

CS 166: Project Description and Phase 1 Requirements

January 31, 2016

1 Introduction (Scenario)

In this project, we will model and build an Online Messaging application. We are expecting a really large amount of users and daily accesses when we will put it online (it will be the new WhatsApp!), so we want to use a database to support efficient data management.

Before we bring our idea to the start-up funders, we should have a demo system to show our design. To attract these funders, we will design a desktop application with friendly UI, having 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 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 iLearn 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:

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

The due date for this phase is **February, 8th**.

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 iLearn.

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.

The due date for this phase is **February, 15th**.

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 15 hours work to finish it, think again! It is strongly recommended that you start early and allocate at least 25 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 will be 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 sent to your TA at least 2 hours before your presentation. You have to send it (documentation and final source code) using the iLearn 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, 11th**.

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

- +20% 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 register to be able to use out Messaging application. phoneNum, login and password should be provided during the registration.

Overall user table will have the following information:

- phoneNum (required)
- login (required)
- password (required)
- Status message
- Contact list
- Blocked users list
- List of chats (group as well as private chats)

Whenever user registers an empty contact list will be associated with his account. Later a user can add phone numbers of other users to his contact list. Whenever a user browses his contact list he can see status messages of his contacts.

A user can receive messages from any other member on the system, unless another he is in the user's block list. Messages from users in contact list are indicated by contact's login, others are indicated by phone number.

Upon registration user will have 0 chats. A new chat is added whenever user sends/recieves a messages to/from specific user or a user group. Chats are displayed in the chronological order of their last update time.

2. 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)
- Message Status (required): "Delivered to the user", "Read by user", "Failed to deliver to the user"

Messages can be created, modified and deleted only by their authors.

Whenever a receiver (or all receivers in case on a group chat) will read the messages it's status should be changed from "Delivered to the user" to "Read by user".

Latest message timestamp will be used to identify the chat last update time.

3. Chat

In our application messages are grouped in logical entities: chats. There exist two types of chats: private (one-to-one) and group chats. Chats have the following attributes:

- Initial sender (required)
- Receiver(s) (required)
- Chat type (required)
- Ordered list of messages

Initial sender is the user, who started the chat.

Members can be added or deleted by the sender. The sender can upgrade the chat from private to group chat by adding more recipients. After that all previous messages will become available to the new chat members.

All chat members can see the chat in their list of chats.