

REPORT

MOBILE APPLICATION DEVELOPMENT



TOPIC

Online Music Player

Instructor: Dr. TRAN Giang Son

Group Another

Phạm Gia Phúc	BI10-138
Trần Đức Minh	BI10-113
Đặng Gia Linh	BI10-101
Nguyễn Tấn Dũng	BI10-041
Phạm Đức Thắng	BI10-159

NOVEMBER, 2021

ANOTHER

Contents

I.	Introduction	3
II.	Methods	4
1.	Set - up	4
2.	Library	4
3.	Architecture	5
4.	Component	5
5.	UI Design - Layout	7
a.	Main layout + Online layout	7
b.	Music Player layout (for both Online and Offline)	9
6.	Online Music Library	10
7.	Database - Firebase	10
III.	Result	12
1.	Demonstration	12
a.	Main layout + Online layout	12
b.	Music Player layout (Online and Offline)	13
2.	Functionalities	13
3.	User Interfaces	14
4.	Performance	14
IV.	Conclusion	15
1.	About the application	15
2.	Future improvements direction	15
V.	References	16

I. Introduction

Online music players are the **platforms which enable users to play music on the internet** without downloading any player or song. These are music streaming platforms from where people can stream and listen to any song, any time with internet access available.

Popular online music players: Spotify, YouTube, Amazon Prime Music, Soundcloud...

Why online music player (streaming service)?

- Save up local storage
- Access to large online music libraries
- Better sound quality
- More customizations available
- Music charts
- Interaction with other users
- See what's trending

This project:

- Build an online music player application for android devices (Android 5.0 or higher).
- Aim for a lightweight and well-performing application with basic functionalities of an online music player.

II. Methods

- SET UP
- LIBRARY
- ARCHITECTURE
- COMPONENT
- UI DESIGN - LAYOUT
- ONLINE MUSIC LIBRARY
- DATABASE - FIREBASE

1. Set - up

Workspace setups:

- Java JDK 15:

Java JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

- Android Studio (for implementation and testing):

"Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development."

- Wikipedia

- Android 5.0 Lollipop with Google APIs:

When choosing target SDK version: API 21 (default setting) - compatible with approximately 94% of android devices.

Permissions (Android Manifest):

- Access to external storage (for reading audio files)
- Access to internet (for getting contents from internet)
- Access to internet state (for connecting to online database)

2. Library

```
implementation 'com.karumi:dexter:6.2.2'
implementation 'androidx.media2:media2-player:1.2.0'
implementation 'com.google.firebase:firebase-database:20.0.2'
```

Default library: androidx (for UI Design, Layout and Classes, Functions)

Google.android.material (v1.4.0): for UI Design and Layout

Dexter: to simplify the process of the application requesting permission at runtime

Firebase: to connect to Google Firebase (online database)

3. Architecture

One package: OMP (online music player)

Gradle: build system - **used to automate building, testing, deployment**

5 Activities (Activity class)

5 Layouts - one Layout for each Activity

2 modes: Online and Offline, YouTube Player mode (on developing)

4. Component

The core, the heart of the application: **MediaPlayer**

```
static MediaPlayer mediaPlayer;
```

MediaPlayer is a component of Android Studio that is able to play audio and video from local storage or URL.

MediaPlayer provides methods to control playback of audio/video including play, stop, fast forward, rewind...

MediaPlayer is a component without a interface, it can be combined with a View.

Application Functionalities:

Main activity: read and display audio files list from local storage, controller (when a song is playing)

```
final ArrayList<File> songsInLib = findSong(Environment.getExternalStorageDirectory());
items = new String[songsInLib.size()];
```

Offline player activity: basic functions of a music player: play/pause, forward/rewind, skip/back, shuffle/repeat toggle, seek bar - to change current position of the song

```

mediaPlayer.setOnPreparedListener(mp -> {...
    mediaPlayer.start();
... });

buttonPlay.setOnClickListener(new View.OnClickListener() {...});
buttonNext.setOnClickListener(new View.OnClickListener() {...});
buttonPrev.setOnClickListener(new View.OnClickListener() {...});
buttonForward.setOnClickListener(new View.OnClickListener() {...});
buttonRewind.setOnClickListener(new View.OnClickListener() {...});
btnShuf.setOnClickListener(new View.OnClickListener() {...});
btnRep.setOnClickListener(new View.OnClickListener() {...});
btnList.setOnClickListener(new View.OnClickListener() {...});

```

Online activity: read and display URLs list from database (Firebase), controller

```

value = snapshot.getValue(String.class);
songNameData.add(value);
arrayAdapter.notifyDataSetChanged();

```

Online player activity: the same as Offline player

```

mediaPlayerOnl.setDataSource(songNameOnl);
textSongOnl.setText(songNameOnl);
mediaPlayerOnl.setAudioStreamType(AudioManager.STREAM_MUSIC);
mediaPlayerOnl.prepare();

mediaPlayerOnl.start();

```

YouTube Player: connected to 'Youtube.com' via API - on developing, not functioning

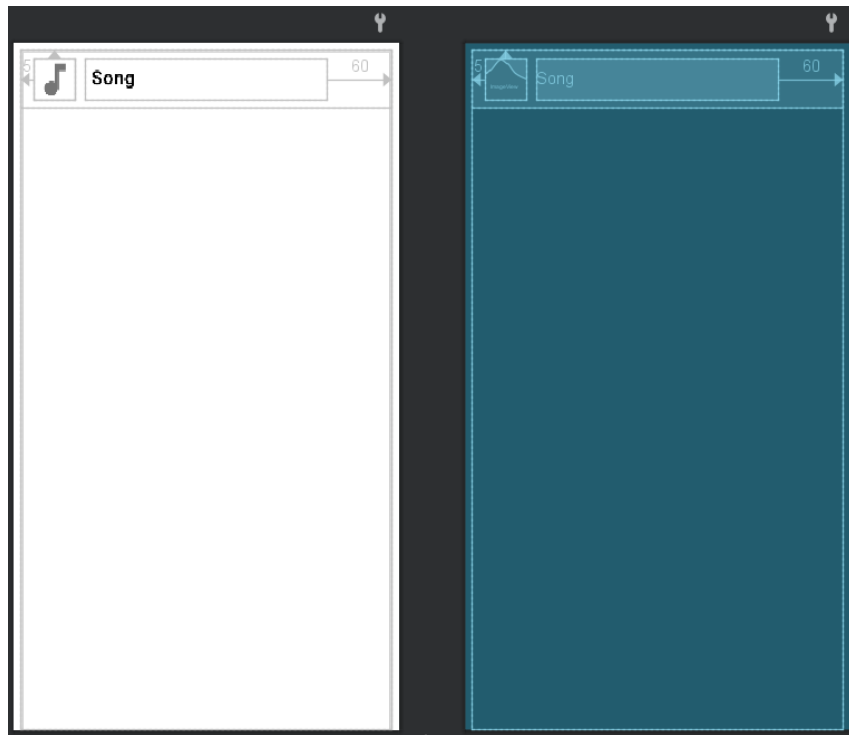
```

public class YoutubeAPI extends YouTubeBaseActivity {
    ...
    YouTubePlayer.OnInitializedListener onInitializedListener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {...
        onInitializedListener = new YouTubePlayer.OnInitializedListener() {...};
    }
}

```

5. UI Design - Layout

a. Main layout + Online layout

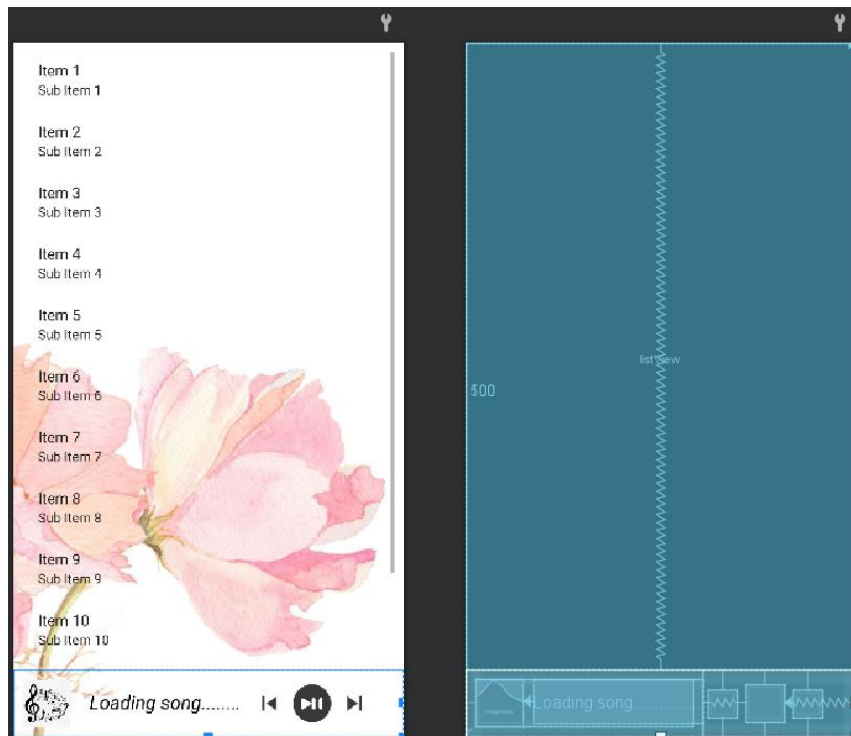


Each item (audio file/URL) will be displayed like above:

- An icon on the left
- The song name/URL beside

Each item is displayed by using `CardView` - an extended class of `FrameLayout` that lets developer display Views on top of each other - on top of a `ListView` in this case.

```
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:elevation="8dp"
app:cardBackgroundColor="#00FFFFFF">...</androidx.cardview.widget.CardView>
```

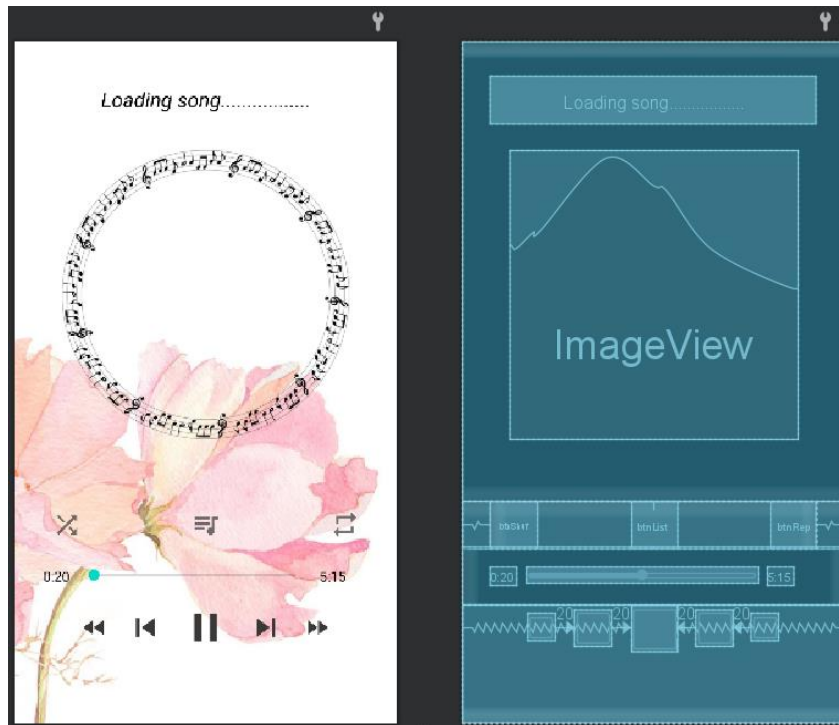


The controller (appear when a song is playing) is on the footer of the layout:

- Icon, song name
- Back - Play/Pause - Skip buttons

```
<LinearLayout
    android:id="@+id/musicController"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:visibility="gone">...</LinearLayout
```


b. Music Player layout (for both Online and Offline)



From top to bottom, left to right:

- TextView: display song name
- ImageView: the music circle - for it to look less empty
- Buttons: Shuffle - Back-to-list - Repeat (for one song)
- TextView: current position of the song - Updated every 1 second
- Seekbar: show and change the current position of the song
- TextView: duration of the song - Updated when changing song
- Controller: Rewind - Back - Play/Pause - Skip – Forward

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PlayActivity"
    android:orientation="vertical"
    android:background="@drawable/b2"
    android:weightSum="10">...</LinearLayout>
```

6. Online Music Library

Music items are stored in the cloud storage of web server.

Browse: <http://darkminh7.infinityfreeapp.com/Musics/> for list of songs in the online library.

The music library has 12 songs (for demonstration).

Directory Listing

Name	Last modified	Size	Description
▲ Parent Directory		-	[unknown item..]
🎵 Lewis Capaldi - Before You Go (Lyrics).mp3	2021-11-09 05:41	3.8M	MPEG Layer 3 Format
🎵 Kage - Feel Me [Monstercat Release].mp3	2021-11-09 05:37	3.5M	MPEG Layer 3 Format
🎵 Jason Derulo - Swalla (feat. Nicki Minaj & Ty Dolla \$ign) [Official Music Video].mp3	2021-11-09 05:33	3.6M	MPEG Layer 3 Format
🎵 It's different - Outlaw (feat. Miss Mary) [NCS Release].mp3	2021-11-09 05:30	3.9M	MPEG Layer 3 Format
🎵 Hyper - Spoiler.mp3	2021-11-09 05:24	4.2M	MPEG Layer 3 Format
🎵 GD X TAEYANG - GOOD BOY M_V.mp3	2021-11-09 05:18	3.9M	MPEG Layer 3 Format
🎵 Fifth Harmony - Worth It (Official Video) ft. Kid Ink.mp3	2021-11-09 05:12	3.5M	MPEG Layer 3 Format
🎵 Era Istrefi - Bonbon.mp3	2021-11-09 04:40	2.8M	MPEG Layer 3 Format
🎵 Designer - Panda (Official Audio).mp3	2021-11-09 05:07	3.8M	MPEG Layer 3 Format
🎵 Clarx - H.A.Y [NCS Release].mp3	2021-11-09 05:04	3.5M	MPEG Layer 3 Format
🎵 BTS - MIC Drop (feat. Designer) (Steve Aoki Remix).mp3	2021-11-09 04:56	3.7M	MPEG Layer 3 Format
🎵 Ash - Mosaique.mp3	2021-11-09 04:53	4.2M	MPEG Layer 3 Format

7. Database - Firebase

Song URLs are stored in the online database - Google Firebase.

“Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps”

- *WhatIs*

→ The application can get the audio files from the internet (music library) via URLs (only supports .mp3, .wav, .wma files).

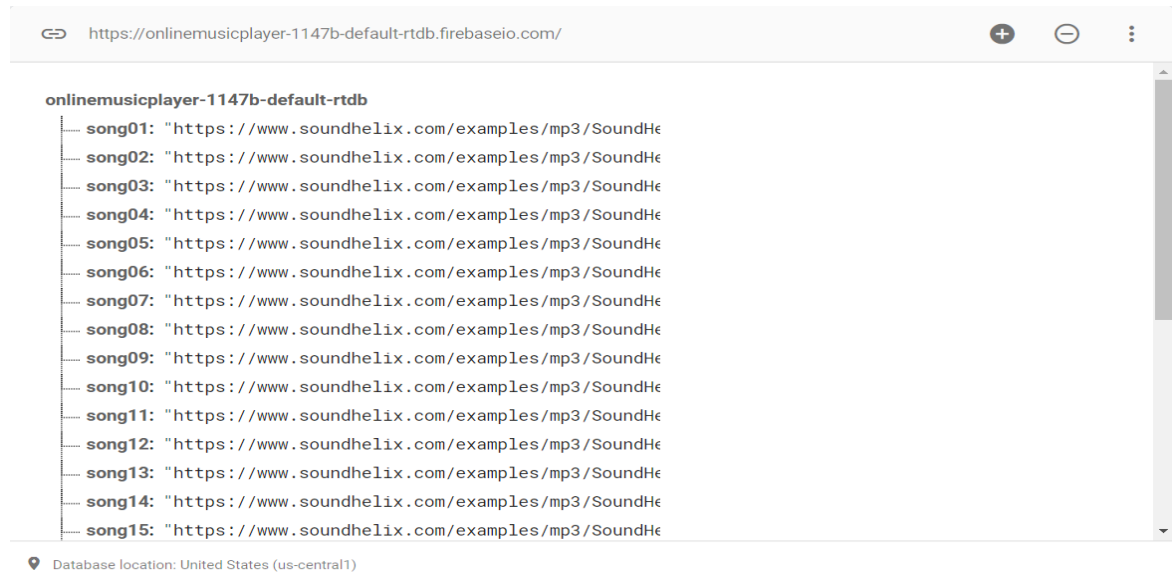
The database has additional 25 song URLs from other websites (apart from 12 songs mentioned above. They had been used for testing before the web server was active (launched)).

```
databaseReference.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String
previousChildName) {
        // run once the data is added to database or when run the app

        // get data, present in literal
        value = snapshot.getValue(String.class);

        // attempt value to list of users
```

```
        songNameData.add(value);  
        arrayAdapter.notifyDataSetChanged();  
    }  
... }));
```

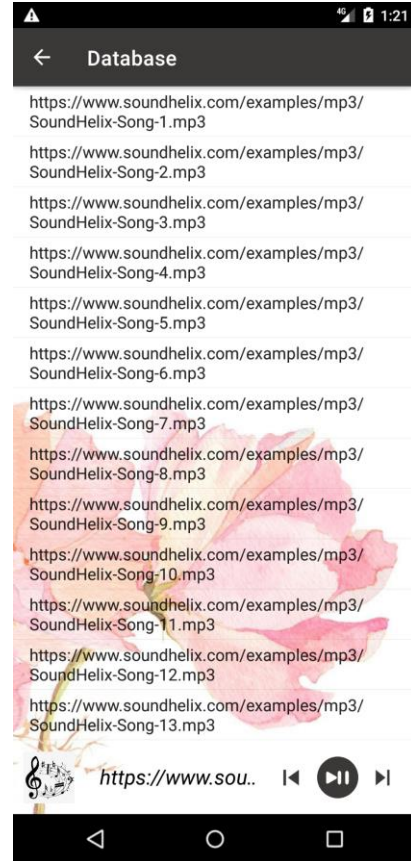


III. Result

1. Demonstration

How the application actually looks like (with action bar - menu and after reading audio files/URLs from local storage and online database):

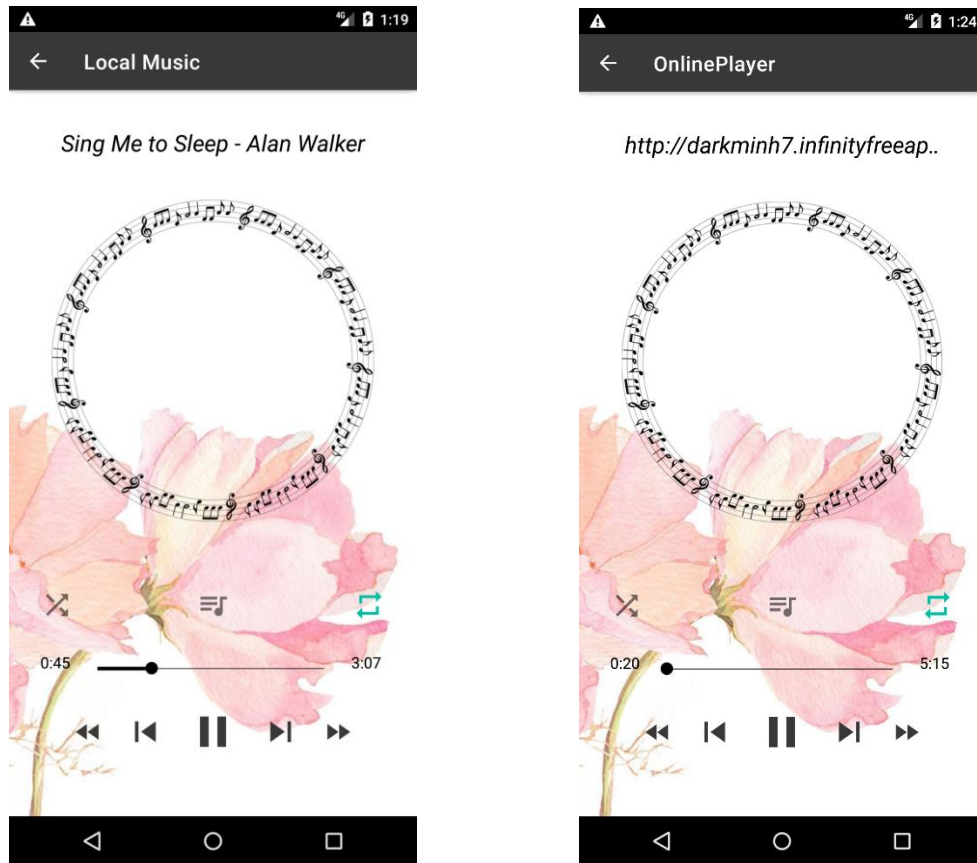
a. Main layout + Online layout



User can change to other activity layout using the menu drop down on the top right of the main layout.

b. Music Player layout (Online and Offline)

They look the same as in the [UI Design - Layout](#) Section:



The Online player layout only differ from the Offline at the title on the action bar.

Notice when the Repeat button is clicked, it turns blue, which means that the function is activated, this goes for the Shuffle button as well.

2. Functionalities

Most of the functionalities work fine, except for some functionalities in the online player activity (skip/back buttons).

This is not because there are errors in the code. It's because the list of items hasn't been indexed and the functions haven't been assigned to these buttons.

→ Press skip/back buttons: nothing happens

The controller in the main activity does not update song's name when changing song from the player activity

→ This is a lack of functions in the code to synchronize between two activities, not an error.

3. User Interfaces

Pretty simple UI, still there are many things that need to improve:

- The floating song's name: when the name is too long, it starts to float, which is annoying to us developers and also to user.
- The flower image in the background: it is maybe subjective but some users will consider this "unprofessional", and it's true. The application needs to have additional functions for customization, or at least "theme changing" setting.
- The interface of online activity layout needs to be improved, it is hard to distinguish between items, especially when the application displays the whole URL on the screen.

→ One more time, this is a lack of function to display the song URL instead of the names.

4. Performance

Ignoring the lack of functionality, the application works beautifully. It is fast, consistent as well as no crash has ever happened.

The online player may take a bit of time before the song is being played, which depends on the internet connection.

IV. Conclusion

1. About the application

The application is really easy to use, anyone who knows how to tap a screen will know how to use it. It has a pretty simple UI and it does not consume much memory in the working time, therefore, it is perfect for users who just want to listen to music while doing something else. Just let the application run in the background without worrying it will crash. Moreover, it also means that the application will save the user a lot of battery power compared to the others.

“Not a lie, the app may look like a joke when compared to YouTube or Spotify but I really like it, except for the pink flower background. It does exactly what a music player needs to do, to play music and just that. Simple and nice.

Of course, I know what I’m saying, I have a music player on my Xiaomi Redmi, the default one. I have about 1000 songs in the storage and every time I want to play a song, I have to wait 5’ for it to read through all the songs and artist thumbnails. Sometimes it even crashes. To make it worse, I also have to wait for another 5’ when changing from a playlist screen to the main screen. So, this app is a life saver :D”

- Minh, User

It may not be a great application in terms of a group project in the course, but it is a great one in terms of application.

2. Future improvements direction

An online music player can be as simple as just playing song from a URL or as complicated as YouTube or Spotify

Of course, we – the development team – are not that delusional to think we can compete with these big companies which have millions of users and subscribers. Thus, the future improvements of the application will be the attention to the functionalities for personal users, which may include:

- An application logo (of course)
- Theme customization
- Search for song/album/artist
- Recent songs
- Favorited songs (like starred in Gmail)
- Create playlists
- Sort songs by (name, number of played times, added date)
- A scroll with alphabetical order

- Tap to add artist image or lyrics
- Display artist/album under song name
- Display artist image (thumbnail) - not sure about this, may take a lot of storage
- YouTube Player - on developing
- **Advanced features:**
 - + Sound visualizer
 - + Sound mixing, editing, enhancement
 - + Running lyrics
 - + YouTube videos to MP3 converter - maybe connect to a website

V. References

<https://github.com/Lib3Rt9/androiddev2022/tree/master/Final>