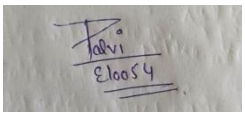
Design and Modeling report

Submitted in partial fulfillment of the requirements for the award of degree of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted to:Palvi Sharma
Project Teacher

Submitted By:
Student Group

NAME : Rishabh Kumar Singh
UID: 19BCS1184
NAME : Aman Raj
UID: 19BCS1182



Mentor Signature
(Name & E-code)

DEPARTMENT OF COMPUTER SCIENCE & ENGG

Chandigarh University, Gharuan

Feb 2021

INTRODUCTION

This report contains designs and models of the android music player, along with the status report. Furthermore it contains UML diagrams like data flow diagrams (Level 0, Level 1) and use case diagram. UML diagrams have been prepared on draw.io and are uploaded to github repository as well.

All the designs have been prepared on adobe XD and final XD file has been uploaded to github repository. This file can be used to simulate the app.

Libraries for App

1. **Room** – It is used to perform database operations for storing information into user's device.

   To use this library we have to add following to our app level build.gradle file:-

   def room_version = "2.2.5"

   implementation "androidx.room:room-runtime:$room_version"

   kapt "androidx.room:room-compiler:$room_version"

2. **Coroutine support for Room** – It is an additional dependency of room library to add coroutine support to room. Using coroutines we can perform database operations effectively in background thread.
   To use this library we have to add following to our app level build.gradle file:-
   implementation "androidx.room:room-ktx:$room_version"
3. **ViewModel and Lifecycle Extensions** – It provides us with LiveData which survives configuration changes of device as well as livedata provides data to our UI. It also provides viewModel scope; a coroutine scope to run livedata operations in lightweight process.
   To use this library we have to add following to our app level build.gradle file:-
   implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.0'
   implementation 'androidx.lifecycle:lifecycle-extensions:2.2.0'
4. **Coroutine** – It provides coroutine support for our app to run in different threads.

To use this library we have to add following to our app level build.gradle file:-

implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.3'

implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.3'

5. **Recycler View** – It provide efficient way to populate list items into UI by recyclying the unused view. Thus improving performance of app. It is recommended way to display list items into android.

   To use this library we have to add following to our app level build.gradle file:-

   implementation 'androidx.recyclerview:recyclerview:1.1.0'

6. **MediaStore** – It is inbuilt library to fetch Media Files from user's device.

   To use this we must add below permissions into our Androidmaifest.xml file:-

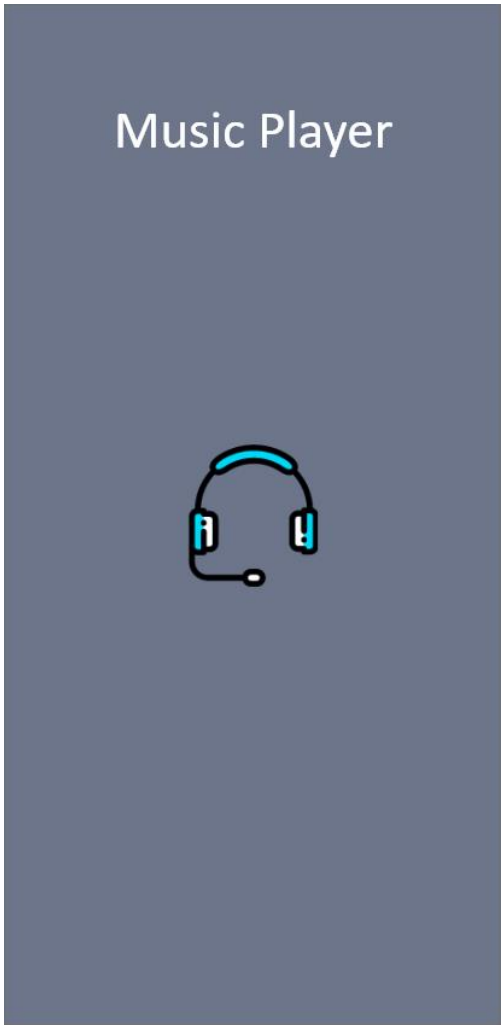
   READ_EXTERNAL_STORAGE

   WRITE_EXTERNAL_STORAGE

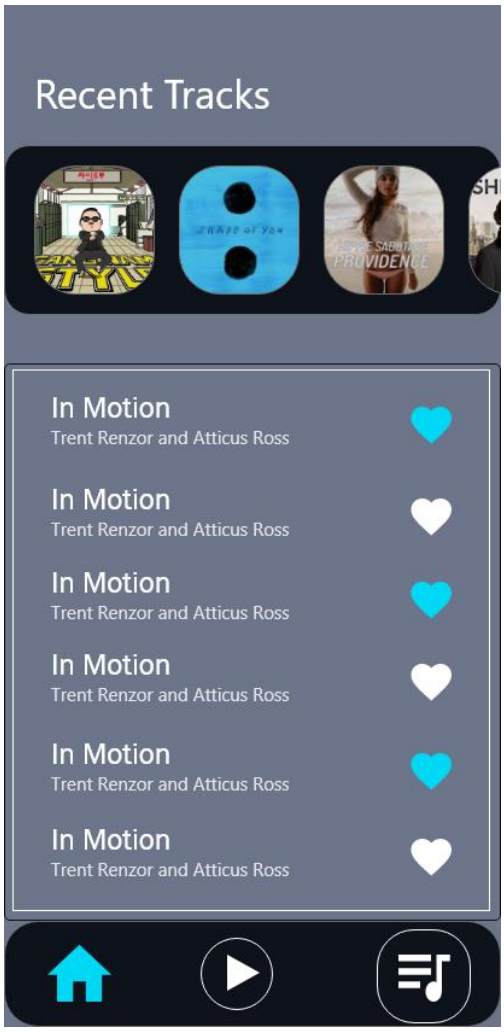
7. **MediaPlayer** - The Android multimedia framework includes support for playing variety of common media types, so that you can easily integrate audio, video and images into your applications. You can play audio or video from media files stored in your application's resources (raw resources), from standalone files in the filesystem, or from a data stream arriving over a network connection, all usingMediaPlayer APIs.
8. **GestureDetector** - Detects various gestures and events using the supplied MotionEvents. The OnGestureListener callback will notify users when a particular motion event has occurred.

SCREEN DESIGNS

Splash Screen

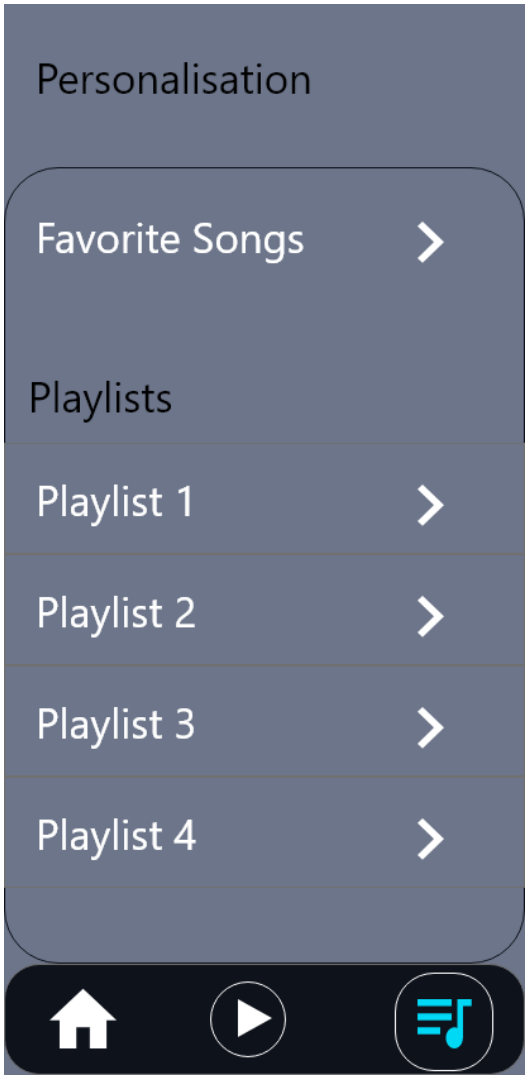


Home Screen

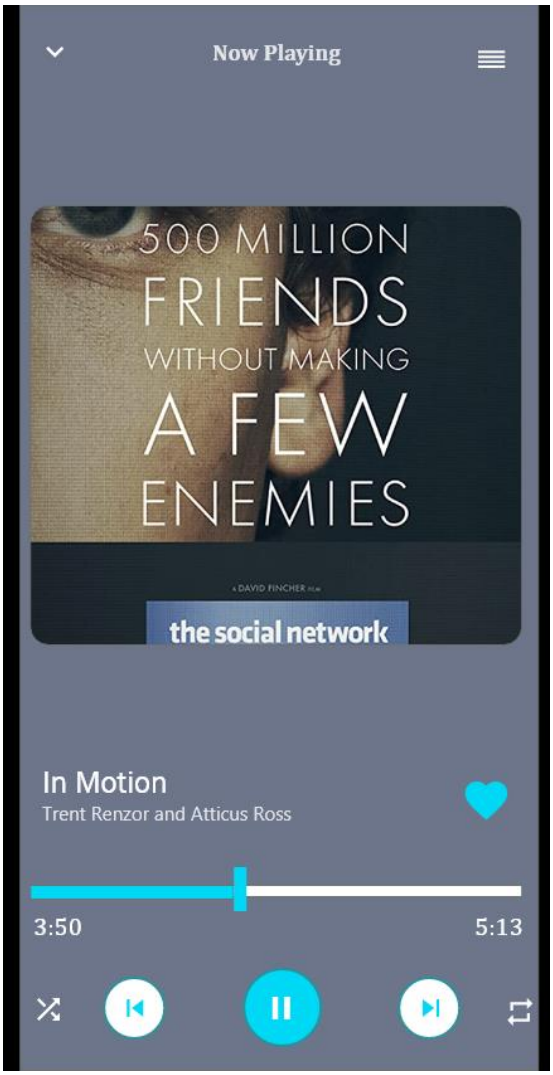


- Splash screen contains name of the app and icon.
- Home screen includes recent tracks, all songs and 3 buttons at the bottom of screen.
  Buttons are for selecting between home screen, now-playing screen and playlists screen.
  Individual song can be added in favorite by pressing heart button on the song or can be added to new playlist by a long press on the song.
  Clicking on the recent songs direct user to the now-playing screen.

Playlist Screen

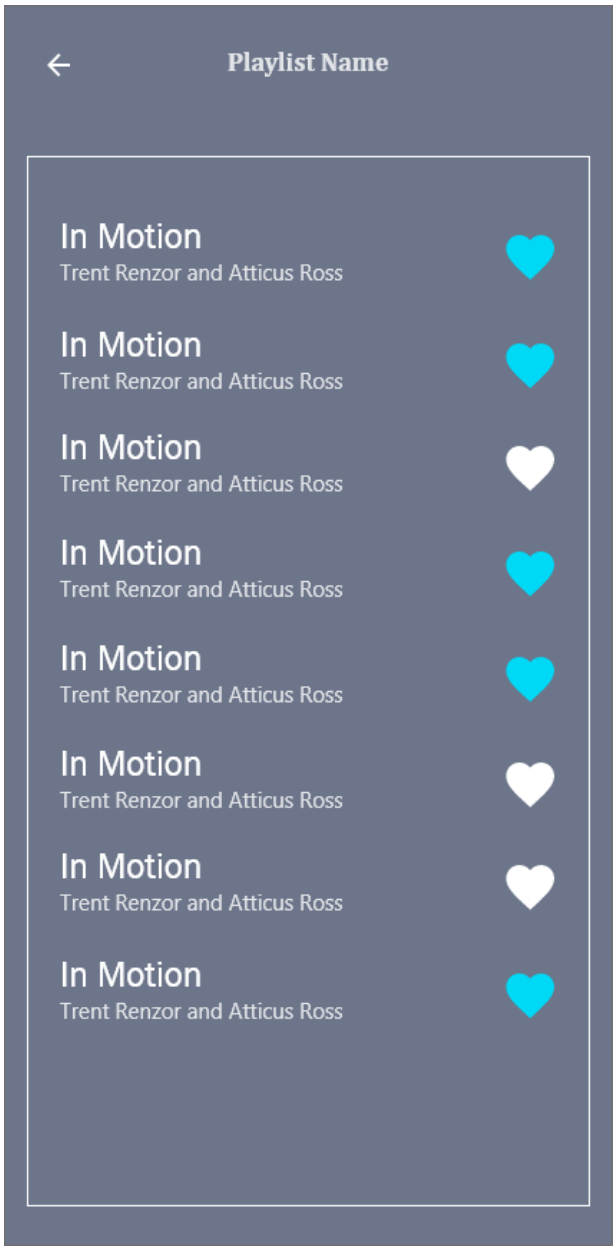


Now Playing Screen



- Playlist screen includes a default playlist called ‘favorite songs’ and user created playlists. Clicking on any playlist will direct user to single playlist screen.
- In now-playing screen, there are multiple controls available to user.
  Drop down option on top left will redirect user to home screen again while homeActionBar on top right will redirect to current playlist being played.
  Play/Pause, next, previous, shuffle, loop options are available at bottom.
  Current song can be added to favorites as well. Seekbar tracks the progress of song.

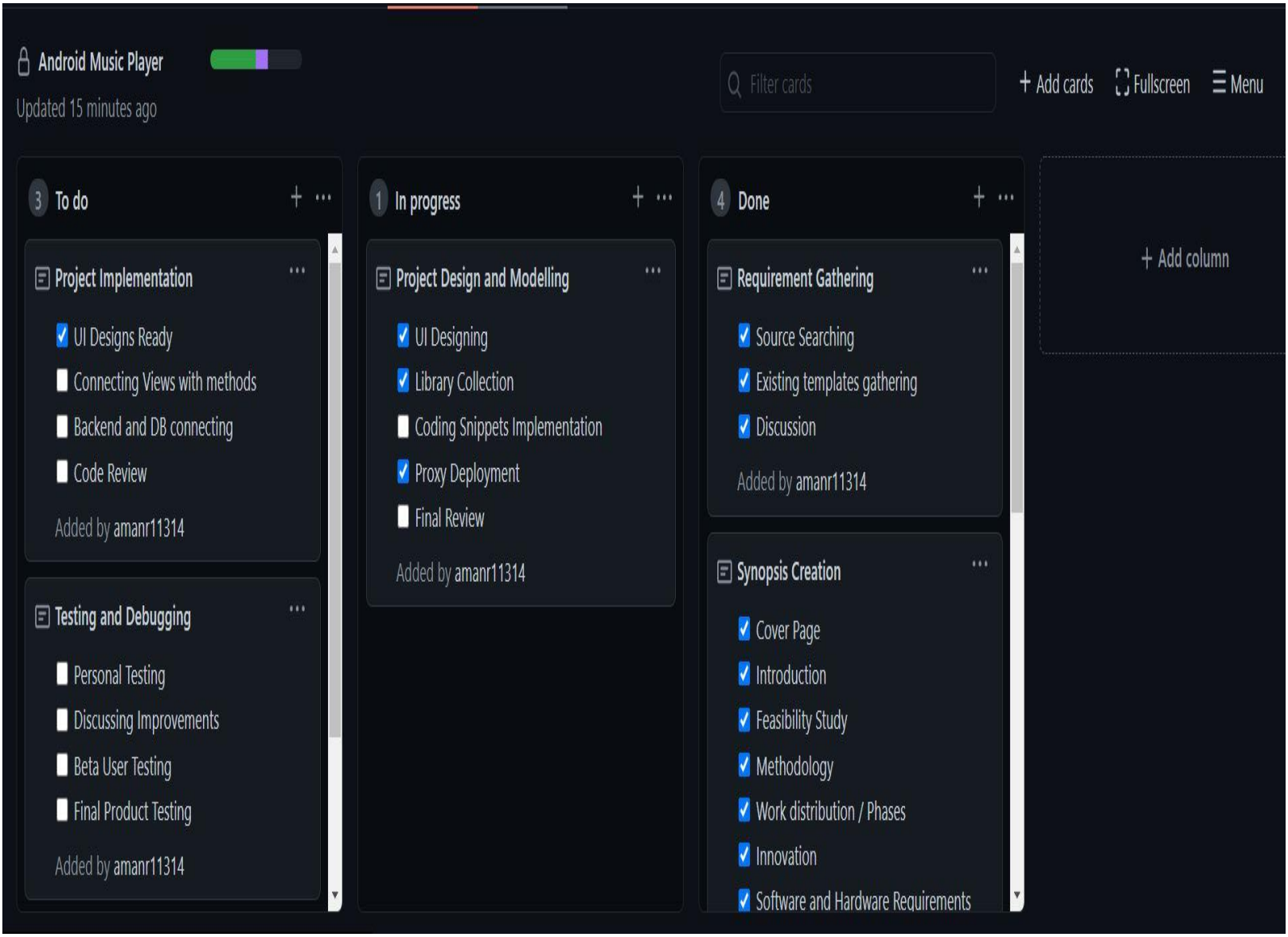
Single Playlist Screen



- It contains list of songs in a single playlist. Each item has song name, artist name and a button to add to favorites.
  There is back button on the top which redirects user to home screen.

STATUS

Status of the project can be traced in github project under the mentioned project.

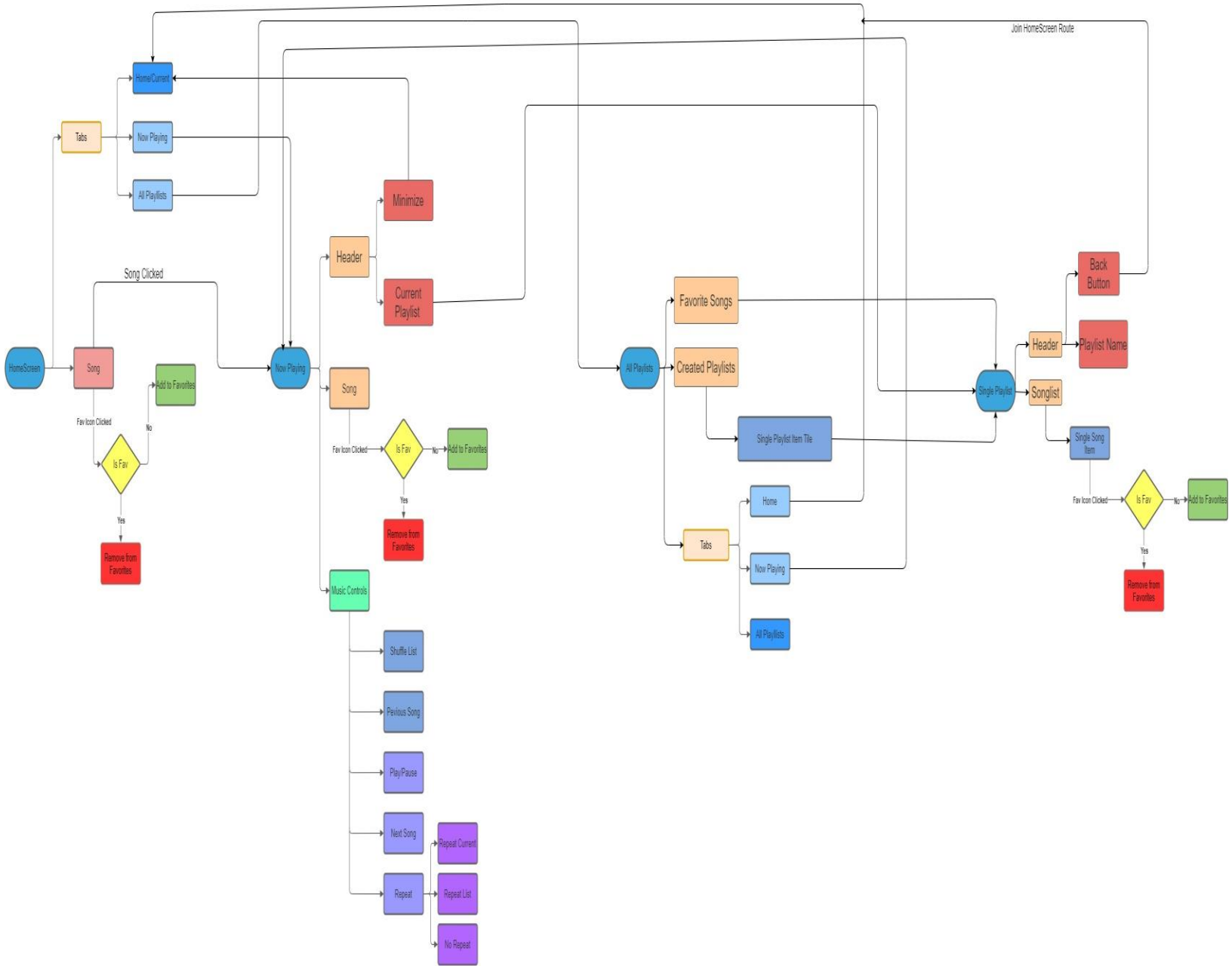


As per the status, project has completed requirement gathering, requirement analysis and synopsis creation and is currently in design and modeling phase.

Designs have been prepared for each screen and basic layout of app is implemented on Adobe XD. XD file can be used to simulate the app and has been uploaded on github repository.

App Flow

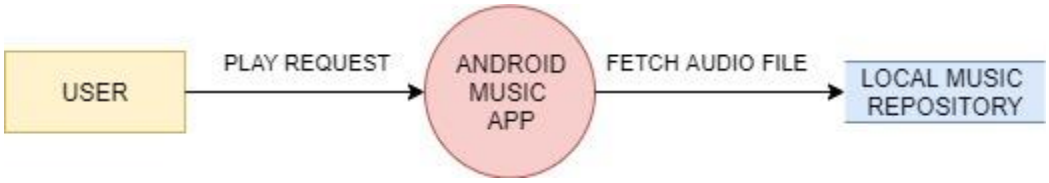
App flow can be depicted using a flow chart. The flow chart is prepared using draw.io and a copy of file has been uploaded to github repository for the same project.



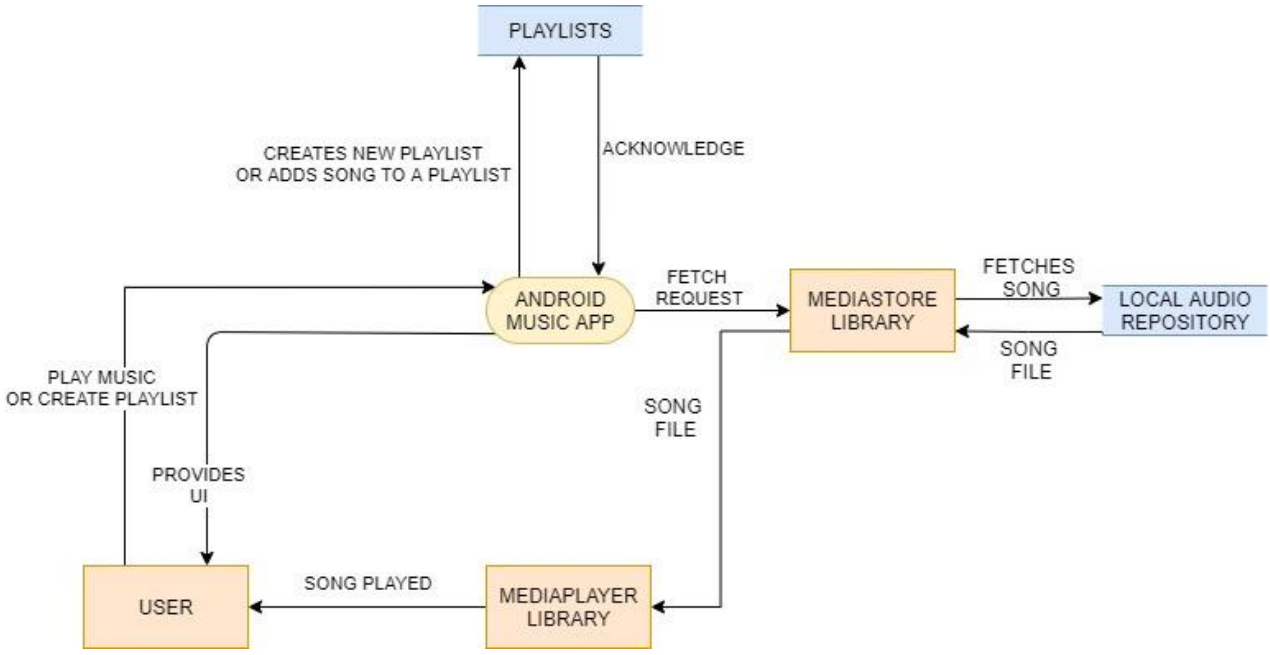
UML diagrams

Several uml diagrams has been created for the designing and modeling phase. This includes multi-level data flow diagrams and use case diagrams.

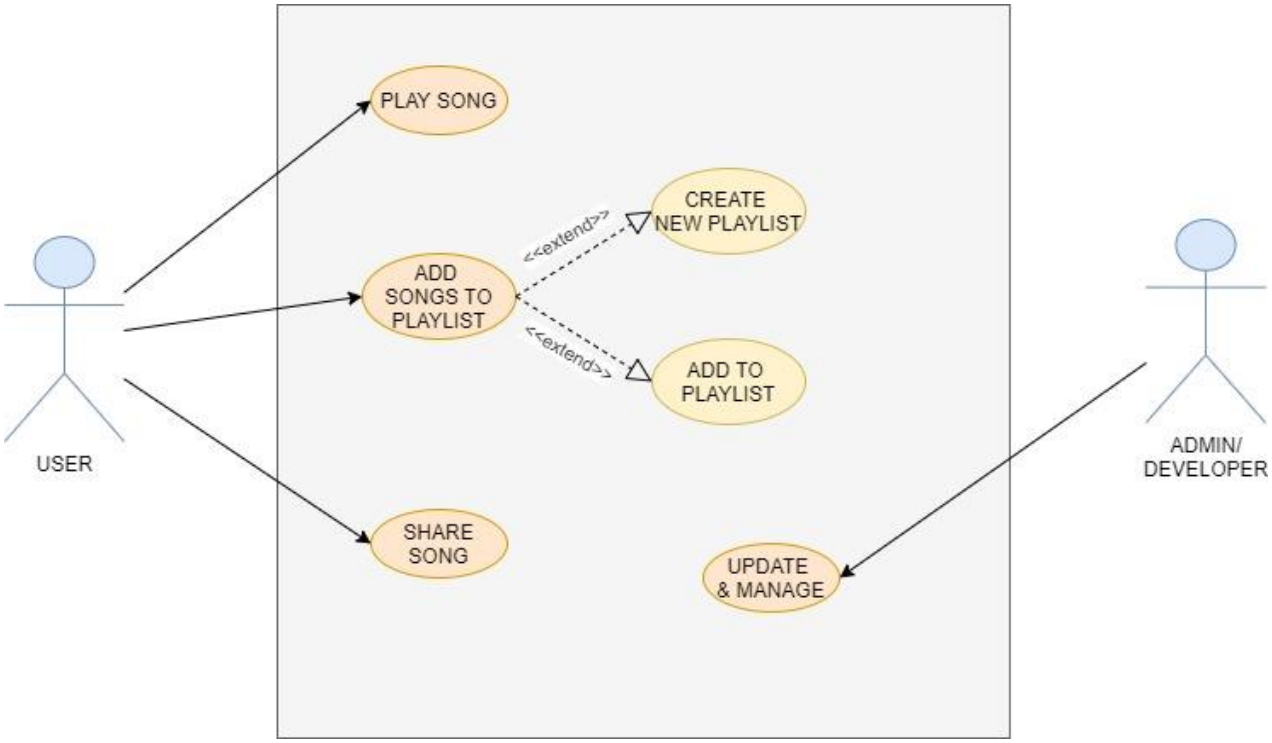
Level 0 DFD



Level 1 DFD



Use Case diagram



Appendix

Github repository link:
https://github.com/Chandigarh-University-students/Music-Player

Softwares:

- Adobe XD (for xd file and simulation of first prototype)
- Draw.io (For uml diagrams)