CS 166: Project Description and Phase 1 Requirements

January 26, 2022

1. Introduction (Scenario)

Imagine that you create a start-up that will build the next Professional Network application. You are expecting a really large number of users and daily accesses when you will put it online (it will be the new LinkedIn!), so you want to use a database to support efficient data management.

Before bringing your idea to the start-up funders, you should have a demo system to show your design. To attract these funders, you will design a desktop application with friendly UI, having the database system as a back-end. You will have three phases for this project: (i) requirements analysis using the ER-model, (ii) Relational schema design, and (iii) implementation. Finally, you will have a short presentation to the funders (in this case course instructors) on all the functionalities of your system.

Phase 1: ER Design

In the first phase you will do the requirement analysis using 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, you should finally create a PDF file.

In this phase we will evaluate the correctness of your ER-diagram. You can make reasonable assumptions on your design, as long as:

• that you state them clearly in the documentation for this phase
• they do not contradict the system requirement analysis we provide.

You have to submit all your files compressed into a single file through eLearn by the deadline. The due date for this phase is **February 12th, 2022**.

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).

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. You have to turn in your script through eLearn. The due date for this phase is **February 22nd, 2022.**

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, you are to use the 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 TAs.

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, efficient queries, 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 sent to your TA at least 2 hours before your presentation. You have to send it (documentation and final source code) using the eLearn system and then send an email to your TA letting him/her know that it is available. 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.

The due date for this phase is **March 12th, 2022.**

1.1 Grading

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

- Phase 1 (30%)
  - – Conceptual Design (ER diagram)

- Phase 2 (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
  - – Extra credit (up to +20%) for good GUI design and interface, any dataset or schema changes/extensions, efficient query writing 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

(a) **User**

A User needs to register to be able to use the application. email, login and password should be provided during the registration. Overall user entity will have the following information:

- email (required)
- login (required)
- password (required)
- Name
- Date of Birth
- Education Details
- Work Experience

Whenever a user registers either s/he can give the personal details then, or update the profile at a later point of time. Note that for a user the data for Education details and Work Experience could be multiple. When a user logs in, s/he can do the following things:

- Search people in the network by name
- View Profile of friends (friends of friends ... can go up to any level)
- Send a connection request (Only possible till 3rd level of connection). For example, if the chain of connection list is A→B→C→D→E, then A can send a connection request to C and D but not E since it is a fourth level connection.
- Message anyone on the network (No connection rules apply.)

(b) Message

The main form of user interaction in our service - exchanging messages. Each message has the following information:

- Unique message id (required)
- Message's text (required)
- Timestamp (required)
- Sender ID (required)
- Receiver ID (required)
- Delete Status
- Message Status (required): "Delivered", "Read", "Failed to deliver"

Messages can only be one to one. They can be created, modified and deleted only by their authors. For example, if user A sent a message to user B the delete would work as follows:

- A deletes the message: The message disappears from A's sent items but still appears in B's received items.
- B deletes the message: The message disappears from B's received items but still appears in A's sent items.
- A and B delete the message: The message disappears from A's sent items and B's received items

(c) Connection

Connection defines which user is friends with whom.

- User id (required)
- Connection id (required)
- Status (required)

Initial sender is the user, who sent the connection request. Connection is the user to whom the request is sent.

2.1 GUI Requirement

1. When a user logs in to her/his page s/he should be able to view her/his entire profile data except password.

2. When a user views the profile of a connection s/he should be able to view Name, Date of Birth, Education Details, Work Experience

3. When a user views the profile of a user who is not a connection s/he should be able to view Name Education Details, Work Experience

4. When n user visits someone’s profile s/he can either add the user as a connection or send a message. The user should not see the add as connection option if the profile is beyond a 3rd level connection.