

ITIS/ITCS 5180 Mobile Application Development In Class Assignment 07

Basic Instructions:

1. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
2. **Export your Android project and create a zip file which includes all the project folder and any required libraries. The file name is very important and should follow the following format: Group#_IC07.zip. Submit the exported file using the provided canvas submission link.**
3. **The required Android Virtual Device (AVD) should have minimum SDK version set to 25 and target SDK at 28+.**
4. Failure to follow the above instructions will result in point deductions.

In Class Assignment 07 (100 Points)

MusixMatch Track Search

song

Limit: 25

Search

Sort by: ☒ Track rating
☐ Artist rating

Search results

(a) Main screen

MusixMatch Track Search

song

Limit: 25

Search

Sort by: ☒ Track rating
☐ Artist rating

Search results

A loading spinner is displayed in the search results area.

(b) Loading data

MusixMatch Track Search

song

Limit: 25

Search

Sort by: ☒ Track rating
☐ Artist rating

Search results

Track: TrackName	Artist: ArtistName
Album: AlbumName	Date: mm-dd-yyyy
Track: TrackName	Artist: ArtistName
Album: AlbumName	Date: mm-dd-yyyy
Track: TrackName	Artist: ArtistName
Album: AlbumName	Date: mm-dd-yyyy
Track: TrackName	Artist: ArtistName
Album: AlbumName	Date: mm-dd-yyyy
Track: TrackName	Artist: ArtistName
Album: AlbumName	Date: mm-dd-yyyy

(c) Display track list

https://www.musixmatch.com/lyrics/

musixmatch

Lyrics

ザ・レイジー・ソング

Bruno Mars

18 Translations available

Spanish Italian Choose Trans

Today I don't feel like doing anything
I just wanna lay in my bed
Don't feel like picking up my phone
So leave a message at the tone

(d) ActionView

Figure 1: app wireframes

In this assignment, you will be developing a music track search application using MusixMatch API. You can search any music related key word (track title, artist name, etc.) and display the search results. Your application should be able to load the details of the selected music track. You will be using JSON parsing to retrieve the music tracks, either dynamic layout or ListView to display the list, and use an intent to show the details of the selected music track using Action View (see Canvas description).

The assignment consists of only one activity, the Main Activity.

JSON API: We will be using Musixmatch API to retrieve the data. Please read the following API details:

API endpoint		http://api.musixmatch.com/ws/1.1/track.search
Parameters	q	Any word in the song title or artist name or lyrics.
	page_size	The limit / number of results defined by you.
	s_artist_rating	Sort by popularity index for artists (use desc).
	s_track_rating	Sort by popularity index for tracks (use desc).
	apikey	The authentication token from the account.
To be fetched	track_name	Name of the track.
	album_name	Name of the album.
	artist_name	Name of the artist.
	updated_time	The time of last update.
	track_share_url	URL to load using ActionView.

Instructions: Please follow the instructions to complete the assignment.

1. The app should contain an EditText as a search bar on top, see Figure 1(a).
2. There should be a SeekBar to set the limit right below the search bar. The minimum should be made 5 and the maximum should be made 25. When the user moves the SeekBar, the value should be changed accordingly. See Figure 1(a).
3. There should be a Button named “Search” right below the SeekBar, see Figure 1(a).
4. There should be a RadioGroup to select the sorting criteria containing two RadioButtons, such as “Track rating”, and “Artist rating”. “Track rating” should be the selected criteria by default, see Figure 1(a).
5. Putting Keyword/s in the search bar and clicking on Search button should start parsing. It should display a ProgressBar (Spinning) while parsing the data, see Figure 1(b).
6. On change of selecting any RadioButton in the RadioGroup should trigger a new API call where you need to change the parameter from s_artist_rating to

s_track_rating and vice versa. The value should be “desc” for both cases. Then reload the list again after you parse the new JSON.

7. There should be a list of results below the RadioGroup. It should be implemented using dynamic layout/ ListView/ RecyclerView.
8. Each item in the list should display the Track Title, Artist Name, Album Name, and Date (in MM-DD-YYYY format). The list must be Scrollable. See Figure 1(c).
9. Clicking on any item should take the user to load the **track_share_url** into the default browser using ActionView and implicit Intent, see Figure 1(d). An example snippet for loading the link would be:

```
String url = "http://www.example.com";  
Intent i = new Intent(Intent.ACTION_VIEW);  
i.setData(Uri.parse(url));  
startActivity(i);
```

Good Luck!!!

Tasks	Points
EditText, SeekBar, RadioGroup, and Button (design, and logic)	10
Creating and encoding API URL with the parameters properly	5
Changing the URL accordingly upon RadioButton change and calling the API again	7.5
AsyncTask to get the data from the API	15
Defining proper classes for keeping track objects and appropriate data structure to keep the data in runtime (ArrayList/ List)	10
Parsing JSON, create track objects, and save them in the data structure you defined	10
ProgressBar	7.5
Dynamic Layout / ListView / RecyclerView to display the result	25
Implicit Intent with ActionView to load the URL	10
Total	100