<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:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="16dp"  
 android:paddingLeft="16dp"  
 android:paddingRight="16dp"  
 android:paddingTop="16dp"  
 android:orientation="vertical"  
 tools:context=".MainActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/comfortaa\_bold"  
 android:text="Music Player"  
 android:textSize="35dp"** />  
  
  
 <**LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="0dp"  
 android:layout\_gravity="center"  
 android:layout\_weight="6"  
 android:orientation="horizontal"** >  
  
 <**ListView  
 android:id="@+id/listViewMP3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorLavender"**>  
  
 </**ListView**>  
  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:id="@+id/musicTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/comfortaa\_bold"  
 android:text="Selected Music: "  
 android:textSize="18dp"** />  
  
 <**TextView  
 android:id="@+id/musicTime"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/comfortaa\_bold"  
 android:text="Time: "  
 android:textSize="18dp"** />  
 <**SeekBar  
 android:id="@+id/seekBar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** />  
  
 <**Button  
 android:id="@+id/btnPlay"  
 android:layout\_width="150dp"  
 android:layout\_height="70dp"  
 android:fontFamily="@font/comfortaa\_bold"  
 android:text="Play"** />  
  
 <**Button  
 android:id="@+id/btnStop"  
 android:layout\_width="150dp"  
 android:layout\_height="70dp"  
 android:fontFamily="@font/comfortaa\_bold"  
 android:text="Stop"** />  
  
</**LinearLayout**>

<JAVA>

**package** org.cnit355jieun.mp3playerapp\_jieunyou;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
**import** androidx.core.content.ContextCompat;  
  
**import** android.Manifest;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.pm.PackageManager;  
**import** android.media.MediaPlayer;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.provider.Telephony;  
**import** android.telephony.SmsMessage;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.ProgressBar;  
**import** android.widget.SeekBar;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.File;  
**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.concurrent.TimeUnit;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
  
 TextView **musicTitle**;  
 TextView **musicTime**;  
 SeekBar **MPlayer**;  
  
  
 Button **btnPlay**;  
 Button **btnStop**;  
  
 ArrayList<String> **mp3List**;  
 String **selectedMP3**;  
  
 MyReceiver **myReceiver**;  
  
 **int count** =1;  
  
 *//String mp3Path = "/storage/music/";* String **mp3Path** = **"/sdcard/download"**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 setTitle(**"CNIT 355 Jieun's MP3 Player"**);  
  
  
 **btnPlay** = findViewById(R.id.***btnPlay***);  
 **btnStop** = findViewById(R.id.***btnStop***);  
 **MPlayer** = findViewById(R.id.***seekBar***);  
  
  
  
 **if** (ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***READ\_SMS***) != PackageManager.***PERMISSION\_GRANTED***) {  
 **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**, Manifest.permission.***READ\_SMS***)) {  
 } **else** {  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***READ\_SMS***},1);  
 }  
 }  
 **if** (ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***RECEIVE\_SMS***) != PackageManager.***PERMISSION\_GRANTED***) {  
 **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**, Manifest.permission.***RECEIVE\_SMS***)) {  
 } **else** {  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***RECEIVE\_SMS***},1);  
 }  
 }  
  
  
  
  
 *//PERMISSIONS* **if** (ContextCompat.*checkSelfPermission*(**this**, Manifest.permission.***READ\_EXTERNAL\_STORAGE***) != PackageManager.***PERMISSION\_GRANTED***) {  
 **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**, Manifest.permission.***READ\_EXTERNAL\_STORAGE***)) {  
  
 } **else** {  
 ActivityCompat.*requestPermissions*(**this**,  
 **new** String[]{Manifest.permission.***READ\_EXTERNAL\_STORAGE***},  
 1);  
 }  
 }  
  
 **mp3List** = **new** ArrayList<String>();  
  
 File[] listFiles = **new** File(**mp3Path**).listFiles();  
 String fileName, extName;  
 **for** (File file : listFiles) {  
 fileName = file.getName();  
 extName = fileName.substring(fileName.length() - 3);  
 **if** (extName.equals((String) **"mp3"**))  
 **mp3List**.add(fileName);  
 }  
  
 ListView listViewMP3 = (ListView) findViewById(R.id.***listViewMP3***);  
  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**,  
 android.R.layout.***simple\_list\_item\_single\_choice***, **mp3List**);  
 listViewMP3.setChoiceMode(ListView.***CHOICE\_MODE\_SINGLE***);  
 listViewMP3.setAdapter(adapter);  
 listViewMP3.setItemChecked(0, **true**);  
  
  
 listViewMP3  
 .setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> arg0, View arg1,  
 **int** position, **long** arg3) {  
 **selectedMP3** = **mp3List**.get(position);  
 *//SET SELECTED MUSIC TEXT WHEN CLICKED ON A MUSIC* **musicTitle** = findViewById(R.id.***musicTitle***);  
 **musicTitle**.setText(**"Selected Music: "** + **selectedMP3**); *// when music is selected, the music title is displayed  
 //seekBar.setMax(player.getDuration());* **btnPlay**.setText(**"Play"**); *//when click other music on a pause button, reset o play button* stopAudio();*// when other music is selected, stop music* **player**.seekTo(0);  
 **MPlayer**.setProgress(0);  
  
 **count** = 1; *//set* }  
 });  
  
  
  
  
  
 **selectedMP3** = **mp3List**.get(0);  
 **musicTitle** = findViewById(R.id.***musicTitle***);  
 **musicTitle**.setText(**"Selected Music: "** + **selectedMP3**);  
  
  
  
 *//PLAY MUSIC BUTTON -> OnClickListener* **btnPlay**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
  
  
 **if** (**count** == 1) {  
  
 *// play the music* playAudio();  
  
 *//thread1.start();  
 // Change the button text to "Pause"* **btnPlay**.setText(**"Pause"**);  
  
  
  
 **count**--;  
 } **else if** (**count** == 0) {  
 *// pause the music when pressed again* pauseAudio();  
  
  
 **btnPlay**.setText(**"Play"**);  
 **count**--;  
 } **else** {  
 *//resume music* resumeAudio();  
 **btnPlay**.setText(**"Pause"**);  
 **count**++;  
 }  
  
  
 *// running time has to change as the music plays* }  
 });  
  
 **btnStop**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 *// stop music  
 //stopAudio();* **player**.stop();  
  
 *// music has to start from the beginning (reset seek bar)* **player**.reset();  
 **MPlayer**.setProgress(0);  
 **btnPlay**.setText(**"Play"**);  
 **count** = 1;  
  
  
 }  
 });  
  
  
  
  
  
 **MPlayer**.setOnSeekBarChangeListener(**new** SeekBar.OnSeekBarChangeListener() {  
 @Override  
 **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {  
 *//if(fromUser)* **if**(**player** == **null**) {  
 seekBar.setProgress(0);  
  
 }**else if** (fromUser){  
 **player**.seekTo(progress); *//Play the music with current location progress  
 //player.seekTo(0);* }  
  
 }  
  
 @Override  
 **public void** onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 **public void** onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 });  
  
  
 }  
 *// END OF ON\_CREATE* MediaPlayer **player**;  
 **int position** =0;  
  
 *//PLAY* **public void** playAudio(){  
  
 **try**{  
 *//closePlayer();* **player** = **new** MediaPlayer();  
 **player**.setDataSource(**mp3Path**+**"/"**+**selectedMP3**);  
 **player**.prepare();  
 **player**.start();  
 **player**.setLooping(**true**);  
 Thread();  
 *//initThread();* } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 *//PAUSE* **public void** pauseAudio(){  
 **if**(**player** != **null**){  
 *//어디까지 재생했는지 할당  
  
  
 //pause* **player**.pause();  
 }  
 }  
  
 *//RESUME* **public void** resumeAudio(){  
 **if**(**player** != **null**){  
 *//재생중인지 아닌지 확인* **if**(!**player**.isPlaying()){  
 *//위치를 찾아서 재생 시작* **player**.start();  
  
  
 }  
 }  
 }  
  
 *//STOP* **public void** stopAudio(){  
 **if**(**player** != **null** && **player**.isPlaying()){  
 *//stop* **player**.stop();  
  
 }  
 **MPlayer**.setProgress(0);  
 }  
  
 *// media resource 해제* **public void** closePlayer(){  
 **if**(**player** != **null**){  
 **player**.release();  
 **player** = **null**;  
 }  
 }  
  
  
 **public void** Thread(){  
 Runnable task = **new** Runnable() {  
 @Override  
 **public void** run() {  
  
 **int** current = 0;  
 **while** (**player** != **null** && **player**.isPlaying()) {  
 **try** {  
  
 **int** total = **player**.getDuration();  
 **MPlayer**.setMax(total);  
 **MPlayer**.setIndeterminate(**false**);  
  
 **while**(**player** != **null** && current <total){  
 Thread.*sleep*(10);*//update once per sec* current = **player**.getCurrentPosition();  
 **MPlayer**.setProgress(current);  
  
  
  
 }  
  
  
  
 } **catch** (Exception e) {  
 **break**;  
 }  
  
  
  
 }  
 }  
 };  
 Thread thread = **new** Thread(task);  
 thread.start();  
 }  
  
 *//****TODO:*** *//* ***Add Broadcast receiver: Pause music when SMS is received. Permission?*** @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 unregisterReceiver(**myReceiver**);  
  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 IntentFilter smsFilter = **new** IntentFilter();  
 smsFilter.addAction(Telephony.Sms.Intents.***SMS\_RECEIVED\_ACTION***);  
 registerReceiver(**myReceiver**,smsFilter);  
  
  
 }  
  
 **public class** MyReceiver **extends** BroadcastReceiver {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 Bundle bundle = intent.getExtras();  
 String str = **""**;  
 **if** (bundle != **null**) {  
 Object[] pdus = (Object[]) bundle.get(**"pdus"**);  
 SmsMessage[] msgs= **new** SmsMessage[pdus.**length**];  
 **for** (**int** i = 0; i < msgs.**length**; i++) {  
 msgs[i] = SmsMessage.*createFromPdu*((**byte**[]) pdus[i]);  
 str += **"From: "** + msgs[i].getOriginatingAddress() + **" Msg: "** +msgs[i].getMessageBody().toString();  
 }  
 Toast.*makeText*(context,str, Toast.***LENGTH\_LONG***).show();  
  
 }  
  
 **player**.stop();  
 }  
 }  
  
}  
  
  
<AndroidManifest.xml>

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="org.cnit355jieun.mp3playerapp\_jieunyou"**>  
  
 <**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />  
 <**uses-permission android:name="android.permission.RECEIVE\_SMS"** />  
 <**uses-permission android:name="android.permission.READ\_SMS"** />  
 <**uses-permission android:name="android.permission.SEND\_SMS"** />  
  
  
 <**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/AppTheme"**>  
 <**receiver  
 android:name=".MyReceiver"  
 android:enabled="true"  
 android:exported="true"**></**receiver**>  
  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
  
 <**meta-data  
 android:name="preloaded\_fonts"  
 android:resource="@array/preloaded\_fonts"** />  
 </**application**>  
  
</**manifest**>

My application meets the project guideline except placing the “time change” when the music plays and the broadcast receiver does not stop the music (even though I do have complete code of broadcast receiver which should work.) Also the stop button does stop the music but does not reset the seekbar.