

**In-House Summer Training Report**  
**On**  
**ANDROID MUSIC PLAYER (GLAZBA)**

Submitted in partial fulfillment of the requirement for the award of degree of

**Bachelor Of Technology**

**In**

**Computer                      Science                      Engineering**

Mentors:

Submitted By:

Ms ANJALI KAPOOR

PRAGYA YADAV 40810402715

Ms. AAKANKSHI GUPTA

PIYUSH RAI 40710402715

AKASH RAI 00310407216

YASH MENDIRATTA 40110402715



Department of Computer Science Engineering

Amity School of Engineering and Technology

Guru Gobind Singh Indraprastha University, New Delhi

## **CERTIFICATE**

This is to certify that the students listed below pursuing B.Tech (Computer Science and Engineering) have successfully completed their In-House Summer Training Project entitled “ANDROID MUSIC PLAYER (GLAZBA)”.

**Yash Mendiratta** [40110402715]

**Pragya Yadav** [40810402715]

**Akash Rai** [003110402715]

**Piyush Rai** [40710402715]

This training partially fulfils their Bachelor of Technology course requirements after the second year. The project is completed at Amity School Of Engineering and Technology, an institution affiliated to Guru Gobind Singh Indraprastha University.

**Ms. Anjali Kapoor**

**Asst. Professor**

**Dept. of CSE**

**Ms. Aakankshi Gupta**

**Asst. Professor**

**Dept. of CSE`**

## **ACKNOWLEDGEMENT**

We are thankful to **Late Prof B.P.Singh, Senior Director and Prof. (Dr.) Rekha Aggarwal**, for their moral support and providing us good infrastructure facilities of the institutes which helped us in implementing the project. It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our respected **Asst. Prof. Ms Aakankshi Gupta, Department of CSE and IT** and esteemed guide **Asst. Prof. Ms Anjali Kapoor, Department of CSE and IT** for their valuable guidance and helping us in completing our project.

We are thankful to all the staff member of the Departments of CSE and IT for their help in our work.

At the end, we would like to express our sincere thanks to all others who helped us directly or indirectly during this project work.

**Yash Mendiratta**

**(401101402715)**

**Pragya Yadav**

**(40810402715)**

**Akash Rai**

**(00310402715)**

**Piyush Rai**

**(40710402715)**

## TABLE OF CONTENTS

Certificate.....	i
Acknowledgment.....	ii
Abstract.....	iii
Introduction.....	7
1. Requirements.....	8
2. Design & Layout.....	9
3. Screens.....	18
3.1 Slider Screen.....	18
3.2 Preference Screen.....	20
3.3 Settings Screen.....	21
4. Features.....	22
4.1 Basic Features.....	22
4.2 Additional Features.....	24
5. Conclusion.....	28
6. References.....	29
7. Appendix.....	30

## LIST OF FIGURES

FIGURES	DESCRIPTION
Figure 1	Audio Picker
Figure 2	Full Playback
Figure 3	Library Contents
Figure 4	Media Library Preferences
Figure 5	Permission Requests
Figure 6	Playlist Buttons
Figure 7	Restore Default Settings
Figure 8	Slider Screen 1
Figure 9	Slider Screen 2
Figure 10	Preference Screen
Figure 11	Settings
Figure 12	List View by Genres
Figure 13	Accelerometer Shake
Figure 14	Miscellaneous Features
Figure 15	Theme Selection

## **ABSTRACT**

Music player forms an integral part of today's smartphone. It is generally used by users to listen music files. Many users like to listen music by a smart phone, but the music player has many limitations. With a rapid development of communication and network, multimedia based technology is adopted in this music player. The papers discuss about the study of music player with the help of existing music player which are available in the android market and proposed system for the music player which will provide the uninterrupted enjoyment for the user. With a rapid development of communication and network, multimedia based technology is adopted in this music player. The papers discuss about the study of music player with the help of existing music player which are available in the android market and proposed system for the music player which will provide the uninterrupted enjoyment for the user.

## INTRODUCTION

Glazba is a Croatian word which stands for Music. Glazba music is a simple but an elegant music player that puts very little between you and your music. It allows one to browse music by genres, artists, albums, folders and playlists. On the top of it, it has some additional features which makes it stand out of the crowd. With a rapid development of communication and network, multimedia based technology is adopted in this music player. The papers discuss about the study of music player with the help of existing music player which are available in the android market and proposed system for the music player which will provide the uninterrupted enjoyment for the user.

With a rapid development of communication and network, multimedia based technology is adopted in this music player. It is an interactive music player which features a variety of audio enhancements that tweak your music experience a little more than the usual.

It has an ability to sync your music library in the phone with this app. It tries to do something a little different by providing features like idle timeout, themes, accelerometer shake etc and succeeds. The user interface matches every single detail of the material design guidelines to ensure Phonograph is an eye candy for you. The experience is even better on devices running Android Lollipop and above.

The Android multimedia framework includes support for playing variety of common media types, so that you can easily integrate audio, video and images into your applications. You can play audio or video from media files stored in your application's resources (raw resources), from standalone files in the filesystem, or from a data stream arriving over a network connection, all using [MediaPlayer](#) APIs.

Android is providing MediaPlayer class to access built-in mediaplayer services like playing audio, video e.t.c. In order to use MediaPlayer, we have to call a static Method **create()** of this class. This method returns an instance of MediaPlayer class.

# **CHAPTER 1**

## **REQUIREMENTS**

### **1. SOFTWARE REQUIREMENT**

- JAVA SDK
- Android Studio v2.3
- Microsoft Visual Studio
- DirectX
- Graphics driver

### **2. HARDWARE REQUIREMENTS**

- Intel i3 processor or above
- 2 GB RAM or above
- 128 MB or above dedicated graphics
- Sound card
- 4 GB or more Free space



## CHAPTER 2

### DESIGN & LAYOUT

Android uses a new design metaphor inspired by paper and ink that provides a reassuring sense of tactility. We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design. This spec is a living document that will be updated as we continue to develop the tenets and specifics of material design.

All action takes place in a single environment. Objects are presented to the user without breaking the continuity of experience even as they transform and reorganize.

Motion is meaningful and appropriate, serving to focus attention and maintain continuity. Feedback is subtle yet clear. Transitions are efficient yet coherent.

#### Material Design

The user interface matches every single detail of the material design guidelines to ensure Phonograph is an eye candy for you. The experience is even better on devices running Android Lollipop

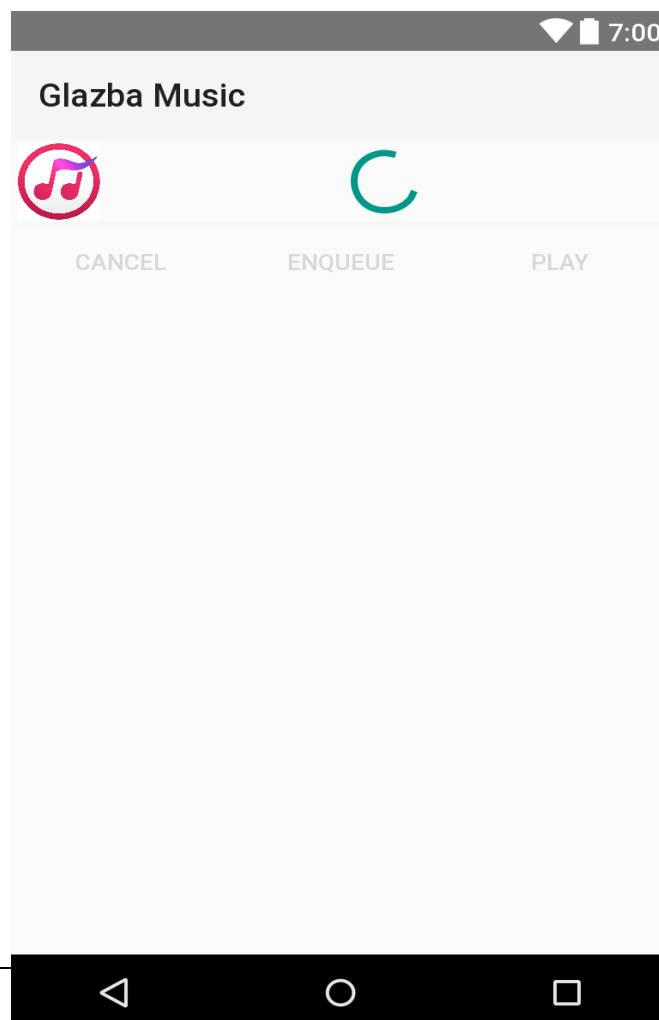
No complicated or but a familiar and

#### Colors

change dynamically main contents base

#### Customizability

inbuilt theme engine different colors to



and above.

#### Easy to Use

overblown menus clean interface.

#### Dynamic

The UI colors to match the color.

There is an with many choose from.

Android provides many ways to control playback of audio/video files and streams. One of this way is through a class called **MediaPlayer**.

Figure 1: AUDIO PICKER

Android is providing MediaPlayer class to access built-in mediaplayer services like playing audio,video e.t.c. In order to use MediaPlayer, we have to call a static Method **create()** of this class. This method returns an instance of MediaPlayer class. Its syntax is as follows –

```
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.song);
```

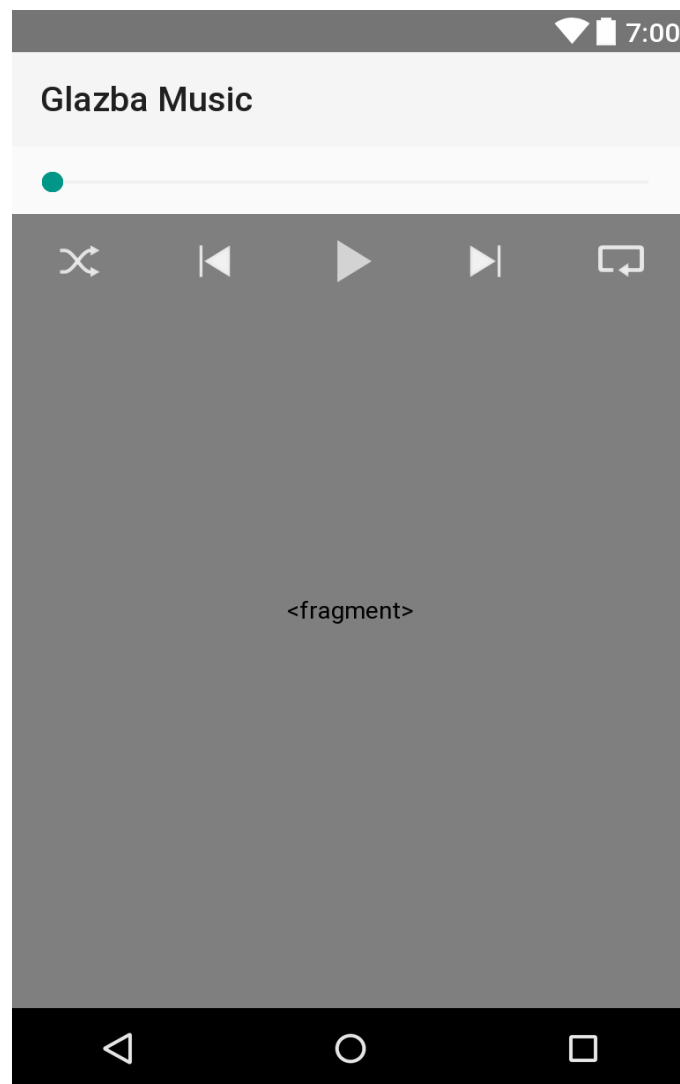


Figure 2: FULL PLAYBACK

The second parameter is the name of the song that you want to play. You have to make a new folder under your project with name **raw** and place the music file into it.

Once you have created the MediaPlayer object you can call some methods to start or stop the music. These methods are listed below.

```
mediaPlayer.start();  
mediaPlayer.pause();
```

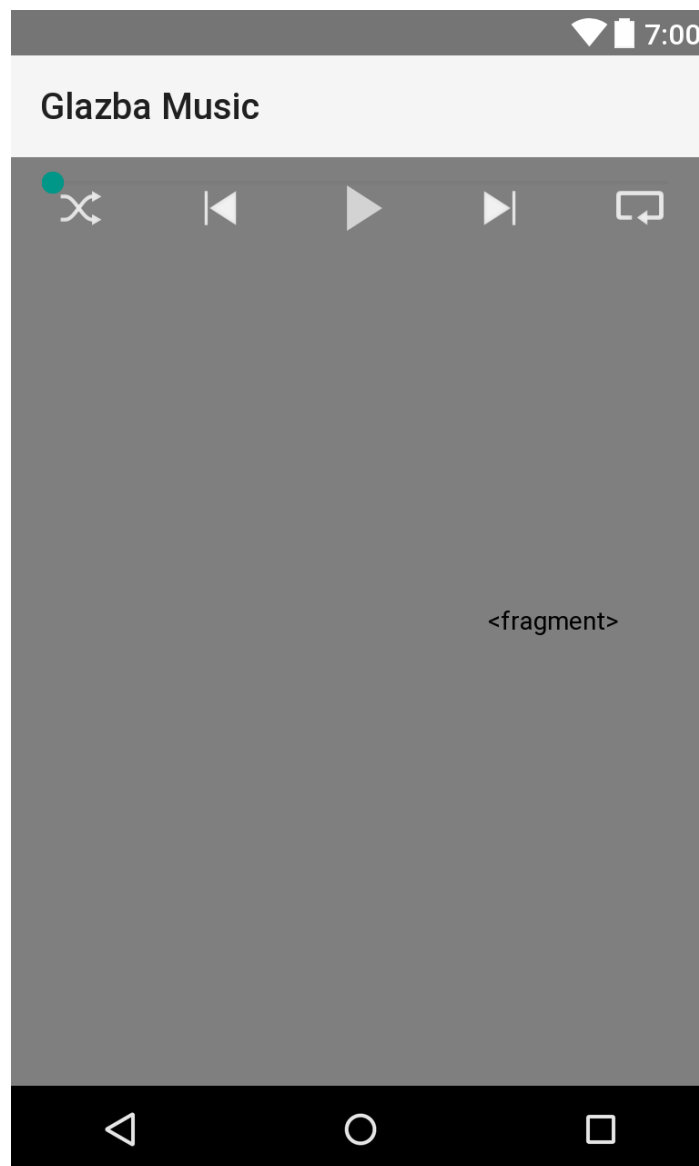


Figure 3: LIBRARY CONTENTS

On call to **start()** method, the music will start playing from the beginning. If this method is called again after the **pause()** method, the music would start playing from where it is left and not from the beginning.

In order to start music from the beginning, you have to call **reset()** method. Its syntax is given below.

```
mediaPlayer.reset();
```

Apart from the start and pause method, there are other methods provided by this class for better dealing with audio/video files.

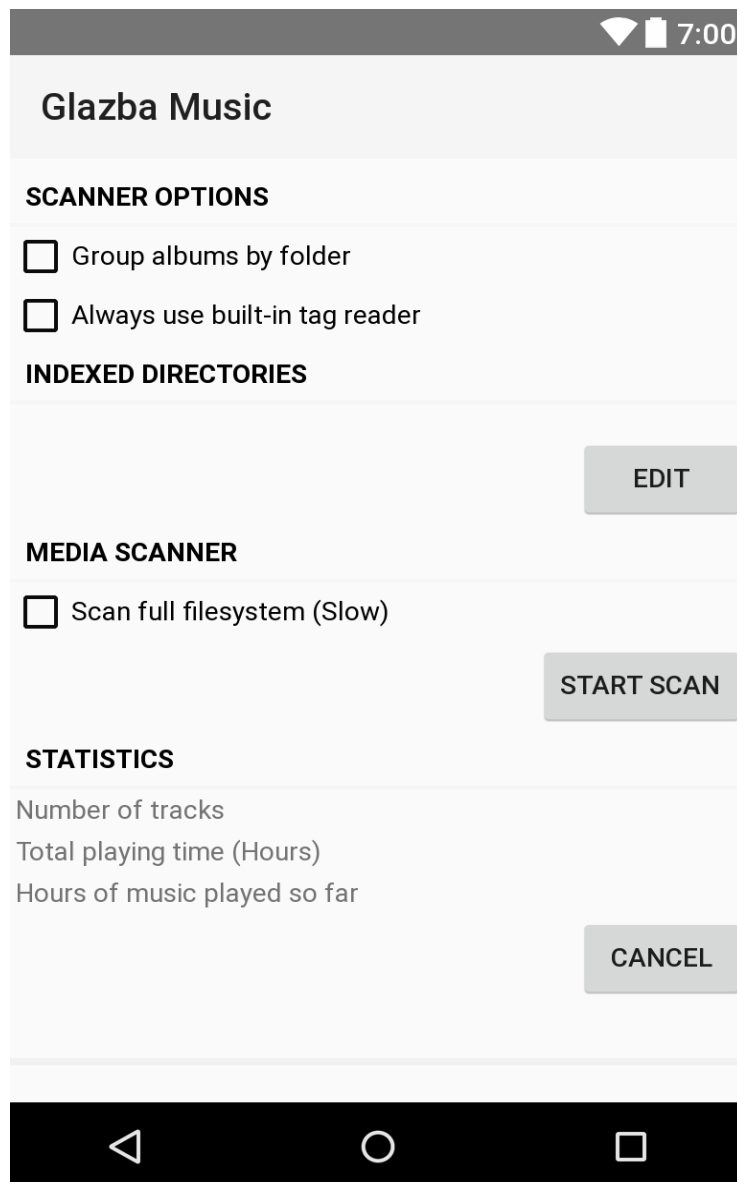


Figure 4: MEDIA LIBRARY PREFERENCES

### **isPlaying()**

1

This method just returns true/false indicating the song is playing or not

### **seekTo(position)**

2

This method takes an integer, and move song to that particular second

### **getCurrentDuration()**

3

This method returns the current position of song in milliseconds

**getDuration()**

4

This method returns the total time duration of song in milliseconds

**reset()**

5

This method resets the media player

**release()**

6

This method releases any resource attached with MediaPlayer object

**setVolume(float leftVolume, float rightVolume)**

7

This method sets the up down volume for this player

**setDataSource(FileDescriptor fd)**

8

This method sets the data source of audio/video file

**selectTrack(int index)**

9

This method takes an integer, and select the track from the list on that particular index

**getTrackInfo()**

10

This method returns an array of track information

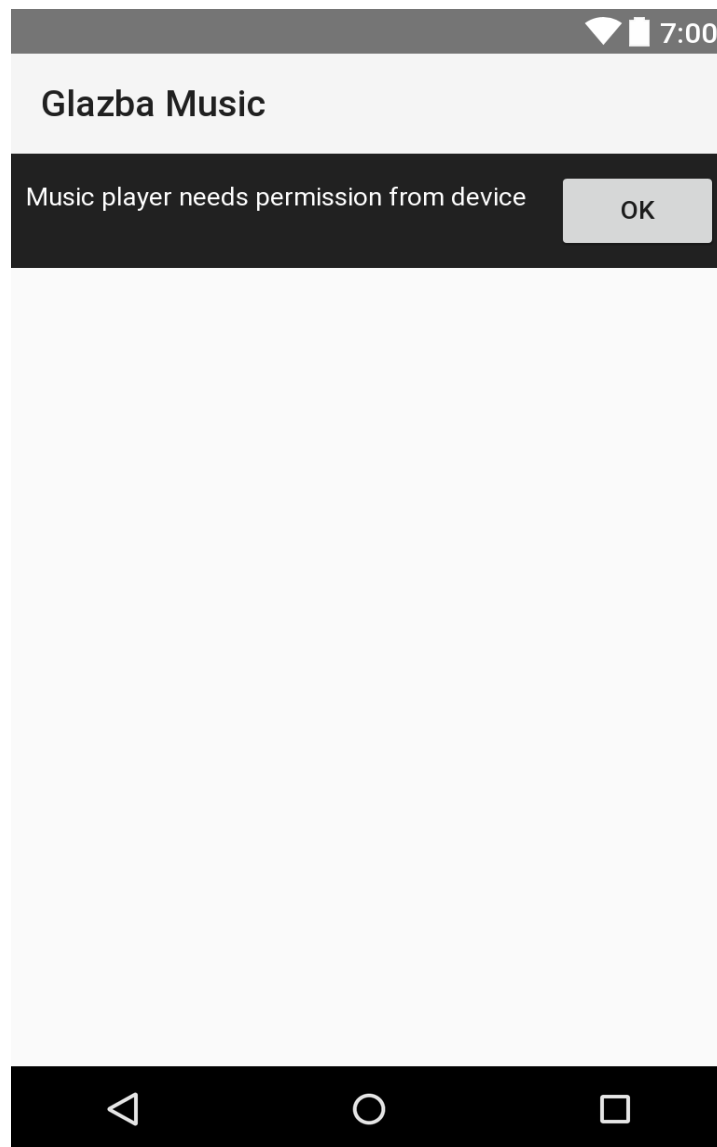


Figure 5: PERMISSION REQUEST

One will use Android studio IDE to create an Android application under a package `com.example.sairamkrishna.myapplication`.

Modify `src/MainActivity.java` file to add MediaPlayer code.

Create a new folder under MediaPlayer with name as raw and place an mp3 music file in it with name as song.mp3

Run the application and choose a running android device and install the application on it and verify the results

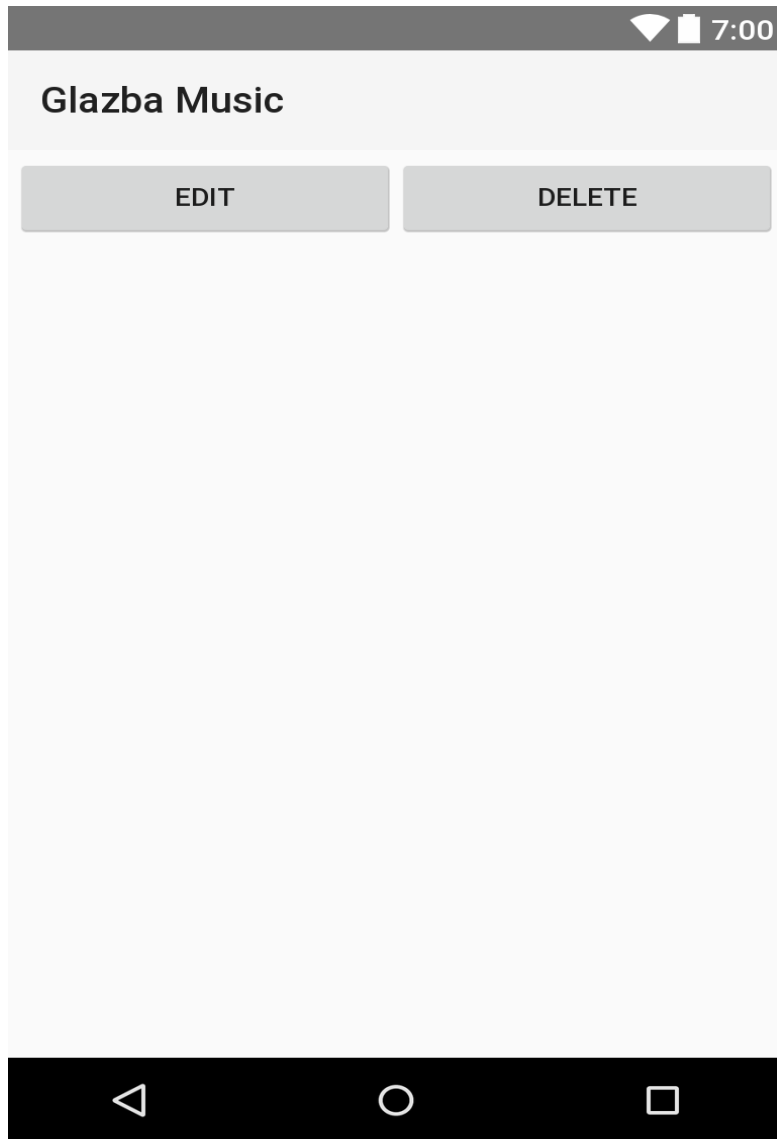


Figure 6: PLAYLIST BUTTONS

The Android multimedia framework includes support for playing variety of common media types, so that you can easily integrate audio, video and images into your applications. You can play audio or video from media files stored in your application's resources (raw resources), from standalone files in the filesystem, or from a data stream arriving over a network connection, all using [MediaPlayer](#) APIs.

This document shows you how to write a media-playing application that interacts with the user and the system in order to obtain good performance and a pleasant user experience.



One can play back the audio data only to the standard output device. Currently, that is the mobile device speaker or a Bluetooth headset. You cannot play sound files in the conversation audio during a call.

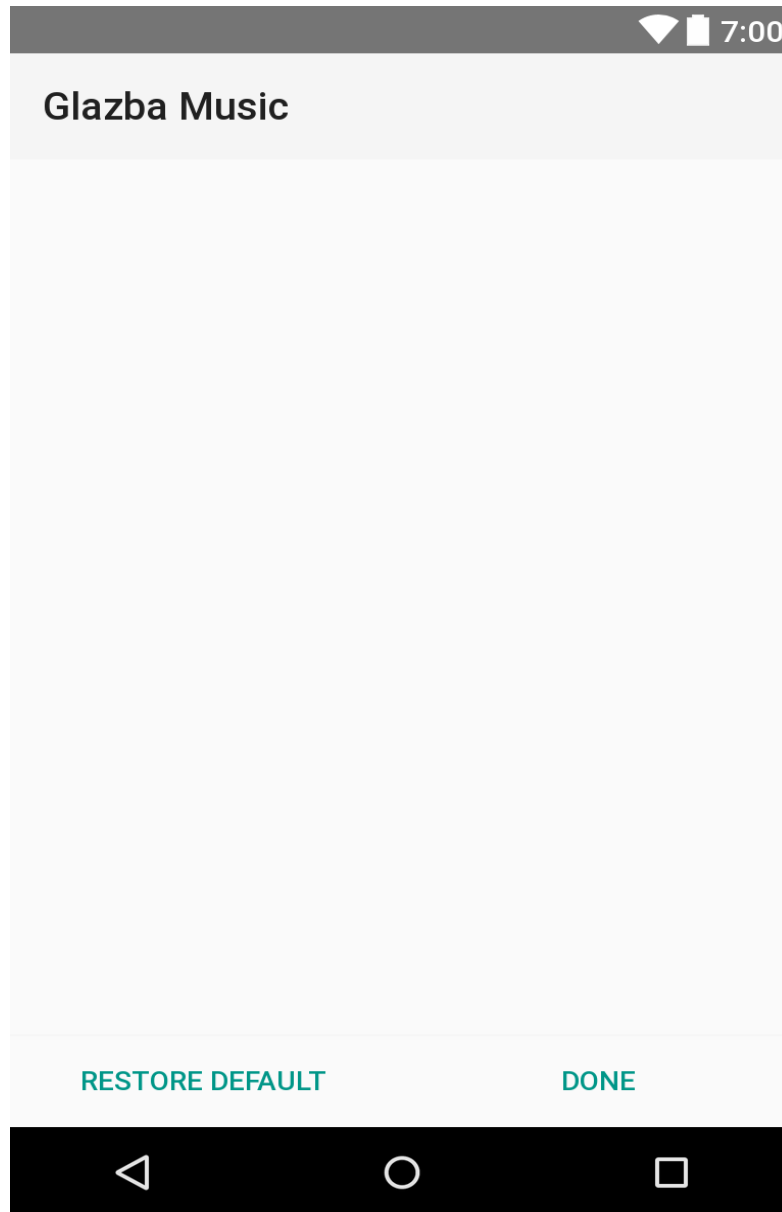


Figure 7: RESTORE DEFUALT SETTINGS

## CHAPTER 3

### SCREENS

#### SLIDER SCREENS

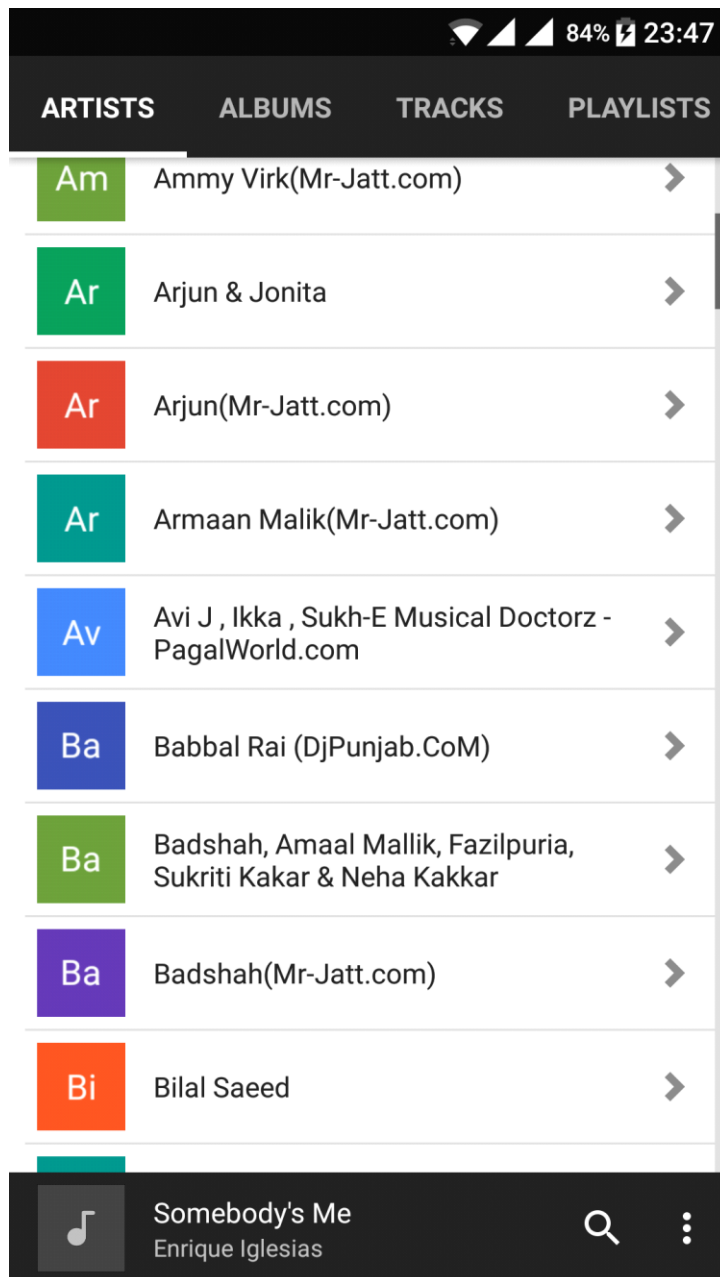


Figure 8: SLIDER SCREEN 1

- This is an example of a slider screen.
- It displays the music in list view.
- It is available by various group by attributes.

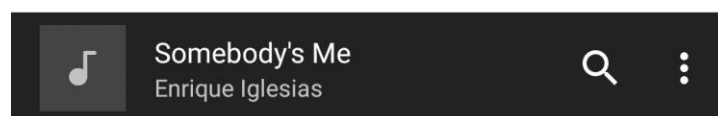
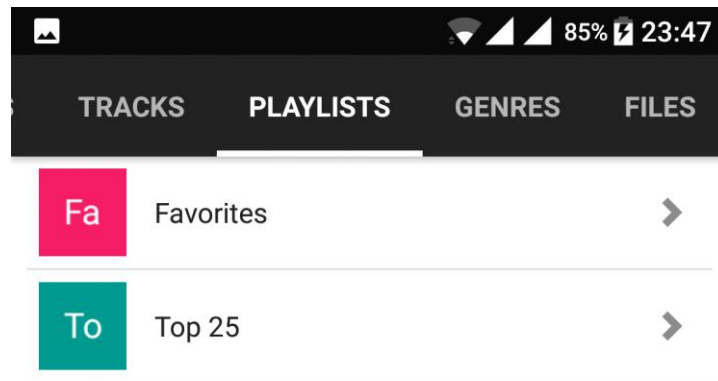


Figure 9: SLIDER SCREEN 2

The following classes are used to play sound and video in the Android framework:

#### [MediaPlayer](#)

This class is the primary API for playing sound and video.

#### [AudioManager](#)

This class manages audio sources and audio output on a device.

## PREFERENCE SCREEN

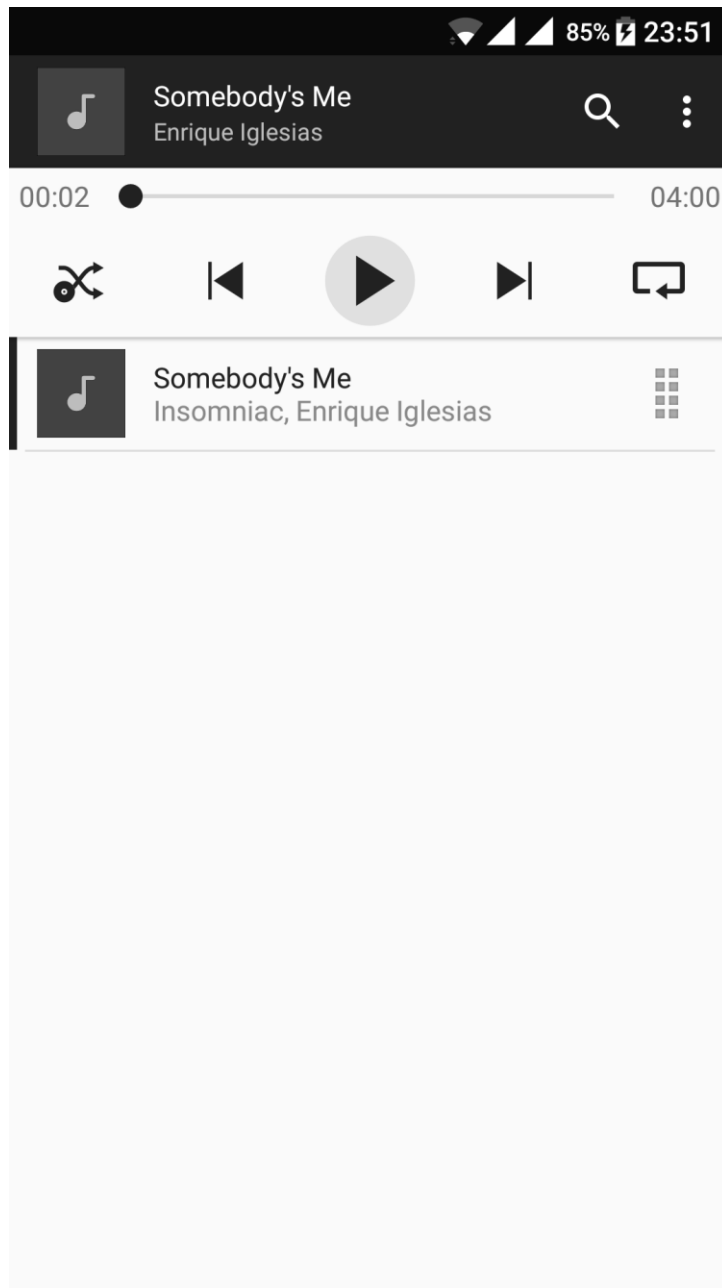


Figure 10: PREFERENCE SCREEN

- This screen displays the queue.
- It provides the media control interface.

## SETTINGS SCREEN

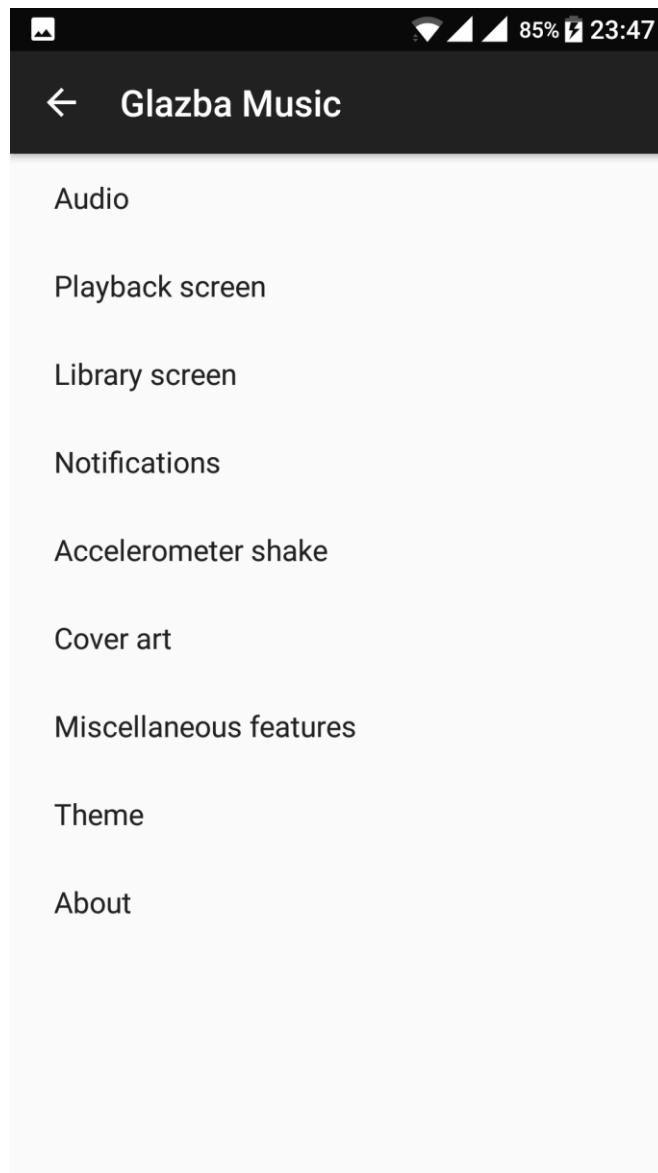


Figure 11: SETTINGS

- Complete Settings tab display
- This screen contains various sub-screens
- "About" tab gives the information about the developers.
- "Accelerometer" tab provides interface for sensor's settings.
- "Miscellaneous" tab contains additional features.
- "Audio" tab provides various features on audio modification.

## CHAPTER 4

### FEATURES

#### BASIC FEATURES

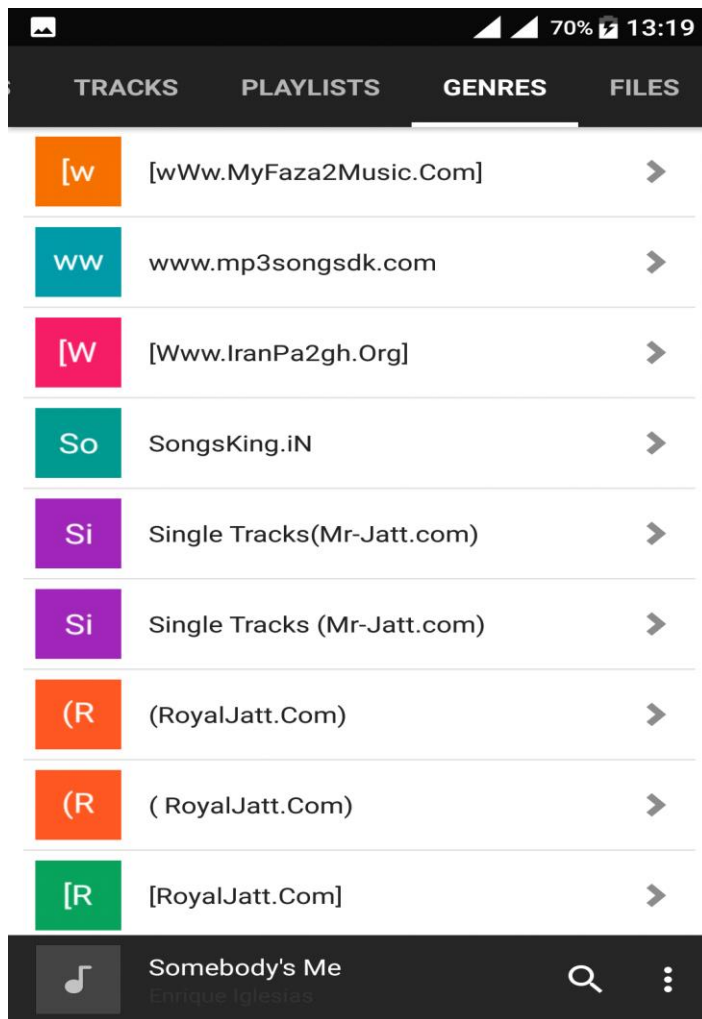


Figure 12: LIST VIEW BY GENRES

- Play and pause buttons
- Next and previous buttons
- Shuffle and repeat buttons
- Search options

- Group by :-
  - i. Artists
  - ii. Albums
  - iii. Tracks
  - iv. Playlists
  - v. Genres
  - vi. Files
  - vii. Reverse
- Favorites
- Notification bars
- Splash screen
- Sort by :-
  - viii. Filename
  - ix. File size
  - x. Modification time
  - xi. Extension
- Queue and Dequeue

## ADDITIONAL FEATURES

### 1. ACCELROMETER SHAKE

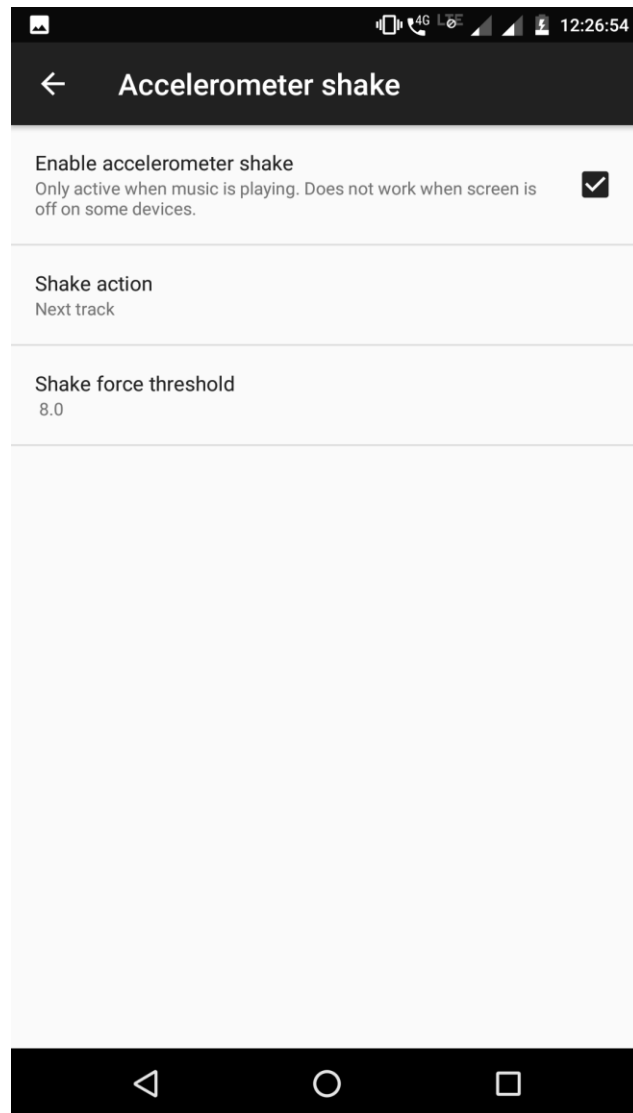


Figure 13: ACCELEROMETER SHAKE

- Accelerometer shake is a unique feature of glazba music
- Accelerometer shake means there are several operations on shaking the phone
- Operations on shaking are : -



- i. Next track
- ii. Previous track
- iii. Seek 10 seconds forward
- iv. Seek 10 seconds backward
- Shake frequency can be changed as per the requirement
- It perform its operations on the background

## 2. Idle Timeout

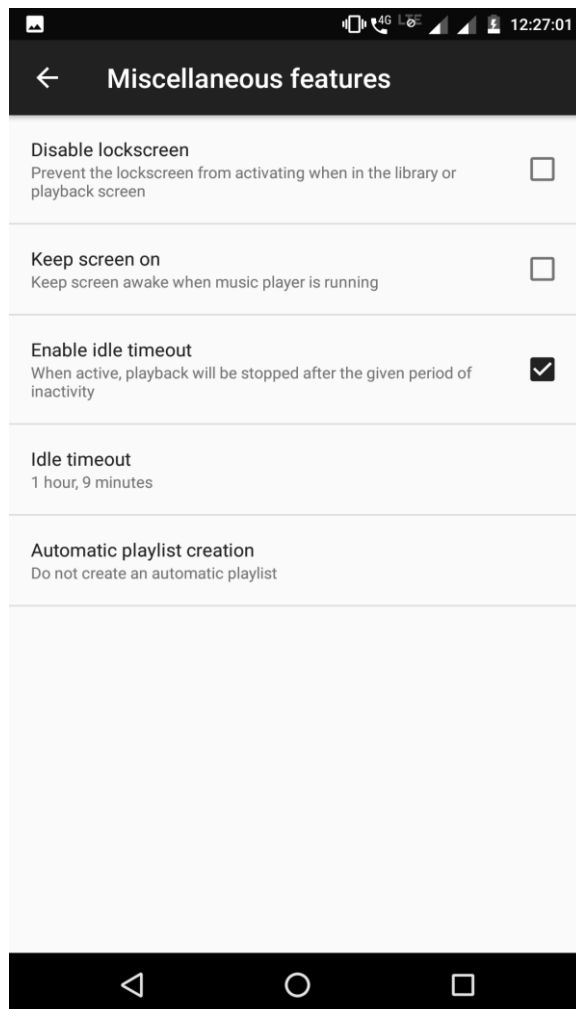


Figure 14: MISCELLANEOUS FEATURES

- Idle timeout is the most unique and attractive features
- Idle Timeout works on the inactivity of the phone
- It automatically closes the applications when the phone is not used
- We can reset the time of the idle timeout
- The timeout can vary from 1 minute to 6 hours

### 3. Theme selection

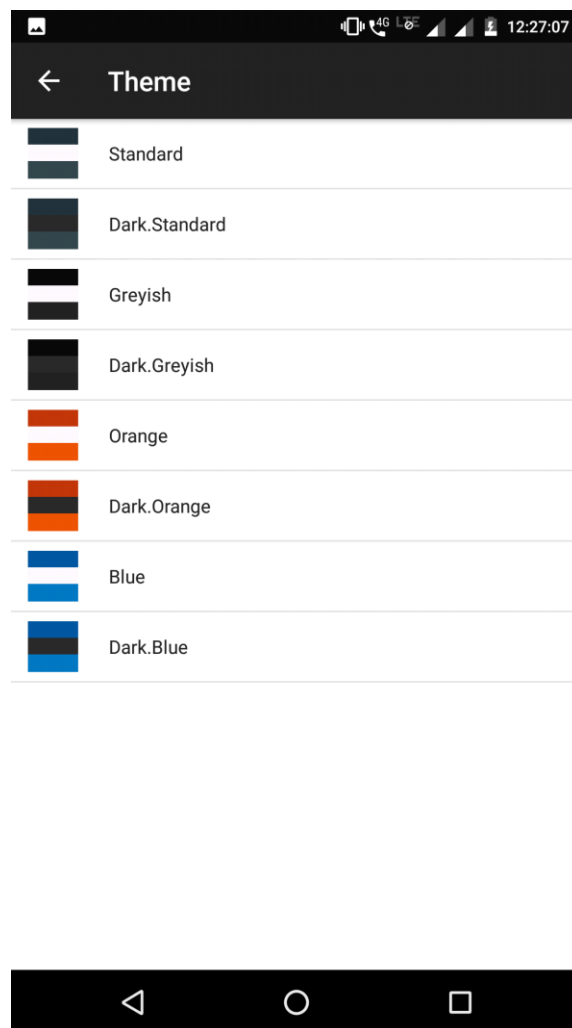


Figure 15: THEME SELECTION

- This feature enables the user to choose among various themes
- These themes can provide various attractive looks to the applications
- There are 8 themes available : -
  - i. Standard
  - ii. Dark Standard
  - iii. Greyish
  - iv. Dark Greyish
  - v. Orange
  - vi. Dark Orange
  - vii. Blue
  - viii. Dark Blue
- This themes also works on the notification bar

## **CONCLUSION**

The music player considers about improving functionality in terms of user interface, sound quality and various looks.

Continuous playing of music without any interruption due to advertisements and it introduces us to many new features like idle time out which stops the music automatically when the smartphone is inactive and accelerometer shake which performs several changes when the device is shaken.

The music application is very compact and the size of apk is mere 650 KB. It testifies its efficiency.

## APPENDIX

### XML CODE FOR AUDIO PICKER

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:divider="?android:attr/dividerHorizontal"
```

```
    android:showDividers="middle"
```

```
    android:orientation="vertical"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content">
```

```
<LinearLayout
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:orientation="horizontal">
```

```
<ImageView android:id="@+id/icon"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_gravity="top"
```

```
    android:layout_marginLeft="4dip"
```

```
    android:layout_marginTop="2dip"
```

```
    android:layout_marginBottom="2dip"
```

```
    android:src="@drawable/icon" />
```

```
<TextView
```

```
    android:id="@+id/filepath"
```

```
    android:visibility="gone"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
```

```

        android:layout_marginBottom="16dip"
        android:layout_marginTop="16dip"
        android:layout_marginLeft="16dip"
        android:layout_marginRight="8dip" />
<ProgressBar
    android:id="@+id/progress"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical|center_horizontal" />
</LinearLayout>

```

```

<LinearLayout
    style="?android:attr/buttonBarStyle"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        style="?android:attr/buttonBarButtonStyle"
        android:id="@+id/cancel"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@android:string/cancel"
        android:enabled="false"
        android:singleLine="true" />
    <Button
        style="?android:attr/buttonBarButtonStyle"
        android:id="@+id/enqueue"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/enqueue"
        android:enabled="false"
        android:singleLine="true" />

```

```
<Button
    style="?android:attr/buttonBarButtonStyle"
    android:id="@+id/play"
    android:layout_width="0px"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/play"
    android:enabled="false"
    android:singleLine="true" />
</LinearLayout>
</LinearLayout>
```

## XML CODE FOR FULL PLAYBACK

```
<?xml version="1.0" encoding="utf-8"?>

<merge xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:vanilla="http://schemas.android.com/apk/res-auto" >
    <ch.blinkenlights.android.glazba.CoverView
        android:id="@+id/cover_view"
        android:layout_height="fill_parent"
        android:layout_width="fill_parent" />
    <LinearLayout
        android:id="@+id/controls_top"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:layout_gravity="top"
        android:background="?overlay_background_color"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/elapsed"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_gravity="center"
            android:paddingLeft="5dip" />
        <SeekBar
            android:id="@+id/seek_bar"
            android:layout_height="wrap_content"
            android:layout_width="0px"
            android:layout_gravity="center"
            android:minHeight="32dip"
            android:maxHeight="32dip"
            android:paddingTop="5dip"
            android:paddingBottom="5dip"
            android:paddingLeft="18dip"
            android:paddingRight="18dip"
```



```

        android:layout_weight="1" />
<TextView
    android:id="@+id/duration"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:paddingRight="5dip" />
</LinearLayout>

<ch.blinkenlights.android.glazba.SlidingView
    android:id="@+id/sliding_view"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:layout_gravity="bottom|left"
    android:layout_marginTop="40dip"
    vanilla:slider_handle_id="@+id/queue_slider">
    <LinearLayout
        android:id="@+id/queue_slider"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:background="?overlay_background_color"
        android:orientation="horizontal"
        android:elevation="2dp">
        <include layout="@layout/controls" />
    </LinearLayout>
    <fragment class="ch.blinkenlights.android.glazba.ShowQueueFragment"
        android:id="@+id/queue"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</ch.blinkenlights.android.glazba.SlidingView>

</merge>

```

## XML CODE FOR LIBRARY CONTENTS

```
<?xml version="1.0" encoding="utf-8"?>
<merge xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:vanilla="http://schemas.android.com/apk/res-auto" >
<LinearLayout
  android:id="@+id/content"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <android.support.iosched.tabs.VanillaTabLayout
    android:id="@+id/sliding_tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:elevation="4dp"
    android:background="?tabs_background" />
  <android.support.v4.view.ViewPager
    android:id="@+id/pager"
    android:layout_width="fill_parent"
    android:layout_height="0px"
    android:layout_weight="1" />
  <HorizontalScrollView
    android:id="@+id/limiter_scroller"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|left">
    <LinearLayout
      android:id="@+id/limiter_layout"
      android:orientation="horizontal"
      android:layout_width="wrap_content"
      android:layout_height="fill_parent" />
  </HorizontalScrollView>
</LinearLayout>
```

```

<ch.blinkenlights.android.glazba.SlidingView
    android:id="@+id/sliding_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_gravity="bottom|left"
    android:orientation="horizontal"
    vanilla:slider_handle_id="@+id/bottombar_controls"
    vanilla:slider_slave_id="@+id/content">
    <LinearLayout
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:background="?overlay_background_color"
        android:orientation="horizontal">
        <include layout="@layout/bottombar_controls"
android:id="@+id/bottombar_controls" />
    </LinearLayout>

```

```

<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:layout_gravity="top"
    android:background="?overlay_background_color"
    android:elevation="2dp"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/elapsed"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:paddingLeft="5dip" />
    <SeekBar
        android:id="@+id/seek_bar"
        android:layout_height="wrap_content"

```

```

        android:layout_width="0px"
        android:layout_gravity="center"
        android:paddingTop="5dip"
        android:paddingBottom="5dip"
        android:paddingLeft="18dip"
        android:paddingRight="18dip"
        android:layout_weight="1" />
    <TextView
        android:id="@+id/duration"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:paddingRight="5dip" />
</LinearLayout>

<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:background="?overlay_background_color"
    android:orientation="horizontal"
    android:elevation="2dp">
    <include layout="@layout/controls" />
</LinearLayout>

<fragment class="ch.blinkenlights.android.glazba.ShowQueueFragment"
    android:id="@+id/queue"
    android:layout_width="596dp" android:layout_height="fill_parent" />
</ch.blinkenlights.android.glazba.SlidingView>
</merge>

```

## XML CODE FOR MEDIA LIBRARY PREFERENCES

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingTop="5dp"
        android:orientation="vertical" >

        <TextView
            style="?android:attr/listSeparatorTextViewStyle"
            android:textColor="?overlay_foreground_color"
            android:text="@string/media_scan_preferences" />
        <CheckBox android:id="@+id/media_scan_group_albums"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/media_scan_group_albums" />
        <CheckBox android:id="@+id/media_scan_force_bastp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/media_scan_force_bastp" />

        <TextView
            style="?android:attr/listSeparatorTextViewStyle"
            android:textColor="?overlay_foreground_color"
            android:text="@string/media_folders_header" />
```

```

<TextView
    android:id="@+id/media_directories"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/empty" />

<Button
    android:id="@+id/edit_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:text="@string/edit">
</Button>

<TextView
    style="?android:attr/listSeparatorTextViewStyle"
    android:textColor="?overlay_foreground_color"
    android:text="@string/media_scan_header" />

<CheckBox android:id="@+id/media_scan_full"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/media_scan_full" />

<Button
    android:id="@+id/start_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:text="@string/media_scan_start">
</Button>

<TextView
    style="?android:attr/listSeparatorTextViewStyle"

```

```
android:textColor="?overlay_foreground_color"
android:text="@string/media_statistics" />
```

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="@string/media_stats_tracks"
            android:padding="3dip" />
        <TextView
            android:id="@+id/media_stats_tracks"
            android:text="@string/empty"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
    <TableRow>
        <TextView
            android:text="@string/media_stats_library_playtime"
            android:padding="3dip" />
        <TextView
            android:text="@string/empty"
            android:id="@+id/media_stats_library_playtime"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
    <TableRow>
        <TextView
            android:text="@string/media_stats_listen_playtime"
            android:padding="3dip" />
        <TextView
            android:text="@string/empty"
            android:id="@+id/media_stats_listen_playtime"
```

```
        android:gravity="right"
        android:padding="3dip" />
    </TableRow>
</TableLayout>
```

```
<Button
    android:id="@+id/cancel_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:text="@android:string/cancel">
</Button>
```

```
<ProgressBar
    android:id="@+id/media_stats_progress_bar"
    android:paddingTop="24dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    style="?android:attr/progressBarStyleHorizontal" />
```

```
<TextView
    android:id="@+id/media_stats_progress_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minLines="3"
    android:text="@string/empty" />
</LinearLayout>
```

```
</ScrollView>
```



## XML CODE FOR PERMISSION REQUEST

```
<?xml version="1.0" encoding="utf-8"?>

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:clickable="true"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="0"
    android:shrinkColumns="0"
    android:paddingTop="8dip"
    android:paddingBottom="8dip"
    android:background="#FF212121">
    <TableRow>
        <TextView
            android:id="@+id/permission_request_title"
            android:layout_marginLeft="8dip"
            android:layout_marginBottom="8dip"
            android:layout_marginTop="8dip"
            android:text="Music player needs permission from device"
            android:textColor="#FFFFFFFF" />
        <Button
            android:id="@+id/permission_request_button"
            android:clickable="false"
            android:layout_marginRight="8dip"
            android:layout_gravity="center"
            android:text="@android:string/ok" />
    </TableRow>
</TableLayout>
```

## XML CODE FOR PLAYLIST BUTTONS

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="3dip"
    android:orientation="horizontal">
    <Button
        android:id="@+id/edit"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/edit" />
    <Button
        android:id="@+id/delete"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/delete" />
</LinearLayout>
```

## XML CODE FOR RESTORING DEAFULT SETTINGS

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:dslv="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:divider="?android:attr/dividerHorizontal"
    android:showDividers="middle"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent">
    <com.mobeta.android.dslv.DragSortListView
        android:id="@+id/list"
        android:divider="@drawable/inset_divider"
        android:dividerHeight="1dip"
        android:layout_height="0px"
        android:layout_width="fill_parent"
        android:layout_weight="1"
        android:choiceMode="multipleChoice"
        dslv:drag_enabled="true"
        dslv:drag_start_mode="onMove"
        dslv:float_background_color="?float_color"
        dslv:drag_handle_id="@+id/dragger" />
    <LinearLayout
        style="?android:attr/buttonBarStyle"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            style="?android:attr/buttonBarButtonStyle"
            android:id="@+id/restore_default"
            android:layout_width="0px"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/restore_default" />
    <Button
        style="?android:attr/buttonBarButtonStyle"
        android:id="@+id/done"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/done" />
</LinearLayout>
</LinearLayout>
```