**GitHub Username**: salatza

# MusicSpot

## Description

It is musical app. It provides information about artists, albums and tracks.  The app is focused on entertaining the user and provides information about the music he loves.

## Intended User

The intended users of this app are people who love music of every age, gender and nationality.

## Features

- Displays information that the user chooses, through a search option for artists, tracks or albums
- Saves favorite tracks
- Plays the tracks selected by the user

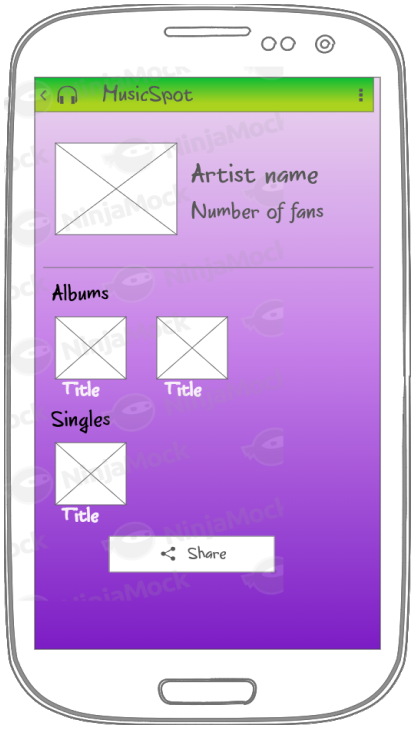
# User Interface Mocks

## Screen 1

In the main screen the user enter a search query and selects what he is searching for(artist, track,album). The list shows information about Artist, Track and album name and if any of these does not exist , becomes invisible in the UI. There is a menu option that shows favorite tracks saved by the user in a databse.


## Screen 2

This is a details activity that is different according to the user’s search criteria. The first activity launches when the user searches for an artist, the second for an album, the third for a track and the last if he selects for the menu to display his favorite tracks. The album and track UI allows the user to save the tracks in his favorite list. The play button to play the tracks using exoplayer. The user can also share in the first UI the artists web page, in the second the album’s web page , in the third the track and in the fourth his list.







On the right is the UI for the widget, that will display the list with the favorite artists and tracks of the user.

# Key Considerations

## How will your app handle data persistence?

The tracks,artists, albums selected by the user will be saved in a database and data will be handled with a Content Provider. Loaders will be used with Content Providers to move the data fetched to the views. IntentService will be used to move between main activity and details activity and also shareIntent service, to allow the user to share music data. The app will use content descriptions for the Search button in MainActivity, for Share buttons and play/pause buttons.

## Describe any edge or corner cases in the UX.

- When the screen is rotated the content should be preserved
- When the user closes the app exopalyer should be realsed
- When the app is paused exoplayer will still work
- ConnectivityManager and NetworkInfo will be used to ensure that the device is connected to the internet and avoid network errors
- When the user presses the back button the app behaves naturally,  exopalyer will still play as a background but it will not be allowed to play multiple songs at the same time

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

I will use Butterknife for binding views, Picasso to handle the loading and caching of images from the albums, artists, etc and Exoplayer to play the tracks.

## Describe how you will implement Google Play Services or other external services.

I will use Google mobile ads and I will try one of these as the second required service:System voice actions (the user can say his search criteria instead of entering his search query) or Cast Application Framework (CAF) for Android. If the second option appears to be infeasible, I will use as a backup plan, the Google mobile ads and the Google drive. The user will save his favorite list in Google drive.

# Next Steps: Required Tasks

## Task 1: Project Setup

App will be written solely in the Java Programming Language. I will use the 3.1 Android Studio version, 4.9 version of Gradle, version 2.71828 (2018-03-07) of picasso library, 8.8.1 version of butterknife, version r2.8.2 exoplayer,

- Configure libraries
- Get an API for a music app
- Coonect the RapidAPI with my app
- Add flavors(free with ads and paid without ads)

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

- Build UI for MainActivity
- Build UI for Details ActivityArtist
- Build UI for Details ActivityTrack
- Build UI for Details ActivityAlbum
- Build UI for ActivityFavorites
- Build UI for MainActivity list item
- Build UI for Details ActivityArtistAlbums
- Build UI for Details ActivityArtistSingles
- Build UI for DetailsActtivityAlbumListItem
- Build UI for ActivityFavoritesListItem
- Create menu options UI

## Task 3:

- Create classes for artist, track, album and main activity
- Create database
- Create activity to connect with Json  and fetch results for artists, albums and tracks
- Create MainActivity and DetailsActivity for artists, albums and tracks
- Create adapters for artists, albums ,tracks and favorites

## Task 4:

- Implement Google Play Services
- Create Widget that displays the favorites list
- Handle Error Cases

## Task 5:

- Check that all strings are kept in strings.xml file
- Fix the design(styles, colors, etc)
- Remove unused code or classes or activities
- Add support for RTL languages