Bilkent University
Department of Computer Engineering

CS 491
Senior Design Project

Low-Level Design Report

Project Name:

GrouPub

- A Location-Based Quiz Application -

Group Members:

Arda Ekmekçi	21101065
Ayberk Aksoy	21100623
Ekin Karayalçın	21101919
Merve Tuncel	21102000
Seren Erdoğan	21100693

Supervisor: Fazlı Can
Jury Members: Selim Aksoy, Hakan Ferhatosmanoğlu
Expert: Mehmet Çakır

Project Website: http://groupub.github.io/

Table of Content

Table of Figures

1. Introduction

GrouPub is a location-based quiz application where users can sign up to an event, form or join a group and participate in a quiz event that is hosted in a specific pub or café. Users need to create an account to be able to use GrouPub. After successfully creating an account and logging in, users will see a list of upcoming events. These events represent the actual quiz event that is taking place in a specific location. After selecting an event, users can see the groups that were already formed by other users that belong to the selected event. Users can either join a group that has an available slot or form a new group (maximum of five). Once the user successfully joins or forms a group, he/she can go the specified location at the specified time to join the quiz.

A quiz event will start at the specified time and users that have successfully signed up to that quiz (and present at the specified location) will start to receive questions every ten minutes. Users will have twenty seconds to answer a question. Once the quiz event is over, the winner group will be rewarded with drinks or food by the host and they will be recorded in GrouPub's public leaderboards. The winner group will also be prompted to take a photo of themselves and upload it GrouPub's leaderboards (this is an optional step for the group).

The purpose of GrouPub is to create an environment where even strangers can form groups, share a few drinks and socialize while trying to win a quiz event. It is an alternative activity that saves users from their daily routine, clear their heads, make some new friends and just have fun.

1.1. Object Design Trade-Offs

Functionality vs. Usability: GrouPub is a social application for people who need to relax and have some fun. Because of that GrouPub has simple yet user friendly GUI which favors the usability. Keeping the GUI simple enables us to reach many and wide range of users since the application will not require any complex interactions with the user. At the same time, intended functionality brings requirements that may be reason of complexity for user interface design. Our priority is to provide usability because our purpose is to facilitate user's work. The users will not need to waste time to understand our system.

Compatibility vs. Extensibility: As the developers of GrouPub, our target is to develop it for wide range of users. Therefore, the system design of the application should be compatible with the different versions of the Android OS and iOS. Nevertheless, extensibility of system conflicts with compatibility due to that old version of Android operating system like version 2.2 does not provide flexibility for the programmer unlike Android version 4.0 and the iOS will be implemented in later phases of the development. For the reasons above we decided compatibility over extensibility.

Speed vs. Space: GrouPub should be fast in order to meet user's satisfaction. Since it is a social application where the top priority is providing fun to its users, GrouPub should be fast which means we value speed over space. This does not mean there won't be any space management.

Space vs. Cost: Since this application is developed by students, the fund allocated for this Project is not much. Because of that we will favor cost over space which means there will be space management and a limit to the storage space to decrease cost.

1.2. Interface Documentation Guidelines

The interface documentation guideline will be created in following hierarchy. The general purpose and the introduction is given in the method overview, attributes defined under Attributes with the following pattern; Attribute Name, Type and Description. The method definition consists of the parameters and the return type of the method.

- Class Name (className)
- Method Overview
- Attributes
    - Attribute Name
    - Type
    - Description
- Methods
    - Method
        - Description
    - Parameters
        - Parameter Name
        - Description
    - Return Value

1.3. Engineering Standards

Engineering standards are the documents that indicate the specifications, characteristics and technical details of the system and also cover the development process of the product. Its purpose is to ensure the product/software is consistent. Our project GrouPub aims to reach various people in different places so we need to standardize the social and technical challenges within application development.

We are using different software development environments such as JavaScript, Spring, Java and so on. Therefore, we followed Java Language Specification standards, which represented in Oracle [1]. IEEE Standards Association (IEEE-SA) and IEEE Educational Activities Board (EAB) promote the importance of standards in meeting technical, economic, environmental and societal challenges [2]. In addition to this, ANSI (American National Standards Institute) provides quality management for small businesses standards, which includes requirements and fundamentals of the product quality. It aims to enhance customer satisfaction through effective application of the system, including processes for improvement of the system and also need to demonstrate application's ability to consistently provide services to the customer [3]. GrouPub's customers vary from age, social and cultural status of the application users. Therefore, we need to standardize our development process to build consistent application. Besides these ethical and social-based standards that we followed, as the technical details of the process, we drew GrouPub's subsystem decomposition structure according to UML (Unified Modelling Language) standardizations. By this means, we ensure that GrouPub's classes and objects are created based on the Object Oriented Programming specifications.

1.4. Definitions Acronyms & Abbreviations

Firebase: An API that provides powerful services such as user authentication, static hosting and more… [4]

HTML: Hypertext Markup Language [5]

ISP: Internet Service Provider [6]

JSON: JavaScript Object Notation [7]

Node.js: An API that provides services for scalable network applications. [8]

Spring: A framework that is used to create JVM (Java Virtual Machine)-Based applications and systems. [9]

XML: Extensible Markup Language [10]

GUI: Graphical User Interface

iOS: iPhone Operating System [11]

IEEE-SA: IEEE Standards Association

EAB: IEEE Educational Activities Board

ANSI: American National Standards Institute

UML: Unified Modeling Language [12]

MVC: Model-view-controller [13]

REST: Representational State Transfer [14]

JSP: JavaServer Pages [15]

DAO: Data Access Object [16]

SQL: Structured Query Language [17]

JDBC: Java Database Connectivity [18]

POJO: Plain Old Java Object [19]

CSS: Cascading Style Sheets [20]

JS: JavaScript [21]

2. Packages

2.1. Java Packages

2.1.1. Controller

This package contains the controller classes of our MVC project structure. Controllers are the RESTful services that handles the post and get requests from the JSP pages which are the view components of the project.

controller
ChatController.java
LoginController.java
QuestionController.java
EventController.java
QuizController.java

2.1.2. Model

2.1.2.1. DAO

This package contains the DAO's (Data Access Object). DAO's are the java classes that are responsible of fetching the desired data from the database by executing SQL queries using JDBC (Java Database Connectivity).

dao
EventDAO.java
LocationDAO.java
QuestionDAO.java
QuizDAO.java
UserDAO.java

2.1.2.2. Entity

This package contains the entities that are called POJO's (Plain Old Java Object). POJO's contain only the attributes of the entities created in the database along with the getter and setter methods. DAO's are responsible of creating and filling the instances of these POJO's.

Entity
Event.java
Location.java
Question.java
Quiz.java
User.java

2.1.3. Exceptions

This package is created in order to contain developer created exceptions.

exceptions
UserAlreadyExistException.java

2.2. Resource Package

Resource package contains the XML files that hold the necessary configurations for the project. At this very moment, our project needs only the following XML file that contains database connection credentials.

resources
datasource.xml

2.3. Web Packages

Web package contains the necessary directories that contain files for view component of the MVC structure.

2.3.1. Resources

This package will contain the necessary resources for the web pages such as CSS, JS and image files.

2.3.1.1. CSS

css
leaderboard.css
main.css
question.css
chat.css
login.css
event.css

2.3.1.2. Images

This directory will contain the image files and icons of our project. (Not designed yet)

2.3.1.3. JS

This directory will contain the necessary JavaScript functions in order to be used in the web pages. Not all JS files can be determined before creating all of the web pages, however the following are the currently available ones.

js
countdown.js
questionResult.js
chatServer.js

2.3.2. WEB-INF

WEB-INF is a directory that contains files that are hidden to any user. The files are only available to the controllers for the server-side tasks.

2.3.2.1. Pages

This directory contains the JSP files which are the views of the MVC structure. Those JSP files are shown to users via the controllers' supervision.

pages
chat.jsp
leaderboard.jsp
login.jsp
event.jsp
main.jsp
question.jsp
register.jsp
registerValidation.jsp

3. Class Interfaces

Event
This class gets and sets information about quiz event such as address of it and when it will start.
-int id -Date date -String name -int locationId -int quizId
+getId(): int +setId(int id): void +getDate(): Date +setDate(Date date): void +getName(): String +setName(String name): void +getLocationId(): int +setLocationId(int locationId): void +getQuizId(): int +setQuizId(int quizId): void

Location
To keep track of the location that are registered to GrouPub, this class is used.
-int id -String name -String address
+getId(): int +setId(int id): void +getName(): String +setName(String name): void +getAddress(): String +setAddress(String address): void

Question
To get the question, choices and the correct answer of the question, this class is used.
-int id -String questionText -String answer -String optionA -String optionB -String optionC -String optionD
+getId(): int +setId(int id): void +getQuestionText(): String +setQuestionText(String questionText): void +getOptionA(): String +setOptionA(String optionA): void +getOptionB(): String +setOptionB(String optionB): void +getOptionC(): String +setOptionC(String optionC): void +getOptionD(): String +setOptionD(String optionD): void

Quiz
An event stores a quiz which stores questions. Every quiz has its own id to identify each of them in terms of scores.
-int id
+getId(): int +setId(int id): void

User
This class is the generic model that are used in both server and client.
-String username 'String password -String title -double exp -int correctAnswers -int wrongAnswers
+getPassword(): String +setPassword(String password): void +getUsername(): String +setUsername(String username): void +getTitle(): String +setTitle(String title): void +getExp(): double +setExp(double exp): void +getCorrectAnswers(): int +setCorrectAnswers(int correctAnswers): void +getWrongAnswers(): int +setWrongAnswers(int wrongAnswers): void

ChatController
This class helps communication between group members and when a new member applies to join the group.
+chatPage(ModelMap model): String

LoginController
This class is responsible for authentication to the application via password.
+getLoginPage(ModelMap model): String +checkCredentials(String username, String password, ModelMap model): String +getRegisterPage(): String +addUser(String username, String password, String passwordConfirm, Model model): String

QuestionController
This class works while a user solves a quiz question.
+printQuestion(ModelMap model): String

QuestionDAO
Data access object that provides an abstract interface to question database.
-DataSource dataSource
+setDataSource(DataSource dataSource): void +findByQuestionId(int questionId): Question

UserDAO
Data access object that provides an abstract interface to user database.
-DataSource dataSource
+setDataSource(DataSource dataSource): void +addUser(String username, String password): void +getUser(String username): User

AppTests
The Spring web MVC framework (MockMvc) provides model-view-controller architecture and components to develop flexible and loosely coupled web applications.
- MockMvc mockMvc #WebApplicationContext wax
+setup(): void +simple(): void

Main Controller
Main Controller is the class that controls every user action made after logging in. Therefore, it is the core class that binds all other controllers to each other.
-MockMvc mockMvc # WebApplicationContext wac
+setup(): void +simple(): void

4. Class Relationship Chart

This following chart shows the relationships among classes used in GrouPub. Please note that the classes shown in this chart are not in full detail, only their class names are shown for the sake of simplicity. For detailed information about the classes including the attributes, methods and description, please refer to section 3 of this report.

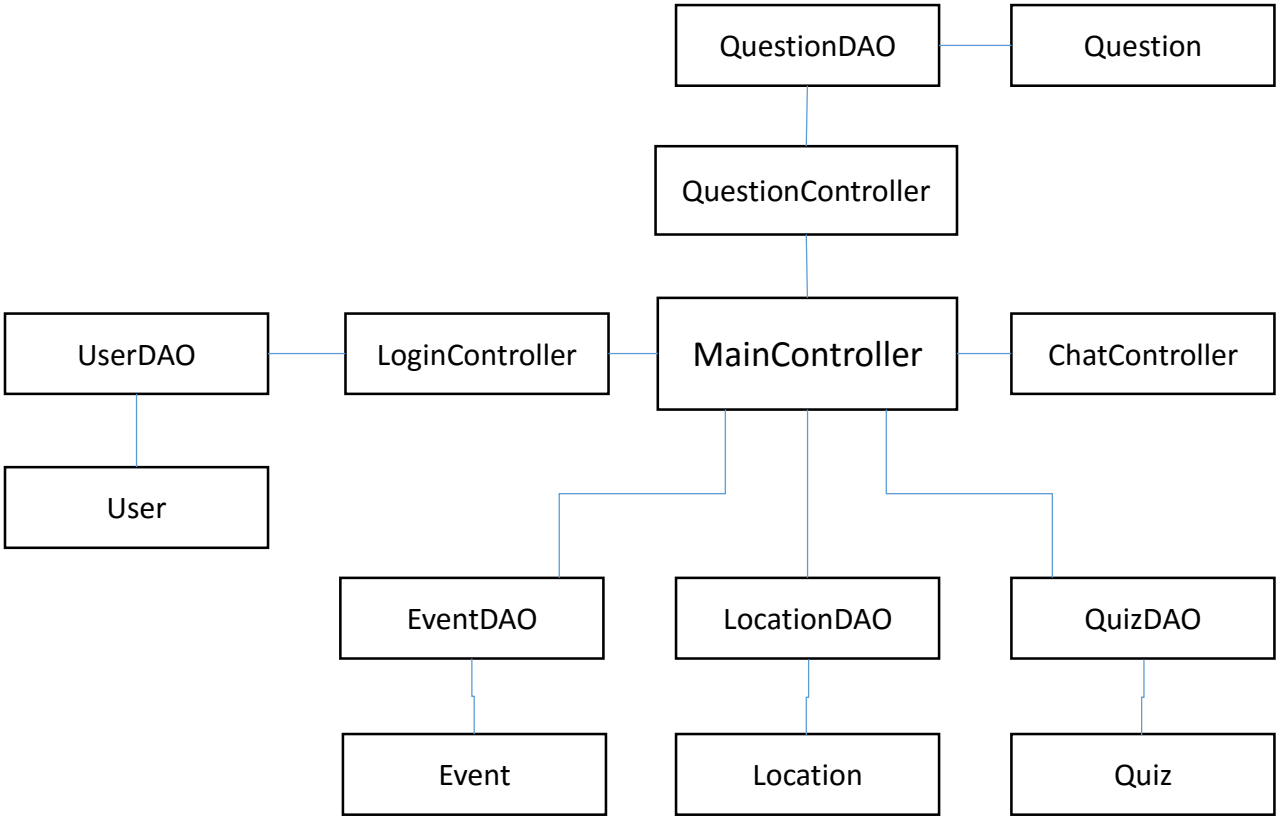

Figure 1: GrouPub MVC Structure Class Diagram

5. Keywords (Glossary)

Event: An event represents a quiz event that is taking place in a specific location (pub, café etc.). It includes information about the quiz event including the start time and the location.

Group: A group (that is formed by a user) represents a group of maximum five users that belongs to a specific event.

Heat Level: Heat level of a user indicates how often that user uses words that are registered in the word blocker's database. Every time a restricted word is used, the heat level will increase by some amount. After reaching a certain threshold, that user will not be able to chat with anyone for a limited amount of time. Note that the heat level of a user will start to cool down if that user does not use a restricted word.

Host: Host is the location where an event will take place. A host can be a pub, a café or anywhere public with tables and chairs where people can sit together and solve quiz questions.

Rating: User A can rate user B based on how polite user B was to user A. The average rating of each user will be publicly displayed, giving a rough description of how polite each user is.

User: A user of GrouPub with a valid account.

Word Blocker: A class (will be written by us) that scans a conversation and censors each word that is registered in its database. The database of the word blocker will be created and maintained by us. The purpose of this class is to censor out insulting and inappropriate words and to help calculate the heat level of a user.

Joker: A joker is a special item that allows users to modify a question such that it is easier to answer it. One example is removing two wrong answers from a question.

6. References

[1] Java Language Specification Standards - Oracle [Online]. Available: http://www.oracle.com/technetwork/java/javase/overview/index.html

[2] IEEE Standards – IEEE Webpage [Online]. Available: http://www.ieee.org/education_careers/education/standards/index.html

[3] ANSI Quality Management – ANSI Webstore [Online]. Available: http://webstore.ansi.org/RecordDetail.aspx?sku=ISO+9001+Quality+Management+for+Small+Businesses+Package&source=homepage

[4] Firebase Homepage [Online]. Available: https://www.firebase.com/

[5] HTML – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/HTML

[6] Internet Service Provider – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Internet_service_provider

[7] JSON – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/JSON

[8] Node.js – Homepage [Online]. Available: https://nodejs.org/

[9] Spring – Homepage [Online]. Available: https://spring.io/

[10] XML – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/XML

[11] iOS – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/IOS_version_history#Overview

[12] UML – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Unified_Modeling_Language

[13] MVC – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[14] REST – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer

[15] JSP – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/JavaServer_Pages

[16] DAO – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Data_access_object

[17] SQL – Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/SQL

[18] JDBC - Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Java_Database_Connectivity

[19] POJO - Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Plain_Old_Java_Object

[20] CSS - Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Cascading_Style_Sheets

[21] JS - Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/JavaScript