A Project/Dissertation ETE Report

on

MUSIC PLAYER APPLICATION

Submitted in partial fulfillment of

the requirement for the award of

the degree of

B. Tech in Computer Science & Engineering

Under The Supervision of
Dr. Sudeept Singh Yadav
Associate Professor

Submitted By: BT4211
JASWINDER SINGH- 18SCSE1010377
Kushagra Siddhit- 18SCSE1010429

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA- 2021-22

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Music Player App"** in partial fulfillment of the requirements for the award of the **B.Tech** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of Semester 7, August 2021-December 2021,under the supervision of Dr. Sudeept Singh Yadav (Associate Professor), Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Jaswinder Singh(18SCSE1010377)

Kushagra Siddit(18SCSE1010429)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Jaswinder Singh(18SCSE1010377) and Kushagra Siddhit (18SCSE1010429) has been held on_____and his/her work is recommended for the award of B. Tech CSE.

Signature of Examiner(s)

Signature of Supervisor(s) Signature of Project Coordinator

Signature of Dean

Acknowledgement

We would like to express our greatest appreciation to the all individuals who have helped and supported us throughout the project. We are thankful to our Guide Dr. Sudeept Yadav for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project.

A special acknowledgement goes to our classmates and seniors who helped us in completing the project by exchanging interesting ideas and sharing their experience.

We wish to thank our parents as well for their undivided support and interest which inspired us and encouraged us to go our own way, without which we would have been unable to complete this project.

At the end, we want to thank our friends who displayed appreciation to our work and motivated me to continue my work.

Abstract

Music has been a way of reducing stress and since we all have different emotions, music comes into play kind of styles. For a music program, you need one application for measuring audio and other streaming of songs and many more. Ours the idea is to combine all of these into one song file that can be helpful to music lovers.

This application includes maps data integrated with JK music player. Some of the features of this app include song sync , audio format checking , granting permissions etc. This proposal reduces the use of a lot of applications and will be a complete music app. The functions of playing music has become essential in one device as a smart phone since the smart phone appeared.

As sound quality is the important part of any music player app, so implementing a code that checks the format of files is must.

We are making this application in Android Studio using Java and XML. After the implementations and gradle run, we will get an optimized music player app which reads the audio files from a device. On concluding the above paragraphs, problems which users felt in music player apps today is optimization requirements which can be solved using our JK Music Player Application.

The functions of playing music has 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, many applications are not perfect , so its hard to choose a good application.

Contents

List of Tables

Table for Student Data:

S. No	Name	Enrollment Number	Admission Number	Program / Branch	Sem
1	JASWINDER SINGH	18021011608	18SCSE1010377	B.Tech CSE	7
2	KUSHAGRA SIDDHIT	18021011660	18SCSE1010429	B.Tech CSE	7

Faculty Data:

Guide Name: Dr. Sudeept Singh Yadav

Designation: Associate Professor

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
AVD	Android Virtual Device
SDK	Software Development Kit
Env.	Environment
UI	User Interface
XML	Extensible Markup Language
SCSE	School of Computing Science and Engineering

Chapter -1

Introduction

1.1 Introduction

The application development sector is switching to advancement everyday . Innovative ideas are born each minute to ease people's work; if not big or ground breaking, but constructive and leading towards a better tomorrow. Sound is the intriguing areas of technology which attract to explore more music into their depths.

With the new developments in technology the sophistication level in software has also increased. JK Media player is a software for playing computer files like audio. It generally display some standard media control icons such as stop buttons , play, pause, fast-forward and back-forward, tape recorders and CD players.

In addition, they also come with progress bars (or "playback bars") for locating the current position in the duration of the media file. It based on artists or albums, as well as on the folder structure and also guide in finding all the music files in seconds, and supports you quick search of downloaded music through artist or track name.

It even has search function which searches the song and show the desired result. It contains an interactive and impressive UI which permit user to control respective songs.

1.2 Formulation of Problem

Music player functions are very important on a single device like a smart phone since the advent of the smart phone. However, most applications are not perfect because of audio format issues , so it is difficult to choose a good application Since audio quality is an integral part of any music player application, then using code that checks file format is appropriate.

Code that will monitor the format of audio files and remove corrupted audio tracks to get a better user experience. For this purpose we need to use methodologies i.e.

i. Android Studio: Android Studio is an integrated development platform for Google's Android operating system. It was developed by Jetbrains and google in the year 2013.

ii.XML and Java: The main language used to develop Android Apps Java. Java and XML (Extensible Language Language) are the basic operating requirements for Android Studio. Android apps need to be based on the Android environment.

1.2.1 Tools and Technology used

Tools which are going to use to make our application are

- Android Studio
- Java
- XML
- Some API’s

i. Android Studio: Android Studio is an integrated development platform for Google's Android operating system. It was developed by Jet-brains and google in the year 2013.

ii. XML and Java: The main language used to develop Android Apps Java. Java and XML (Extensible Language Language) are the basic operating requirements for Android Studio. Android apps need/s to be based on the Android environment.

Chapter - 2

Literature Survey

Music and its use for emotion regulation processes, still remains an unanswered question. Many experimental layouts encompassing its daily life use and clinical applications across different cultures and continents have preserved music as a self-regulative tool.

Music intervention and emotion regulation measures were viewed and included only when at least forms of music participation (singing, playing, listening, and engagement) were noted in the study and effects on emotion regulation were directly measured.

The interrelations between the effects of music on emotion regulation and the use of it as a purposeful instrument, e.g. educational or therapeutic functions, yielded limited results, music interventions for specific.

Music has a' regulative capacity of itself', but is confined as valuable instrument for specific emotion regulation interventions.

Emotions can appear in most parts of human-to-human communication and often provide additional information about a message.

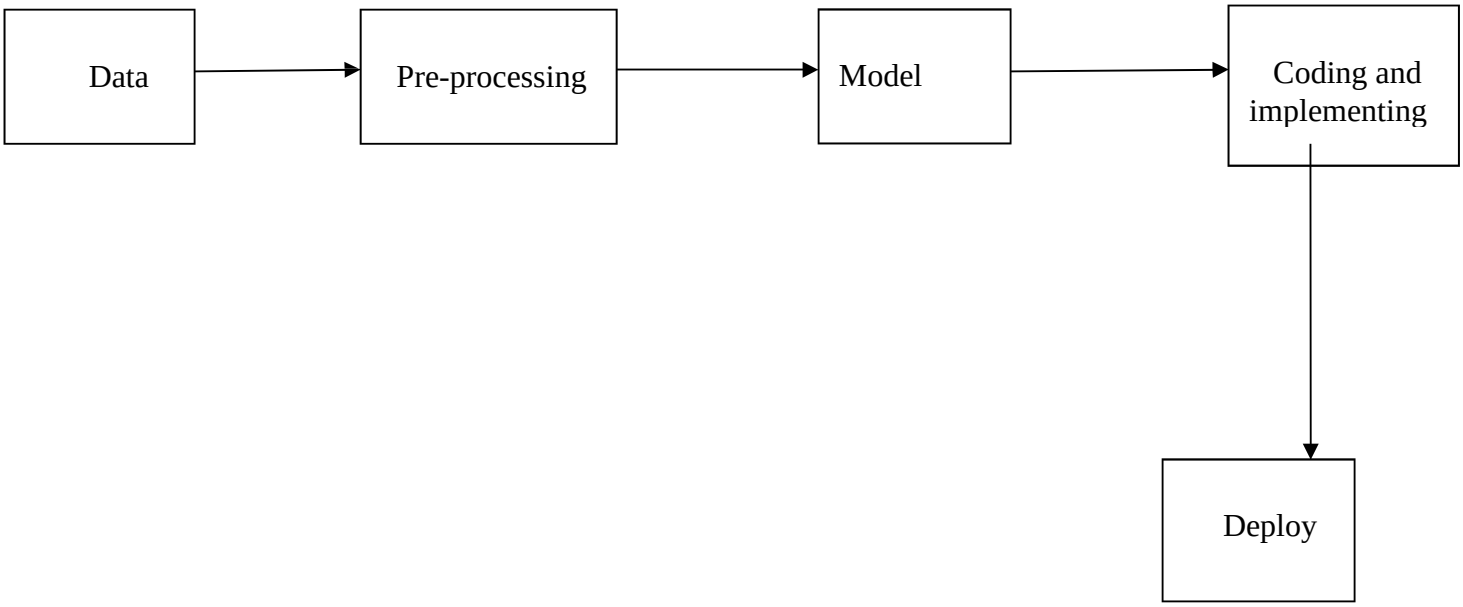
As some emotion expressions are culturally independent-even in a foreign language where we do not understand the meaning of words it is relatively easy for anyone to recognize surprise, scare, anger, etc. in the message.

So, music songs in the recent years have become a popular choice to depict human emotions. Initially, it was a tedious task to label songs based on the emotions they depict from a collection on large database of songs.

But audio and lyrics of songs become ways of extracting the emotions and helped to distinguish the different human emotions.

Also, it is known that individuals perceive emotions within music differently. Knowing the many existing approaches for modelling the ambiguities of musical mood, a complete system would need to incorporate some level of individual profiling to adjust it.

Design



Working Design

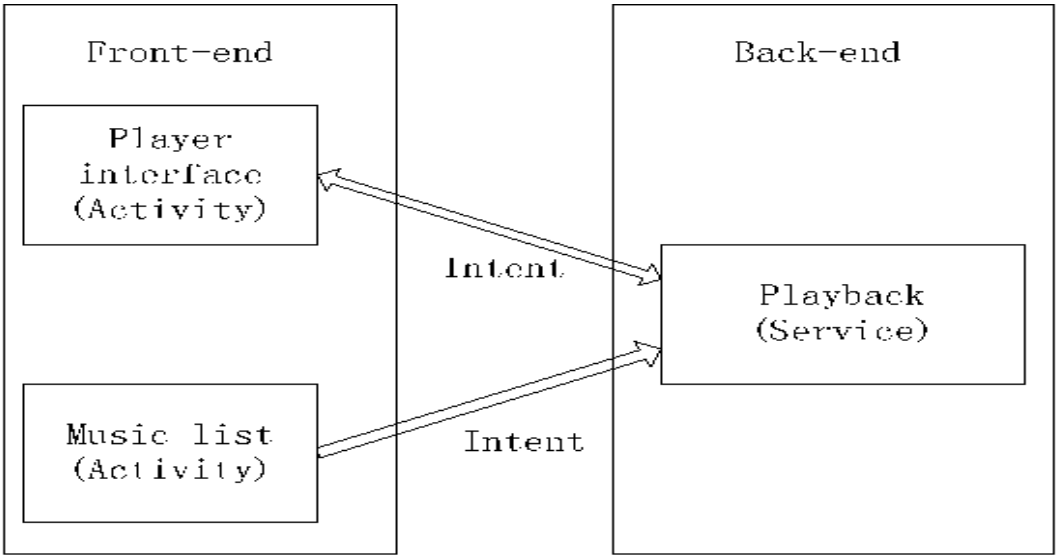
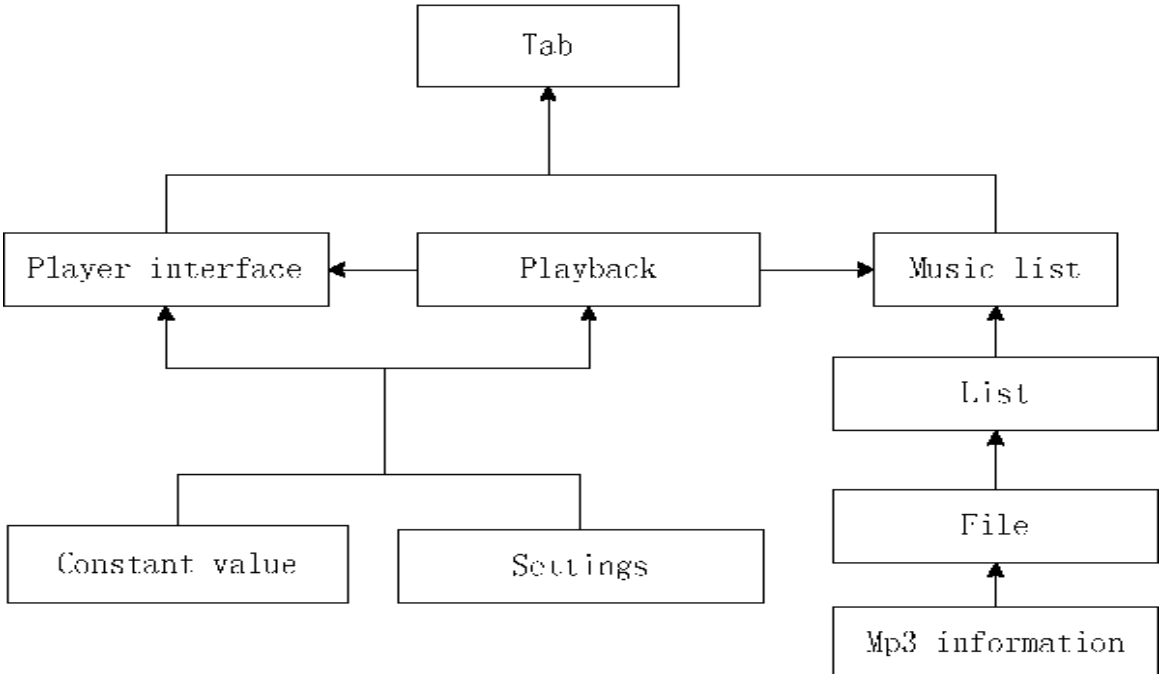


Figure 1. the architecture of music player



Chapter – 3

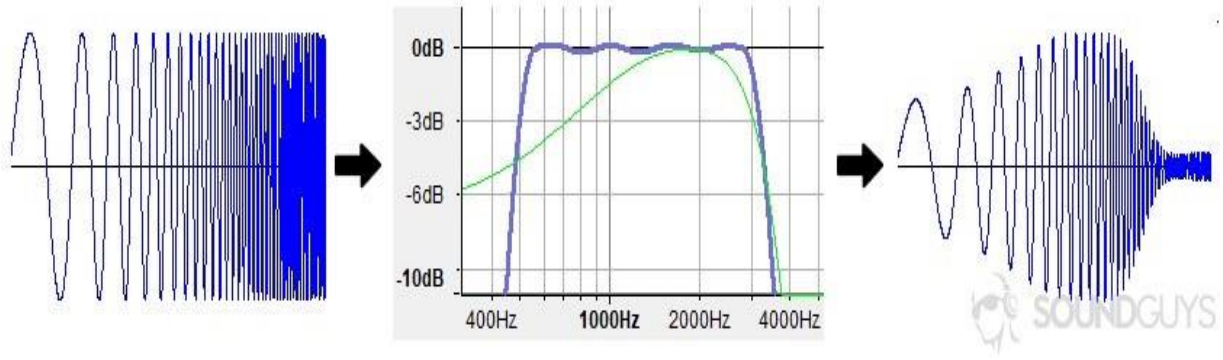
Functionality and working of the project

Function design part/ performance design

Playing the interface's main design The main interface should be constructed with simplicity and realism in mind. Android's overall interface is simple and easy, with its own set of configuration files. We may adapt numerous formats and resource files, such as photos, text, and colour reference, to meet your demands, resulting in a unique visual interface and a glossy appearance.

Adding tracks with a visual interface: - There are no suitable music when they initially enter the system; users must add tracks to play. The performance graph will be like

Existing Systems:-



As we can see that after song finishing, its frequency drops out randomly as class name AllSongFragment.createDataParse in not used but in out proposed model we had used this class which may produce the frequency as shown below in fig.

Proposed Systems:-

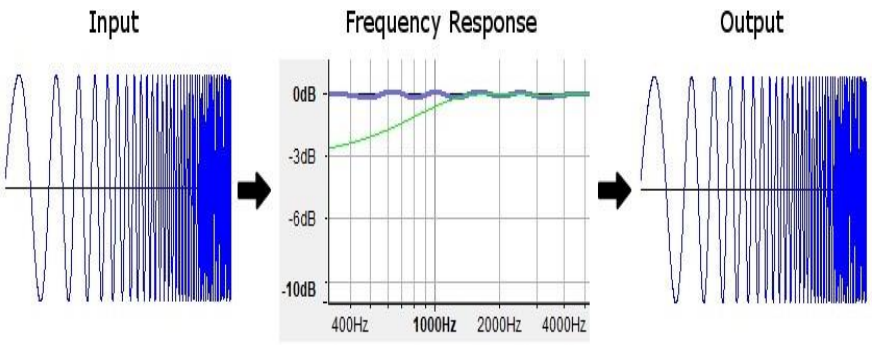


Figure 1: Frequency Response to input

As a result, you'll need to install a song-adding interface. Optional tracks to be loaded on the SD card must be placed to the empty playlist. Next / Move Past is a design for a play activity with music. To watch the activity, click the play button when you need to utilise the player to play the relevant music. When you need to transition from one song to another, you'll need to use the player.

Playlist Design: - A List View on Android, with the involvement of Base Adapter feature feature. This can show the path from top to bottom or left 2 right. This playlist recognises user-defined form, but the default system form for each line simply shows Text View. Each row in List View displays the name of one song.

As a Base Adapter, we defined a class music adapter. Also, using an algorithm, enlarge the adaptor such that the picture and song title may be presented in the first row. Because this Base Adapter is in invisible section, we needed to use the invisible "Get View" method, which returns View. Views can be displayed in Tasks, so a visible playlist connector will pop out.

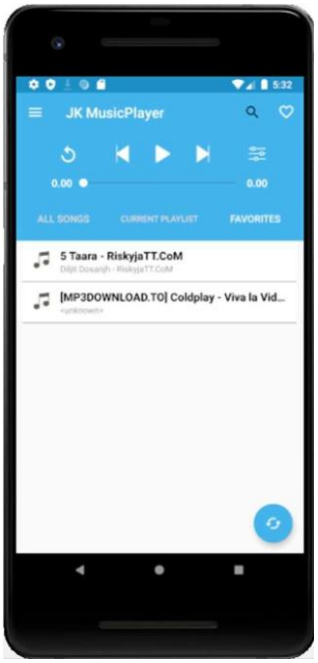


Figure 2: Playlist interface

System Design and comparison: - In this part, the design steps and outcomes of the operating modules in the system including other applications comparison are given in detail.

***AppStarting Module*:-**Any App-Starting requires AndroidManifist.xml file to get started. And any of latest/new project content will/be automatically generate/s/d AndroidManifst. XML file.

The configuration of those files are the core part of the entire system, which contains the Android SDK~version, as well as the default function of the operating system. Systems will automatically check the logo on AndroidManifest to respond to the corelated functionality when any part of the program triggers events.However existing one had hard coded the strings which is either old term supported or deprecated.

To define this system, the first phase is to launch the Android Activity combining features such as the action of <intent - filter>. Many of them are d-default system values. To set the action and category be aware of the switch between specific Tasks.

The announcement should be in Android Manifest. Xml files if any aspects of the system are to be used. The authority must be identified as a provider statement, to be unambiguous. Each part has various attributes, which the system will define differently depending on the demands. An overview of programme architecture The Android project's core architecture contains SRC (source code), gen (time generated automatically by the Android system), res (resource file), and file and picture layout in the programme archive's interface.

Interfaces and Optimization:-Existing systems provide interfaces which are not much attractive or take much time to response. For Example: iSangeet App of playstore which takes a long time to play next song and its also not optimized solution for the mentioned problem. For this a different class namely

SongAdapter.java must be implemented so that it acts like a bridge between view of Adapter and the data view which helps in optimization of the app.

Ratings in Navbar:-As per the quote “Customer is the owner” existing models don’t have in built ratings .Yes ofcourse ,they have playstore ratings but that ratings are publically visible which impacts the customer traffic for downloading the app. For this a highly confidential rating in-built is available which will help developer to make a changes accordingly as well as continuously which may increase traffic.

Part of the work design:- Key interactive game design Simple and effective play should be completely considered under/in the design/ing of the interface. The entire Android interface is an intuitive inter-face, with its own unique configuration files. We can customize various formats and resource file/s accordingly to needs, and create a unique visual interface and a great effect.

No matching songs when the first time user entering into the system; users need to add songs to play. Therefore, we need to install an interface to add songs. Empty playlist needs to add optional songs to the SD card to be added. Next play / music design / past music Click the play button to see the activity if you need to use the player to play the proper music.

When you need to use the player for switching back to the previous/s song, click the "Move Previous Music" buttun to see the activity. If you need to use the player to play the next song, click the "next music" button to see the activity .Comparing to other applications, our music player provides simple and attractive layout including favourite songs list,current playing songs and playlist like features which are not provided in applications like iSangeet App, Music P,etc.

The basic structure content of Android project includes: the SRC (source code), gen (constant that Android system automatically generates), res (resource file), and the layout of file and pictures in the main storage program interface.

Part of the function design:-

The main play interface design Convenience and practical should be fully considered in the design of the main interface. Every Android interface is a visual interface, which has its unique layout configuration files.

We can configure various layout and resources files according to the requirements, such as images, text and color reference, which can form different visual interface and glaring effect.

Interface design of adding songs There are no corresponding songs for the first time login entering the program; users need to add songs to play. Therefore, you need to enter the adding songs' interface. The empty playlist needs to add songs which can choose from the SD card to add. Function design of play and Next/Move Previous music

When need to use the player to play appropriate music, click the play button to realize the function. When need to use the player to switch to the previoussong, click on "Move Previous music" button to realize the function.

When need to use the player to play the next song, click on "the next music" button to realize the function and add MainActivity.java as well as PlaySong.java files as shown below…

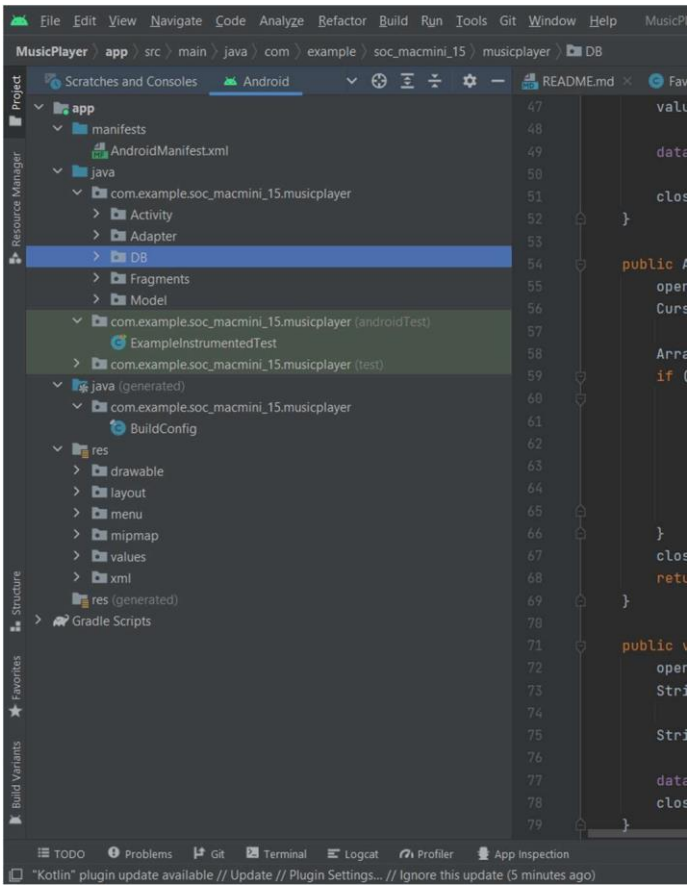


Figure 3: Files used

Part of the function design

The main play interface design Convenience and practical should be fully considered in the design of the main interface. Every Android interface is a visual interface, which has its unique layout configuration files. We can configure various layout and resources files according to the requirements, such as images, text and color reference, which can form different visual interface and glaring effect.
Interface design of adding songs .

There are no corresponding songs for the first time login entering the program; users need to add songs to play. Therefore, you need to enter the adding songs' interface. The empty playlist needs to add songs which can choose from the SD card to add. Function design of play and Next/Move Previous music When need to use the player to play appropriate music, click the play button to realize the function.

When need to use the player to switch to the previous song, click on “Move Previous music” button to realize the function. When need to use the player to play the next song, click on “the next music” button to realize the function.

Playlist interface design:-

View can be displayed on the Activity, so the playlist interface will come out.

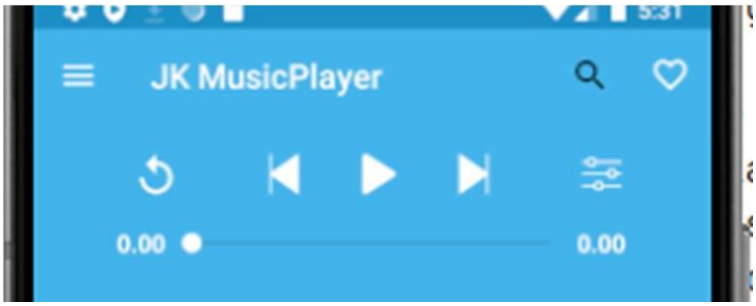


Figure 4: Activity panel

Working and Implementation

Step 1: Open a new android project
After opening the Android Studio you have to create a new project using **the** Empty

Activity with language as Java and give your project a unique name as you wish but don't

forget to keep the first alphabet capital.

1. Go to the top left corner and then hit **File->New->New Project** as shown in the following screenshot.

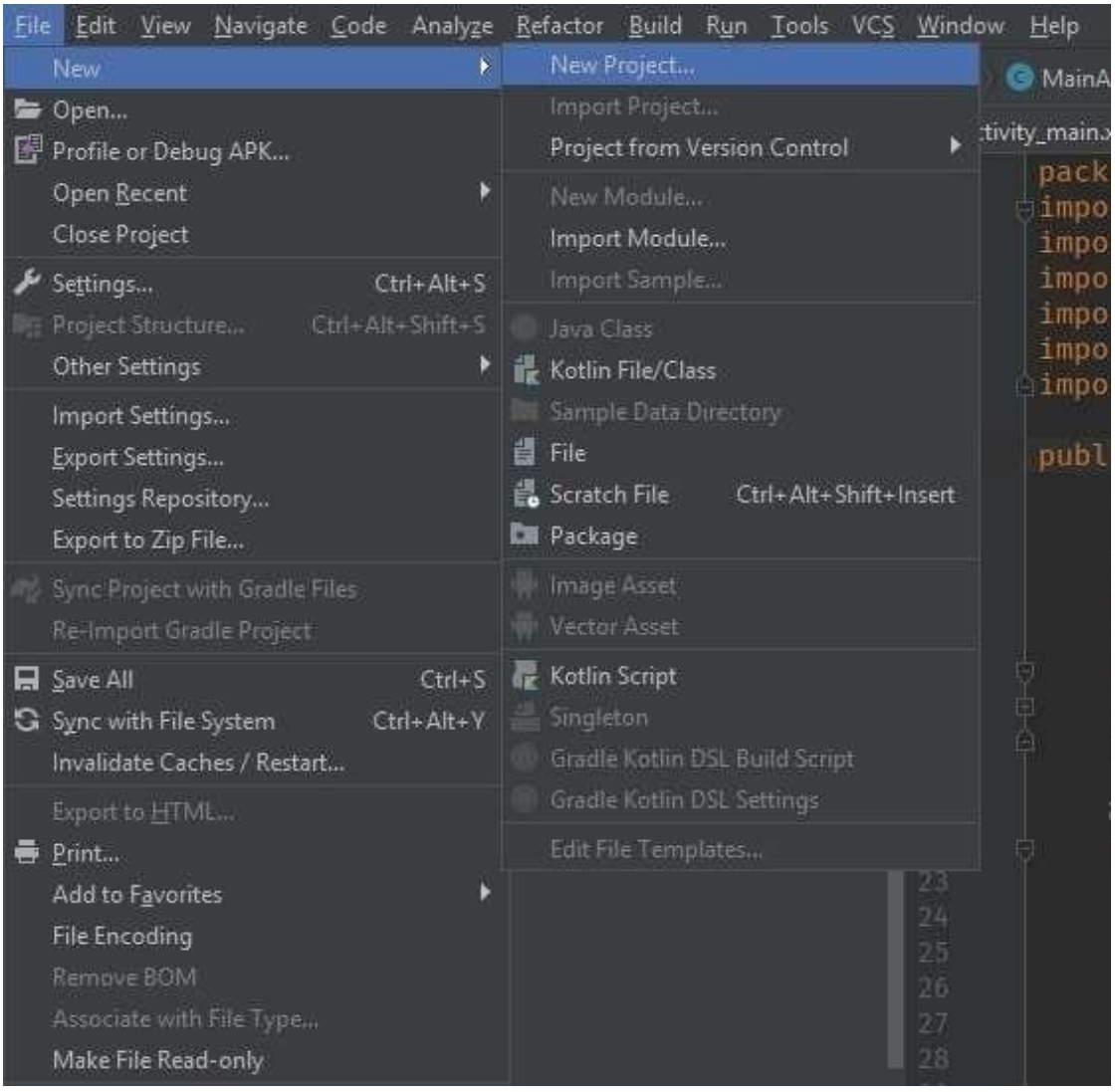
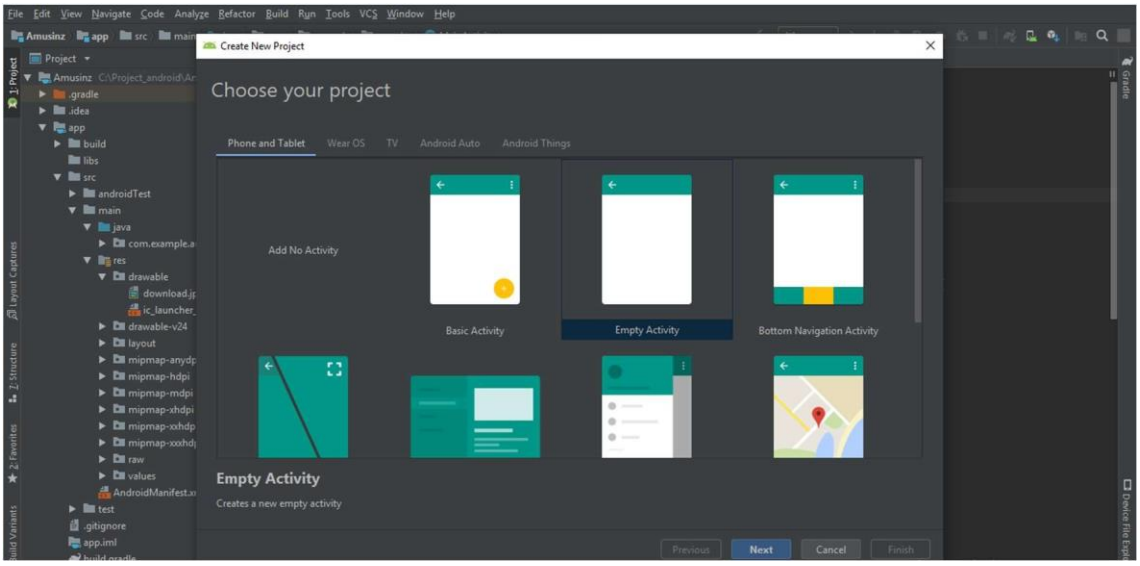
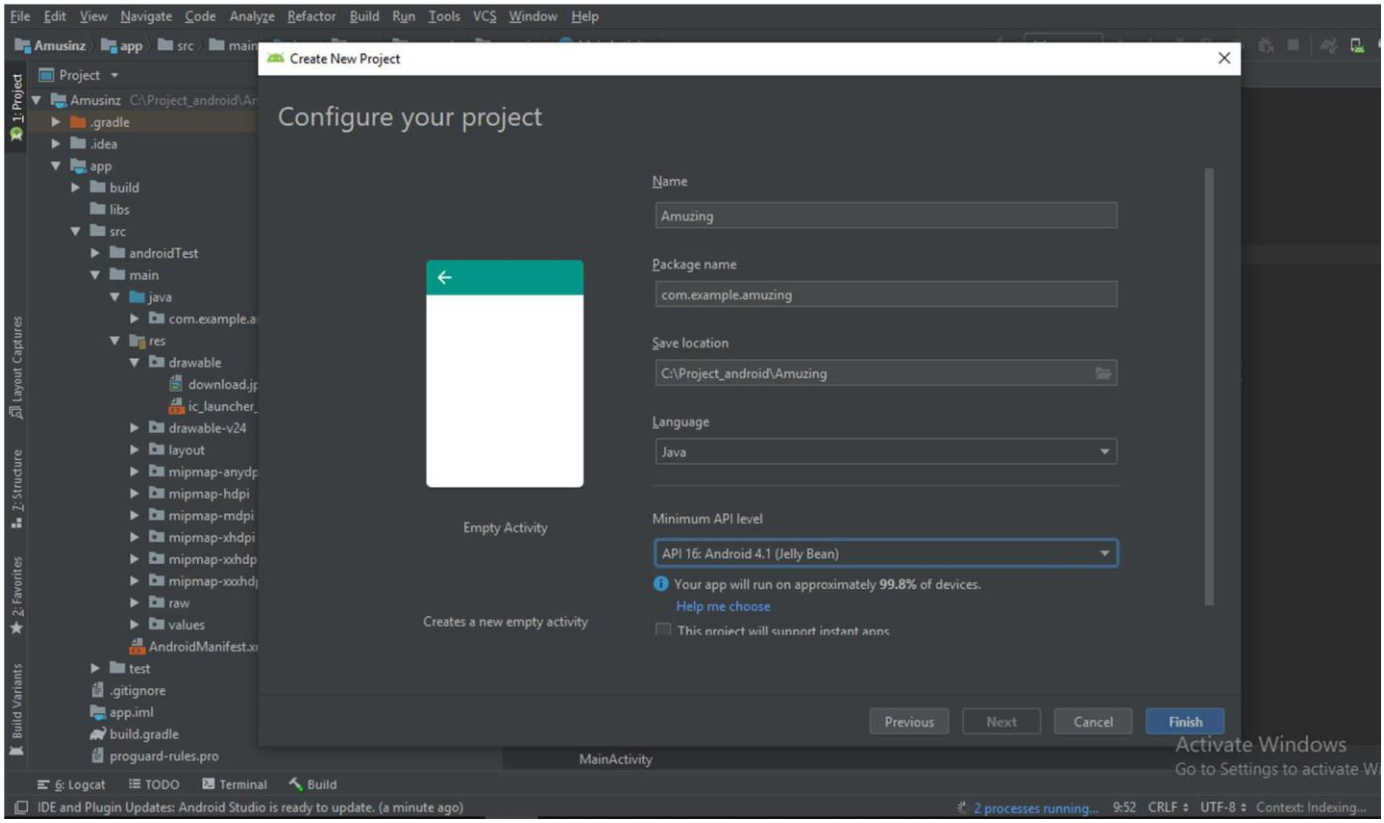


Figure 5: Steps to operate

2. Select **Empty Activity** as shown in the following screenshot.



3. Give your project a name, choose java and use lower level API so that your app can run on older version of android phones(I am using Api 16: Android 4.1 Jelly Bean).



Step 2: Designing the User Interface of the app

In this app, we have used 4 components:

- a imageView– to show our given image for the song
- 3 Buttons:
    - a play button to play our song
    - a pause button to pause our song
    - a stop button to stop our song

(Note: if we press play after pressing the pause then our song will continue playing immediately after where it was paused but if we press play button after stop then our song will play from the beginning)

These components are implemented on the below two layouts:

- Vertical LinearLayout
- Horizontal LinearLayout

Inside the LinearLayout (vertical) there are two components:

- imageView component
- LinearLayout(horizontal)

This layout will vertically divide our app screen in two halves.
The **imageView** component will be on upper half and the **Horizontal Linear**

Layout will be on the lower half. The horizontal layout will contain three buttons (play, pause and stop button). This horizontal layout will align these three buttons one after another horizontally on the lower half of our app screen.

To understand this clearly, please go through the following blue print and **Component Tree** of our app:

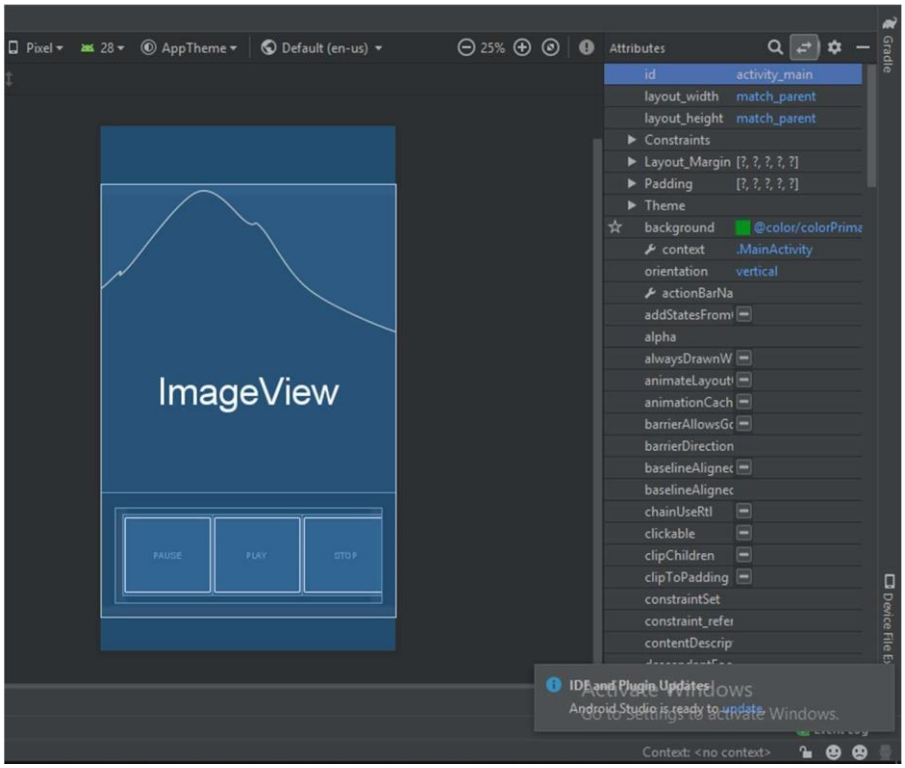


Figure 6: Layout view

In our app I have used different styles for play, pause and stop button by adding the following line of code:

android:background="@android:drawable/ic_media_play" for play button
android:background="@android:drawable/ic_media_pause" for pause button
android:background="@android:drawable/ic_delete" for stop button

Here JK Music logo is used in the app. Select the image and then paste it in the drawable directory. The path of the directory:

project->app->src->main->res->drawable

Step 3: Adding the music file to our app

Add the mp3 file to the raw folder. You can reach there by:
app-> res-> raw

If there is no raw folder, then create it by right-clicking on res directory then:

res-> new-> directory

Name the newly created directory as raw and add all the audio files in this folder. Drag and drop files here is not allowed. You have to copy your source file, then right-click on raw directory and click paste. Use "**show in explorer**" (if you are using windows) to go to that particular file. Make sure that the new name contains all small alphabets. The only valid characters are (a-z and 0-9 and _)

Step 4: Let's code the functionality of our App

1. Make a object of **MediaPlayer** class named **music**. It is an inbuilt class in **android package**. All the properties of the MediaPlayer class can be used by this music object:

   MediaPlayer music

2. We will add our music file to this newly created object by using **create** function :

   music = MediaPlayer.create(this, R.raw.sound);
   Please note that there is no need to add .mp3 or .wav or whatever filetype you are using. Just add the name of the file. (I have named my file as sound.mp3 so used R.raw.sound)

3. MediaPlayer class has an inbuilt function called **start** we will use this function
4. for play button. It will start the song.
   public void playSong(View v){
   music.start();
   }

5. For pause button we will use the inbuilt function **pause**. This will pause the song.
   public void pauseSong(View v) {
   mp.pause();

   }

6. For **stop** button we will use the inbuilt **stop** function. This function also deletes the object (music), so we create a new object with the same name.

```
public void stopSong(View v) {
mp.stop();
}

music = MediaPlayer.create(this, R.raw.sound);
```

Rest code is in MainActivity.java

Step 5: Let’s Run our app

Click the “Run” button at the Toolbar at the top to run our code.

You can run your app two ways:

- using Android Virtual Device (emulator)
- by connecting your phone using USB

Chapter – 4

Results and Discussion

Through the development of music player on Android platform, we get a clear understanding of overall process of the system. The core part of the music player is mainly composed of main interface, playlists, menus, play Settings, file browsing and song search.

Grasping the development of the six parts, the music player has had the preliminary scale. Based on the function of the six categories, add some other small features. Music player system realized the basic function of player: play, pause, and stop, up/down a, volume adjustment, and other functions.

This development implicated the popular mobile terminal development technology. This is the combination management of Java language in the open source mobile platform .

The system realized the music player programming. This design of music player based on Android system requires elaborate design of the music player framework, by adopting Eclipse3.5 + Java language as technical support of this system, with the Android plug-in tools, and combination of Android SDK2.1 version lead to the comprehensive and smoothly design and development of the mobile terminal.

This application can work effectively with high processing speed compared to the traditional methods. This proposed framework will reduce much complexity. Presently, the user has to download many. It will ease the people's difficulty in combining different features into a single app. These APIs can be made compatible with all the music applications people use.

The lower versions of Android like 7 or 8 should be able to work in this. This system could be made to work with other operating systems like Windows, iOS.

The above files including many other files placed in res folder namely Drawable , layout, menu etc will produce the final output as:-

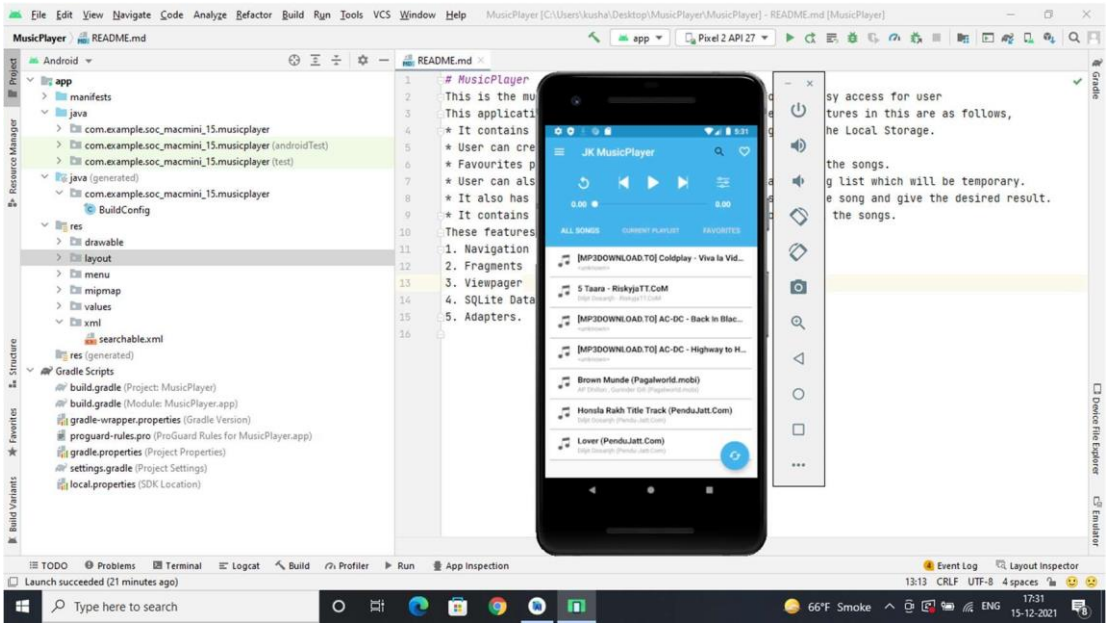
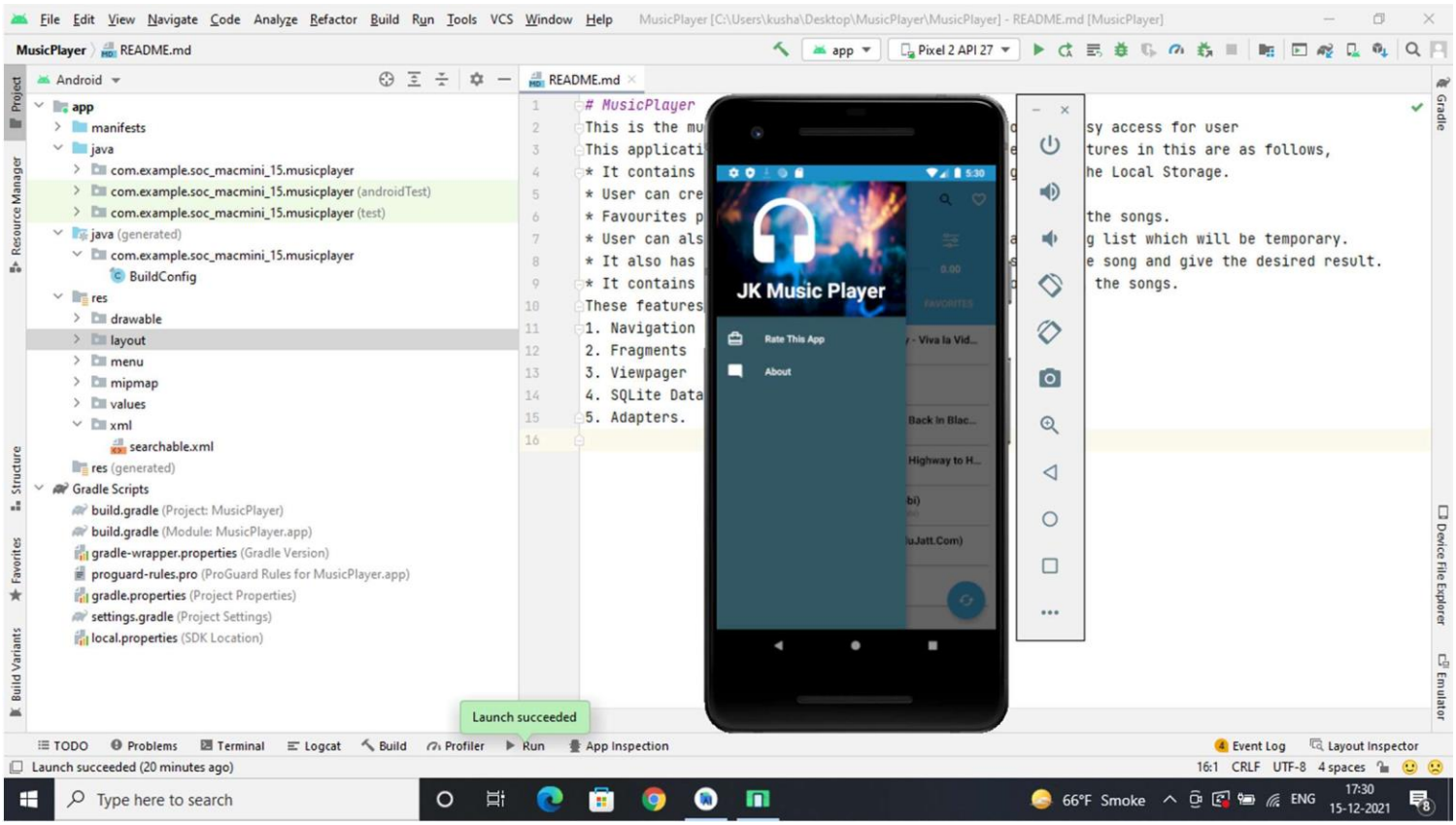
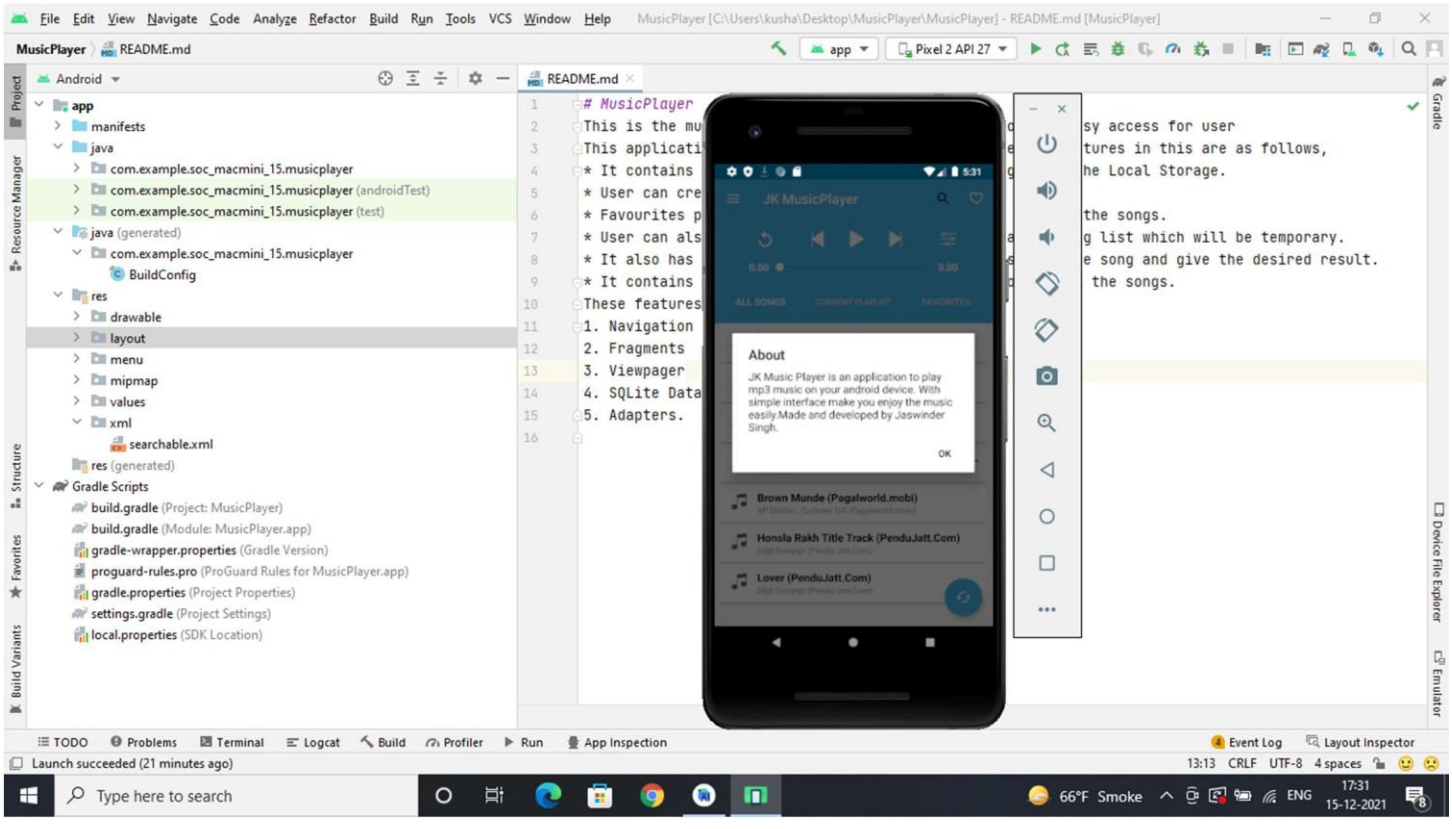


Figure 7: Implementation and working of Interface


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicPlayer [C:\Users\kusha\Desktop\MusicPlayer\MusicPlayer] - README.md [MusicPlayer]
MusicPlayer README.md
app Pixel 2 API 27
Android
app
manifests
java
com.example.soc_macmini_15.musicplayer
com.example.soc_macmini_15.musicplayer (androidTest)
com.example.soc_macmini_15.musicplayer (test)
java (generated)
com.example.soc_macmini_15.musicplayer
BuildConfig
res
drawable
layout
menu
mipmap
values
xml
searchable.xml
res (generated)
Gradle Scripts
build.gradle (Project: MusicPlayer)
build.gradle (Module: MusicPlayer.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for MusicPlayer.app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
README.md
MusicPlayer
This is the mu
This applicati
* It contains
* User can cre
* Favourites p
* User can als
* It also has
* It contains
These features
1. Navigation
2. Fragments
3. Viewpager
4. SQLite Data
5. Adapters.
JK Music Player
Rate This App
About
sy access for user
tures in this are as follows,
he Local Storage.
the songs.
g list which will be temporary.
e song and give the desired result.
the songs.
Launch succeeded
TODO Problems Terminal Logcat Build Profiler Run App Inspection
Event Log Layout Inspector
Launch succeeded (20 minutes ago)
16:1 CRLF UTF-8 4 spaces
Type here to search
66°F Smoke
ENG
17:30
15-12-2021


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicPlayer [C:\Users\kusha\Desktop\MusicPlayer\MusicPlayer] - README.md [MusicPlayer]
MusicPlayer README.md
app Pixel 2 API 27
Android
app
manifests
java
com.example.soc_macmini_15.musicplayer
com.example.soc_macmini_15.musicplayer (androidTest)
com.example.soc_macmini_15.musicplayer (test)
java (generated)
com.example.soc_macmini_15.musicplayer
BuildConfig
res
drawable
layout
menu
mipmap
values
xml
searchable.xml
res (generated)
Gradle Scripts
build.gradle (Project: MusicPlayer)
build.gradle (Module: MusicPlayer.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for MusicPlayer.app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
README.md
MusicPlayer
This is the mu
This applicati
* It contains
* User can cre
* Favourites p
* User can als
* It also has
* It contains
These features
1. Navigation
2. Fragments
3. Viewpager
4. SQLite Data
5. Adapters.
JK MusicPlayer
About
JK Music Player is an application to play mp3 music on your android device. With simple interface make you enjoy the music easily.Made and developed by Jaswinder Singh.
OK
Brown Munde (Pagalworld.mobi)
Honsla Rakh Title Track (PenduJatt.Com)
Lover (PenduJatt.Com)
sy access for user
tures in this are as follows,
he Local Storage.
the songs.
g list which will be temporary.
e song and give the desired result.
the songs.
TODO Problems Terminal Logcat Build Profiler Run App Inspection
Event Log Layout Inspector
Launch succeeded (21 minutes ago)
13:13 CRLF UTF-8 4 spaces
Type here to search
66°F Smoke
ENG
17:31
15-12-2021

Chapter – 5

Conclusion and Future Scope

We gain a comprehensive grasp of the whole process of the system, both existent and intended, via the construction of a music player on the Android platform, which includes various research paper analysis and findings.

The primary interface, playlists, menus, play settings, file browsing, and song search make up the majority of the music player.

Add some tiny features based on the six-phase work. The music player's essential operations are recognised by the system: play, pause, pause, up/down a, volume adjustment, and other functions.

The Java programming language is discussed in an open source forum. The music player application was identified by the system.

This Android system-based music player design necessitates a comprehensive design of the music player framework, with Android plug-in tools, and a combination of Android version SDK2.

1 leading to full design and smoothness and development of the portable terminal. This programme has the ability to Compared to older approaches, operate more efficiently with faster processing.

The suggested framework will simplify things. Meanwhile, the user should continue to download. It will make it easier for individuals to combine diverse functionalities into a single application.

These APIs may be adjusted to work with any music application. Low-end Android versions like 7 and 8 should be able to handle it.

Other operating systems, such as Windows and iOS, can be used to alter this programme.

The goal of this work was to look at music player applications and published studies, as well as our own JK music player, which reads files via external storage. Users who desire music based on their downloads and just formats and emotional behaviour will benefit greatly from the programme.

It will aid in lowering music search time, hence eliminating superfluous computation time and increasing the system's overall accuracy and efficiency.

The system will bring more delight to music listeners by providing a large number of songs that are acceptable or fit for the user's needs. We also provided the planned system exposure in this study.

Future Scope

This application can work effectively with high processing speed compared to the traditional methods. This proposed framework will reduce much complexity. Presently, the user has to download many. It will ease the people's difficulty in combining different features into a single app. These APIs can be made compatible with all the music applications people use. In future we will be using AI for face-detection and fingerprint scanner and also try to implement it for mood based song playing.

Reference

[1] Matthew E. P. Davies, Yoshii, and Masataka, "Automatic Creation of Multi-Song Music Mashups" , IEEE/ACM transactions on audio, speech, and language processing, vol. 22, no. 12, December 2014.

[2] ieeexplore.ieee.org/document/5628891 author Youg Cai Pan"Development and Research of Music player app based on android

[3] eprint.utar reference on music player development based on Android author Anonymous.

[4] Mark L.Murphy, Li,X. interpreted Guide of Android Development journal , People's Posts and Telecommunications Publishing House,2010-12:128～156.

[5] Emotion Based Music Player by Nikhil Zaware B.E.Computer, B.E. Computer, Department of Com Eng., Pune, March 2014.

[6] Music Detector: A Prototype Software Tool for Playlist Generation by Luís C

[7] Karthik Nathan ; Manasi Arun IEEE International Symposium on Music application and Information Technology (ISSPIT)

[8] Neural Networks for Emotion Classification. At Leiden institute of Advanced Computer Science, August 2003