

[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:** nishtha3006

# MusicMasti

## Description

So many times you would like to sing a song but can not understand the lyrics. Don't worry, MusicMasti is here to make your life easier by providing you a song with lyrics also so you can enjoy singing without abusing singers :p!!!

## Intended User

Any music lover who would like to sing songs but can not remember lyrics which is everyone.

## Features

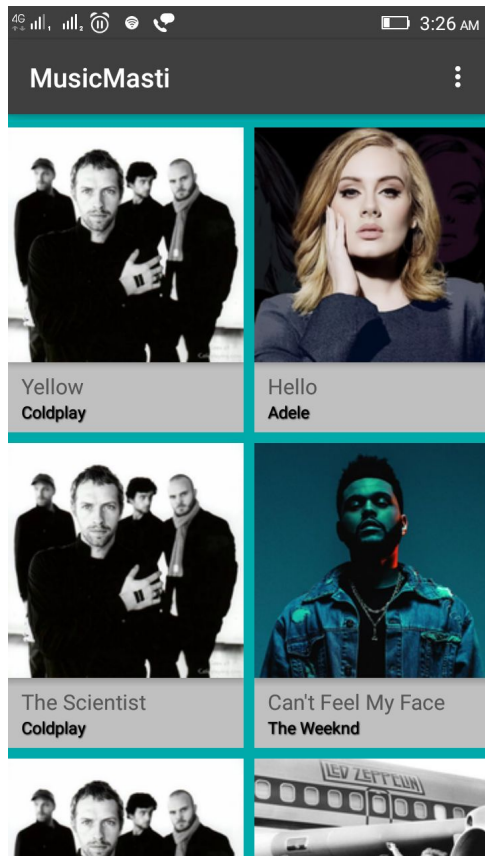
List the main features of your app. For example:

- Add to your favourite list
- List popular songs according to your area
- For each artist, list his/her top tracks

## User Interface Mocks

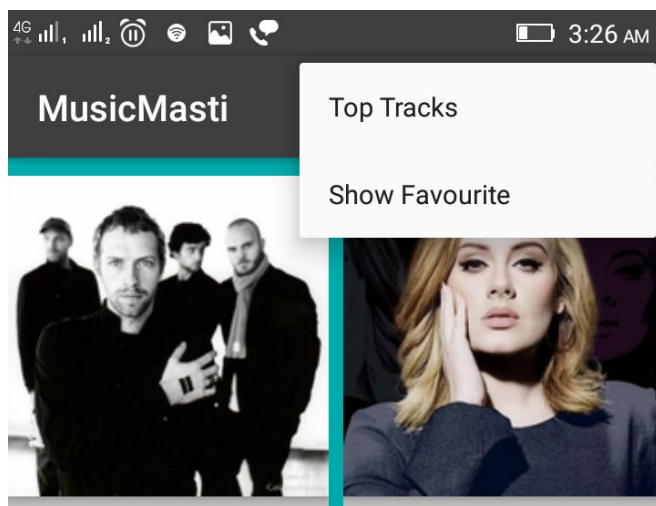
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



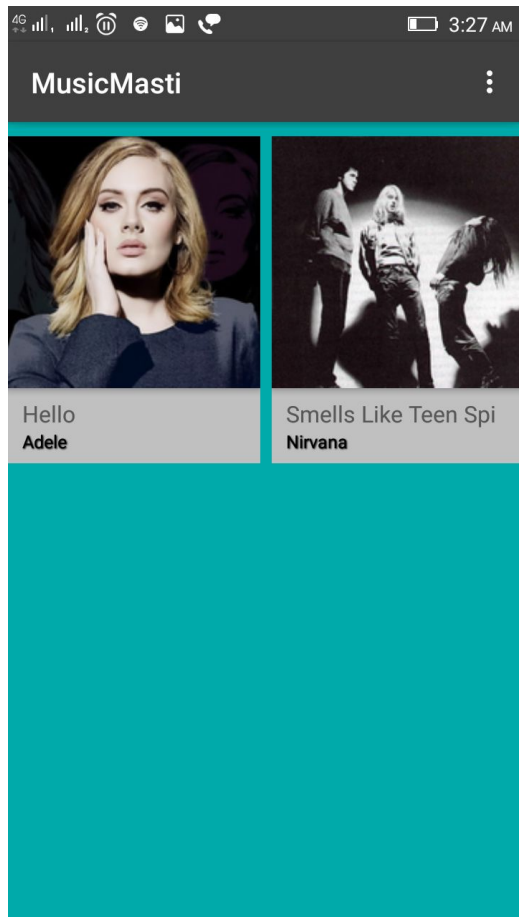
Main grid view screen which contains popular songs based on location

## Screen 2



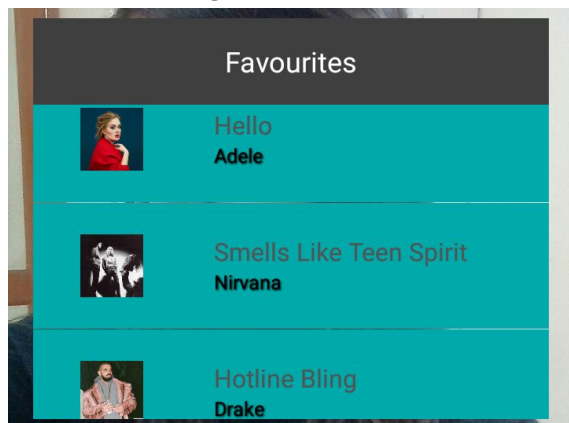
Menu item : Let user choose what to list(Top tracks or Fav songs)

## Screen 3:



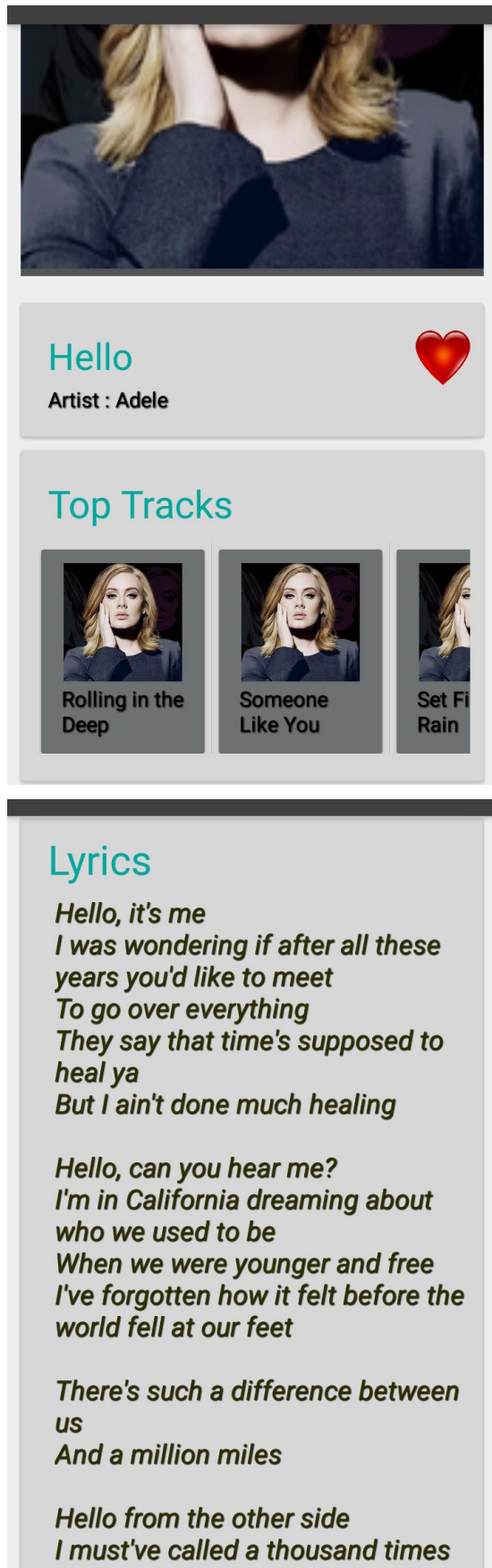
Favourite list UI : This screen will show the user all the songs, he/she added to the favourite list.

#### Screen 4 : Widget



Widget UI : A widget from which users will be able to consult updated info related to the favourite song he/she added.

#### Screen 5 : Details Screen



Detail Song UI : Shows the lyrics of the song and all the top tracks of that artist's. User can click those top tracks which will open the Last\_Fm api link of that track which includes youtube video of the song.

## Key Considerations

### How will your app handle data persistence?

Using content provider

### Describe any corner cases in the UX.

-- While loading the popular songs and their lyrics, due to some reason screen can be blanks so some default data should be there.

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

-- Picasso to load images  
 -- OkHttp and HttpURLConnection will be used to access the REST endpoints of the lastfm api to discover songs, and popular artists/songs in a location.

### Describe how you will implement Google Play Services.

-- AdMob: Display an ad banner within the app  
 -- Location: Locations will be used to list popular songs which are popular in the user's geographic location

## Next Steps: Required Tasks

### Task 1: Project Setup

- Configure libraries
  - Using maven jMusixMatch needs to be packaged into a jar file and imported into the project
  - OkHttp
  - Picasso

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Grid view to each different list : popular, favourite

### Task 3: Google Services

- Admob
- Google location service

#### **Task 4: Data Persistence**

- Create SQL database with different tables to save different category songs
- Loader to update UI with each search
- Implement cursor adapter to fetch the data easily.

#### **Task 5: Implement MusicMasti widget**

- The app must provide a widget from which users will be able to consult updated info related to the favourite song he/she added.

#### **Task 6: Configure app building**

- Configure app signing, including the keystore and passwords in the repository.