VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi – 590 018

A

Mini Project Report

On

"Media Player Application"

Submitted in partial fulfillment of Bachelor of Engineering Degree

In

COMPUTER SCIENCE AND ENGINEERING

18CSMP68- Mobile Application Development Laboratory

Submitted by:

SANOBER FATIMA	**1HK20CS142**
SANTOSH KUMAR PAITAL	**1HK20CS143**

Under the guidance of

Prof. D. Humera

Assistant Professor, Department of Computer Science & Engineering

JULY 2023

Department of Computer Science and Engineering

HKBK COLLEGE *of* ENGINEERING

(Approved by AICTE & Affiliated to VTU)

No.22/1,Opp., Manyata Tech Park Rd, Nagavara, Bengaluru, Karnataka 560045

Email: info@hkbk.edu.in URL: www.hkbk.edu.in

HKBK COLLEGE *of* ENGINEERING

Nagawara, Bangalore–560 045

Approved by AICTE & Affiliated to VTU

Department of Computer Science and Engineering

Certificate

Certified that the Mini Project Work entitled **"Media Player Application",** carried out by **SANOBER FATIMA (1HK20CS142) and SANTOSH KUMAR PAITAL (1HK20CS143)** are bonafide students of **HKBK COLLEGE *of* ENGINEERING**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University**, Belgaum, during the year **2022–23**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of **18CSMP68- Mobile Application Development Laboratory** prescribed for the said Degree.

Prof. D.Humera	**Dr. Smitha Kurian**	**Dr. Tabassum Ara**
Guide	HOD - CSE	Principal – HKBKCE

External Viva

Name of the Examiners	Signature with Date
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

We would like to express our regards and acknowledgement to all who helped us in completing this mini project successfully.

First of all we would take this opportunity to express our heartfelt gratitude to the personalities, **Mr. C M Ibrahim**, Chairman, HKBKGI and **Mr. C M Faiz Mohammed**, Director, HKBKGI for providing facilities throughout the course.

We express our sincere gratitude to **Dr. Tabassum Ara**, Principal, HKBKCE for her support towards the attainment of knowledge.

We consider it as a great privilege to convey our sincere regards to **Dr. Smitha Kurian.**, Associate Professor and HOD, Department of CSE, HKBKCE, for her constant encouragement throughout the course of the project.

We would specially like to thank our guide **Prof. D. Humera,** Assistant Professor, Department of CSE, HKBKCE for her vigilant supervision and her constant encouragement. She spent her precious time in reviewing the project work and provided many insightful comments and constructive criticism.

Finally, we thank Almighty, all the staff members of CSE Department, our family members and friends for their constant support and encouragement in carrying out the project work.

SANOBER FATIMA (1HK20CS142)

SANTOSH KUMAR PAITAL (1HK20CS143)

ABSTRACT

The functions of playing music have become essential in one device as a smart phone since the smart phone appeared. It is very convenient, but it contains controversial arguments about sound quality, so many smart phone users use the music player application. By using these music applications, people start to think about the relationship between music playing and sound quality. However, those applications are not perfect, so it is hard to choose a good application. This project is about the music player application development using Android. As newcomers to the scene of android application development, we tried to put our knowledge and understanding to build a simple music player application.

TABLE OF CONTENTS

LIST OF FIGURES:

CHAPTER 1:

INTRODUCTION

INTRODUCTION

Mobile Application

A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available. The term "app", short for "software application", has since become very popular; in 2010, it was listed as "Word of the Year" by the American Dialect Society**.**

What is Android Development?

Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008.The official Android app distribution mechanism to end users is Google Play; it also allows staged gradual app release, as well as distribution of pre-release app versions to testers.

Advantages of Android Development:

- **Open system**: Android is naturally open to more of the inner workings of the system. This accessibility helps developers to create and apply features that would probably be restricted by iOS.
- **Flexibility:** With its open-source software, Android offers developers a low barrier to entry. Developers can utilize various crowdsourcing frameworks and plugins and play around with the features and functionality, which makes the platform more flexible than iOS.
- **Design**: Google’s extensive developer guidelines to design are pretty helpful in building an aesthetically appealing layout and intuitive Android user interface.
- **Release**: In contrast to iOS, publishing apps is easier and quicker on Google Play. Once the Android Package (APK) is uploaded, it takes only a few hours before the app goes live.

Disadvantages of Android Development

- **Fragmentation**: There are a multitude of devices with different resolutions and screen sizes to take care of. This makes the app design and UI development much more challenging.

Android development teams should take this into account if they want to deliver a responsive app design that runs seamlessly across many devices, as well as to assess the impact of launching new features since a malfunction on a device can prevent users from running the app as anticipated. Therefore there are so many apps of poor quality in the Play Store.

- **Testing**: Given the multiple Android devices and versions, QA specialists have to spend more time to thoroughly test apps on all models.
- **Cost**: Due to the fragmentation and a large amount of testing required, it may cost more to develop in this platform. Still, it depends on the app's complexity.

Android Components:

There are four main Android app components: activities, services, content providers, broadcast receivers and manifest. Whenever we create or use any of them, we must include elements in the project manifest.

1. Activities:

An Android activity represents any single screen on the user interface. For example, a simple login screen with username and password is one activity of the application. Applications invoke activities based on the workflow.

2. Services:

A service in Android is a background process. Services are typically used for processes that are ongoing or that take a significant period of time.

3. Content Providers

Content providers do the job of supplying data to other applications based on specific requests. Content providers are one of the best ways for cross-application data sharing.

4. Broadcast receivers

Android applications continuously broadcast messages so that other applications know about the event and could possibly trigger some action.

5. Manifest

This is an XML file that contains all configuration parameters of an Android application. In Android projects, a manifest file is included with the project's software and resources when the project is built. This file tells the Android system how to install and use the software in the archive that contains the built project. The manifest file is in XML, and the ADT plug-in provides a specialized XML editor to edit the manifest.

CHAPTER 2:
MEDIA PLAYER APPLICATION

MEDIA PLAYER APPLICATION

INTRODUCTION

Android is open-source code mobile phone operating system that comes out by Google. Music player in this project is application software based on Google Android. Music is one of the best ways to relieve pressure in stressful modern society life. The purpose of this project is to develop a player which can play the mainstream file format. To browse and query the storage space as well as operation of playing can be realised. Meanwhile, this software can play, pause, and select songs with latest button and next button according to sets requirement as well as set up songs.

PROBLEM STATEMENT

Design an Android app that can play music/songs from the local storage and the external using appropriate API, which seamlessly stream uninterrupted music/song. The app needs to be designed in a way so that it delivers audio files which supporting format of mp3/mpv and many other

OBJECTIVE OF THE PROJECT

- List the different available audio files
- Play the selected Audio
- Options for Play, Pause, Next, Previous
- Scroll through the audio file timeline to play required parts

CHAPTER 3:

REQUIREMENTS AND IMPLEMENTATON

REQUIREMENTS & IMPLEMENTATION

Software Requirements:

- Operating System : Windows 8/10/11, Linux, MacOS(10.14 Mojave or later), ChromeOS
- IDE: Android Studio
- Android Emulator

Hardware Requirements:

- 2nd Generation Intel CPU or newer, AMD CPU with support for Windows Hypervisor
- RAM size of 8GB
- Free Storage of 8GB minimum (recommended 30gb and SSD)
- GPU (optional)
- Android Device

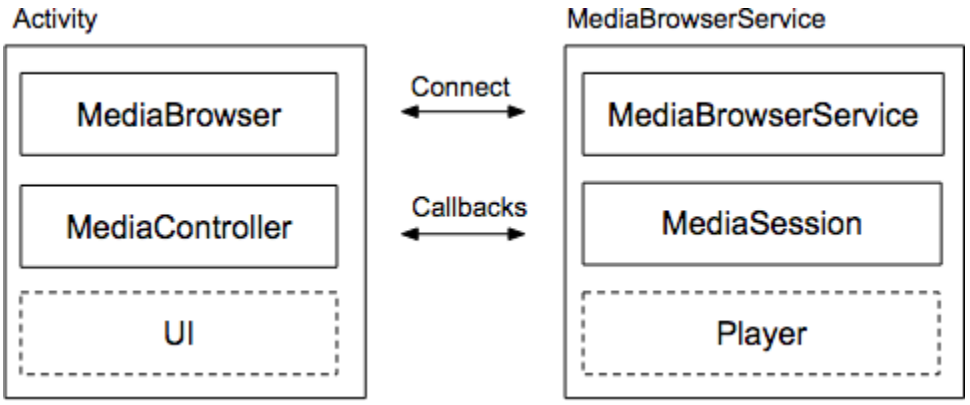
Functional Requirements:

- Now Playing Console
- Ability to play audio files smoothly
- Simple user interface
- Basic Audio files codec support
- Song duration with song completion status bar
- Music Library consisting of all audio files available on phone storage

CHAPTER 4:
SYSTEM DESIGN

SYSTEM DESIGN

System Architecture:



The preferred architecture for an audio app is a client/server design. The client is an Activity in your app that includes a MediaBrowser, media controller, and the UI. The server is a MediaBrowserService containing the player and a media session.

A MediaBrowserService provides two main features:

- When you use a MediaBrowserService, other components and applications with a MediaBrowser can discover your service, create their own media controller, connect to your media session, and control the player. This is how Wear OS and Android Auto Applications gain access to your media application.
- It also provides an optional browsing API. Applications don't have to use this feature. The browsing API lets clients query the service and build out a representation of its content hierarchy, which might represent playlists, a media library, or some other kind of collection.

- **onCreate():**

  It is called when the activity is first created. This is where all the static work is done like creating views, binding data to lists, etc. This method also provides a Bundle containing its previous frozen state, if there was one.

- **onStart():**

  It is invoked when the activity is visible to the user. It is followed by onResume() if the activity is invoked from the background. It is also invoked after onCreate() when the activity is first started.

Android Lifecycle:

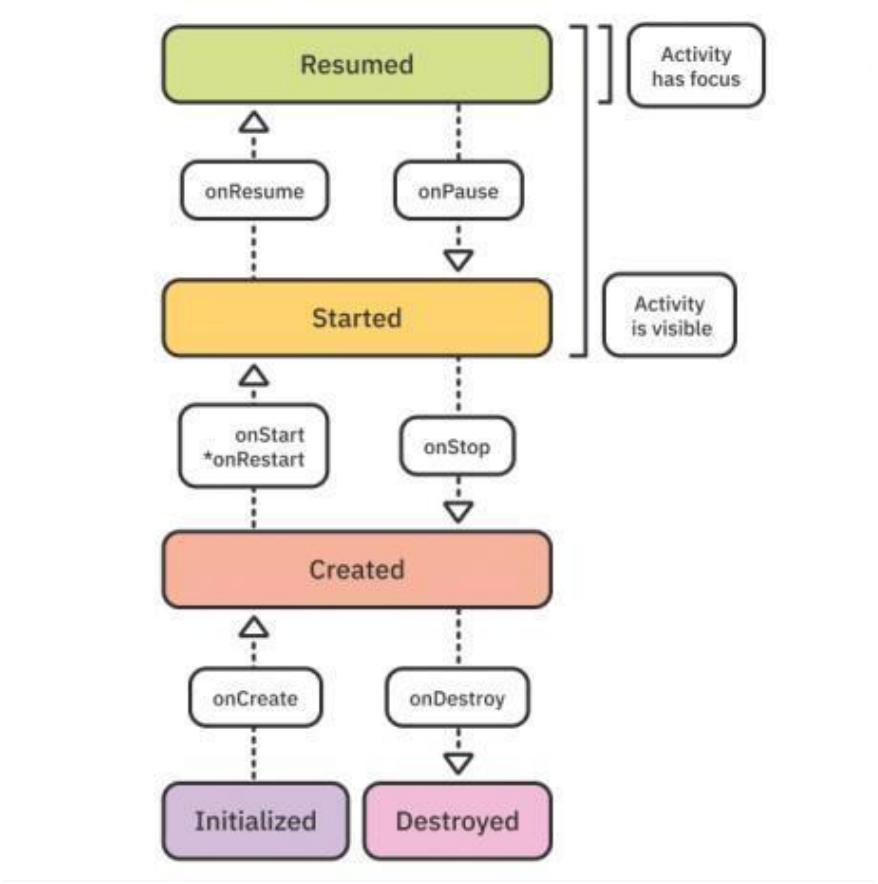


Figure 4.2.1 A simplified illustration of the activity lifecycle.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

Modules Selection

module:

This module accesses the phone storage to retrieve all audio files with mp4, mp3, wav formats. The user can select the song or audio he wants to listen to. The selection module presents each song as a selectable entity.

Player module:

The player module is for the audio to be played. Once a song is pressed on in the selection module, we get transitioned to the player modules where the music controls are present. The screen consists of song/audio name, an image, music scrollbar with time stamps and control buttons, The available control buttons are previous button, pause/play button, forward button.

CHAPTER 5:
IMPLEMENTATION

IMPLEMENTATION

SOURCE CODE:

XML Files:

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools">
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"  android:label="@string/app_name"
    android:requestLegacyExternalStorage="true"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true" android:theme="@style/Theme.SrdMusic" tools:targetApi="31">
    <activity
      android:name=".MusicPlayerActivity" android:exported="false" />
    <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>
```

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"
  tools:context=".MainActivity">
  <TextView
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:id="@+id/songs_text"  android:textColor="@color/black"
    android:text="SONGS"  android:textSize="20dp" android:textStyle="bold" android:padding=
"10dp"
    android:layout_centerHorizontal="true"/>
  <TextView
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:id="@+id/no_songs_text"   android:text="NO SONGS FOUND"
    android:layout_centerInParent="true"  android:visibility="gone"/>
  <androidx.recyclerview.widget.RecyclerView
    android:layout_width="match_parent"  android:layout_height="wrap_content"
    android:id="@+id/recycler_view"  android:layout_below="@id/songs_text"/>
</RelativeLayout>
```

activity_music_player.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="#00BCD4"  tools:context=".MusicPlayerActivity">
<TextView
   android:layout_width="match_parent"  android:layout_height="wrap_content"
   android:id="@+id/song_title"   android:text="Title of the song"
   android:singleLine="true"  android:ellipsize="marquee"
   android:textColor="@color/white" android:textSize="20dp"
   android:layout_margin="20dp"   android:padding="20dp"  />
<ImageView
   android:layout_width="240dp"  android:layout_height="240dp"
   android:id="@+id/music_icon_big" android:layout_centerHorizontal="true"
   android:padding="20dp" android:layout_above="@id/controls"
   android:src="@drawable/music_icon_big"/>
<RelativeLayout
   android:layout_width="match_parent" android:layout_height="wrap_content"
   android:layout_alignParentBottom="true" android:id="@+id/controls"
   android:padding="40dp">
 <SeekBar
    android:layout_width="match_parent"  android:layout_height="wrap_content"
    android:id="@+id/seek_bar"  android:layout_margin="10dp"
    android:backgroundTint="@color/white"/>
  <TextView
     android:layout_width="wrap_content" android:layout_height="wrap_content"
       android:id="@+id/current_time" android:layout_below="@id/seek_bar"
       android:text="0:00"  android:layout_alignParentStart="true"
       android:textColor="@color/white"  android:layout_margin="20dp"/>
   <TextView
       android:layout_width="wrap_content" android:layout_height="wrap_content"
       android:layout_below="@id/seek_bar" android:id="@+id/total_time"
       android:text="0:00"  android:layout_alignParentEnd="true"
       android:textColor="@color/white"  android:layout_margin="20dp"/>
   <RelativeLayout
       android:layout_width="match_parent"  android:layout_height="wrap_content"
       android:layout_below="@id/total_time"  android:padding="20dp">
   <ImageView
           android:layout_width="wrap_content"  android:layout_height="wrap_content"
           android:src="@drawable/baseline_skip_previous_24"
           android:layout_centerVertical="true" android:layout_alignParentStart="true"
           android:id="@+id/previous"/>
    <ImageView
           android:layout_width="wrap_content"  android:layout_height="wrap_content"
           android:src="@drawable/baseline_skip_next_24"
           android:layout_centerVertical="true"    android:layout_alignParentEnd="true"
           android:id="@+id/next"/>
      <ImageView
           android:layout_width="wrap_content" android:layout_height="wrap_content"
           android:src="@drawable/baseline_pause_circle_outline_24"
           android:layout_centerInParent="true" android:id="@+id/pause_play"/>
   </RelativeLayout>  </RelativeLayout> </RelativeLayout>
```

Java Files:

MainActivity.java:

```
package com.example.srdmusic; import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView; import android.content.pm.PackageManager;
import android.os.Bundle; import android.widget.TextView;
import android.widget.Toast; import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.Manifest; import android.database.Cursor;
import android.provider.MediaStore; import android.view.View;
import java.io.File; import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
  RecyclerView recyclerView;  TextView noMusicTextView;
  ArrayList<AudioModel> songsList = new ArrayList<>();
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
    recyclerView = findViewById(R.id.recycler_view);
    noMusicTextView = findViewById(R.id.no_songs_text);
    if(checkPermission() == false){
      requestPermission();  return;      }
    String[] projection = {
        MediaStore.Audio.Media.TITLE,  MediaStore.Audio.Media.DATA,
        MediaStore.Audio.Media.DURATION      };
    String selection = MediaStore.Audio.Media.IS_MUSIC +" != 0";
    Cursor cursor = getContentResolver().query(MediaStore.Audio.Media.
EXTERNAL_CONTENT_URI,projection,selection,null,null);
      recyclerView.setLayoutManager(new LinearLayoutManager(this));
      recyclerView.setAdapter(new
MusicListAdapter(songsList,getApplicationContext()));  }  }
  boolean checkPermission(){
    int result = ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.READ_EXTERNAL_STORAGE);
    if(result == PackageManager.PERMISSION_GRANTED){
      return true; }else{ return false;  }  }
    if(ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,Manifest.perm
ission.READ_EXTERNAL_STORAGE)){
      Toast.makeText(MainActivity.this,"READ PERMISSION IS REQUIRED,PLEASE
ALLOW FROM SETTTINGS",Toast.LENGTH_SHORT).show();
    }else
      ActivityCompat.requestPermissions(MainActivity.this,new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},123);  }
  @Override
  protected void onResume() { super.onResume();
    if(recyclerView!=null){
      recyclerView.setAdapter(new MusicListAdapter(songsList,getApplicationContext()));
    } }}
```

MusicListAdapter.java:

```
package com.example.srdmusic; import android.content.Context;
import android.content.Intent; import android.graphics.Color;
import android.view.LayoutInflater; import android.view.View;
import android.view.ViewGroup; import android.widget.ImageView;
import android.widget.TextView; import java.util.ArrayList;
import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView;
```

```
public class MusicListAdapter extends RecyclerView.Adapter<MusicListAdapter.ViewHolder>{
  ArrayList<AudioModel> songsList;  Context context;
  public MusicListAdapter(ArrayList<AudioModel> songsList, Context context) {
    this.songsList = songsList;  this.context = context;   }
  @Override
  public ViewHolder onCreateViewHolder( ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(context).inflate(R.layout.recycler_item,parent,false);
    return new MusicListAdapter.ViewHolder(view);
      holder.titleTextView.setTextColor(Color.parseColor("#FF0000"));
    }else{
      holder.titleTextView.setTextColor(Color.parseColor("#000000"));       }
    holder.itemView.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        MyMediaPlayer.getInstance().reset();
        MyMediaPlayer.currentIndex = position;
        Intent intent = new Intent(context,MusicPlayerActivity.class);
        intent.putExtra("LIST",songsList);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(intent);    }   });  }
  @Override
  public int getItemCount() { return songsList.size();   }
  public class ViewHolder extends RecyclerView.ViewHolder{
    TextView titleTextView;
    ImageView iconImageView;
    public ViewHolder(View itemView) {  super(itemView);
      titleTextView = itemView.findViewById(R.id.music_title_text);
      iconImageView = itemView.findViewById(R.id.icon_view);
    }}}
```

MusicPlayerActivity.java:

```
package com.example.srdmusic;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; import android.media.MediaPlayer;
import android.os.Handler; import android.util.Log;
import android.widget.ImageView; import android.widget.SeekBar;
import android.widget.TextView; import java.io.IOException;
import java.util.ArrayList; import java.util.concurrent.TimeUnit;
ublic class MusicPlayerActivity extends AppCompatActivity {
  TextView titleTv,currentTimeTv,totalTimeTv;
  SeekBar seekBar;ImageView pausePlay,nextBtn,previousBtn,musicIcon;
  ArrayList<AudioModel> songsList;  AudioModel currentSong;
  MediaPlayer mediaPlayer = MyMediaPlayer.getInstance();int x=0;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_music_player);   titleTv = findViewById(R.id.song_title);
    currentTimeTv = findViewById(R.id.current_time);
    totalTimeTv = findViewById(R.id.total_time); seekBar = findViewById(R.id.seek_bar);
    pausePlay = findViewById(R.id.pause_play); nextBtn = findViewById(R.id.next);
    previousBtn = findViewById(R.id.previous); musicIcon =
findViewById(R.id.music_icon_big);
    titleTv.setSelected(true);
    songsList = (ArrayList<AudioModel>) getIntent().getSerializableExtra("LIST");
```

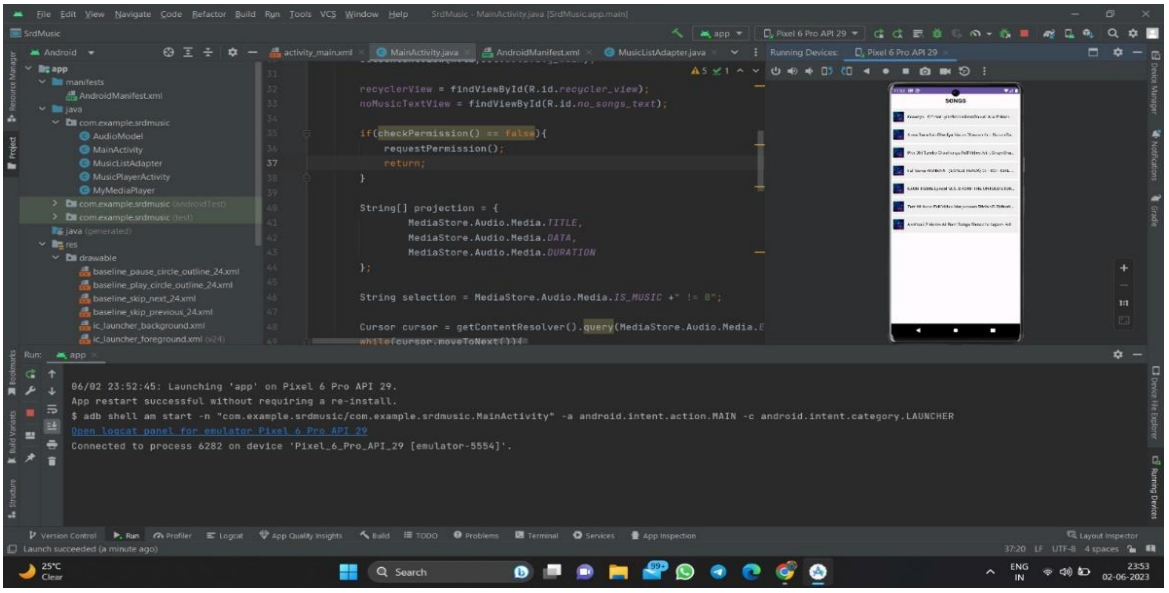
```
        setResourcesWithMusic();
        MusicPlayerActivity.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                if(mediaPlayer!=null){
                    seekBar.setProgress(mediaPlayer.getCurrentPosition());
                    currentTimeTv.setText(convertToMMSS(mediaPlayer.getCurrentPosition()+""));
                    if(mediaPlayer.isPlaying()){
                        pausePlay.setImageResource(R.drawable.baseline_pause_circle_outline_24);
                        musicIcon.setRotation(x++);
                    }else{
                        pausePlay.setImageResource(R.drawable.baseline_play_circle_outline_24);
                        musicIcon.setRotation(0);} }
                new Handler().postDelayed(this,100); } });
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                if(mediaPlayer!=null && fromUser){
                    mediaPlayer.seekTo(progress);  }    }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {  }
            try {
            mediaPlayer.setDataSource(currentSong.getPath());
            mediaPlayer.prepare();  mediaPlayer.start();
            seekBar.setProgress(0); seekBar.setMax(mediaPlayer.getDuration());
        } catch (IOException e) { e.printStackTrace(); }   }
    private void playNextSong(){
        if(MyMediaPlayer.currentIndex== songsList.size()-1)  return;
        MyMediaPlayer.currentIndex +=1;  mediaPlayer.reset();
        setResourcesWithMusic();   }
    private void playPreviousSong(){
        if(MyMediaPlayer.currentIndex== 0)  return;
        MyMediaPlayer.currentIndex -=1;  mediaPlayer.reset();
        setResourcesWithMusic();   } private void pausePlay(){
        if(mediaPlayer.isPlaying())  mediaPlayer.pause();
        else { mediaPlayer.start();  }
    public static String convertToMMSS(String duration){
        Long millis = Long.parseLong(duration);  return String.format("%02d:%02d",
                TimeUnit.MILLISECONDS.toMinutes(millis) % TimeUnit.HOURS.toMinutes(1),
                TimeUnit.MILLISECONDS.toSeconds(millis) % TimeUnit.MINUTES.toSeconds(1));
    }}
```

MyMediaPlayer.java:

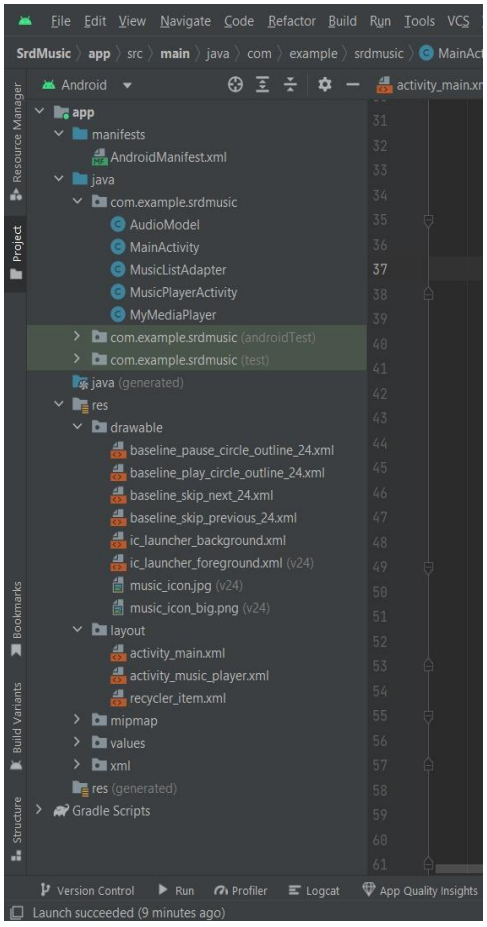
```
package com.example.srdmusic;
import android.media.MediaPlayer;
public class MyMediaPlayer {
    static MediaPlayer instance;
    public static MediaPlayer getInstance(){
        if(instance == null){
            instance = new MediaPlayer();       }
        return instance;   }
    public static int currentIndex = -1;
}
```

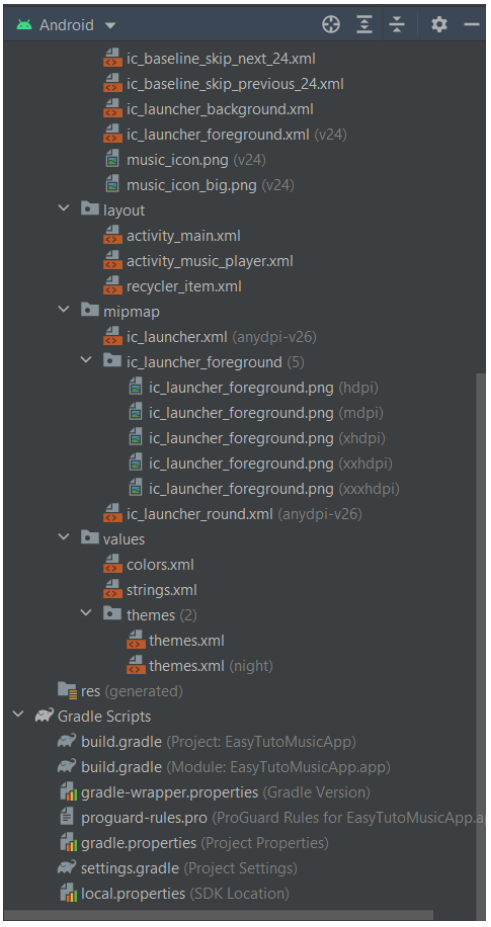
Screenshot:

Project Structure:

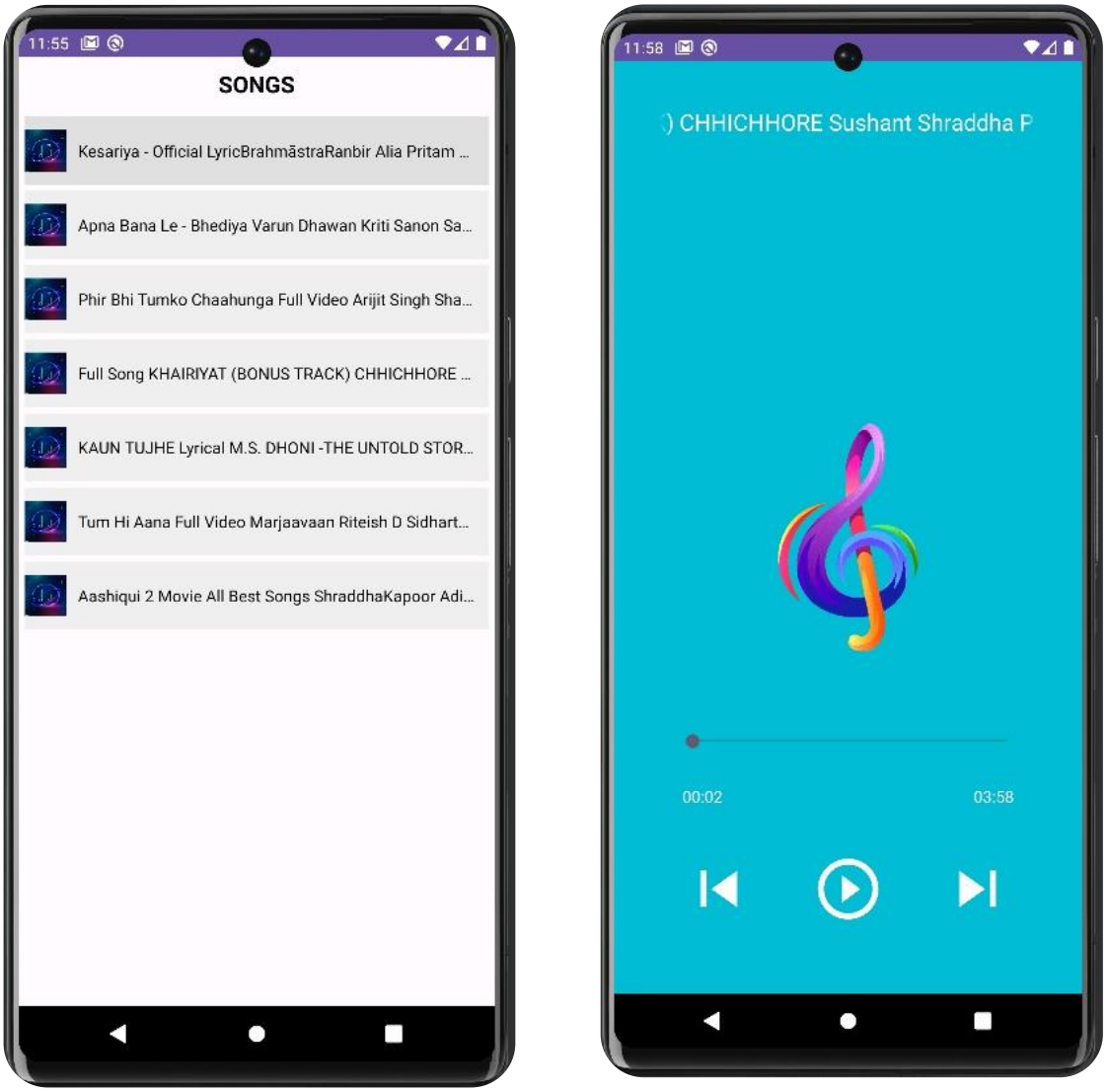


activity_main.xml file is an important component in Android development that defines the layout and appearance of the main user interface (UI) for an Android app. It is typically created as an XML (Extensible Markup Language) file and is associated with the main activity of an app.





The MainActivity.java file is a Java class that serves as the main entry point for an Android app. It is responsible for controlling the behavior and logic of the main activity, which represents the primary user interface for the app.



The output shows the sound track image two seekbars , play, pause and navigate buttons,Along with these functionalities the volume can also be increased and decreased.

Seekbar(1): To show the duration of current song

Navigation button:To navigate through the Playlist

Play-pause button:To play and pause the songs

Seekbar(2):To increase or decrease the volume

Image View :Shows the album picture of current song being played.

Installation Setup:

1. Android Studio Installation:

Step 1: Head over to this link to get the Android Studio executable or zip file.

Step 2: Click on the Download Android Studio Button.

Click on the "I have read and agree with the above terms and conditions" checkbox followed by the download button.Click on the Save file button in the appeared prompt box and the file will start downloading.

Step 3: After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.Click on next. In the next prompt, it'll ask for a path for installation.

Step 4: It will start the installation, and once it is completed, it will be like the image shown below.

Step 5: Once "Finish" is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not. It is better to choose the 'Don't import Settings option'.

Step 6: This will start the Android Studio.It will be finding the available SDK components.

Step 7: After it has found the SDK components, it will redirect to the Welcome dialog box.

Choose Standard and click on Next. Now choose the theme, whether the Light theme or the Dark one. The light one is called the IntelliJ theme whereas the dark theme is called Darcula.

Step 8: Now it is time to download the SDK components.Click on Finish.

Emulator Setup:

An emulator (also known as an AVD) is an Android Virtual Device that looks, acts, walks, and talks (well, maybe not walks and talks) like a physical Android device. AVDs can be configured to run just about any particular version of Android.

Steps to create AVD:

Step 1: Choose Tools→Android→AVD Manager.

The AVD Manager dialog box opens.

Step 2: Click Create a Virtual Device.

Click the Nexus 5 item and click Next.

Step 3: In the System Image dialog box, select the Lollipop x86 item.

Step 4: In the Configure AVD dialog box, use the default AVD name or change it to a haiku.

Step 5: Click the Finish button.

The figure shows the completed AVD Manager dialog box. You should now see your new AVD listed under Your Virtual Devices.

CHAPTER 6:
SYSTEM TESTING

SYSTEM TESTING

6.1 System Architecture:

System Testing is alevel of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system

Testing Hierarchy

1. Unit testing
2. Integration testing

Unit Testing:

Unit testing performed on each module or block of code during development. Unit testing is normally done by the programmer who writes the code.

Unit Test case 1:

Title: user module testing

Step1: create user by entering information in user form and save

Step 2: make sure the newly created user is listed in user list.

Step 3: edit user and make sure all the values are filled up in user form apart from password

Unit Test case 2:

Tittle: Project module testing

Step 1: create project by entering information in project form and save.

Step 2: Make sure the newly created project is listed in project list.

CONCLUSION
REFERENCES ,FUTURE SCOPE

CONCLUSION

The System has been developed for the given condition and is found working effectively. The developed System is Flexible and changes can be made accordingly whenever required. Using the facilities and functionalities of android studio the application has been developed in neat and simple manner, thereby reducing operator's work.

The speed and accuracy is maintained in a proper way. The user friendly nature of this application developed in android studio very easy to work with both the higher management as well as users with little knowledge of Computer. The results obtained were fully satisfactory from the user point of view.

FUTURE SCOPE

We aim to improve the app by performing some fundamental changes. We would like to add a file system to organize the audio files and also sort them based on Artist, Name, Album, Genre. We would also like to add a settings page where user can change app theme and access a equilizer to tune the audio. The current app is also missing a status bar now playing section which we aim to add.

LIMITATION

While the Mp3 Media Player Application developed using Android Studio and Java offers valuable functionality, it also has certain limitations. These limitations include:

1. Limited Audio Format Support: The application primarily focuses on playing MP3 files, which may restrict the compatibility with other audio formats such as WAV, AAC, or FLAC. Enhancements would be required to support a wider range of audio formats.
2. Lack of Streaming Capability: The application does not provide direct streaming capabilities from online music platforms. Integration with popular streaming services would be needed to enable users to access and stream music directly from these platforms.
3. Absence of Advanced Features: The application lacks certain advanced features commonly found in commercial music player applications. These may include features like advanced equalizer settings, crossfade effects, smart playlists, or automatic song recommendations based on user preferences.

BIBLIOGRAPHY:

1. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197

2. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341

3. Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Nerd Ranch Guide”, 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

Websites:

- https://developer.android.com
- GeeksforGeeks | A computer science portal for geeks

HKBK College of Engineering
No.22/1, Opp., Manyata Tech Park Rd, Nagawara, Bengaluru, Karnataka 560045.
Approved by AICTE & Affiliated by VTU

Department of Computer Science & Engineering

DEPARTMENT MISSION AND VISION

VISION

To advance the intellectual capacity of the nation and the international community by imparting knowledge to graduates who are globally recognized as innovators, entrepreneur and competent professionals.

MISSION

M-1. To provide excellent technical knowledge and computing skills to make the graduates globally competitive with professional ethics.

M-2. To involve in research activities and be committed to lifelong learning to make positive contributions to the society. Institute

INSTITUTE MISSION AND VISION

VISION

To empower students through wholesome education and enable the students to develop into highly qualified and trained professionals with ethics and emerge as responsible citizen with broad outlook to build a vibrant nation.

MISSION

M - 1. To achieve academic excellence through in-depth knowledge in science, engineering and technology through dedication to duty, innovation in teaching and faith in human values.

M - 2. To enable our students to develop into outstanding professionals with high ethical standards to face the challenges of the 21st century

M-3. To provide educational opportunities to the deprived and weaker section of the society, to uplift their socio-economic status.

PROGRAM OUTCOMES(PO'S)

PO-1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO-4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend.

PO-11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1. Problem-Solving Skills: An ability to investigate and solve a problem by analysis, interpretation of data, design and implementation through appropriate techniques, tools and skills.

PSO-2. Professional Skills: An ability to apply algorithmic principles, computing skills and computer

science theory in the modelling and design of computer-based systems.

PSO-3. Entrepreneurial Ability: An ability to apply design, development principles and management skills in the construction of software product of varying complexity to become an entrepreneur.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO-1. To provide students with a strong foundation in engineering fundamentals and in the computer science and engineering to work in the global scenario.

PEO-2. To provide sound knowledge of programming and computing techniques and good communication and interpersonal skills so that they will be capable of analyzing, designing and building innovative software systems.

PEO-3. To equip students in the chosen field of engineering and related fields to enable him to work in multidisciplinary teams.

PEO-4. To inculcate in students professional, personal and ethical attitude to relate engineering issues to broader social context and become responsible citizen.

PEO-5. To provide students with an environment for life-long learning which allow them to successfully adapt to the evolving technologies throughout their professional career and face the global challenges.

COURSE OUTCOMES(COs)

CO-1. Create, test and debug Android application by setting up Android development environment.

CO-2. Implement adaptive, responsive user interfaces that work across a wide range of devices.

CO-3. Infer long running tasks and background work in Android applications.

CO-4. Demonstrate methods in storing, sharing and retrieving data in Android applications.

CO-5. Analyse performance of android applications and understand the role of permissions and security.

CO-6. Describe the steps involved in publishing Android application to share with the world.