MP3 Music Player

IKRAMUL HOSSAIN
ID:191-15-2505
Pc-c
Department of CSE
Daffodil International University

Introduction: **This project is about the mp3 music player application development using Android. The biggest difference between the music player and existing applications is that it is completely free for users to use. It will integrate the advantages of existing music players on the market, as far as possible to mining out the existing music players' function, and then do the filtering in order to eliminate function that not practical or low cost-effective**

Scope of the project:

To create a simple mp3 player with playlist that has functionalities like Play, Pause, Stop, Next, Preview, Shuffle and mute.

Aim and objective of this project :

The aim of this assignment is to produce a suitable design for a software application,
followed by an implementation in java using Qt library and a documentation of such
application.
Objective is to design a simple mp3 player with user-friendly GUI and basic and some
advance functionalities of mp3 player.

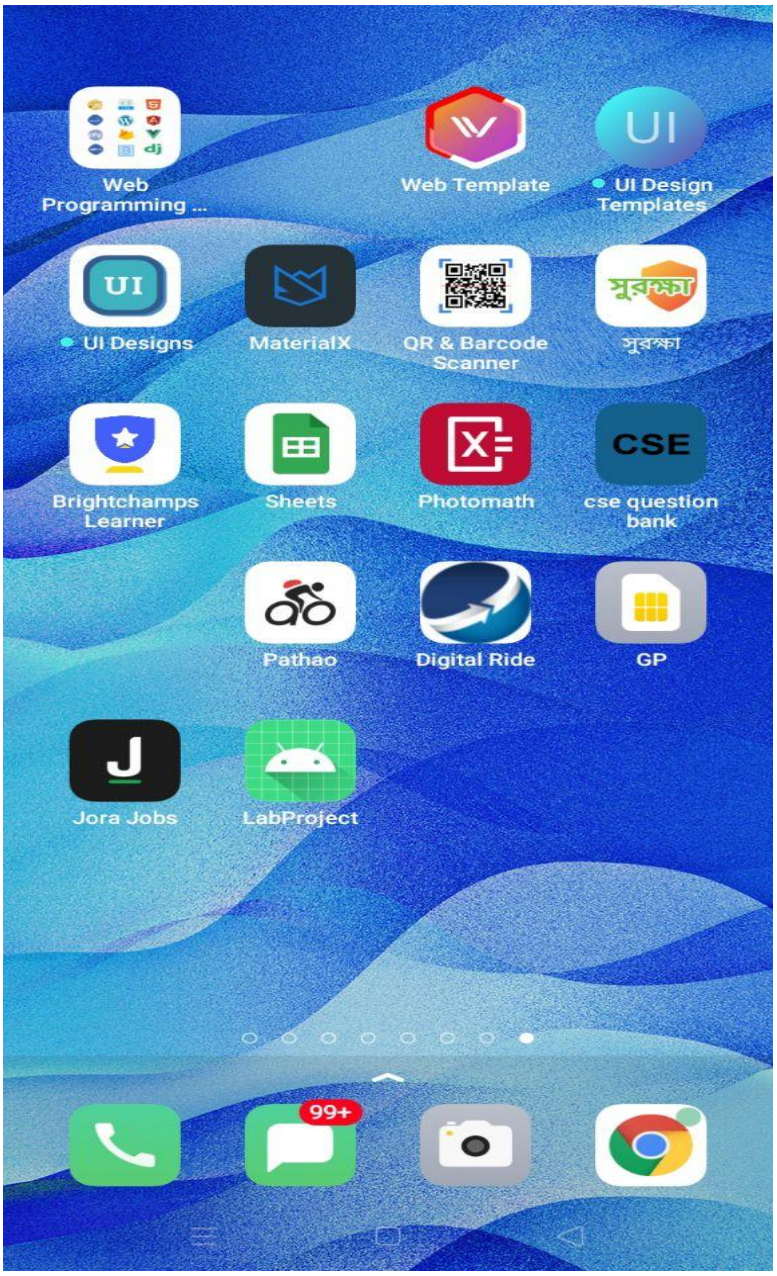
Functionalities :

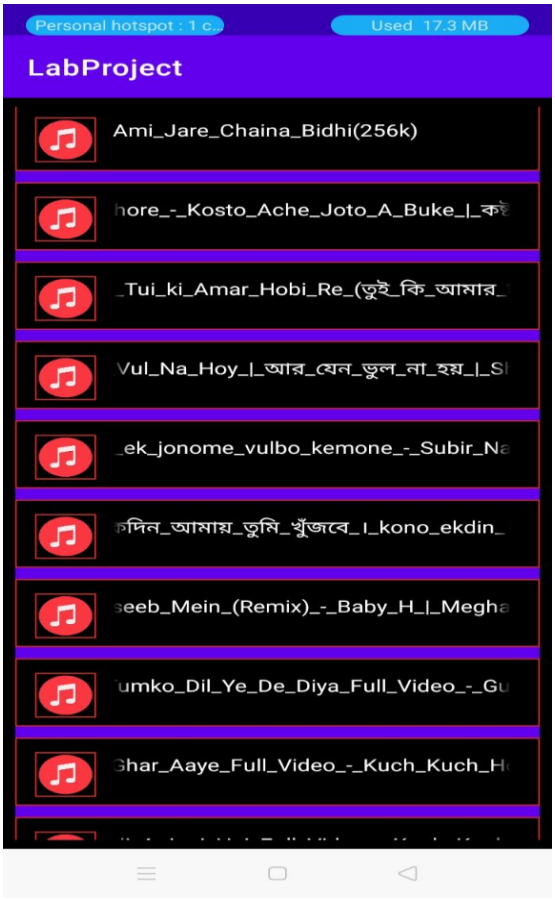
Program functionalities are followed
pause it by either clicking the play button on the tool bar or clicking the GUI button placed in bottom of the window or pressing Space button. And the current status initialized in a label under volume bar.

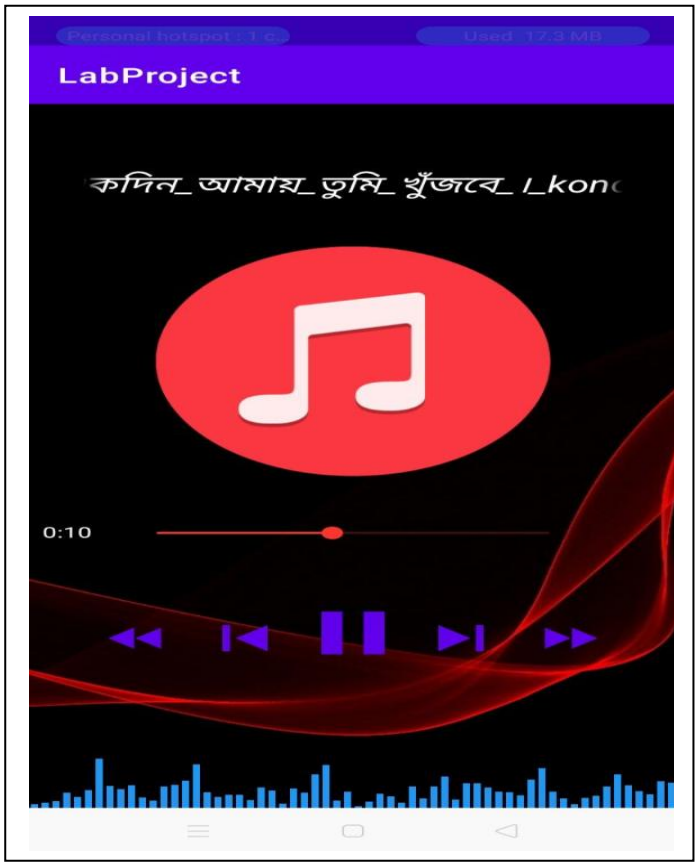
clicking the GUI button placed in bottom of the window or pressing esc button. And the current status initialized in a label under volume bar.

clicking the GUI button placed in bottom of the window or pressing PgUp button. And the current status initialized in a label under volume bar.

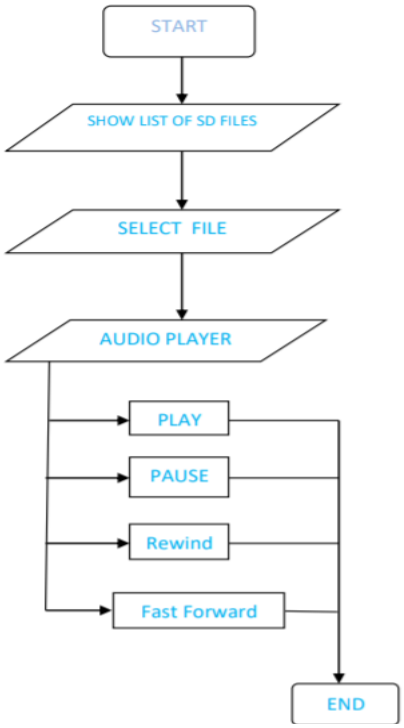
UI (User Interface)




Personal hotspot : 1 c...
Used 17.3 MB
LabProject
Ami_Jare_Chaina_Bidhi(256k)
hore_-_Kosto_Ache_Joto_A_Buke_|_ক
_Tui_ki_Amar_Hobi_Re_(তুই_কি_আমার_
Vul_Na_Hoy_|_আর_যেন_ভুল_না_হয়_|_S
_ek_jonome_vulbo_kemone_-_Subir_Na
দিন_আমায়_তুমি_খুঁজবে_I_kono_ekdin_
eeb_Mein_(Remix)_-_Baby_H_|_Megha
umko_Dil_Ye_De_Diya_Full_Video_-_Gu
Ghar_Aaye_Full_Video_-_Kuch_Kuch_H


LabProject
কদিন_আমায়_তুমি_খুঁজবে_I_kon
0:10

Flow Chart:


START
SHOW LIST OF SD FILES
SELECT FILE
AUDIO PLAYER
PLAY
PAUSE
Rewind
Fast Forward
END

Purpose:

In this music main purpose also show my friend and my family member. The project, particularly its purpose, must therefore be clearly defined before the project developer accepts personal responsibility for its success. The first step, after accepting the mandate, should then be to share the project's purpose with the Project teacher and with all friend.

Benefits:

(1) **Reliability The reliability of the software design must be determined. The reliability of the software system refers to the ability to avoid fault occurred in the process of system running, as well as the ability to remedy troubles once the fault occurs.**

(2) **Reusability Look for commonness of similar codes, and come out new method abstractly and reasonably. Pay attention to the generic design.**

(3) Understandability The understandability of software not only require clear and readable document, but the simplified structure of software itself, which requires the designer possess keen insight and creativity, and know well about the design objects.

(4) Simple program To keep the program simple and clear, good programmers can use simple program to solve complex problems.

(5) Testability Testability means that the created system has a proper data collection to conduct a comprehensive test of the entire system.

(6) The Open-Closed Principal Module is extensible but cannot be modified. That is to say, extension is open to the existing code in order to adapt to the new requirements. While modify is closed to the categories. Once the design is completed, the categories cannot be modified.

Coding:

PlayerActivity.java:

```
package com.example.labproject;

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

import androidx.appcompat.app.AppCompatActivity;

import com.gauravk.audiovisualizer.visualizer.BarVisualizer;

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

public class PlayerActivity extends AppCompatActivity {

    Button btnplay,btnnext,btnprev,btnff,btnfr;
    TextView txtsname,txtstart,txtsstop;
    SeekBar seekmusic;
    BarVisualizer visualizer;
    //ImageView imageView;

    String sname;
    public static final String EXTRA_NAME="song_name";
    static MediaPlayer mediaPlayer;
    int position;
    ArrayList<File> mySongs;
    Thread updateseekbar;

//
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_player);




       btnplay=(Button) findViewById(R.id.btnprev);
       btnnext= (Button) findViewById(R.id.btnnext);
       btnplay = (Button) findViewById(R.id.playbtn);
       btnff= (Button) findViewById(R.id.btnff);
       btnfr= (Button) findViewById(R.id.btnfr);
       txtsname=findViewById(R.id.txtsn);
       txtsstop=findViewById(R.id.txtsstop);
       seekmusic=findViewById(R.id.seekbar);
       visualizer=findViewById(R.id.blast);
       //imageView=findViewById(R.id.imageView);
```

```
    if(mediaPlayer != null){
        mediaPlayer.stop();
        mediaPlayer.release();
    }

    Intent i=getIntent();
    Bundle bundle=i.getExtras();

    mySongs=(ArrayList) bundle.getParcelableArrayList("songs");
    String songnameZ=i.getStringExtra("songname");
    position=bundle.getInt("pos",0);
    txtsname.setSelected(true);
     Uri uri=Uri.parse(mySongs.get(position).toString());
     sname=mySongs.get(position).getName();
     txtsname.setText(sname);

      mediaPlayer=MediaPlayer.create(getApplicationContext(),uri);
      mediaPlayer.start();
      updateseekbar = new Thread()
      {
          public void run(){
              int totalDuration=mediaPlayer.getDuration();
              int currentposition = 0;
              while (currentposition<totalDuration)
              {
                  try {
                      sleep(500);
                      currentposition=mediaPlayer.getCurrentPosition();
                      seekmusic.setProgress(currentposition);
                  }
                  catch (InterruptedException | IllegalStateException e)
                  {
                      e.printStackTrace();
                  }
              }
          }
      };

       seekmusic.setMax(mediaPlayer.getDuration());
       updateseekbar.start();
       seekmusic.getProgressDrawable().setColorFilter(getResources().getColor(R.color.ColorPrimary),
PorterDuff.Mode.MULTIPLY);

seekmusic.getThumb().setColorFilter(getResources().getColor(R.color.ColorPrimary),PorterDuff.Mode.SRC_I
N);

       seekmusic.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());
```

```
        }
    });

//
        btnplay.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(mediaPlayer.isPlaying()){
                btnplay.setBackgroundResource(R.drawable.ic_baseline_play);
                mediaPlayer.pause();
            }
            else{
                btnplay.setBackgroundResource(R.drawable.ic_baseline_pause_24);
                mediaPlayer.start();

            }
        }
    });

    mediaPlayer.setOnCompletionListener(mediaPlayer -> btnnext.performClick());


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

    }
    btnnext.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view) {
            mediaPlayer.stop();
            mediaPlayer.release();
            position=((position+1)%mySongs.size());
            Uri u=Uri.parse(mySongs.get(position).toString());
            mediaPlayer=MediaPlayer.create(getApplicationContext(),u);
            sname=mySongs.get(position).getName();
            txtsname.setText(sname);
            mediaPlayer.start();
            btnplay.setBackgroundResource(R.drawable.ic_baseline_pause_24);
            //StartAnimation(imageView);

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

        btnff.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(mediaPlayer.isPlaying()){
                mediaPlayer.seekTo(mediaPlayer.getCurrentPosition()+10000);
            }
        }
    });
    btnfr.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(mediaPlayer.isPlaying()){
```

```
                mediaPlayer.seekTo(mediaPlayer.getCurrentPosition()-10000);
            }
        }
    });
  }
}
```

activity_player.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    android:background="@drawable/background_img"
    android:orientation="vertical"
    android:weightSum="10"
    tools:context=".PlayerActivity">


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="7"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/txtsn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="20dp"
            android:ellipsize="marquee"
            android:marqueeRepeatLimit="marquee_forever"
            android:padding="10dp"
            android:singleLine="true"
            android:text="Song Name"
            android:textAlignment="center"
            android:textColor="@color/cardview_light_background"
            android:textSize="22sp"
            android:textStyle="italic"></TextView>

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="250dp"
            android:layout_height="250dp"
            android:layout_marginBottom="8dp"
            android:src="@drawable/music"></ImageView>

        <RelativeLayout
            android:layout_width="399dp"
            android:layout_height="60dp">

            <SeekBar
                android:id="@+id/seekbar"
                android:layout_width="250dp"
                android:layout_height="wrap_content"
                android:layout_alignParentBottom="true"
                android:layout_centerInParent="true"
```

```
            android:layout_marginStart="20dp"
            android:layout_marginTop="20dp"
            android:layout_marginEnd="20dp"
            android:layout_marginBottom="20dp"></SeekBar>

        <TextView
            android:id="@+id/tstsstar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="false"
            android:layout_centerInParent="true"
            android:layout_marginLeft="20dp"
            android:layout_toLeftOf="@id/seekbar"
            android:text="0:10"
            android:textColor="#FFF"
            android:textSize="14sp"></TextView>

        <TextView
            android:id="@+id/txtsstop"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="false"
            android:layout_centerInParent="true"
            android:layout_marginRight="20dp"
            android:layout_toRightOf="@id/seekbar"
            android:text="4:10"
            android:textColor="#FFF"
            android:textSize="14sp"></TextView>
    </RelativeLayout>
</LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="3">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/playbtn"
                android:layout_width="70dp"
                android:layout_height="70dp"
                android:layout_centerHorizontal="true"
                android:background="@drawable/ic_baseline_pause_24"

                ></Button>

            <Button
                android:id="@+id/btnnext"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_marginTop="15dp"
                android:layout_toRightOf="@+id/playbtn"
                android:background="@drawable/ic_baseline_skip_next_24"

                ></Button>

            <Button
                android:id="@+id/btnprev"
                android:layout_width="50dp"
```

```
            android:layout_height="50dp"
            android:layout_marginTop="15dp"
            android:layout_toLeftOf="@+id/playbtn"
            android:background="@drawable/kip_previous_24"></Button>

        <Button
            android:id="@+id/btnff"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginTop="20dp"
            android:layout_marginLeft="15dp"
            android:layout_toRightOf="@+id/btnnext"
            android:background="@drawable/fast_forward"

            ></Button>
        <Button
            android:id="@+id/btnfr"
            android:layout_toLeftOf="@+id/btnprev"
            android:layout_marginTop="20dp"
            android:layout_marginRight="15dp"
            android:background="@drawable/ic_baseline_fast_rewind_24"
            android:layout_width="40dp"
            android:layout_height="40dp"
             ></Button>

        <com.gauravk.audiovisualizer.visualizer.BarVisualizer
            android:id="@+id/blast"
            android:layout_width="match_parent"
            android:layout_height="70dp"
            android:layout_alignParentBottom="true"
            android:layout_marginBottom="0dp"
            app:avColor="@color/av_dark_blue"
            app:avDensity="0.5"
            app:avSpeed="normal"
            app:avType="outline"
            app:avWidth="4dp" />

    </RelativeLayout>
</LinearLayout>
</LinearLayout>
```

MainActivity.java

```
package com.example.labproject;

import android.Manifest;
import android.annotation.SuppressLint;
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.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;
```

```
import androidx.appcompat.app.AppCompatActivity;

import com.karumi.dexter.Dexter;
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.single.PermissionListener;

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

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).withPermission(Manifest.permission.READ_EXTERNAL_STORAGE)
                .withListener(new PermissionListener() {
                    @Override
                    public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {
                        displaySongs();
                    }

                    @Override
                    public void onPermissionDenied(PermissionDeniedResponse permissionDeniedResponse) {

                    }

                    @Override
                    public void onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest,
PermissionToken permissionToken) {
                        permissionToken.continuePermissionRequest();
                    }
                }).check();
    }
    public ArrayList<File> findSong (File file){
        ArrayList<File> arrayList = new ArrayList<>();
        File[] files= file.listFiles();
        for (File singlefile : files){
            if(singlefile.isDirectory() &&! singlefile.isHidden())
            {
                arrayList.addAll(findSong(singlefile));
            }
            else
                {
                if (singlefile.getName().endsWith(".mp3") || singlefile.getName().endsWith(".wav"))
                {
                    arrayList.add(singlefile);
```

```
            }
        }

    }
    return arrayList;
}
void displaySongs(){
    final ArrayList<File> mySongs =findSong(Environment.getExternalStorageDirectory());
        items = new String[mySongs.size()];
    for (int i=0;i<mySongs.size();i++){
        items[i] = mySongs.get(i).getName().toString().replace(".mp3", "").replace(".wav", "");


    }

    /*ArrayAdapter<String> myAdapter =new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,items);
    listview.setAdapter(myAdapter);*/
    customAdapter customAdapter=new customAdapter();
    listview.setAdapter(customAdapter);

    listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            String songName = (String) listview.getItemAtPosition(i);
            startActivity(new Intent(getApplicationContext(), PlayerActivity.class).putExtra("songs",
mySongs).putExtra("songname", songName).putExtra("pos", i));
        }
    });
}


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) {
        @SuppressLint("ViewHolder")
        View myView=getLayoutInflater().inflate(R.layout.list_item,null);
        TextView textsong=myView.findViewById(R.id.txtsongname);
        textsong.setSelected(true);
        textsong.setText(items[i]);
```

```
            return myView;
        }
    }
}
```

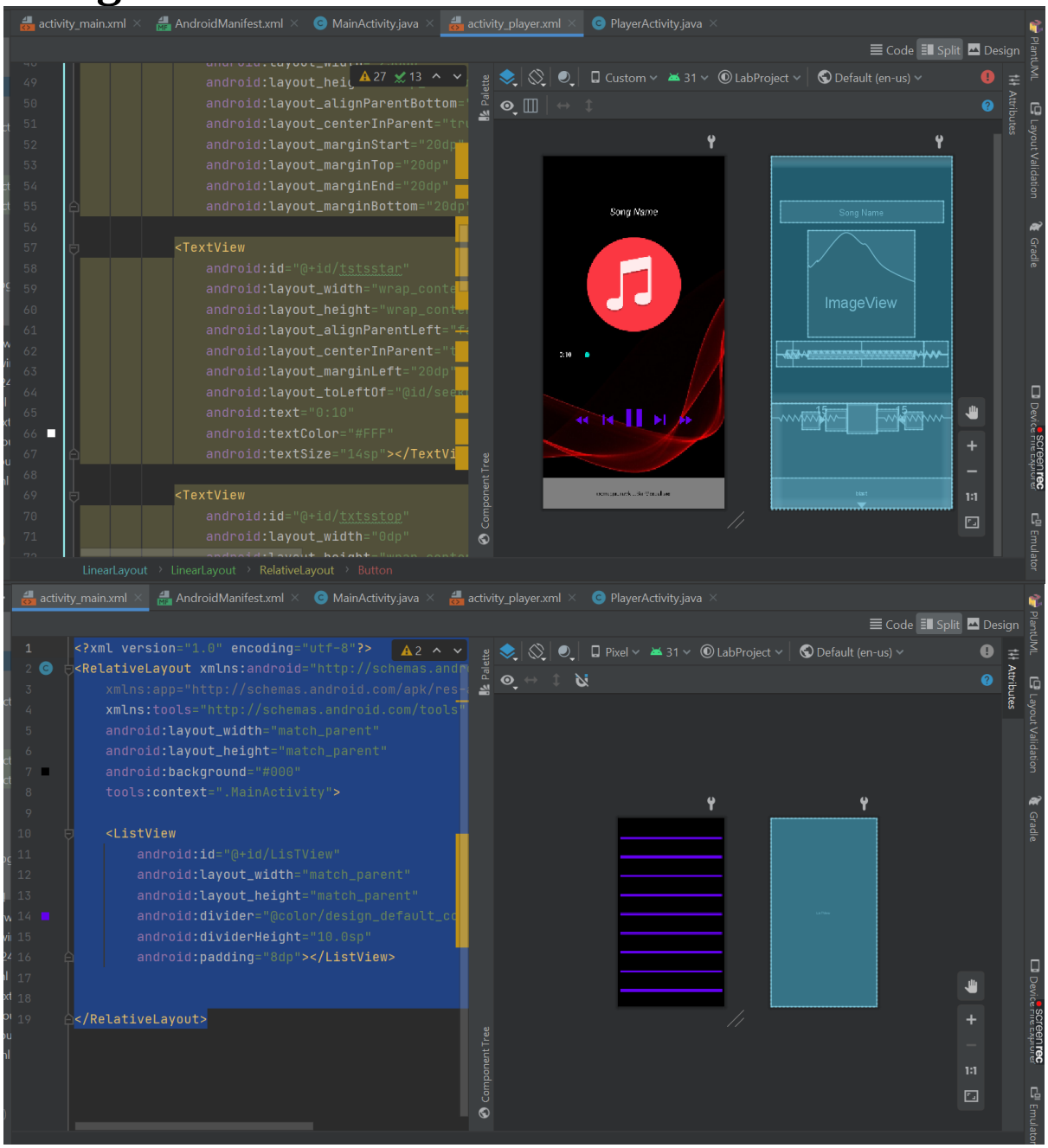
activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="#000"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/LisTView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:divider="@color/design_default_color_primary"
        android:dividerHeight="10.0sp"
        android:padding="8dp"></ListView>


</RelativeLayout>
```

AndroidMainfest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.labproject">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"

        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LabProject">
        <activity
            android:name=".PlayerActivity" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Design and code Scan short:



Conclusion: Through the development of music player on Android platform, we get a clear understanding of overall process of the system. The core part of the music player is mainly composed of main interface, file browsing and song listing, Grasping the development of the music player has had the preliminary scale small features. Music player system realized the basic function of player: play, pause, rewind and fastforward a, volume adjustment is performed through the Android System

Itself, play mode, song search, seekbar, This development implicated the popular mobile terminal development technology. This is the combination management of Java language in the open source mobile platform based on Linux system configuration file. The system realized the music player programming. This design of music player based on Android system requires elaborate design of the music player framework, by adopting ANDROID STUDIO 3.1.2 + Java language as technical support of this system, with the Android plug-in tools, and combination of Latest Android SDK version lead to the comprehensive and smoothly design and development of the mobile terminal

“Best of Luck”