

System Design Document for StudyBuddy

Bashar Oumari, Benjamin Sannholm, Mathias Drage,
Pontus Nellgård, Sophia Pham

2020-10-02
version 1

1 Introduction

The project is a mobile application developed for Android mobile phones using Android Studio. The main purpose of the app is to allow students to locate and interact with other students registered to the same courses and universities who want to study together or are looking for new study partners. The idea is loosely based on a Snapchat extension app called “Beer Buddy” in which people can see other friends and what they are currently drinking and request to join them.

1.1 Definitions, acronyms, and abbreviations

Abbreviation	Definition
Activity	A user interface “scene” in which the current session is logged and the user can interact with different Android UI components.
Android Studio [1]	The framework used for developing the GUI, logic, and running the Android phone emulator.
Broadcast	The team’s technical term for the users to display that they are actively seeking a study partner. It holds information about the course, description, and the location of the study session (Broadcast). On the map, they are seen as pins.
Firebase [2]	A Google-developed “backend as a service” that works well with the Google Maps location integration and account setup.
Fragment	Similar to an Activity (scene) in which the user can interact with and navigate between different Android components (UI).
Jetpack Navigation [3]	A framework for Android development to make it easier to navigate between different UI screens.
GPS	A self-developed class to handle geolocation and requesting the user’s device (and the user) to use geolocation activities

	(longitude and latitude).
Pin (see also “dot on the map”)	A red Google Maps icon that displays where a specific broadcast is located.
Google Maps [4]	A framework provided by Google to use their Maps API in Android applications
MVC	Model View Controller, a design pattern
OOP	Object-Oriented Programming, a design pattern
GUI	Graphical User Interface
JUnit [5]	A framework used for testing
Gradle [6]	A framework for managing dependencies
Git [7]	A version control manager
GitHub [8]	A repository for managing projects using Git

2 System architecture

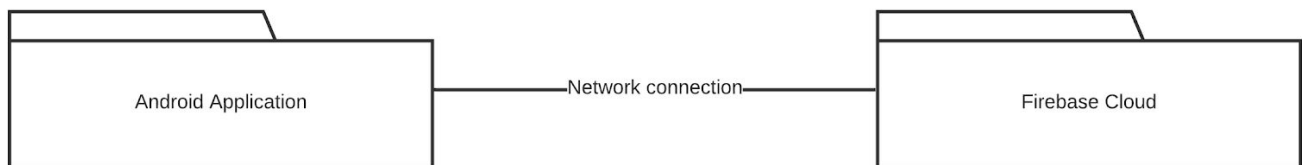


Figure 1: System components.

3 System design

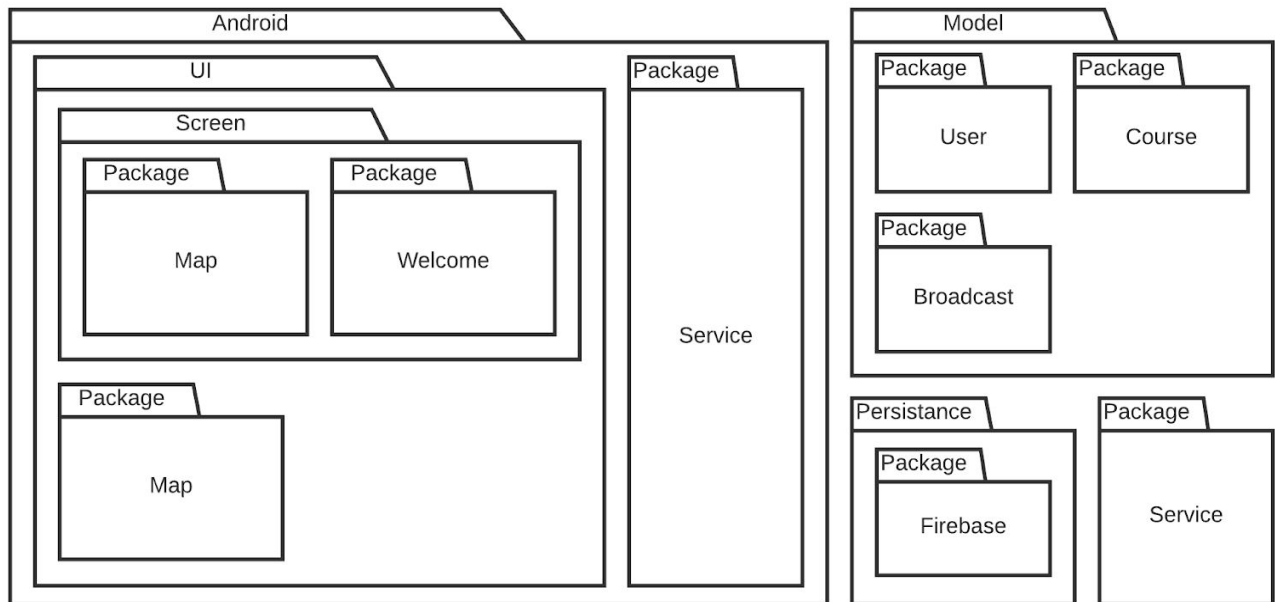


Figure 2: *Package hierarchy.*

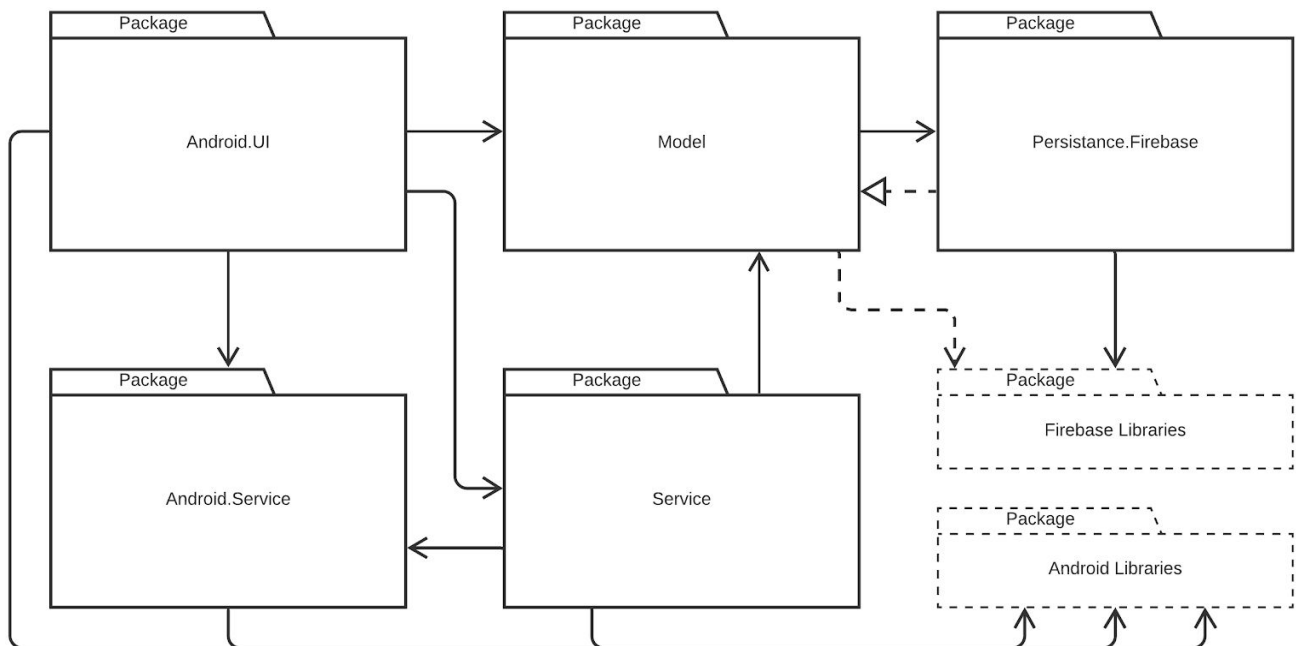


Figure 3: *High-level dependencies between packages.*

4 Persistent data management

Our application uses the Firebase realtime database for saving users data. On the other hand we use Firebase authentication to store users when making an account. The real time Firebase

database enables our application to store the coordinates of broadcast, last time active, course code, description and when the broadcast is created.

On the authentication side, we will check if the users have already an account or if a user wants to register an account when using the application in order to login and use the application.

5 Quality

At this stage, our tests consist mostly of unit tests for the classes which belong to the model. We have not applied any continuous integration tests. We will however work more on instrumental tests in the coming time.

We currently have no major issues with regards to our completed user stories.

See attachment for dependencies in UML diagram.

5.1 Access control and security

In our application we use a login and a registration fragment to gain access to the rest of the application. Login details are saved in our database provider firebase, after completing the registration process. Currently we have no other security protocols than login credentials, and we have not implemented any further security measures than saving the login credentials as plaintext in the database. On the other hand Firebase provides security protocols, but since we are in test mode now, our database is not secure. The test mode in firebase will be switched to locked mode, and security rules will be added when we feel that our application is starting to be ready.

6 References

Platforms:

- Android (<https://www.android.com/>)
- Firebase (<https://firebase.google.com/>)
- GitHub (<https://github.com/>)

External Tools:

- Android Studio (<https://developer.android.com/studio>)
- Gradle (<https://gradle.org/>)
- Git (<https://git-scm.com/>)
- JUnit (<https://junit.org/junit4/>)

Libraries:

- Android SDK (Bundled with Android Studio)
- Jetpack AndroidX Support Libraries (<https://developer.android.com/jetpack/androidx>)
- Jetpack Navigation Graph (<https://developer.android.com/guide/navigation>)

- Jetpack Architecture Components
(<https://developer.android.com/topic/libraries/architecture>)
- Google Play Services
 - Maps (<https://developers.google.com/maps/documentation/android-sdk>)
 - Auth (<https://developers.google.com/identity/sign-in/android>)
- InteractiveInfoWindowAndroid
(<https://github.com/Appolica/InteractiveInfoWindowAndroid>)
- Firebase (<https://github.com/firebase/firebase-android-sdk>)
 - Realtime Database
 - Authentication
 - Analytics

7 Attachments

