INDUSTRIAL TRAINING REPORT

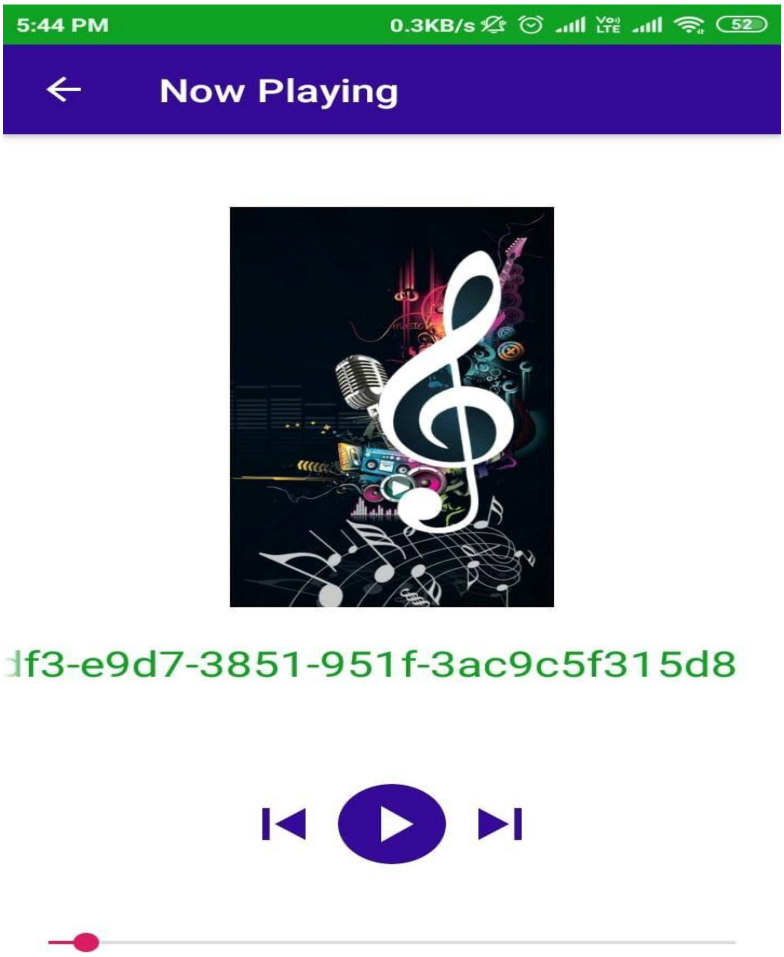
1.1 First Project:

1.1.1 The project

MUSIC PLAYER APPLICATION

For my first application, I was not really comfortable with android, so in this application there are some different screen, and so different activities, but it was a good training to try how to communicate between the activities, how to display elements (TextView, Buttons, Radio buttons, Radio group, Image view, Plain text view, Web view ...) on the screen and how to interact with them.

- A basic music player application for Android.
- This app presents a list of songs on the user device, so that the user can select songs to play.
- The app also presents controls for interacting with playback such as play, pause, next and previous, and will continue playing when the user moves away from the app.



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:padding="10dp"
    tools:context="com.example.musicplayerapp.MainActivity">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/mySongListView"
        android:background="@drawable/image8">
    </ListView>

</RelativeLayout>
```

MainActivity.java

```
package com.example.musicplayerapp;

import android.Manifest;
import android.content.Intent;
import android.os.Environment;
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.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 = (ListView) findViewById(R.id.mySongListView);
        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[] 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);
        listView.setAdapter(myAdapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int i, long l) {

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

                startActivity(new
Intent(getApplicationContext(),PlayerActivity.class).putExtra("songs",mySongs).putExtra("
songname",songName).putExtra("pos",i));
            }
        });
    }
}
```

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:orientation="vertical"
    android:weightSum="10"
```

```
tools:context="com.example.musicplayerapp.PlayerActivity">

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

    <ImageView
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:src="@drawable/music_note" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Song Name"
        android:textAlignment="center"
        android:textSize="22sp"
        android:layout_marginTop="20dp"
        android:id="@+id/songLabel"
        android:singleLine="true"
        android:marqueeRepeatLimit="marquee_forever"
        android:ellipsize="marquee"
        android:scrollHorizontally="true"
        android:textColor="@color/colorPrimaryDark"/>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="3"
```

```
android:orientation="horizontal">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="5dp">

    <SeekBar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/seekBar"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="40dp"/>

    <Button
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_centerHorizontal="true"
        android:background="@drawable/icon_pause"
        android:id="@+id/pause"
        android:layout_marginTop="5dp"/>

    <Button
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:id="@+id/next"
        android:layout_toRightOf="@id/pause"
        android:layout_marginTop="15dp"
        android:background="@drawable/icon_next"/>

    <Button
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_marginTop="15dp"
```

```
            android:layout_toLeftOf="@id/pause"
            android:background="@drawable/icon_previous"
            android:id="@+id/previous"/>


        </RelativeLayout>

    </LinearLayout>

</LinearLayout>
```

PlayerActivity.java

```
package com.example.musicplayerapp;

import android.content.Intent;
import android.graphics.PorterDuff;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.FileObserver;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MenuItem;
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 PlayerActivity extends AppCompatActivity {

    Button btn_next, btn_previous, btn_pause;
```

```
TextView songTextLabel;
SeekBar songSeekBar;

static MediaPlayer myMediaPlayer;
int position;
String sname;

ArrayList<File> mySongs;
Thread updateseekBar;

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

    btn_next = (Button) findViewById(R.id.next);
    btn_previous = (Button) findViewById(R.id.previous);
    btn_pause = (Button) findViewById(R.id.pause);
    songTextLabel = (TextView) findViewById(R.id.songLabel);
    songSeekBar = (SeekBar) findViewById(R.id.seekBar);

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

    updateseekBar = new Thread(){
        @Override
        public void run() {

            int totalDuration = myMediaPlayer.getDuration();
            int currentPosition = 0;

            while (currentPosition<totalDuration){
```

```
            try {
                sleep(500);
                currentPosition = myMediaPlayer.getCurrentPosition();
                songSeekBar.setProgress(currentPosition);
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
};
if (myMediaPlayer != null) {
    myMediaPlayer.stop();
    myMediaPlayer.release();

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

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

sname = mySongs.get(position).getName().toString();

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

songTextLabel.setText(songName);
songTextLabel.setSelected(true);
position = bundle.getInt("pos",0);

Uri u = Uri.parse(mySongs.get(position).toString());
myMediaPlayer = MediaPlayer.create(getApplicationContext(),u);
myMediaPlayer.start();
songSeekBar.setMax(myMediaPlayer.getDuration());
updateseekBar.start();
```

```
        songSeekBar.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) {
                myMediaPlayer.seekTo(seekBar.getProgress());
            }
        });

        btn_pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                songSeekBar.setMax(myMediaPlayer.getDuration());

                if (myMediaPlayer.isPlaying()){
                    btn_pause.setBackgroundResource(R.drawable.icon_play);
                    myMediaPlayer.pause();
                }
                else
                {
                    btn_pause.setBackgroundResource(R.drawable.icon_pause);
                    myMediaPlayer.start();
                }
        }
    });

    btn_next.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            myMediaPlayer.stop();
            myMediaPlayer.release();
            position = ((position+1)%mySongs.size());
            Uri u = Uri.parse(mySongs.get(position).toString());

            myMediaPlayer = MediaPlayer.create(getApplicationContext(),u);
            sname = mySongs.get(position).getName().toString();
            songTextLabel.setText(sname);
            myMediaPlayer.start();
        }
    });

    btn_previous.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            myMediaPlayer.stop();
            myMediaPlayer.release();
            position = ((position-1)<0)?(mySongs.size()-1):(position-1);
            Uri u = Uri.parse(mySongs.get(position).toString());
            myMediaPlayer = MediaPlayer.create(getApplicationContext(),u);

            sname = mySongs.get(position).getName().toString();
            songTextLabel.setText(sname);

            myMediaPlayer.start();
        }
    });
}
```

```
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        if (item.getItemId() == android.R.id.home){
            onBackPressed();
        }
        return super.onOptionsItemSelected(item);
    }
}
```