CS 166: Project Description and Phase 1 Requirements

1 Introduction (Scenario)

In this project we will model and build a Pizza Delivery Application. We are expecting it to allow users and employees to see the real-time changes on their orders, so we want to use a database to support efficient data management.

We want to build a demo system to show our design. To attract customers, we are planning to show some of our frequent customers our application with a friendly UI, having a database system as a back-end. We will have three phases for this project: (i) requirements analysis using the ER-model, (ii) Relational schema design, and (iii) implementation. Finally we will have a short presentation to the funders (in this case course instructors) on all the functionalities of our system.

Phase 1: ER Design

In the first phase we will do the requirement analysis using the ER-model. All the requirements can be obtained from Section 2 of this document.

After this phase you should generate an ER-diagram with any other supporting documentation for the system. For the ER-diagram, you can use any graphical editor you want, and you should finally create a PDF file using ER notations from the lectures/labs/book. You have to submit all your files compressed into a single file through elearn (Canvas) by the deadline. In this phase we will evaluate the correctness of your ER-diagram. You can make reasonable assumptions on your design, as long as:

- you state them clearly in the documentation for this phase, and
- they do not contradict the system requirement analysis we

provide.

Phase 2: Relational Schema Design

In this phase we will provide you with a common (final) ER-diagram (so that the whole class will proceed with the same design). This final ER-diagram will be the starting point for the second phase, which involves the creation of the relational schema.

Your task in this phase will be to translate the provided ER design to a PostgreSQL relational database schema. The database schema will be in form of a single recreatable SQL script (*.sql file with SQL statements), and you have to turn in your script through elearn (Canvas).

For this phase, you will be evaluated for the correctness and completeness of your Relational schema. You may find some constraints in the model and/or system requirement analysis that are not possible to represent or enforce in the relational schema. You may specify all these issues in the documentation and it will be considered in your final grade.

Phase 3: Implementation

After collecting all your submissions on the relational schema design, we will combine them into a final common schema to implement our system in phase 3. Your task in this phase will need to:

- Design Physical Database (with respect to DB performance tuning using indexes)
- Develop Client-Application (console application in Java program language from which the various functionalities of the system can be executed)
- Write a profound documentation.

In order to keep the costs of the overall system down, we are to use PostgreSQL open source database management system.

The Client Application development should be done in Java. Do not worry if you are not familiar with Java. A skeleton program and examples will be given after the second phase. Different programming languages other than Java (i.e. Ruby, PHP, C++, C#) can be used for this phase, but don't expect to get any detailed help on implementation from the TA.

Don't underestimate this phase. If you think it will take only 10 hours work to finish it, think again! It is strongly recommended that you start early and allocate at least 20 hours per person to get it finished. Don't forget that each group has to schedule a presentation to show the system running with all its functionalities to the TA. Slots for the presentation are available online on a first come-first served basis.

For this phase, you will be evaluated based on the system requirements for phase 3. Your GUI and source code will also be taken into consideration in your final evaluation. Groups that implement systems with user-friendly interfaces, extra functionalities and error handling (i.e. invalid values, wrong operations, meaningfulness messages) will receive an extra credit. A final report about your system along with its source code has to be submitted within the due date. You have to submit it (documentation and final source code) using the elearn (Canvas) system. Please keep in mind that we have already prepared a set of data, which you can load in the database once you create it. This dataset will be provided to you and you can modify it but you must report it in the final documentation.

1.1 Grading

Your contribution to this start-up project will be graded based on the following characteristics:

- Phase 1 (30%)
  - 30% Conceptual Design (ER diagram)
- Phase 2 (10%)
  - 10% Logical DB Design (Relational Database Schema)

- Phase 3 (60%)
  - 30% Implementation of SQL queries/reports in the Client-Application Development
  - 10% Physical DB Design (DB performance tuning using indexes)
  - 10% Client-Application Development (a console application in Java is expected)
  - 10% Documentation of the project including any assumptions that you have made
  - +10% Extra credit for good GUI design and interface, any dataset or schema changes/ extensions, etc.

This project will be performed in groups of TWO students. No individual submissions are allowed. Choose your partner wisely because the final evaluation is based on the group performance! In your report explicitly enumerate the tasks that each member of your group was responsible for. If one of the group members does most of the work, the grade will be proportional to the effort.

2 Requirement Analysis

1. **User**

   A User needs to register to be able to use the application.

   Overall user table will have the following information:

   - phoneNum (required)
   - login (required)
   - password (required)
   - role (required)
   - address (required)
   - favoriteItem

   Any newly created accounts should automatically receive a role of ”Customer”. Only the Manager can set the role of a User from the default ”Customer” to ”Driver” or ”Manager”.

   After logging in, the users can view the menu, place an order, view delivery status, and view/edit their profile. Drivers and Managers have an additional option of updating delivery info for an order. Managers can also update the menu and change the role of a user.

   In their profile menu, customers are free to update any field except for login and role.

2. **Item**

   An item is an available food item that can be ordered (pizza, drinks, etc...)
   - itemName (required)

- type (required)
- price (required)
- ingredients
- description
- imageURL

Only the Manager can update the list of items.

3. **Order**

   The main form of interaction between the pizza store and the customers is through an order.

   - orderID (required)
   - login(required)
   - orderTimestamp (required)
   - totalPrice (required)
   - orderStatus (required)

   Once a user places an order, we record the user's login, the time they placed the order, the total price, and we set the orderStatus to "Order Received" Afterwards, we issue a unique orderID for this specific order.

4. **Store**

   Customers will place an order at a specific store. Information about the store is here.

   - storeID (required)
   - address (required)
   - city (required)
   - state (required)
   - isOpen (required)
   - reviewScore

   An order is placed at a specific store.