B.M.S. College of Engineering, Bangalore-19

Final Review

"Music Player App"

Submitted by:
Injeel Fatima (1BM19IS060)
Nidhi Prakash (1BM19S100)

Work done as part of the Alternative Assessment for the V Semester Undergraduate (UG) Course **Mobile Application Development (20IS5PWMAD)** during the Academic Year 2021-22.

Faculty Incharge
Dr. Shubha Rao V

Department of Information Science & Engineering

Table of Contents

Sl No Page No.

Abstract

Music is a part of every person's life. No matter what your mood is, you have a song to sustain that mood. If you wish to play your desired songs on Android devices, you need a music player. So, through this article, we will try to build our music player app using Android.

The music player app that we will develop in this article would allow the users to play the songs present on the device. You can download songs on your device and then use the music player to play those songs. Let's see the quite exciting features that you get along with this app.

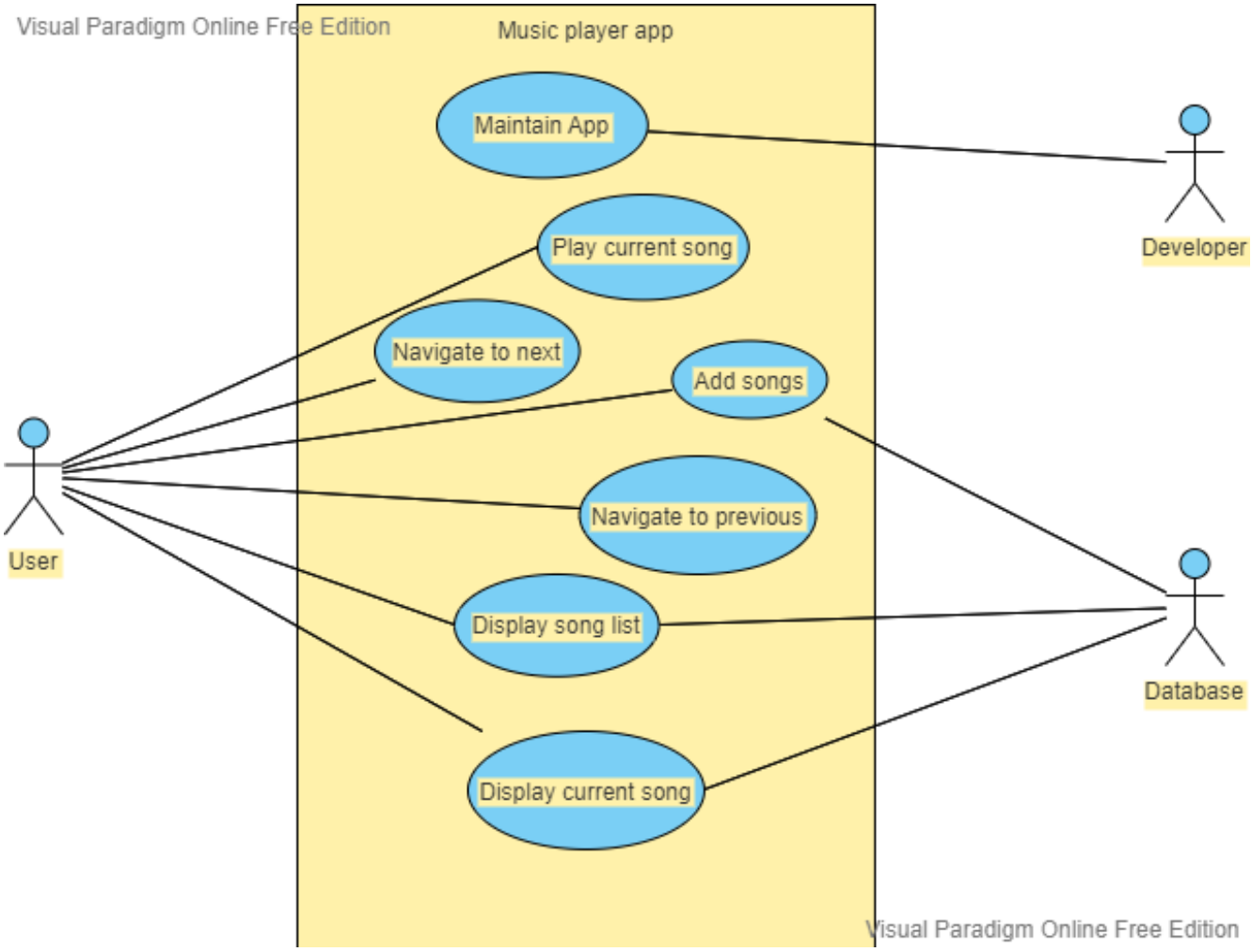
1. Introduction

Characters of varying degrees that are found in music, can affect one's mood. Music can raise someone's mood, get them excited, or make them calm and relaxed. Music also - and this is important - allows us to feel nearly or possibly all emotions that we experience in our lives. The possibilities are endless.

We have created a music app for all moods. This music app reads a list of songs from an external file, shows the list of songs stored in the external storage. It has all the basic characteristics

2. System Architecture

Use Case Diagram



3. User Interface Design and Navigation

Providing an user-friendly interface is of the functional requirements of the application and hence care has been taken to use best practices while designing the overall UI

The Various UI components used are the following :

- TextView- It is used to display texts which are meant to be not editable by the user by default (can change parameters to allow editing). This has been used for headings in the app
- EditText- It is used to collect text from the user. It's an editable text box where users can enter values.
- ImageView- ImageView is used to display images in android applications. An image can be displayed by assigning it to the ImageView control and including the android:src attribute in the xml definition. Images can be assigned dynamically.
- Button- A UI component which when clicked performs a certain task.
- Seekbar - Android SeekBar is a kind of ProgressBar with a draggable thumb. The end user can drag the thumb left and right to move the progress of the song, file download etc. In our project, it is used to track the progress of songs playing, and to seek to the position that we want to go to in the song.
- Intent- A navigation component used for communication and navigating from one activity to another
- Animation- Animation has been used for the initial splash screen as well in navigating between forms in the application.
- Dexter - Dexter is the library that will help us to make this task easy for handling runtime permissions in Android.

4. Design

XML Designs

Activity_main.xml:

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

<androidx.constraintlayout.widget.ConstraintLayout
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"

    android:background="#099"

    tools:context=".MainActivity">



    <ListView

        android:id="@+id/listView"

        android:divider="#099"
```

```
        android:dividerHeight="10sp"

        android:padding="8dp"

        android:layout_width="match_parent"

        android:layout_height="match_parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_play_song.xml:

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

<androidx.constraintlayout.widget.ConstraintLayout
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"

    android:background="@drawable/music_background"

    android:orientation="vertical"
```

```
    tools:context=".PlaySong">


    <ImageView

        android:id="@+id/imageView2"

        android:layout_width="199dp"

        android:layout_height="342dp"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.424"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.303"

        app:srcCompat="@drawable/logo" />


    <TextView

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

```
        android:ellipsize="marquee"

        android:textColor="#FFF"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:textSize="24sp"

        android:textStyle="bold"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.435"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.698" />



    <SeekBar

        android:id="@+id/seekBar"

        android:layout_width="304dp"
```

```
        android:layout_height="21dp"

        android:outlineAmbientShadowColor="#FFF"

        android:outlineSpotShadowColor="#FFF"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.451"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.757" />



    <TextView

        android:id="@+id/txtsstrart"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentLeft="true"

        android:layout_centerInParent="true"
```

```
        android:layout_toLeftOf="@+id/seekBar"

        android:text="0:10"

        android:textColor="#FFF"

        android:textSize="14sp"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.041"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.758"

        tools:ignore="MissingConstraints,UnknownId">



    </TextView>



    <TextView

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

```
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentLeft="true"

        android:layout_centerInParent="true"

        android:layout_toRightOf="@+id/seekBar"

        android:text="4:10"

        android:textColor="#FFF"

        android:textSize="14sp"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.916"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.758"

        tools:ignore="MissingConstraints,UnknownId">
```

```
    </TextView>


    <LinearLayout

        android:id="@+id/linearLayout"

        android:layout_width="360dp"

        android:layout_height="88dp"

        android:orientation="horizontal"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.313"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.9">


        <ImageView

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

```
            android:layout_width="match_parent"

            android:layout_height="match_parent"

            android:layout_weight="1"

            app:srcCompat="@drawable/previous" />



        <ImageView

            android:id="@+id/play"

            android:layout_width="match_parent"

            android:layout_height="match_parent"

            android:layout_weight="1"

            app:srcCompat="@drawable/play" />



        <ImageView

            android:id="@+id/next"

            android:layout_width="match_parent"

            android:layout_height="match_parent"
```

```
            android:layout_weight="1"

            app:srcCompat="@drawable/next" />



    </LinearLayout>


    <com.gauravk.audiovisualizer.visualizer.BarVisualizer

        xmlns:custom="http://schemas.android.com/apk/res-auto"

        android:id="@+id/blast"

        android:layout_width="384dp"

        android:layout_height="60dp"

        android:layout_alignParentBottom="true"

        app:avColor="#089"

        app:avDensity="0.5"

        app:avSpeed="normal"
```

```
        app:avType="outline"

        app:avWidth="4dp"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.592"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.994" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

List_item.xml:

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

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_marginEnd="8dp"

    android:layout_marginTop="8dp"
```

```
    android:layout_marginStart="8dp"

    >

    <RelativeLayout

        android:padding="8dp"

        android:background="@drawable/list_bg"

        android:layout_width="match_parent"

        android:layout_height="wrap_content">



        <ImageView

            android:id="@+id/imgsong"

            android:layout_alignParentStart="true"

            android:layout_alignParentTop="true"

            android:layout_marginStart="5dp"

            android:layout_marginTop="5dp"

            android:background="@drawable/list_bg"

            android:layout_width="40dp"
```

```
            android:layout_height="40dp"

            android:src="@drawable/music_note"

            android:contentDescription="TODO" />



        <TextView

            android:id="@+id/textsongname"

            android:layout_alignParentEnd="true"

            android:layout_marginStart="5dp"

            android:layout_marginEnd="5dp"

            android:layout_toEndOf="@+id/imgsong"

            android:padding="6dp"

            android:textColor="#FFF"

            android:text="Song Name"

            android:textSize="15sp"

            android:singleLine="true"

            android:marqueeRepeatLimit="marquee_forever"
```

```
            android:ellipsize="marquee"

            android:scrollHorizontally="true"

            android:layout_width="wrap_content"

            android:layout_height="40dp"



            />

    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

5. Implementation

MainActivity.java:

```
package com.example.my_music;

import static android.os.Environment.getExternalStorageState;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;
import com.karumi.dexter.listener.single.PermissionListener;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    ListView listView;
    String[] items;
    @Override
    protected void onCreate(Bundle savedInstanceState) {


        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = findViewById(R.id.listView);
        runtimePermission();
    }
    public void runtimePermission()
    {

Dexter.withContext(this).withPermissions(Manifest.permission.READ_EXTERNAL_STORAGE,Manifest.permiss
```

```
ion.RECORD_AUDIO)
                .withListener(new MultiplePermissionsListener() {
                    @Override
                    public void onPermissionsChecked(MultiplePermissionsReport
multiplePermissionsReport) {
                        displaySongs();
                    }

                    @Override
                    public void onPermissionRationaleShouldBeShown(List<PermissionRequest> list,
PermissionToken permissionToken) {
                        permissionToken.continuePermissionRequest();
                    }
                }).check();




    }
    class customAdapter extends BaseAdapter
    {

        @Override
        public int getCount() {

            return items.length;
        }

        @Override
        public Object getItem(int i) {
            return null;
        }

        @Override
        public long getItemId(int i) {
            return 0;
        }

        @Override
        public View getView(int i, View view, ViewGroup viewGroup) {
            View myview = getLayoutInflater().inflate(R.layout.list_item,null);
            TextView textsong = myview.findViewById(R.id.textsongname);
            textsong.setSelected(true);
            textsong.setText(items[i]);
            return myview;
        }
    }
    public ArrayList<File> fetchSongs(File file) {
        ArrayList arraylist = new ArrayList();
        File[] songs = file.listFiles();
        if (songs != null) {
            for (File myFile : songs) {
```

```
                if (!myFile.isHidden() && myFile.isDirectory()) {
                    arraylist.addAll(fetchSongs(myFile));
                } else {
                    if (myFile.getName().endsWith(".mp3") && !myFile.getName().startsWith(".")) {
                        arraylist.add(myFile);
                    }
                }
            }

        }
        return arraylist;
    }
    void displaySongs()
    {
        ArrayList<File> mysongs= fetchSongs(Environment.getExternalStorageDirectory());
        Toast.makeText(getApplicationContext(),getExternalStorageState(),Toast.LENGTH_SHORT).show();
        items = new String[mysongs.size()];
        for(int i=0;i<mysongs.size();i++)
        {
            items[i] = mysongs.get(i).getName().replace(".mp3","");

        }

        customAdapter customAdapter = new customAdapter();
        listView.setAdapter(customAdapter);
                        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                            @Override
                            public void onItemClick(AdapterView<?> adapterView, View view, int
position, long id) {
                                String songname = (String) listView.getItemAtPosition(position);
                                startActivity(new Intent(getApplicationContext(),PlaySong.class)
                                .putExtra("songList",mysongs)
                                .putExtra("songname", songname)
                                .putExtra("pos",position)
                                );


                            }
                        });
        /*listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position , long id)
            {
                Intent intent = new Intent(MainActivity.this,PlaySong.class);
                String currentSong = listView.getItemAtPosition(position).toString();
                intent.putExtra("songList",mysongs);
                intent.putExtra("currentSong", currentSong);
                intent.putExtra("position", position);
                startActivity(intent);
            }
        });*/
    }
```

```
}
```

PlaySong.java:

```
package com.example.my_music;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.TextView;

import java.io.File;
import java.util.ArrayList;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import com.gauravk.audiovisualizer.visualizer.BarVisualizer;

import java.io.File;
import java.util.ArrayList;

public class PlaySong extends AppCompatActivity {
    TextView txtsstart,txtsstop;
    ImageView imageView;
    BarVisualizer visualizer;
     protected void onDestroy() {
         if(visualizer != null) {
             visualizer.release();
         }
        super.onDestroy();
        mediaPlayer.stop(); //stops media player
        mediaPlayer.release(); //releases media player
        updateSeek.interrupt();
    }
    TextView textView;
```

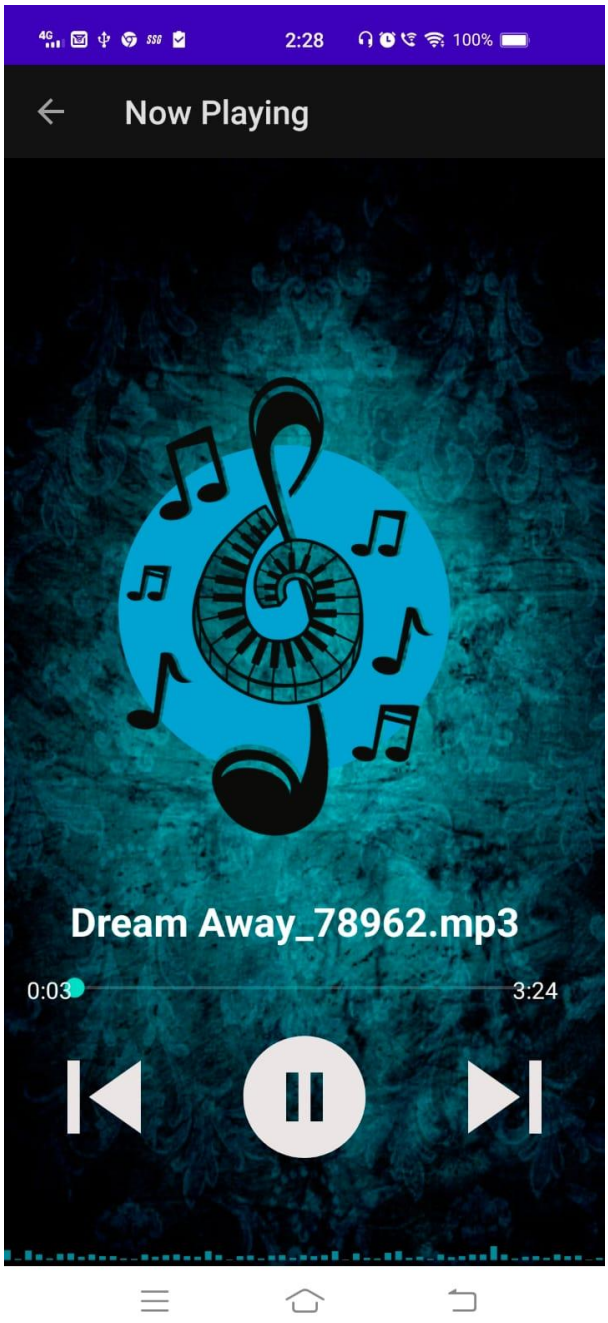
```
    ImageView play, previous, next;
    ArrayList<File> songs;
    MediaPlayer mediaPlayer;
    String textContent;
    int position;
    SeekBar seekBar;
    Thread updateSeek;

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            onBackPressed();
        }
        return super.onOptionsItemSelected(item);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getSupportActionBar().setTitle("Now Playing");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);


        setContentView(R.layout.activity_play_song);
        textView = findViewById(R.id.textView);
        play = findViewById(R.id.play);
        previous = findViewById(R.id.previous);
        next = findViewById(R.id.next);
        seekBar = findViewById(R.id.seekBar);
        visualizer= findViewById(R.id.blast);
        txtsstart= findViewById(R.id.txtsstrart);
        txtsstop= findViewById(R.id.txtsstop);
        imageView = findViewById(R.id.imageView2);

        Intent intent = getIntent();
        Bundle bundle = intent.getExtras();
        songs = (ArrayList)bundle.getParcelableArrayList("songList");
        textContent = intent.getStringExtra("currentSong");
        textView.setText(textContent);
        textView.setSelected(true);
        position = intent.getIntExtra("position", 0);
        Uri uri = Uri.parse(songs.get(position).toString());
        mediaPlayer = MediaPlayer.create(getApplicationContext(), uri);
        mediaPlayer.start();
        seekBar.setMax(mediaPlayer.getDuration());

        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int i, boolean b) {

            }
```

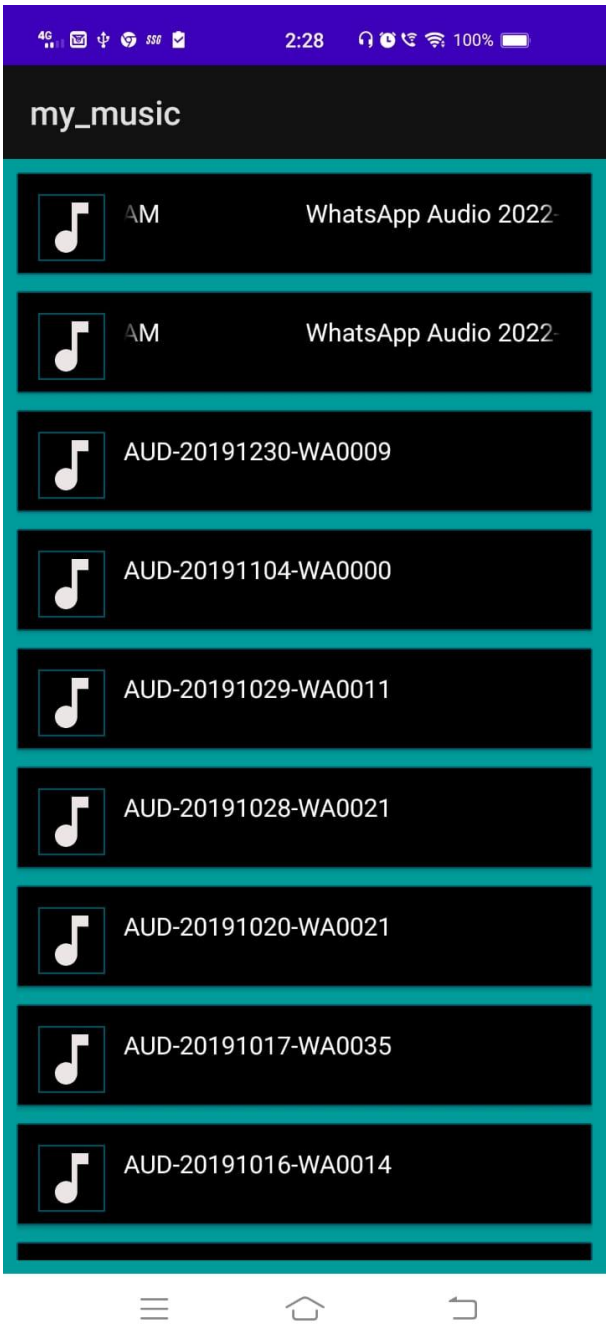
```
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                mediaPlayer.seekTo(seekBar.getProgress());
            }
        });
        updateSeek = new Thread() {
            public void run() {
                int currentPosition = 0;
                try {
                    while(currentPosition<mediaPlayer.getDuration()) {
                        currentPosition = mediaPlayer.getCurrentPosition();
                        seekBar.setProgress(currentPosition);
                        sleep(800);
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        };
        updateSeek.start();

        String endTime = createTime(mediaPlayer.getDuration());
        txtsstop.setText(endTime);

        final Handler handler = new Handler();
        final int delay = 1000;
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                String currentTime = createTime(mediaPlayer.getCurrentPosition());
                txtsstart.setText(currentTime);
                handler.postDelayed(this, delay);
            }
        }, delay);

        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(mediaPlayer.isPlaying()) {
                    play.setImageResource(R.drawable.play);
                    mediaPlayer.pause();
                }
                else {
                    play.setImageResource(R.drawable.pause);
                    mediaPlayer.start();
                }
```

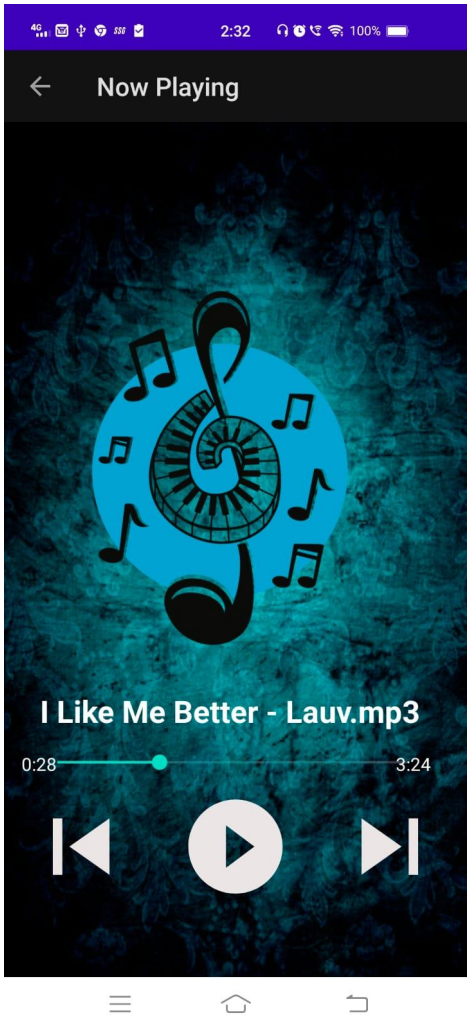
```
            }
        });
        mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
            @Override
            public void onCompletion(MediaPlayer mediaPlayer) {
                next.performClick();
            }
        });

        int audiosessionId = mediaPlayer.getAudioSessionId();
        if(audiosessionId != -1) {
            visualizer.setAudioSessionId(audiosessionId);
        }
        previous.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mediaPlayer.stop();
                mediaPlayer.release();
                if(position-1 >= 0) {
                    position = position - 1;
                } else {
                    position = songs.size() - 1;
                }
                Uri uri = Uri.parse(songs.get(position).toString());
                mediaPlayer = MediaPlayer.create(getApplicationContext(), uri);
                mediaPlayer.start();
                play.setImageResource(R.drawable.pause);
                startAnimation(imageView);
                seekBar.setMax(mediaPlayer.getDuration());
                textContent = songs.get(position).getName().toString();
                textView.setText(textContent);
                int audiosessionId = mediaPlayer.getAudioSessionId();
                if(audiosessionId != -1) {
                    visualizer.setAudioSessionId(audiosessionId);
                }
            }
        });
        next .setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mediaPlayer.stop();
                mediaPlayer.release();
                position=((position+1)%songs.size());
                Uri uri = Uri.parse(songs.get(position).toString());
                mediaPlayer = MediaPlayer.create(getApplicationContext(), uri);
                mediaPlayer.start();
                play.setImageResource(R.drawable.pause);
                startAnimation(imageView);

                seekBar.setMax(mediaPlayer.getDuration());
                textContent = songs.get(position).getName().toString();
                textView.setText(textContent);
```

```
                int audiosessionId = mediaPlayer.getAudioSessionId();
                if(audiosessionId != -1) {
                    visualizer.setAudioSessionId(audiosessionId);
                }
            }
        });

    }
    public void startAnimation(View view) {
        ObjectAnimator animator = ObjectAnimator.ofFloat(imageView, "rotation", 0f, 360f);
        animator.setDuration(1000);
        AnimatorSet animatorSet = new AnimatorSet();
        animatorSet.playTogether(animator);
        animatorSet.start();
    }
    public String createTime(int duration) {
        String time = "";
        int min = duration/1000/60;
        int sec = duration/1000%60;
        time+=min+":";
        if(sec < 10) {
            time +="0";
        }
        time+=sec;
        return time;
    }
}
```

6. Snapshots


2:28
100%
Now Playing
Dream Away_78962.mp3
0:03
3:24


2:28
100%
my_music
AM
WhatsApp Audio 2022-
AM
WhatsApp Audio 2022-
AUD-20191230-WA0009
AUD-20191104-WA0000
AUD-20191029-WA0011
AUD-20191028-WA0021
AUD-20191020-WA0021
AUD-20191017-WA0035
AUD-20191016-WA0014


2:32
100%
Now Playing
I Like Me Better - Lauv.mp3
0:28
3:24


2:32
100%
Now Playing
Tu Aake Dekhle(PagalWorld).mp3
0:00
3:24


2:34
100%
Now Playing
Tu Aake Dekhle(PagalWorld).mp3
1:20
3:24

6. Testing

1. List songs in the external storage

Test Scenario ID	List-songs	**Test Case ID**	LS-1A	
Test Case Description	Creative-Positive Case	**Test Priority**	High	
Prerequisite	Songs in External storage	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Launch Application	Press the app icon	App opens	App opens	Vivo v17	Pass	[Injeel 21/01/2022 12:00 PM]: Launch successful
2.	List view of songs in the external storage	Open App	List of songs appear on the screen	List of songs appear on the screen	Vivo v17	Pass	[Injeel 21/01/2022 12:02 PM]: Songs appear successfully

2. Song status

Test Scenario ID	Song status	**Test Case ID**	SS-1A	
Test Case Description	Play-Positive Case	**Test Priority**	High	
Prerequisite	Song is paused	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Launch Application	Press the app icon	App Screen Displays	App Screen Displayed	Vivo v17	Pass	[Injeel 21/01/2022 12:07 PM]: Launch successful
2.	Play song	Tap on the play button	Current song starts playing	Current song starts playing	Vivo v17	Pass	[Injeel 21/01/2022 12:09 PM]: Play successful

Test Scenario ID	Song status	**Test Case ID**	SS-1B	
Test Case Description	Pause-positive case	**Test Priority**	High	
Prerequisite	Song is Playing	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Pause current song	Press the pause button	Current playing song pauses	Current playing song pauses	Vivo v17	Pass	[Injeel 21/01/2022 12:12 PM]: Pause successful
2.	Pause current song, play and pause	Tap the play button, then the pause, then the play button	Song plays, pauses, and plays	Song plays, pauses and plays successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:13 PM]: Consecutive play pause successful

Test Scenario ID	Song status	**Test Case ID**	SS-1C	
Test Case Description	Seekbar-Positive case	**Test Priority**	High	
Prerequisite	Song played	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Display seekbar progress	Tap the play button	The seekbar progresses in accordance with the song	The seekbar progresses in accordance with the song successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:15 PM]: seekbar progress successful
2.	Seek to desired timestamp	Tap on seekbar and drag to the desired time stamp	Song plays from the seeked time stamp	Song plays from the seeked timestamp successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:23 PM]: Seekbar updated successfully

3. Navigate song-list

Test Scenario ID	Navigate song-list	**Test Case ID**	NSL-1A	
Test Case Description	Navigate to next song	**Test Priority**	Moderate	
Prerequisite	Current song is playing	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Play next	Tap on the next button	Next song starts playing	Next song starts playing successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:35 PM]: next song plays successfully
2.	Play next on last song	Tap on the next button	First song starts playing	First song starts playing successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:36 PM]: next song plays successfully

Test Scenario ID	Navigate song-list	**Test Case ID**	NSL-1B	
Test Case Description	Navigate to previous song	**Test Priority**	Moderate	
Prerequisite	Current song is playing	**Post-Requisite**	Nil	

Test Execution Steps:

S.No	**Action**	**Inputs/Actions**	**Expected Output**	**Actual Output**	**Test Phone**	**Test Result**	**Test Comments**
1.	Previous song	Tap on the previous button	Previous song plays	Previous song plays successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:45 PM]: Previous song play successful
2.	Play previous on first song	Tap on the previous button	Last song starts playing	Last song starts playing successfully	Vivo v17	Pass	[Injeel 21/01/2022 12:50 PM]: Last song play successful

References:

- Android Studio docs: https://developer.android.com/docs
- Material IO UI components: https://material.io/design/color/the-color-system.html
- Coolors for color palette: https://coolors.co/
- For the java code: https://www.javatpoint.com/android-seekbar-example

https://www.javatpoint.com/android-intent-tutorial
https://www.tutorialspoint.com/android/android_mediaplayer.htm