignIn.SystemUsers(SignIn.UserIdNo).UserNameStr)
            .Send()
        End With
        objOutlookMsg = Nothing
        objOutlook = Nothing

        MsgBox("Your Message Has Been Sent")
    End If
End Sub

```

```

    Dim SendTo As String
    Private Sub CbEmailPicker_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles CbEmailPicker.SelectedIndexChanged
        Dim ChangeOver As Integer
        ChangeOver = CbEmailPicker.SelectedIndex 'behins the scenes convers the
teachers name into an emial
        CbHidden.SelectedIndex = ChangeOver 'In short has hidden text box, when user
slects one it change other with the relavent data
    End Sub

End Class

```

### Test –

```

Imports System.IO
Public Class TestPage
    Dim SR As StreamReader
    Dim SW As StreamWriter
    Dim NameLable As Label
    Dim ACheckBox, BCheckBox, CCheckBox, DCheckBox As CheckBox
    Dim Tcounter As Integer
    Dim Void As Boolean = False
    Dim Score As Integer = 0
    Dim RecordNumber As Integer

    'Save test score
    Sub SaveTestDataFile()
        SR = New StreamReader("TestScore.txt")
        RecordNumber = SR.ReadLine
        SR.Close()
        TeacherPage.ReadTestDataFile()
        RecordNumber = RecordNumber + 1
        ReDim Preserve TeacherPage.TestArray(RecordNumber)
        TeacherPage.TestArray(RecordNumber).TestNo = RecordNumber
        TeacherPage.TestArray(RecordNumber).TestName = CbtestNames.Text
        TeacherPage.TestArray(RecordNumber).UserName =
(SignIn.SystemUsers(SignIn.UserIdNo).UserNameStr)
    End Sub
End Class

```

```

TeacherPage.TestArray(RecordNumber).Score = Score
TeacherPage.TestArray(RecordNumber).Time = Tcounter
TeacherPage.TestArray(RecordNumber).Mark = (Score / Tcounter) * 100
SW = New StreamWriter("TestScore.txt")
SW.WriteLine(RecordNumber)
For Counter = 1 To RecordNumber
    SW.WriteLine(TeacherPage.TestArray(Counter).TestNo)
    SW.WriteLine(TeacherPage.TestArray(Counter).TestName)
    SW.WriteLine(TeacherPage.TestArray(Counter).UserName)
    SW.WriteLine(TeacherPage.TestArray(Counter).Score)
    SW.WriteLine(TeacherPage.TestArray(Counter).Time)
    SW.WriteLine(TeacherPage.TestArray(Counter).Mark)
Next
SW.Close()
MsgBox("Your score has been saved")
End Sub

Sub Mark()
    Dim Amount As Integer
    For counter = 1 To 5
        ACheckBox = Me.Controls("ACheckBox" & counter) 'checks the check boxes
        BCheckBox = Me.Controls("BCheckBox" & counter)
        CCheckBox = Me.Controls("CCheckBox" & counter)
        DCheckBox = Me.Controls("DCheckBox" & counter)
        If ACheckBox.Checked = True Then
            Amount = Amount + 1 'Adds one to a counter, marks right
            ACheckBox.ForeColor = Color.Red
        End If
        If BCheckBox.Checked = True Then
            Amount = Amount + 1
            BCheckBox.ForeColor = Color.Red
        End If
        If CCheckBox.Checked = True Then
            Amount = Amount + 1
            CCheckBox.ForeColor = Color.Red
        End If
        If DCheckBox.Checked = True Then
            Score = Score + 1
            DCheckBox.ForeColor = Color.LawnGreen
            Amount = Amount + 1
        End If
    Next
    If Amount > 5 Then
        MsgBox("You had too many posuble answers selected- Your score has been set to
0, please retake the test")
        Score = 0 'if too many have been selected tests is marked as all wrong
    End If
End Sub

Private Sub BtnStart_Click(sender As Object, e As EventArgs) Handles BtnStart.Click
    TeacherPage.ReadTestLayoutData()
    LayoutQ() 'sets out the test
    TestTimer.Start() ' Starts the timer
    BtnSave.Visible = True
End Sub

```

```

Sub LayoutQ()
    Dim RandomClass As New Random()
    Dim RememberSet As New HashSet(Of Integer)
    Dim RandomNumber As Integer = 0
    Dim Before As Integer
    Dim First As Integer = 25
    Dim Second As Integer = 260
    Dim RowOne As Integer = 160
    Dim RowTwo As Integer = 180
    Dim PointA As Point
    Dim PointB As Point
    Dim PointC As Point
    Dim PointD As Point

    For counter = 1 To TeacherPage.TestDataID
        If CbtestNames.Text = TeacherPage.TestQ(counter).TestName And Before <
TeacherPage.TestQ(counter).TestIDno Then
            Before = TeacherPage.TestQ(counter).TestQno
            NameLable = Me.Controls("Label" & TeacherPage.TestQ(counter).TestQno)
            NameLable.Text = TeacherPage.TestQ(counter).Q

            ACheckBox = Me.Controls("ACheckBox" & Before)
            BCheckBox = Me.Controls("BCheckBox" & Before)
            CCheckBox = Me.Controls("CCheckBox" & Before)
            DCheckBox = Me.Controls("DCheckBox" & Before)

            PointA = New Point(First, RowOne) 'Sets check boxes to their locations
            PointB = New Point(Second, RowOne)
            PointC = New Point(First, RowTwo)
            PointD = New Point(Second, RowTwo)

            RandomNumber = RandomClass.Next(1, 5) 'generates random number 1-4

            If (RandomNumber) Then
                If RandomNumber = 1 Then
                    ACheckBox.Location = PointA 'Randomises the locations they boxes
can be in
                    BCheckBox.Location = PointB
                    CCheckBox.Location = PointC
                    DCheckBox.Location = PointD

                ElseIf RandomNumber = 2 Then
                    ACheckBox.Location = PointB
                    BCheckBox.Location = PointD
                    CCheckBox.Location = PointC
                    DCheckBox.Location = PointA

                ElseIf RandomNumber = 3 Then
                    ACheckBox.Location = PointD
                    BCheckBox.Location = PointB
                    CCheckBox.Location = PointA
                    DCheckBox.Location = PointC

                ElseIf RandomNumber = 4 Then
                    ACheckBox.Location = PointA
                    BCheckBox.Location = PointC
                    CCheckBox.Location = PointB
                    DCheckBox.Location = PointD

            End If
        End For
    End Sub

```

```

        End If
        ACheckBox.Text = TeacherPage.TestQ(counter).A
        BCheckBox.Text = TeacherPage.TestQ(counter).B 'sets the location
        CCheckBox.Text = TeacherPage.TestQ(counter).C
        DCheckBox.Text = TeacherPage.TestQ(counter).D
        RowOne = RowOne + 120
        RowTwo = RowTwo + 120
    End If

    Next
End Sub
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles TestTimer.Tick
    Tcounter = Tcounter + 1
    TimerLab.Text = Tcounter 'shows the timer ticking
End Sub

Private Sub BtnSave_Click(sender As Object, e As EventArgs) Handles BtnSave.Click
    TestTimer.Stop()
    Mark()
    MsgBox(Score) 'stops the test, displays the score and saves
    SaveTestDataFile()
    MarkAsDone()
    Tcounter = 0
    BtnSave.Visible = False
End Sub

Sub MarkAsDone()
    Call SignIn.ReadUserFile()
    ReDim Preserve SignIn.SystemUsers(SignIn.RecordNumber)
    SignIn.SystemUsers(SignIn.UserIdNo).FinishedSta = True
    SW = New StreamWriter("UserData.txt")
    SW.WriteLine(SignIn.RecordNumber) 'Writes in the number of records (to be
read for next time)
    For Counter = 1 To SignIn.RecordNumber 'Does not need to be decrypted as its
written in the form
        SW.WriteLine(SignIn.SystemUsers(Counter).IdNumberInt)
        SW.WriteLine(SignIn.SystemUsers(Counter).UserNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).EmialStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).FNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).SNameStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).DoBStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).PasswordStr)
        SW.WriteLine(SignIn.SystemUsers(Counter).AcessLvInt)
        SW.WriteLine(SignIn.SystemUsers(Counter).TestComName)
        SW.WriteLine(SignIn.SystemUsers(Counter).CompBy)
        SW.WriteLine(SignIn.SystemUsers(Counter).FinishedSta)
    Next
    SW.Close()
    SignIn.EncryptFile()
End Sub

Private Sub TestPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    CbtestNames.Items.Clear()
    TeacherPage.ReadTestLayoutData()
    For Counter = 1 To TeacherPage.TestDataID
        If CbtestNames.Items.Contains(TeacherPage.TestQ(Counter).TestName) = False
Then
            CbtestNames.Items.Add(TeacherPage.TestQ(Counter).TestName)

```

```

        End If
    Next
End Sub

Private Sub BtnStartTest_Click(sender As Object, e As EventArgs)
    TeacherPage.ReadTestLayoutData()
    LayoutQ()
    TestTimer.Start()
    BtnSave.Visible = True
End Sub

```

End Class

### Simulation –

```
Public Class Algorithm
```

```

    Dim BubbleArray() As Integer = {44, 67, 75, 55, 999, 1, 23, 67, 3} ' set array
    Dim ChosenArraySiz As Integer
    Dim NewArray() As Integer
    Dim AC As Integer = 0

    Sub BubbleWithSetArray()
        Dim ArrayLength As Integer = BubbleArray.Length - 1           'Has to be -1 here
        otherwise array out of bound (computer registers 0 as first)
        Dim Temp As Integer                                           'Holds the stored
        value                                                         value
        Dim Swapp As Boolean                                           'Flag to show if pass
        was made
        Do
            Swapp = False
            For Counter = 0 To ArrayLength - 1
                If BubbleArray(Counter) > BubbleArray(Counter + 1) Then 'preforms passes,
                swaps numbers if necessary
                    Temp = BubbleArray(Counter)                       'stores the value
                to be moved
                    BubbleArray(Counter) = BubbleArray(Counter + 1)
                    BubbleArray(Counter + 1) = Temp
                    Swapp = True                                       'Tells if there
                is swapp is needed
                End If
            Next
            Loop Until Swapp = False
            For Counter = 0 To (BubbleArray.Length - 1)
                DbBubSor.Items.Add(BubbleArray(Counter))             'Places the items in teh
            Display box
            Next
        End Sub

        Sub BubbleWithMadeArray()
            Dim ArrayLength As Integer = NewArray.Length - 1         'Has to be -1 here
            otherwise out of bound

```

```

    Dim Temp As Integer
'^^^^ was due to coputer starting at 0 not 1
    Dim Swapp As Boolean
    Do
        Swapp = False
        For Counter = 0 To ArrayLength - 1
            If NewArray(Counter) > NewArray(Counter + 1) Then
'same sorting as before
                Temp = NewArray(Counter)
is adaped for the new array
                NewArray(Counter) = NewArray(Counter + 1)
                NewArray(Counter + 1) = Temp
                Swapp = True
            End If
        Next
    Loop Until Swapp = False
    For Counter = 0 To (NewArray.Length - 1)
        DbBubSor.Items.Add(NewArray(Counter))
    Next
End Sub

Private Sub BtnBubSor_Click(sender As Object, e As EventArgs)
Handles BtnBubSor.Click
    DbBubSor.Items.Clear()
    If CbUseNo.Checked = True Then
        BubbleWithMadeArray()
    Else
        BubbleWithSetArray()
    End If
End Sub

Sub StepWithPreset()
    Dim ArrayLength As Integer = BubbleArray.Length - 1
    Dim Temp As Integer
Sort
    For Counter = 0 To ArrayLength - 1
press
    If BubbleArray(Counter) > BubbleArray(Counter + 1) Then
insread of
        Temp = BubbleArray(Counter)
loop)
        BubbleArray(Counter) = BubbleArray(Counter + 1)
        BubbleArray(Counter + 1) = Temp
    End If
    Next
    DbBubSor.Items.Clear()
    For Counter = 0 To (BubbleArray.Length - 1)
        DbBubSor.Items.Add(BubbleArray(Counter))
    Next
End Sub

Sub StepWithChooosen()
    Dim ArrayLength As Integer = NewArray.Length - 1
otherwise out of bound
    Dim Temp As Integer
coputer starting at 0 not 1
    For Counter = 0 To ArrayLength - 1

```

This is a generic bubble sort. It first makes a pass of all the data comparing the next piece of data to the last, if the next item is smaller, it holds the item, swaps the current and then saves the held data back into the array. This is done until no more passes are made with the data.

'But

'If the use own numbers if checked  
'Changes the sorting mode from  
'set array to made array

'Generic Bubble

'Uses the button

'as a couter,

'A fixded (set

'Has to be -1 here

'^^^^ was due to

```

        If NewArray(Counter) > NewArray(Counter + 1) Then           'Same as before
            Temp = NewArray(Counter)                               'Uses stepper rather
            NewArray(Counter) = NewArray(Counter + 1)             'than Loop
            NewArray(Counter + 1) = Temp
        End If
    Next
    DbBubSor.Items.Clear()
    For Counter = 0 To (NewArray.Length - 1)
        DbBubSor.Items.Add(NewArray(Counter))
    Next
End Sub

Private Sub BtnBubStep_Click(sender As Object, e As EventArgs) Handles
BtnBubStep.Click
    DbBubSor.Items.Clear()
    If CbUseNo.Checked = True Then
        StepWithChosen()
    Else
        StepWithPreset()
    End If
End Sub

Private Sub BtnAdd_Click(sender As Object, e As EventArgs) Handles BtnAdd.Click
    If AC <= ChosenArrySiz Then
        If IsNumeric(TbIntAdd.Text) = True Then                   'Validation to ensure number
entered is a number
            NewArray(AC) = TbIntAdd.Text                          'sets the array postion to
the number
            TbIntAdd.Text = ""                                    'resets the textbox
            AC = AC + 1                                           'Increases counter
            DbBubSor.Items.Clear()
            For Ycounter = 0 To NewArray.Length - 1
                DbBubSor.Items.Add(NewArray(Ycounter))
            Next
        Else
            MsgBox("Please enter an integer")                       'Error message if validation
filed
        End If
    Else
        MsgBox("The Array is full")                               'If the array is too big error
message
    End If
End Sub

Private Sub NumericUpDown1_ValueChanged(sender As Object, e As EventArgs) Handles
NUDArraySize.ValueChanged
    If (NUDArraySize.Value) > 20 Then ' range check
        MsgBox("Be reasonable, you're not going to input over 20 numbers")
        NUDArraySize.Value = 1
    Else
        ChosenArrySiz = NUDArraySize.Value - 1                   'Changes the array size (for
array format hense the -1)
        ReDim NewArray(ChosenArrySiz)                            'Makes the new array big enough
to hold the new data
        AC = 0                                                    'Re sets the array couter
    End If
End Sub

```

```
End Sub
```

```
Private Sub CbUseNo_CheckedChanged(sender As Object, e As EventArgs) Handles  
CbUseNo.CheckedChanged  
    NUDArraySize.Visible = True           'Changes the display to visable to users  
    BtnAdd.Visible = True                 '^ and useabililty  
    TbIntAdd.Visible = True  
End Sub
```

```
Private Sub BtnRnFib_Click(sender As Object, e As EventArgs) Handles BtnRnFib.Click  
    Fibonattchi()  
End Sub
```

```
Sub Fibonattchi()  
    Dim Num As Integer = 0  
    Dim Num1 As Integer = 1  
    Dim Num2 As Integer = 0  
    For counter = 0 To 15  
        Num = Num1 + Num2 'Adds numbers  
        Num1 = Num2       'Sets first number as the last  
        Num2 = Num        'Sets second number as the generated  
        DbFib.Items.Add(Num) 'Adds the new number to the  
display box  
        Next  
    End Sub
```

This is the Fibonacci sequence. It works by adding two numbers together, an then continuing to add the product of the new number to the last, in doing so this replicates the sequence.

```
    Dim NumStep As Integer = 0  
    Dim Num1Step As Integer = 1  
    Dim Num2Step As Integer = 0  
    Private Sub BtnFibStep_Click(sender As Object, e As EventArgs) Handles  
BtnFibStep.Click  
        NumStep = Num1Step + Num2Step 'Works similar to before however  
        Num1Step = Num2Step           'This Does not have a counter  
        Num2Step = NumStep            'And only completes when the user presses the  
button  
        DbFib.Items.Add(NumStep)  
    End Sub
```

```
Dim SearchFor As String  
Dim StartNo As Integer  
Dim EndNo As Integer  
Dim LookingAt As Integer  
Dim Found As Boolean = False  
Dim ArrayNum As Integer  
Dim OverCounter As Integer  
Dim SetString() As String = {"Alex", "Dog", "Hat", "IceCube", "Legion", "LegionCube",  
"Penguin", "Zebra"} 'Set string to search  
Sub Setup() '-----Sets up the items to be found  
    SearchFor = TbBinary.Text 'selects the name of the item to be found  
    StartNo = 1 'Starting size of the array  
    EndNo = SetString.Length 'uses length of array to find the end size  
    Found = False 'Sets flag to false  
    OverCounter = 0  
    For Counter = 0 To SetString.Length - 1
```

```

        DbBinSearch.Items.Add((Counter + 1) & " " & SetString(Counter)) 'Adds all the
items in array to the display box
    Next
    DbBinSearch.Items.Add("-----") 'Graphical purposes -
used to help user
End Sub
Sub BinarySearch()
    Do While Found = False
        LookingAt = ((StartNo + EndNo) / 2) 'Selects the middle of
the array to search
        ArrayNum = LookingAt - 1 'Puts into array format (0 is the
first place)
        DbBinSearch.Items.Add("(" & StartNo & "+" & EndNo & ")") & "/"
2 = " & LookingAt) 'Display use for user
        If SetString(ArrayNum) = SearchFor Then
'IF found Set found to true to stop loop
            MsgBox((SetString(ArrayNum)) & " " & SearchFor)
            DbBinSearch.Items.Add(LookingAt & " = " &
SetString(LookingAt - 1)) 'Display
            DbBinSearch.Items.Add(SearchFor & " Found! ") 'Display
            Found = True
        ElseIf SetString(ArrayNum) < SearchFor Then
'IF not found, but selected is bigger
            DbBinSearch.Items.Add(LookingAt & " = " &
SetString(LookingAt - 1)) 'Display ^then change array size (cuts
of front)
            DbBinSearch.Items.Add(SearchFor & " Not yet Found ")
'Display
            StartNo = LookingAt + 1
            EndNo = EndNo
            OverCounter = OverCounter + 1
            DbBinSearch.Items.Add("list is now " & StartNo & " to " &
EndNo) 'Display

        Else ' If selected is smaller then change array size (cuts of end)
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt - 1))
'Display
            DbBinSearch.Items.Add(SearchFor & " Not yet Found ") 'Display
            StartNo = StartNo
            EndNo = LookingAt - 1
            OverCounter = OverCounter + 1
            DbBinSearch.Items.Add("list is now " & StartNo & " to " & EndNo) 'Display

        End If
        If OverCounter - 1 > Math.Log(SetString.Length, 2) Then 'Used as safety
catch
            Found = True 'Uses log 2 of
array size to ensure
            MsgBox("Not Found") 'The max number
of possible selects is not used
        End If
    Loop
End Sub

Private Sub BtnBinary_Click(sender As Object, e As EventArgs) Handles BtnBinary.Click
    BinarySearch()

```

The binary search works by taking the sorted data array, checking the middle value of the array to see if it matches the correct item being searched for. If the item looked at is lower than the one being found it shortens the array size by cutting off the end of the search parameter. If higher than expected it lowers the search parameter. Then looks at the middle value of the new search parameter again.

```

End Sub

Private Sub BtnStep_Click(sender As Object, e As EventArgs) Handles BtnStep.Click
    If Found = False Then
        LookingAt = ((StartNo + EndNo) / 2) 'Same as before
        ArrayNum = LookingAt - 1           'Uses a counter (set by button click)
rather than
        If SetString(ArrayNum) = SearchFor Then 'an incrementing loop
            MsgBox((SetString(ArrayNum)) & " " & SearchFor)
            Found = True
            DbBinSearch.Items.Add(LookingAt & " = " & SearchFor) 'Display
            DbBinSearch.Items.Add(SearchFor & " Found! ") 'Display
        ElseIf SetString(ArrayNum) < SearchFor Then
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt - 1))
'Display

            DbBinSearch.Items.Add(SearchFor & " Not yet Found ") 'Display
            StartNo = LookingAt + 1
            EndNo = EndNo
            OverCounter = OverCounter + 1
            DbBinSearch.Items.Add("list is now " & StartNo & " to " & EndNo) 'Display
        Else
'Display
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt - 1))

            DbBinSearch.Items.Add(SearchFor & " Not yet Found ") 'Display2
            StartNo = StartNo
            EndNo = LookingAt - 1
            OverCounter = OverCounter + 1
            DbBinSearch.Items.Add("list is now " & StartNo & " to " & EndNo) 'Display
        End If
        If OverCounter - 1 > Math.Log(SetString.Length, 2) Then
            Found = True
            MsgBox("Not Found")
        End If
    End If
End Sub

Private Sub BtnSLoad_Click(sender As Object, e As EventArgs) Handles BtnSLoad.Click
    If TbBinary.Text = "" Then 'Presence check
        MsgBox("Please enter an item to be searched?")
    Else
        DbBinSearch.Items.Clear() 'Clears DisplayBox
        Setup() 'Runs Setup
        BtnBinary.Visible = True
        BtnStep.Visible = True
    End If
End Sub

Private Sub BtnFrac_Click(sender As Object, e As EventArgs) Handles BtnFrac.Click
    number = NuDFracTo.Value 'Takes number from up down box
    Fractoral() 'runs fractional code
    MsgBox(hold) 'Shows user the end number
End Sub

```

This is the factorial algorithm, it multiplies the number it is given by the number -1, and that by the number -2 until it reaches one. Then the system stops calling its self.

This sub calls its self if the number started is greater than 1, if not it terminates. Then subtracts one from the number, multiplies by the last, then calls itself again.

```

.....
'Fracorial! code
Dim hold As Integer = 1
Dim number As Integer
Sub Fractoral()
    If number = 1 Then
        MsgBox(hold) 'ends/terminates reccursion
    Else
        DbFac.Items.Add(hold) 'Adds numeber to display box
        DbFac.Items.Add(number)
        DbFac.Items.Add(hold & " * " & number & " = " & hold * number) 'Shows user
equation
        hold = hold * number
        number = number - 1 'lowers the number by one
        Call Fractoral() 'calls its self to do the remaining numbers
    End If
End Su

Private Sub BubbleSort_Load(sender As Object, e As EventArgs) Handles MyBase.Load
'upon the page loading
    AdminPage.ReadProgramFile() 'Reads the proam file
    For Counter = 1 To AdminPage.RecNo
        If CbProgramName.Items.Contains(AdminPage.ProgramLocation(Counter).Name) =
False Then 'if the name of a progam hasnt
            CbProgramName.Items.Add(AdminPage.ProgramLocation(Counter).Name)
'already been added to the combo box
        End If
    Next
'it gets added
    Next

    For Counter = 0 To SetString.Length - 1
        DbBinSearch.Items.Add((Counter + 1) & " " & SetString(Counter))
    Next
    HideGB()
End Sub

Private Sub BtnLoadPro_Click(sender As Object, e As EventArgs) Handles
BtnLoadPro.Click
    Dim returnValue

```

```

        returnValue = Process.Start(LocalPass)    'Loads the program up
    End Sub
    Dim LocalPass As String    'Allows passing of the name
    Private Sub CbProgramName_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles CbProgramName.SelectedIndexChanged
        For counter = 1 To AdminPage.RecNo
            If CbProgramName.SelectedItem = AdminPage.ProgramLocation(counter).Name Then
'finds the correct recordf
                TbDesc.Text = AdminPage.ProgramLocation(counter).Description
'loads up the corresponding description
                LocalPass = AdminPage.ProgramLocation(counter).Location
'and location
            End If
        Next
    End Sub

```

```

Sub HideGB()    'Hides all groupboxes
    GbBinary.Hide()
    GbBubble.Hide()
    GbFib.Hide()
    GbFrac.Hide()
    GbLoad.Hide()
End Sub

```

```

Private Sub BtnShoBub_Click(sender As Object, e As EventArgs)
Handles BtnShoBub.Click
    HideGB()    'Shows the groupBox thars selected
    GbBubble.Show()
End Sub

```

```

Private Sub BtnBinShow_Click(sender As Object, e As EventArgs) Handles
BtnBinShow.Click
    HideGB()
    GbBinary.Show()
End Sub

```

```

Private Sub Fibonacci_Click(sender As Object, e As EventArgs) Handles Fibonacci.Click
    HideGB()
    GbFib.Show()
End Sub

```

```

Private Sub BtnFracShow_Click(sender As Object, e As EventArgs) Handles
BtnFracShow.Click
    HideGB()
    GbFrac.Show()
End Sub

```

```

Private Sub BtnLoadShow_Click(sender As Object, e As EventArgs) Handles
BtnLoadShow.Click
    HideGB()
    GbLoad.Show()
End Sub
End Class

```

Used for graphical effects, hides parts that aren't needed at that time, shows the parts that's needed when the user selects the algorithm to load.

## Bot Page line –

```
Imports System.IO
Public Class BotPage
    Dim PathConnect As Boolean
    Dim BotInfo As String
    Dim Stats As String
    Dim TranferFileName As String
    Public PassoverArray() As String

    Sub Display()
        Dim counter As Integer = 0
        Dim lineNo = File.ReadAllLines("CUBE.txt").Length 'find the number of lines used
in total
        DbBotpath.Items.Add("Total Number of lines" & lineNo)
        Dim Array() = File.ReadAllLines("CUBE.txt") 'saves the file as an
array
        For counter = 0 To (lineNo - 1) 'Subtracts one to keep in
range as it uses 0 as the first line
            DbBotpath.Items.Add(Array(counter))
        Next
    End Sub

    Private Sub BtnExtract_Click(sender As Object, e As EventArgs) Handles
BtnExtract.Click
        Dim counter As Integer = 0
        PathConnect = False
        Dim ofd As New OpenFileDialog 'This part opens a file view and filers
for text files
        ofd.FileName = ""
        ofd.Filter = "|*.txt"
        ofd.Title = "Select memory card"
        ofd.ShowDialog()
        If ofd.FileName = "" Then 'Needed to stop
crashing if cancelled
            MsgBox("Please select a file!", MsgBoxStyle.Critical)
            BtnShowPath.Visible = False
        Else
            CbEraseCover.Visible = True
            BtnShowPath.Visible = True
            DbBotpath.Items.Clear()
            Dim lineNo = File.ReadAllLines(ofd.FileName).Length 'find the number of lines
used in total
            DbBotpath.Items.Add("Total Number of lines:" & lineNo)
            Dim Array() = File.ReadAllLines(ofd.FileName) 'saves the file as
an array
            ReDim PassoverArray(lineNo - 1)
            For counter = 0 To (lineNo - 1) 'Subtracts one to
keep in range as it uses 0 as the first line
                DbBotpath.Items.Add(Array(counter))
                PassoverArray(counter) = Array(counter)
            Next
            TranferFileName = ofd.FileName

        End If
    End Sub
End Class
```

```
End Sub
```

```
Private Sub BtnErase_Click(sender As Object, e As EventArgs) Handles BtnErase.Click  
    Dim style = MsgBoxStyle.YesNo 'Message to ensure user wants to erase  
the file.
```

```
    Dim Result As Integer
```

```
    Result = MsgBox("Are you sure? ", style) 'Y/N are you Sure
```

```
    If Result = 6 Then
```

```
        File.WriteAllText(TranferFileName, "") ' writes nothing into the file
```

```
        MsgBox("The Data has been erased") 'ersading any data in it.
```

```
    End If
```

```
End Sub
```

```
Private Sub CbEraseCover_CheckedChanged(sender As Object, e As EventArgs) Handles  
CbEraseCover.CheckedChanged
```

```
    BtnErase.Visible = True
```

```
    If CbEraseCover.Checked = False Then
```

```
        BtnErase.Visible = False
```

```
    End If
```

```
End Sub
```

```
'.....  
.....
```

```
'All needed for drawing the path that the bot follows
```

```
Dim G As System.Drawing.Graphics
```

```

Dim NESW As Integer = 2
Dim StartX As Integer = 250
Dim StartY As Integer = 250
Dim FinishX As Integer = 250
coordinates
Dim FinishY As Integer = 250
Dim A As New Point(StartX, StartY)
Dim B As New Point(FinishX, FinishY)
Dim Rturn As Integer = 0
turns
Dim Lturn As Integer = 0
Dim uniCounter As Integer = 0
Dim HoldTime As Integer
Dim OldTime As Integer
Dim HoldCount As Integer

Public Sub Line()
    For uniCounter = 0 To PassoverArray.Length - 1
        If ((uniCounter + 1) Mod 3) <> 0 Then
            time then does below
            G = Panel1.CreateGraphics()
            G.DrawLine(Pens.Black, A, B)
            StartX = FinishX
            StartY = FinishY
            NESWCheck()
            stat of new one
            Else
                OldTime = HoldTime
                HoldTime = Mid(PassoverArray(uniCounter), 17,
                    Len(PassoverArray(uniCounter))) / 250
                ChangePoints()
                A = New Point(StartX, StartY)
                B = New Point(FinishX, FinishY)
                coordinates
                G.DrawLine(Pens.Blue, A, B)
            End If
        Next
    End Sub

Sub NESWCheck() 'Comapas system, works via numeric representation
    Dim HoldForTurn As Integer
    HoldForTurn = Mid(PassoverArray(uniCounter), 4,
        Len(PassoverArray(uniCounter)))
    If HoldForTurn > Rturn Then
        NESW = NESW + 1
        Rturn = HoldForTurn
    ElseIf HoldForTurn > Lturn Then
        NESW = NESW - 1
        Lturn = HoldForTurn
    End If
    If NESW > 4 Then
        NESW = 1
    ElseIf NESW < 1 Then
        NESW = 4
    End If
End Sub

```

Sets the starting position to the middle of the 500\*500 display area

Reads the bot data from the array.

If the third line (one saying the time), it draws the line of the last coordinates in black, and sets the last coordinate positions to the new ones.

If not the third line then the program checks the left or right position of the bot, changes the direction it should be facing, and then draws a new line in blue.

```

End Sub

Sub ChangePoints() 'Changes the coordinates of the lines
    If NESW = 1 Then
        FinishY = FinishY - (HoldTime - OldTime) 'Changes the
coordinate values ← 'X and Y
    ElseIf NESW = 2 Then
        FinishX = FinishX + (HoldTime - OldTime) 'Based of
the direction of the bot
    ElseIf NESW = 3 Then
        FinishY = FinishY + (HoldTime - OldTime)
    ElseIf NESW = 4 Then
        FinishX = FinishX - (HoldTime - OldTime)
    End If
End Sub

Private Sub BtnShowPath_Click(sender As Object, e As EventArgs) Handles
BtnShowPath.Click
    Line()
End Sub

End Class

```

Changes the coordinates for the line of the bot to be drawn. Which coordinate is changed and by what value is based off the direction of the bot.

### Bot Program mode –

```

Imports System.IO
Public Class BotProgram
    Dim Nesw As Integer = 1 'Acts as a compas
    Dim Size As Integer = -1 'Fist additon sets to 0 (array size 1)
    Dim Program() As String

    Private Sub BtnUp_Click(sender As Object, e As EventArgs) Handles
BtnUp.Click
        Size += 1 'Makes array bigger
        ReDim Preserve Program(Size) 'keeps old aray while
increasing size
        Program(Size) = "Up" 'Write in the correct
instructuion
        If Nesw = 1 Then
            Me.Bot.Top -= 20
        ElseIf Nesw = 2 Then ← 'uses directon bot is facing
            Me.Bot.Left += 20 'moves the virtual bot correcty
        ElseIf Nesw = 3 Then
            Me.Bot.Top += 20
        ElseIf Nesw = 4 Then
            Me.Bot.Left -= 20
        End If
    End Sub 'Gives Movement to Btnn as needed based off Direction (Nesw)
    Private Sub BtnDown_Click(sender As Object, e As EventArgs) Handles BtnDown.Click
        Size += 1
        ReDim Preserve Program(Size)
        Program(Size) = "Down" 'Similar to before - add the instructon
        If Nesw = 1 Then 'that corrsponds to the bots
            Me.Bot.Top += 20
        End If
    End Sub
End Class

```

Used to move the on screen bot in a direction.  
The direction the bot is moved in is dependant on its NESW position.

```

ElseIf Nesw = 2 Then
    Me.Bot.Left -= 20      'Changes location
ElseIf Nesw = 3 Then
    Me.Bot.Top -= 20
ElseIf Nesw = 4 Then
    Me.Bot.Left += 20
End If
End Sub 'Gives Movement to Btn in reverse

Private Sub BtnLturn_Click(sender As Object, e As EventArgs) Handles BtnLturn.Click
    Size += 1
    ReDim Preserve Program(Size)
    Program(Size) = "Left"
    Nesw = Nesw - 1 'Changes Direction of the bot
    NeswBtn()      'Changes colour of the compas to highlight directon.
End Sub 'Changes dication of Nesw
Private Sub BtnRturn_Click(sender As Object, e As EventArgs) Handles BtnRturn.Click
    Size += 1
    ReDim Preserve Program(Size)
    Program(Size) = "Right"
    Nesw = Nesw + 1
    NeswBtn()
End Sub

Sub NeswBtn()
    If Nesw > 4 Then      'Stops Nesw from being over 4
        Nesw = 1        'Loops it back to 1
    ElseIf Nesw < 1 Then 'Vice versa
        Nesw = 4
    End If
    CompasVisual()
End Sub 'sets Nesw
Sub CompasVisual()
    If Nesw = 1 Then
        BtnN.BackColor = Color.Green
        BtnE.BackColor = Color.Gray
        BtnS.BackColor = Color.Gray
        BtnW.BackColor = Color.Gray
    ElseIf Nesw = 2 Then      'If the direction is (this)
        BtnN.BackColor = Color.Gray 'Change colour of button (this) green
        BtnE.BackColor = Color.Green 'change coulour of (not this) green
        BtnS.BackColor = Color.Gray
        BtnW.BackColor = Color.Gray
    ElseIf Nesw = 3 Then
        BtnN.BackColor = Color.Gray
        BtnE.BackColor = Color.Gray
        BtnS.BackColor = Color.Green
        BtnW.BackColor = Color.Gray
    ElseIf Nesw = 4 Then
        BtnN.BackColor = Color.Gray
        BtnE.BackColor = Color.Gray
        BtnS.BackColor = Color.Gray
        BtnW.BackColor = Color.Green
    End If
End Sub 'Changes colour of compas

Private Sub BotProgram_Load(sender As Object, e As EventArgs) Handles MyBase.Load

```

```

        NeswBtn()
    End Sub 'Upon Loading Sets compas Colour

    Private Sub BtnSave_Click(sender As Object, e As EventArgs) Handles BtnSave.Click
        Dim counter As Integer = 0
        Dim ofd As New OpenFileDialog          'This part opens a file view and filters
for text files
        ofd.FileName = ""
        ofd.Filter = "|*.txt"
        ofd.Title = "Select memory card"
        ofd.ShowDialog()
        If ofd.FileName = "" Then              'Needed to stop
crashing if cancelled
            MsgBox("Please select a file!", MsgBoxStyle.Critical) 'Sends
        Else
            DbShowEnd.Visible = True
            For counter = 0 To Size            'Goes through the array one by
one
                DbShowEnd.Items.Add(Program(counter)) 'adds to item to the display box
line by line
            Next
            File.WriteAllText(ofd.FileName, "") 'Erases anything in the file
            Dim SW As StreamWriter
            SW = New StreamWriter(ofd.FileName)
            For counter = 0 To Size
                SW.WriteLine(Program(counter)) 'Writes in the array to the file.
            Next
            SW.Close()
            MsgBox("Your Program Has Been Saved") 'Shows the user the program has been
saved
        End If
    End Sub

End Class

```

### Bot Room Calculation-

```

Imports System.IO
Public Class BotRoom
    Public PassoverArray() As String 'DELETE IF NOT USED

    Dim LineNoDiv3 As Integer
    Structure RLT
        Dim Right As Integer
        Dim Left As Integer
        Dim Times As Integer
    End Structure
    Dim BotDataArray() As RLT

    Private Sub BtnExtract_Click(sender As Object, e As EventArgs) Handles
BtnExtract.Click
        Dim counter As Integer = 0
        Dim ofd As New OpenFileDialog          'This part opens a file view and filters
for text files
        ofd.FileName = ""
        ofd.Filter = "|*.txt"
        ofd.Title = "Select memory card"

```

```

ofd.ShowDialog()
If ofd.FileName = "" Then                                     'Needed to stop
crashing if cancelled
    MsgBox("Please select a file!", MsgBoxStyle.Critical)
    BtnShowPath.Visible = False
Else
    CheckBox1.Visible = True
    BtnShowPath.Visible = True
    ListBox1.Items.Clear()
    Dim lineNo = File.ReadAllLines(ofd.FileName).Length 'find the
number of lines used in total
    ListBox1.Items.Add("Total Number of lines:" & lineNo)
    Dim Array() = File.ReadAllLines(ofd.FileName)
'saves the file as an array (one line is one part of the array)
    LineNoDiv3 = lineNo / 3
    ReDim BotDataArray((lineNo / 3) - 1)
'Calculate num of records needed and then makes structured array
    For counter = 0 To ((lineNo / 3) - 1)
'Subtracts one to keep in range as it uses 0 as the first line
        BotDataArray(counter).Right = Mid(Array(counter * 3), 4,
Len(Array(counter * 3))) '*3 to keep up
        BotDataArray(counter).Left = Mid(Array(counter * 3 + 1),
4, Len(Array(counter * 3 + 1)))
        BotDataArray(counter).Times = Mid(Array(counter * 3 + 2),
17, Len((Array(counter * 3 + 2)))) 'Just collects raw time
        ListBox1.Items.Add(BotDataArray(counter).Right)
        ListBox1.Items.Add(BotDataArray(counter).Left)
        ListBox1.Items.Add(BotDataArray(counter).Times)
    Next
End If
End Sub

Private Sub BtnShowPath_Click(sender As Object, e As EventArgs) Handles
BtnShowPath.Click
    Line()
End Sub

```

Strips the bots data down to the raw components (removes any unassay data, just holds the number of R and L turns and the time) This data is stored into an array.

```

'...../-----
-----

Dim G As System.Drawing.Graphics
Dim NESW As Integer = 2
Dim StartX As Integer = 250
Dim StartY As Integer = 250
Dim FinishX As Integer = 250                                     'Holds values to make coordinates
Dim FinishY As Integer = 250
Dim A As New Point(StartX, StartY)                               'Makes the coordinates
Dim B As New Point(FinishX, FinishY)
Dim Rturn As Integer = 0                                         'Counters holding the turns
Dim Lturn As Integer = 0
Dim uniCounter As Integer = 0
Dim HoldTime As Integer
Dim OldTime As Integer
Dim HoldCount As Integer

Public Sub Line()
    Dim LLast As Integer = 0

```

```

Dim RLast As Integer = 0
G = Panel1.CreateGraphics()
For uniCounter = 0 To LineNoDiv3 - 1
    StartX = FinishX 'changes end of previous line to start of new one
    StartY = FinishY 'Making B the new A
    'Change start point to end point
    OldTime = HoldTime
    HoldTime = BotdataArray(uniCounter).Times
    RLast = BotdataArray(uniCounter).Right
    If HoldTime - OldTime > 2999 Then
        NESW = NESW - 1 'Turn
        KeepInRange()
        ChangePoints() 'Set new coordinates
        DrawLine() 'Up to show movement straight after
    Else
        NESW = NESW + 1
        KeepInRange()
        A = New Point(StartX, StartY)
        ChangePointsOfReverse()
        G.DrawLine(Pens.Blue, A, B)
        ChangePoints()
        DrawLine()
    End If
Next
End Sub

Dim RLast As Boolean = False

Sub DrawLine()
    A = New Point(StartX, StartY)
    B = New Point(FinishX, FinishY)
    G.DrawLine(Pens.Black, A, B)
End Sub

Sub ChangePoints() 'Changes the coordinates of the lines
    If NESW = 1 Then
        FinishY = FinishY - (HoldTime - OldTime) / 250 'Changes the
coordinate
    ElseIf NESW = 2 Then
        FinishX = FinishX + (HoldTime - OldTime) / 250 'X and Y values
'Based of the
direction of the bot
    ElseIf NESW = 3 Then
        FinishY = FinishY + (HoldTime - OldTime) / 250
    ElseIf NESW = 4 Then
        FinishX = FinishX - (HoldTime - OldTime) / 250
    End If
End Sub

Sub ChangePointsOfReverse() 'Changes the coordinates of the lines
    If NESW = 3 Then
        B = New Point(FinishX, FinishY - 10)
    ElseIf NESW = 4 Then
        B = New Point(FinishX + 10, FinishY)
    ElseIf NESW = 1 Then
        FinishY = FinishY + (HoldTime - OldTime) / 250
        B = New Point(FinishX, FinishY + 10)
    ElseIf NESW = 2 Then

```

```
        B = New Point(FinishX - 10, FinishY)
    End If
End Sub

Dim GuideLineX As Integer
Dim GuideLineY As Integer
Dim ForLeft As Integer = NESW - 1

Sub KeepInRange()
    If NESW > 4 Then
        NESW = 1
    ElseIf NESW < 1 Then
        NESW = 4
    End If
End Sub
End Class
```

## Bot Code (Arduino script)-

```
Finished_Bot_code
#include <SPI.h> //Allows serial connection
#include <SD.h> //Allows SD writing and reading
File myFile;
unsigned long previousMillis = 0;
unsigned long ToPrint = 0;

const int M1a = 4; //Motor A positive
const int M1b = 5; //Motor A negative
const int M2a = 6;
const int M2b = 7;

const int MicroSwRPin = 2;
int MicroSwRCounter = 0; //Num of times sw pressed
int MicroSwRState = 0; //Closed or Open (on or off)
int LastMicroSwRState = 0; // What it was before

const int MicroSwLPin = 8;
int MicroSwLCounter = 0; //Num of times sw pressed
int MicroSwLState = 0; //Closed or Open (on or off)
int LastMicroSwLState = 0; // What it was before

int Mode = 0; //varibale for mode selection

//-----
```

```

void setup() {
  pinMode(A1, OUTPUT);
  Buzz();
  pinMode(MicroSwRPin, INPUT); //MicroSwitches
  pinMode(MicroSwLPin, INPUT);
  pinMode(M1a, OUTPUT); // Motor A
  pinMode(M1b, OUTPUT);
  pinMode(9, OUTPUT); //PWM control for A
  pinMode(M2a, OUTPUT); // Motor B
  pinMode(M2b, OUTPUT);
  pinMode(3, OUTPUT); //PWM control for B

  Serial.begin(9600); //Allows the board to write to the screen
  while (!Serial) {
    ; // waits for the board to talk to the computer - to ensure that both a ready
  }
  Serial.print("Regstering SD card...");
  if (!SD.begin(10)) { //Uses imported SD function (starts form pin 10)
    Serial.println("Registration Error");
    return;
  }
  Serial.println("Done.");
  Mode = 0; //Stats the program on Room Calc mode
  ModeSELECT(); //Runs mode selection so allows user to change the mode if they wish
}

void ModeSELECT(){
  MicroSwRState = digitalRead(MicroSwRPin); //Reads in the current value on the switches
  MicroSwLState = digitalRead(MicroSwLPin);
  if (MicroSwRState == LOW) {
    Serial.println("Mode select = 1"); //Changes mode based on the input the user gives
    Mode = 1;
    Buzz();
  }
  else if (MicroSwLState == LOW) {
    Serial.println("Mode select = 2");
    Mode = 2;
    Buzz();
    delay(500);
    Buzz(); //Buzzes to indicate mode (audio output)
  }
}

//-----

void loop() {
  if (Mode == 1) {CollisionMode();} //checks mode
  else if (Mode == 2) {CompasMode();} //Runs 'mode'selected
  else if (Mode == 0){RoomCalcMode();}
}

```

```

void RoomCalcMode() {
  Serial.println("Room mode begins");
  unsigned long currentMillis = millis(); //set time to correct time (that the arduino uses
  MicroSwRState = digitalRead(MicroSwRPin); // Reads switch value

  if (MicroSwRState != LastMicroSwRState) { //Checks with the last state
    if (MicroSwRState == LOW) {
      MicroSwRCounter++; //If changed adds to the counter
      Serial.print("Switch Hit"); //If hooked up to computer users can see the 'thought'
      previousMillis = currentMillis; //process of the bot
      ToPrint = previousMillis;
      ReadIntoFile();
    }
    MoveReverse();
    delay(1000); //Delay to allow the bot to move and turn
    MoveSharpRight(); //Turns right to next part
    delay(400);
  }
  delay(50);
  MoveForward();
}
LastMicroSwRState = MicroSwRState; //saves last state for loop, as to let the bot know when a change has been made

if (currentMillis - previousMillis > 3000){//if its been over 'three' seconds
  MicroSwLCounter++;
  previousMillis = currentMillis;
  ToPrint = previousMillis; //Stores the time value of the bot to be saved to the SD
  ReadIntoFile(); //Saves the left turn into the file
  Serial.print("Left turn");
  MoveSharpLeft(); //turn left
  delay (400);
  MoveForward(); //move forward
  Serial.println("Forward (towards wall)");
}
LastMicroSwRState = MicroSwRState;
}
}

```

```

void CollisionMode() {
  Serial.println("Collision mode begins");
  unsigned long currentMillis = millis(); //Find the value of the current time
  MicroSwRState = digitalRead(MicroSwRPin);
  MicroSwLState = digitalRead(MicroSwLPin); //Reads the Switch values

  if (MicroSwRState != LastMicroSwRState) { //Checks with the last state
    if (MicroSwRState == LOW) { //If changed adds to the counter
      MicroSwRCounter++; //Adds one to counter if switch is changed
      Serial.println("on");
      Serial.print("Number of SW on: ");
      Serial.println(MicroSwRCounter); //Prints the value to the serial port for user to see
      previousMillis = currentMillis; //Sets the current time as the old time
      ToPrint = previousMillis;
      ReadIntoFile(); //saves to the SD card
    }
    MoveReverse(); //moves back
    delay(1000);
    MoveSharpRight(); //Turns right
    delay(300);
  }
  delay(50); //Used to stop the input being overloaded
}
LastMicroSwRState = MicroSwRState; //saves last state for loop
if (MicroSwLState != LastMicroSwLState) { //Checks with the last state
  if (MicroSwLState == LOW) { //If changed adds to the counter
    MicroSwLCounter++;
    Serial.println("on");
    Serial.print("Number of SW on: ");
    Serial.println(MicroSwLCounter);
    previousMillis = currentMillis; //Same as before but
    ToPrint = previousMillis; // for the left bumper
    ReadIntoFile(); //saves into file
    MoveReverse();
    delay(1000);
    MoveSharpLeft(); //Movement
    delay(300);
  }
}
}

```

```

if (MicroSwLState != LastMicroSwLState) { //Checks with the last state
  if (MicroSwLState == LOW) { //If changed adds to the counter
    MicroSwLCounter++;
    Serial.println("on");
    Serial.print("Number of SW on: ");
    Serial.println(MicroSwLCounter);
    previousMillis = currentMillis; //Same as before but
    ToPrint = previousMillis; // for the left bumper
    ReadIntoFile(); //saves into file
    MoveReverse();
    delay(1000);
    MoveSharpLeft(); //Movement
    delay(300);
  }
  delay(50);
}
LastMicroSwLState = MicroSwLState; //saves last state for loop
MoveForward(); //Ensures bot is always moving forward of not doing anything else.
}

void CompasMode(){
  Serial.println("Compas mode begins"); //Message if serial port is connected
  myFile = SD.open("cube.txt"); //Opens the file saved in the SD card
  if (myFile) { //If found
    Serial.println("cube.txt:");
    while (myFile.available()) {
      String line = myFile.readStringUntil('\n'); //Read part file until the line is missing
      Serial.println(line); //Prints to user

      if (line.startsWith("Up")) {MoveForward(); delay (1000);} //If line read contains the correct instructuon
      else if (line.startsWith("Down")) {MoveReverse(); delay (1000);} //Loads the correcent movement for allocated time
      else if (line.startsWith("Right")) {MoveSharpLeft(); delay (300);}
      else if (line.startsWith("Left")) {MoveSharpRight(); delay (300);}
    }
    myFile.close();
  } else {
    Serial.println("error opening File"); //If couldnt open, output a message for serial port.
  }
}

//-----
//File Reading and Writing
void ReadIntoFile(){
  myFile = SD.open("cube.txt", FILE_WRITE); //Opens file in SDcard
  if (myFile) {
    Serial.print("Saveing to cube.txt...");
    myFile.print("R: "); //saves information line by line
    myFile.println(MicroSwRCounter); //Inputs data
    myFile.print("L: ");
    myFile.println(MicroSwLCounter);
    myFile.print("At the time of: ");
    myFile.println(ToPrint);
    myFile.close();
    Serial.println("done.");
  } else {
    Serial.println("error opening File"); //If the sd card is not found outputs via serial port
  }
}
}

```

```

//-----
//The movent subs for the bot
//Forward
void MoveForwardC(){ //Both Forward
  digitalWrite (M1a, HIGH); //Motor A
  digitalWrite (M1b, LOW);
  analogWrite (9, 180);
  //
  digitalWrite (M2a, LOW); //Motor B
  digitalWrite (M2b, HIGH);
  analogWrite (3, 160);
}

//Reverse
void MoveReverseC(){ //Both wheels backwards
  digitalWrite (M1a, LOW); //Motor A
  digitalWrite (M1b, HIGH);
  analogWrite (9, 180);
  //
  digitalWrite (M2a, HIGH); //Motor B
  digitalWrite (M2b, LOW);
  analogWrite (3, 160);
}

//Stpp
//SharpLeft
void MoveSharpLeftC(){ //MA=R and MB=F
  digitalWrite (M1a, LOW); //Motor A
  digitalWrite (M1b, HIGH);
  analogWrite (9, 180);
  //
  digitalWrite (M2a, LOW); //Motor B
  digitalWrite (M2b, HIGH);
  analogWrite (3, 160);
}

//SharpRight
void MoveSharpRightC(){ //MA=F and MB=R
  digitalWrite (M1a, HIGH); //Motor A
  digitalWrite (M1b, LOW);
  analogWrite (9, 180);
  //
  digitalWrite (M2a, HIGH); //Motor B
  digitalWrite (M2b, LOW);
  analogWrite (3, 160);
}

void BuzzC(){
  digitalWrite(A1, HIGH); //Power to the buzzer for 1 second
  delay (1000);
  digitalWrite(A1, LOW);
}

```

## Example encryption code (Written in Python):

```
def en():
    Message = input('what is your message? ') #User inputs the message.
    key = input('what is your keyword? ') #User inputs the keyword.
    output=[] #This is where stages of the
    keyadd=[] #message will be saved while
    coded=[] #the program runs.
    Hold=""
    for char in Message:
        if ord(char)<97 and ord(char)>64: #Checks for capitals, if found
            char=chr(ord(char)+32) #program changes it into lowercase.
            output.append(ord(char)) #Saves message as ASCII
    for chard in key:
        if ord(chard)<97 and ord(chard)>64: #Checks for capitals like before.
            chard=chr(ord(chard)+32)
            keyadd.append(ord(chard)-96) #Takes 96 to make ASCII into 1,2,3

    keylen=len(Message)//len(key) #Makes the length of the key equal
    keyl=(keyadd*keylen)+keyadd #to the length of the message.
    count=0
    while count<len(Message):
        code=output[count]+keyl[count] #Adds the keyword to the message.
        coded.append(code)
        if coded[count]>122: #If new ASCII code is over 122(z)
            coded[count]=(coded[count]-26) #the program will bring it back to a
        if coded[count]<71: #If the number is too small to be a
            coded[count]=32 #character, it needs to be a space
        Hold=Hold+(chr(coded[count])) #Program concatenates the
        count=count+1 #finished code, then displays it
    print('Your coded message is ... =',Hold) #to the user.
def de():
    Message = input('what is the encrypted message? ')
    key = input('what was the keyword? ') #Inputs for the user.
    output=[]
    keytake=[] #Stores concatenated message.
    decoded=[]
    Hold=""
    for char in Message:
        output.append(ord(char)) #Converts message into ASCII.
    for chard in key:
        keytake.append(ord(chard)-96) #Changes key into alphabet number.(1-26)

    keylen=len(Message)//len(key) #Makes key length equal to message.
    keyl=(keytake*keylen)+keytake

    count=0
    while count<len(Message):
        code=output[count]-keyl[count] #Takes key away from message.
        decoded.append(code) #Concatenates message
        if decoded[count]<97: #If the new ASCII message is
            decoded[count]=(decoded[count]+26) #under 97(a), code brings it back to z.
        if decoded[count]<71:
            decoded[count]=32 #Program finds where a space should be.
        Hold=Hold+(chr(decoded[count]))
        count=count+1
    print('Your decoded message is ... =',Hold) #Displays message to user.
count=0
while count==0:
    taskLoop=input("Do you want to Encrypt or Decrypt? (E or D) \n\
(any other key to exit)\n")
    if taskLoop=='e' or taskLoop=='encrypt' or taskLoop=='E' or taskLoop=='Encrypt':
        en()
    elif taskLoop=='d' or taskLoop=='decrypt' or taskLoop=='D' or taskLoop=='Decrypt':
        de()
    else:
        print('Goodbye') #improved program loop, will allow the
        count=count+1 #user to encrypt or decrypt with one menu
```

Main Program Data Dictionary:

Variable	Type	Function	Used
SR	StreamReader	Used to read the data in a file	AdminPage SignIn
RecordNumber	integer	Holds the number of records kept in the file	AdminPage SignIn
Counter	Integer	A generic counter used to hold a number incrementing by a value	All apart menu page
LoggedIn	Boolean	Allows the system to keep track if a user is signed in or not	SignIn
UserIdNumber	Integer	Holds the record number of the user currently logged in	SignIn
SW	StreamWriter	Allows the writing into files	AdminPage SignUp
Proceed	Boolean	Flag to tell the program if to continue or not.	
Extra	String	Holds message that's sent to the user	
PicNum	Integer	Holds the number of which photo should be displayed.	WelcomePage
RecNo	Integer	Holds the record number for the program selected	
RCText	String	Holds the information to be sent in the email, this will keep the data in case of a mistake.	AdminPage
HolderID	String	Holds the information to be sent in the email, this will keep the data in case of a mistake	AdminPage
HolderEmial			
HolderFName			
HolderSName			
HolderDob	Object	Used for sending an email with outlook	TeacherPage AdminPage
HolderPass			
HolderLv			
objOutlookmsg			
objOutlook	Object	Used for sending an email with outlook	TeacherPage AdminPage

Ofd	OpenFileDialog	Allows complete reading and writing into a file.	AdminPage TeacherPage
Add	Integer	Counter used to increase by one if needed	AdminPage TeacherPage
Found	Boolean	Flag used to show if record has been found or not.	AdminPage TeacherPage AlgorithmPage
TestDataID	Integer	Used to hold the current test record ID	TeacherPage TestPage
TestAmount	Integer	A counter that's holds the question number of the test.	TestPage TeacgerPage
Allow	Boolean	Used as a flag to allow or disallow the procedure to continue	TeacherPage TestPage AlgorithmPage
Index	Integer	Holds the selected index of a combo box when changed	TeaccherPage AdminPage LessonPage
Style	Msgbox Style	Used to make a message box that gets y/n response from a user.	SetTests BotPage BotRoom
Result	integer	Used to make a message box that gets y/n response from a user.	SetTests BotPage BotRoom
SearchFor	String	Holds the search criterial	LearnPage AlgorithmPage
StartNo	Integer	Numbers used in the binary search, holds the stat and end of the new array	LearnPage AlgorithmPage
EndNo	Integer	Numbers used in the binary search, holds the stat and end of the new array	LearnPage AlgorithmPage
LookingAt	Integer	Flag for when found	LearnPage AlgorithmPage
ArrayNum	Integer		LearnPage AlgorithmPage
Found	Boolean		LearnPage AlgorithmPage SignIn
Address	String	Holds the email address of there the email will be sent to	LearnPage
SendTo	String	Holds the hidden address, to be switched over behind the scenes	LearnPage
Namelable	Label	Label that is controlled by code	TestPAGE
CheckBoxA	CheckBox	Are textbox's that are controlled with code.	TestPage

CheckBoxB	CheckBox	Are textbox's that are controlled with code.	
CheckBoxC			
CheckBoxD	Integer	Counter that increments with timer	TestPage
Tcounter			
Void	Boolean	Used as a flag for if the student score is to be accepted or not	TestPage
Score	Integer	The students score from the test	TestPage
Amount	Integer	Counter of how many boxes have been checked	TestPage
RandomClass	Random	Allows a random number to be generated	TestPage
RandomNumber	Integer	Holder for the random number	TestPage
Before	Integer	Used to set the location of the combo boxes	TestPage
First	Integer Point	Used to set the location of the combo boxes Coordinates that will randomize the location of the correct answer.	TestPage
Second			TestPage
RowOne			TestPage
RowTwo			TestPage
PointA			TestPage
PointB	Point Integer	Coordinates that will randomize the location of the correct answer. Array to be sorted	TestPage
PointC			TestPage
PointD			TestPage
BubbleArray()			AlgorithmPage
NewArray()	Integer	An array that the user generates	AlgorithmPage
ArrayCounter	Integer	Counter to check array does not go out of bounds	AlgorithmPage
ArrayLength	Integer	Holds the length of an array	AlgorithmPage
Temp	Integer	Holds a value to be swapped	AlgorithmPage
Sawpp	Boolean	Used as a flag for if a swap was made in the pass.	AlgorithmPage
Num	Integer	Numeric holders to be added for the Fibonacci sequence	AlgorithmPage
Num1	Integer Integer	Numeric holders to be added for the Fibonacci sequence Used to ensure the counter does not go over this calculated value	AlgorithmPage
Num2			AlgorithmPage
OverCoutner			AlgorithmPage
Hold	Integer	Holds the last value in the fractural calc	AlgorithmPage
Number	Integer	The number entered by the user	AlgorithmPage

LocalPass	String	Used to pass the location of the selected program.	AlgorithmPage Admin Page
PathConnect	Boolean	Used as flag for if a variable is correct or not	AlgorithmPage
LineNo	Integer	Used to hold number of lines from file	BotProgram BotPage BotRoom
G	System.Graphics	Past of VB's graphics system	BotProgram BotPage BotRoom
NESW	Integer	Numeric compass system	BotProgram BotPage BotRoom
StartX	Integer	Used to make the coordinates of the onscreen bot	BotProgram BotPage BotRoom
StartY	Integer Point	Used to make the coordinates of the onscreen bot Holds where the on screen bot moves to	BotProgram BotPage BotRoom
FinishX			BotProgram BotPage BotRoom
FinishY		Holds where the on screen bot moves to Starting coordinate	BotProgram BotPage BotRoom
A		BotProgram BotPage BotRoom	
B	Point	Finish coordinate a-b	BotProgram BotPage BotRoom
Rturn	Integer	Counter for right turns by the bot	BotPage
Lturn	Integer	Counter for left turns by the bot	BotPage
UniCounter	Integer	Counter used for multiple loops	BotProgram BotPage BotRoom
HoldTime	Integer	Hold where the bot was	BotPage BotRoom
OldTime	Integer	Holds the time the bot used to be	BotPage BotRoom
HoldForTurn	Integer	Used to see if the bot has changed direction since last checked	BotPage BotRoom
Size	Integer	Holds the size of the array	BotProgram BotPage

			BotRoom
Program()	String	Holds an array that's generated by the user	BotProgram
Key	string	Holds the key that's used to encrypted and decrypt the file	SignIn
IdNumberInt	Integer	The record number of the user.	SignIn
UserNameStr	String	The generated user name	SignIn
FNamrStr	String	Users First Name	SignIn
SNameStr	String	Second name	SignIn
DoBSrt	String	Users date of birth	SignIn
PasswordStr	String	Password of user	SignIn
AccesLvInt	Integer	Users access level	SignIn
TestComName	String	Name of the Set Test	SignIn
CompBy	Date	Date that it should be competed by	SignIn
FinishedSta	Boolean	Flag for if it has been completed or not	SignIn
Reccord	Integer	Record number	AdminPage AlgorithmPage
Name	String	Holds name of the program	AdminPage AlgorithmPage
Description	String	A small description of the program	AdminPage AlgorithmPage
Location	String	Holds the location of the program	AdminPage AlgorithmPage
TestIDno	Integer	Record number	TeacherPage
TestQno	Integer	Question number 1-5	TeacherPage
TestName	String	Holds the name of the test	TeacherPage
A	String	Holds fake answers to multi choice	TeacherPage
B	String		TeacherPage
C	String		TeacherPage
D	String	Holds the real answer	TeacherPage
TestNo	Integer	Record Number	TestPage TeacherPage
UserName	String	The students name who took the test	TestPage TeacherPage
Score	Integer	The score the student got on the test	TestPage TeacherPage
Time	Integer	The time they took to complete the test	TestPage TeacherPage
Mark	Integer	Output of a calculation with the score and time.	TestPage TeacherPage
Message	String	Holds the message to be encrypted or decrypted	Encryption Program
Key	string	Holds the key the user enters	Encryption Program

Hold	String	Where the final message is slowed added	Encryption Program
Count	Integer	Generic counter	Encryption Program

Bot Pins:

Pins Used	Labled	Use	Type
Pin 4	MotorA+	Power supply to motor A	Digital Output
Pin 5	MotorA-		Digital Output
Pin 6	MotorB+	Power supply to motor B	Digital Output
Pin 7	MotorB-		Digital Output
Pin 2	MicroSwRPin	Inputs for the switches	Digital Input
Pin 8	McroSwLPin		Digital Input
Pin A1	Buzzer	Buzzer Output	Digital Output
Pin 9	PWM OutA	Allows speed control of motor A	Analog Output
Pin 3	PWM OutB	Allows speed control of motor B	Analog Output
Pin 10	SD- Cs	Used with imported function to allow reading and writing with the Sd card	SD –Read and Write
Pin 11	SD- Di		
Pin 12	SD- Po		
Pin 13	SD- Clk		

Bot Code Data Dictionary:

Variable	Type	Function
previousMillis	Unsigned Long integer	Holds the 'old' time (in mili seconds)
MicroSwRCounter	Integer	Holds the Number of time the R switch has been activated
MicroSwRState	Integer	Gives the current state od the SW
LastMicroSwRState	Integer	Gives the last state (used to stop Sw being read as on more than once per hit)
MicroSwLCounter	Integer	Holds the Number of time the L switch has been activated
MicroSwLState	Integer	Gives the current state od the SW
LastMicroSwLState	Integer	Gives the last state (used to stop Sw being read as on more than once per hit)

Mode	Integer	Allows the user to select a mode for the bot
CurrentMillis	Unsigned long Integer	Read the time at that moment from the Arduino
ToPrint	Unsigned Long Integer	Holds the Time to later be saved into the file

### Final forms and fiished bot:

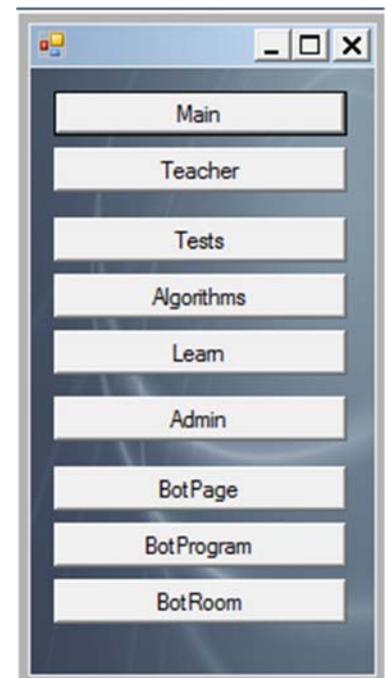


The Sign in form was designed to be simplistic. As a result, it only gives the user two options, sign in or sign up. This keeps the form as simple as possible allowing users to login or sign up without any hassle or problems

To grab the user's attention, the image on the main part of the screen changes periodically. This means that the users don't just see 'the same' screen everytime.

The imagery and background chosen was picked as the colours are 'relaxing' a blue tinted background has a subconsiuls effect to help relax people and make them think with more care, some studys have shown this. As my system is educational thinking with care is an important part of it.

As for the Quick menu, this small form helps user quickly navigate the system, its only purpose its to make navigation easer for the user. It accomplaces this by havinh all the buttons to all the different forms (apart from its self and the sign up from) in it. This gives user quick access to all the form from one location, vastily helping with navigation.



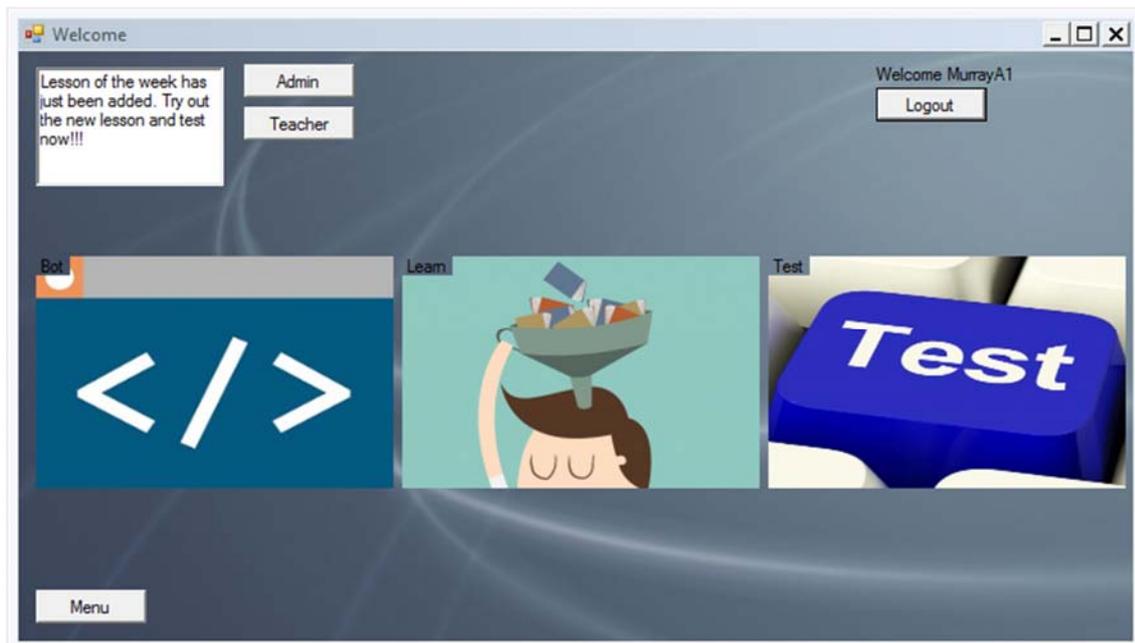
The Signup form can only be accessed by a button on the login form. This helps stop users from making more than one account by accident.

To make an account, it's designed to be self-explanatory with each field clearly labeled. All the fields needed to hold the user information are all on the page and can be entered by the user.

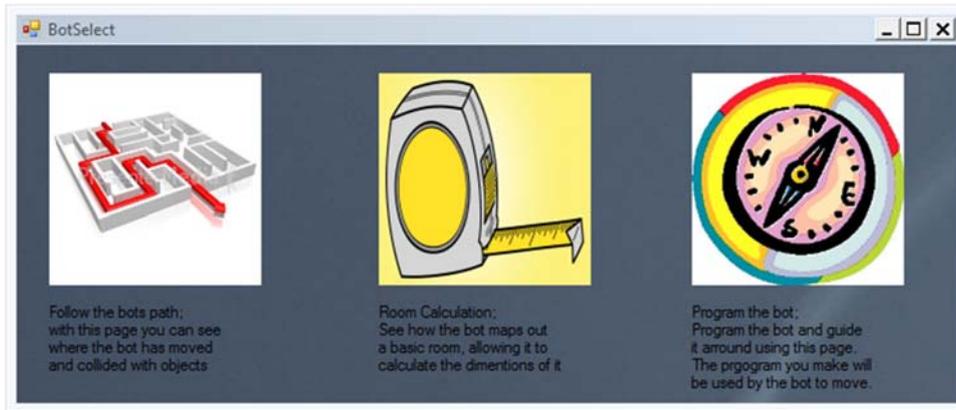
Double entry is required for the passwords to stop the user being locked out of the account the first time they join it.

Once the user has finished they press the clear sign up button.

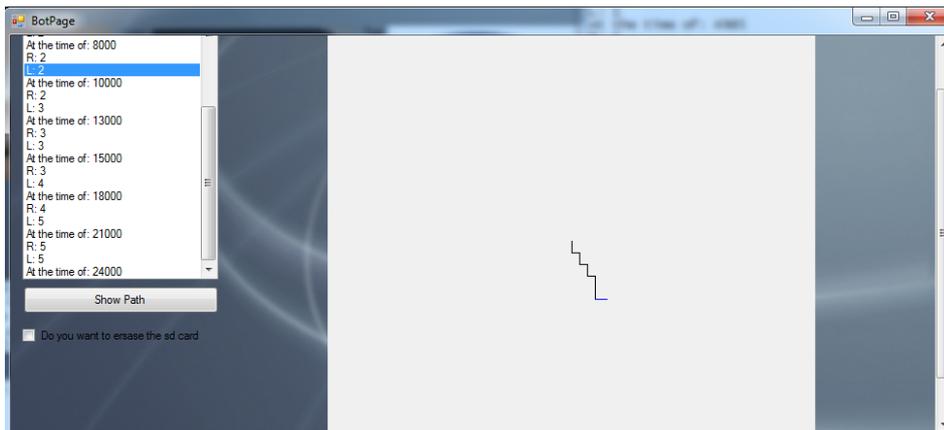
All the clear and self-explanatory fields make this form easy to use, even for a novice user.



The Main page (The welcome page) was also designed to be clear and simple for users to see and understand. It has all the needed links/features for the user to start the system. E.g. menu link, links to the main pages. Admins and teachers can access their special pages from this page too. For the student users, the page was kept with the professional 'blue' and "calm" tinted background. With pictures acting as the main buttons to keep the users interested, (as before these change on a timer).



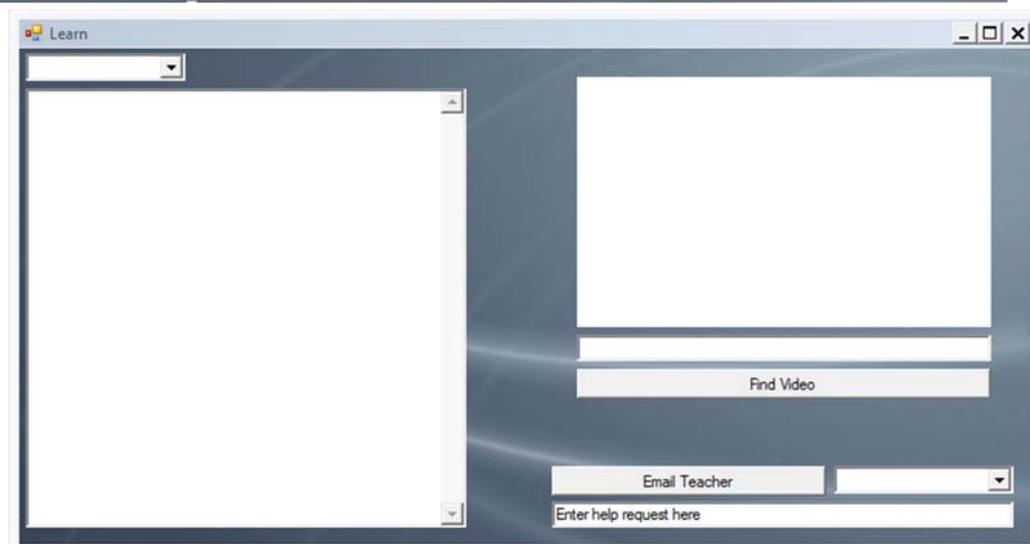
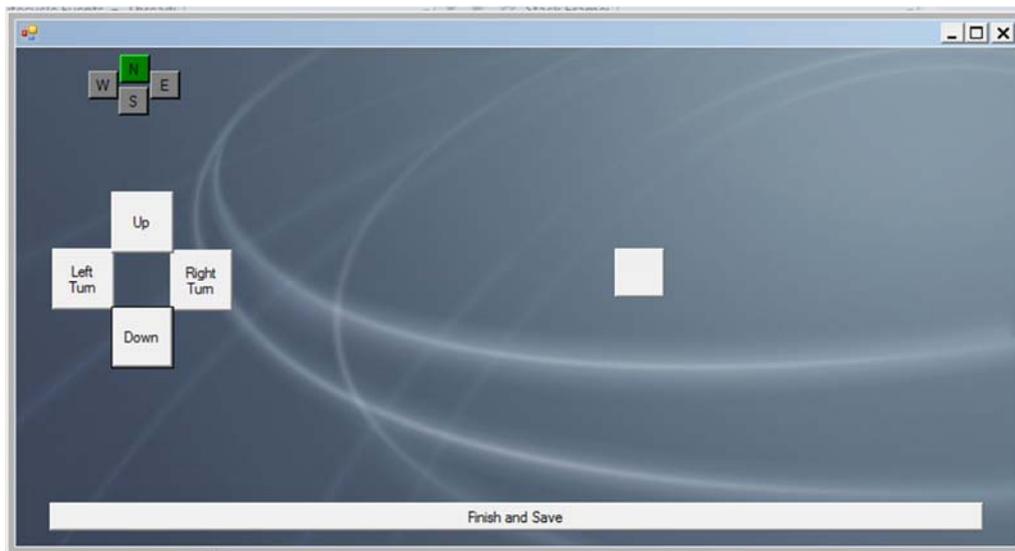
This is more of a sub menu for the bot's section of the program, designed to again help with navigation of the page, the picture buttons were used to make the program more appealing to the user. The buttons also have text below them to help the user decide which mode they want, the idea of this was given from a similar system I found on my investigation.



The bot's collision mode allowed data to be collected by the bot, this page converts the data given into a graphical display that can be shown to the user. The map/path shown to the user is done so clearly on a space (background and obstruction free) panel, this allows it to be seen by the user completely.

The user also has very clear buttons to get them started to begin with they can only extract the data, once extracted they can show the path, and once shown, they have the option to erase the data if they so wish. This subtly guides the user in how to use this page without the need for an actual guide.

The program mode for the bot is designed to be simple to use but also fun and entertraing. The compass at the top makes it very easy to see which way the virtual bot is pointing and what direction it will in, as the green colour of the direction clearly indicates this to the user. The use of buttons as a sort of D-pad design allows user to easily click the way they want the bot to move, as they are arranged in the D-pad formation this makes it easy for new user to grasp, as it's a common format that most people are accustom too ( due to vido games and smiilar to arrow keys.)



The learning page is where the sudent can take in the knoladge, I kept this page partial formal for the pulrey text base lessons, so the student would not get distracted, but I ensured to balance it out with entertraiment by adding the video feature to it, this ensures the students don't get bored. As the rest of the program the blue tinted background helps with both looks and learning.

The email feature was also used well on this form, as the student can just send off a message when they encounter a problem, they don't have to wait to load up another form.



The test page was kept simple to keep the students focused on just the test they were taking. At the top the timer shows the student how long they have taken for the test upon starting it. The student can use the dropbox to see the available tests, they then can select one easily by simply clicking on the list that appears. This makes is very easy for the students to set up a test.

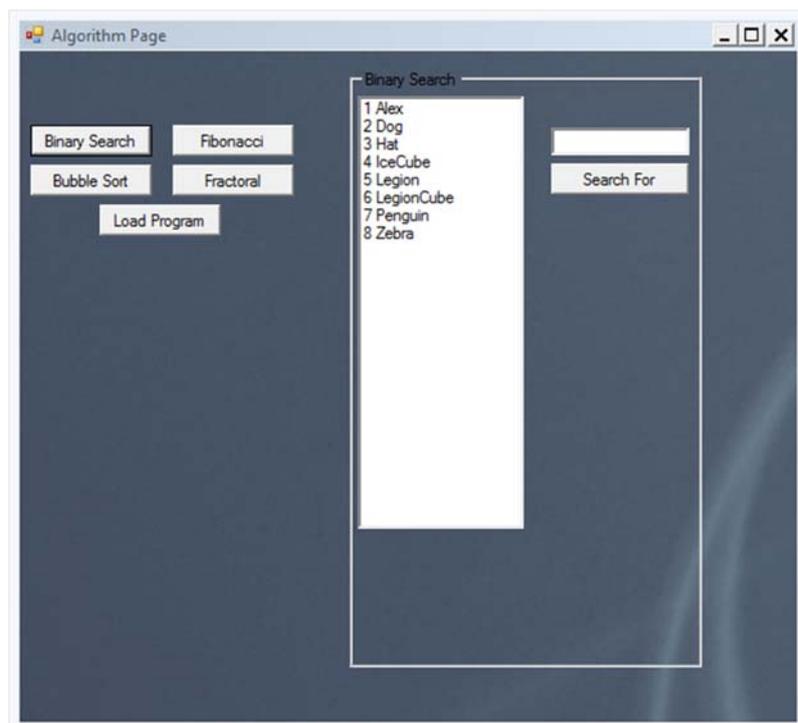
The questions and answers are then loaded up into the lables and check boxes, for the student to see (but this is randomised for the correct answer). This allows the user to just read and click, it even allows for changing an answer before they mark it if they made a mistake.

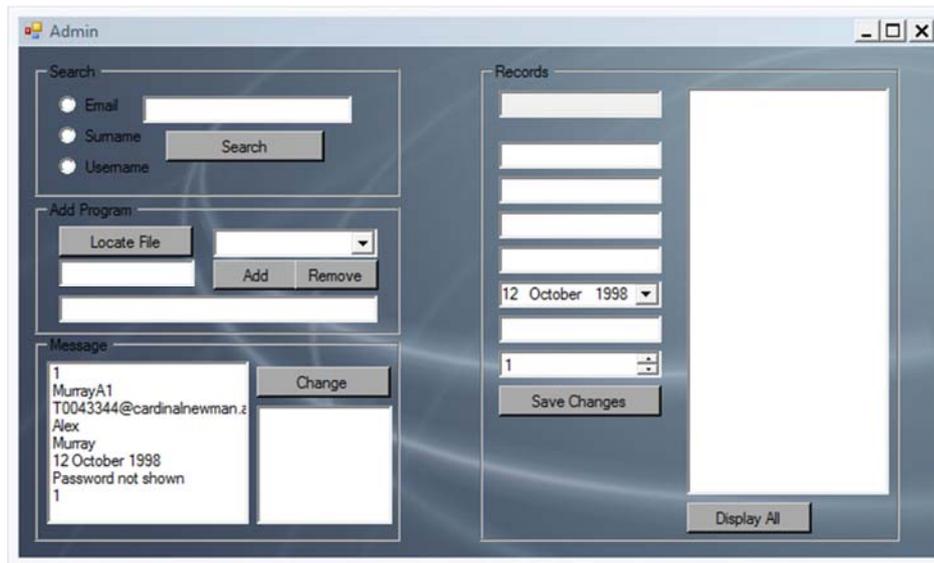
Once the test has been finished the student can mark the test, this show the correct answers to the test in green and the incorrect in red, allowing the student to take in where they did well and where they did poor. The colour scheme helps the users as green is positive and red negative.

The algorithm page holds all the imteractive algorithms that the student can use, this allows them to learn and develop their own understanding of the algorithm better.

The studnts just have to press the one they wish to load and the algorithm will appear. This method of hiding and showing the algorithm helps to keep the form clear so the student can focus on just one at a time (as some users could get distracted.)

The algorithm are all stored in a group box, within this group box the algorithms are easy to use, each having only a few buttons and or textboxes needed to oporate them, this keeps them simple to use and makes them easy for new users to learn how to use them.

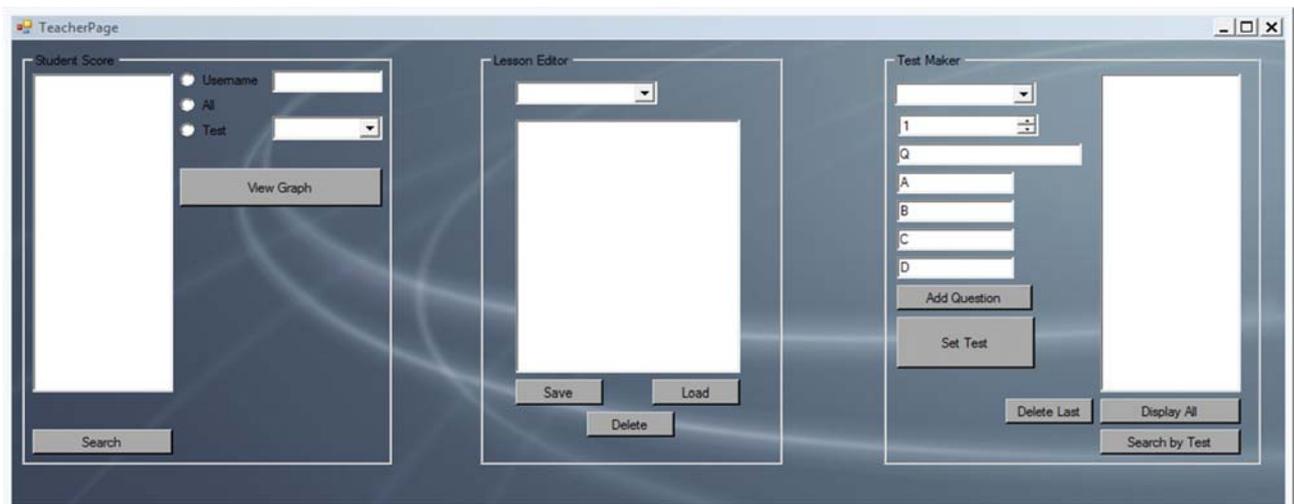




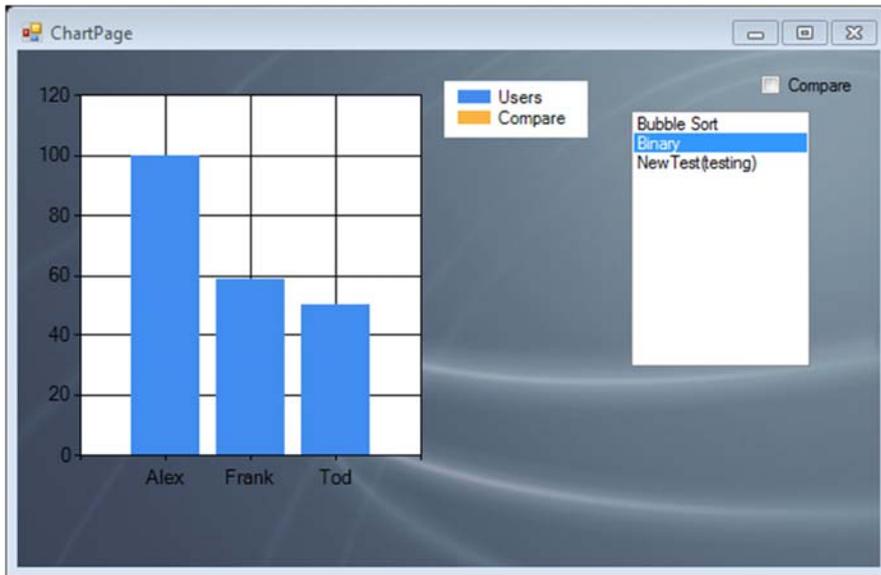
The Admin page is the most powerful page on my system, this allows admins and only admins (as only an admin access will grant access to the page) to edit and change user files, and any other important details in the file.

The page allows the admins to search for a particular user on the user file and even change the data on the file. As admins are supposed to be (higher than average) users of computers and computer systems. Therefore the page is a lot more formal and professional, as to allow the admins to perform the tasks more efficiently.

The admin can use the main display box to set the user's data into the correct fields at the press of a mouse click by clicking the user's record number. This allows the admin to save time when changing or checking details.



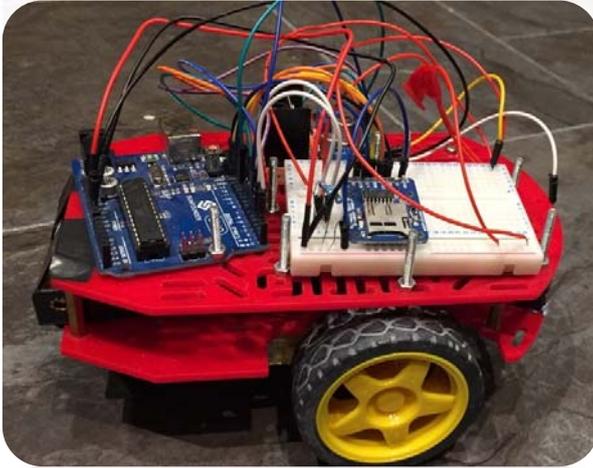
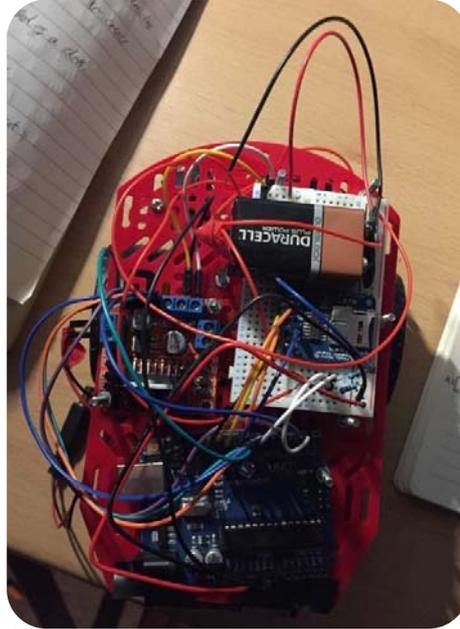
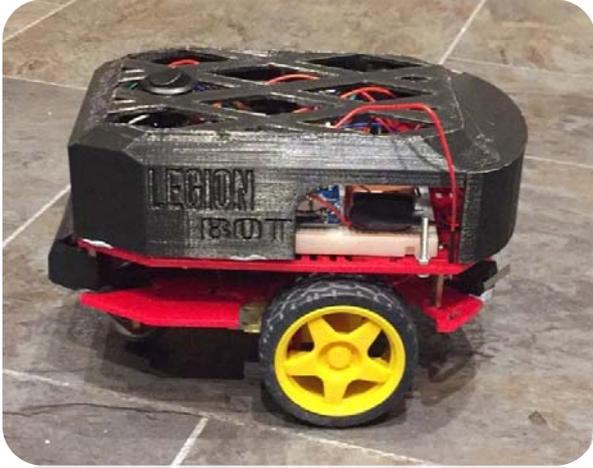
The teacher page is organized to allow the teachers to focus on one of three main tasks at a time: student scores, Lesson editor, and test maker. This way the teacher is less likely to veer from the task at hand. The grouped tasks all have the options to perform file editing with the correct task. For the teachers' convenience they were separated. For further events the teachers can select another option for the buttons linked to the correct methods.



The chart page allows teacher to compare the students test results with one another, this allows them to see how well a student is doing in comparison with another. A bar chart was chosen over another type of chart as it's the easiest to read quickly and the easiest to compare between other subjects on the chart.

The setting test page was designed to show the teachers if the student has been keeping up with set work, the page allows the teachers to easily add tests for the student to complete. If the students have completed the test set it appears as green, if not red. This gives a clear yes or no indication to if the student has completed the work on time or not.

The page also allows the teacher to easily set test by selecting the students name from a dropdown list if needed or selecting all students to set the test to. The deadline for the test can also be easily selected by using the window's dropdown timetable, this makes it easier for the teachers as they will have most likely used the built-in function before.



The bot was built to be easy to use by any user with a very short guide. The switch that turns on the bot can be seen on the black cover case of the bot, the simple flick switch sits on top of the case and allows the user to access it without hassle.

The user does not have to see (or worry about pulling out wires as they are fragile) the bot's wiring as the case covers it, this 3D printed case has gaps in the top and side.

The gap in the top allows the user to see the built-in lights that the Arduino, SD card reader, and motor driver all emit, this allows the user to have a small check to see if it's working properly (visual check.) The diamond holes in the top also have a secondary purpose, they enabled the bot to print faster when using the 3D printer (as it takes a long time) but also to improve the look of the case (as the material has a unflattering look at for just a flat surface) The other hole at the side allows access for the SD and battery this lets the user access them without having to take the case off.

The bot also has a buzzer on it, this allows for an audible output to the user, when the bot is turned on, to ensure the user is aware of what mode the bot is in.

Lastly the bot has two 'micro' switches attached that act as both the human input and the bot's sensors, they are located at the front of the bot below the main case.

## Developmental testing:

### Core Components:

- Bot can move around correctly
- Bot collects Data (saves to SD card)
- Graphics Show Movement of the bot
- Data Saved correctly
- Edits to data
- Charts load correct data
- Deletion of a data set
- Only high level uses can access the correct pages

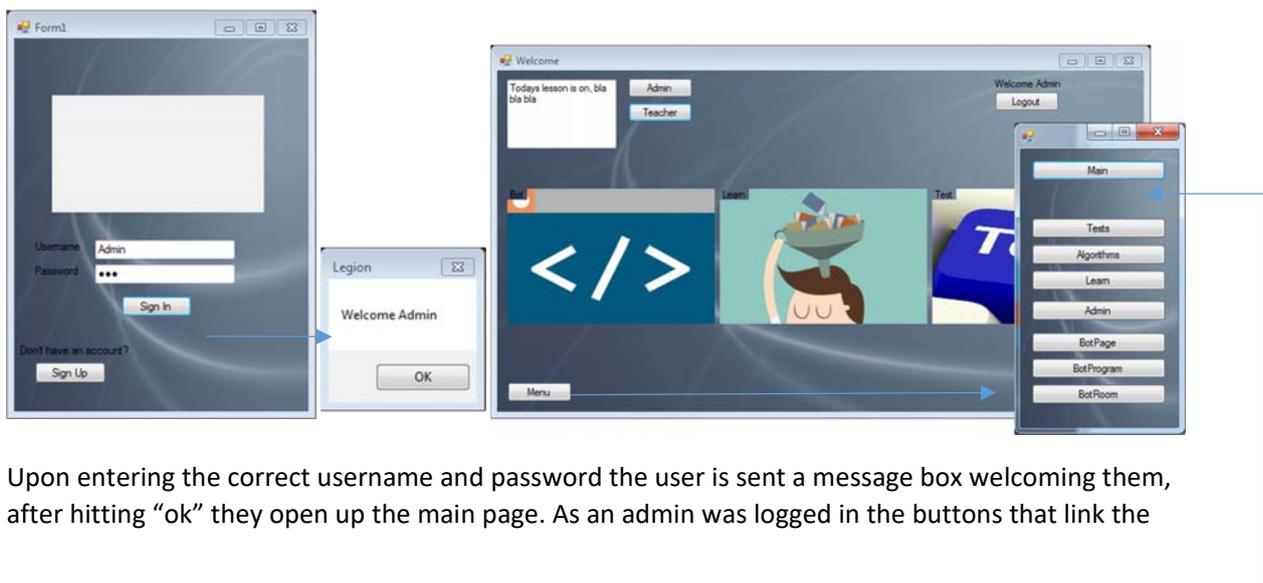
### General features:

- All List and combo boxes have the data load into them
- All buttons link the correct pages
- When taking an input, the input is used from the correct location
- All sorts work properly
- All searches work
- Images are as expected
- layouts or population of items (show and hide) works as expected

### Tests:

Test for allowing only the correct access level to open the correct pages-

This test will be ensuring that only the users will the correct level can use the button links that take them to the correct page. I will log into my system with accounts I have generated, they will each have different Access levels, (Student, Teacher, and Admin) This way I will be able to ensure that each user has the correct restrictions to my system.



admin and teacher page have loaded onto the welcome form as expected. However, I have found a small error in my system. The 'Teacher page' button on the menu has not appeared.

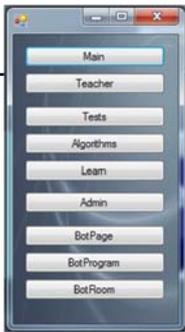
```

end if
If TBUsername.Text = SystemUsers(Counter).UserNameStr And TBPASSWORD.Text = SystemUsers(Counter).PasswordStr Then
    UserIdNo = Counter
    LoggedIn = True 'when name and password match lets value to true, stopps checking for more than needed.
    MsgBox("Welcome " & TBUsername.Text)
    Welcome.LABUsername.Text = ("Welcome " & TBUsername.Text)
    Welcome.Show()
    Me.Hide()
    If SystemUsers(Counter).AccessLvInt = 3 Then 'hides or shows the right buttons for admin or teacher
        Welcome.Student()
    ElseIf SystemUsers(Counter).AccessLvInt = 2 Then
        Welcome.Teacher()
        MenuPage.BtnTeacher.Visible = True
    ElseIf SystemUsers(Counter).AccessLvInt = 1 Or 0 Then
        Welcome.Admin()
        MenuPage.BtnAdmin.Visible = True
    End If
Else
End If
End If
If TBUsername.Text = SystemUsers(Counter).UserNameStr And TBPASSWORD.Text = SystemUsers(Counter).PasswordStr Then
    UserIdNo = Counter
    LoggedIn = True 'when name and password match lets value to true, stopps checking for more than needed.
    MsgBox("Welcome " & TBUsername.Text)
    Welcome.LABUsername.Text = ("Welcome " & TBUsername.Text)
    Welcome.Show()
    Me.Hide()
    If SystemUsers(Counter).AccessLvInt = 3 Then 'hides or shows the right buttons for admin or teacher
        Welcome.Student()
    ElseIf SystemUsers(Counter).AccessLvInt = 2 Then
        Welcome.Teacher()
        MenuPage.BtnTeacher.Visible = True
    ElseIf SystemUsers(Counter).AccessLvInt = 1 Or 0 Then
        Welcome.Admin()
        MenuPage.BtnAdmin.Visible = True
    End If
Else
End If
End If

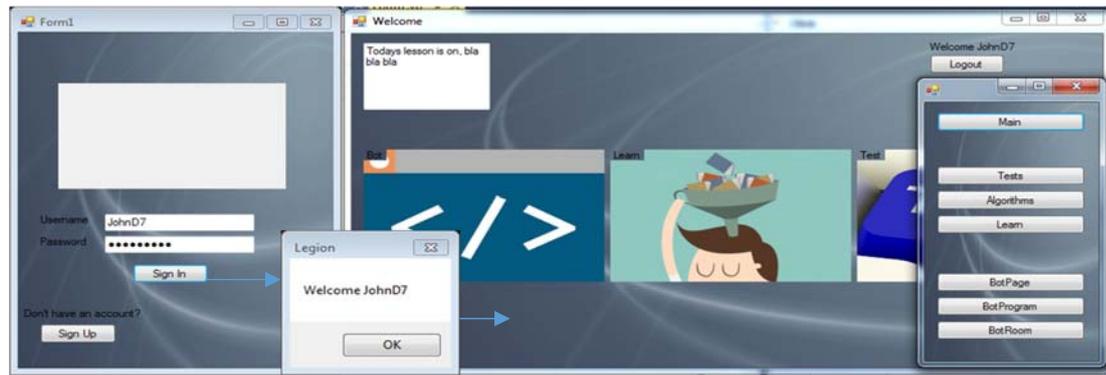
```

A line of code had been missed, The button was not turned from invisible to visible.

I added the line of code I had missed, once added I r-started the test  
This was the result



I then retook the test but with a student and teacher login details, both these test worked as expected. (e.g. Below the logged in student does not have the access levels, therefore they don't have the teacher and admin page buttons on the menu or welcome page)

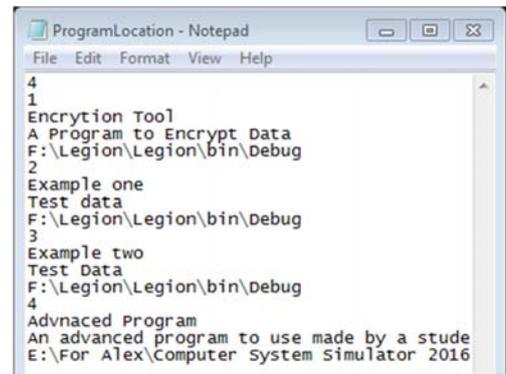


Deletion of data –

To ensure my program has the capability of erasing/removing data that is unwanted/unneeded. I will test the function of my program to ensure I can remove a record from one of my text files. This will allow the data to be removed from my system. In this test I removed a file from the program list (on the admin page.) Here is the File of the program list before a record was removed, at the moment it has four records in it, two of which are test data the other two are real life data

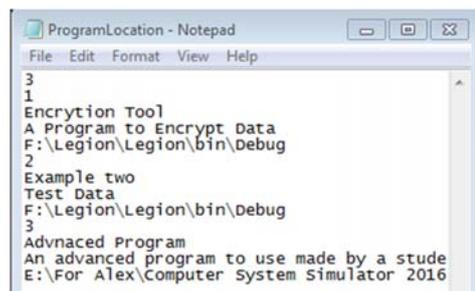


selected is removed.



When the remove button is presses the record is removed and the user is sent a message box to show the task has been completed.

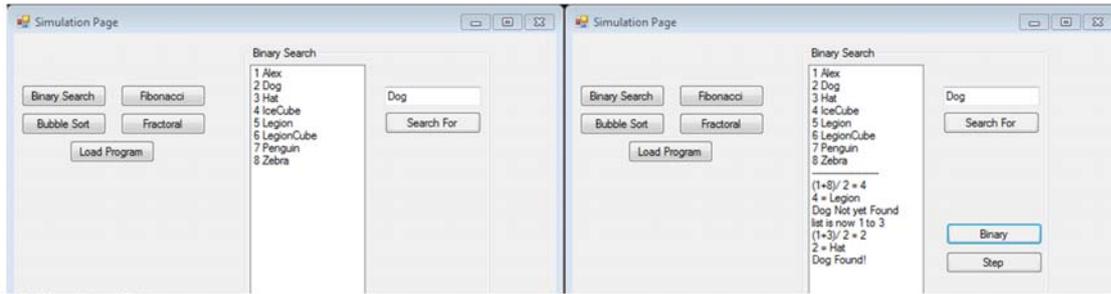
The Text file is amended and the record



Data is loaded correctly into combo and textboxes –

Its important data in my system was loaded into the correct textbox, I used multiple forms to ensure this worked. My forms all loaded in the correct data when the correct circumstances were met (e.g. Button pressed, or form loaded. [below is evidence form the teacher page]



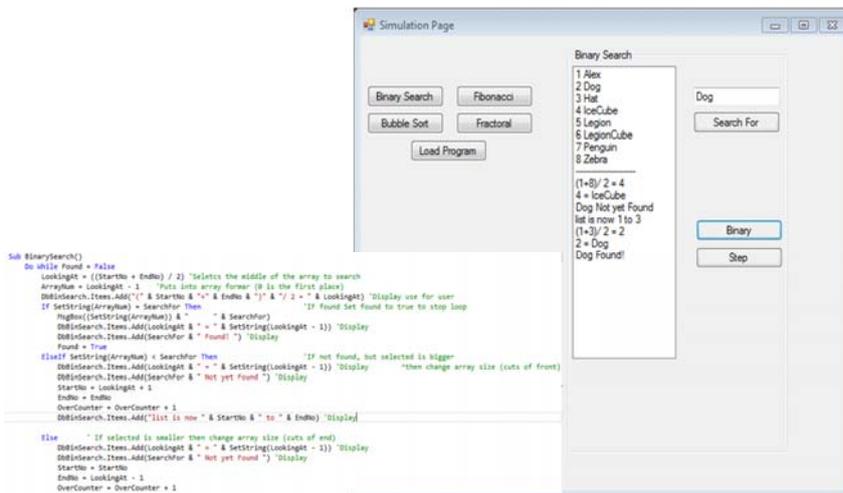


```

Sub BinarySearch()
    Do While Found = False
        LookingAt = ((StartNo + EndNo) / 2) 'Selects the middle of the array to search
        ArrayNum = LookingAt - 1 'Puts into array format (0 is the first place)
        DbBinSearch.Items.Add("(" & StartNo & "+" & EndNo & ") & "/ 2 = " & LookingAt) 'Display use for user
        If SetString(ArrayNum) = SearchFor Then 'IF found Set found to true to stop loop
            MsgBox((SetString(ArrayNum)) & " " & SearchFor)
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt)) 'Display
            DbBinSearch.Items.Add(SearchFor & " Found! ") 'Display
            Found = True
        ElseIf SetString(ArrayNum) < SearchFor Then 'If not found, but selected is bigger
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt)) 'Display 'then change array size (cuts of front)
            DbBinSearch.Items.Add(SearchFor & " Not yet Found ") 'Display
            StartNo = LookingAt + 1
            EndNo = EndNo
            OverCounter = OverCounter + 1
            DbBinSearch.Items.Add("list is now " & StartNo & " to " & EndNo) 'Display
        Else ' If selected is smaller then change array size (cuts of end)
            DbBinSearch.Items.Add(LookingAt & " = " & SetString(LookingAt)) 'Display
            DbBinSearch.Items.Add(SearchFor & " Not yet Found ") 'Display
            StartNo = StartNo
            EndNo = LookingAt - 1
            OverCounter = OverCounter + 1
    End Do
End Sub

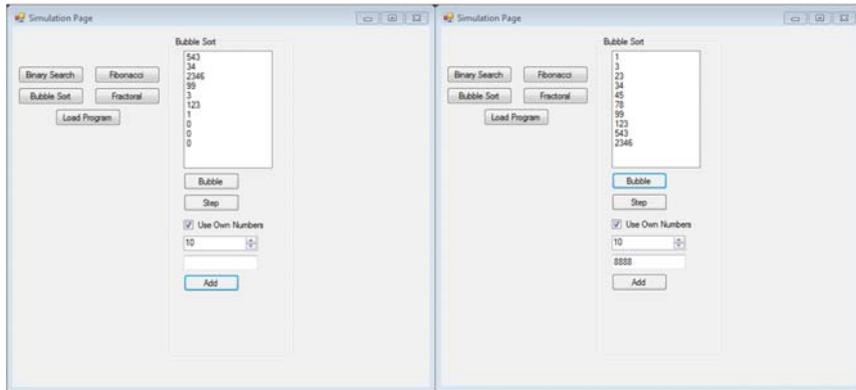
```

With my Test I found that the search did work as it should, however I was getting a wrong value back for the word, but the correct value for the search number. This was due to an error I had made, as the program reads arrays as the first item being in position 0, and I had used the first item as position 1.



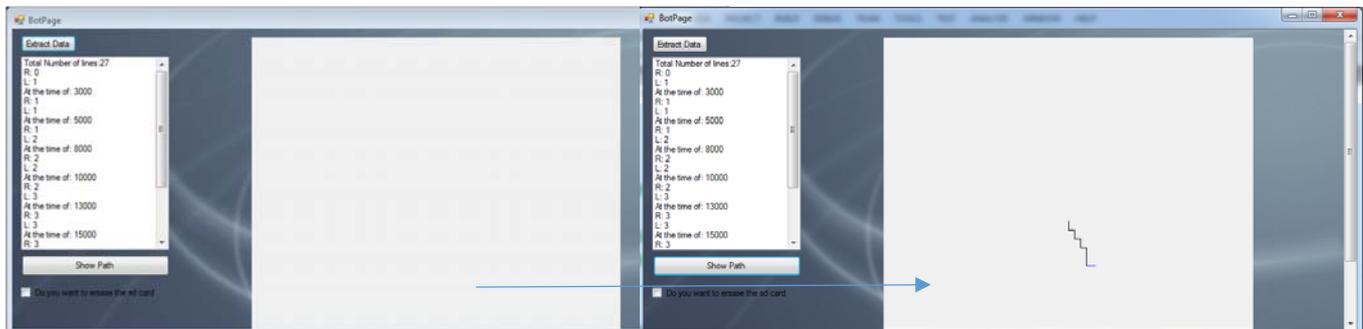
After I had amended my code the correct results was sent to the text box at the correct place. Therefore the test now had passed.

Similar to the binary search I tested my bubble sorting algorithm to ensure that I can use it with other parts of my system.



The test was completed and passes, once the array was filled the numbers were placed into the correct order by the system.

Lines can be drawn from the bot's data –



Once the User presses the show path button the program converts the text data from the bot into coordinates that draw lines onto the mapping area. This test was a success as the lines drawn were correct.

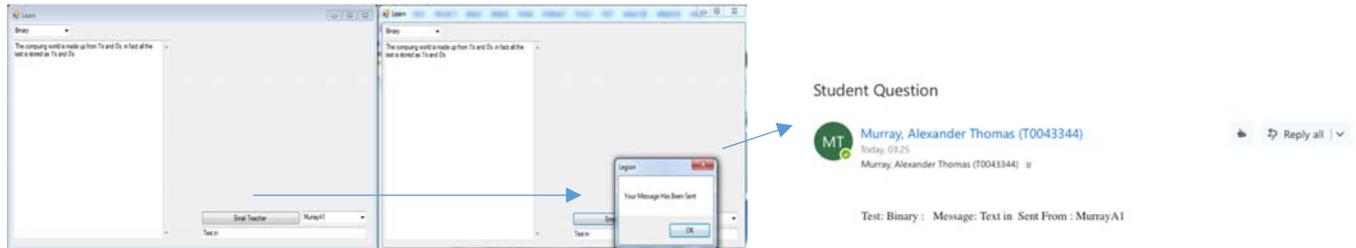
The bot can store data and take inputs from the real world-

I connected the bot up to a computer via a serial port connection, (using the built in serial connection function the Arduino sketch allows) This connection allowed me to 'see' what the bot processed based on the 'flags' I left in the code. After connecting the bot and setting its mode I randomly started pressing the switches at the front of the bot, as this changed the flag on screen and saved the data to the sd card, I believe this test was successful. (Video proof – SW Test)

Test

### Email Function-

My system should be able to send out an email to a teacher from the Learn page for if a student is stuck or struggling with a lesson, this test will see if my system can send an email from its self to a user, all while only using data given into the system. The user selects a teacher and then inputs the wording they wish the email to have.



The test was a success, the email was sent by the program and then received later at the correct teacher's address. The information on the email was correct.

### Loading form display –

For the UI to look more presentable than just a page full of simulations, on the simulation page I made sure to hide the algorithms that went being used until the user wanted to see them, when the user presses a button the algorithms appear, and when a different one is selected, a different one appears. The test was a success as the simulations couldn't be seen unless the selected simulation button was pressed.



### Movement of the virtual bot-

On the bot program page, the user moves a virtual bot around on screen when a button is pressed, this test is to make sure the on screen bot can be moved and sent to the correct direction on the screen, this will ensure that the graphical side of my system is working.



Up 5 times

Right and UP 7 times

This test was a success as the on screen bot moved as expected allowing the user to see it.

Python encryption code-

With the exemplar code I created to give as an example the user enters a message and a keyword to be encrypted, this test will make sure I can actually encrypt a message with the code I have written.

The code should allow me to input a message to the computer and then allow a keyword to be entered. Both will then be used to make an encrypted message. From the encrypted message and the keyword

```
Python 3.4.3 Shell
Python 3.4.3 (v3.4.3:9b72f0c4dd1, Feb 22 2015, 02:32:09)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> """
===== RESTART =====
>>> Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
E
what is your message? hat
what is your keyword? c
Your coded message is ... = kdw
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
d
what is the encrypted message? kdw
what was the keyword? c
Your decoded message is ... = gzs
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
"""
```

the program should be able to decrypt the given message. For this test I used the message "hat", and keyword of "c" the output should be kdw.

The encryption worked well, but when decrypting the wrong message was given, instead of 'hat' I got 'gzs' this was one letter lower than it should have been.

<pre>keytake=[] decoded=[] Hold="" for char in Message:     output.append(ord(char)) for char in key:     keytake.append(ord(char)-95)  keylen=len(Message)//len(key) keyl=(keytake*keylen)+keytake</pre>	<pre>#Stores concatenated message. #Converts message into ASCII. #Changes key into alphabet number #Makes key length equal to</pre>	<pre>decoded=[] Hold="" for char in Message:     output.append(ord(char)) for char in key:     keytake.append(ord(char)-96)  keylen=len(Message)//len(key) keyl=(keytake*keylen)+keytake</pre>
---	---	--

After looking into the code, I found I had made a small logic error. The alphabet in ASCII code starts at 97, therefore in my code I subtract 96 to represent the letter 'a' as the number 1, (b = 2, c = 3, etc) However I must have misread the ASCII table and typed in 95 instead of 96. Once I corrected this mistake the program worked as it was intended to.

```
>>> ===== RESTART =====
>>> Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
E
what is your message? hat
what is your keyword? c
Your coded message is ... = kdw
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
d
what is the encrypted message? kdw
what was the keyword? c
Your decoded message is ... = hat
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
"""
```

Images at the welcome page change periodically –

The images on the welcome page should change every two seconds, making them load up a different photo, they should cycle three photos for each main 'button' and then change back to the original ones before redoing the loop.



These screenshots show the different images the program cycled through, it repeated the same sequence after, proving that the loop will continue therefore the images will keep changing.

#### Bot test audible output-

I will test that the bot can sound the audible output (the buzzer) it has connected to it, this buzzer needs to be able to produce a sound loud enough for the user to hear.

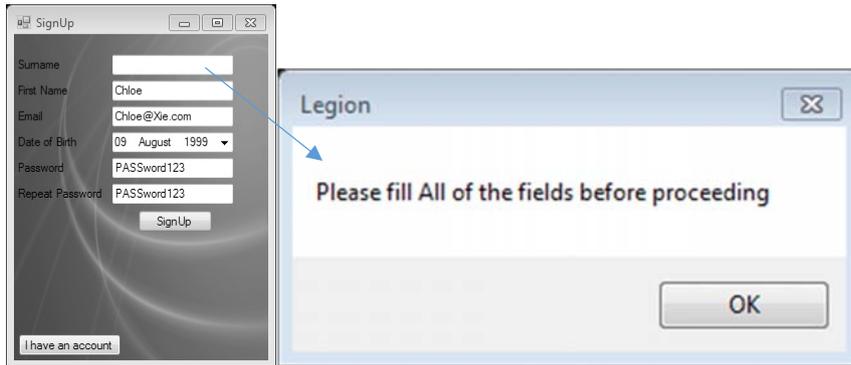
Upon testing the bot I couldn't hear the buzzer however the bot was still functioning properly with mode selection and movement (this is shown in video buzzer test. Upon further investigation I found this was not due to a coding error, but a physical one. One of the wires had become loose and was not connected to the breadboard, this meant the buzzer could not generate the correct sound, after reconnecting the wire, the buzzer worked as normal (buzzer connection video.) If I had the resources I would use a more stable way of connecting the bot's wiring up, such as a prototype circuit board.

#### Test for mode selection –

I tested that the modes of the bot could be correctly selected by the user when the bot is turned on. I used the micro switches to change the mode of the bot and load up the different code it performs, for the different tasks in the mode. Proof in video bot mode test. This test was a success but it can still be difficult to select the correct mode for the bot. In an ideal world I could add a real button or form of menu system to the bot, however due to limitations on hardware I cannot.

#### Validation test –

While making my system I had to ensure no data could be entered to break it, therefore I validated any data entry points that could cause potential system crashes, when I made a validated data entry point I gave it a quick test to ensure it worked. This test was a success as the program picked up there was no data inputted into a field, and therefore should not be save the information.



#### Bot movement and control-

As the bot is one of the key features of the system I needed to ensure it was working properly through its stages of development. When I had got the collision mode code developed I tested it on the bot (see video Bot without PWM.) As you can see in the video the bot moved and turned as and when it should, however the moment was very jumpy and fast, a but too fast for the small bot. Another key issue I found upon this test was the bots ability to move in a straight line, there was no errors in the code of the bot to prevent it moving in a straight line. Upon further investigation I found that this was due to the wheels of the bot bot being identical, one was ever so slightly misshaped, meaning that the wheels would never be able to turn at the same rate for the same movement. In order to fix this issue I did some research into motor controls for DC motors. From this research I got the idea to use Pulse Width Modulation. This allowed me to control the individual motor speeds by sending an analog signal to them that pulses at a varying voltage, rather than just one continues set voltage, this allowed me to slow down one motor and speed up the other just enough to make up the for physical shape and move the bot in more of a straight direction. See bot with case after video.

The code and wiring of the bot did have to be changed to allow PWM control, this took up extra recourses on the Arduino board and on the memory it could store on it.

```

//Forward
void MoveForward(){ //Both Forward
  digitalWrite (M1a, HIGH); //Motor A
  digitalWrite (M1b, LOW);
  analogWrite (9, 180);
  //
  digitalWrite (M2a, LOW); //Motor B
  digitalWrite (M2b, HIGH);
  analogWrite (3, 160);
}

```

Here I changed the code to ensure the motors had different pulses going to them, the '9' and '3' are the analogue pins that had to be used and the '180' and '160' are the different pulse frequency that are sent to the motors.

## Testing

Testing is very important, as in order for a system to work properly it needs to have no mistakes when given to the real world. So to ensure that any mistakes are picked up and fixed before the system is released vigorous testing is needed.

### Validation and verification –

One of the most common and best ways to ensure data allowed, is to validate the entering of the data (checking the data 'looks' correct.) Another way of ensuring data is correct is to verify it, checking that the data that was accepted into the system is the actual correct data and not just similar. In order to ensure my data is valid and verified I will have to enter different types of valid and invalid data into my system deliberately to ensure of its acceptable or not.

### Functionality-

I will need to be sure all parts of my system works, all the files are saved into, all the buttons are connected to perform their tasks, all the forms work as they are expected. The program should function as its intended and not produce errors or unexpected results. In order to test these functions, I will use the function, using a range of data (if required) and provide the result given with the expected result. Some of the functions I will be testing are:

- Encryption and Decryption
- File handling (adding, editing, deleting, and reading records)
- Sorts
- Searches
- Simulations
- Graphics (lines drawn, charts populated)
- Emails
- Tests
- Loading of programs
- Loading of videos
- Bot usage

### Interface –

My system was developed to help teaching and learning, therefore the interface must be 'easy to read' With any text/list/combo boxes working properly and populated as needed, charts and 'visibility' of the displays also need to be tested to ensure they work. I can manually test that textboxes are populated correctly and that textboxes and graphical systems work by seeing what result should be given, and

which result is given. I will check the interface is functioning properly throughout all my functionality testing and then from a questionnaire.

#### Compatibility Testing –

As my system was designed to work with any windows based computer, I will need to ensure it can work on different processors (32, 64 bit processors) so long as its on a system using windows my system should run normally. I will also be testing the compatibility of the programs that can be added by the user. As some of the programs that can be loaded are not pure executable files there could be compatibility issues with the programs loading on different computers (if the language has not been installed to the system for example.) I will also test the compatibility of the actual Arduino board and ensure that the code I have written can be loaded on to any Arduino Uno, or other appropriate micro controllers. These tests will involve using my code on different computers/systems to ensure they still function, or to see the restriction a system would have, and what hardware/software my system requires to work to its full extent.

#### Beta Testing-

In order to test my system to the full extent I will need to use data that represent the real world and how users would actually enter data (as I could potentially miss something when testing that could lead to the system breaking.) To test the system, I will use the help of beta testers who will be able to give a much wider range of testing as they might think of different data to enter that could crash the system. I'm going to use multiple testers to give a wider range of data and allow more overall time to be spent testing the system (as this will allow more vigorous testing) After the beta testers are finished I will get them to fill in a questionnaire, this will provide me with feedback of how the system preformed on different areas, including usability.

#### Tests:

##### Validation and verification testing-

Test Number	Test Type	Description and use	Criteria	Expected Result	Actual result and proof
1	Valid – Test on presence check	Used to check data has been entered into the text box's (cannot be left blank. For all data on sign up form	The text box must contain data of some form.  Data entered: "Chloe"	The program should allow the data and proceeds with the next stage	Data was entered and accepted  Screenshots 1,2

2	Invalid - Test on presence check	Used to check data has been entered into the text box (cannot be left blank. For all data on signup form	The text box must contain data of some form.  Data entered: <i>-Null-</i>	The program should not allow the data to be acceded and not proceed Output an error message	Data was rejected  Screenshots 3,5
3	Invalid - Test on presence check	Used to check data has been entered into the text box (cannot be left blank. For all data on signup form	The text box must contain data of some form.  Data entered: <i>-Null-</i>	The program should not allow the data to be acceded and not proceed. Output an error message	Data was rejected  Screenshots 4,5
4	Valid test – Double entry	Used to check the user entered the password correctly Used in- Signup form	The two textboxes must contain the same data Data entered: "PASSword123" "PASSword123"	The system should accept the password and proceed with next step	Data was accepted  Screenshots 1,2
5	Invalid test- Double entry	Used to check the user entered the password correctly Used in- Signup form	The two textboxes must contain the same data Data entered: "PASSword123" "123passWORD"	The system should not allow the data and send out the correct error message	Data was rejected  Screenshots 6,7
6	Invalid presence check	Used to see of the data entered had a particular item/value Used in Sign Up form	The password must have a capital letter, lowercase letter and a number. Data used: "PASSword"	The system should not allow the data and send out the correct error message	The data was rejected and the error message sent out Screenshots 8,9
7	Invalid presence check	Used to see of the data entered had a particular item/value Used in Sign Up form	The password must have a capital letter, lowercase letter and a number. Data used: "password123"	The system should not allow the data and send out the correct error message	The data was rejected and the error message sent out Screenshots 9,10

8	Valid presence check (regular expressions)	Used to see of the data entered had a particular item/value Used in Sign Up form	The email must contain some form of string, then an @ sign then another string and a .com/.co/etc Data used: "Chloe@Xie.com"	The system should accept the password and begin the next stage of the code.	The data was accepted and the next stage of the code began.  Screenshots: 1,2
9	Invalid presence check (regular expressions)	Used to see of the data entered had a particular item/value Used in Sign Up form	The email must contain some form of string, then an @ sign then another string and a .com/.co/etc Data used: "Chloe.com"	The data entered should be rejected with an error message telling the user its not valid.	The data was rejected and the correct message sent out.  Screenshots: 11,12
10	Invalid presence check (regular expressions)	Used to see of the data entered had a particular item/value Used in Sign Up form	The email must contain some form of string, then an @ sign then another string and a .com/.co/etc Data used: "ChloeXie@gmail"	The data entered should be rejected with an error message telling the user its not valid.	The data was rejected and the correct message sent out.  Screenshots: 12,13
11 12 13 14	Valid Range check	Used to stop the user trying to set the size of an array to a ridiculously big size Used on the "use own numbers section "on bubble sort	The number selected must be <21 and >than 0 Data Used: "10", "12", "19", "5"	The number selected should be accepted and the program should generate an array with the correct length	The value was accepted and program generated the array to the correct length.  Screenshots: 14, 15, 16, 17

15 16 17 18	Invalid Range check	Used to stop the user trying to set the size of an array to a ridiculously big size Used on the "use own numbers section "on bubble sort	The number selected must be <21 and >than 0  Data Used: "23", "89", "-5", "0"	The number selected should be rejected and a message sent to the user.	The positive values were rejected and the message sent to the user then the box reset. For the negative and 0 values the box reset to 1 automatically Screenshots: 20,21,23,24,25, 26,27
19 20	Extreme Range check	Used to stop the user trying to set the size of an array to a ridiculously big size Used on the "use own numbers section "on bubble sort	The number selected must be < 21 and < 0  Data Used: "20", "1"	The number selected should be accepted and the program should generate an array with the correct length	The values were accepted and program generated the array to the correct length.  Screenshots: 18,19
21 22	Valid Verification check	Used to ensure the link entered can only connect to YouTube Used on the learn page	The link entered must be a YouTube link that has the correct link format Link entered: "https://www.youtube.com/watch?v=95kv5BF2Z9E" " https://www.youtube.com/watch?v=-KafyF6M4rw"	This link should be accepted and the system should load up the correct video.	The link was accepted and the video was loaded  Screenshots: 28,29

23 24	Invalid verification check	Used to ensure the link entered can only connect to YouTube Used on the learn page	The link entered must be a YouTube link that has the correct link format Link entered: "Logic Gates" "Ram and rom video"	The data should be rejected, no video should load and a message will be sent out	The links were rejected and the video was not searched for, a message was sent to the user. Screenshots: 30,31
25	Extreme verification check	Used to ensure the link entered can only connect to YouTube Used on the learn page  A damaged link will be entered	The link entered must be a YouTube link that has the correct link format Link entered: <a href="https://www.youtube.com/watch?v=-KaF6M4w">https://www.youtube.com/watch?v=-KaF6M4w</a>	The data should be accepted ,but no video should load.	The link was accepted but no video loaded but the video box did change to black (not an issue) Screenshots: 32
26 27	Valid Length Check	To ensure no one enter useless data (or a string long enough to slow or break the system) a length check will be preformed on the names of a user in the signup page.	The length of the first and second name should be below 15 characters each.  Data entered: "Chloe" "Xie"	I expect the data to be accepted and the program to continue as normal.	The data was accepted and the system contended as expected  Screenshots: 1,2
28 29	Invalid Length Check	To ensure no one enter useless data (or a string long enough to slow or break the system) a length check will be preformed on the names of a user in the signup page.	The length of the first and second name should be below 15 characters each.  Data entered: "abcdefghijklmnopqrstuvwxyz" "abcdefghijklmnopqrstuvwxyz"	The names should be rejected as they are over the allowed length and a message should be sent out to the user as they attempt to enter the data.	The system stopped the user entering data after the 20 <sup>th</sup> letter, sent out a message and cleared the data entered as expected. Screenshots: 33,34
30	Valid Presence check	Its important that when data is saved into the file	All the fields must be populated in	The system should accept the data,	The data was accepted, a message was

		that information is actually entered into the file and not left blank.	order for the file to be changed. For the tests all the fields were filled.	change the file and send a message to the user.	sent to the user Screenshots: 35,36
31	Invalid Presence check	Its important that when data is saved into the file that information is actually entered into the file and not left blank.	All the fields must be populated in order for the file to be changed. For the tests I removed data from different fields	The data should be rejected, the program will not run, and a message to the user saying so	The data was rejected, a message was sent to the user.  Screenshots: 37,38,38,40
32 33 34	Valid – Numeric check	When the user enters data (to be added to a numeric array) the data HAS to be numeric in order to prevent errors – found in bubble sort own number section	The data entered must be a number. It cannot be string or other data type, only a number. Test Data: "1", "999999", "18"	The data should be accepted, the program will add the numbers to the array.	The data was accepted and the numbers were added too the array.  Screenshots: 41,42,43
35 36 37	invalid – Numeric check	When the user enters data (to be added to a numeric array) the data HAS to be numeric in order to prevent errors – found in bubble sort own number section	The data entered must be a number. It cannot be string or other data type, only a number. Test Data: "one", "Hat", "7e"	The data should be rejected by the system, and a message popup the user saying so.	The data was rejected and a message sent out to the user.  Screenshots: 44,45,46
38 39 40	Extreme – Numeric check	When the user enters data (to be added to a numeric array) it has to be numeric but will more values than just integers be accepted	The data entered must be a number. It cannot be string or other data type, only a number. Test Data: "5.6", "0.732", "-54.63"	I'm expecting the data to still be accepted as they are still numerical values if not a common integer.	The data was accepted into the system, however it was rounded. This is acceptable as rounding has no effect on the task its self Screenshots: 47,48,49,50

Functionality Testing-

Test Number	Test Type	Description	Criteria/data entered	Expected Result	Actual result and proof
41 42	Encryption Test	Test to see if the students user data can be securely encrypted and cannot be easily read.	The text with the information on it must be unreadable by a human after encrypted.	The data should be encrypted and impossible to read.	The file was successfully encrypted, making it unreadable without the decryption tool. Screenshots: 51,52,54,55
43 44	Decryption Test	Test to see if the data can be decrypted back into its original form.	The text must go from being encrypted to a readable form	The data in the file should return to its original form.	The file was decrypted, making it readable again Screenshots: 52,53,55,56
45	File Handling: Addition of data – program list	Test to ensure the 'program file' can have data added into it correctly.	The data added has to be saved into the text after the 'add' button has been pressed. Data entered: "Test Name" "Test Description-this is a test" Location of a program	I expect the data to be in the file, in the correct order as the data already in the file	The data was saved into the file correctly.  Screenshots: 57,58,59

46	File Handling: Removal of data– program list	This test is to ensure I can remove selected data from a file if needed.	The record selected will be removed from the file. Record Selected: “CatSoftware”	I expect the record to be removed and for all the other records to move up one in its place.	The selected record was removed and the other record moved up in its place as expected. Screenshots: 59,60,61
47	File Handling: Addition of a record – Learn Data	Test to add in a record to the Learn data file. The test will see if the program can add data entered by a teacher	The record created should be added to the file, exactly as it was written.	I expect the file will be saved into correctly and the new record will be added. Record added “TestData” “this is test data”	The record was correctly saved into the file and can be seen when the file is opened. Screenshots: 62,63,64
48	Sorting – Bubble sort of learn data file	This test checks the sorting of the Learn data is correct, data will be inputted to be sorted to see if its in the correct order or not	The file should be in alphabetic order, any records are arranged by the alphabetic order of the names. I will input data {Abc, A, Z, E, Q, P} in that order	I expect the file to have sorted the data when I open it, and to contain A,Abc,E,P,Q,Z in the order above (with any data already in the file moved accordingly)	The sort worked, the records were all placed into the correct location, and other moved around it accordingly.  Screenshots: 64,65
49	File Handling: Removal of a record– Learn Data	This test is for ensuring I can remove data from the file, (the correct/chosen data)	The file should have the selected record removed from it, without disrupting the others. Record to remove is “Abc”	I would expect the record to be removed leaving the others untouched.	The test was a success. The data removed was only “Abc” and it kept the other data in order and in tact Screenshots: 65,66,67
50	File Handling: Editing of a record– Learn Data	Editing the data in the file need to be tested to ensure the correct data is	The file should only have the record edited, no other records apart from the chosen.	I expect the file will have the record changed appropriately and leave the	The record was changed accordingly and the other files remained untouched.

		being changed in the record	I will edit the record "E"	others unchanged.	Screenshots: 67,68,69
51	Security of system: logging in (reading user file) – valid login	This test is to see if the login is secure, it will also act as a test to see if the user file is being read properly as it needs to read the file to know usernames and passwords	Only if the correct password and username is entered should the program allow me entry For this test I will use the correct login and username I created before	The program should recognize that my username and password match, and allow me (MurrayA1) entry to the system.	The test worked as expected, a message box welcomed me, and the I was sent to the correct page which showed my username Screenshots: 70,71
52	Security of system: logging in (reading user file) – invalid login	This test is to see if the login is secure, it will also act as a test to see if the user file is being read properly as it needs to read the file to know usernames and passwords	Only if the correct password and username is entered should the program allow me entry For this test I will use the incorrect password for the login.	The program should find no match for the password to the username and request me to check my username and password are correct	As expected I was denied entry to the system And the message was given to check my login details. Screenshots: 72
53	Security of system: logging in (reading user file) – invalid login	This test is to see if the login is secure, it will also act as a test to see if the user file is being read properly as it needs to read the file to know usernames and passwords	Only if the correct password and username is entered should the program allow me entry For this test I will use my username but another accounts password for the login.	The program should find no match for the password to the username and request me to check my username and password are correct	As expected I was denied entry to the system And the message was given to check my login details. Screenshots: 73
54	File handling: Writing and saving message for the main page	The main page has a message that is displayed to all users, this message is saved by the admin into a text file.	The file should have no parts of the old message left in it, and hold the new message as it was written by the admin.	The file will change to the new message as it should this will save the new message and erase the old.	The test worked as expected, the file was changed correctly Screenshots: 74,75,76

55 56	File handling: Saving Tests to the Test Layout file  Prevention of having the question entered twice	The teacher should be able to generate test for the students to use and take, this test is about ensuring the teacher can save and store the tests made.	The test should write into the file one question at a time, and prevent the teacher from entering a question on the same test twice. I will generate a test using test data.	The system should save the test questions to the text file without any problems. And alert the user if the question number already exists.	Worked as expected, the questions were saved into the file one by one, and a message stopped the user from entering the question twice Screenshots: 77,78,79,80,81
57  58  59	File handling: Reading the tests from the test layout file.  Graphics: Loading in of the check boxes randomly  Search: the correct test data should be found and loaded	This test will be ensuring the system can read the test layout file correctly, collect the required data and then display the data in a randomized order (as a multiple choice test).	The test selected must be the correct test to load, all the questions should be in order and the answers randomized.	I expect the system to work as intended and load the answers randomly from the correct file.	The correct data from the file was loaded correctly and the answers were randomized in an unpredictable pattern.  Screenshots: 82,83,84,85,86, 87
60 61 62 63	Functionality Valid marking tests	In order to save the students, score the system needs to mark the students test and save the score. This test will ensure the correct score is being marked.	The system should check the student's answers are correct, marking them green if the answer is correct and red if not. I will mark the correct answer "3" "4" "5" and "1" times	I expect the system to mark the students answers correctly, and output the data as a message to the user.	The test was a success the data was marked and the correct scores given. With the correct color changes  Screenshots: 88,90,91

64	Functionality Invalid marking tests	In order to prevent cheating In the tests the system stops the students from checking more than five boxes. This test it to ensure the system works.	I the system should pick up that more than five boxes have been checked and output a message to the user and set the score to 0 For this test I will check over five check boxes.	The system should see that more than 5 boxes has been checked, and reject the students answers.	The system picked up on the student entering more than five check boxes and made their test score invalid.  Screenshots: 89
65	File handling Data should be saved to the "student score" file after being marked	This test is to ensure the data saved has been correctly added to the students score text file.	The file should contain a new record that holds the students username, test data, and score.	The text will be a success if the data has been added after a test was taken.	The file had the data added to it as expected, the correct data was added to the record into the file Screenshots: 92,93
66	Sorts: Simulation on bubble sort – Test A	This test will check if my bubble sort algorithm works at it should	The system should take the numbers and sort them into the correct order and display this to the user. Data Used: 12,18,99,0,1,107	I expect the data to be outputted it the correct order of: 0,1,12,18,99, 107	The numbers were sorted correctly and as expected.  Screen shots: 94,95
67	Sorts: Simulation on bubble sort – Test B	This test will check if my bubble sort algorithm works at it should	The system should take the numbers and sort them into the correct order and display this to the user. Data Used: 54,22,22,98,34,0, 31,63,72,3	I expect the data to be outputted it the correct order of: 0,3,22,22,31, 34,54,63,72, 98	The numbers were sorted correctly and as expected.  Screen shots: 96,97
68	Sorts: Simulation on bubble sort – Test C	This test will check if my bubble sort algorithm works at it should	The system should take the numbers and sort them into the correct order and display this to the user. Data Used: -2,-8,-99,99,0,1,4	I expect the data to be outputted it the correct order of: -99,-8,-2, 0,1,4,99	The numbers were sorted correctly and as expected.  Screen shots: 98,99

69	Sorts: Simulation page, stepping bubble sort -Test A	This test will check that my program can step through a bubble sort part by part.	The system should only produce one pass on the data Numbers used: 13,20,12,42,16	The data should be in the order 13,12,20, 16,42	The data was in the correct order and displayed to the user Screenshots: 100,101
70	Sorts: Simulation page, stepping bubble sort -Test B	This test will check that my program can step through a bubble sort part by part.	The system should only produce one pass on the data Numbers used: 13,12,20, 16,42	The data should be in the order of 12,13,16, 20,42	The data was in the correct order and displayed to the user Screenshots: 101,102
71	Sorts: Simulation page, stepping bubble sort -Test C	This test will check that my program can step through a bubble sort part by part.	The system should only produce one pass on the data Numbers used: 999,5,998, 1000,12	The data should be in the order 5,998,999, 12,1000	The data was in the correct order and displayed to the user Screenshots: 103,104
72	Sorts: Simulation page, stepping bubble sort -Test D	This test will check that my program can step through a bubble sort part by part.	The system should only produce one pass on the data Numbers used: 5,998,999, 12,1000	The data should be in the order: 5,998,12, 999,1000	The data was in the correct order and displayed to the user Screenshots: 104,105
73 74	Searches: Binary search algorithm page	This search is a simulation of how a binary search would work, I need to ensure that this search works in order to show others how it should preform correctly	The search must find the correct item to be searched for. Without selecting the incorrect one. I will search for "Dog" in the set array Then "LegionCube"	The program should find the word dog without any issues and show the 'thought process' in the display box	The test was a success, the words were correctly found and the 'thought' process was shown Screenshots: 106,107,108
75 76 77 78	Searches: Binary search algorithm	This search is a simulation of how a binary search would	The search must run one step if the process. Without	The program should look at one piece of	The test was a success, after stepping through the

	page –step search	work, I need to ensure that this search works in order to show others how it should preform correctly	selecting the incorrect one. I will search for “Penguin” in the set array	information in the array and cut the array list down per step	algorithm the word was found  Screenshots: 109,110,111, 112
79	Searches: Binary search algorithm page – when word not in array	This test ensures the system shows the correct result if the data is not found in the array	In order to pass the search must run through the ordering of the list, cutting down the size until there are no other possible options. Test data: “Andromeda”	In order to pass the program must search thought the correct part of the list and output a ‘not found’ result.	Test passed, the program worked as expected and an output message was given Screenshots: 113
80	Simulation: Fractal check Test - A	This test will check the recursion fractal simulation, to ensure it provides the correct result	To pass the tests the simulation, the program must produce the correct result. Find fractal “5”	The program should run and calculate “5*4*3*2*1” = 120	The test worked, the correct output was given  Screenshot: 114
81	Simulation: Fractal check Test - B	This test will check the recursion fractal simulation, to ensure it provides the correct result	To pass the tests the simulation, the program must produce the correct result. Find fractal “9”	The program should run and calculate “9*8*7*6*5*4*3*2*1” = 362880	The test was a success the correct output was given  Screenshot: 115
82	Simulation: Fibonacci	This test will ensure the Fibonacci sequence generator will produce the numbers.	To pass this test the Fibonacci sequence must be generated correctly	I expect the numbers [1,1,2,3,5,8, 13,21,34,55,8 9...etc]	The numbers were generated correctly and loaded into the correct display box. Screenshot: 116

83	Simulation: Fibonacci - step	This test will check that Fibonacci sequence will work when stepped.	The sequence should be produced one number at time.	Numbers should be produced one at a time.	The numbers were generated and shown to the user each time they stepped the program. Screenshots: 117,118,119
84	Simulation: Loading up a program	This test will ensure I can load up a program that has been added by an admin	For this test to work the program made by the admin should load up and be available to use	I expect the selected program to load up and for the user to be able to use it.	The file loaded as expected and the program was useable.  Screenshots: 120,121
85 86	Search: Searching for the correct username – Admin page	The search function will be tested to ensure the correct/relevant information can be brought up when requested.	For the search to be a success it should find the correct record, and display it to the user. Searched for: "MurrayA1" "TestS5"	As the username is unique, only one correct record should be found, and presented to the user.	The search found the correct record, and presented it to the user.  Screenshots: 129,130,131
87 88	Search: Searching for the correct email – Admin page	The search function will be tested to ensure the correct/relevant information can be brought up when requested.	For the search to be a success it should find the correct email and match the record, displaying it. "T0043344@cardinalnewman.ac.uk" " "	The system should match the email and show the record in the display box.	The correct records were found and presented to the user in the display box.  Screenshots: 126,136
89 90	Search: Searching for the correct Second name – Admin Page	The search function will be tested to ensure the correct/relevant information can be brought up when requested.	The search should find the correct records with the same last name as the one entered. "Murray" "TestData"	The search function should bring up all the records with the the same last name and show them to the admin.	The correct records were shown to the admin.  Screenshots: 132,133, 134,135
91 93	Search:	I will load the file with a large	The record found should be	I expect the file to be	The correct record was

92 94	Binary search Learn data on a big file – Learn page Function: Finds the correct record relatively quickly for a big file.	amount of test data (to act as real life data sample) the data will then be searched for a particular record.	the record selected. The time taken to find the record should be relatively quick (within a second) I will search for the records “Mango” and “Alphabet”	found well within the time, and for the correct data to be located.	found and loaded into the display box, the time taken to load the data as miniscule, results were found instantly to the human eye. Screenshots: 137,138, 139,140
95 96	Email: Admin Record	This Tests will be checking the email function of the admin page – ensuring changes to a record are documented.	When the admin changes information in the user record file, this data (before its changed) should be emailed to an address	I expect the record (before its been changed) to be sent by email. Creating a copy.	The test worked as expected, and email holding the unchanged data was sent to a secure email address Screenshots: 146,147,148, 149, 150,151, 152,153
97 98	Email: Student to Teacher	To ensure the emailing system works. I will need to test the ‘student to teacher email’	The student writes a a small email to the teacher, with the lesson they need help with.	The correct teacher should receive an email with the student request for help	An email was sent to the correct address with the correct message held within it. Screenshots: 141,142,155, 153,154,156
99 100	Search: Students Score - Username	Making sure the system can search and find the correct students score from all the different records from the username of the student.	The system has to be able to find the correct, students score and display the details. Username searched: “MurrayA1” “Tod”	The system should find the correct students data and show it to the teacher.	The correct records were brought up from the file and displayed to the user  Screenshots: 157,158 159,160

101 102	Search: Students Score - Tests	The teacher can also search the score using the tests as the search criteria. This test will ensure the system works.	The system should search the data file and find the records holding the same tests name as the search test. I will test "binary" and "TestData" tests	The system should output all the saved data to the display box about students who have taken the test.	The test worked as expected the correct output was given to the user. The correct records were found and shown. Screenshots: 161,162, 163,164
103 104	Graphics: Chart loading Chart page	This tests will be conducted to ensure the bar chart will be populated properly.	The bar chart should be populated with the correct data from the students scores dependent on the test selected I will select "Binary" and then "Bubble sort" to be displayed.	I expect the graph to show all the students who took the test and there scores in comparison to each other.	The graph showed the correct test data and displayed the correct students and score for the two different tests in graphical form. Screenshots: 165,166 167,168
104	Graphics: Chart loading two data sets	This tests will be conducted to ensure the bar chart will be populated properly when two tests are selected.	The bar chart should be populated with the correct data from both the students scores form both tests	I expect the graph to show all the students who took the test and there scores in comparison to each other, but for both test selected.	The graph showed the correct test data and displayed the correct scores when comparing the tests. Screenshots: 169,170
105 106	File handling: Bot data open and read by system (Bot Program) (bot room)	The bots data needs to be able to be read by the main program, this test will ensure the data can be read by the main system.	The data from the bot file should be read into the main system properly an saved into an array. I will use two different data	The data should be read into the system properly and the data displayed to the user	Data was read and inputted into the system without problems. Screenshots: 171,172 173,174

			sets. "Cube" and "Box"		
107 108	Graphics: Bot Path Drawn	This test will see if the bots path can be drawn from the data it has collected. I will first test 'perfect data' and then real data collected.	The data should be converted into graphics the user should see the movement of the bot from the data. I will use "perfect data" and then "test data"	The data should indicate when the bot has turned and how far the bot moved in-between the turns.	The bots path was shown on screen by the system, however it wasn't entirely accurate to to the angle that the bot can turn. Regardless the R and L turning and distance was displayed correctly. Screenshots: 182,183,184,185
109 110	Graphics: Room path drawn	The bots path should be drawn on screen for the user to see, this test will ensure the system works as intended.	The test will be a success if the room is displayed to the user, this should show the user the layout of the walls of the room. I will use "perfect data" and then "real data"	I expect the bot to draw out the map of a room, showing the user where the wall line is and how long it is (relevant to the size of the grid)	The path of the room was given, in regards to the wall of the room, the reverse lines in blue and the forward motion in black. Screenshots: 186,187
111	Graphics: Virtual bot moves -Program mode	On the 'bot program' page an on screen bot moves when the user presses a button, this test will ensure the system works as it should.	The display shown to the user has to represent the mapping of the room. I will use perfect data and then real data.	I expect the bot to move up and down, depending on the direction is facing and the instruction given by the user	The on screen bot moved up and down the page as it should, dependent on the buttons (for direction) pressed. Screenshots: 175-181
112	Graphics: On screen compass changes- Program mode	The program should change the direction of the compass when the on screen bot changes direction.	The compass must change to show the correct direction of the bot after the user has moved the bot.	I expect the compass to change color and direction when the left or right button is pressed.	The color of the compass changed, showing me which way the bot was facing. Screenshots: 176,177, 179,180
113	Bot – The sensors on the bot	I will connect the bot to the computer as to	The computer should display when the	When the sensors have been pressed	The bot sensors activated when presses sending

	activate when hit	obtain real-time feedback. This will prove if the bot can pick up when something activates the sensors.	sensors have been hit, as the serial port will be connected.	the computer should display the information to prove it has been hit.	out a different message to the information on the computer screen. Videos: SwTest And all othes
114	Bot – collision mode	To test the bot mode, I will turn the bot on and then select the wanted mode. With the collision mode I will have the bot move and collide with obstacles to make a path for the computer.	In order to pass this test the bot should hit an obstacle (Wall or other) and then save the data	I expect the bot to turn when it hits an object, then move back, turn and set off again.	The bot moved as it should, stopping when it hit something to turn and move forward again.  Video: Collision Mode
115	Bot – Program Mode	To test the bots mode, I will turn the bot on and then select the wanted mode. It will then try to run the program made on the user before hand.	In order to pass the next test, the bot should read the SD card and follow the preset instructions written on it.	The bot should read the data file and then sett off following the instructions the user has made before hand.	The bot moved on its own, following the instructions the user had made.  Video: Program mode
116	Bot – Room Mode	To test the bot mode, I will turn the bot on and then select the wanted mode. The bot should begin mapping out the room	To pass the bot should navigate around the room. Making a path that goes around the room, and stores the information.	The bot should stick to the left of the room, 'hugging' the left and keep returning left if not right	The bot followed the wall and stuck left finding its way around the room. However the bot did not always hit the sensors on the first try. Video: Room Mode
117 118	Python Example encryption mode	The code I wrote as an example needs to function properly and encrypt the message it was given	As the program uses a keyword and a message it has to change the words by the correct value I will enter messages " Hat"	The program should take the message and encrypt it, by using the keyword as the	The message was successfully encrypted and the message outputted to the user.

			and "Hello World" With the keyword of "C" and "Hat" respectively	encryption amount.	Screenshots: 124,122
119 120	Python Example decryption mode	The code I wrote as an example needs to function properly and decrypt the encrypted message	The program needs to decrypt the message to get back to the original (used encrypted messages from above tests)	The program should return the encrypted message to its original form.	The message was sent back to its original form as expected.  Screenshots: 125,122

### Interface testing results –

Along with the other testing I have performed, I have kept my eye on the displays and combo boxes to ensure the forms load in the data correctly, throughout the testing all the forms have loaded the correct information when requested. The forms all loaded correctly and appeared as intended. The screens and charts loaded properly and were populated accordingly. However, to make ensure all users feels the interface is as it should be I will obtain feedback from the questionnaire I will hand out to the beta testers.

### Compatibility Testing Results –

During my tests, I used the system on different computers to see how compatible and adaptable it was. During the tests I attempted to use the system on different computers, this produced the following (but expected results.) On the windows operated computers the system worked and loaded correctly, the only issue with the main program was when I attempted to load it on a OSX operating system, as expected my system would not load as it was designed for a windows system.

Although the main system worked as intended, the ability within to load other programs has causes some minor compatibility issues. When a program has been added to the system, (unless the program has been converted into an exe file) the code might not load correctly unless the languages IDE or library has been installed upon the computer. Fortunately, this will not effect the system dramatically, as only the admin will have the ability to add the programs, therefor this means they should also have the ability to add/download/install the needed software to the computer they wish the system to work. In short, in order for my system to be compatible the admins need to keep the software on the computer they choose to run my system on up to date with the IDE's and programming software that they need to run the code on. Screenshots:

As for the microcontroller compatibility, I found that the code will work for multiple microcontrollers, such as an Arduino Uno, and a similar board (such as a board made with different but similar components such as a 'sunflower' board.) This means I can allow the code to be loaded onto different

boards and therefore use different microcontrollers to make new bots (set with the code I provided.) I did also test uploading my code to other microcontrollers, such as a node mcu, however this test was unsuccessful as the microcontroller was not compatible with the code written, only certain micro controllers will accept the code and be able to perform the bots task. See screenshots:

### Beta Test review –

After giving my system to the beta testers, they vigorously tested the system using real data and real life scenarios. This should allow any faults or any oversights to be detected. The beta tests results did not reveal any new or system breaking problems that weren't already considered. There were crashes in the system, however they were mainly due to the compatibility issues such as not having the installed software to load the added programs. But this is not a system damaging issue. No beta testers found any major floors in the system, however a minor error was found with the system.

The error was an oversight with the system. One of the beta testers deliberately cut off the Wi-Fi connection from the device they ran the system on, this interrupted the emailing system as emails could not be sent from the system (a much minor side effect from this problem was the learn page video function did also not work.) This was not a huge problem in terms of having a working system, but it does contain the potential for undocumented editing of the user file. As this problem is unlikely to occur, and an admin login is still required to change the data.

### Acceptance Questionnaire results-

Criteria	Agree	Disagree
The system was easy to navigate	89%	11%
The system was a useful addition to the classroom	94%	6%
Information stored in the system (e.g. lessons) could be accessed correctly	100%	0%
Students could contact easily teacher from within the program	96%	4%
There was no need to leave the program after it was setup (for knowledge)	71%	29%
Topics needed for the syllabus were able to be added easily	82%	18%
The system helped to teach and educate students	78%	22%
Teachers could keep track of a students progress well	90%	10%
Admins have complete control over the system	99%	1%

The user interface is appealing and useful to the user	69%	31%
The simulations were use full and helped the education	83%	17%
Users felt their data was secure in the system	95%	5%
The bot was a useful learning tool	92%	8%
Users adapted to how the system worked and how to use it easily	79%	21%
All user of the system would like to use it again	97%	3%

### Analysis:

Overall I had some very positive feedback from the acceptance testers, they thought the system was good on the whole. However, like every system it wasn't perfect, it had some small problems with it. One of the areas it was let down on was the ability to stay in the program for all the classroom needs, this appeared to be scored relatively low compared to the rest of the feedback as some users wanted to use the internet to search for more information, and perform research tasks. Or load up a word document to make a report or take notes.

Another area for improvement was the user interface, this section was the biggest loss for the acceptance test. This could be as users found the design, 'bland' and wanted a more 'fluent system' such as hovering over something to reveal more text rather than clicking to get info. The UI wanted by the students was hard to understand until a similar one was shown.

Some areas did get positive feedback, for example the bot as an educational tool was a success, earning a positive 92% on feedback, this shows that the physical device was a good idea/ add on to the main system as it received such a high positive approval rating from the acceptance test.

Admins seem to really like their side of the system, they believed the admins had full control over the system and could perform any task they were required to, with a positive feedback of 99% I believe this was a full success as almost all the admins were happy with the system, as for the one percent nothing will ever be perfect, the acceptance was up to the individual, each has their own views on how something should/could be.

In all the system worked as it should function, all of the acceptance test seem to agree with this, as none of them were below 60% of positive, (only one below 70 at 69%) this means my system was effective at setting out what it had achieved to do. However, this means the system was nowhere near perfect as it didn't get all the criteria at least in 90%. This shows I have room to improve on my system. If I had more time I could go back and make changes to make it even closer to what the users want.

### **Future refinements (developed from testing):**

Upon testing it was clear although my system performed correctly in the test, it was clear that there were some overlooked areas in the system. Areas that could be improved even more if I had the time. These were not broken, but could use more care or adjustments for the future of the system.

The bot's movement one something that I could change, the wheels on the bot were adapted to turn at the correct speed using (PWM) however if the battery was low, the power could not be supplied correctly and the bot wouldn't move correctly. If I had some more time I could change the way the bot uses the power, and therefore increase the battery life.

Another improvement to the bot could be to use longer sensors, as the small microswitches, didn't always pick when the bot had hit something (as the contact area of the switches was so small compared to the bot's size)

The User Interface was disliked at different stages of testing (even my improvements made after beta testing didn't seem to fulfill its needs) this is something I could change, it would be best to bring in the help of another person, if not just for a second opinion as this seems to be a problem for me, but also someone who works on design, more than just the code. If I was to improve the UI on the current system I would make the graphics more welcoming, and maybe use a slowly changing background (like a screen saver) to make the student forms a little less 'formal'

Buttons were another point the students tended to dislike, not the function but the look of them, I had the right idea turning them into pictures, but just something like rounded edges on the buttons rather than just the rectangular ones. However, this is more of a graphics standpoint than UI, as it just makes the page more welcoming.

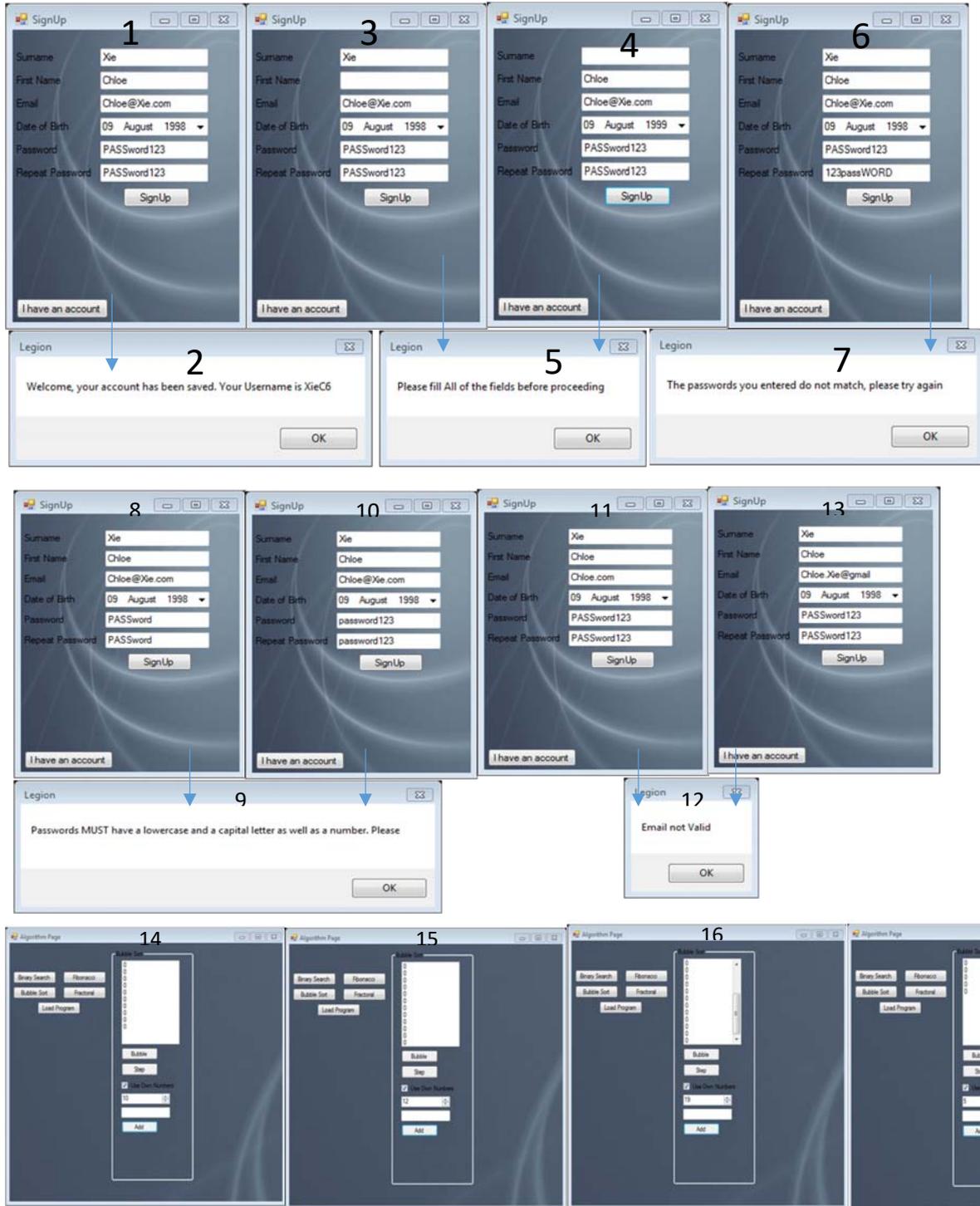
While on the subject of buttons, the layout I had on the bot program page (for the movement control) although a good design, could use further improvement. Instead of making the user click the buttons I could have allowed them to use the arrow keys on the keyboard to select the movement process, this isn't a huge change, but it would allow a user to use the program slightly faster.

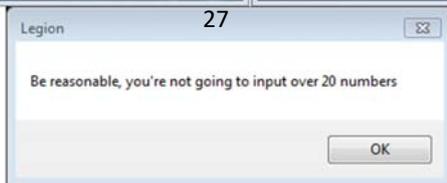
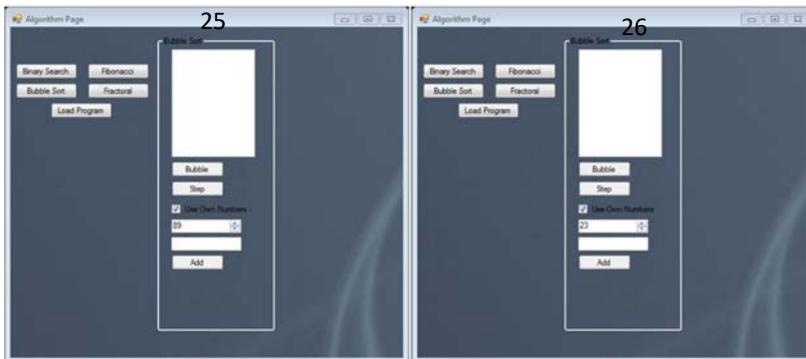
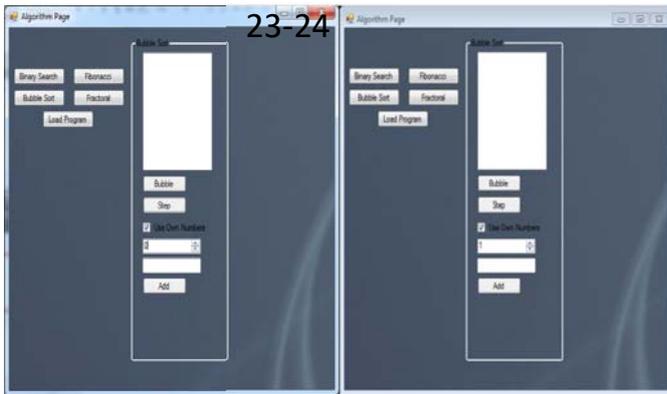
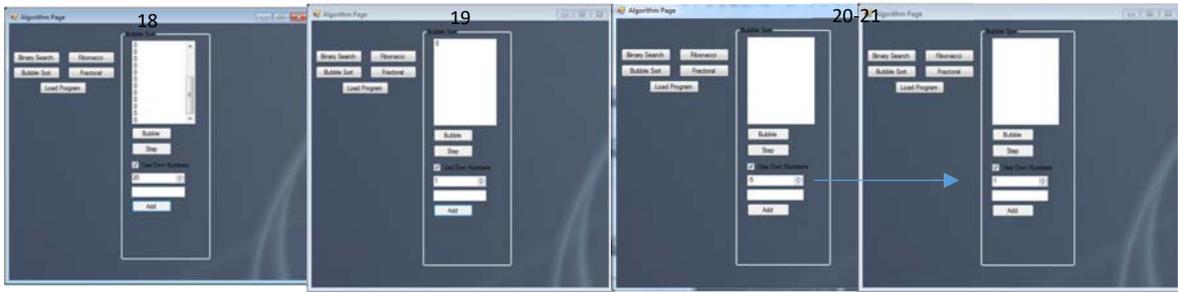
On the subject of graphics the system could be improved with some additional graphics, as seen from screen shots (175-181, 182-187, 166-170) The graphics here all did their part and showed the user the correct information, but they were very basic limited in functionality. With some refinement and maybe a new graphics system I could call upon better graphics that are more detailed and give me more freedom of what to do with them.

Another future refinement I could implement would be editing the email system to work with any form of an email account, currently the system requires an outlook email account in order to send emails, I could change this to allow any form of email account to be used, and not rely on the computer having outlook installed on it already.

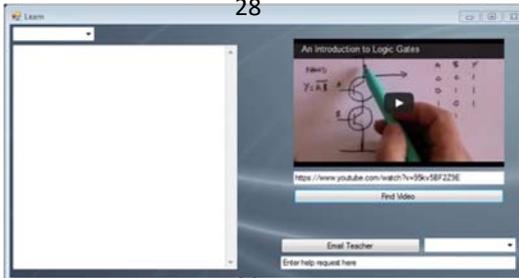
Lastly I would improve on the way the user can search for an item in the system, not the search method itself, but the feedback the user gets. As in testing I found the user had to remember the exact details they wanted to search for, if I were to change this I would add a degree of error, allowing for similar records or names to be found if the search wasn't completely spot on. (this would allow small misspellings to make it more user-friendly)

## Screenshots:





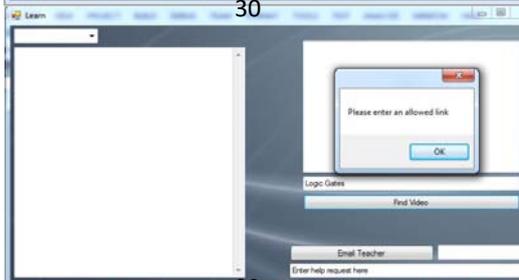
28



29



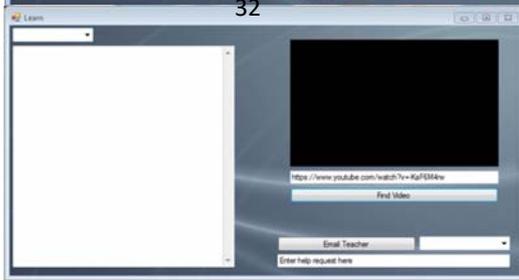
30



31



32



33

SignUp

Surname: abcdefghijklmnop

First Name: \_\_\_\_\_

Email: \_\_\_\_\_

Date of Birth: 14 September 2016

Password: \_\_\_\_\_

Repeat Password: \_\_\_\_\_

SignUp

I have an account

Legion

Please enter a shortened second name

OK

34

SignUp

Surname: \_\_\_\_\_

First Name: abcdefghijklmnop

Email: \_\_\_\_\_

Date of Birth: 14 September 2016

Password: \_\_\_\_\_

Repeat Password: \_\_\_\_\_

SignUp

I have an account

Legion

Please enter a shortened first name

OK

35

Admin

Search: Email, Surname, Username

Add Program: Locate File, Add, Remove

Message: MurnyA1, 70043544@cardinalnem...  
 Rex: Murny  
 12 October 1998  
 Password not shown

Records:

1	MurnyA1	70043544@cardinalnem...	12 October 1998	asd	1
2	Murny	70043544@cardinalnem...	12 October 1998	asd	2
3	Murny	70043544@cardinalnem...	12 October 1998	asd	3

Save Changes

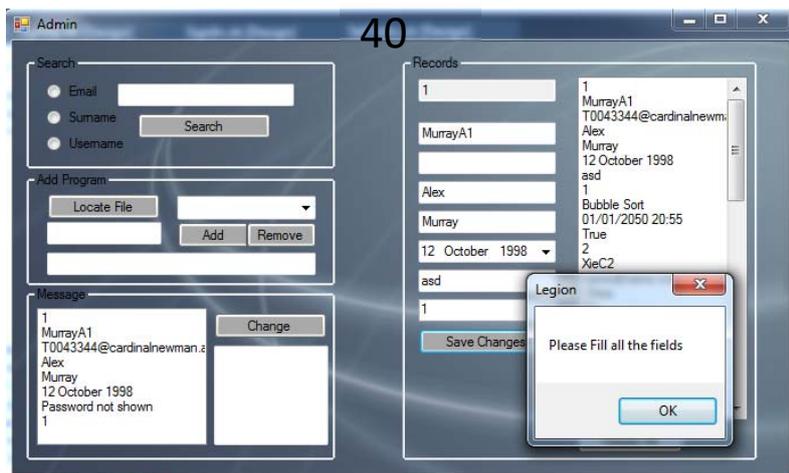
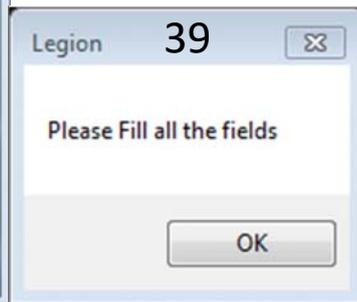
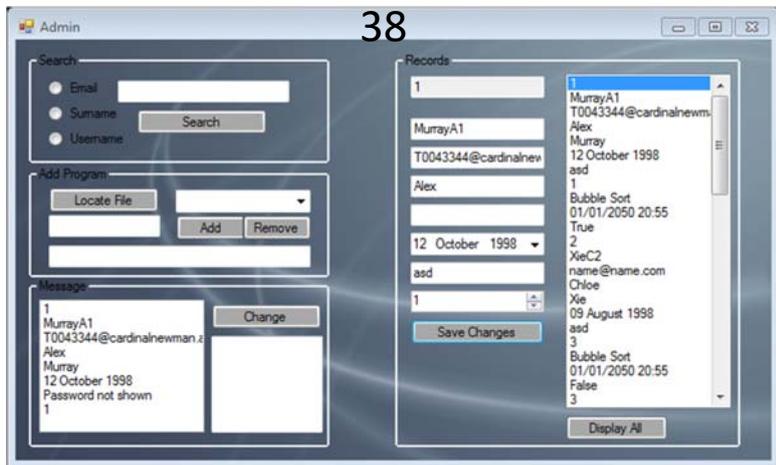
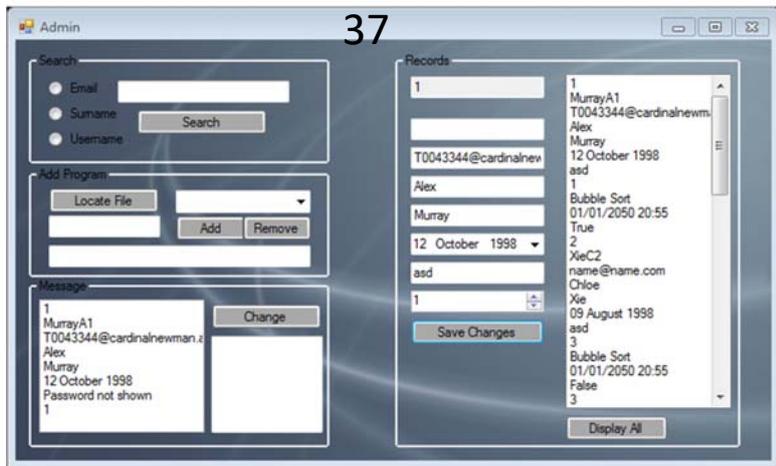
Display All

36

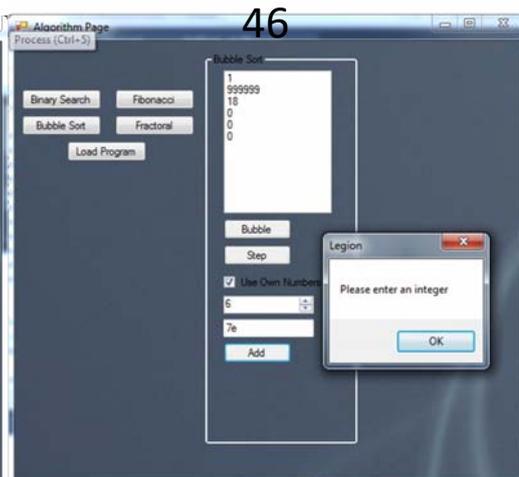
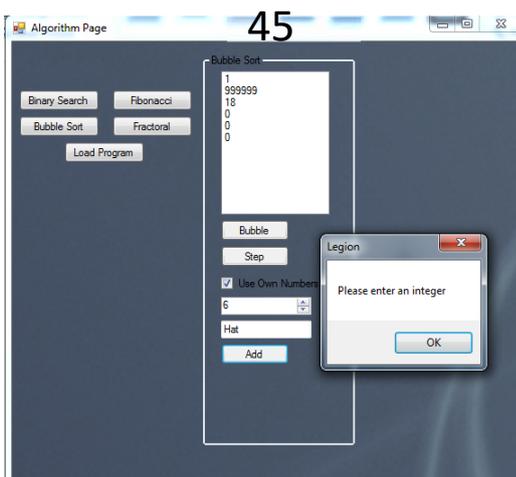
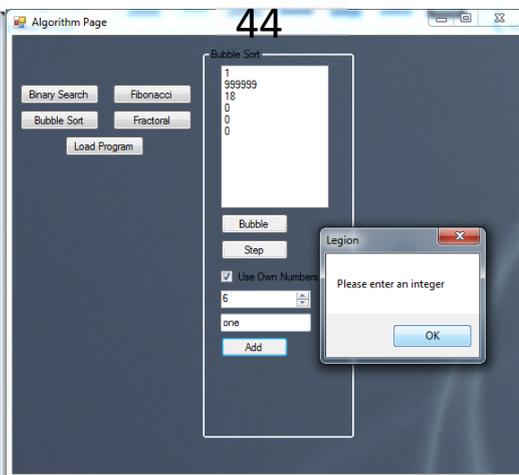
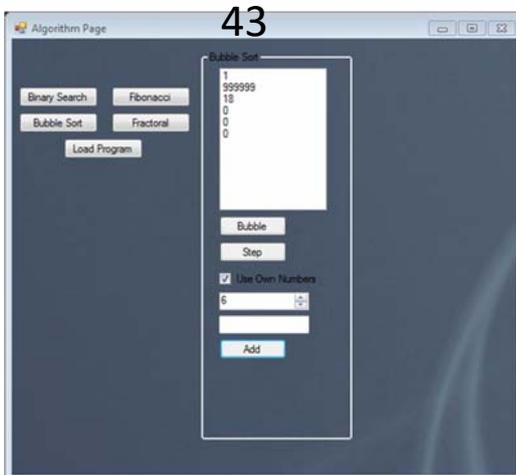
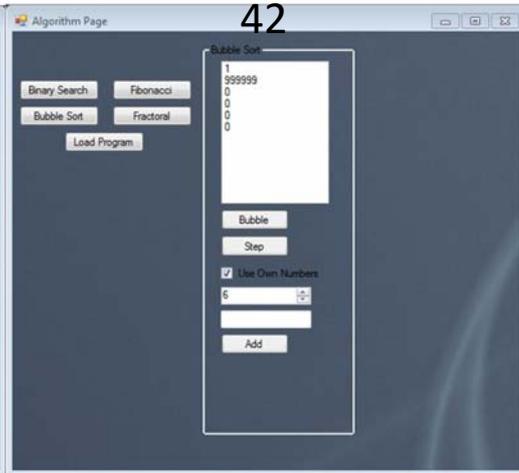
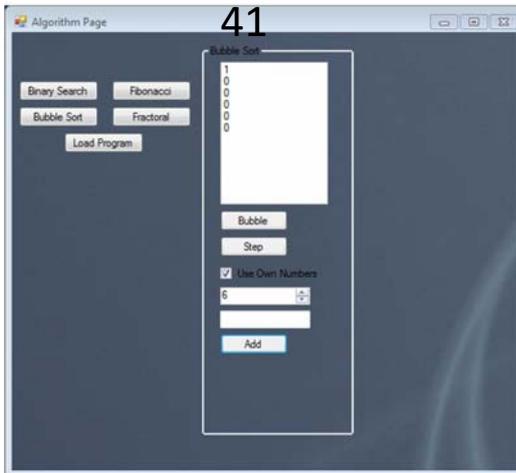
Legion

The data has been changed

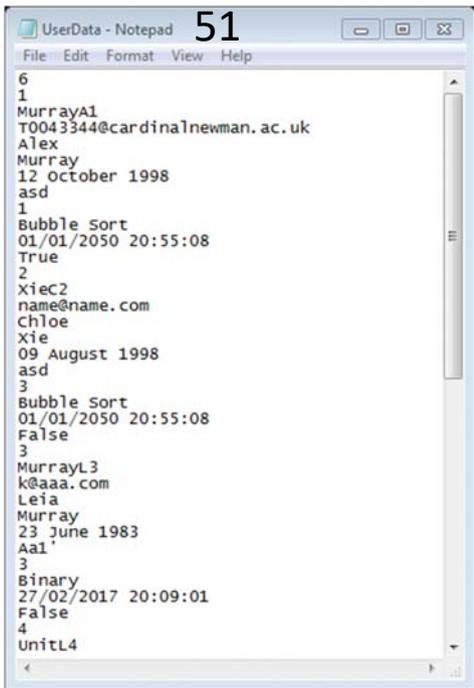
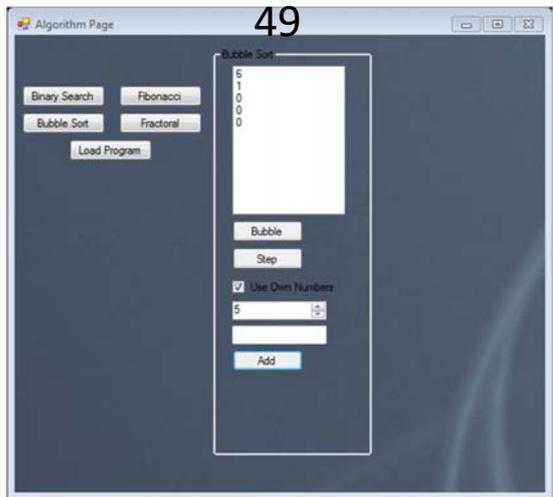
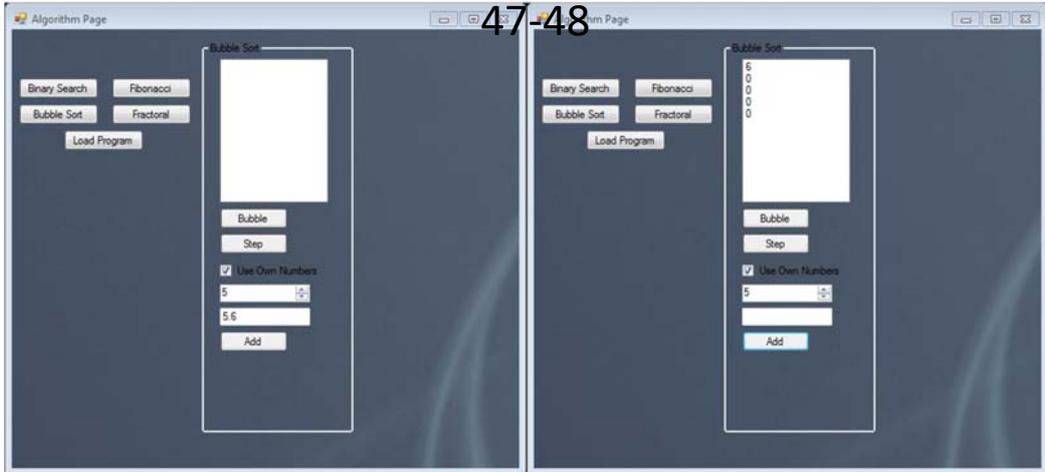
OK

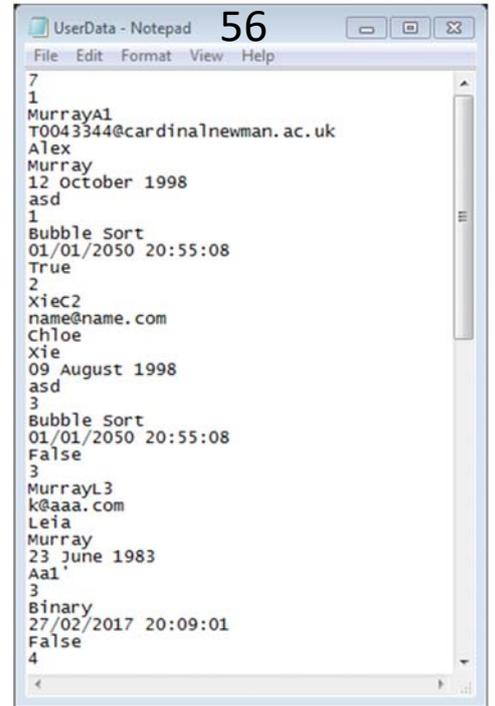
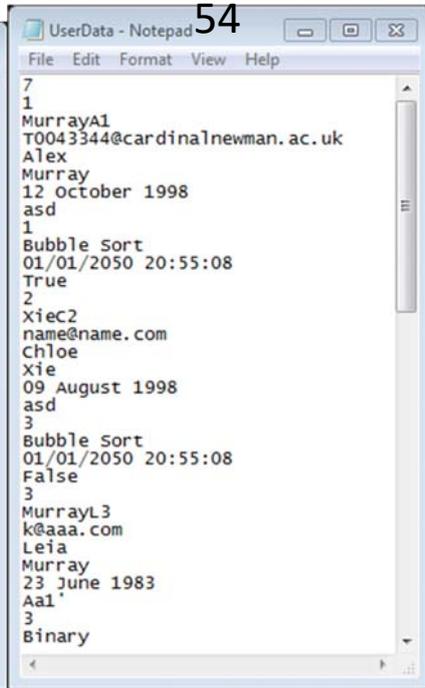
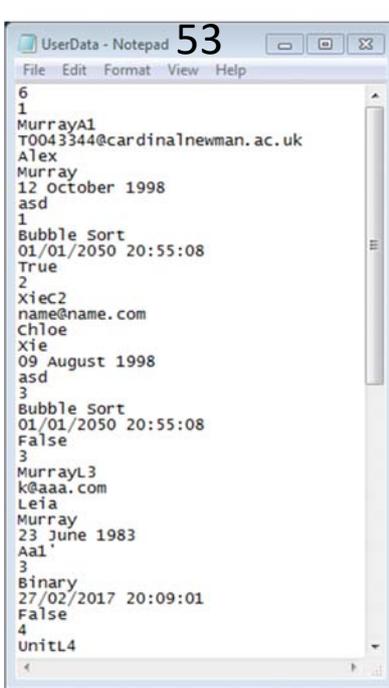


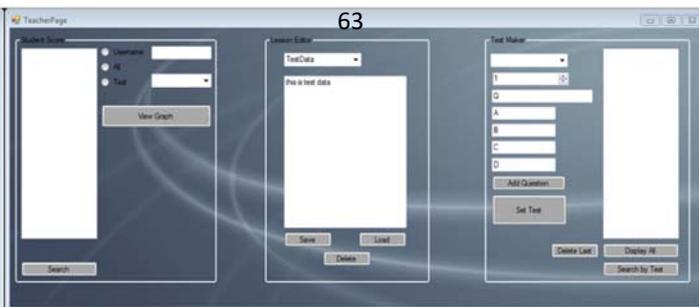
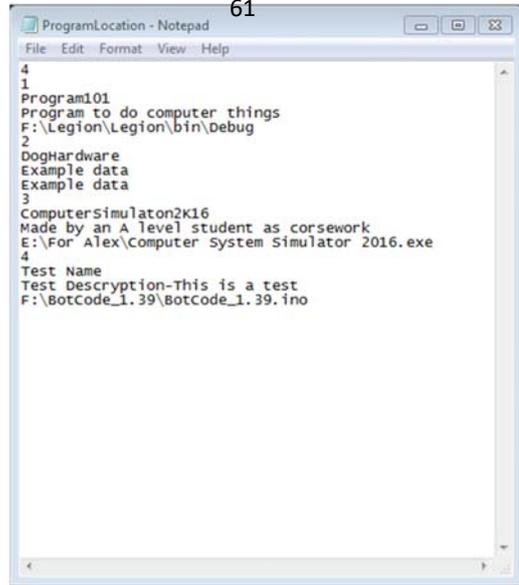
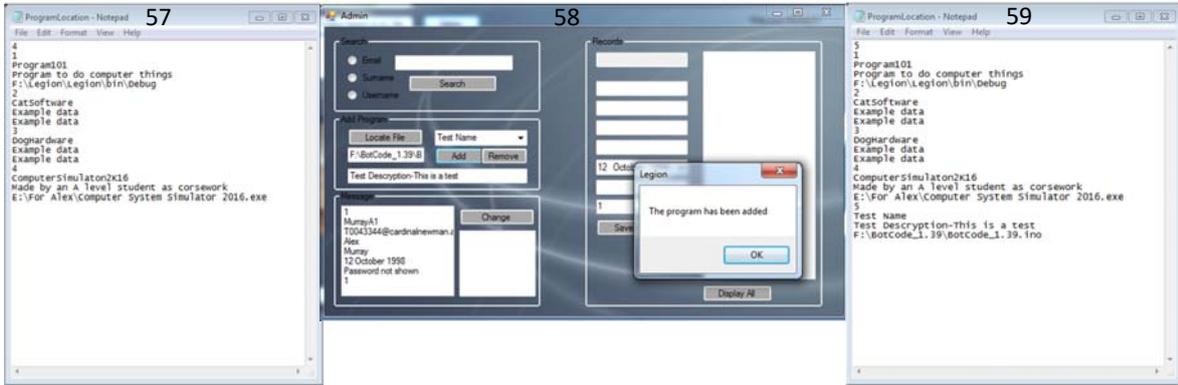




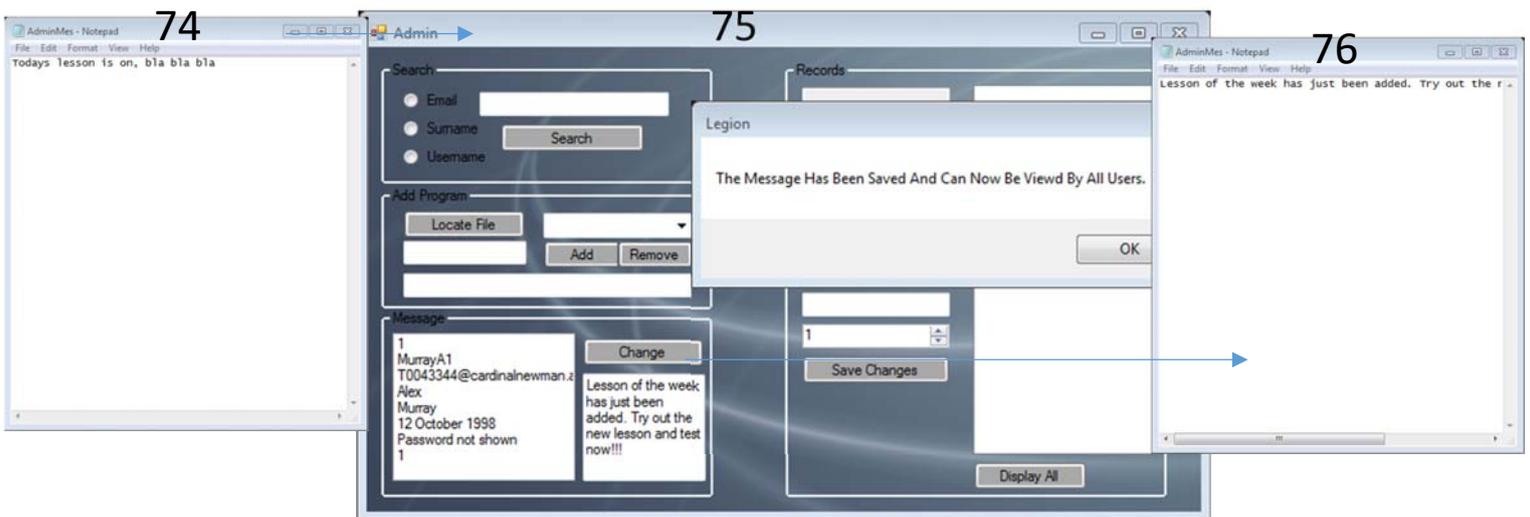
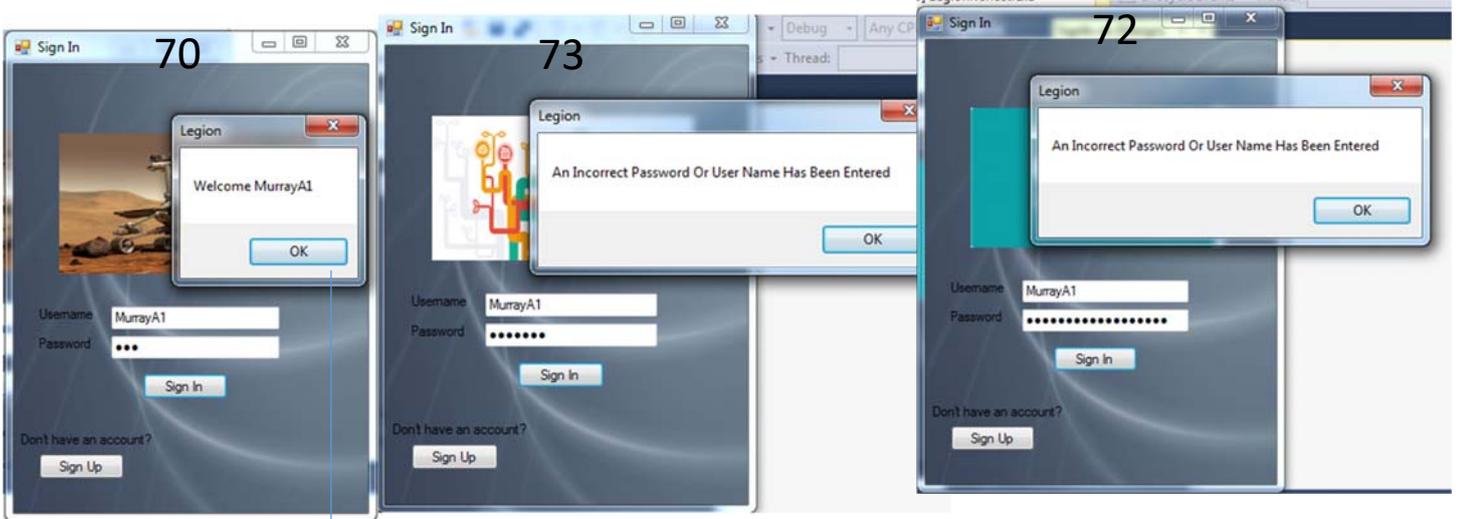


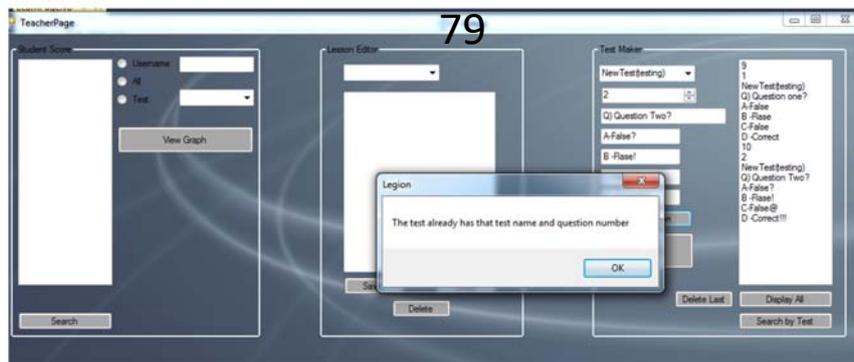
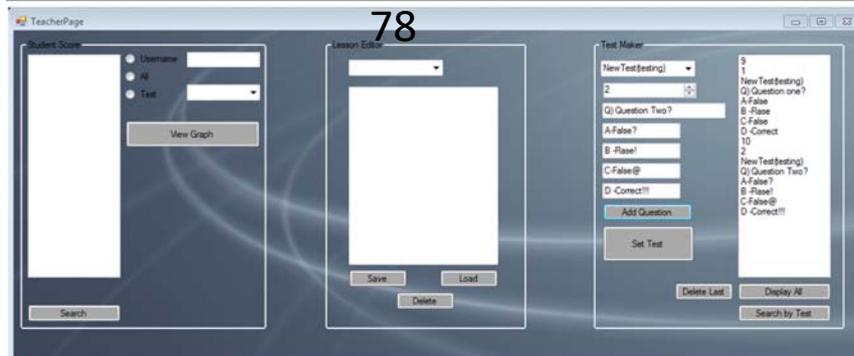
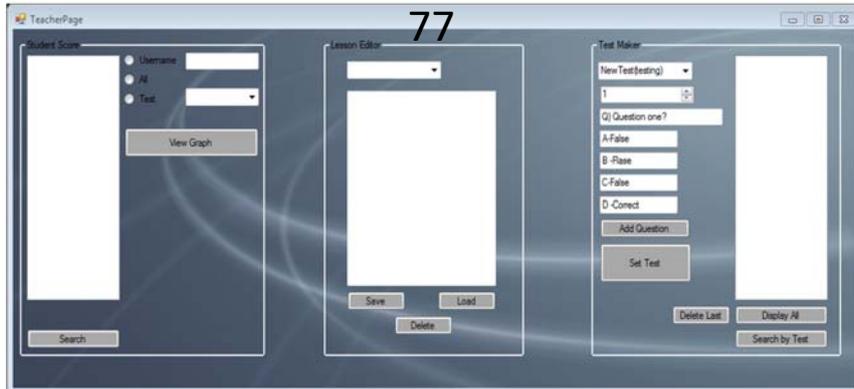








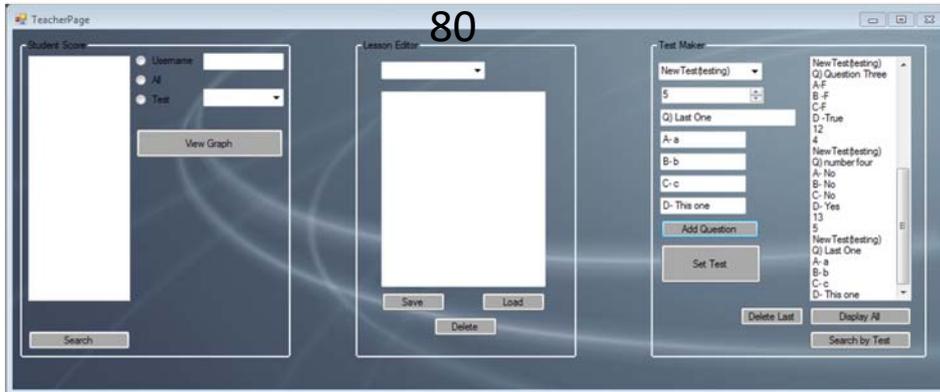


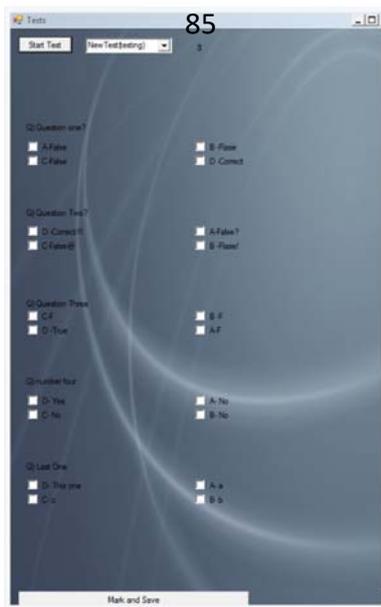
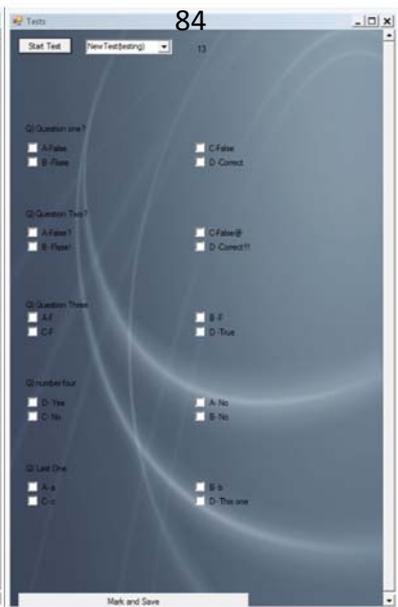
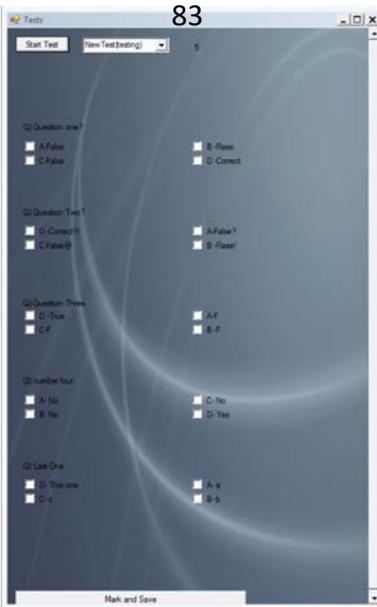
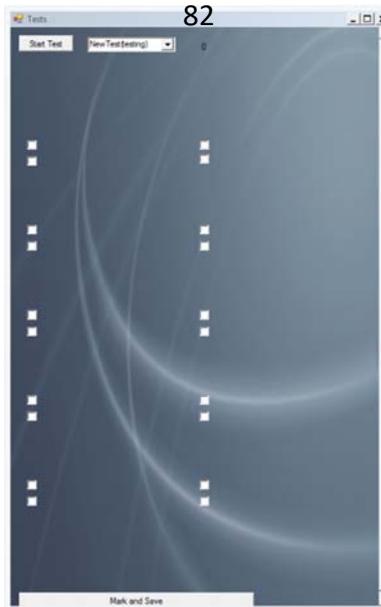


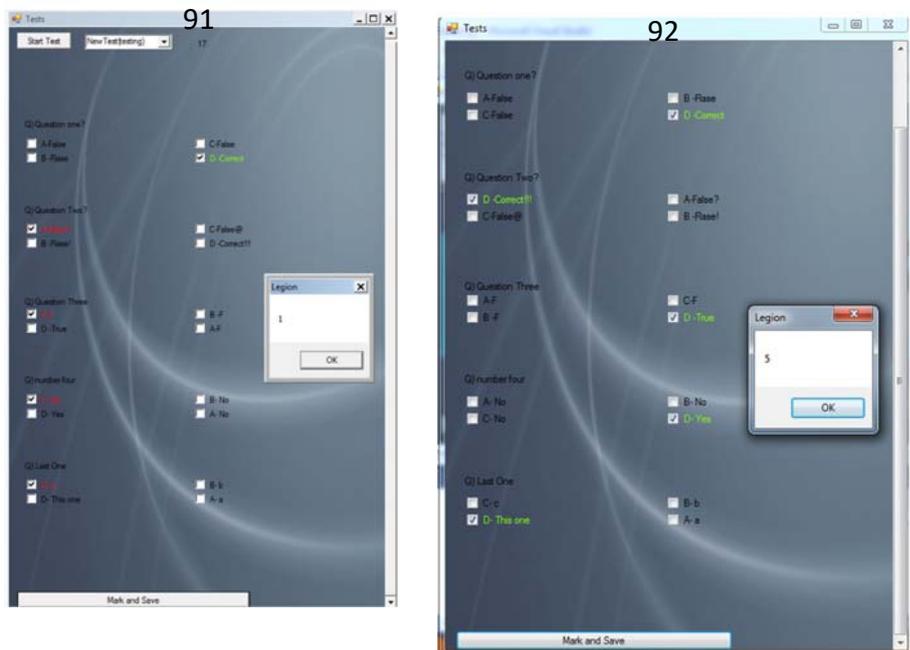
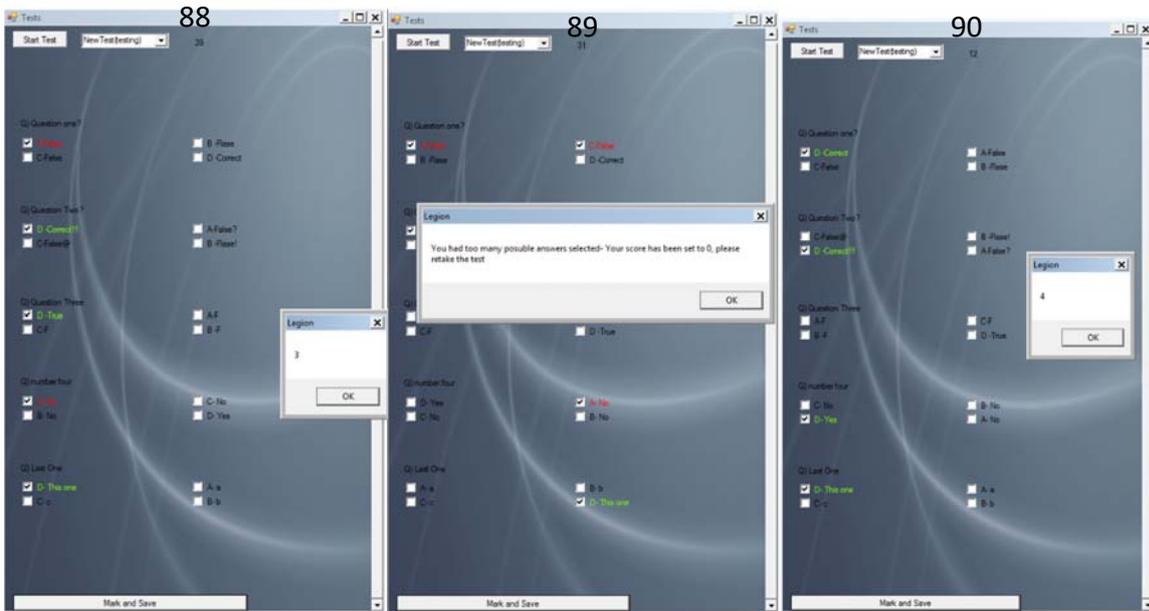
81

```

TestLayout - Notepad
File Edit Format View Help
D -Correct
10
2
NewTest(testing)
Q) Question Two?
A-False?
B -False!
C-False@
D -correct!!!
11
3
NewTest(testing)
Q) question Three
A-F
B -F
C -F
D -True
12
4
NewTest(testing)
Q) number Four
A- NO
B- NO
C- NO
D- Yes
13
5
NewTest(testing)
Q) Last one
A- a
B- b
C- C
D- This one
  
```



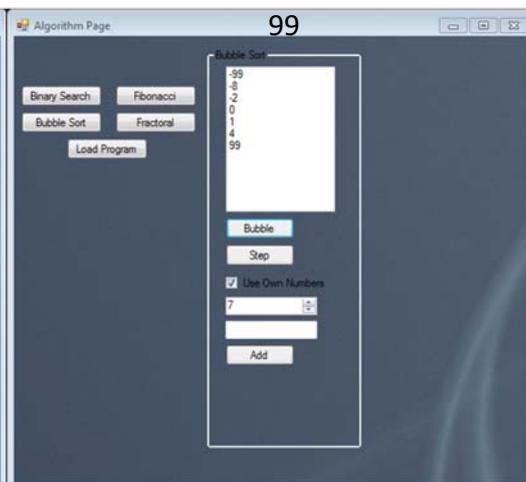
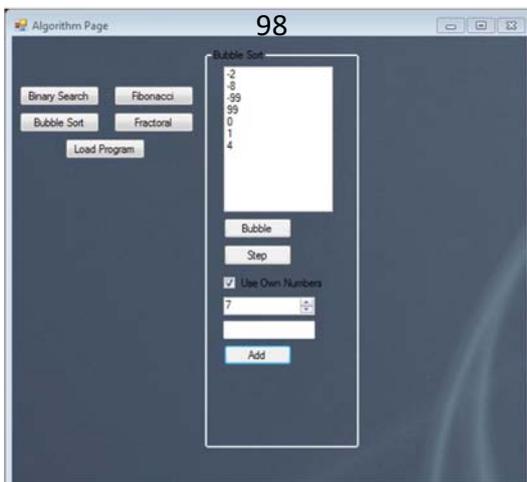
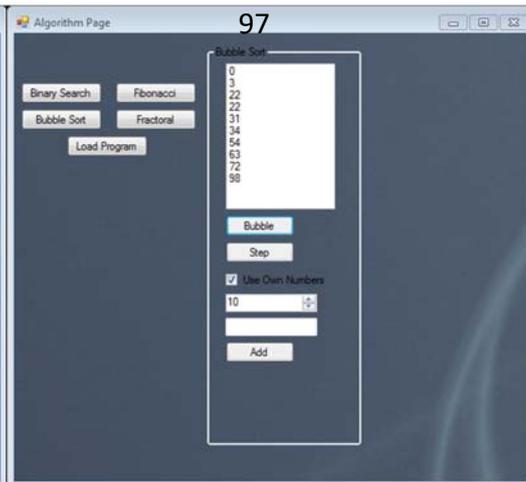
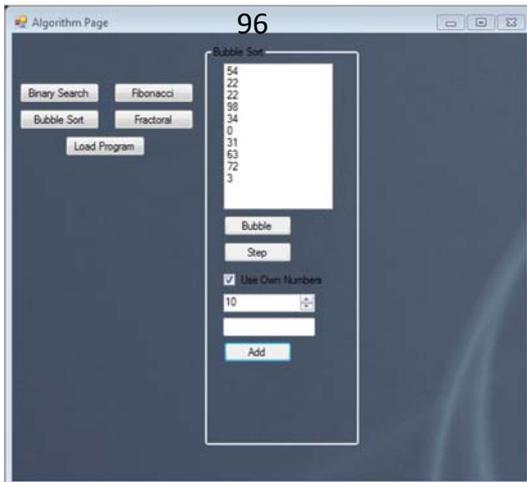
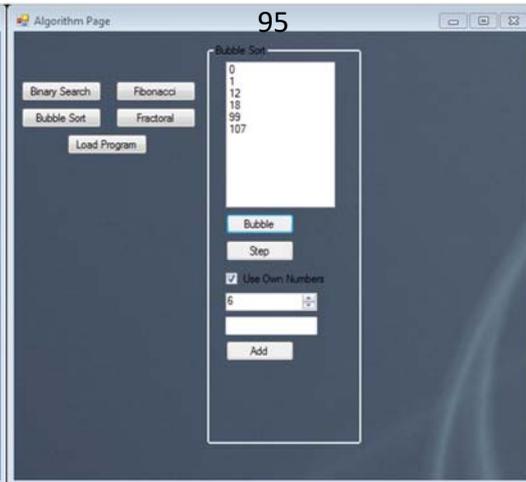
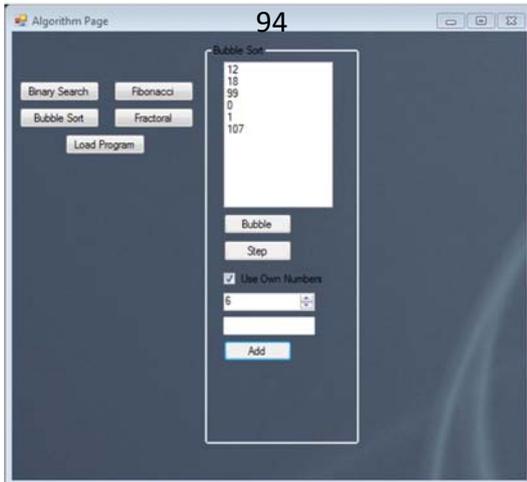




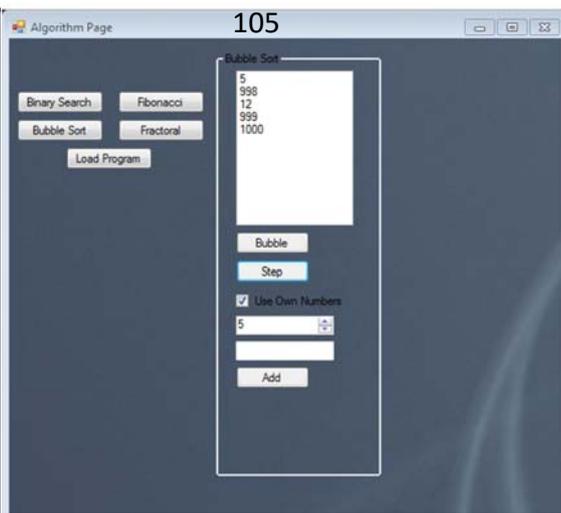
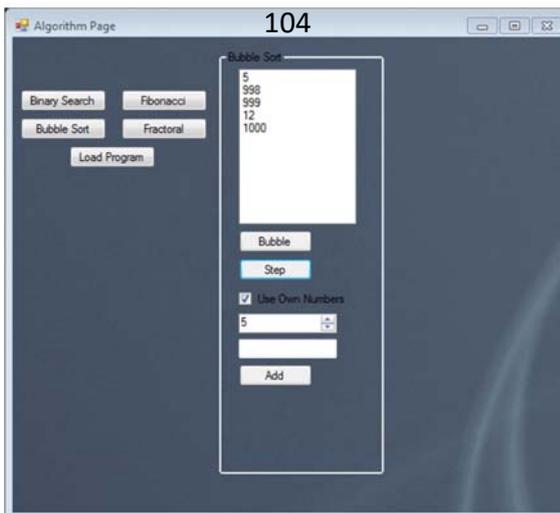
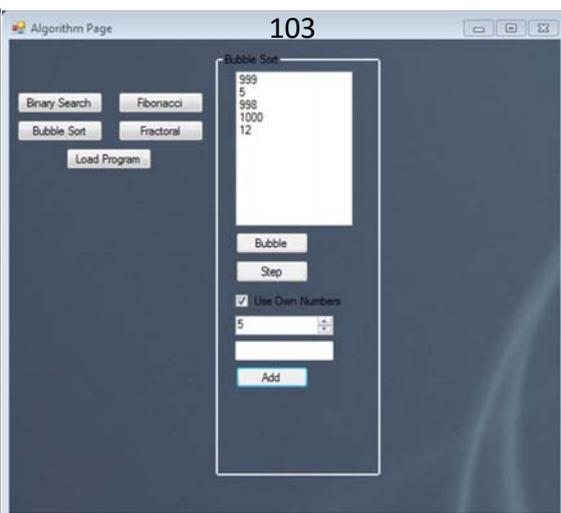
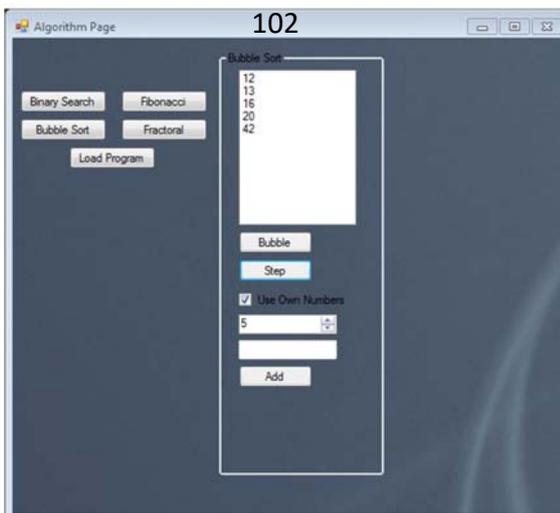
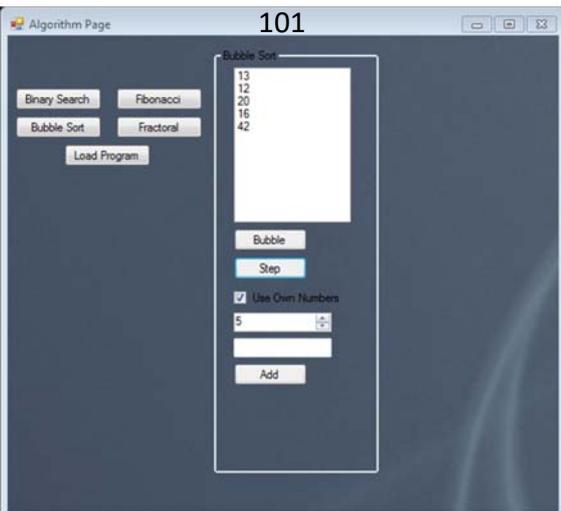
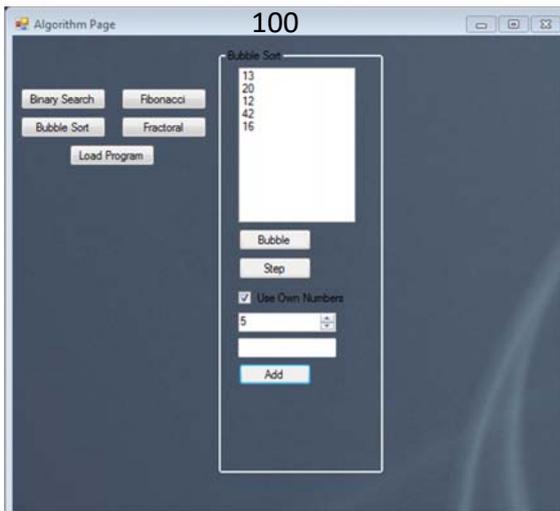
93

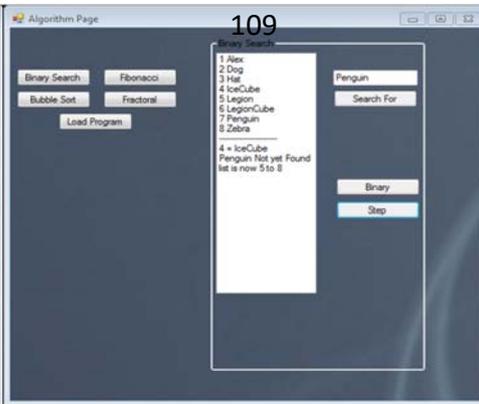
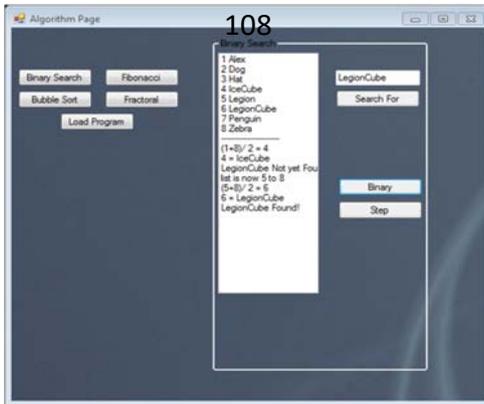
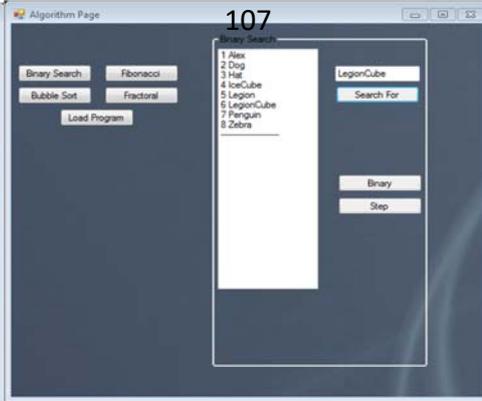
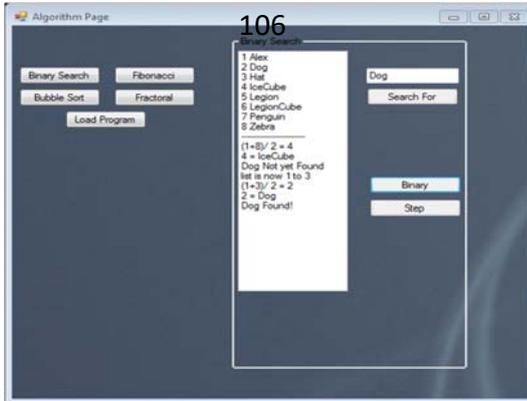
```

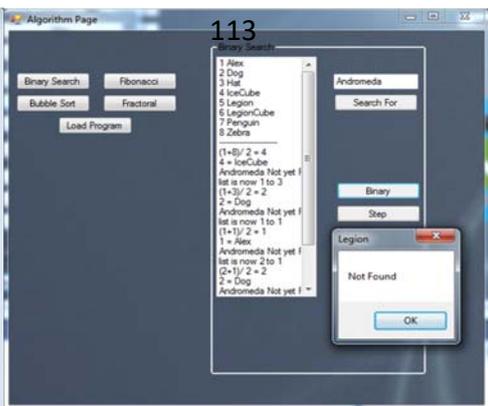
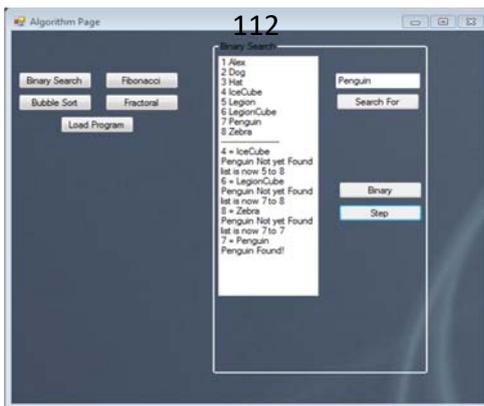
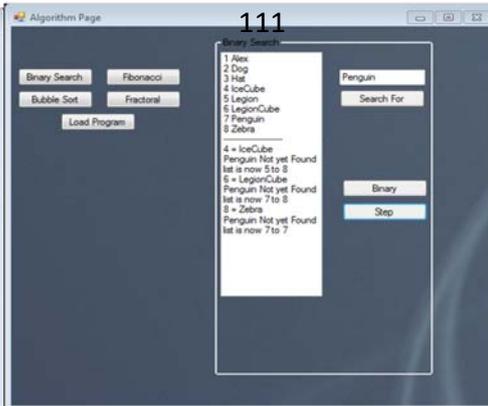
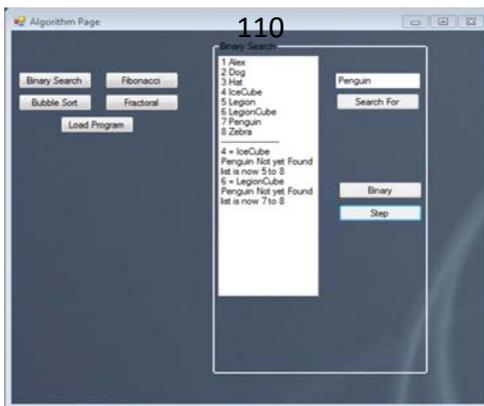
TestScore - Notepad
File Edit Format View Help
5
10
2
3
Bubble sort
Cat
5
10
50
4
Binary
Frank
5
10
59
5
Bubble sort
Eveee
5
10
78
6
Binary
Tod
5
26
50
7
NewTest(testing)
MurrayA1
5
21
24
  
```

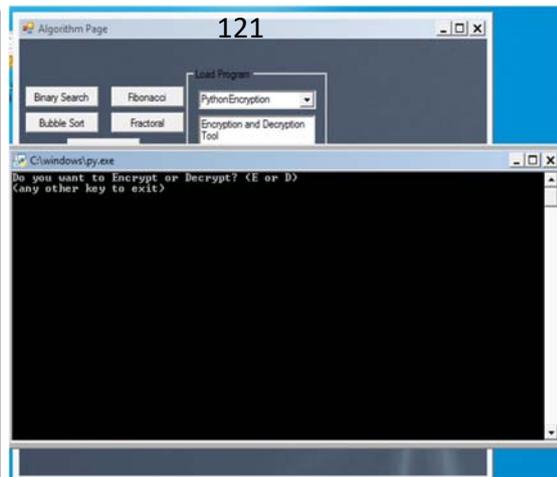
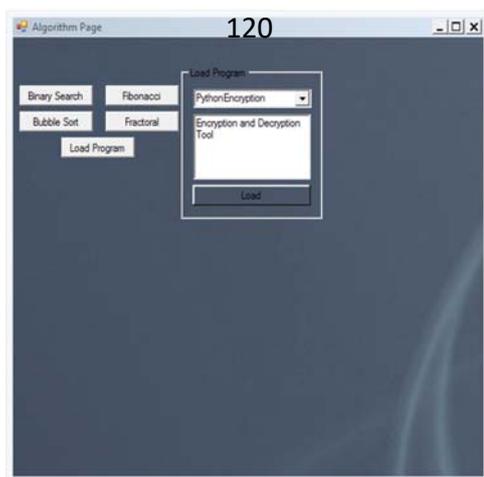
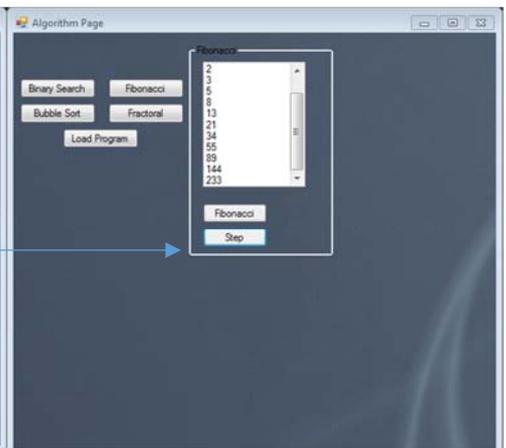
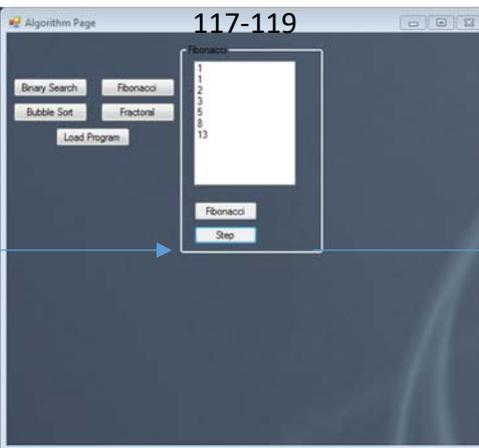
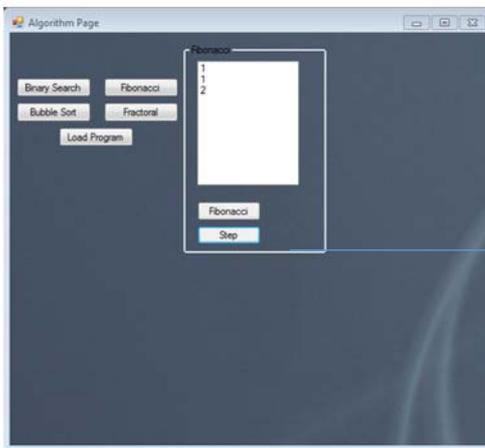
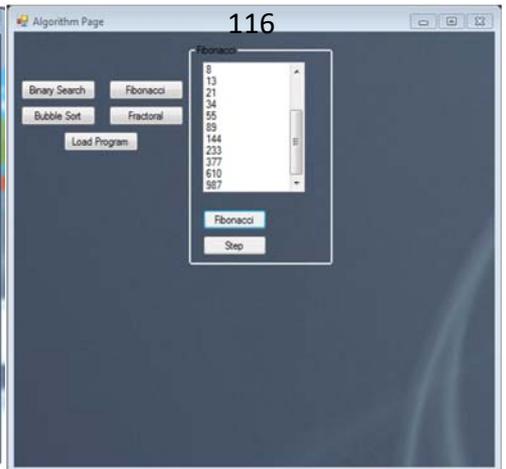
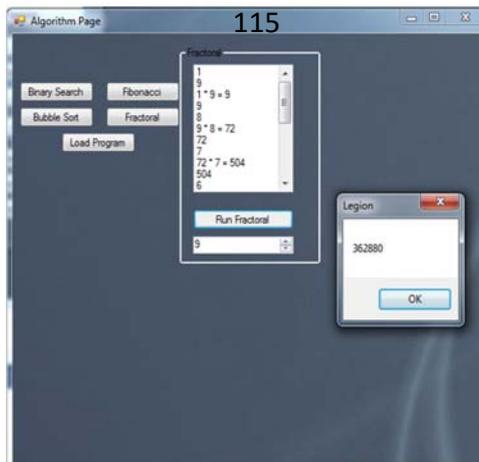
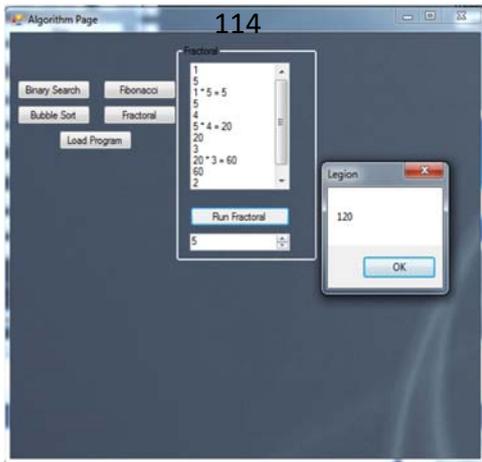












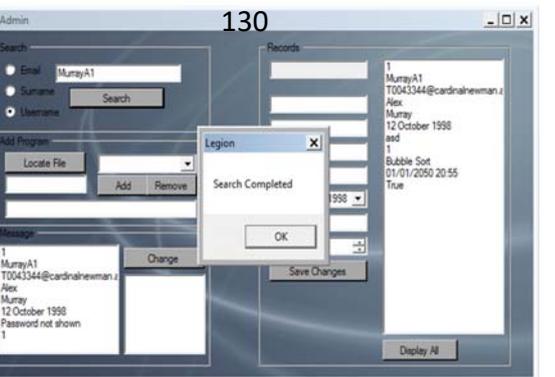
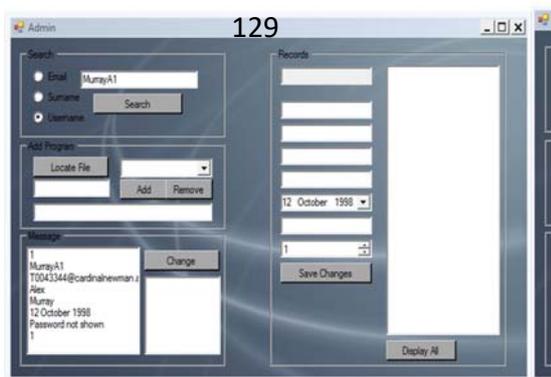
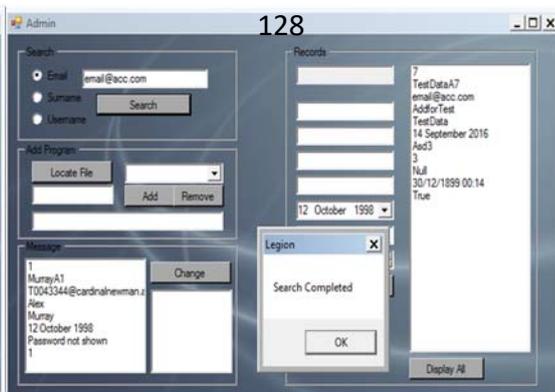
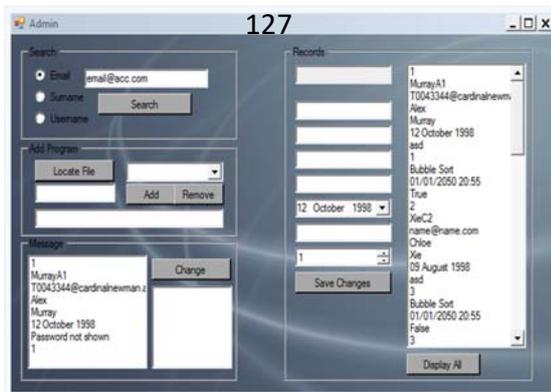
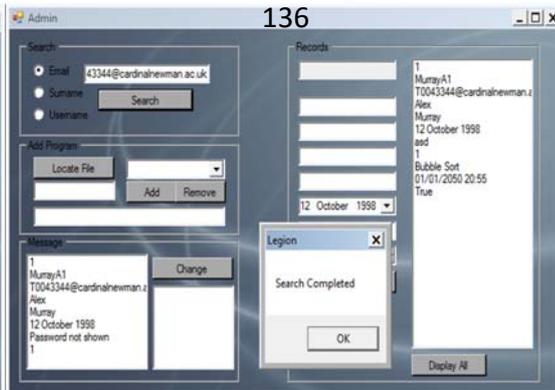
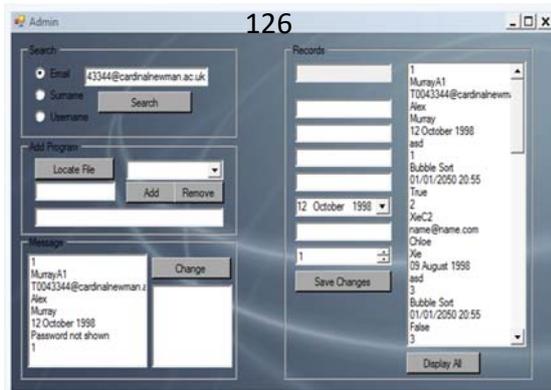
```
C:\windows\py.exe 122
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
E
what is your message? Hello World
what is your keyword? Hat
Your coded message is ... = pfftp eplte
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
```

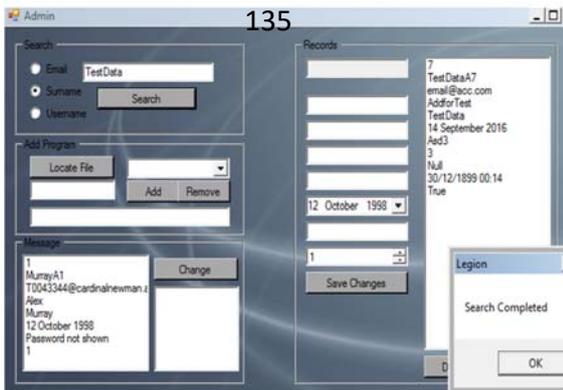
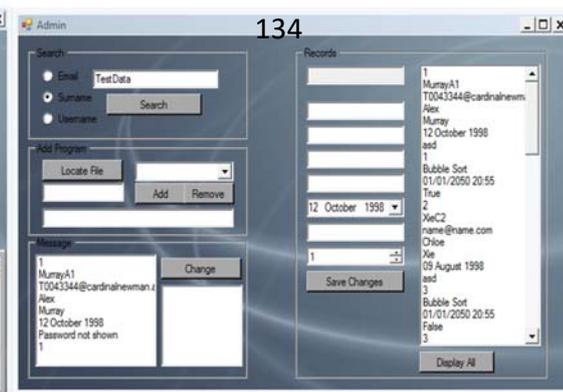
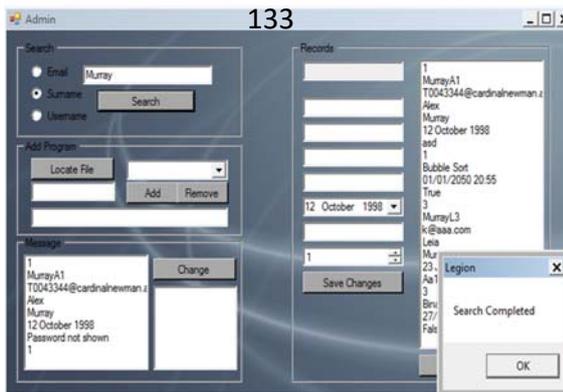
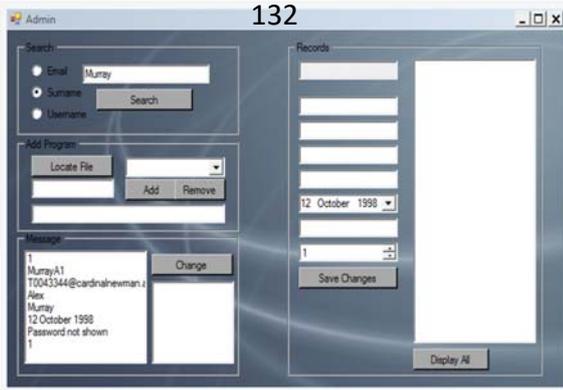
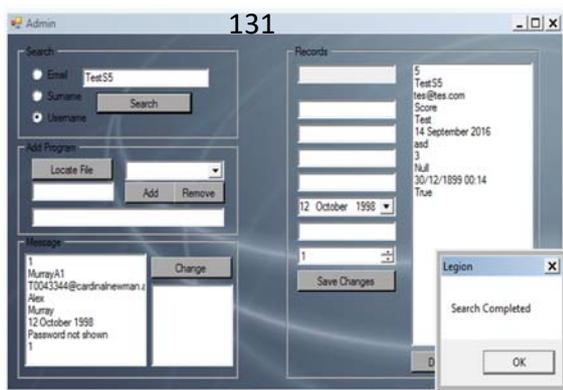
```
C:\windows\py.exe 123
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
E
what is your message? Hello World
what is your keyword? Hat
Your coded message is ... = pfftp eplte
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
D
what is the encrypted message? pfftp eplte
what was the keyword? hat
Your decoded message is ... = hello world
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
```

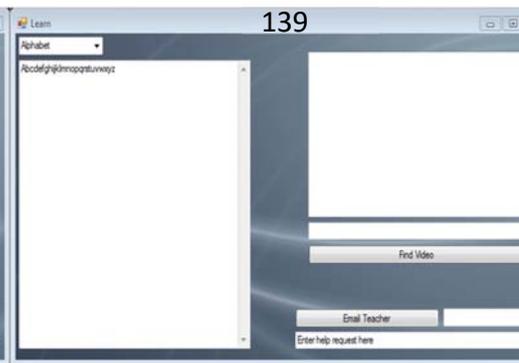
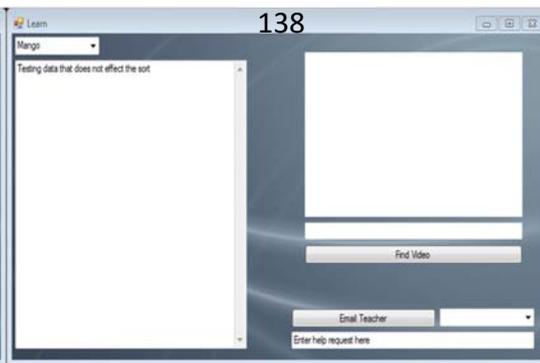
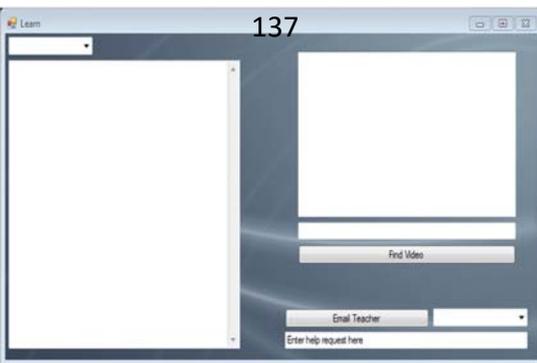
```
C:\windows\py.exe 124
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
E
what is your message? hat
what is your keyword? c
Your coded message is ... = kdv
Do you want to Encrypt or Decrypt? (E or D)
(any other key to exit)
```

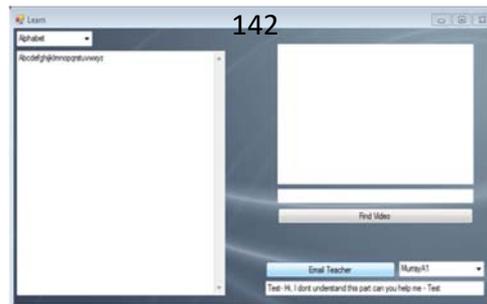
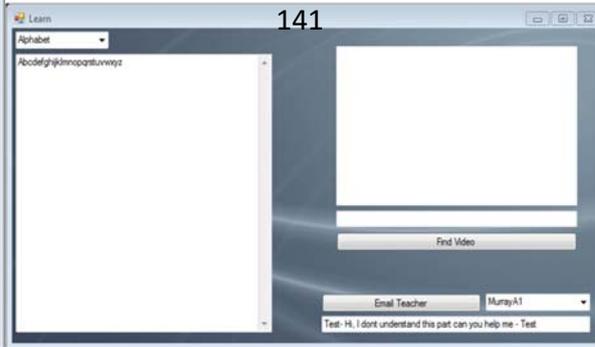
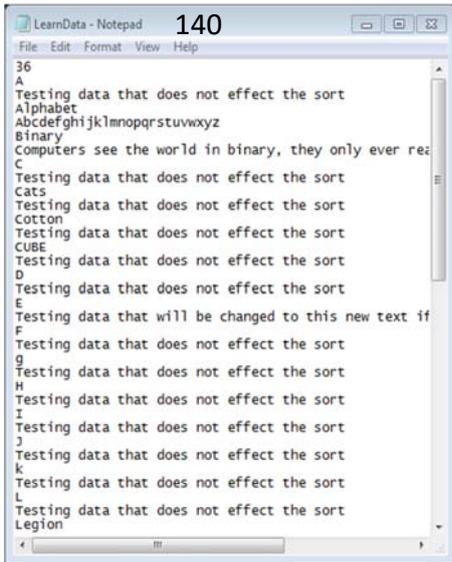


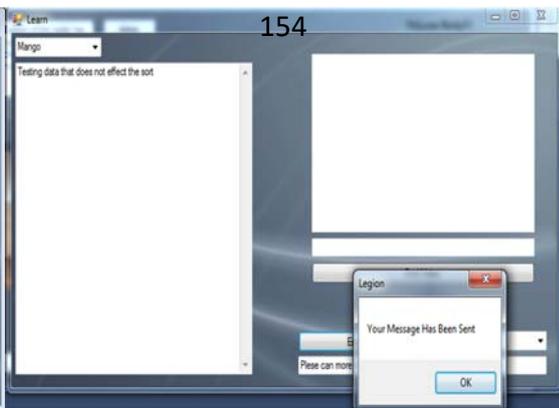
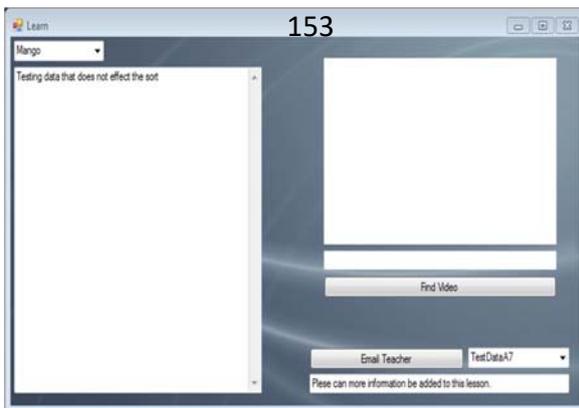
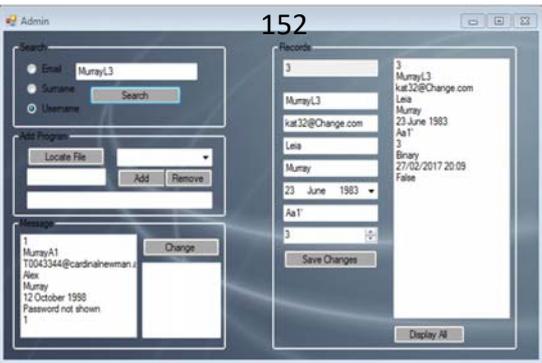
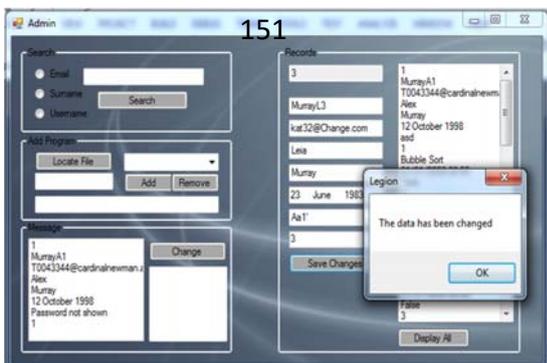
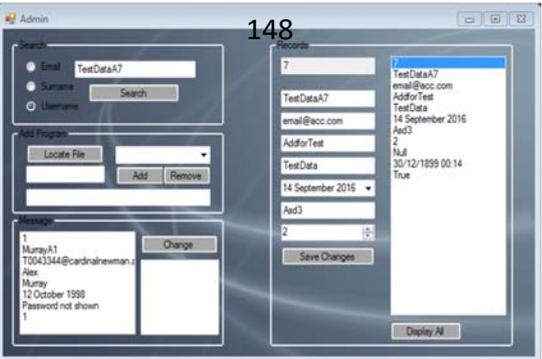
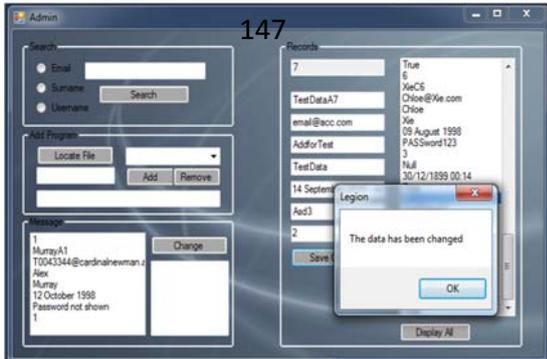
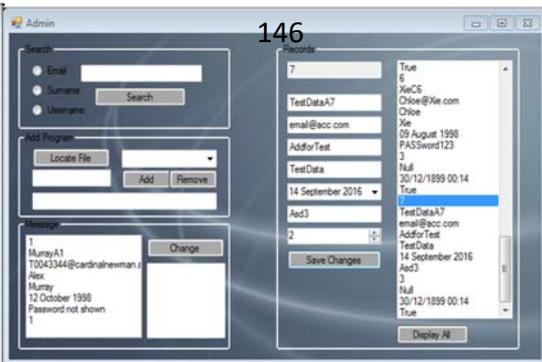
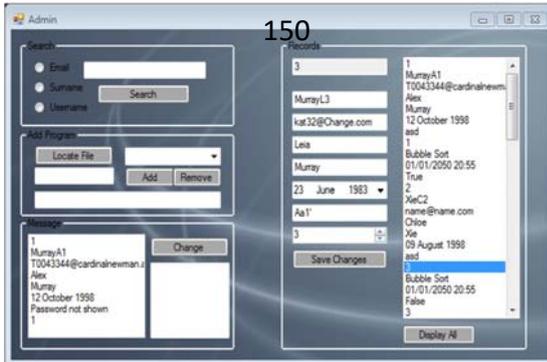
```
C:\windows\py.exe
Do you want to Encrypt or Decrypt? <E or D>
(any other key to exit)
D
what is the encrypted message? kdv
what was the keyword? c
Your decoded message is ... = hat
Do you want to Encrypt or Decrypt? <E or D>
(any other key to exit)
```













Murray, Alexander Thomas (T0043344)

Test: Mango : Message: Plese can more information be added to this lesson. Sent From : Mur...

Yesterday, 21

### Student Question



Murray, Alexander Thomas (T0043344)

Yesterday, 21:02

Murray, Alexander Thomas (T0043344) ✕

Reply all |

Test: Alphabet : Message: Test- Hi, I dont understand this part can you help me - Test Sent From : MurrayA1



Murray, Alexander Thomas (T0043344)

Yesterday, 21:27

Murray, Alexander Thomas (T0043344) ✕

Reply all |

Record Changed: 3. This was changed by: blablabla. Record details before change: ID=MurrayL3  
Emial=k@aaa.com FirstName=Leia SecondName=Murray DateOfBirth=23 June 1983 Password=Aa1'  
AccesLevel=3



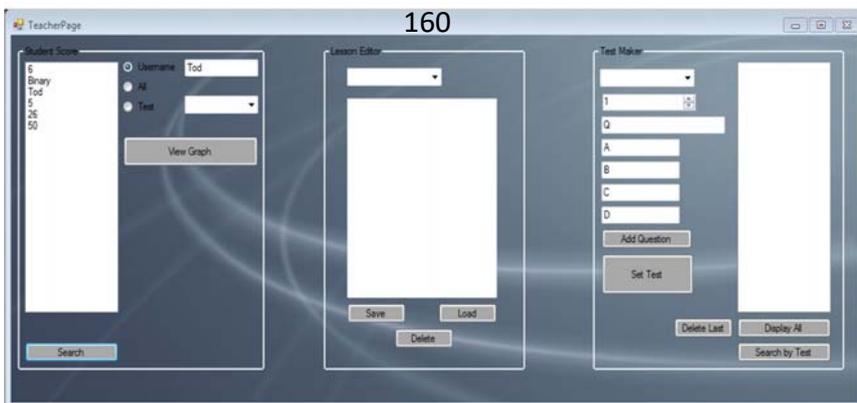
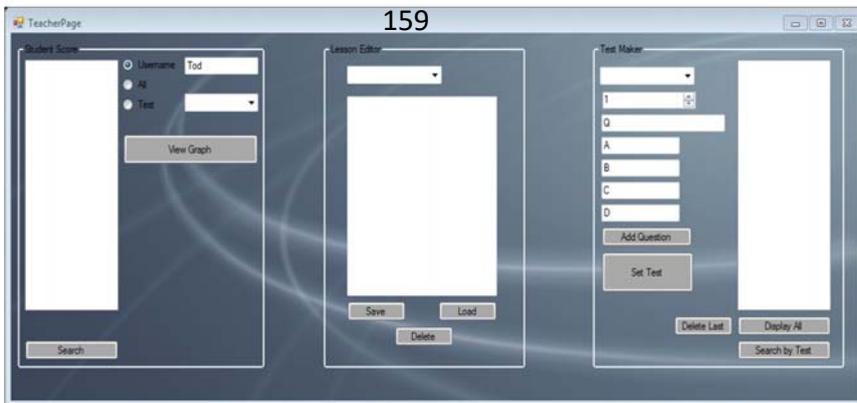
Murray, Alexander Thomas (T0043344)

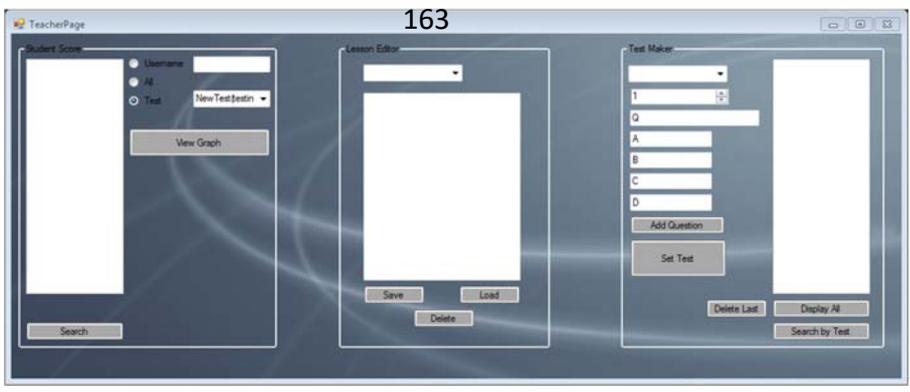
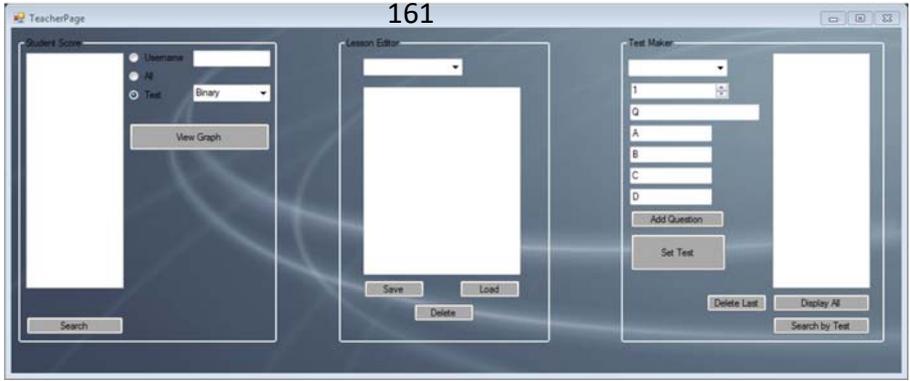
Yesterday, 21:05

Murray, Alexander Thomas (T0043344) ✕

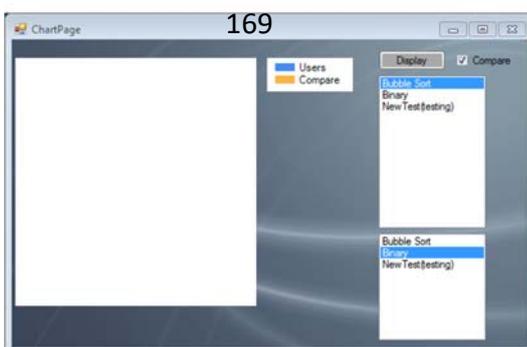
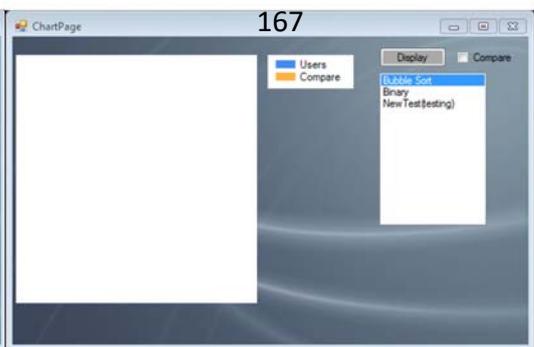
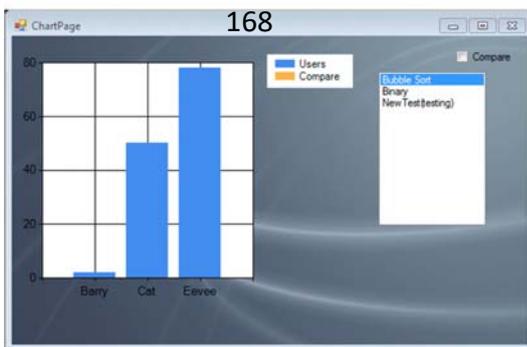
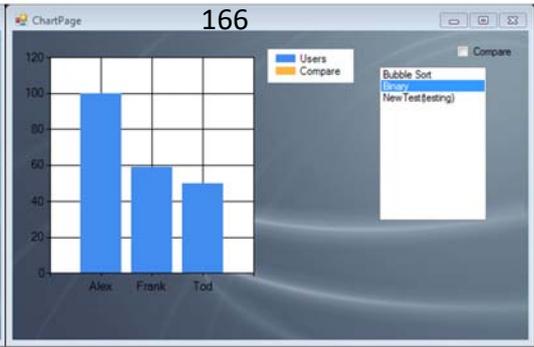
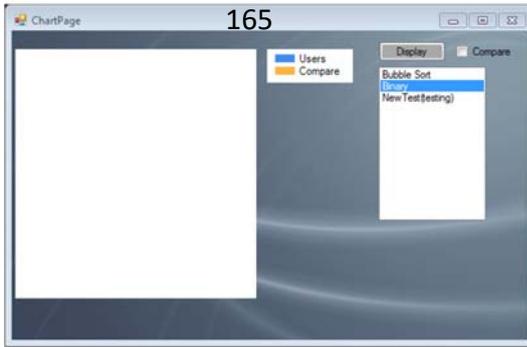
Reply all |

Record Changed: 7. This was changed by: blablabla. Record details before change: ID=TestDataA7  
Emial=email@acc.com FirstName=AddforTest SecondName=TestData DateOfBirth=14 September  
2016 Password=Asd3 AccesLevel=3







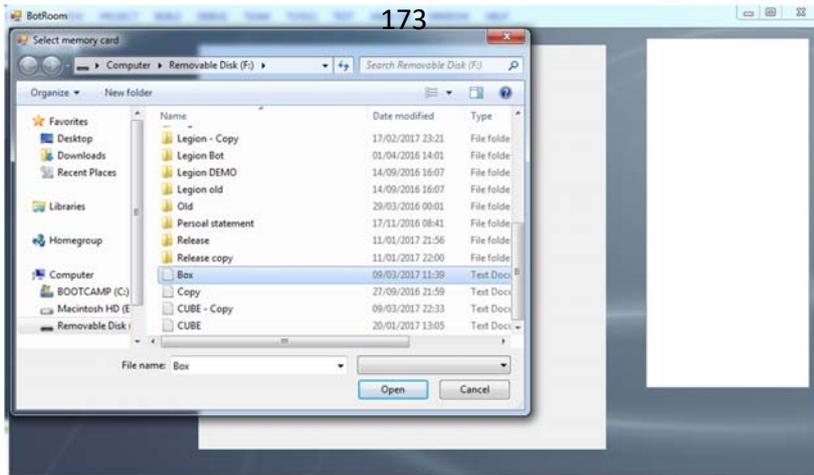


171

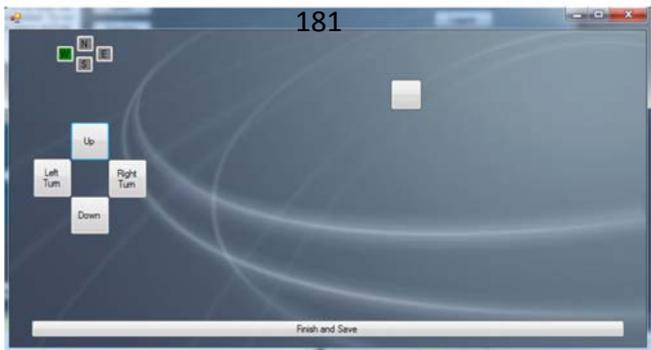


172











## Evaluation

### **Programming language:**

For my main program I decided to use Visual basics. I chose visual basics as its an object oriented language, therefore it fit my needs to make a system that humans could easily interact with. For example, when the user presses a button this then sets off a chain of events for system to carry out (mainly behind the scenes) with an end results given from the user interacting with the 'object' or button.

While my programming language was Visual Basics (VB), I used the visual studio development environment (IDE) this also further my programming needs, as it had lots of useful features and tools that helped speed up development of my program. For example, when I occurred a problem with my code and I was unsure why It had crashed, I the debugging tools given by visual studio to find the problem. I was able to place break points in my code, which allowed me to stop execution of the code on one line and then check the variables at the point in time, allowing me to see what was happening to them before the program crashed. This was very useful to me as a programmer as it sped up the time it would take to manually test the system and variables on paper. This confirms to me that using the visual studio IDE was the correct decision for me. As it gave me access to help for the language and debugging tools, as well as a compiler and a form designer.

My choice of using VB as the language was also justified, as my main program for my system is constructed form window based forms. The IDE allowed me to drag and drop in buttons and objects that are to be interacted with by the user, the code for theses objects were automatically generated when I dropped in the object to the form, having this code automatically generated for me saved me a huge amount of time and effort, that I would have had to do making the objects myself, all I had to do one I had the object was give it instructions and bounds to deal with upon being interacted with. This meant I could set up forms quickly and edit them even faster. It gave me a stable base code that held the forms and objects, that I was able to develop on and use to my advantage.

Some of the inbuilt functions that visual basics offers were also able to help me with my programming, for example the file reading function (StreamReader) that could be used once I had imported the 'system I.O' library, allowed me to open up files and read the contents of the file, for my system this plays a very crucial part of how the system stores its data. Without this feature being offered I would have had to find another way of reading data from a file. While there are others this was a very quick and easy way to do so. As almost all my forms on the main program need to be able to read and write data this helped me more than I realised at the time of writing the code.

After I had got the data from the files I needed a way of displaying it for the user, I had considered using a more graphics friendly system for this, such as python's new Turtle graphics, as it's a surprisingly powerful system. I had a think about it, but in the end I'm glad I chose VB, as the python's graphics system was powerful, but also hard to learn and change to my needs. Plus, it would mean either having to switch data between the VB system and the python graphics, or use python to open up and files where the data was stored, thus solving the same problem in a different way wasting time and memory in the system. As VB already had a chart system, that allowed me to use an object as a chart and populate it with data as and when needed. This was very easy to use and was wonderful at generating simple charts to show the students scores. VB has a lot of useful features like that, that help to save time when developing code.

One drawback/problem that I found when using VB however was its graphical limitations, as useful as it was to be able to populate data into charts, that was more or less the best of the languages capabilities when it came to any form graphics, above populating a chart the language becomes hard to use. I realised this when programming the 'show path' mode of the bot. in order to draw one line in the path, the language requited me to make points (coordinates) for where I wanted the line to end and start. This wouldn't be too big an issue for drawing functions such as  $f(x)=2x/7$  I but the coordinates system was just not ideal for my purposes of drawing the path the of something that has to constantly change direction that is random to the system, I think for this part, I would have been better using a more graphic heavy language for this part, such as something like unity, where I could use the 3D objects and move them around to help visualise the bot, or I could use python's new graphic system. Both would take some time to learn but could potentially have given a better result for the bot's path. That being said I was still able to use VB to get the bots path, the language was able to draw the line to my specification, it was just time consuming and not efficient to program.

In all I was happy with my choice to use Visual Basics as my language for my project. Its combination of IDE tools, code generation and ease of use of the language allowed me to make a fully working system from the ground up. The level of complexly and integration with items outside the program (such as charts and file handling) could not be preformed as easily on another language, such as python. Visual basics was the correct language to use for my main program.

For the bot, I did not use Visual Basics, I programmed the Arduino board using an Arduino sketch, this is a language similar to C/C+ The choice to use this language was more swayed by the compatibility of the ease to upload the Arduino code to the board form the IDE.

The Arduino language was very useful when it came to programming the bot, as it gave me the freedom to set the pins on the board to an input or output based on how I wanted to wire the bot up, this gave me a lot of freedom with how I added parts such as sensors to the bot as I didn't have to make sure I had an input pin available, I could just program it to be an input.

The language had other features that helped save me some time, for example when accessing the file, I used a micro SD card reader, the Arduino library had a function I could import, (SD.h) this allowed me to easily read the data on the SD card without having to find another way to do so, this saved me time in connecting the SD card up and writing to it. Other functions allowed me to open a serial port to the computer when the bot was hooked via cable, this allowed me to run my own real time tests in the board and get a text output on the computer screen.

Although the Arduino IDE did not contain any debugging tools, which was frustrating at times, it did come with a compiler that compiled the written program, converted it into assembly code then loaded it into the Arduino. This is a lot more helpful than the rival language I almost used (python and raspberry pi) The Arduino IDE did send out a message why the code failed to compile, (E.g. ; expected)

The Arduino's freedom to change and map pins along with an extensive library made it very useful for creating a bot, I think it's a very powerful language despite its difficulties when debugging. And it saved me a lot of time as I could upload it straight from the computer I programmed it on.

### **Comparison evaluation:**

Similar system functions:

- Sign in with a google or Facebook account
- Web based
- Way of searching the site
- User friendly visual displays
- Chat forums
- Tests for students
- Hold data on a range of topics
- Progress could be tracked for each lesson

My system could never fully compare up to on of the systems I investigated, this is because those systems are built with big development teams who have more knowledge and skills than I possess in a much wider range of expertise. Although my system will never be as good as theses others, I do think my system can hold its own for what an educational system is.

The main reason my project cannot compare to the bigger systems is due to mine not being web based. All the other systems were web based and this gave them a huge advantage over mine as they can be accessed from any where in the world so long as the user had a stable internet connection. It also reduced compatibility issues as a web based system can run on almost any device, for example mobiles can run them, OSX and Windows devise can use them, it allows or a much bigger audience.

It also allows the login system to be better, as you can login form anywhere, it doesn't matter which decide you made the account on. With my system, each device that holds the program is the only device the user has an account on. Being web bases also allows for other features like 'sign up with google account' something like this would save a lot of time for the users, but without even being a web bases system I would not be able to even consider adding this kind of function.

Chat forums are another reason I don't think my system was a good as others out there. The chats added another reason to use the system, it made it more social and even allowed for further learning as the users could post questions that could then be answered by other students who have a greater knowledge, this could have added a lot of new and extra features to my system as it makes room for improving and user feedback better (the uses can say problems on the chat that the designer can then take in to consideration when changing the system.)

Dispute not having all the features of the big systems that currently exist, mine does have some of the same features, and some new ones. My system may not be web based or have a smart login feature but it does match the other systems with:

- Way of searching the site
- User friendly visual displays
- Tests for students
- Hold data on a range of topics

My system withstood the tests and was able to prove it can meet this criterial that the other systems had, my system isn't perfect but it can hold its own when compared. My system even had other features

that the bigger ones didn't, such as the bot. This was a new feature that none of the other systems had. The bot was even liked by the testers in my system who enjoyed having something physical to work with to see how it worked and how it moved. I believe my system was better at keeping the users' attention due to features like the bot. Possibly even better at keeping the attention than the big systems as this is something new, that the other systems don't offer.

I also think my system did succeed when it came to the holding and sorting the lessons, although my search was less sophisticated (as in mine the user had to click on the item they wanted from a list but other systems used a search engine that worked in real time when the user entered a new letter into the search bar) My searching method was still effective, as the data was sorted when saved, this allowed me to use a binary search on the data that quickly found the correct result regardless of how much data was in the system.

System evaluation:

- *The program should be able to hold information for a range of different topics, with the teacher being able to add new topics as and when they see fit. The data should be accessible by the students.*  
– In my system the teachers can add lessons or topics by typing in text or linking a video. Students can then access this data on a different page. I believe I met these criteria as the teacher can add any topic on any lesson to the system. This is proved by tests 47,49, and 50. Where the data is loaded into the file correctly and the students can read the file having the information displayed to them on the learn page ( all students have access to all lessons from the start.)
- *The program should be interactive. (such as simulations the user can try) Too much of the current system and old system was 'boring' The new system will need to grab the attention of its users and keep it making sure its interesting and fun to use.(not a pain/drag)*  
– I believe I have met this criteria, but this is a subjective matter as the view what is 'fun' changes from person to person, therefore I implemented the ability to add more simulations/programs as time goes on. This way it can always have new content added to it. However, I did also make my own simulation and add in a video function (for if it was appropriate.) I believe this with the bot made my system a lot more interactive and engaging than the old classroom textbook. This is backed up by the feedback I got in my questioners. I had a positive feedback regarding how entertaining the bot was and how much the students had enjoyed the given simulations.
- *The system must be easy to use. As some of the users may not be experienced with computers, or this type of system. It should be easy to pick up and navigate the system, as well as intuitive.*

- Unlike some of the other criteria I

believe I neglected this one a little, I made the original system without much thought into how the users would navigate, I did however take this more seriously once I received feedback from the prototype, there was a negative response to how well navigation of the system went, because of this I did then correct my errors and make it easier to navigate the system with a menu.

- *A way of testing the student's knowledge will be multiple choice test, these tests need to be able to be created by the teachers and taken by the students, with the results of the test saved. The location of the answers to the test should also be randomized each time to prevent cheating.*
  - The teachers have the ability to make tests for students, these tests can be later taken by students. When the student loads a test the location of the answer is randomized each time a student loads up a test, this prevents the cheating from simply selecting the same box as another. This is proved in tests 55 to 65. I even made it so teachers could edit tests and remove them if they wanted to, therefore I believe I have defiantly met this objective
- *Topic lessons should load up relatively quickly when requested. The number of topics should not drastically increase the time to load up one of the topics.*
  - Tests 91-94 prove that the data is loaded up quickly when a large number of topics were used. This shows that the topics load quickly regardless of the amount. I ensured I would meet this criteria by building in both a sorting algorithm and binary search algorithm to the topics section, this meant I could quickly search through a very large file relatively quickly.
- *Admins and teachers will have special access that the students don't. They will be able to check and change data stored by the program as and when necessary.*
  - I think I met this criteria, as I ensured admins had access to all of the system, every part. They were the only authority on the system that could access all the forms and pages. Teachers did also have special access to other important pages too but not full access. I believe I met this part of the criteria well as I received positive feedback from the admins when I asked if they enjoyed the loading of data into fields to help them edit it later. The ability to change data stored in the program can also be shown by tests 50, 54, 49, 47, 46, 45. With the feedback and the tests I believe the users were happy with the system, and the system worked for this one.
- *Sensitive/private data stored in the program must be kept safe and secure, so no unauthorized personal can access it.*
  - In order to keep personal data, secure I made sure to use a very strong form of encryption, I had to research and spent a lot of time adding DES encryption in order to properly secure the users data. Once the data has been encrypted it can not be read until the decryption process is used. Even if someone gets into the text file without the key the data they can read would be useless. Tests 41, 42, 43, 44. This shows that the data is secure. I fully believe that no unauthorized user can access this data.
- *New users should be able to easily join without hassle. Already existing users should be able to login easily, using a secure password and username.*
  - Users of the system can login quickly as they only need a username and password. New users can create an account just by entering their information, no need for email verification or waiting for an admin to conform the account. This is shown in tests 52, and 53 prove this. However, I do realize that there are quicker ways a user could login. E.g. a 'remind me' feature, that saves passwords, this could make it easier for the user to login. But I am happy with the current speed it takes to login, and a remind me feature produces security risks.

- *Test will be able taken by the students, and then marked automatically. The teachers should be able to view the results from the individual tests.* – The teachers have the ability to view the students score and test details, feature was originally limited, but after the prototype feedback I revisited this and improved it, making it so a teacher can see the students scores in a chart. This was added due to feedback therefor although it wasn't in my original criteria, I still believe its better to add in this feature for the teachers. The teacher can use it to view the students scores, this is shown by tests 103, 104, 99, 100, 101, 102

I believe that my system does meet the criteria it was set out to preform. It does have its various strengths and weakness, for example navigation of the system was (to a minor extent) still is lacking on meting the mark. But I believe this is made up for in some of the areas of the system, such as the enjoyment and educational factor, I made sure to keep my system interesting for the user through as much of it as I could to ensure users were engaged with it a didn't just see it as a virtual text book. The system can be interacted with, that's what it was designed for. To be more than just text the user reads. My system has three main users, and each of them has different needs and requirements from it. Although I have done my best to unsure I fulfill all these different needs I know I haven't fulfilled them 100% as the feedback wasn't fully positive.

I understand some of my users disliked the user interface, this was shown in the beta testing. This shows to me that in order to really fulfill the needs of the user I would have to bring in a designer, someone who could with with me as part of a team to improve the UI. Navigation and ease of user for first time users seem to be my systems downfall that (despite changes) I could not bring up to the standards required.

But the strength in the system was shown once they were shown how to use it, admins liked how the data fields self populated. Loading times were reduced by using the correct sorting and searching methods, data in the system flowed well. And what I think if the main strength of my program is the fact it wont become outdated easily, as teachers and admins can add in new lessons, topics, tests and simulations. There is no need to go back into development to add in new information to the system.

Self evaluation:

Over this project I have learnt some very valuable lessons about how I personally cope with projects as this as the first major project I have worked on. I found I have some very valuable strengths and some costly weaknesses.

Time management was a key issue for me. Over the project I wasted valuable time at the start working on side ideas that had nothing to do with the criteria and didn't even end up making the cut for the final system in the end. Later on in the project I spent far too long on parts of the system, work did need to be done, but I was making very slow progress, this came back to haunt me later as It dug into time towards the end of the project that I had set aside. It had a huge knock on effect that I didn't expect. And certainty didn't account for. I also underestimated how long certain parts of the project would take

to complete. This made my time at the end for documentation and testing shorter than I would have preferred. In hindsight the I should have sat down at the begging of the project and made a time scale of roughly how long each part should take, I should have then set aside relative big gaps in-between them to allow room for falling behind. If I had stuck so a schedule like that it would have drastically reduced the pressure and given me more time to check over all my work, maybe even add some more features to the system.

One of my strengths was being able to find out new things on my own that I thought would help me, I spent a decent amount of time online finding some sort of solution to a problem, and if I could I would use the solution and adapt it to fit my purposes, this really helped when I was running out of pins on the Arduino board, I was worrying I might have to call it quits for adding an SD card, when I saw an Arduino hooked up to a bread board instead of directly to the motor driver for a model tank, this gave me the idea to use the breadboard for the SD card, and Switches, that way I could use one the of GND pins for multiple connections instead of just one. Finding similar solutions and seeing different methods of completing the same task really showed me how creative you can be even with a lest list of instructions.

Another weakness had was jumping form part to part of the project, I kept becoming bored of parts, I would then move onto the next promising myself to come back later (and then forgetting) this was especially true for parts of the system I disliked such as the data flow diagrams, I forgot about them and was just ready to move onto he next stage when I remembered about them. More self-control could be used here to stop me from skipping over parts and jumping ahead, as it will only make me fall back behind later.

Another strength was problem solving, when I had an error in my program (which often occurred) I felt I was very good and capable at finding out what the issue was and resolving it. This didn't matter which language I was working on or which IDE I was working in. For example, if I was in visual studio I would use the break points to find exactly what caused the problem looking at the variables as I went, this normally would show a logical error I had implemented by mistake. But where there was no break point available I had to get creative, I used the serial connection on the bot to allow me to flag up any errors in my bot's code, or connections, like the switch test I had to hook it up to the computer and fiddle around with the bot until I found the message I was expecting on screen, and therefore found the root of the problem.

Remembering to document my work became a clear weakness I didn't know I had until I looked back on my work. In excitement of getting a new piece of code to work or in getting the bot to move in the intended way I would forget to document the work I had done, this includes commenting code, making adjustments to the forms and writing about them, anything that didn't affect the code or performance would slip my mind until I had to go back and retrace my steps.

Change of approach:

If I was to restart the project using the information that I have now, there would be some things I would do very differently.

For example, although a waterfall method is one of the most common and reliable developing techniques. I would use a more agile approach, programming, then testing and documenting as I go, I believe this would work better for me personally as I found doing all the testing at the same stage tedious. I don't enjoy staying on one part for too long as it becomes a chore for me and I begin to slow down. It would be more productive to complete one form (or one interlocking part of the system) test the part document it, and then move onto the next part. Working in this style would also help to ensure I don't forget about the criteria I'm working towards for that section.

When I started the project I would create a time management plan, a gantt chart of my deadline and where I should be in regards to the dead line, I would also leave room for unforeseen events, such as being off sick, or becoming stuck on a part of the code. This would allow me to ensure I wouldn't have to power through a night or cut of parts of my system as I would be running out time, instead it would allow me to see how long I have left to make changes and if needed move my time scale around accordingly.

I would also change how I built the bot. I ran into a few problems when creating the bot, such as memory issues, hardware limitations, running of digital to analogue (such as the bot curving due to the wheels not being straight) I could solve these problems in a lot of different ways, but I think mainly, I would change the parts of the hardware I used. I would add encoders on to the wheels to solve the curving problem more effectively than the PWM (as this drained the battery) I would also have a better interface with the bot than the sensors. As this was a very fiddly and caused some issues. The sensors on the bot weren't always guaranteed to be activated when the bot hit a wall or object due to the switches small contact area, I would change the switches into either an ultrasound sensor or into a bumper with a much larger switch surface.

Another thing I would do differently if I had a second chance, would be to add an in system message board, allowing the students to talk to each other and the teachers to talk to the students. This could allow the student to help each other with questions rather than as the teacher, it would also remove the need to email the teacher therefor removing the need to have emails in the system at all, this would reduce the data stored by the program and also possible mean the system could work without the need for Wi-Fi (as another solution could be made to keep the unchanged user file)