CS 2300 - Database Project Guidelines

The objective of this course's project is to instill the concepts learned in class through the implementation of a database-dependent application. Each project will consist of between 2 or 3 members[1], each needing to agree upon an application idea that would be useful to some external user and that heavily depends on an underlying database.

The project consists of three phases and it constitutes 30% of your total grade for this course:

- Phase 1: Conceptual database design (20%) – 60 pts
- Phase 2: Logical and physical database design (20%) – 60 pts
- Phase 3: Database implementation and testing (50%) – 150 pts
- Demonstration (10%) – 30 pts

Details of the various activities and the deliverables of each phase are defined below.

Phase 1: Conceptual database design.

The first step is to identify an application. Examples include, but are not limited to, retail inventory control, airline reservations, a searchable and manageable music library, or supermarket purchase analytics. The second step is to analyze the intended use of this application. In particular, you need to define the functionality that will be offered by your application to an end user. The third step is to design the conceptual schema of the database that will power the application using an ER/EER diagram.

The deliverable from this phase is a project proposal, including the following sections:

1. **Problem statement**: Concise but thorough description of the application for which you propose to build, and why a database system is essential to this application.
2. **Conceptual database design**: Design ER/EER diagrams and briefly describe the meaning of your design and any assumptions made in a similar format as the examples shown in class. It is required that your design includes all types of ER/EER components (e.g. derived attributes, participation constraints) that were covered in class. Your design is expected to contain at least 5 entity sets (a weak entity set is also counted as one entity set) with a reasonable number of relationships among them.
3. **Functional requirements**: Briefly describe the functions that your application will provide, and list the entities involved in each function.

Take an airline reservation system for example.

Some example functionality offered by such a system are "search for a flight given the departure location and time", "book a ticket", "check the itinerary for the flight associated with a reservation number". The "book a ticket" function may involve three entity sets, such as a "flight" entity set which stores flight information, a "customer" entity set which stores customer information, and a "ticket" entity set which stores ticketing information of all the flights.

[1] Not advised; however, you may work independently if wish to

Phase 2: Logical and physical database design

In this phase, you will perform the logical and physical database design, and user interface design. The deliverable from this phase is the report including the following sections:

1. *Revised problem statement:* (from phase 1).

2. *Revised conceptual database design*: (from Phase 1)

3. *Logical database design:* Convert the ER/EER diagrams into a set of relations. Clearly indicate all primary and foreign keys. Also, prepare a table which summarizes the data type, meaning of each attribute, and integrity constraints on each attribute.

4. *Application program design:* Based on the revised functional requirements from Phase 1, provide a high-level description (e.g., pseudo code) of main steps in the major functions. The system should have the following basic functions: **insertion, deletion,** and **modification**. The system should have some queries that need to **join multiple tables.** The system should also have some **query functions** which allow users to provide input. For example, a user can search his itinerary by entering his ticket number. Moreover, the system should have some data aggregation functions using **average, sum, max, min, etc.** For example, list the number of available seats for each flight.

   An example design of the "book the ticket" function may look like this:

```
Function 1: Book_the_ticket

// This function is to book the selected air ticket.
// It accesses "ticket" table, "flight" table and "customer" table.
Input: customer ID, the ID of the selected flight
Steps:
(1) Check the ticket availability in the "ticket" table based on the
    flight ID
(2) Reserve a seat in the "ticket" table by updating the seat status
    to "booked" and recording the customer ID.
(3) Retrieve the flight information from the "flight" table,
    the seat number and ticket price from the "ticket" table,
    and create the itinerary for the customer.
(4) Insert the itinerary to the "customer" table.
(5) Display the itinerary.
```

5. *User interface design:* Sketch your system interface or provide screenshots of your system prototype. **Extra credit will be awarded for Web Interface after the completion of the project.**

Phase 3: Database implementation and testing.

In this phase, you are going to fully implement your proposed system. Consider possible optimization of your design by using functional dependency. Generate the appropriate data to populate the relation schemas. The volume of the data should be such that interesting queries can be posed and all user transactions will produce meaningful results. Transactions will be implemented, and the system will be fully tested.

NOTE: The database project must be implemented with a **RELATIONAL** database/DBMS system. If you are not sure about the database system of your choice is appropriate or not, consult with the TAs or the instructor first.

The deliverables for phase 3 include two files—the source code and the final project report.

1. Your source code in a .zip archive named "**GROUPNAME_source.zip**", where GROUPNAME is your group's name (if you are working alone, use your project name if you desire or simply use your LASTNAME).
    a. You must also include a single **.sql** file that will instantiate a new database with all tables, user credentials, data, and relationships, and anything else pertinent to the *initial setup* of your database.
    b. Additionally, include an *optional* installation script (e.g. Bash script, installation .exe) that can be used for installing all dependencies of your project.
2. Your final project report named as "**GROUPNAME_report.pdf**". It should have the following sections:
    a. Problem statement (revised from Phase 1)
    b. Conceptual database design (revised from Phase 1)
    c. Logical database design (revised from Phase 2)
    d. Application program design (revised from Phase 2)
    e. Installation instructions:
        - intended operating system
        - steps to installing your system on another computer, and a list of dependencies.
    f. User manual:
        - This is for end-users who may not have database knowledge.
        - Describe precisely how to use your system step by step with screenshots of your system interface and sample outputs.

Demonstration

Your team will need to demonstrate your database system in class according to a later-defined schedule. This demonstration is intended to show off what you've implemented, and should be conducted in such a way as if you were presenting to other non-Computer Science undergrads - no technical jargon, no EER/relational diagrams, no walk-throughs of the code. Only a walk-through of the user interface and examples of how to accomplish various tasks that are useful to end users.

For those designing a web application, it is recommended to deploy your system to a free cloud environment such as OpenShift and Heroku; otherwise, you may also self-host it. Regardless, you also need to provide me with source code and instructions to running your system on my own server.

For those creating a desktop application / command-line application, I must be able to compile

and run your application from the user manual provided in your project report.

For those creating an Android / iPhone app, you will need to detail the process of compiling the source code into an installation package and installing it to an Android smartphone / iPhone. Your submission ZIP archive must include both the source code for your application and a pre-compiled app installation file.