3. Music player application project

3.1 Object: Song and AudioPlayer

The Song's object is responsible for containing the information of an mp3 file including: song titles, authors, links, loudness, image path, isLiked. The Class Song includes the constructor function and the Getter, Setter functions.

The AudioPlayer's object includes public static, data storage.

3.2 Permissions

[3] The purpose of a permission is to protect the privacy of an Android user. Android apps must request permission to access sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet). Depending on the feature, the system might grant the permission automatically or might prompt the user to approve the request.

To have permission to read and write data, the application needs to be granted these permissions. As of Android 6.0 operating system, besides having to declare in Manifest.xml, these rights must be approved by the users when running the application.

3.3 Data scanning

Once authorized, the application will search for all the files ending in .mp3 in the phone's memory. Song's objects will be made up of the found .mp3 files. At the end of the search, Song's objects will be stored as Arraylist of Song.

3.4 SQLite Database

[4] Android SQLite is a very lightweight database which comes with Android OS. Android SQLite combines a clean SQL interface with a very small memory footprint and decent speed. For Android, SQLite is “baked into” the Android runtime, so every Android application can create its own SQLite databases.

SQLite is a typical relational database, written by Richard Hipp in the form of a library in C programming language, containing tables (which consists of rows and columns), indexes etc. We can create our own tables to hold the data accordingly. This structure is referred to as a schema.

Once an Arraylist of type Song is available, an SQLite Database and a data table will be created. This data table will have the same data columns as Song-type properties.

```
CREATE TABLE MP_TABLE_NAME( MP_ID + integer primary key,
MP_PATH + text, MP_NAME + "text, MP_DURATION text, MP_SIZE text
MP_COMPOSER text, MP_IMG text, MP_ISLIKED text);
```

Data will be added to the rows corresponding to the column of each Song’s objects.

```
INSERT INTO MP_TABLE_NAME( MP_ID, MP_PATH, MP_NAME,
MP_DURATION, MP_SIZE, MP_COMPOSER, MP_IMG, MP_ISLIKED)
VALUES( i, path, name, duration, size, composer, img, isliked)
```

If users hit like or dislike a song, the data of that song in the database will also be updated.

```
UPDATE  MP_TABLE_NAME
SET MP_ISLIKED =  songisLiked
WHERE MP_NAME  = songgetName
```

If the user wants to delete a song, the delete command will be called and the corresponding data row will be deleted.

```
DELETE FROM MP_TABLE_NAME
WHERE id = songID
```

3.5 Custom Listview

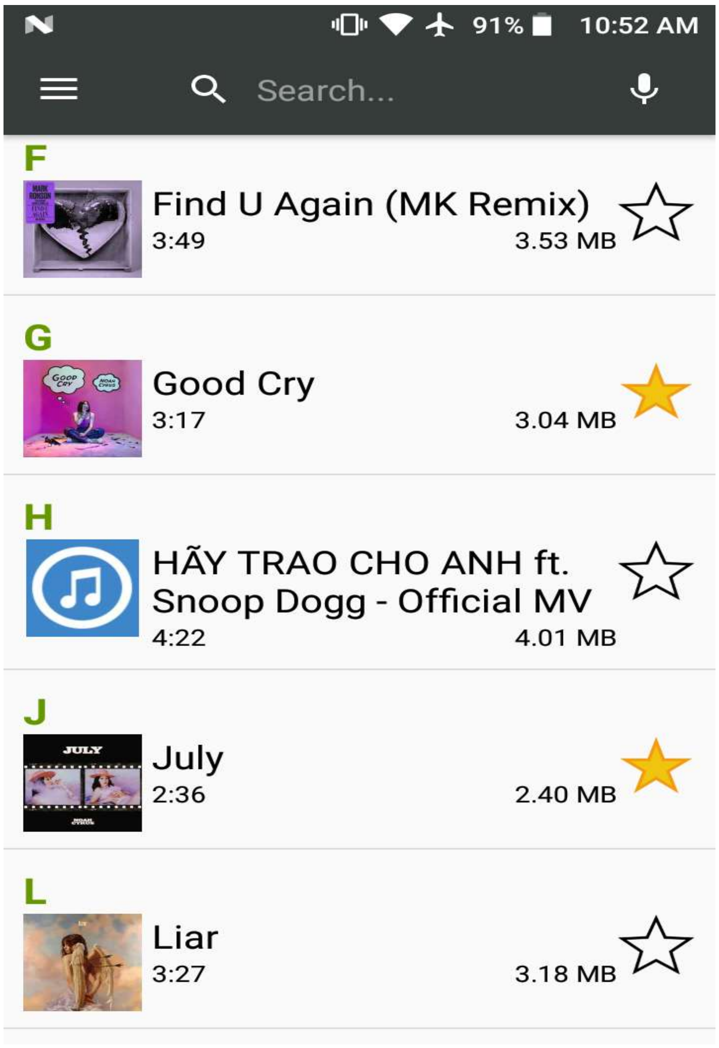
An Adapter class will be created to customize a layout file.

This layout file includes the TextView to display the information of the song, an ImageView to display the image of the song and ImageButton for selecting the song of the user.

All data in Database will be accessed. Use Cursor to read the data of each row of the returned database, forming a Song-type Arraylist. This array continues to pass to CustomAdapter objects.

```
SELECT * FROM MP_TABLE_NAME
```

This set of objects creates a complete ListView, displaying all the songs in alphabetical order in the phone.
The corresponding song will be played when the user clicks on it.
When the user make long clicks on a song, there are two options that appear: delete the song or view more details about this song.



Find U Again (MK Remix)
3:49 3.53 MB

Good Cry
3:17 3.04 MB

HÃY TRAO CHO ANH ft. Snoop Dogg - Official MV
4:22 4.01 MB

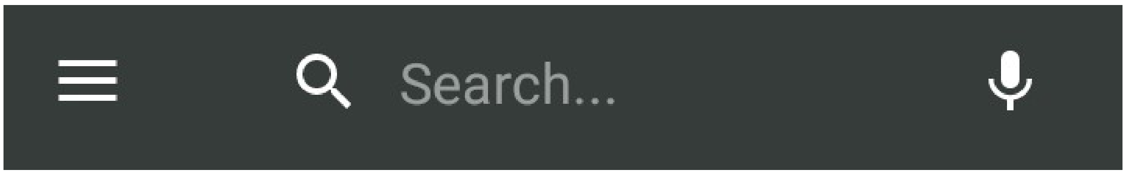
July
2:36 2.40 MB

Liar
3:27 3.18 MB

3.6 Search for songs in playlist

First, a database to store suggestions will be created. This database only stores the id and name of all songs in the list.

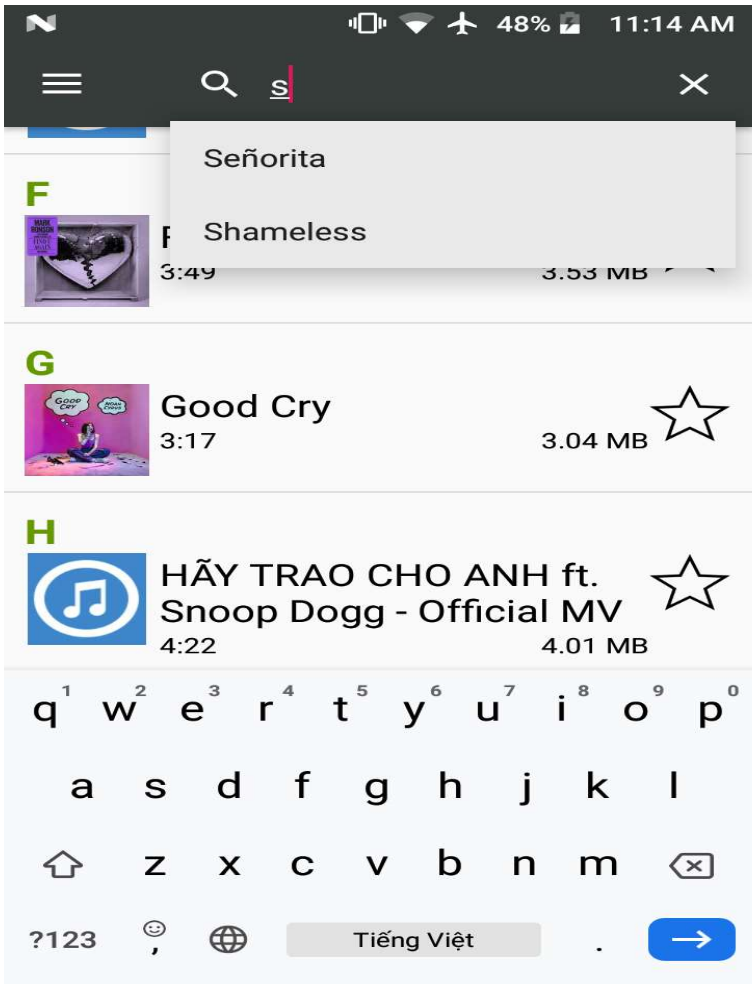
```
CREATE TABLE TABLE_SUGGESTION( FIELD_ID integer primary key
autoincrement, FIELD_SUGGESTION text, UNIQUE (FIELD_SUGGESTION)
ON CONFLICT REPLACE)
```



3.6.1 In text

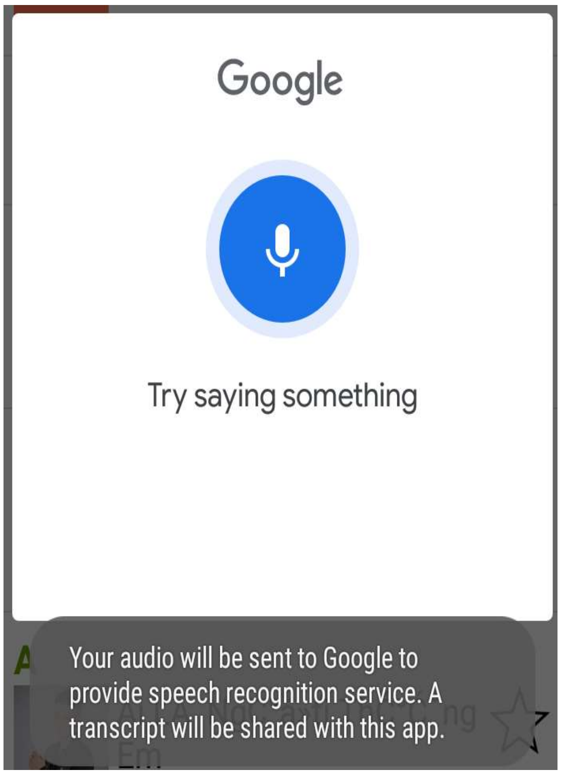
When the user enters the name of the song in the search box, an SQLite demand statement will be retrieved in the Suggestion Database to find the song with the corresponding name.

```
SELECT * FROM TABLE_SUGGESTION WHERE FIELD_SUGGESTION LIKE text
```



If the user clicks on a suggestion, the application will play that track.

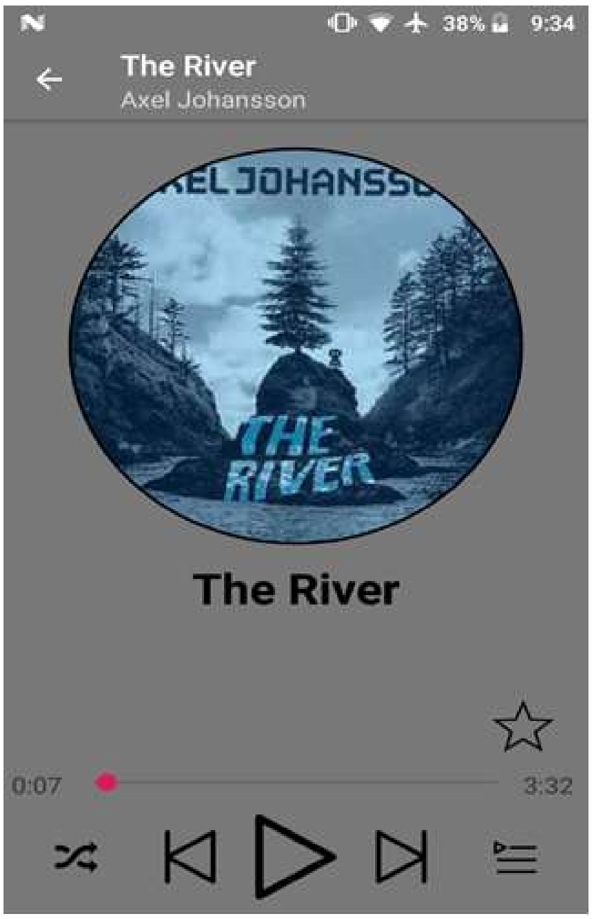
3.6.2 In voice

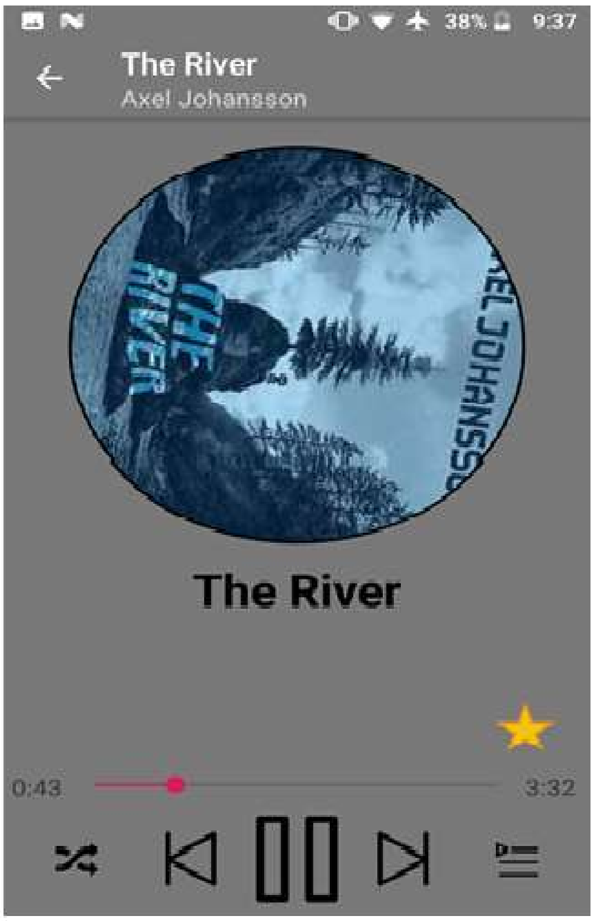


Your audio will be sent to Google to provide speech recognition service. A transcript will be shared with this app.

To use this feature, the application needs an Internet connection and is authorized to record audio. When users read the name of the song they want to find, they create an audio file. This file will be sent to Google and returned as a text. The text will be input into the search box and continue to execute the same as the search by typing text.

3.7 Screen to play music





3.7.1 Playing music

Use the android.media.MediaPlayer library to play music:

```
MediaPlayer player = new MediaPlayer();
player.setDataSource(path);
player.prepare();
player.start();
```

Because this is an Android Service, the music player will continue to play even when switching screens or exiting apps.

3.7.2 Seekbar

Seekbar shows the running of the song. Use the 2 functions below to find the duration of the song playing. Thereby we find the percentage of completion of the song running.

```java
long totalDuration = player.getDuration();
long currentDuration = player.getCurrentPosition();
int progress = (int) (getProgressPercentage(currentDuration,
totalDuration));
```

3.7.3 Circle Image View

This is a library introduced on Github by Henning Dodenhof.
This library will create a CircleImageView object that displays images in a circular frame. Can customize the values like a normal ImageView.

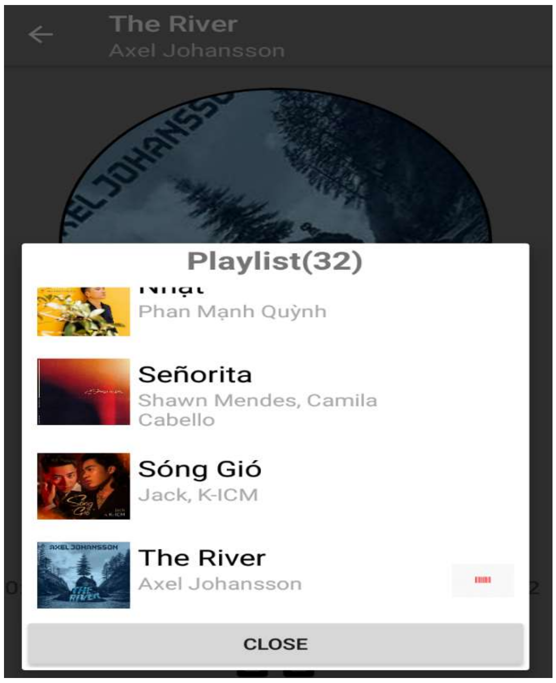
3.7.4 Animation

Using library android.animation.ObjectAnimator;
Combine using Circle Image View and Animation, the song image in a circular frame will rotate while the song is being played.

```java
Objectanimator anim = ObjectAnimator.ofFloat(imageView,
"rotation", 0, 1440);
anim.setDuration(60000);
anim.start();
```

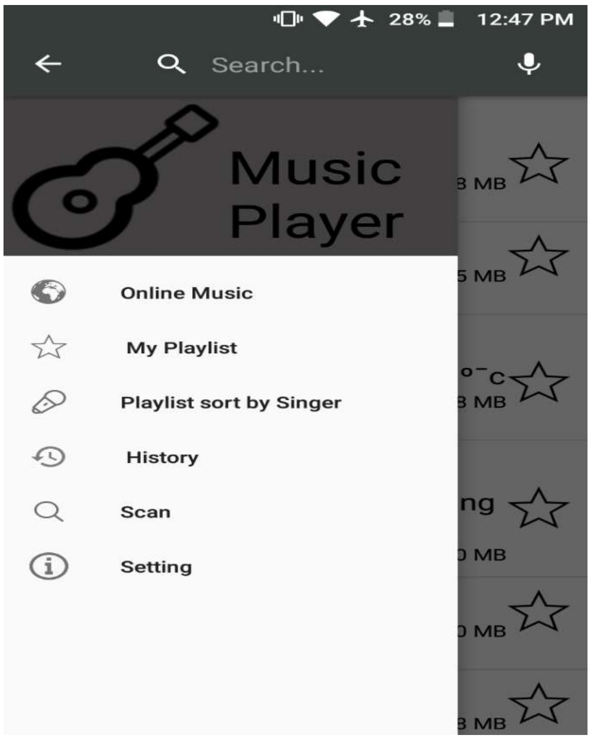
3.7.5 DialogFragment

Use DialogFragment to display the playlist on the screen.
The red animation on the right shows which song is being played. This animation uses the library of Karol Wrótniak to display.



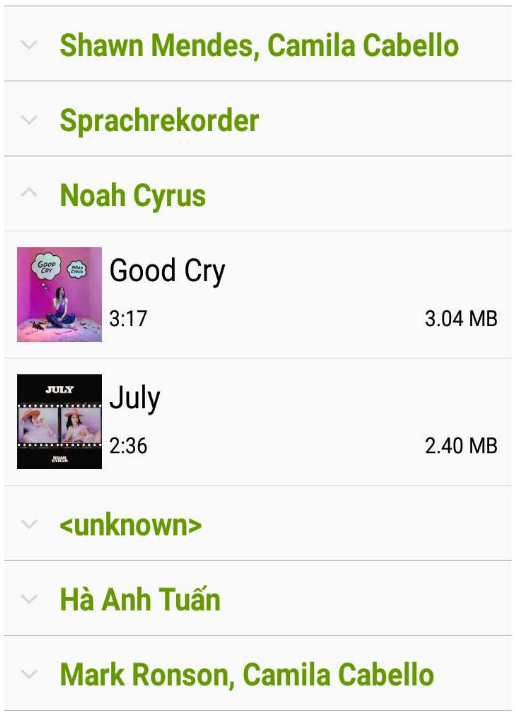
3.8 Navigation Drawer

The Navigation Drawer is a pull menu that displays as a stack at the edge of the screen. It is hidden when not in use, but appears when users swipe their finger from the left edge of the screen or the user touches the toolbar icon.



3.9 Expandable ListView

Use Expandable ListView to display the list of songs by artist name.
Only the singer name will be shown. When the user clicks on a singer's name, songs will appear.



3.10 Deezer Android SDK

3.10.1 About Deezer

[5] Deezer is an online music streaming service. It allows users to listen to music from firms such as Sony, Universal Music, and Warner Music Group on multiple devices, whether online or offline. Deezer was founded in Paris, France and currently has 53 million licensed songs in the music store, along with more than 30,000 radio channels, 14 million monthly users, and 6 million paid subscribers as of April 3. 2018.

3.10.2 About Deezer Android SDK

[6] Deezer Android SDK allows embedding many features from Deezer directly into the application. It mainly includes the following functions:

• Communicating with Deezer API, accessing to entire playlists as well as user information.
• Authenticating users with their credentials.
• Easily playing music from Deezer playlists

3.10.3 Authorization

Applications need to be granted the following permissions for the Deezer SDK to function correctly:

android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
android.permission.WAKE_LOCK

3.10.4 Initializing SDK

Developer needs to register a developer account with Deezer and apply via this account. Deezer will then provide an applicationID to make the connection.

```
String applicationID = "123456";
DeezerConnect deezerConnect = DeezerConnect.
forApp(applicationID).withContext(this).build();
```

3.10.5 Authenticating users

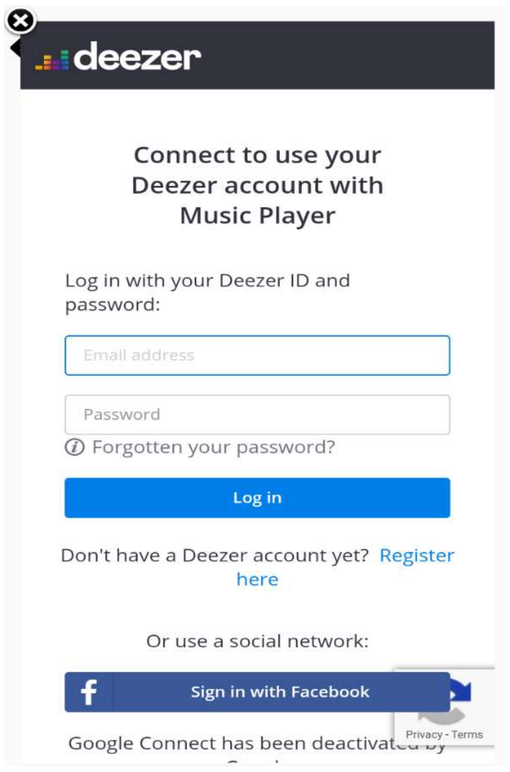
Some permissions need to be granted to the application to be able to listen for authentication events.

```
String[] permissions = new String[] {
Permissions.BASIC_ACCESS,
Permissions.MANAGE_LIBRARY,
Permissions.LISTENING_HISTORY };
```

The user authentication process will then be executed.

```
deezerConnect.authorize(activity, permissions, listener);
```

Users need to use Deezer account or Facebook account to be able to login.



Connect to use your Deezer account with Music Player

Log in with your Deezer ID and password:

Email address

Password

Forgotten your password?

Log in

Don't have a Deezer account yet? Register here

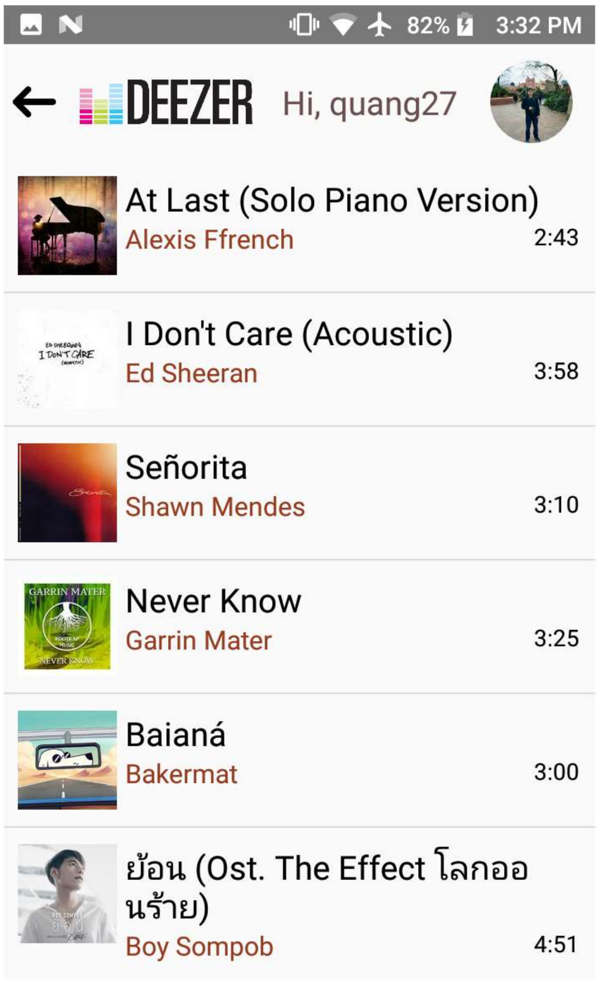
Or use a social network:

Sign in with Facebook

Google Connect has been deactivated by

The User Session will also be saved, avoiding the need to log back in.

```
SessionStore sessionStore = new SessionStore();
sessionStore.save(deezerConnect, context);
```



At Last (Solo Piano Version)
Alexis Ffrench 2:43

I Don't Care (Acoustic)
Ed Sheeran 3:58

Señorita
Shawn Mendes 3:10

Never Know
Garrin Mater 3:25

Baianá
Bakermat 3:00

ย้อน (Ost. The Effect โลกออนร้าย)
Boy Sompob 4:51

3.10.6 Deezer Track

The Track object has information about the song, including the song titles, artist names, duration, album names and other attributes.

Use the following function to retrieve the top 10 favorite songs, the return it returns a list of Track type.

```
DeezerRequest request = DeezerRequestFactory.requestChartTracks();
```

Continuing transferring data from the list of Track types on to CustomListView to display the playlist.

Using a variable of TrackPlayer type will play the song.

```
TrackPlayer mTrackPlayer = new TrackPlayer(app, deezerConnect,
new WifiAndMobileNetworkStateChecker());
mTrackPlayer.playTrack(track.getId());
```

3.10.7 Deezer User

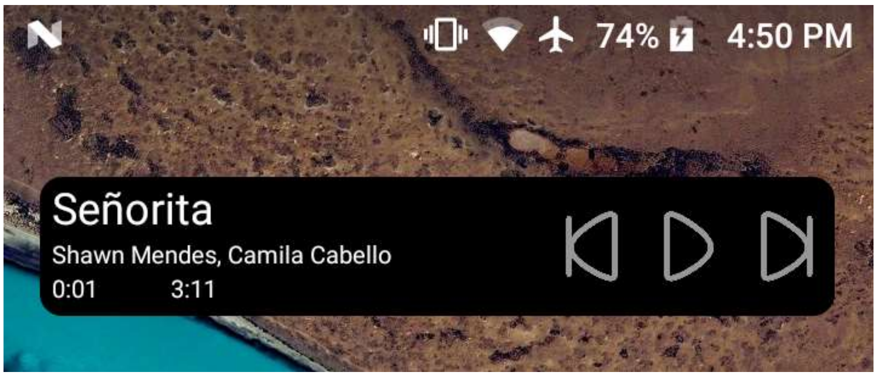
The Users have personal information about the user, including names, date of birth, email, status, gender, avatar and other information.

Use the following 2 functions to get the user's name and avatar:

```
deezerConnect.getCurrentUser().getName()
deezerConnect.getCurrentUser().getMediumImageUrl()
```

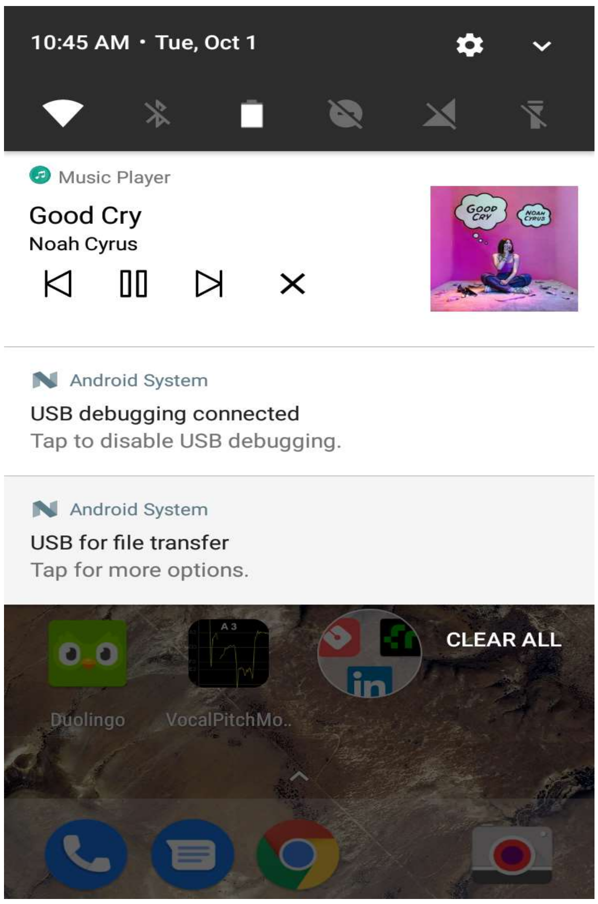
3.11 Widget

A widget is a simple application extension that is usually part of a larger application that is installed on the device. Widgets come in all shapes and sizes, can be customized, and on any of the available Home screens for quick access.



Because this application uses the MediaPlayer library, as a Service, to play music, the Widget application can easily listen for events from the main application. Thereby, the Widget application can update the information about the song, the progress of music playback, as well as play the music itself, move the next song.

3.12 Notification


10:45 AM • Tue, Oct 1

Music Player
Good Cry
Noah Cyrus

Android System
USB debugging connected
Tap to disable USB debugging.

Android System
USB for file transfer
Tap for more options.

[7] A notification is a message that Android displays outside the app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

Notification will pop out when the user plays a song. Similar to Widgets, it displays the song titles, artist names and also has the ability to stop, move to next songs.