Prj4: Project Final Requirements

# 25 Points. Submit all documents to Canvas.

# Deliverables

1. An ACCESS database file containing all your tables and the interface. (Steps 1 & 3)
2. A Word file containing Steps 2 & 4.
3. If relevant, supporting documents for Step 4.

# Submission of project

**Submit all documents through Canvas.** Make sure you have attached all files before hitting the submit button. Note that multiple files can be attached, one-by-one, through the submission area.

**Step 1 (0 Points).** If you didn’t create all tables in ACCESS in Prj2, implement all tables in ACCESS in this stage. Enter some data in each table. Define the PKs and the FKs. You may use fictitious data. For tables similar to Customer, Emp, and Product, enter 5 or more rows of data; for tables similar to Order, Purchase, Transaction, Class, enter 7 or more rows of data; For tables similar to TransactionDetails; OrderDetails (i.e., tables created from m-n relationships), enter 12 or more rows of data. (If you have already done this for Prj2 and submitted it for grading, there is nothing to do for Step 1.)

**Step 2: (5 Points):** Describe the functionalities that users of this database would like implemented. This is the basis for your interface design. This is where you discuss functionalities from the user's perspective, before the interface is designed. This is not where you discuss the forms in your database. Note that all tables in your database must be used in some fashion; otherwise you’ll lose points in the Interface Design step (Step 3). Submit this in a Word document with a cover page that lists 1) name of project, course number and title, semester and year, 2) your name.

**Step 3: (20 Points) Interface Design**

Design the ACCESS interface on top of the tables you created in Step 1 to allow users to interact with them (retrieve/insert/delete/update). ). Your interface should provide functionalities described in **Step 2**. Since a database allows users to retrieve, update, delete, and insert data, you may want to design your interface to provide users with easy or background retrieval/updating/deleting/inserting. For example, you may want to design some parameter queries to retrieve information. You may also allow users to perform searches. For another example, if you are designing a database for a library, the switchboard should contain a textbox or a combo box for entering a PatronId for checking out books. Upon entering the PatronId, a new row is automatically added to Table "Transaction." This new row records current date/time and PatronId. Next, there should be a textbox or a combo box to enter barcode of each book checked out. Upon checking out each book, a new row is automatically added to Table "TransactionDetails" for each barcode so entered so which books the patron is taking out are recorded. Table "Book" should be updated to record that these books are checked out. Users should be able to navigate from one form to another in a logical, easy-to-use fashion. To achieve that, use various controls discussed in class such as combo boxes and command buttons. Subforms are also a very good way to allow users access to multiple inter-connected tables in the database. Use Macros to automate automating routine business operations such as updating book status upon checking out as discussed above. Just build a form for each table, which would allow users to insert, delete, and update rows, would not earn you a good grade for this step. Your design of the interface should validate your data requirements stated in Prj1 (if users cannot interact with certain data, why store them?), and should correspond to required functionalities addressed in Step 2.

Note: Try to use all principles discussed in the handout: e.g., 1). user combo boxes for foreign keys and other places, 2) use auto look up forms, 3) use subforms, 4), use unbound controls to solicit inputs for searches, 5) provide ADD, DELETE, SAVE, and CLOSE buttons, 6) Use Update, Insert, and Delete queries that are called by macros, 7) Use macros for automation.

**Step 4: State project source (see Prj0 one-page description requirements document) and Provide Supporting Documents, if applicable**

If your project is a real-world project, say so and provide supporting documents. Attach blank paper forms for the application environment that you are designing the database for. In your write-up, provide the actual mailing or web address of the company. You may provide other contact info as well (such as phone numbers) and any information that may substantiate your claim that it is a real-world project. If no claim is made or no supporting documents are provided, your project will be considered a made-up project. (Note, paper forms are forms people fill out using a pen, not ACCESS forms.) If you’re reverse engineering a database, say so and provide web links and supporting documents if possible.

# Grading of the project:

The total project score is adjusted up for complex projects and for real-world projects.