SUBJECT: MOBILE COMPUTING

PROJECT NAME: MUSIC PLAYER

REPORT SUBMISSION DATE: -06-2023

*Music Player Android App Documentation*

**Introduction**

The Music Player Android app is a user-friendly music streaming application that allows users to play, manage, and organize their music library on their Android devices. The app provides an intuitive interface for users to browse, search, and play their favorite songs, as well as add and delete songs from their library.

**Widget**

* Layout
* Constraint
* Text view
* List view
* Image view
* Seek Bar
* Linear layout

**Working**

The Music Player app works as follows:

1.Song Library: The app scans the user's device for music files and creates a library of songs.

2.Song Playback: Users can select a song from the library and play it using the app's built-in media player.

3.Song Management: Users can add new songs to their library, delete existing songs, and organize their songs into playlists.

4.Alphabetic List: The app displays the song library in an alphabetical list, making it easy for users to find and play their favorite songs.

**Addition of Songs**

To add a new song to the library, users can follow these steps:

1. Browse: Users can browse their device's storage for music files.

2. Select: Users can select the music file they want to add to their library.

3. Add: The app will add the selected song to the library.

**Alphabetic List**

The app displays the song library in an alphabetical list, making it easy for users to find and play their favorite songs. The list is updated in real-time as users add or delete songs from their library.

**Code**

**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="@drawable/bg1"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="1dp"

android:fontFamily="@font/ubuntu"

android:gravity="center"

android:text="iMusic"

android:textAppearance="@style/TextAppearance.AppCompat.Body1"

android:textColor="#263268"

android:textSize="34sp"

android:textStyle="bold"

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

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/nosong"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="80dp"

android:fontFamily="@font/ubuntu"

android:gravity="center"

android:padding="15dp"

android:textColor="#DD2C00"

android:textSize="24sp"

android:textStyle="bold"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/suggest"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="50dp"

android:fontFamily="@font/ubuntu"

android:gravity="center"

android:textColor="#64DD17"

android:textSize="20sp"

android:textStyle="bold"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.5" />

<ListView

android:id="@+id/listView"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_marginStart="1dp"

android:layout\_marginLeft="1dp"

android:layout\_marginEnd="1dp"

android:layout\_marginRight="1dp"

android:layout\_marginBottom="1dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView">

</ListView>

</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:background="@drawable/bg2"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".PlaySong">

<ImageView

android:id="@+id/imageView"

android:layout\_width="231dp"

android:layout\_height="213dp"

android:layout\_marginTop="80dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/music\_icon" />

<TextView

android:id="@+id/textView"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="100dp"

android:ellipsize="marquee"

android:fadingEdge="horizontal"

android:fontFamily="@font/ubuntu"

android:gravity="center"

android:marqueeRepeatLimit="marquee\_forever"

android:scrollHorizontally="true"

android:singleLine="true"

android:textColor="@color/violet"

android:textSize="20sp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView" />

<SeekBar

android:id="@+id/seekBar"

android:layout\_width="360dp"

android:layout\_height="32dp"

android:layout\_marginTop="10dp"

android:thumbTint="@color/violet"

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

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView"

app:layout\_constraintVertical\_bias="0.5" />

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="250dp"

android:layout\_height="80dp"

android:layout\_marginTop="50dp"

android:layout\_marginBottom="50dp"

android:orientation="horizontal"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/seekBar">

<ImageView

android:id="@+id/previous"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1"

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

<ImageView

android:id="@+id/pause"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1"

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

<ImageView

android:id="@+id/next"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1"

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

</LinearLayout>

<TextView

android:id="@+id/currentDuration"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="32dp"

android:layout\_marginLeft="32dp"

android:textColor="#490072"

android:textSize="16sp"

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

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/seekBar"

app:layout\_constraintVertical\_bias="0.0" />

<TextView

android:id="@+id/totalDuration"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="32dp"

android:layout\_marginRight="32dp"

android:textColor="#490072"

android:textSize="16sp"

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

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/seekBar"

app:layout\_constraintVertical\_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

package com.sajjal.imusic;

import android.Manifest;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

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.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 {

@SuppressLint("MissingInflatedId")

@Override

protected void onCreate(Bundle savedInstanceState) {

ListView listView;

TextView nosong,suggest;

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

listView = findViewById(R.id.listView);

nosong = findViewById(R.id.nosong);

suggest = findViewById(R.id.suggest);

Dexter.withContext(this)

.withPermission(Manifest.permission.READ\_EXTERNAL\_STORAGE)

.withListener(new PermissionListener() {

@Override

public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {

ArrayList<File> mySongs = fetchSongs(Environment.getExternalStorageDirectory());

String[] items = new String[mySongs.size()];

for (int i = 0; i < mySongs.size(); i++) {

items[i] = mySongs.get(i).getName().replace(".mp3", ""); }

if(mySongs.size()== 0)

{

nosong.setText("No Songs Found!");

suggest.setText(" Please save some offline songs to your phone to enjoy music.");

}

ArrayAdapter<String> adapter = new ArrayAdapter<String>(MainActivity.this, android.R.layout.simple\_list\_item\_1, items) {

@Override

public View getView ( int position, View convertView, ViewGroup parent){

View view = super.getView(position, convertView, parent);

TextView tv = view.findViewById(android.R.id.text1);

tv.setTextColor(Color.BLACK);

return view;

} };

listView.setAdapter(adapter);

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);

}}); }

@Override

public void onPermissionDenied(PermissionDeniedResponse permissionDeniedResponse) {

}

@Override

public void onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest, PermissionToken permissionToken) {

permissionToken.continuePermissionRequest();

}

})

.check();

}

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;}}

**PlaySong.java**

package com.sajjal.imusic;

import android.content.Intent;

import android.media.MediaPlayer;

import android.net.Uri;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.ImageView;

import android.widget.SeekBar;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.io.File;

import java.util.ArrayList;

import java.util.concurrent.TimeUnit;

public class PlaySong extends AppCompatActivity {

TextView currentDuration;

MediaPlayer mediaPlayer;

ImageView next;

ImageView pause;

int position;

ImageView previous;

TextView totalDuration;

SeekBar seekBar;

ArrayList<File> songs;

boolean stopThread;

String textContent;

TextView textView;

Thread updateSeek;

public void onDestroy() {

super.onDestroy();

stopThread = true;

mediaPlayer.stop();

mediaPlayer.reset();

mediaPlayer.release();

}

public void onCreate(Bundle bundle) {

super.onCreate(bundle);

setContentView(R.layout.activity\_play\_song);

textView = findViewById(R.id.textView);

currentDuration = findViewById(R.id.currentDuration);

totalDuration = findViewById(R.id.totalDuration);

pause = findViewById(R.id.pause);

previous = findViewById(R.id.previous);

next = findViewById(R.id.next);

seekBar = findViewById(R.id.seekBar);

Intent intent = getIntent();

songs = (ArrayList) intent.getExtras().getParcelableArrayList("songList");

String stringExtra = intent.getStringExtra("currentSong");

textContent = stringExtra;

textView.setText(stringExtra);

textView.setSelected(true);

int intExtra = intent.getIntExtra("position", 0);

position = intExtra;

MediaPlayer create = MediaPlayer.create(this, Uri.parse(songs.get(intExtra).toString()));

mediaPlayer = create;

create.start();

seekBar.setMax(this.mediaPlayer.getDuration());

stopThread = false;

seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

public void onStartTrackingTouch(SeekBar seekBar) {

}

public void onProgressChanged(SeekBar seekBar, int i, boolean z) {

Log.d("duration", mediaPlayer.getCurrentPosition() + " and " + mediaPlayer.getDuration());

if (mediaPlayer.getCurrentPosition() >= mediaPlayer.getDuration() - 200) {

next.callOnClick();

}

}

public void onStopTrackingTouch(SeekBar seekBar) {

mediaPlayer.seekTo(seekBar.getProgress());

}

});

updateSeek = new Thread() {

public void run() {

while (!stopThread) {

try {

if (mediaPlayer != null) {

long currentPosition = mediaPlayer.getCurrentPosition();

final String format = String.format("%02d:%02d", Long.valueOf(TimeUnit.MILLISECONDS.toMinutes(currentPosition)), Long.valueOf(TimeUnit.MILLISECONDS.toSeconds(currentPosition) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(currentPosition))));

runOnUiThread(new Runnable() {

public void run() {

currentDuration.setText(format);

}

});

seekBar.setProgress(mediaPlayer.getCurrentPosition());

sleep(200);

Log.d("threadCode", "Updating Success");

}

} catch (Exception e) {

e.printStackTrace();

Log.d("threadCode", "Updating Failed");

}}}};

pause.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

if (mediaPlayer.isPlaying()) {

pause.setImageResource(R.drawable.play);

mediaPlayer.pause();

}

else {

pause.setImageResource(R.drawable.pause);

mediaPlayer.start();

}

}

});

previous.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

mediaPlayer.stop();

mediaPlayer.release();

if (position != 0) {

position--;

} else {

position = songs.size() - 1;

}

Uri parse = Uri.parse(songs.get(position).toString());

mediaPlayer = MediaPlayer.create(getApplicationContext(), parse);

mediaPlayer.start();

pause.setImageResource(R.drawable.pause);

seekBar.setMax(mediaPlayer.getDuration());

textContent = songs.get(position).getName();

textView.setText(textContent);

long duration = mediaPlayer.getDuration();

totalDuration.setText(String.format("%02d:%02d", Long.valueOf(TimeUnit.MILLISECONDS.toMinutes(duration)), Long.valueOf(TimeUnit.MILLISECONDS.toSeconds(duration) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(duration)))));

}

});

next.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

mediaPlayer.stop();

mediaPlayer.release();

if (position != songs.size() - 1) {

position++;

} else {

position = 0;

}

Uri parse = Uri.parse(songs.get(position).toString());

mediaPlayer = MediaPlayer.create(getApplicationContext(), parse);

mediaPlayer.start();

pause.setImageResource(R.drawable.pause);

seekBar.setMax(mediaPlayer.getDuration());

textContent = songs.get(position).getName();

textView.setText(textContent);

long duration = mediaPlayer.getDuration();

totalDuration.setText(String.format("%02d:%02d", Long.valueOf(TimeUnit.MILLISECONDS.toMinutes(duration)), Long.valueOf(TimeUnit.MILLISECONDS.toSeconds(duration) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(duration)))));h

}

});

long duration = mediaPlayer.getDuration();

totalDuration.setText(String.format("%02d:%02d", Long.valueOf(TimeUnit.MILLISECONDS.toMinutes(duration)), Long.valueOf(TimeUnit.MILLISECONDS.toSeconds(duration) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(duration)))));

updateSeek.start();

}}

**Scope**

The Music Player app has the following scope:

1. Music Library Management: The app allows users to manage their music library, including adding and deleting songs.

2. Song Playback: The app provides a built-in media player for playing songs.

3. User Interface: The app provides an intuitive and user-friendly interface for users to interact with their music library.

4. Device Compatibility: The app is compatible with a wide range of Android devices.

**Future Development**

The Music Player app has the potential for future development in the following areas:

1. Cloud Integration: Integrating the app with cloud storage services to allow users to access their music library from multiple devices.

2. Social Sharing: Adding social sharing features to allow users to share their favorite songs and playlists with friends.

3. Personalization: Adding personalization features to allow users to customize their music experience, such as creating custom playlists and radio stations.

4. By providing a user-friendly interface and robust music management features, the Music Player app aims to become the go-to music streaming app for Android users.

**Conclusion**

The conclusion of the music player in Android is that it is a successful project that achieves its purpose of developing a player that can play mainstream file formats. The music player application is designed with a user-friendly interface, allowing users to browse and query storage space, play, pause, and select songs, and even create and store songs in a playlist. Additionally, the application includes features such as dark and light mode, adding songs as favorites, and searching for songs.

The project follows the principles of software design, including reliability, reusability, understandability, simplicity, testability, and the open-closed principle. The system adopts a modularized program design, dividing the system function into function modules, including UI function module design and backstage function module design.

The feasibility analysis verifies that it is feasible to add a music player on