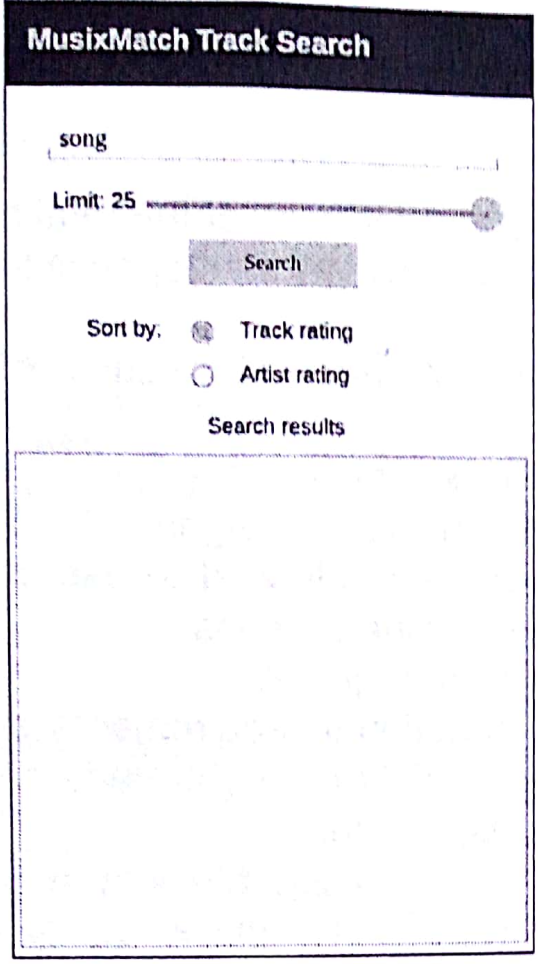
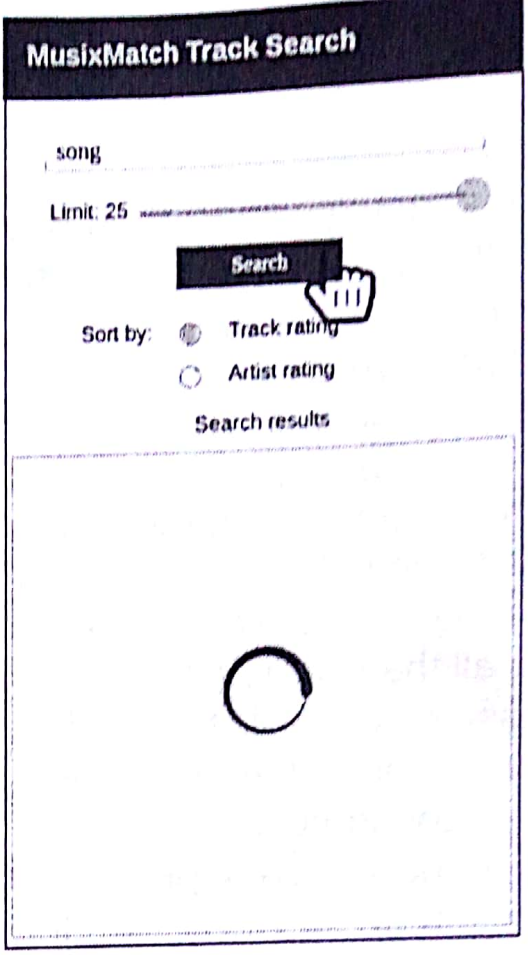
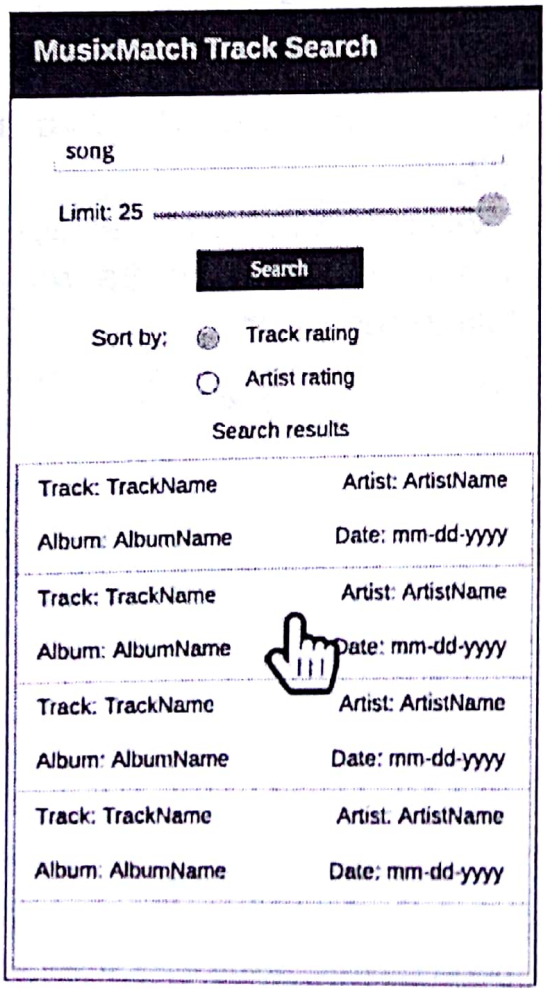


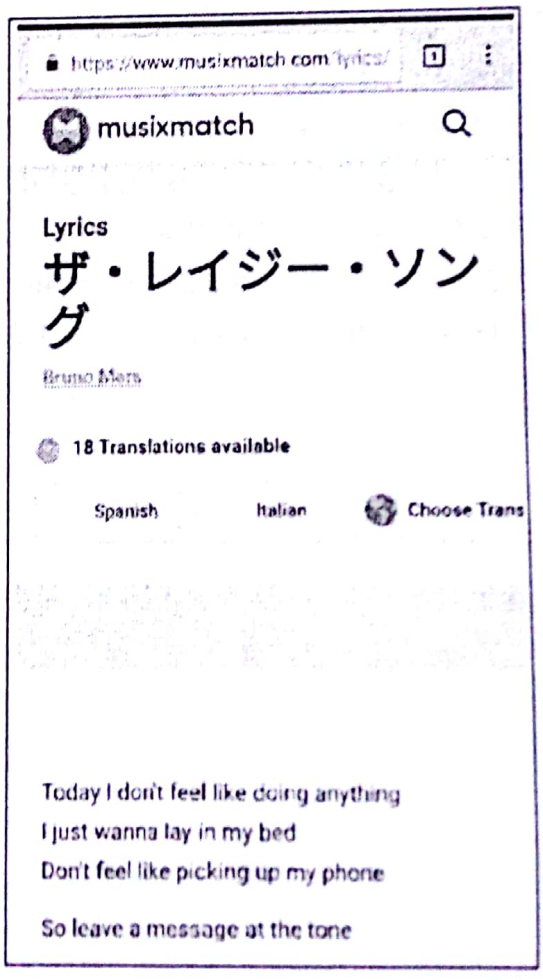
(a) Main screen



(b) Loading data



(c) Display track list



(d) ActionView

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!!!