Rida Haider
01-134162-036

Social Music Application

Bachelor of Science in Computer Science

Supervisor: Dr. Kashif Naseer

Department of Computer Science

Bahria University, Islamabad

April 26th, 2020

Chapter 1

Introduction

1.1 Project Background/Overview

In this era of digitalization, sharing data over Bluetooth and keeping it on our personal hard drives belongs to the antiquity and everything is shifting towards cloud computing since cloud computing introduced secure, cost cut and resizable solution. The dilemma we plan to cater with our application is of facilitation sort revolutionizing the music experience for users in a phenomenal way. We believe in the power of music and have an eye for it as a medium of connecting with people.

Our application, like other social applications, uses music to be shared among your followers, allows you to live stream music for your followers, inserts an interactive element between users and creates a gateway to music library on internal storage of phone.

1.2 Problem description

Enhancing faith in the power of music and analyzing the current social media trends we see LinkedIn as a place for professionals, we see Instagram as a place for photographers and models, we see Twitter as a 140 character news broadcaster, we see Tumblr as a society for poets and we see nothing for the art of music. We aim to create a platform where music enthusiasts get the chance to interact and reflect upon their skills to the society. The vastness of this app in terms of future is huge considering the number of music lovers' existent in the society.

1.3 Objective

The main objective of the proposed project is to design a social mobile application related to music that allows users to listen to live music that is being streamed by their friends on their respective devices. And give users a socially interactive platform where they can share the music to their followers.

1.4 Project Scope

The project is aimed towards facilitating music lovers in sharing their music and expanding their tastes. It also helps multiple devices play the same song at the same time providing the users with a socially interactive application to share their music.

Chapter 2

Literature Review

There are many mobile applications available in the market related to music providing facilities to the user. Below mentioned are some of the applications and the detail of their basic functionalities and how this project may differ from them.

2.1 Previous Works

2.1.1 Spotify

Spotify is a paid application which allows user to listen to songs from their own limited library, and it does not contain all songs according to the user preferences. For instance, doing my research on various applications, I realized that Spotify doesn't have any search results for the genre of "Qawaali" which in my humble opinion is unjust to all the Qawaali music makers. Similarly, our findings also suggested the non-existence of psychedelic rock music. Plus, there might be several other genres that Spotify fails to cover.

2.1.2 Soundcloud

Soundcloud is another streaming platform for music enthusiasts and it revolutionized the music industry by allowing anyone to create music. This, in turn, brought phenomenal music artists in the industry but it came with a few drawbacks which we aim to cater. Firstly, Soundcloud does not give an interactive interface, has no feature to stream on local storage and most importantly, does not allow users to stream live music for their followers.

2.1.3 YouTube

YouTube itself is a multi-purpose application which covers everything from music and education to comedy and web-series. However, the biggest drawback it comes with is the unavailability of music to be streamed without video. Moreover, it comes with no element of interaction between the users. YouTube is not specifically designed for music artists which in turns creates a need to develop a proper and specific application for music artist so as to appreciate the art of music.

2.1.4 8tracks

8tracks is another webpage in which users can create their own playlists which their followers can stream. However, the music is linked to YouTube and it caters to the issue of music not being available without video. Furthermore, users do not have access to stream live music from their internal storage.

2.2 Proposed System

It is evident from above data that although many applications are available to cater the needs of music but, they do not contain many social elements of interaction as that of Instagram, Facebook etc. Hence this application solves the problem of providing a specific platform which attracts music enthusiasts to showcase their music to their followers through their profile along with the option of live streaming of tracks using their phone's internal storage.

Chapter 3

Requirement Specifications

3.1 Product Scope

I am designing an android application that is going to allow users to create music rooms in which other users will be able to connect to and listen to music. It is a social media-based application in which users will be able to follow other users so that they are notified in their feed when their friends start a music room.

3.2 Overall Description

3.2.1 Product Functions

1. Creating a user through authentication.
2. Searching and listing mp3 songs from the phones memory to the application interface.
3. Creating a room.
    a) Sending the song file to the server.
    b) Starting the song timer on the database.
    c) Initializing the chat in the room.
4. Other users connecting to a room.
    a) Searching for available rooms in the database and displaying them on the application interface.

b) Connecting to a specific room.

c) Stream song from the server at the current time.

5. Follow/Unfollow functionality.

3.2.2 Operating Environment

The application requires an android phone with at least marshmallow (6.0) OS. It requires a stable internet connection at all times.

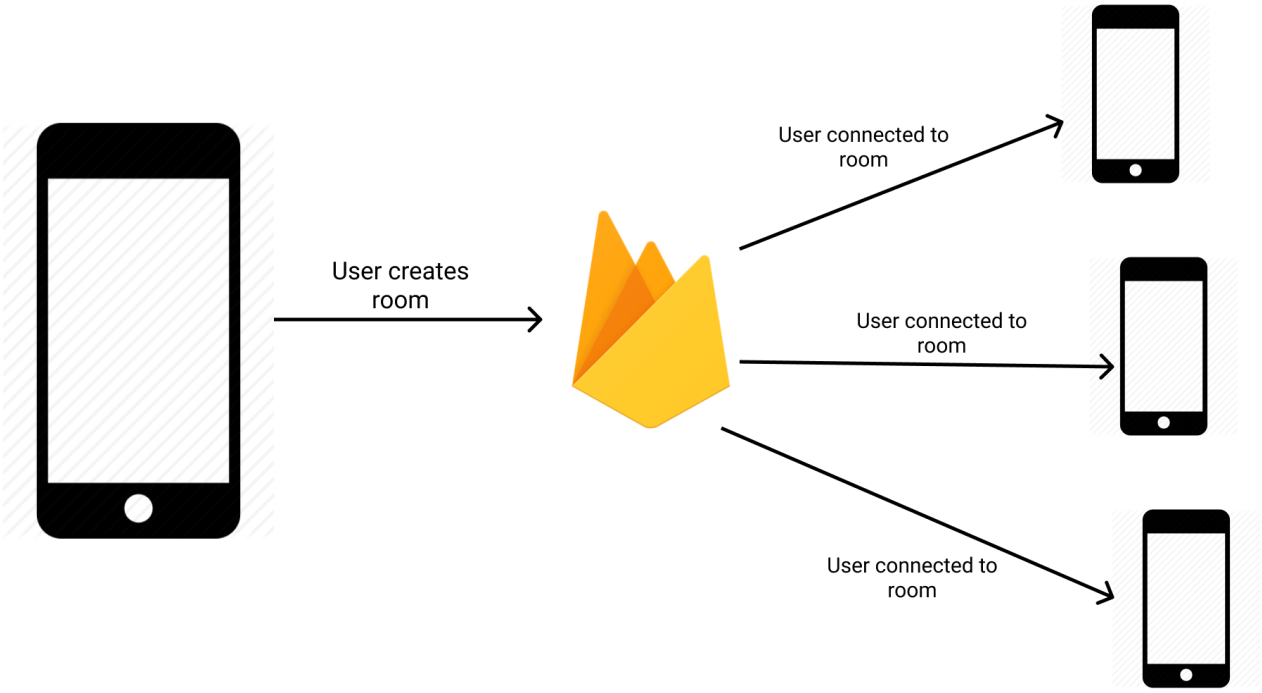


Figure 3.1 System Diagram

3.2.3 External Interface requirements

3.2.3.1 User Interface

- The application will have an easy to understand interface.
- All forms will have proper validation checks so that user does not enter incorrectly formatted information in the forms.
- A simple and attractive color scheme will be applied.

3.2.3.2 Software Interface

Java will be used in the android studio to develop the logic of the application and android frameworks XML will be used to develop the user interface.

3.2.3.3 Communication Interface

All the communication done between the application and the servers, which includes music streaming, chat data, database uploads and downloads require an internet connection.

3.2.4 System Requirements

3.2.4.1 Functional Requirements

- The application requires internet access at all times.
- The user must register in order to use the application.

3.2.4.2 Non-functional Requirements

- The application requires minimum Android Marshmallow version (6.0).
- There should be screen adaptation since android devices come in many sizes.
- Strong security rules of firebase cloud should be written.
- The application should be reliable to use.

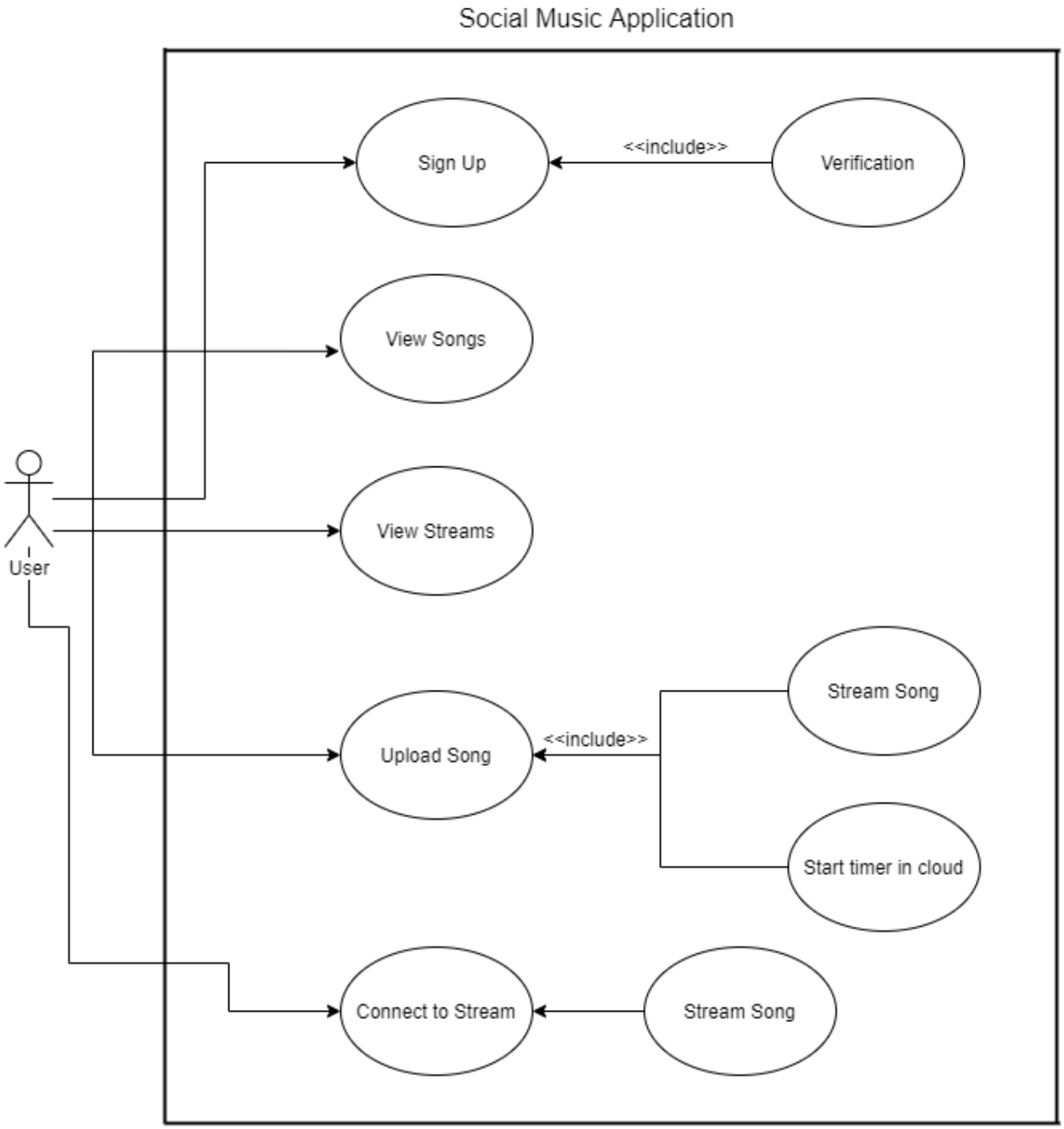


Figure 3.2 Use Case Diagram

Table 3.1: Sign Up

Use Case	Sign Up.
Actors	User.
Scope	Registering a new user.
Pre-Condition	User fills the sign up form correctly
Post-Condition	User is registered in the cloud
Triggers	Button Click (onClickListener)
Basic Flow	1. User enters the information in the sign up form. 2. User clicks the submit button. 3. Email verified. 4. A new user is created in the cloud.

Table 3.2: View songs

Use Case	View songs
Actors	User.
Scope	Display songs on UI from storage
Pre-Condition	Internal storage permission is granted.
Post-Condition	Songs are displayed on the UI.
Triggers	Pressing on the View Songs tab.
Basic Flow	1. Permission dialog opens when activity is loaded. 2. On granting storage access the intent gathers mp3 files from internal and external storage. 3. The names of the song are loaded into the recyclerview that displays the list of songs on UI.

Table 3.3: View Streams.

Use Case	View Streams.
Actors	User, Firebase Cloud.
Scope	Load online streams from the cloud.
Pre-Condition	Streams are available on the cloud and the user is authenticated.
Post-Condition	Available streams are downloaded from database and displayed on UI.
Triggers	Pressing on View Streams tab.
Basic Flow	1. On activity load the relevant streams are downloaded from the database. 2. The streams are bind to the view of recycler view and displayed on the UI.

Table 3.4: Upload Song

Use Case	Upload Song
Actors	User, Firebase Cloud
Scope	Uploading the song to firebase Storage
Pre-Condition	User is authenticated.
Post-Condition	User has started a stream.
Triggers	Selecting a song from the view songs tab.
Basic Flow	1. The user presses on one of the songs from the view songs tab. 2. The song is uploaded to the cloud storage and referenced in database. 3. User moves to new activity. (Stream Room)

Table 3.5: Connect to Stream

Use Case	Connect to Stream
Actors	User, Firebase Cloud
Scope	Connect to an available stream and stream its song.
Pre-Condition	User is authenticated and has streams available on View Streams tab.
Post-Condition	User connected to the stream room and is streaming its song.
Triggers	Pressing on one of the streams available on view streams tab
Basic Flow	1. User presses on a stream. 2. The application gets access to the relevant streams data (song, chat etc.) 3. User moves to new activity (Stream Room) and song starts streaming.

Chapter 4

System Design

In this chapter we will explain architecture and design of the project

4.1 User Interface

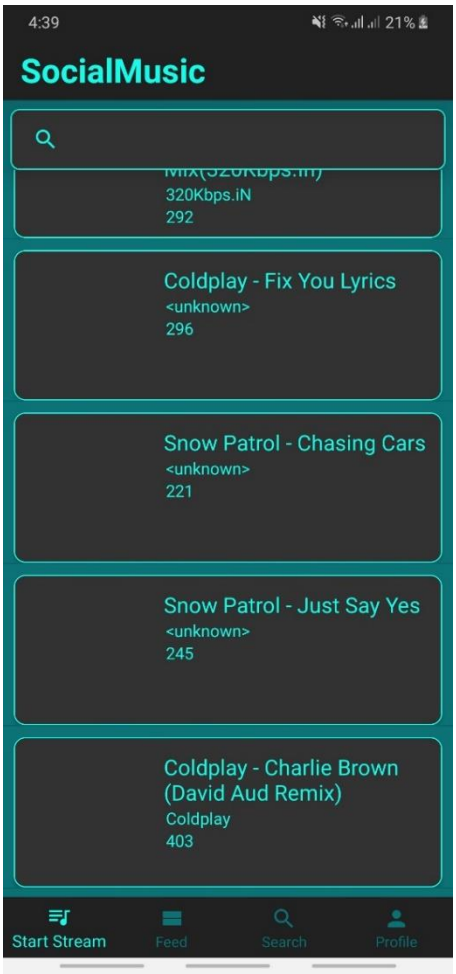


Figure 4.1 Song List UI

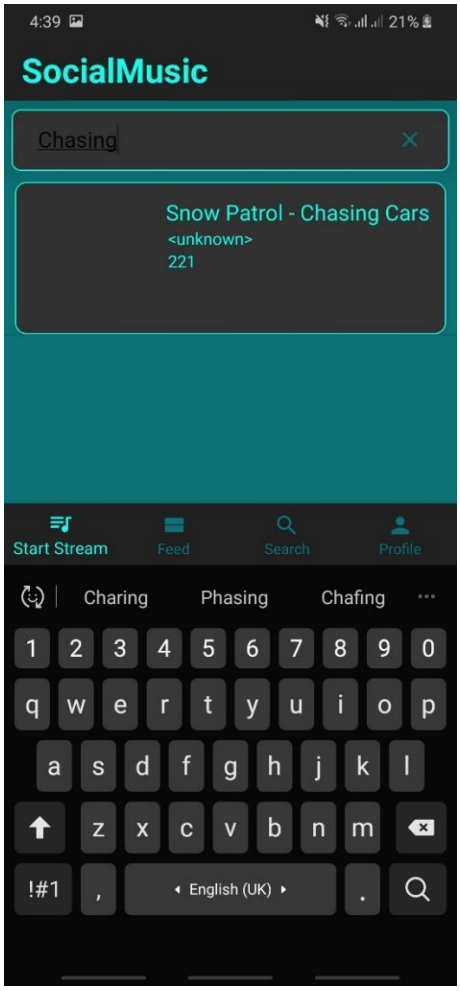


Fig 4.2 Song List Search

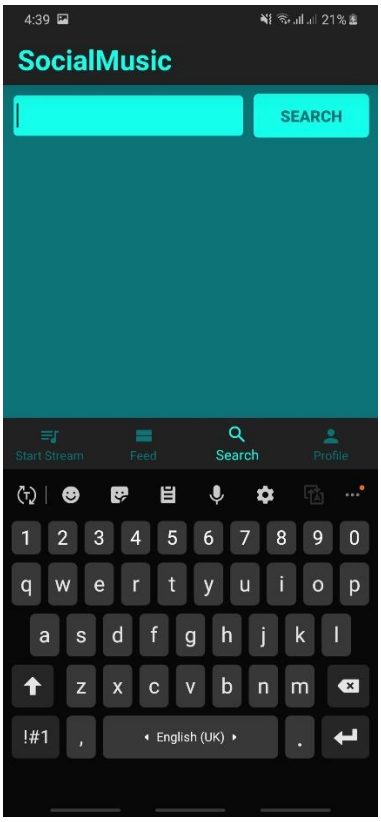


Fig 4.3 Profile Search

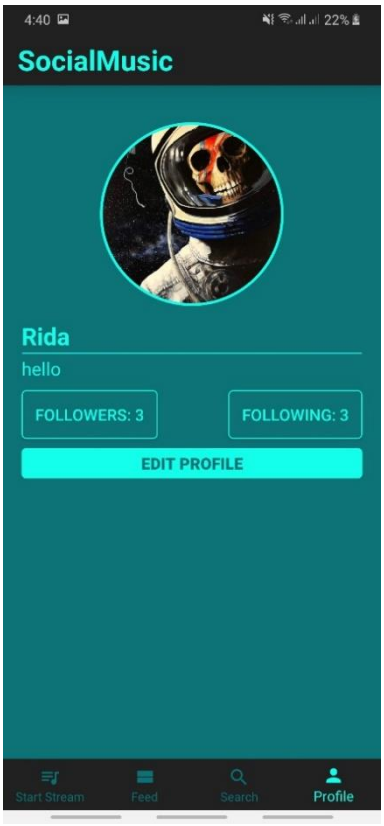


Fig 4.4 Profile View

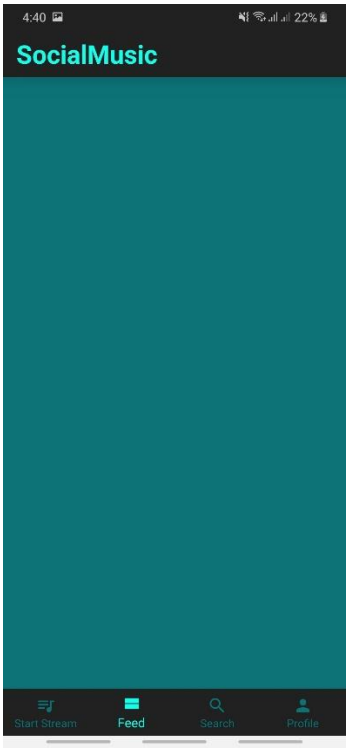


Fig 4.5 Available Streams

4.2 Low Level Design

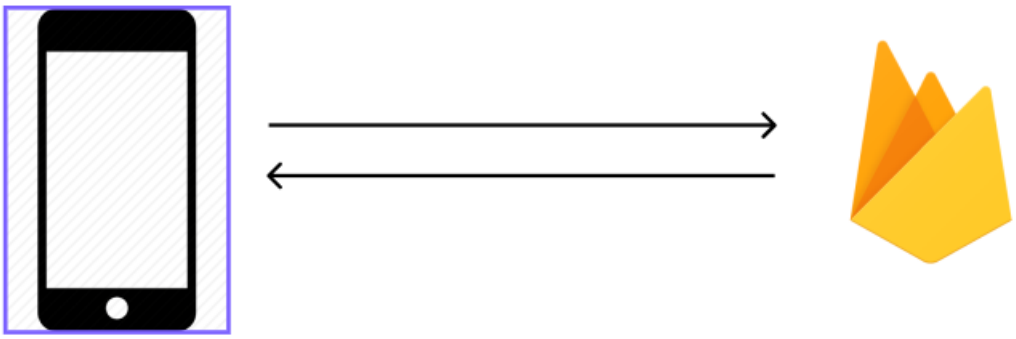


Figure 4.6 Low Level Design

4.3 High Level Design

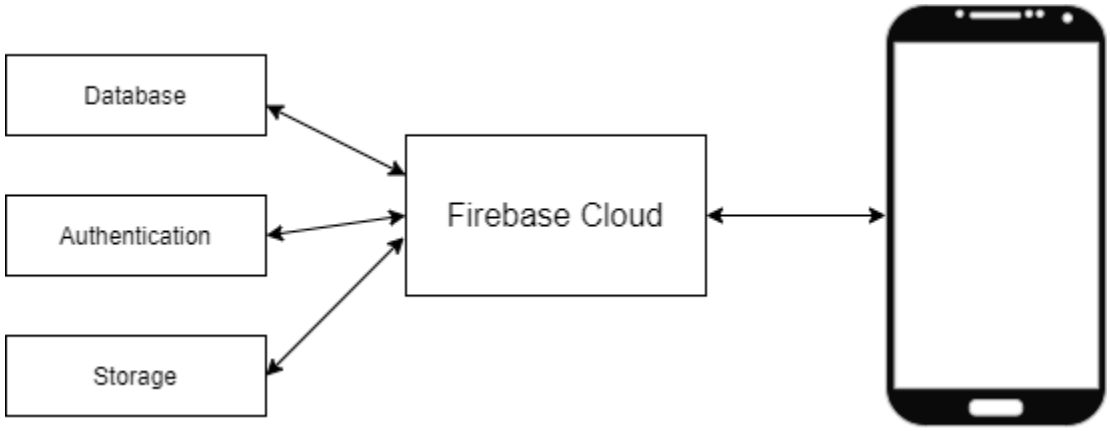
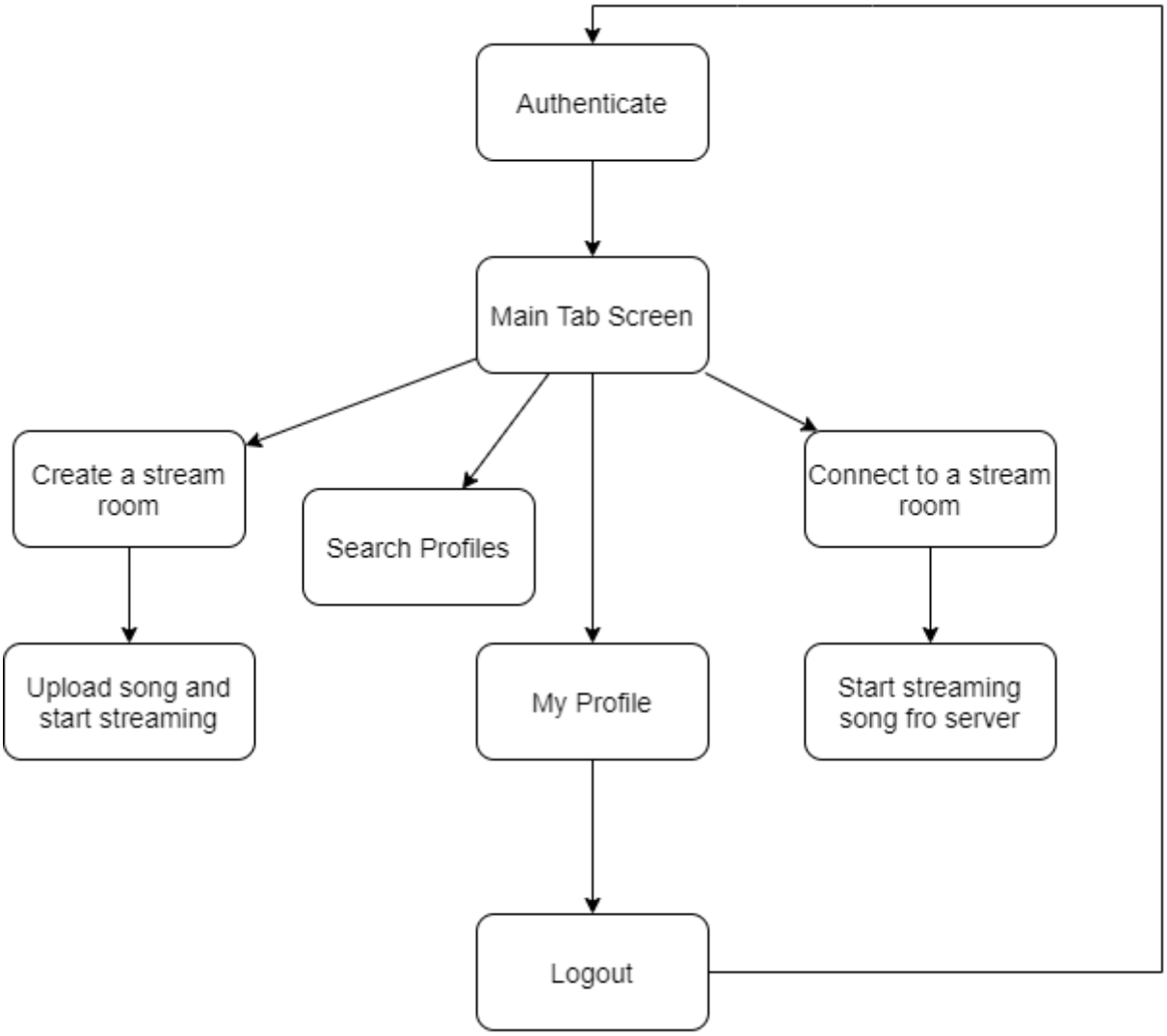


Figure 4.7 High Level Design

4.4 Sequence Diagram



Chapter 5

System Implementation

This chapter will focus on the implementation of the system and the tools and technologies used to create it.

5.1 System Architecture

System Architecture explains overall architecture of the system, which comprises of the overall system components i.e. external and internal. Our system contains one main component which is "Web Application".

5.1.1 Mobile Application

Through the mobile application users will sign up using their email address. The users will be able search for their friends and follow them. The users can either start a song stream or join a stream that is being played by their friends.

5.2 System Internal Components

5.2.1 SignUp

Users will be able to register a profile using their email addresses.

5.2.2 Login

Users will be able to log on to their profiles through a login form.

5.2.3 Search Users

Users can search for their friends or other users in the search tab of the application dashboard.

5.2.4 Follow/Unfollow

Users can follow or unfollow other users after searching for their profiles. Only the streams of users that you are following will be available in your feed tab.

5.2.5 Edit Profile

A user can edit his profile name and bio from the edit profile section in his profile.

5.2.6 Start Stream

Users will be able to start a stream searching and clicking on a song that is shown in the start stream tab. On clicking the song file will be uploaded to firebase storage (if not already uploaded) and then streamed.

5.2.7 Join Stream

Users will be able to join a stream that is online in the feed tab of the application.

5.2.8 Chat

Users connected to the same stream can chat with each other.

5.3 Libraries and Dependencies

- Glide (To download and set images)
- Firebase UI (To fill recyclerView from database)

5.3.1 Firebase Realtime Database

A cloud-hosted NoSQL database used to store and sync data (audio timings) between users in realtime.

5.3.2 Firebase Authentication

The backend used to authenticate user logins.

5.3.3 Firebase Cloud Storage

The storage service used to store the music data for the application.

5.3.4 Firebase UI

To fill recyclerView from database

5.3.5 Glide

To download and set images.

Chapter 6

System Testing and Evaluation

6.1 Graphical User Interface Testing

The GUI of the system is simple to use. Users can easily navigate between all functionalities including a clearly marked login button, prominent search bar, and a bottom navigation bar that makes it simple to move between searching for a song, viewing their profile or feed, or starting a stream.

6.2 Usability Testing

No technical experience is required to use this application as it is user friendly. The application shows a clear display of options for users to control it.

6.3 Exception Handling

Some modules of the application have integrated exception handling so that the project runs smoothly.

6.4 Test Cases

6.4.1 Test Case I: Non-matching password

Test Case ID	SCBU-T1
Title	SCBU-02
Description	The user will enter their information to create an account.
Pre-condition	The user is not already registered.
Steps	• The user fills the name, email, password, and confirm password fields. • The user enters Submit
Expected Result	• Error displayed: "Passwords don't match" • Asked to reenter password
Status	Pass

6.4.2 Test Case II: Account Already Exists

Test Case ID	SMBU-T2
Title	Account Already Exists
Description	The user will enter their information to create an account.
Pre-condition	The user is not already registered.
Steps	• The user fills the name, email, and password fields. • The user enters Submit
Expected Result	• Error - User already exists
Status	Pass

6.4.3 Test Case III: Incorrect password

Test Case ID	SMBU-T3
Title	Incorrect password
Description	The user will enter their credentials to log in to the system.
Pre-condition	User must be registered.
Steps	• The user fills the email and password fields. • Email and password are verified against the database.
Expected Result	• Error - Incorrect Password
Status	Pass

6.4.4 Test Case IV: Non-existing Account

Test Case ID	SMBU-T4
Title	Incorrect password
Description	The user will enter their credentials to log in to the system.
Pre-condition	User must be registered.
Steps	• The user fills the email and password fields. • Email and password are verified against the database.
Expected Result	• Error - Account doesn't exist
Status	Pass

Chapter 7

Conclusion

The Social Music application allows users to share their music among their followers by creating a live stream of whatever they are listening to. This application has been completed to the best of my abilities and has been a great learning point for me. However, it can still be improved greatly.

7.1 Future Improvements

This is just a testing version of the application that includes a simple GUI and not that many functionalities. The system can be further improved by:

7.1.1 Better GUI

The GUI can be further improved for a better user experience, one that is more visually appealing.

7.1.2 Better Audio Synchronisation

Song play for a stream can be further optimized so there is no delay between playback.

7.2 Learning Outcomes

Developing this project has tested the abilities that I have learned during my degree. I have practically implemented the software development life-cycle by gathering the required information and planning the development. I also learned how to integrate the Firebase database with the front end of my application in

real time. This project has been a great way for me to showcase what I have learned throughout my Computer Science degree.

Chapter 8

References

[1] Firebase. (2019). Firebase Authentication | Simple, free multi-platform sign-in. [online] Available at: https://firebase.google.com/products/auth/ [Accessed 1 Oct. 2019].

[2] Firebase. (2019). Cloud Storage for Firebase | Store and serve content with ease. [online] Available at: https://firebase.google.com/products/storage/ [Accessed 1 Oct. 2019].

[3] Firebase. (2019). Firebase Realtime Database | Store and sync data in real time. [online] Available at: https://firebase.google.com/products/realtime-database/ [Accessed 1 Oct. 2019]./

[4] Android Developers. (2019). Meet Android Studio | Android Developers. [online] Available at: https://developer.android.com/studio/intro/?gclid=Cj0KCQjw8svsBRDqARIsAHKVyqHjQE2M8_uQouc3xDPdrxdnHVCH_nlu_NnS5QsYj02lcc2ZKq_f3xMaAlqFEALw_wcB [Accessed 1 Oct. 2019].