

GOVERNMENT POLYTECHNIC, PUNE-16

(An Autonomous Institute of Government of Maharashtra)



Project report on

ANDROID MUSIC PLAYER

Submitted By

Atharva Girish Deshmukh

(1706017)

Tejas Ramesh Atkale

(1706001)

Department of Computer

Engineering Government

Polytechnic, Pune. (2019-2020)

ABSTRACT

In order to solve the problem of complex functions and large required memory of mobile phone music player on the current market, a new music player of simple, convenient, less required memory as well as user-friendly is developed. Based on the Android technology, using the Java language and Eclipse programming tools lead to design and coding of music player. The new design mainly realizes six core functions including main play interface, playlists, menus, play settings, file browsing and song search. This player has merits of high performance, simple operation, and run independently on the Android mobile devices. At the same time, the player can also browse and access files in mobile phones.

This software is used for android operating system, it can be used easily for music and it's convenient and quick, use simple UI realized the use function, in this software music can free play, and you can use it to change the state of the mobile phone, unique menu can give the users perfect experience. This paper briefly introduces the system of project background and significance, emphatically elaborated the database design and the realization of the function of order and related to the demand analysis of the system, module design, database design to the module realize each task such as for a detailed analysis and description.

INDEX

Chapter No	Topic	Page No
1.	Introduction	4
2.	Android Operating System 2.1. Build Developing Environment for Android 2.2. Design Principal of Android Application 2.3 Functions and Structure of Android System	5
3.	Android Studio	9
4.	Working of Application 4.1 Introduction of Player Project 4.2 Part of Function Design 4.3 Play-Pause, Next-Previous Button 4.4 Sliding Menu 4.5 Favourites List 4.6 Current Playing Song 4.7 Searching a Song 4.8 Replaying & Looping 4.9 Refreshing 4.10 About-Us Box	10
5.	Test Cases	22
6.	Conclusion	24

1. INTRODUCTION

Android is open source code mobile phone operating system that comes out by Google in November 2007. Its appearance has broken the traditional closed mobile phone operating system. Anyone can modify the mobile phone operating system as well as function according to personal preference, which is also the most attractive merit of Android. Music player in this article is application software based on Google Android.

Android's application on mobile terminals also completely broke the traditional understanding of the mobile terminals. And appreciate music is one of the best ways to relieve pressure in stressful modern society life. Therefore, many kinds of mobile phone players are also developed. However, a lot of players devote to fancy appearance and function, while caused resources wasting to the user's mobile phone, such as large required memory and CPU, which brings a lot of inconvenience as multiple programs running at the same time. For the most ordinary users, many functions are useless.

The purpose of this article is to develop a player which can play the mainstream music file format. To browse and query the storage space as well as operation of adding, deleting, and playing can be realized. Meanwhile, this software can play, pause and select songs with latest Btn and next Btn according to users' requirement as well as set up songs' order and etc.,

Music player based on Android application is popular in the market at the present. The completing development of Android operating system gives developers a nice platform, which can learn the popular computer technology combining with learned knowledge, and master the latest knowledge, enrich oneself, and enjoy entertainment.

2. ANDROID OPERATING SYSTEM

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The current stable version is Android 10, released on September 3, 2019. The core Android source code is known as Android Open Source Project (AOSP), which is primarily licensed under the Apache License. This has allowed variants of Android to be developed on a range of other electronics, such as game consoles, digital cameras, PCs and others, each with a specialized user interface. Some well known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google.

Android's source code has been used as the basis of different ecosystems, most notably that of Google which is associated with a suite of proprietary software called Google Mobile Services (GMS), that frequently comes pre-installed on said devices. This includes core apps such as Gmail, the digital distribution platform Google Play and associated Google Play Services development platform, and usually apps such as the Google Chrome web browser. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google. Other competing Android ecosystems include Amazon.com's Fire OS, or LineageOS. Software distribution is generally offered through proprietary application stores like Google Play Store or Samsung Galaxy Store, or open source platforms like Aptoide or F-Droid, which use software packages in the APK format.

2.1. BUILD DEVELOPING ENVIRONMENT OF ANDROID

The applications of Android need to run based on Android environment. The following is the configuration requirement and installation steps of Android development environment.

The required software of the developing environment

- Operation system: WindowsXP, Linux, Windows 7 or above
- Software: Android SDK (Software Development Kit), ADT(Android Development Tool)
- IDE environment: Eclipse IDE + ADT Eclipse3 or higher, Android Studio
- JDK : Java Runtime Environment virtual machine, Java Development Kit(JDK)

Installation steps of the developing environment

- Step 1: Install the Java virtual machine JDK version - 6 or above
- Step 2: Install Android Studio; download address: <https://developer.android.com/studio>
- Step 3: Install the Android SDK: first download the Android SDK
Download address: <http://developer-android-com/sdk/index-html>
- Step 4: Install Android ADT plug-in, run Eclipse and select help - > install new software and select

add. Input SDK tools path in the SDK location: D: \ android \ software \ android SDK

—

Windows and click OK. The Android environment is set up successfully.

2.2. DESIGN PRINCIPLE OF ANDROID APPLICATION

Twice the result with half the effort will get if an overall study of the principles done before the design and follow them in the operation.

The principle of software design mainly includes the following points:

(1) Reliability

The reliability of the software design must be determined. The reliability of the software

system refers to the ability to avoid fault occurred in the process of system running, as well as the ability to remedy troubles once the fault occurs.

(2) Reusability

Look for commonness of similar codes and come out new method abstractly and reasonably. Pay attention to the generic design.

(3) Understandability

The understandability of software not only require clear and readable document, but the simplified structure of software itself, which requires the designer possess keen insight and creativity, and know well about the design objects.

(4) Simple program

To keep the program simple and clear, good programmers can use simple program to solve complex problems.

(5) Testability

Testability means that the created system has a proper data collection to conduct a comprehensive test of the entire system.

(6) The Open-Closed Principal

Module is extensible but cannot be modified. That is to say, extension is open to the existing code in order to adapt to the new requirements. While modify is closed to the categories. Once the design is completed, the categories cannot be modified.

2.3. FUNCTIONS AND STRUCTURE OF ANDROID SYSTEM

This system adopts the modularized program design, and system function is correspondingly divided into function modules, the main modules include:

(1)UI function module design of mobile terminal: the index screen, play screen, music adding page, file management page are realized.

(2) Backstage function module design of mobile terminal: the specific function, music file data storage function and other function are implemented.

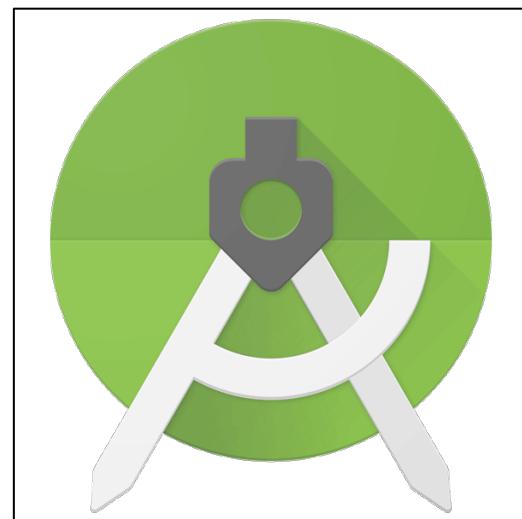
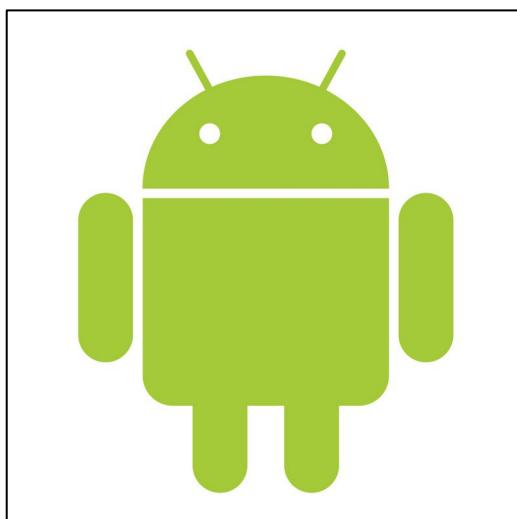


Fig: Android & Android Studio

3. ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

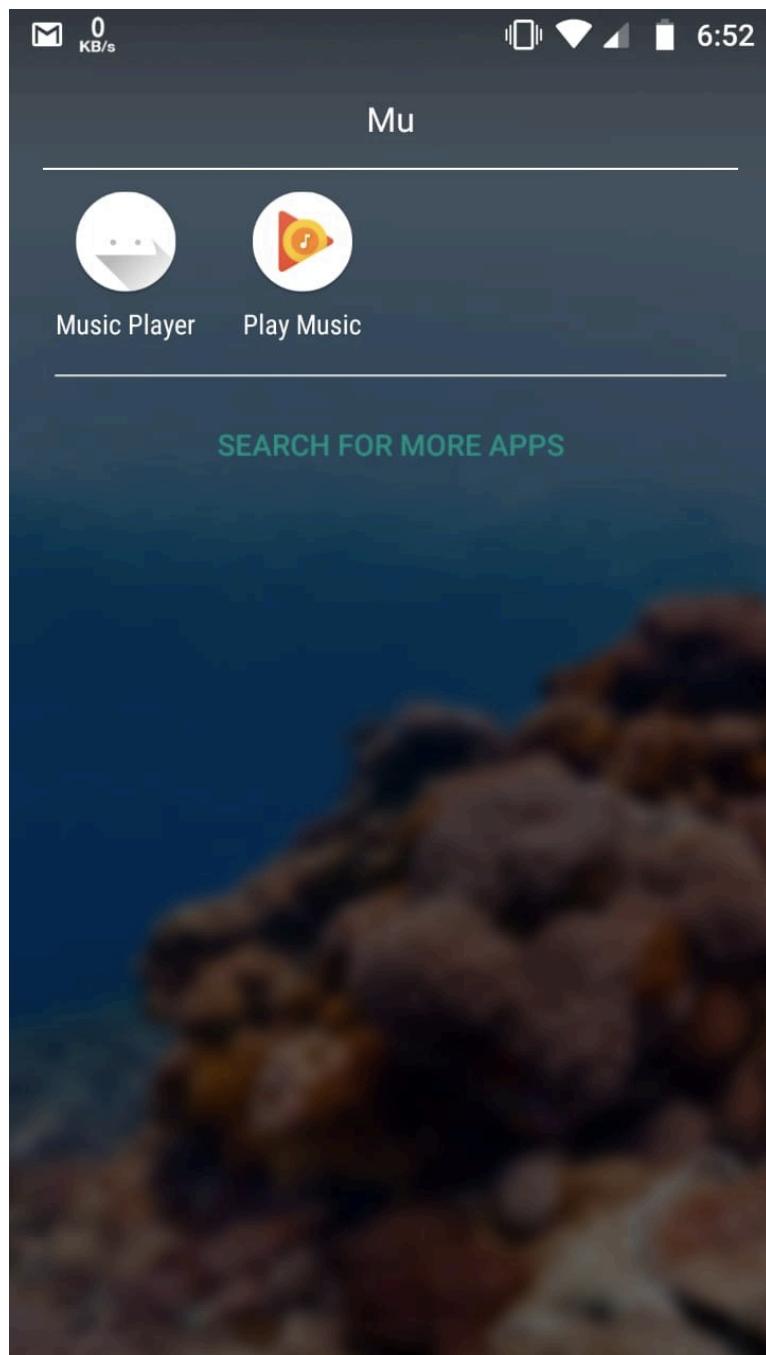
On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

FEATURES OF THE CURRENT STABLE VERSION:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

4. WORKING OF APPLICATION

In this chapter, design steps and the results of functional modules in the system are given in details.



4.1. INTRODUCTION OF PLAYER PROJECT

In this section, the App Starting module of the player in the project is introduced, as well as the Android engineering program structure, etc.

Introduction of AppStarting module in the project

Any AppStarting needs AndroidManifest. XML file to start. And any new project content will automatically generate an AndroidManifest. XML file. Configuration files are the core of the whole program, which contains the Android SDK version, and the default Activity in program running. The systems will automatically looking for a logo in AndroidManifest to react the corresponding operation when any component of the program triggers events.

To define the system, the first thing is launching the Activity: Android Activity. There are properties such as action and category in < intent - filter >. Most of these are the default values of the system. Setting the action and category realize the switch between different Activities. When any components of the program is about to use, declaration must be in the Android Manifest. Xml files.

To be clear that authorities must be illustrated as the statement of provider. Each component has a lot of attributes; the program will define different attributes according to different needs.

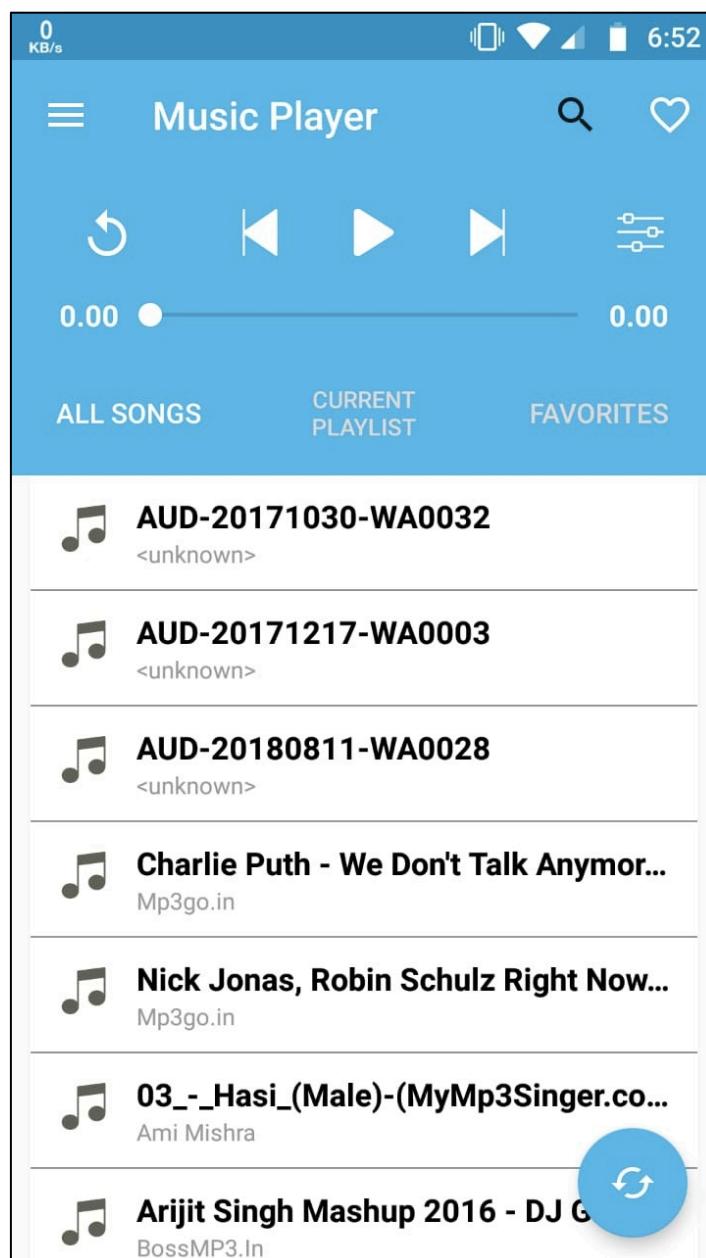
4.2. PART OF FUNCTION DESIGN

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.

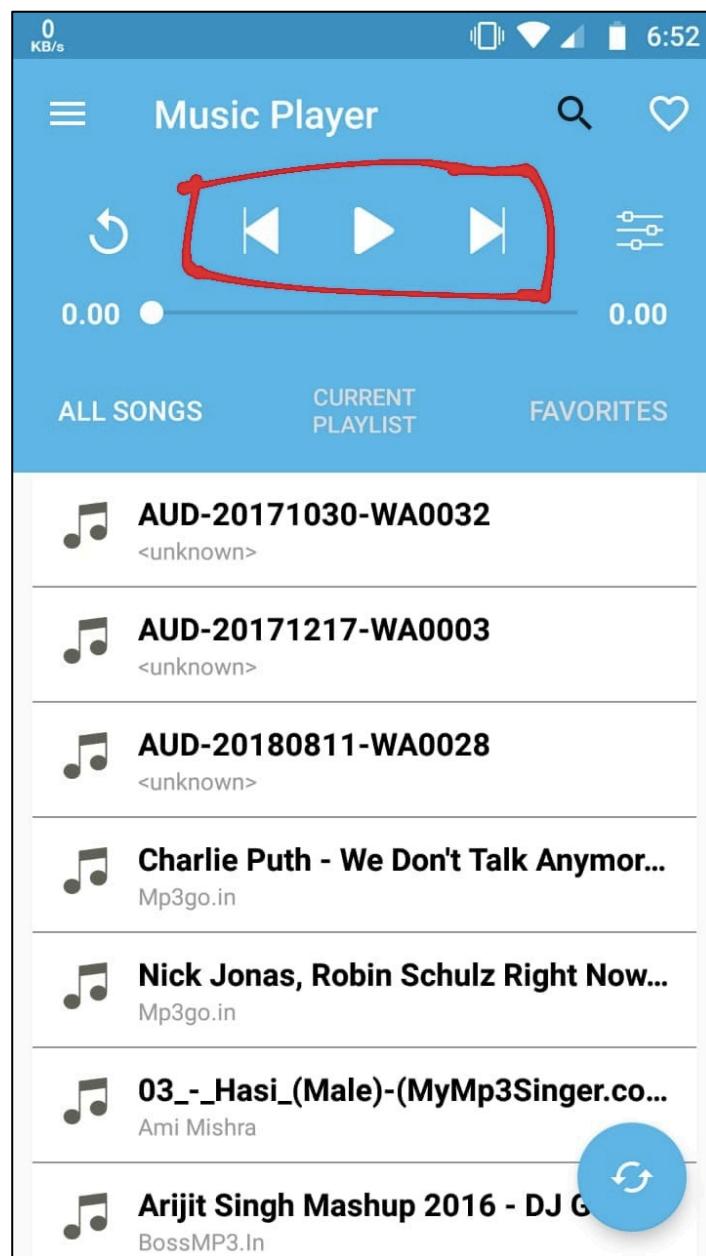
Interface design of adding songs: There are no corresponding songs for the first-time login entering the program; users need to add songs to play. Therefore, you need to enter the adding songs' interface. The empty playlist needs to add songs which can choose from the SD card to add.



4.3. PLAY-PAUSE, NEXT-PREVIOUS BUTTON

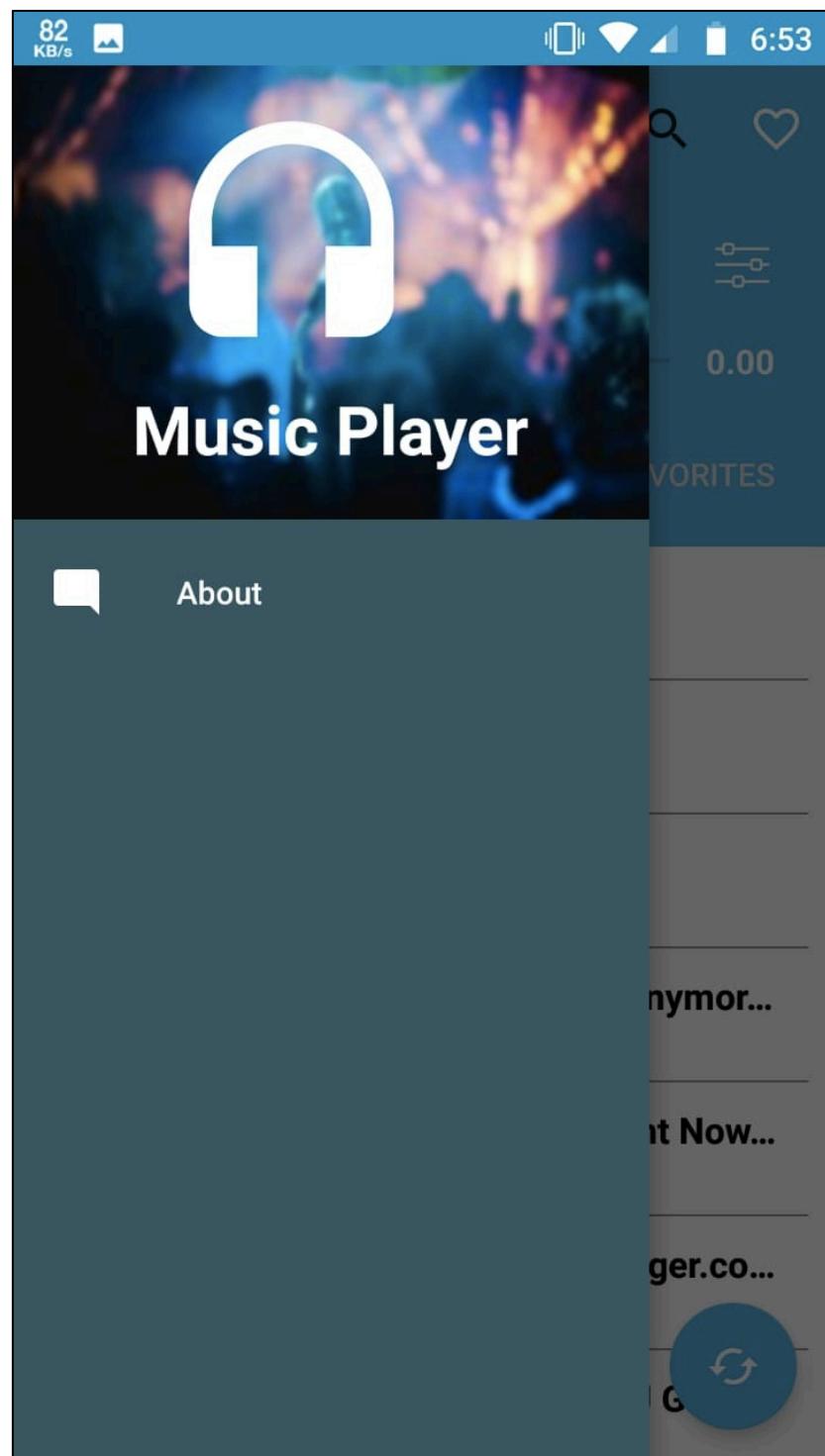
Play- Pause button is used to play the paused song or pause the playing song in the music player.

Next button is used to play the next song in the queue and likewise the previous button is used to play the previous button in the queue.



4.4. SLIDING MENU

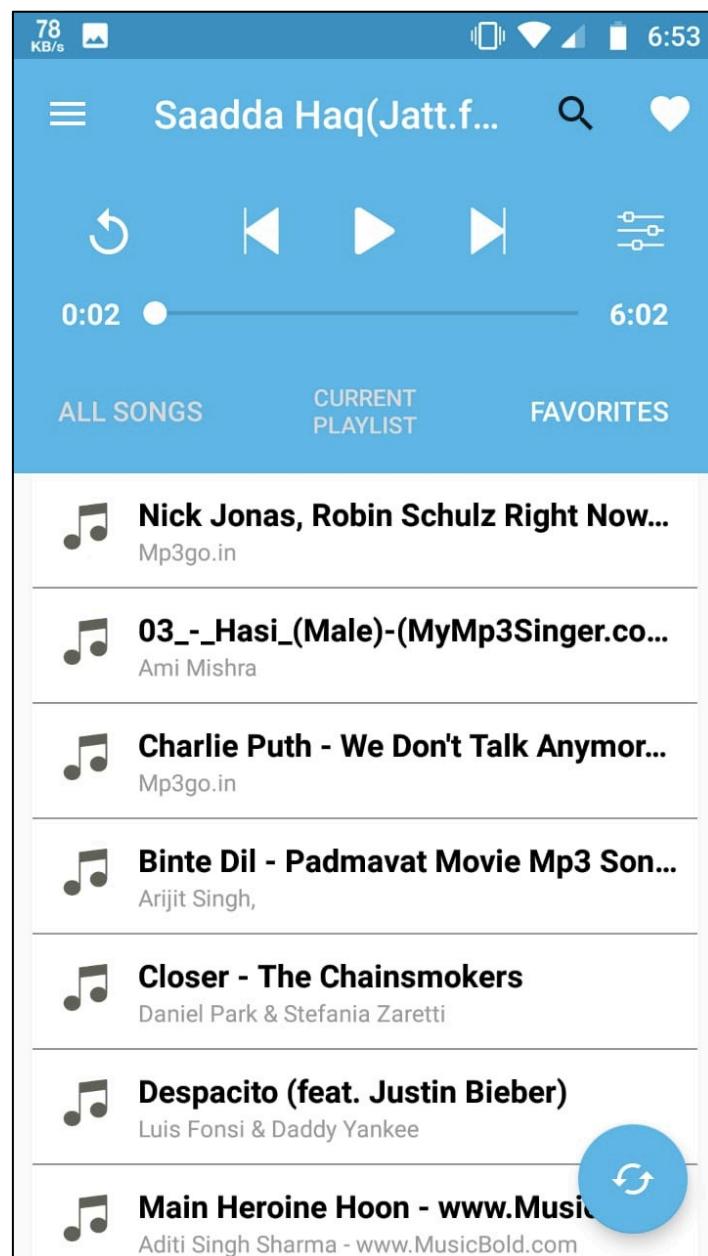
Sliding Menu with options is created using the Navigation Drawer to provide user navigate through other features of app.



4.5. FAVOURITES LIST

The music player also allows the user to add his favourite/liked song to the favourites list. This then reduces the efforts of the user to search and play his favourite songs.

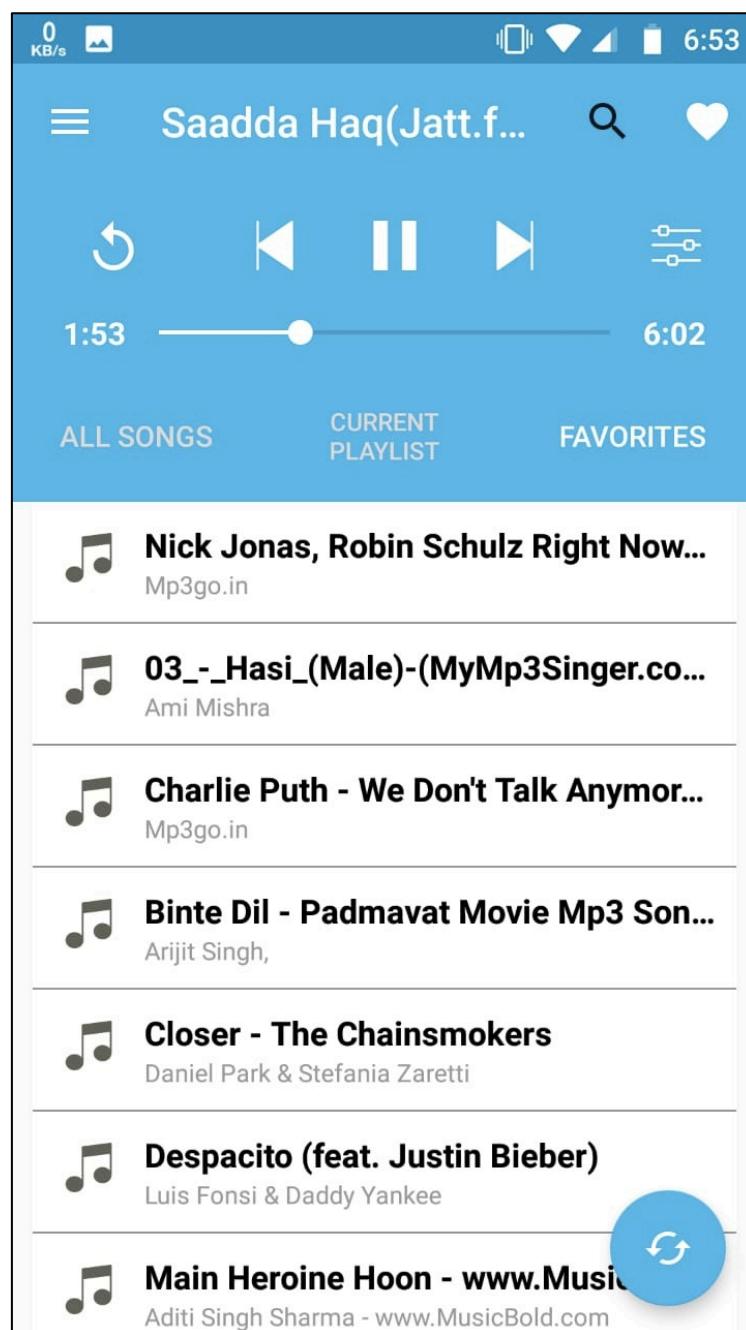
The user can add his/her favourite song to the favourites by pressing on the heart icon right beside the search icon. When the heart icon is filled, the song is added to the favourites list.



4.6. CURRENT PLAYING SONG

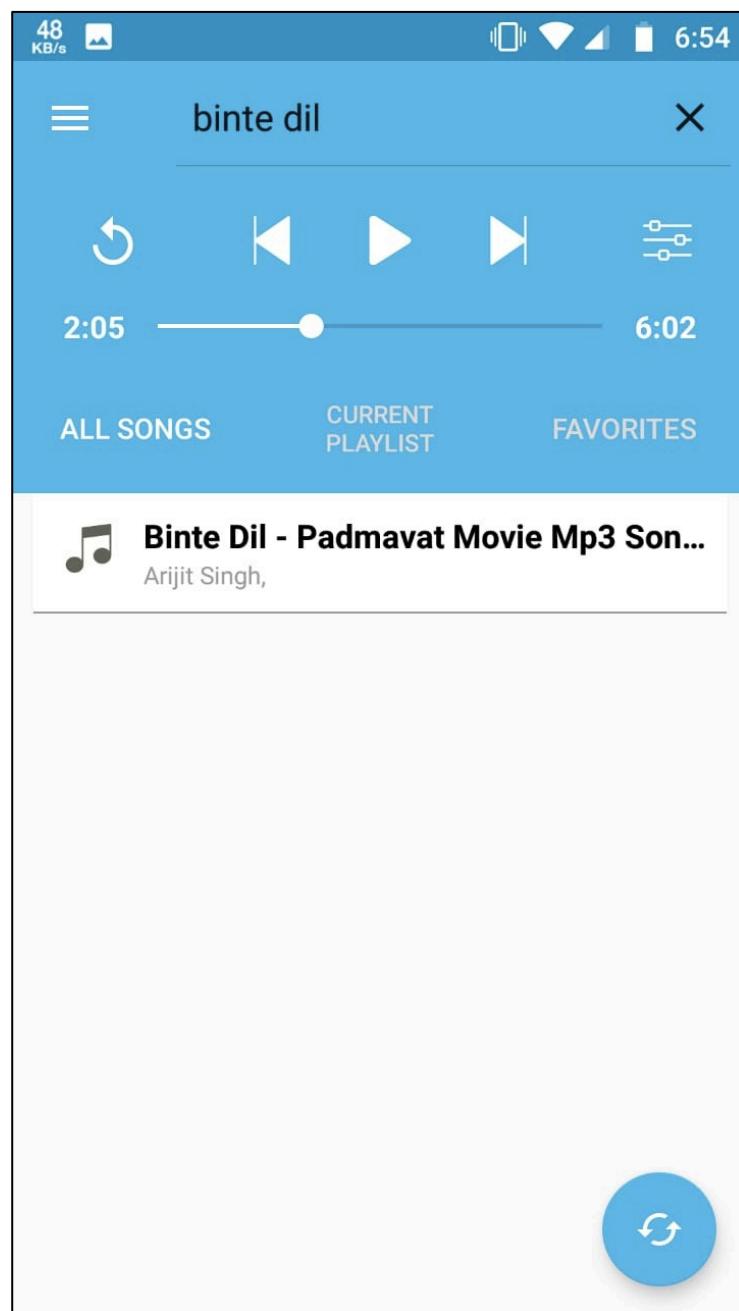
While the current song is being played on the android music player, the user can see the stream length of the song, the current stream time.

The user is also able to see the name of the current playing song on the extreme top of the app window.



4.7. SEARCHING A SONG

The music player app allows the user to search his preferable song from the Songs list. The user just has to write the song name or the song initials in the search bar by pressing the search icon. When the required song appears in the list, the player plays it by pressing on it.

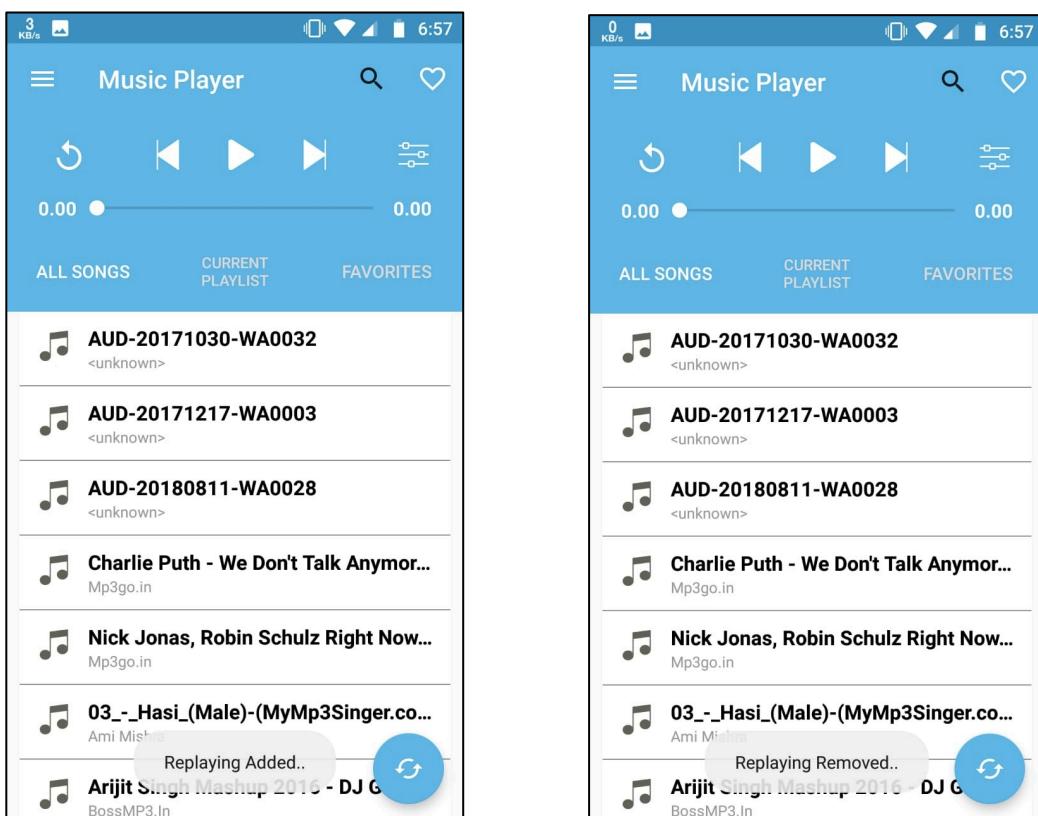


4.8. REPLAYING & LOOPING

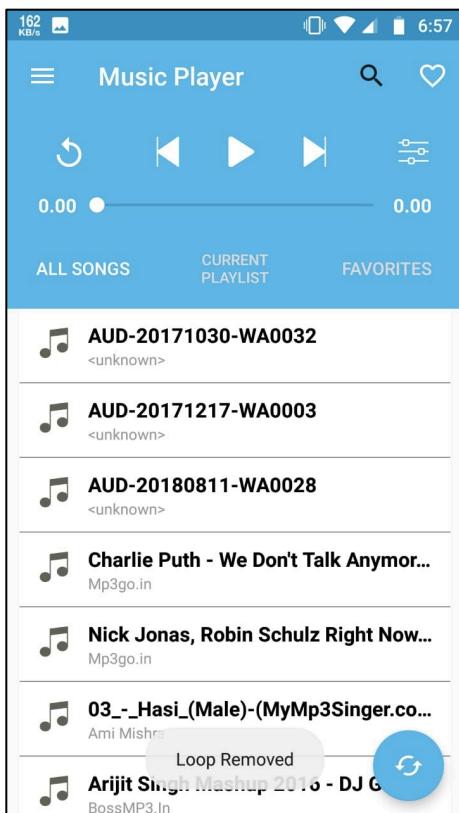
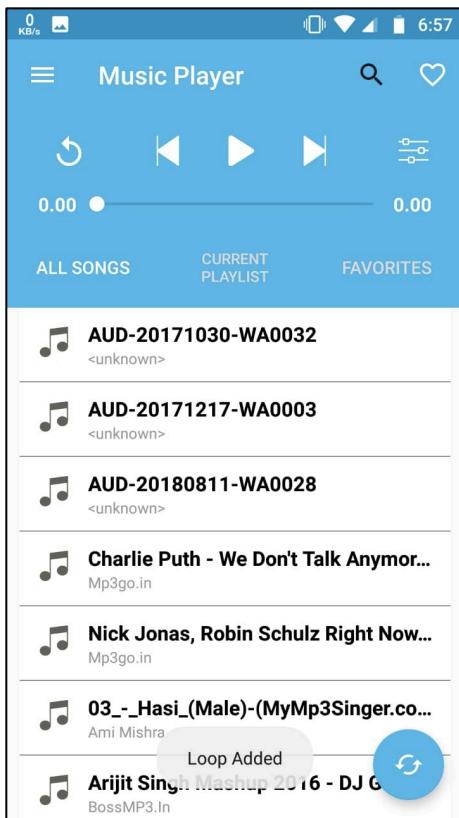
Replaying allows the user to replay a specific song once in the queue or in the playlist.

Looping allows the user to replay a specific song for infinite times until the user has pressed looping button for unlooping in the queue or in the playlist.

Replaying:

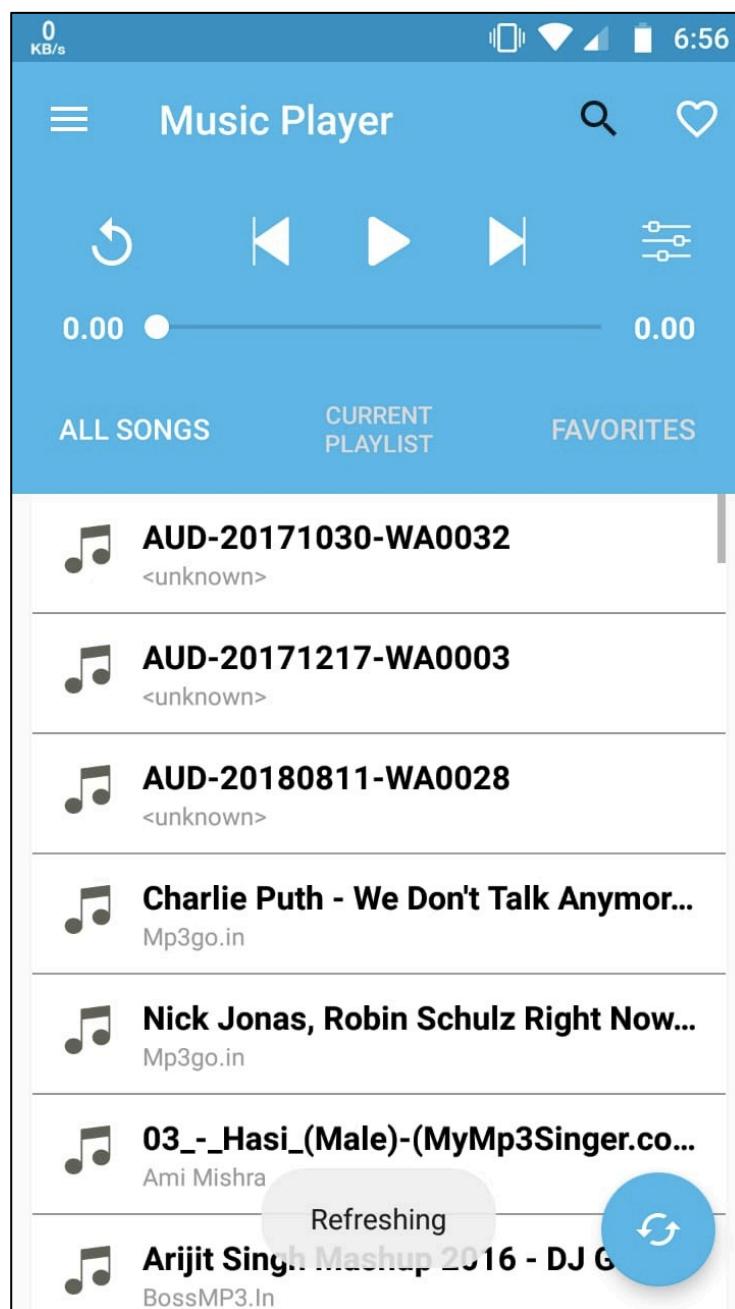


Looping:



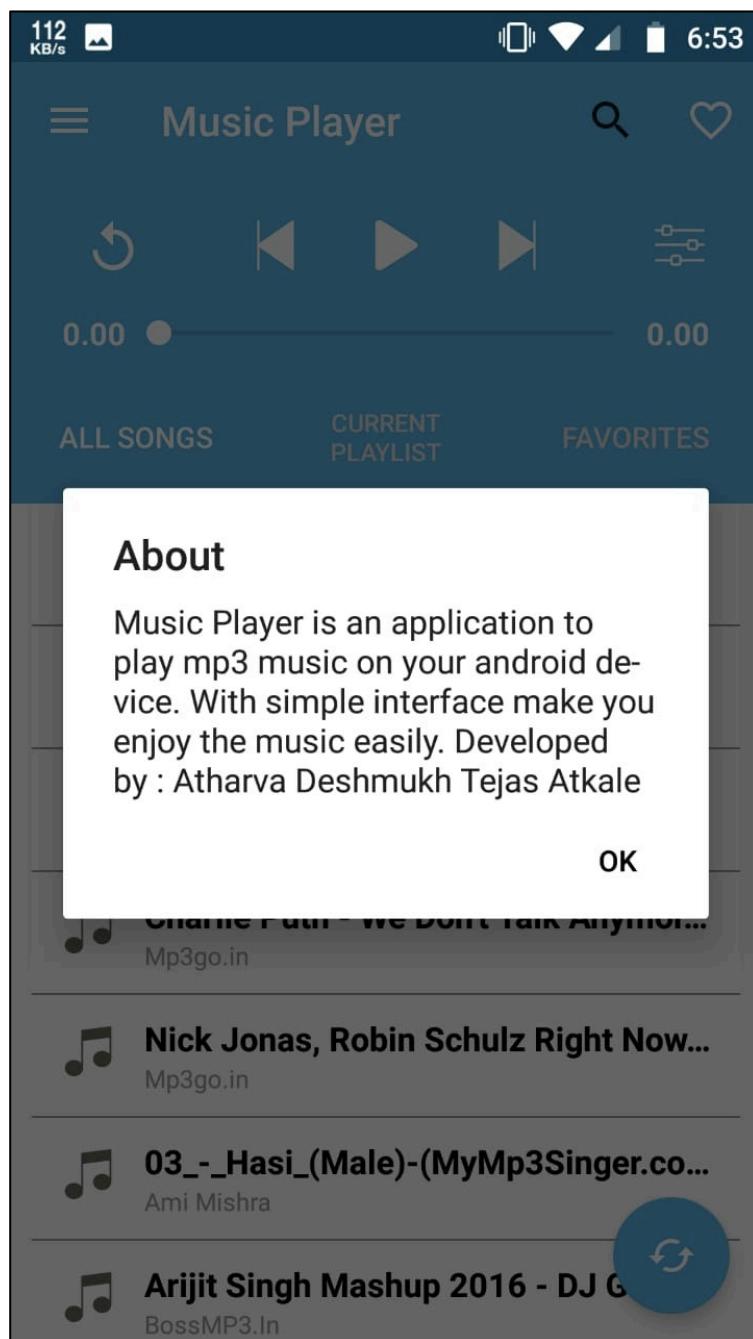
4.9. REFRESHING

Refreshing allows the user to refresh the song, queue, playlist and the music player once. This can be done by pressing the refresh icon on the left of previous icon. This feature works just like the windows refresh feature.



4.10. ABOUT US BOX

About-Us box provide the user with the information on the music player application and also some information about the developers of the application project.



5. TEST CASES

Ref No.	Test Data	Expected Outcome	Final Result
1.	Install the Android Application on the Android Operating System.	Application is installed on the Android Device Successfully	Pass
2.	Start the Application by clicking its icon	Application is started	Pass
3.	Splash Screen of the Application	Splash Screen is coming after the starting of Application	Pass
4.	Coming of Main Screen after the Splash Screen	Intent has been successfully working	Pass
5.	Opening of Song Playlist after pressing playlist Button	Song Playlist comes	Pass

6.	Selecting Any song by pressing song on song playlist.	Song plays	Pass
7.	Press Forward Button	Next Song Plays	Pass
8.	Press Back Button	Previous Song Plays	Pass
9.	Press Pause Button	Song Pause	Pass
10.	Press Play Button	Pause song plays	Pass
11.	After the Ending of Song Next Song in the Playlist Plays	Inline Playlist is working	Pass
12.	Press the Playlist during the playing of song	Playlist open without any trouble in current music playing	Pass
13.	Pressing the exit Button on the main screen	Song stop and application exits successfully	Pass

6. CONCLUSION

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, lyrics display, play mode, song search, file browser, playlists query, 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 based on Linux system+ + SQLite database support+ SharePreference configuration file. 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.

**THANK
YOU**