TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 491 – High Level Design Report

by

Alperen ASLAN

Yiğithan ÖCAL

Housey

A Mobile Application Project

15/01/2022

1. Introduction

This is a high-level design report for a mobile application Housey.

1.1 Purpose of The System

In the project, main purpose is the create an online social platform that can bring people together to make different kinds of activities together with people users know or never met before. Users can create an activity they want according to their own wishes; users can view activities created by other users and can send a join request to participate in those activities.

1.2 Design Goals

Performance: Nowadays, mobile applications that do not cause performance problems are preferred, therefore, we consider this when implementing the application so, we preferred to use systems that will not cause performance issues for our initial number of user goal.

User-friendly UI: We thought that a mobile application should have an user interface that would not cause any problems for the user. Therefore, we try to design user-friendly interface by trying to maintain the balance of simplicity and functionality.

Security: In the application, we stored user information in Firebase Databases. Data is stored in a way that cannot be accessed from outside in any way unless there is any authorization.

Low Cost: Housey will be a completely free application and users will not have to pay to user basic system features.

1.3 Definitions, acronyms, and abbreviations

JSON: JavaScript Object Notation

Firebase RDBMS: Firebase Realtime Database

UI: User Interface

User: Any person who has properly created an account and is using the Housey.

MVC: Model-View-Controller

1.4 Overview

Housey is a mobile application both for Android and IOS systems which is designed helping people to socialize and people who need someone or something for any activity. The application is especially can be good for people who have different hobbies to reach other people who are engaged in that hobby without being alone.

2. Current software architecture (if any)

We did our research on how to use Firebase real-time database and how to design user interfaces using Flutter. While trying to use Firebase and connect the real-time database, we mostly getting help from Firebase's own document. While designing the user interfaces, we mostly examined the mobile application user interface designs on sites such as Dribble and Behance then we tried to implement them by using different sources from the internet. So, we need to use Flutter's widget structure when implementing user interfaces. We used Flutter's own documentation when using the widget structure. In the current architecture of the application, login and register operations have been implemented and interfaces have been designed, but better-looking interfaces will be implemented in future. Operations such as user creating and searching activity were taken care of and tested using a simple interface. Lastly, profile etc. the interfaces of the other pages are still under design phase.

3. Proposed software architecture

3.1 Overview

This section contains detailed information about Housey application.

3.2 Subsystem decomposition

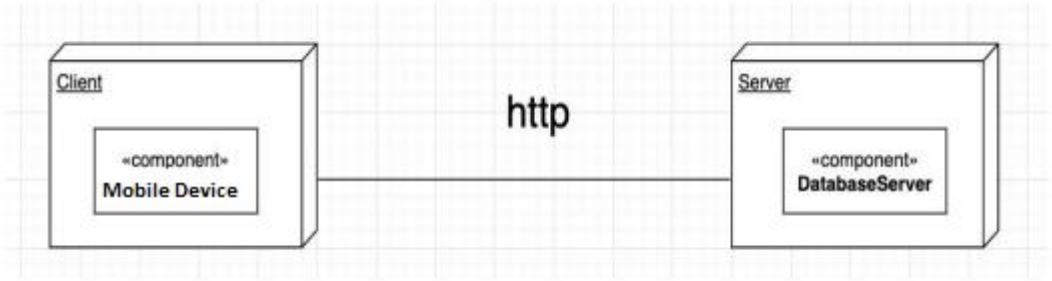
3.2.1 Android and IOS Application

Flutter is used for an implementation which is a cross-platform framework, application is developing for both Android and IOS at the same time, and a direct connection is established between Flutter and Firebase for data management and control.

3.2.2 Database Management

Firebase is using for the data management and database. All users and activities information stored in the Firebase.

3.3 Hardware/software mapping



Housey has two main hardware side: server and client. Client node connects users and their activities. Server, which is Firebase, holds the data of users and activities and hosts the application. Client must communicate with the server to be able to get data. This communication done with HTTP requests.

3.4 Persistent data management

Housey, has a medium sized data that needed to be managed. There are usernames, passwords, email addresses and profile information for user. Activity titles, descriptions, images, and locations for a single activity. Lastly, there will be message data for real-time chat included this project. Thanks to Firebase, we will manage this data with 3 parted database which is users, activities, and chats.

3.5 Access control and security

For security, we are developing the app by don't saving any information in the front end of app. When it is needed, we will use tokens instead of. And [1]"flutter_secure_storage" library provides encryption for it.

Secondly, we will use extra layer of security thanks to Firebase. When the hackers try to decrypt and private key, they will never know which is the right one.

Additionally, any case of brute force attack, we will implement to locking log in operation. If they type incorrect password 3 times, the backend will lock them out.

3.6 Global software control

For our system architectural style, there will be 3-tier control which are database, user interface and web services. First control will be at the time of registration and login. Thanks to Firebase, the registered data control will be handled from UI and database. On the other hand, there will be web service requests in case of when the user wants the nearest activity of him. By doing that, it provides a smooth connection between web services of Google and Firebase.

3.7 Boundary conditions

3.7.1 Initialization

Users will be accessing the app via App Store or Google Play Store. After downloading process, users can easily register to app.

3.7.2 Termination

For the better user experience, users will stay active until they sign out. Termination could only be done by clicking sign out button.

3.7.3 Failure

First case of failure is trying to log in with invalid information. In case of typing wrong values 3 times, app will be block log in operation for 5 minutes.

Second case of failure is trying to access app without internet connection. When application is open, there will be pop up which warns user to check internet connection.

4. Subsystem services

We have tried to use to MVC pattern while implementing the application. We decided that this was most suitable design pattern for the purpose of the project. We thought that the MVC pattern was suitable for our user integrations and communication between components.

4.1 Model

The user model keeps information about the user and arranges information for the CRUD operations.

The activity model keeps information about the activity and arranges information for the CRUD operations.

4.2 View

The LoginScreen is responsible for displaying login screen to user and sending request to Firebase server for the user login operation.

The RegisterScreen is responsible for displaying register screen to user and sending request to Firebase server for the user registration operation.

The HomeScreen is responsible for displaying activities and gives search option to user for searching desired activity and sending request to Firebase server and getting response that will render in the screen.

The AddActivityScreen is responsible for displaying add activity screen to user and sending request to Firebase server for the add activity request.

The ProfileScreen is responsible for displaying profile screen to user, it shows profile information to user.

4.3 Controller

Controllers perform the necessary operations according to the requests coming from the view classes. Using the Firebase, it performs the functional operations and sends the incoming data to the view class. Currently, all controller

classes and functions are together. In the future stages of the project, separate controller classes will be implemented according to the necessary project needs.

5. Glossary

No glossary is needed for this report.

6. References

[1] Flutter Secure Storage: https://pub.dev/packages/flutter_secure_storage

https://docs.flutter.dev/

https://firebase.google.com/docs

https://firebase.google.com/docs/database

https://www.behance.net/

https://dribbble.com/

https://www.developerlibs.com/2018/11/flutter-firebase-realtime-database-crud.html