Lab 3

CSE 438

Spring 2020

1 Warning

This assignment is SIGNIFICANTLY MORE DIFFICULT than the previous two assignments. You are receiving plenty of time and a partner (if you choose to have one) to work on the assignment with because the assignment's difficulty is much greater than the previous two assignments.

2 Setup

Navigate to the repository for this assignment and make a copy for yourself. Please include your first and last name in the repositories name to make grading easier. To submit your project, commit and push your code to the repository by the due date. Any pushes after the due date will be counted as late.

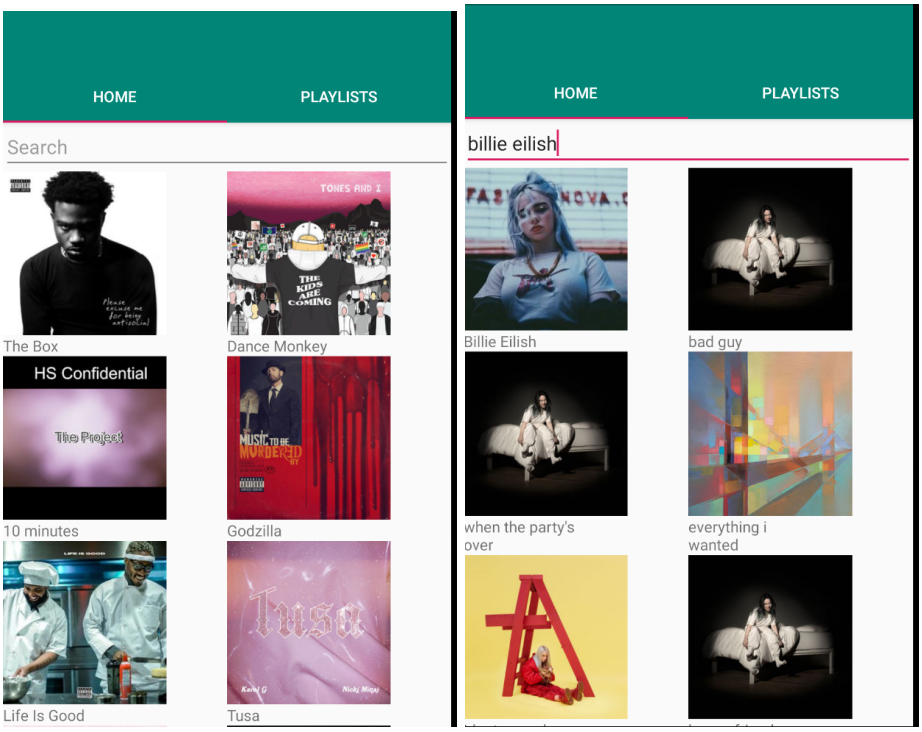
3 Introduction

After hearing this great song, you have been inspired to listen to more music. However, there is so much music, you cannot keep track of it all. As a talented computer science student and mobile developer have decided to try to make it easier for people to keep track of their music choices using an app.

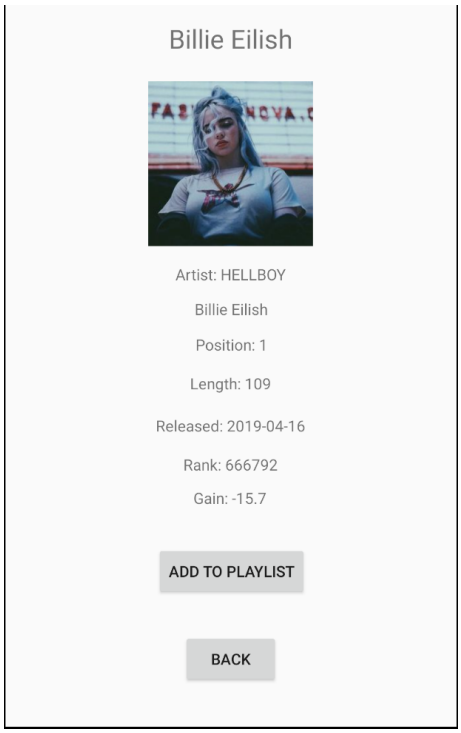
The app is an app that lets users search for music using the Deezer Api and lets the user create playlists that are saved to their phone. You will be using the Deezer Api to get information about various music and artists using RetroFit and you will use Room to manage a SQLlite database for the playlists.

4 Assignment

The app should begin by loading the main activity. This Main Activity should use a RecyclerView to display current popular music from the charts. The music should be displayed in a GridView using a RecyclerView as shown below. From this screen the user should also be able to search by BOTH artists and tracks and display the results in the same view. This should all be categorized in the Home Section of a TabView. An example of a search result and popular view is shown below.



When one of the displayed images is clicked on, the user should be taken to a screen that will give them information about the track and the option to add the track to a playlist. An example of the layout is shown below.

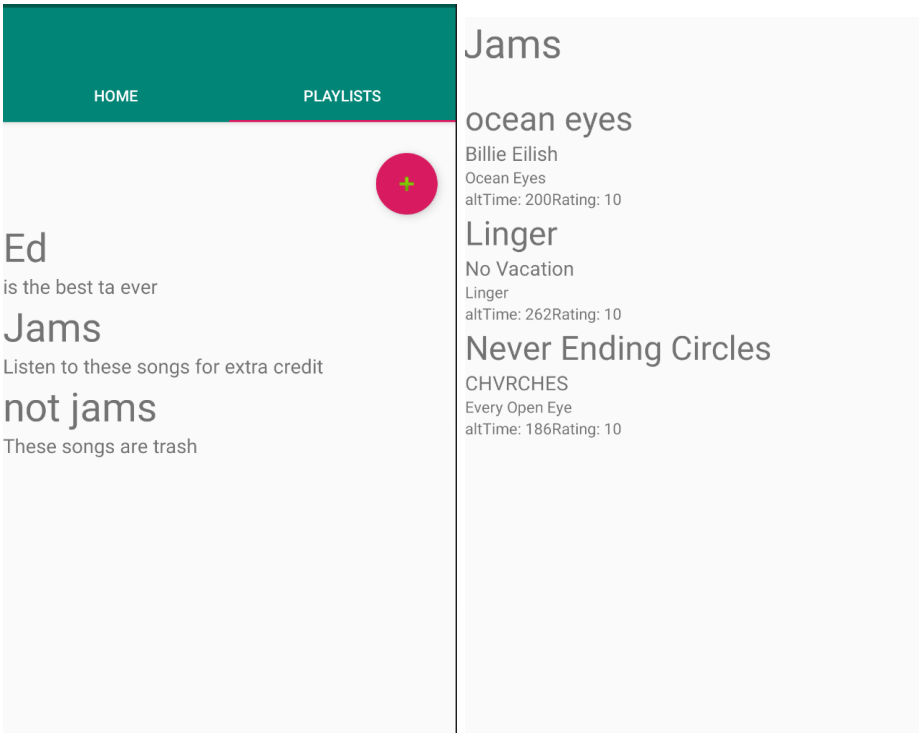


If the user clicks the Add To Playlist button, the user will then receive the option to add the song to any number of user created playlists. The user should be able to select a playlist they want to add the song to.

In the playlist section of the application, the user should be able to both add to and view a RecyclerView of playlists with descriptions. The user should be able to navigate to a form that lets the user create a new

playlist. Every playlist should have at least a Name, Description, Rating, and Genre.

When a playlist is clicked on, the user should be shown all of the tracks in the individual playlist. These tracks should include at least the track name, artist, playlist genre, time, and playlist rating. Getting the playlist genre and playlist rating for EACH song may require a SQL Join. The user should also be able to remove tracks from any playlist.



5 Displaying Images

Notice that the GridView and Track Information Activity use images that are returned from the API. We have NOT covered how to use image views and dynamically load images in this class. You will need to do some research on this. I would recommend using the Picasso Library for this as it keeps loading images simple and like RetroFit is supported by a well known company: Square.

6 Tips

- Make use of the RetroFit and Room as well as a Repositories and ViewModels when retrieving data. Note that ViewModels and repositories are made relative to data models and not network/database queries. For example, any calls involving a playlist object to either an API or the database should be encapsulated in a Playlist ViewModel and Playlist repository and calls for Tracks should be made from a Track ViewModel and Track repository. You will want to have more than one ViewModel on this assignment.
- Remember to sanitize and consider edge cases. No values should ever be null or blank.
- Observers and LiveData within ViewModels will be helpful in making the UI dynamic and avoiding blocking code.
- Draw out your database schema before you start creating your database. Make sure you have all the fields you need and that there are no repeat values in tables.

- The Moshi and Picasso libraries will likely be very helpful in parsing data and displaying images respectively.

7 Creative Portion

For every homework assignment, you will be asked to think of an additional feature to be added to the application that will improve the user experience and provide you an opportunity to learn about concepts that you are personally interested in. Put yourself in the shoes of your users: what features would they like to see in an app like this? Try to make it something new and substantially different from what the app already does - do not just rehash existing requirements.

When you submit your assignment, please include a ReadMe.txt file that explains your creative portion. You should explain what the feature is, why you chose to implement that particular feature, and how you went about implementing it.

To receive full credit, your feature needs to be substantial as compared to the rest of the assignment. You have an entire API, database, and operating system available to you. Use them and be creative! Examine the rubric below to get a feel for how much weight we are putting on the creative portion of the assignment.

8 Requirements

Total Points: 100

1. (10 Points) Creative portion
2. (8 Points) The UI uses LiveData in multiple ViewModels as well as observers to update the UI dynamically by calling Repositories to make both REST and SQL queries.
3. (8 Points) Users can search for tracks by both artist and track name using the Deezer API
4. (8 Points) Search results and chart information is displayed in a GridView with images
5. (8 Points) Users can view all of their playlists in a RecyclerRiew with descriptions of each playlist
6. (7 Points) Users can view the contents of their playlist in a RecyclerView with track name, artist, playlist genre, time, and playlist rating. This is done by joining two tables together to avoid data redundancy
7. (7 Points) Detailed information about a track including song name, artist, album, track position in album, track length, release date, and rank are present when a track is clicked
8. (5 Points) Retrieves and displays information about charts with RetroFit and the Deezer API when the app loads
9. (6 Points) Users can create and store multiple playlists
10. (6 Points) Users can add songs to playlist
11. (5 Points) Users can delete songs from a playlist
12. (5 Points) A tab view is used to toggle between Home and Playlist Views
13. (5 Points) Uses coroutines to avoid blocking calls to the API or database
14. (5 Points) The app does not crash and sanitizes inputs
15. (4 Points) UI is ascetically pleasing and responsive across all devices. Selected text boxes should be visible while typing
16. (3 Points) Code is well organized in folders and easy to read