**Introduction:**

Music Player is a software that plays audio files encoded in mp3 and other audio formats. This project is to design and implement platform independent music using Java.

We used Android Studio 3.4.1 to develop this standalone application which can be used in devices which have android version 8.1.0 or lower.

The Music Player consists of these files:

* Splash Screen - activity\_splash.xml and splash.java
* List of Songs - activity\_main.xml and MainActivity.java
* Player Control - activity\_main2.xml and Main2Activity.java

1. **Splash Screen:**

Android Splash Screen is the first screen visible to the user when the application's launched. Splash screen is one of the most vital screens in the application since it's the user's first experience with the application.

* 1. **Layout Output: (activity\_splash.xml) design view:**



* 1. **Layout Code: (activity\_splash.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:orientation="vertical"

tools:context=".splash"

android:background="@drawable/gg1"

>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center\_horizontal|center\_vertical"

android:orientation="vertical">

</LinearLayout>

</LinearLayout>

* 1. **Java Code: (splash.java)**

package com.example.musicplayer;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class splash extends AppCompatActivity {

private static int SPLASH\_TIME\_OUT = 4000;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splash);

new Handler().postDelayed(new Runnable(){

@Override

public void run(){

Intent homeIntent = new Intent(splash.this,MainActivity.class);

startActivity(homeIntent) ;

finish(); } },SPLASH\_TIME\_OUT); }}

1. **List Screen:**

A List of songs is displayed and the user and select and play any particular song.

* 1. **Layout Output: (activity\_main.xml) Design View:**

****

* 1. **Layout Code: (activity\_main.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"

tools:context=".MainActivity"

android:orientation="vertical">

<ListView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#d8bfd8"

android:id="@+id/Lv1"

android:padding="10dp" >

</ListView>

</LinearLayout>

* 1. **Java Code: (MainActivity.java)**

package com.example.musicplayer;

import android.Manifest;

import android.content.Intent;

import android.os.Environment;

import android.support.constraint.solver.ArrayLinkedVariables;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

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.io.FileDescriptor;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

ListView lv1;

String[] items;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

lv1 = (ListView)findViewById(R.id.Lv1);

runtimepermission(); }

public void runtimepermission()

{ Dexter.withActivity(this).withPermission(Manifest.permission.READ\_EXTERNAL\_STORAGE).withListener(new PermissionListener() {

@Override

public void onPermissionGranted(PermissionGrantedResponse response) {

display();

}

@Override

public void onPermissionDenied(PermissionDeniedResponse response) {

} @Override

public void onPermissionRationaleShouldBeShown(PermissionRequest permission, PermissionToken token) { token.continuePermissionRequest(); } }).check(); }

public ArrayList<File>findsong(File file) { ArrayList<File>arrayList = new ArrayList<File>();

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 display (){

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);

lv1.setAdapter(myadapter);

lv1.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

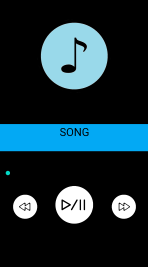
String songName = lv1.getItemAtPosition(i).toString();

startActivity(new Intent(getApplicationContext(),Main2Activity.class).putExtra("songs",mysongs).putExtra("songname",songName).putExtra("pos",i) );}});}}

1. **Player Control:**

Player control is the screen which controls the audio player. It consists of a pause/play button, a forward button, a rewind button, a seek bar and image view.\

* 1. **Layout Output: (activity\_main2.xml) Design View:**

****

* 1. **Layout Code:( activity\_main2.xml)**

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

<android.support.constraint.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"

tools:context=".Main2Activity"

android:background="#000000">

<ImageView

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_alignParentStart="true"

android:layout\_alignParentLeft="true"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="104dp"

android:layout\_marginLeft="104dp"

android:layout\_marginTop="50dp"

android:layout\_marginEnd="8dp"

android:layout\_marginRight="8dp"

android:layout\_marginBottom="8dp"

android:background="@drawable/music"

app:layout\_constraintBottom\_toTopOf="@+id/textView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.01"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.037" />

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="74dp"

android:text="SONG"

android:textColor="#000000"

android:textSize="30dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.515"

android:background="#03A9F4" />

<SeekBar

android:id="@+id/sk"

android:layout\_width="fill\_parent"

android:layout\_height="30dp"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginRight="8dp"

android:layout\_marginBottom="8dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.66" />

<Button

android:id="@+id/play"

android:layout\_width="110dp"

android:layout\_height="110dp"

android:layout\_centerHorizontal="true"

android:background="@drawable/play2"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.811" />

<Button

android:id="@+id/forward"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_marginStart="8dp"

android:layout\_marginLeft="8dp"

android:layout\_marginRight="36dp"

android:layout\_marginBottom="8dp"

android:background="@drawable/ff"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.81" />

<Button

android:id="@+id/backward"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_marginStart="36dp"

android:layout\_marginLeft="36dp"

android:layout\_marginBottom="8dp"

android:background="@drawable/bb"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.155"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.81" /></android.support.constraint.ConstraintLayout>

* 1. **Java Code: (Main2Activity.java)**

package com.example.musicplayer;

import android.content.Intent;

import android.media.MediaPlayer;

import android.net.Uri;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.SeekBar;

import android.widget.TextView;

import java.io.File;

import java.util.ArrayList;

public class Main2Activity extends AppCompatActivity implements View.OnClickListener { Button play,forward,backward;

TextView songName;

static MediaPlayer mp;

SeekBar s;

int position;

ArrayList<File>mySongs;

Thread updateseekbar;

Runnable runnable;

Handler handler = new Handler();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main2);

play = (Button) findViewById(R.id.play);

forward = (Button) findViewById(R.id.forward);

backward = (Button) findViewById(R.id.backward);

s = (SeekBar) findViewById(R.id.sk);

songName = (TextView)findViewById(R.id.textView);

updateseekbar = new Thread()

{

@Override

public void run(){

int totalDuration = mp.getDuration();

int currentpos = 0;

while(currentpos<totalDuration)

{

try {

sleep(500);

currentpos = mp.getCurrentPosition();

}

catch (InterruptedException e){

e.printStackTrace();

} } };

if(mp!=null)

{

mp.stop();

mp.release();

}

Intent i = getIntent();

Bundle bundle = i.getExtras();

mySongs = (ArrayList)bundle.getParcelableArrayList("songs");

String s1 = mySongs.get(position).getName().toString();

String se = i.getStringExtra("songname");

songName.setText(se);

songName.setSelected(true);

position = bundle.getInt("pos",0);

Uri uri = Uri.parse(mySongs.get(position).toString());

mp = MediaPlayer.create(getApplicationContext(),uri);

mp.start();

s.setMax(mp.getDuration());

forward.setOnClickListener(this);

backward.setOnClickListener(this);

play.setOnClickListener(this);

mp.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

s.setMax(mp.getDuration());

mp.start();

changeSeekbar();

} });

s.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

@Override

public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser){

if(fromUser){

mp.seekTo(progress);

} }

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override

public void onStopTrackingTouch(SeekBar seekBar) {

mp.seekTo(seekBar.getProgress());

}});}

private void changeSeekbar() {

s.setProgress(mp.getCurrentPosition());

if(mp.isPlaying()){

runnable = new Runnable() {

@Override

public void run() {

changeSeekbar(); } };

handler.postDelayed(runnable,1000); }

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.play:

if(mp.isPlaying()) {

mp.pause();

}

else

{

mp.start();

changeSeekbar();

}

break;

case R.id.forward:

{

mp.seekTo(mp.getCurrentPosition()+5000);

break;

}

case R.id.backward:

{

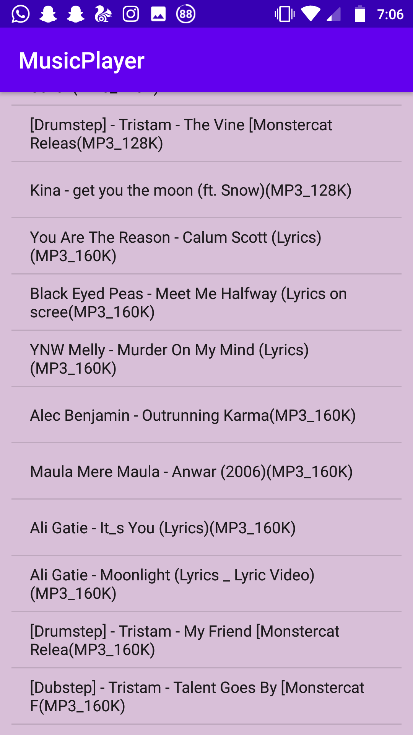
mp.seekTo(mp.getCurrentPosition()-5000);

break;

}

}}}

**OUTPUT:**

**** **** 

Thus we have successfully developed a simple music player application.

**References:**

* **Wikipedia.com**
* **Github.com**
* **GeeksforGeeks.com**