

How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you’ll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: Your GitHub username here

Popular Music

Description

Want to know about the popular music trends in your area, then popular music is the app you’re looking for. It provides trending/popular music and through smooth animations and swipe controls make this player most convenient music player. The interface is design specifically for Android™, making it intuitive and simple to use.

Intended User

This app is for music lovers, who want to keep upbeat with the new music.

Features

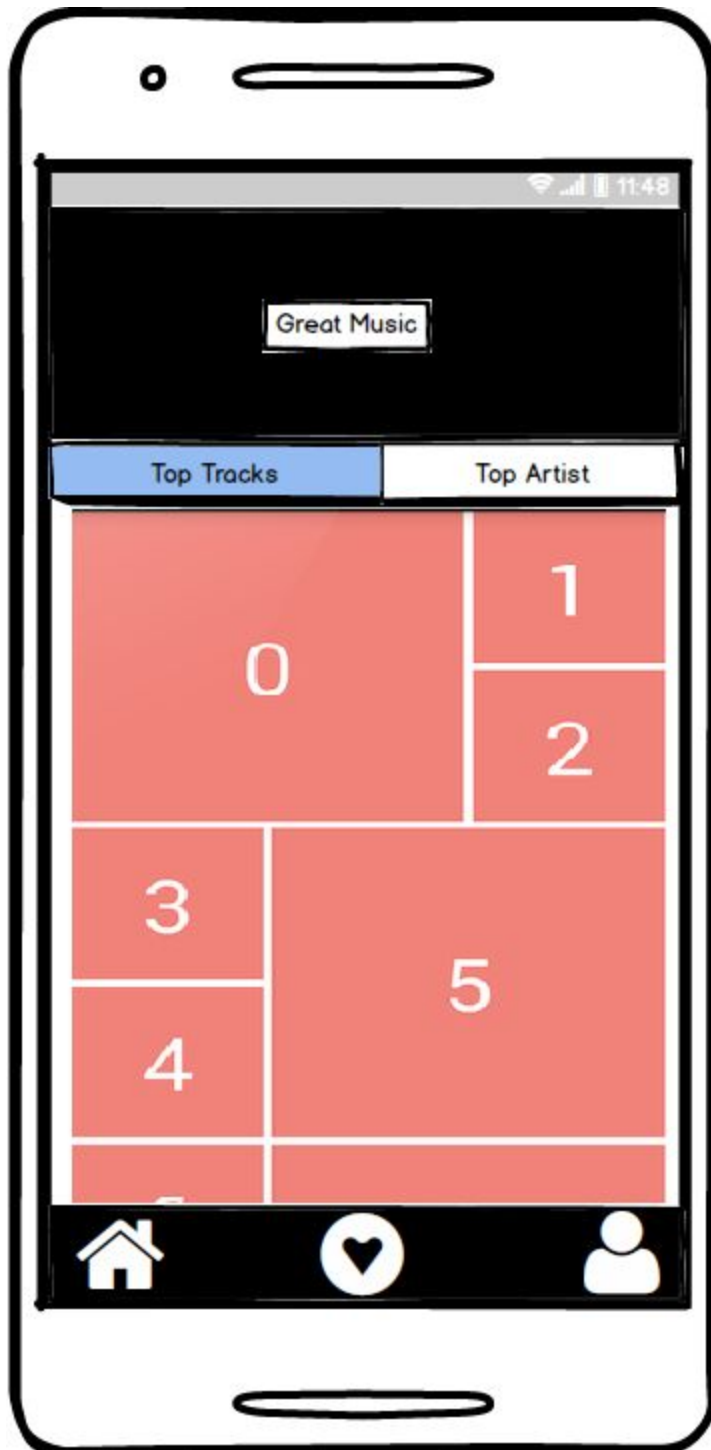
Main features include:

- Always updated with popular music
- A complete music player
- User favourite and most played tracks

User Interface Mocks

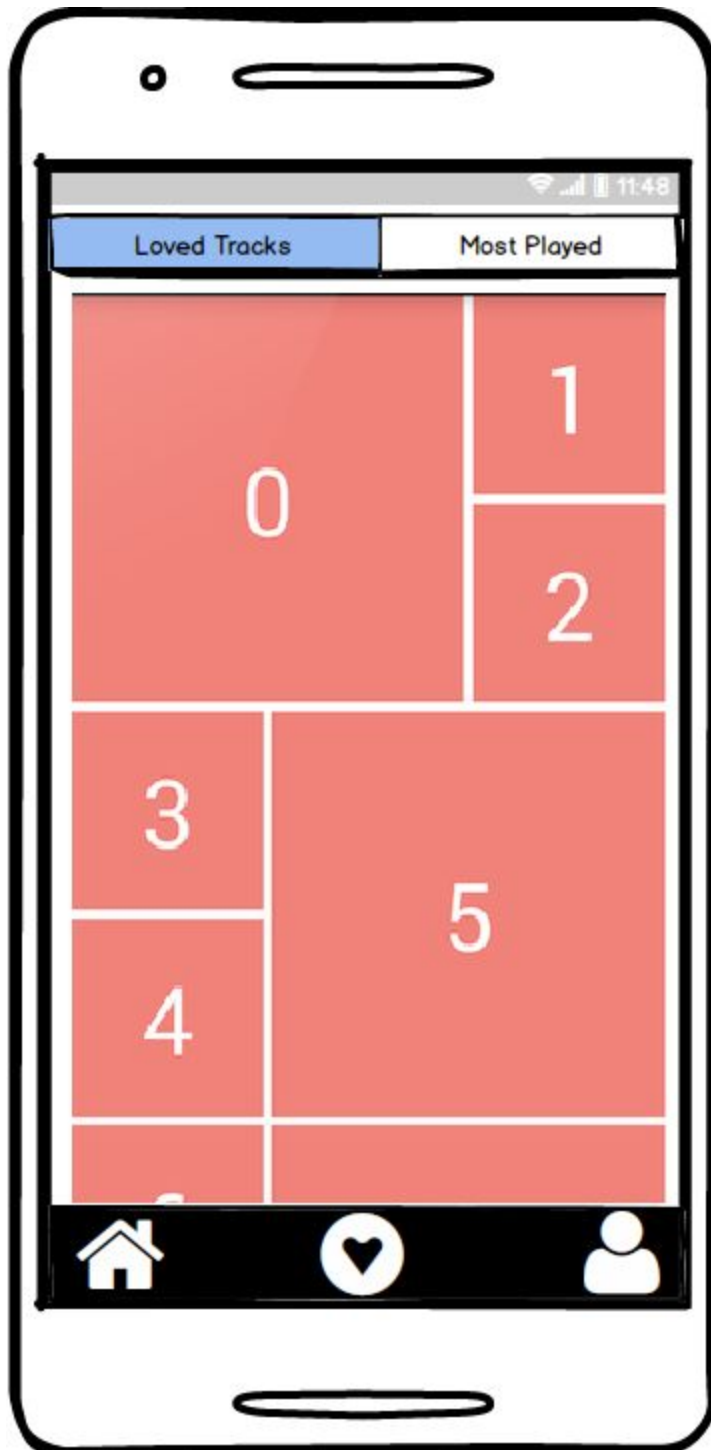
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



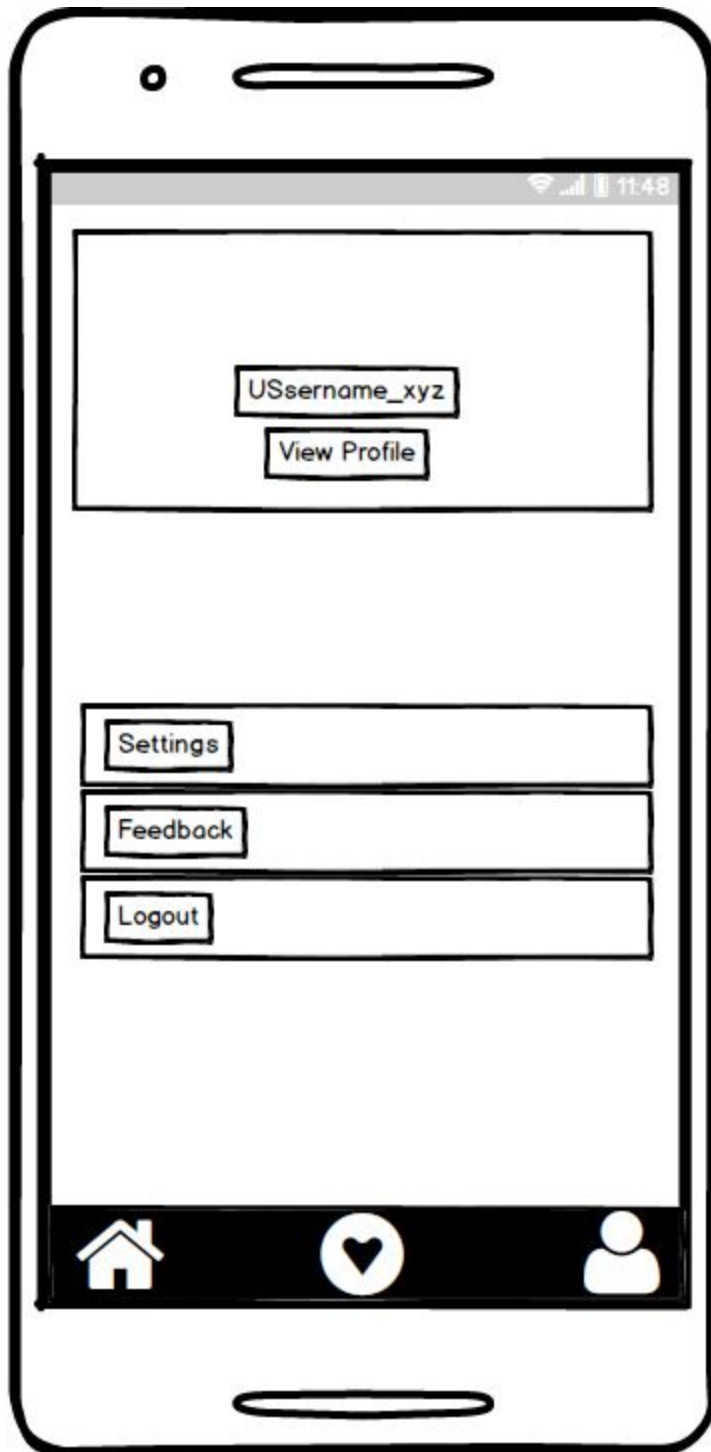
The home activity presented after login through last.fm api.
It is using asymmetric grid view to show the top tracks or top artist

Screen 2



Screen 2 - it comprises of the user favorited tracks and user most played tracks in asymmetric grid view

Screen 3



Screen 3 - It's the profile pf the user showing username, settings, feedback and logout.

Key Considerations

How will your app handle data persistence?

There should be a syncing service and syncadapter which pull data from last.fm api associated with the particular user on regular interval and store it in app SQLite db.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

Describe any libraries you'll be using and share your reasoning for including them.

Glide - for sharing and caching images.

Volley - to handle all the http request from the last.fm api

Describe how you will implement Google Play Services.

Google location service - to get the location of the user and use it to fetch top tracks w.r.t to user's location

Admob - to display ads

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Setup one module for the app

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for BaseActivity
- Build UI for music player activity
- Build UI for fragments home, favourites and profile
- Build UI for settings activity

Task 3: Your Next Task

- Implement Volley for http last.fm REST API
- Implement Google play services for both location and admob
- Implement Utils classes for common functions
- Implement class to handle error cases

Task 4: Your Next Task

- Implement database
- ContentProvider to access locally stored data
- Implement data syncing service

Task 5: Your Next Task

- Write all UI to pull data from the database
- Loader class to move data to its views

Task 6: Your Next Task

- Create a widget to show the top charting tracks

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"