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**"

GitHub Username: **sLakshmiprasad**

SongWiki

Description

This app helps you discover new music and find out more about your favorite songs and artists

Features:

1. Discover top charts
2. Artist biography
3. Search for artist, album or tracks
4. Save artist info for offline viewing
5. Attractive material design

Intended User

Music lovers who want to know more about their favorite songs and artists and also for those who want to discover new music

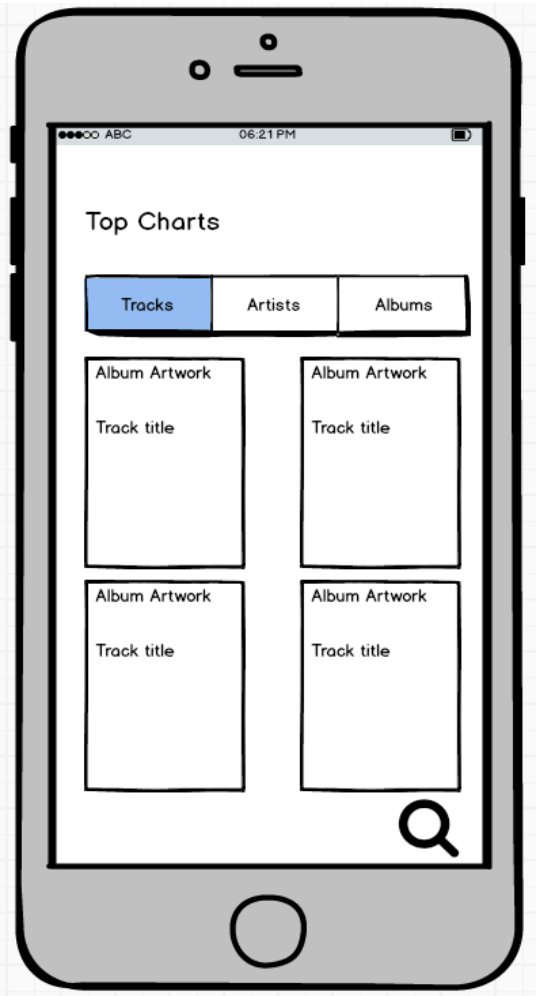
Features

List the main features of your app. For example:

- Search for artists, tracks and albums
- Top charts
- Save artist info for offline viewing

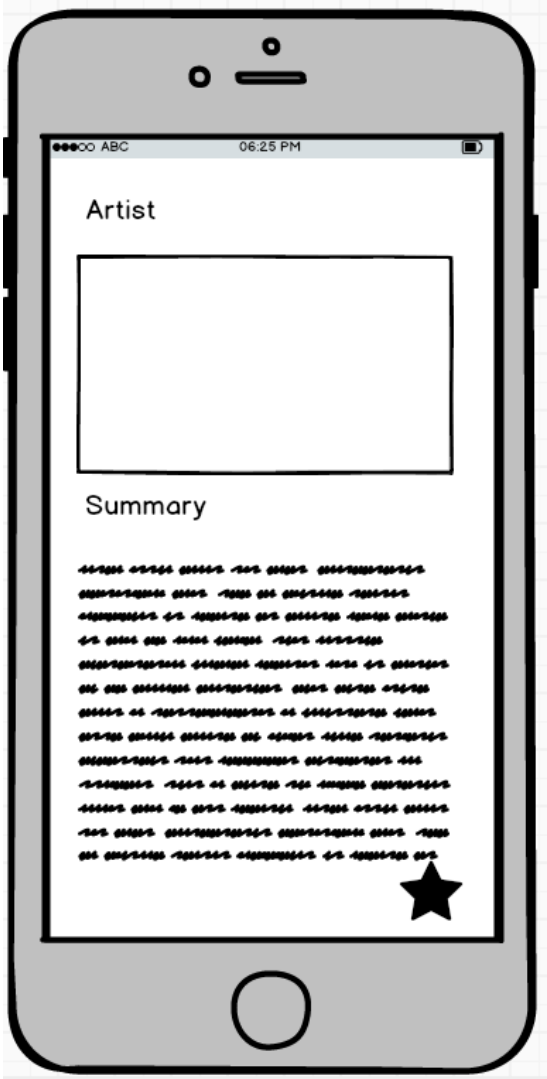
User Interface Mocks

Screen 1



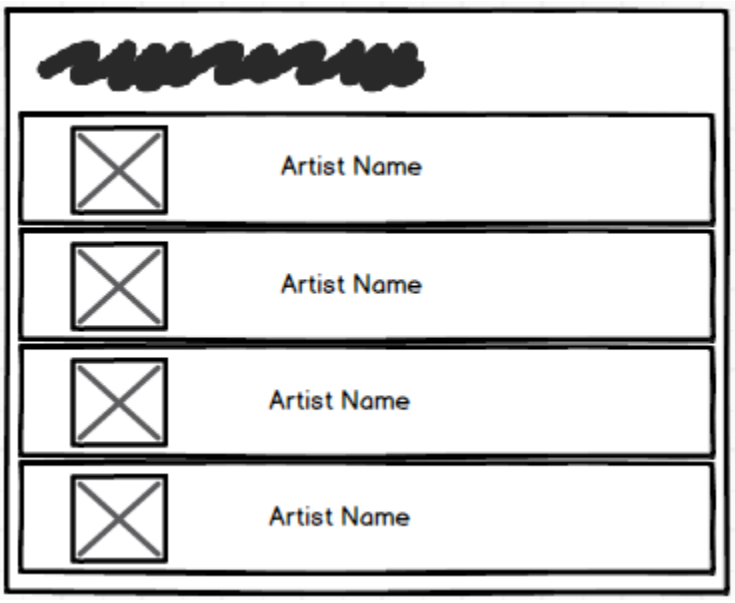
- This is the starting screen of the app which shows current top charts.
- It's a TabLayout with 3 tabs.
- On click of the item, the user will be redirected to the detailed activity.
- This screen also contains search feature using which the user will be able to search the corresponding item (e.g. - search for a song in TrackFragment)
- Similar to the above screen, there will be another activity which contains favorite artists whose information will be fetched from the database.

Screen 2



- The above screen mocks the ArtistDetailActivity which will be opened onClick of the artist from the MainActivity.
- This screen may contain the albums created by the artist and similar artists.
- Similar to the above screen, the app will contain TrackDetailActivity and AlbumDetailActivity respectively.
- This screen will contain other information provided by last.fm API
- Share functionality will be their which might share the respective last.fm url

Screen 3 - widget



The widget displays list of popular artists. On click of the list items, the user will be directed to ArtistDetailActivity

Key Considerations

How will your app handle data persistence?

New contentProvider supported by SQlite database will be created to save the artist info.

Describe any corner cases in the UX.

When the response from the last.fm API is null or invalid show appropriate message

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

Picasso for showing the display the images
Android support library
Butterknife for object injection etc.

Describe how you will implement Google Play Services.

Google analytics will be used which analyses the artist activity
Admod will be used

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

- Configure the libraries like Picasso, butterknife, support library and other libraries required for google analytics and admod
- Configure all the endpoints from last.fm API that is used by the app
- Design the manager class which has methods that returns the appropriate objects (List<Artist>) for the given input (artistName)
- Verify if all the methods are returning the correct objects

Task 2: Implement UI for Each Activity and Fragment

- Build UI for TopChartsActivity and the details activity
- Build UI for search functionality
- Build UI all the detail activities (track, artist and album)

Task 3: Google play Services

- Implement banner ads in the details activity
- Use Google analytics in the ArtistDetailActivity to get the artist name

Task 4: Implement the user flow

- Implement the flow MainActivities and the corresponding DetailActivities

- Use shared element transition for the same
- Implement search functionality
- Adding favorite artists' info to the database
- Displaying favorite artists in offline mode
- Build a widget to show the artists

Task 5: Error handling, accessibility and RTL layouts

- Handle the errors like invalid or no response from the endpoints
- Offline mode – show message using snackbar
- Add content description for the images and image buttons
- Design the layouts to support RTL layouts

Task 6: Unit Testing

- Test all the functionalities of the app in at least 2 devices
- Write unit test cases to verify if the services are working properly

Task 7: Build

- Add keystore to the project

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**"