Music Streaming and Event Booking Application

by Satyendu Panda

Submission date: 28-Dec-2020 02:22PM (UTC+0530)
Submission ID: 1481635823
File name: 1NZ19MCA23.pdf (1.44M)
Word count: 3919
Character count: 19586

Name- Satyendu Panda

USN-1NZ19MCA23

Section-B

Sem-2nd

Department-MCA

Project Title-Music Streaming and Event Booking Application

CHAPTER 1

INTRODUCTION

This chapter is going to provide an introduction about this project. These following topics have been described in this chapter, General information about Project, Project Description, Existing System, Proposed System with Methodology and Review of Literature.

1.1 General Introduction

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.

1.2 Music Streaming and Event Booking Application

In Music Streaming and Event Booking app you can listen multiple languages music and if there any event or music concert you want to go then by through this application you can book any musical event in your area. As It is a mini project, I'm only using core java and database and in frontend I'm using CSS and HTML. It works like if you want to listen your favorite music, when you open this application it will ask you select your preference like which genre of music you like and language also. In event booking it will ask your area where you want to go for any concert or event. If particular concert or event mention in the application then you can book. There will be music icons like. 80S,90S, acoustic, classical, pop etc. and also there will be player icons like. pause, next, loop etc.

1.4 Proposed System with Methodology

- **Sign up-** Before using this application you have to give some details about you when you signing up.
- **Login-** After signing up you have your email id and password that you have to enter in login page.
- **Profile page-** In profile you have your name, email, DOB etc.
- **Add Playlist -**You can make your own playlist
- **Search –** You can search your favourite music to listen and events you want to go.
- **Search Filter-** You can choose different filter like. language, music genre, specific artist and in event booking also like. music event or any other event.
- **My Booking -**All the booking information you done in past all the information will be here.
- **Find Fest –** Here you can find all the event are happening in near future.
- **Icons –** Artist icons, music icons etc.
- **Best of artist –** Here you can find hit music of your favourite artists.

CHAPTER 2

LITERATURE REVIEW

Many of the articles written about music streaming services often refer to them as the modern form of piracy. In the last twenty years, the music industry has gone through many changes due to technological innovations. Technology has had both a negative and a positive effect on the industry. The issue of piracy has existed in the music industry since the beginning, however the nature of it has changed. Piracy as defined by the Oxford English Dictionary refers to 'the unauthorized reproduction or use of an invention or work of another, as a book, recording, computer software, intellectual property, etc.' ('piracy, n.', 2016). Before music was available on CDs, music fans would record their favorite artist's music via cassette tape while the song was being played on the radio.

One of the earliest forms of music piracy occurred when people would produce recordings of songs and then sell the recorded songs on cassettes tapes. Initially when CDs became a popular format for audio files, many CDs were coded to restrict the disc from being played on computers, as this was how users could create pirate CDs. People would rip (upload) the files from the CD onto a computer and then burn (copy) the files onto an empty CD. This became the second generation of music piracy. However, due to advances in technology, it has become easier to acquire music for free online.

As a result, this has led to an increase in piracy. Online music stores such as Apple's iTunes were seen as a necessary evil in the battle against piracy. Users were given the option of buying music cheaper and many in the industry believed this would offer users the option to morally purchase music rather than download it illegally. However, in more recent years many believe that streaming services such as Spotify have become the most recent case of music piracy.

This is because Spotify offers its users a free subscription, thus allowing users to listen to music for free. The way music is distributed has developed in line with technology, however the laws regarding copyright and the payments of artists and producers have not been updated to deal with the addition of music streaming services. Gardner (2015) quotes Maria Pallante, the director of the Copyright Office discussing this issue where she states, 'the structures that evolved in the

previous century to facilitate the lawful exploitation of musical works 4 and sound recordings, while perhaps adequate for the era of discs and tapes, are under significant stress'.

In a report issued by the US Copyright Office, Spotify stated that 'we are competing with piracy, it's a reality that we all face on every level of the ecosystem, we are all competing with free'. This shows that even well-established streaming services are attempting to combat music piracy on a daily basis. The idea of music piracy has been consistently changing alongside the development of new technology. It seems that initially the music industry is untrustworthy of each new technological development until it has existed for a number of years or been overtaken by the next big thing.

The idea of giving music away for free on streaming services has been met with angst from many music artists such as Taylor Swift. However, as part of the artists contract/record deal with their record company they often have to give music away for free. When an album is ready to be sold in stores, record companies make deals with the individual stores to decide on how many albums the store will purchase in order for them to sell on to consumers. In an attempt to get stores to stock more of an album they give away 10% or more of the albums that are ordered for free (Passman, 2015).

These albums do not accumulate any royalties for the artists. Albums which are offered to radio stations and competitions by the artist for the promotion a new album, lead to a loss of royalties received by the artist due to the distribution of music for free (Passman, 2015). Thus, showing that the artist fails to make royalties in many aspects of music publishing and distributing, not just from streaming. Advances in technology have forced the industry to reevaluate its current laws and practices. Each new technological innovation has been met with backlash initially, however it is only when the record companies invest in these new innovations themselves, that they become common use for the industry as a whole. It is clear to see that the music industry is still in control of what happens with its products and artists.

2.1 Spotify

Spotify was founded in Sweden in 2006 and was officially launched as a music streaming platform in 2008. It has since expanded across the world, offering users the opportunity to listen to the music of their favorite artists without needing to own the music themselves. Spotify offers users

based in European countries a paid subscription service of €10 a month. There is also a free subscription service which is offered across the website and tablet versions only. Paid users get added benefits of no advertisements between songs, unlimited skipping through the tracks, the ability to use the app on their mobile devices and also the option to have the music they like available offline.

Although the option of a free service is welcomed by their consumers, it is not appreciated by many music artists as there is controversy over the payment, they receive from Spotify against the payment they would normally receive from the sale of an album. Nonetheless, the record companies still use the service as a method to promote their artists. This chapter intends to highlight the impact which Spotify has had on the industry, its artists and also on the consumer.

2.1.1 The Growing Popularity of Spotify

When Spotify was launched in the UK and US during February 2009, it was seen as an alternative to music piracy by the media. Many articles report that it is difficult to find accurate statistics of the membership numbers of the music streaming sites and every report advocates for a different amount. Swanson (2013) wrote that 'Spotify is the fastest growing music streaming service in the world, with over 24 million users worldwide and nearly 6 million paying between US$5 and $10 a month to use the service'. At the time the service was only four years old, showing great progress for the future. Spotify initially offered users three subscription options, free, unlimited and premium. Unlimited would cost €5 while premium was €10.

The unlimited subscription offer is no longer available on the site, however its main difference from the free version was that it removed advertisements. Nonetheless, it still did not allow offline content to be accessed on any device (Gilmour, 2016). The removal of unlimited may be seen as Spotify wishing for more of their users to pay for the premium service and thus not giving the middle option. Either the users pay to receive unlimited music with no distractions on any device or they must be 8 subjected to advertisements.

In 2009, media sources stated that the public's perception was that Spotify was the 'future of music' (Lewis, 2009). In a more recent article on Spotify, Camp (2015) states that Spotify has 'more than fifty million users through the Americas, Europe and Australasia'. Therefore showing that according to media sources the amount of Spotify subscribers had doubled in two years. This

could be associated with the continued growth of mobile phone technology, which means many consumers may use their mobile data and thus would be able to easily access the mobile application of Spotify.

It is clear to see that Spotify is the most popular streaming service judging by the amount of users its service has attracted (Roettgers, 2015), however there is increased competition from new services such as Tidal and Apple Music. Spotify is popular with users due to its easy to use app and website which are available across many devices. It introduces users to new artists that they may like, in the 'discover weekly' playlist.

This playlist is made up of songs and artists that Spotify thinks the user would like judging by the genres and artists they listen to while using the service (Luckerson, 2015a). Although the service is popular with its users, there has been controversy in the media regarding some artist's relationships with Spotify.

CHAPTER 3

SYSTEM CONFIGURATION

In this chapter tells that, what type of hardware and software system required to do the project.

3.1 Hardware requirements

Processor	Pentium IV
Memory	512 MB RAM
Keyboard	105 Keys
Monitor	CRT or LCD
Mouse	USB Mouse or PS/2 Mouse

Tab. 3.1.1 (Hardware Requirements)

3.2 Software requirements

Operating System	Windows XP/NT
Ide	Eclipse
Database	MySQL
Language	Javafx Programming

Tab. 3.2.1 (Software Requirements)

CHAPTER 4

MODULE DESCRIPTION

4.1 Music Management Module

The Music management Module divided into 3 Parts-(I)Best of Artists (ii)Top Genres (iii) Top Singles. In Best of Artists section all the songs that are popular from that particular artists are there in this section. In Top Genres Section all the song Genres are there like-Pop, EDM, Rock etc.

4.2 Playlist Management Module

In playlist management module a user can make a playlist using a playlist name (ex. Pop song, etc.). After the playlists created the user can add songs to that playlists and they can share the same playlists with your friends. And you can delete that playlists if you want to using delete option.

4.3 Login Management Module

Login Management Module manages all the login related work. After a user signed up then a unique user id created and it stored in the database. If signed up is successful then the user can login using his email and password. The system identifies the specific user using that unique user id. If the user enters two different email ids like while signing up one email id and while login using another email id then the application through exception like email not correct etc.

4.4 Album Management Module

In Album management module the admin can upload the albums and can alter them. When a user listing to this album he can got know the album name and the artists name when that album created and all the information. The album divided into three different sections I.e. Best artists, Top Genres, Top singles.

4.5 Event Management Module

Event management apps come in all shapes and sizes—and promise all sorts of successes. Here, we're showcasing apps that save you time and energy when managing logistics, deadlines, and attendee engagement. The event management apps listed here are a combination of companion tools that can be used together for ultimate productivity as well as end-to-end event planning software.

4.6 User Management Module

Play mode Use a component called Radio Group which has the function of the single item choice. There is a Radio Button in Radio Group. Multiple Radio Button items can only select one; play mode of the player includes single cycle, random play, play in order etc. Media Player have a monitor, which ensures the songs' playing and complete. When songs played complete, method OnCompletionListener will be triggered. In the method the operation after completion of play will be processed.

CHAPTER 5

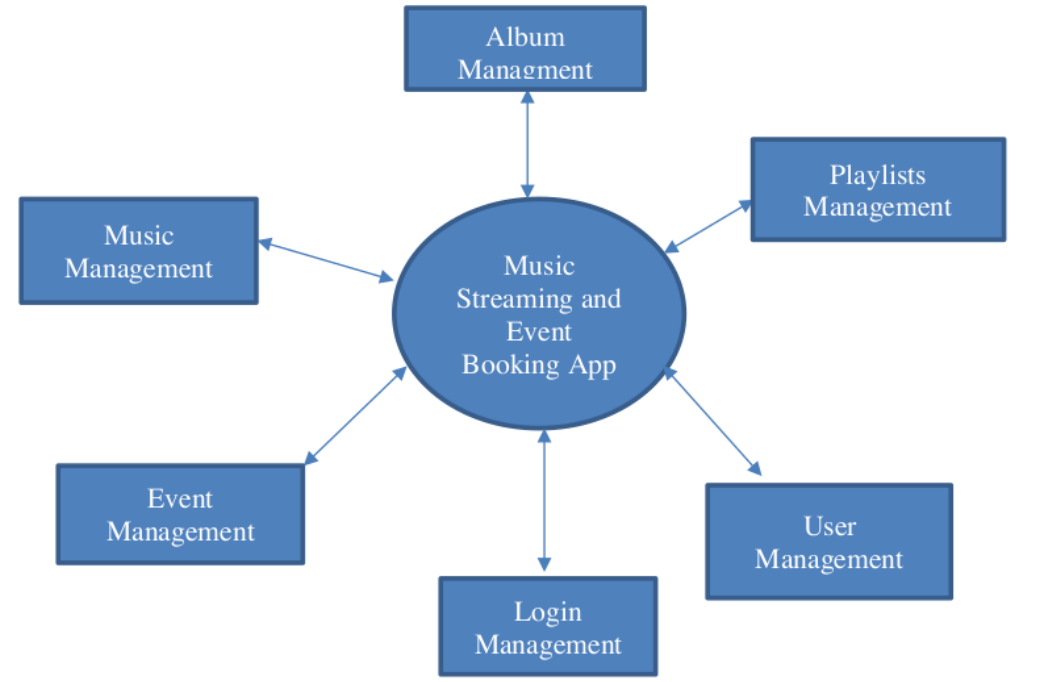
SYSTEM DESIGN

The System Design Document is a required document for every project. It should include a high-level description of why the System Design Document has been created, provide what the new system is intended for or is intended to replace and contain detailed descriptions of the architecture and system components.

5.1 DATA FLOW DIAGRAM

A data Flow Diagram or a bubble chart is a graphical tool for future analysis. DFD modules a system by using external entities from which data flows to a process, which transforms the data and creates output data flows which go to other process or external entities or files. Data in files may also flow to processes as inputs.

FLOW DIAGRAM



Zero Level DFD-MS&EBA

Fig.5.1.1 DFD-1

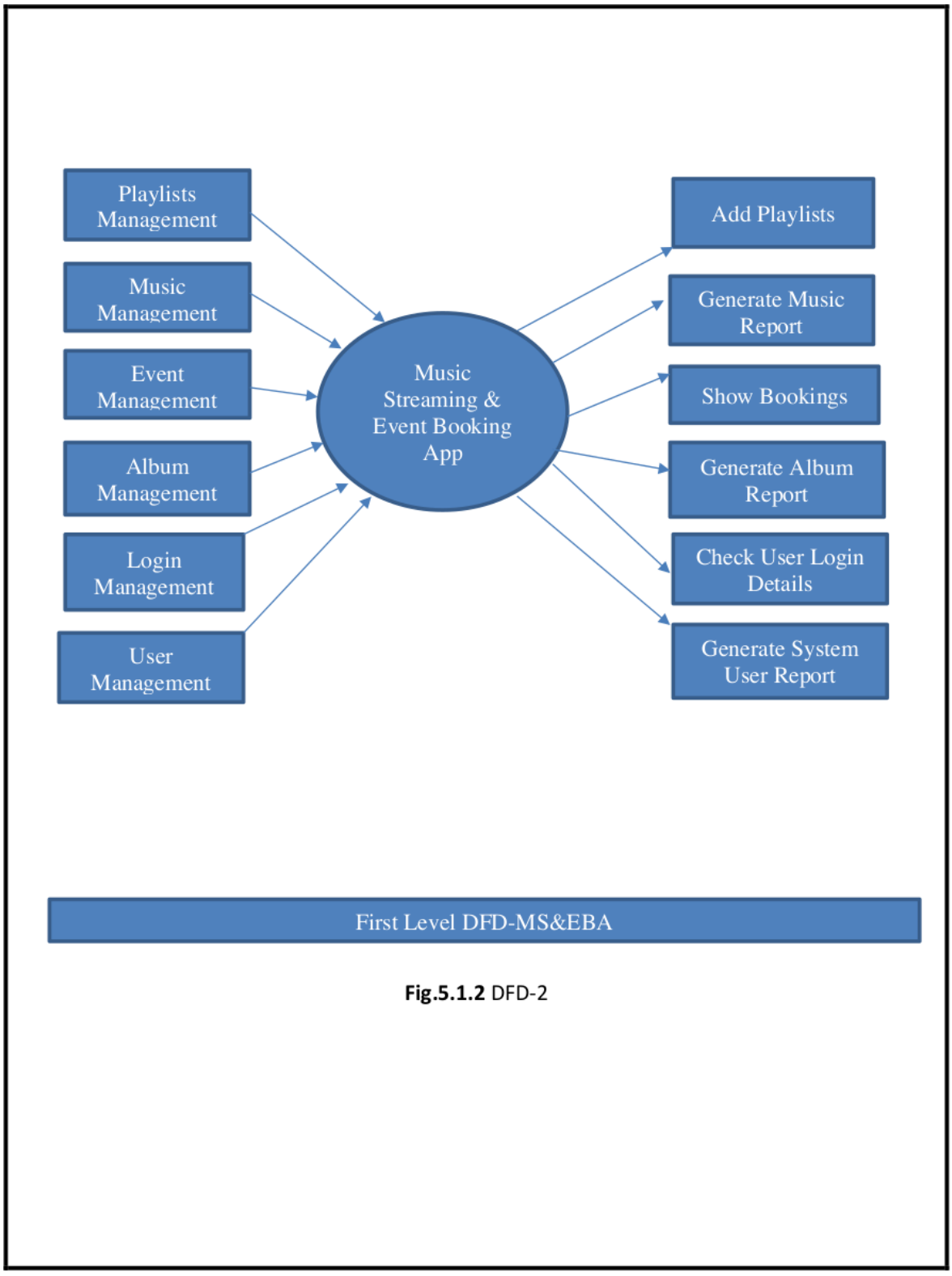

Playlists Management
Music Management
Event Management
Album Management
Login Management
User Management
Music Streaming & Event Booking App
Add Playlists
Generate Music Report
Show Bookings
Generate Album Report
Check User Login Details
Generate System User Report
First Level DFD-MS&EBA

Fig.5.1.2 DFD-2

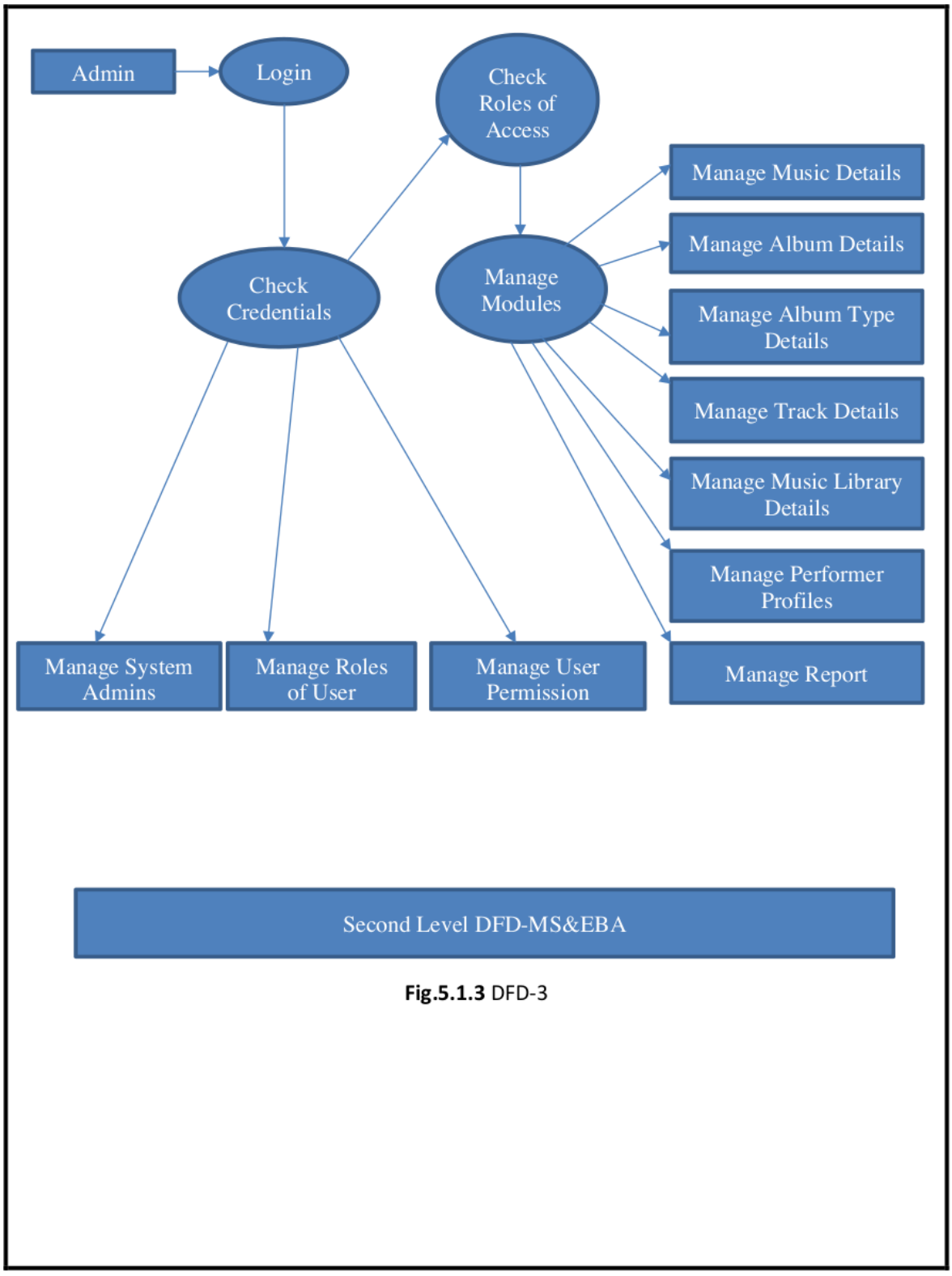

Admin
Login
Check Roles of Access
Manage Music Details
Manage Album Details
Check Credentials
Manage Modules
Manage Album Type Details
Manage Track Details
Manage Music Library Details
Manage Performer Profiles
Manage System Admins
Manage Roles of User
Manage User Permission
Manage Report
Second Level DFD-MS&EBA

Fig.5.1.3 DFD-3

Chapter 6

SYSTEM IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The system can be implemented only after through testing is done and if it is found to work according to the specification. The implementation phase comprises of several activities. The hardware and software requisition are carried out. Implementation of a computer system is by replacing a manual system. The problem encountered are converting files, training users, creating accurate files, and verifying printouts for integrity.

6.1 Implementation

User.java

```java
package Core;
import java.text.DateFormat;

public class User{
    private int user_id;
    private String name;
    private String country;
    private String email;
    private String DOB;
    private String gender;
    public User(int user_id, String name, String country, String email, String DOB, String gender){
        this.user_id = user_id;
        this.name = name;
        this.country = country;
        this.email = email;
        this.DOB = DOB;
        this.gender = gender;

    }
    public void checkFormatting() throws MyException {
        String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\."+ "[a-zA-Z0-9_+&*-]+)*@" + "(?:[a-zA-Z0-9-]+\\.)+[a-z" + "A-Z]{2,7}$";
        Pattern pat = Pattern.compile(emailRegex);
        if(!pat.matcher(email).matches()) throw new InvalidEmailException("Enter a correct email address");

        try {
            DateFormat df = new SimpleDateFormat("yyyy-mm-dd");
            df.setLenient(false);
            df.parse(DOB);
        }
        catch (ParseException e){
            throw new IncorrectDateException("Enter a valid date");
        }

    }
    public void display(){
        System.out.println(user_id + " " + name + " " + country + " " + email + " " + DOB + " " + gender);

    }
    public int getAge(){
        return Calendar.DATE;
    }

    public int getUser_id() {
```

Fig 6.1.1 USER Login

(Before using this application, user must login into this application giving their credentials like. Email, name, Country, unique user_id, DOB, Gender etc. After giving all the credentials then it checks all the credentials are valid or not if the credentials are valid then you successfully registered)

UserAuth.java

```java
package Core;
import java.util.regex.Pattern;


public class UserAuth{
    private int user_id;
    private String password;

    public UserAuth(int user_id,String password){
        this.user_id = user_id;
        this.password = password;
    }

//    Be between 8 and 40 characters long
//    Contain at least one digit.
//    Contain at least one lower case character.
//    Contain at least one upper case character.
//    Contain at least on special character from [ @ # $ % ! . ].

    public void checkFormatting() throws  MyException{
        String passRegex = "((?=.*[a-z])(?=.*\\d)(?=.*[A-Z])(?=.*[@#$%!]).{8,40})";
        Pattern pat = Pattern.compile(passRegex);
        if(!pat.matcher(password).matches()) throw new InvalidPasswordException("Enter a correct password");

    }

    public int getUser_id() {
        return user_id;
    }

    public String getPassword() {
        return password;
    }

}
```

Fig 6.1.2 USER Authorization

(In user authorization, after user successfully registered then the user has to login using the Email and Password if the credentials are correct then it shows you the home screen)

UserPlaylist.java

```java
package Core;

public class UserPlaylist {
    private int user_id;
    private int playlist_id;
    private String name;
    private int song_num;

    public UserPlaylist(int user_id, int playlist_id, String name, int song_num){
        this.user_id = user_id;
        this.playlist_id = playlist_id;
        this.name = name;
        this.song_num = song_num;
    }

    public int getPlaylist_id() {
        return playlist_id;
    }

    public int getSong_num() {
        return song_num;
    }

    public int getUser_id() {
        return user_id;
    }

    public String getName() {
        return name;
    }
    public void display(){
        System.out.println(playlist_id + " " + user_id + " " + name + " " + song_num);
    }
}
```

Fig 6.1.3 USER Playlists

(In user playlists the user can create playlist giving user_id, playlist_id, name and song_num. After giving all the details the user can add any kind of songs in that playlists and listen to it)

DataController.java

```java
package Database;
import Core.*;

public class DataController {
    public Connection connection;

    public void addToPlaylist(int playlist_id, int song_id) throws MyException{

        try {
            System.out.println(playlist_id + " " + song_id);
            DBConnection con = new DBConnection();
            connection = con.getConnection();
            if(connection == null) throw new ConnectionInvalidException("Connection not Estabilised");
            Statement stmt = connection.createStatement();
            String query = "INSERT INTO PLAYLIST_SONGS VALUES (" + playlist_id + "," + song_id + ")";
            stmt.executeUpdate(query);
            System.out.println("done");
        }
        catch (SQLException e){
            e.printStackTrace();
        }

    }

    public static ArrayList<Advertisement>  ads = new ArrayList<Advertisement>();
    public int getSongID(String name) throws MyException{
        int songID=0;
        String songIDString="";

        try {
            DBConnection con = new DBConnection();
            connection = con.getConnection();
            if(connection == null) throw new ConnectionInvalidException("Connection not Estabilised");
            Statement stmt = connection.createStatement();
            String query = "select song_id from ALL_SONGS WHERE title = '" + name + "'";
            ResultSet rs = stmt.executeQuery(query);
            while(rs.next()) {
                songIDString = rs.getString(1);
            }
            songID = Integer.parseInt(songIDString);

        }
        catch (SQLException e){
```

Fig 6.1.4 Data Controller

(In Data Controller, the admin manages all the modules for ex. To add songs to a playlist it uses the playlist_id and song_id to control the playlist module. Also, to control the module and adding data to database some SQL Query uses in this Data Controller)

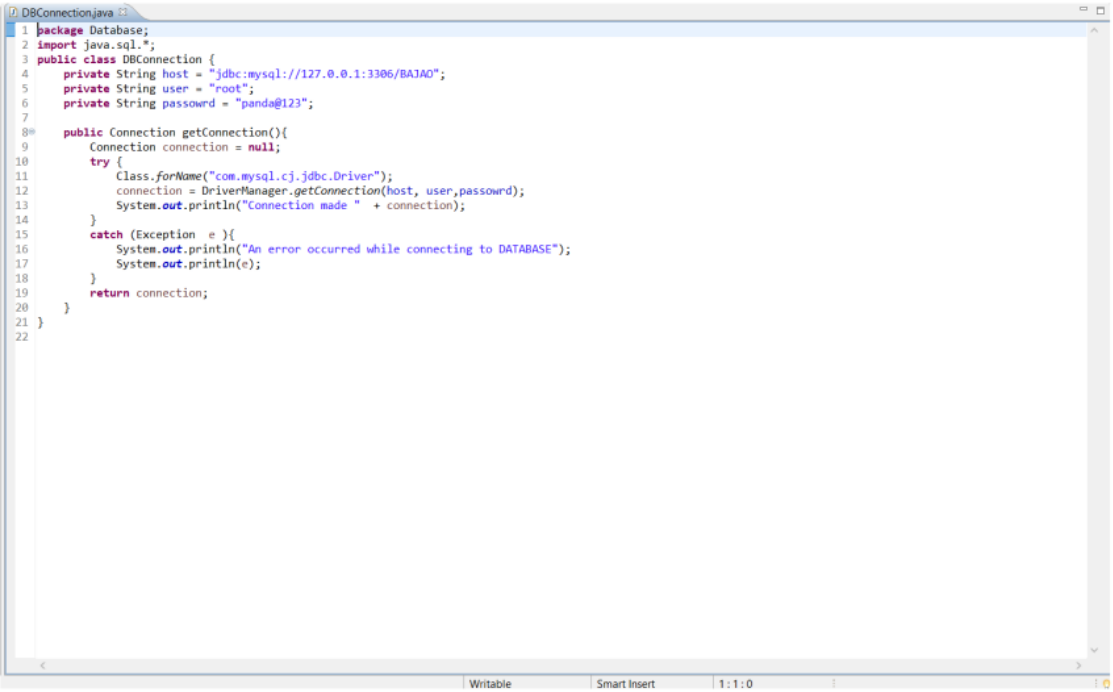


Fig 6.1.5 Database Connection

(For database connectivity it required the host, user and password of that database. My database name is “BAJAO”. Before running this application, we have to connect the application to the database. If it not connected to the database it should showing some errors like “An error occurred while connecting to DATABASE”)

ConnectionInvalidException.java

```java
package Exception;

public class ConnectionInvalidException extends MyException{
    public ConnectionInvalidException(String message) {
        super(message);
    }
}
```

Writable | Smart Insert | 8 : 1 : 169

Fig 6.1.6 Connection Invalid Exception

If any kind of exception occurs during connection then throw some messages and it extends MyException class.

MainScreenController.java

```java
package Ui.MainPage;

import Core.SaveFileCache;
public class MainScreenController implements Serializable {
    public Stage mainPageWindow;
    public Scene scene;
    public Button userProfileButton;
    public ImageView userProfilePic;
    public Button homeButton;
    public Button searchButton;
    public Button yourLibraryButton;
    public Button addPlaylistButton;
    public Button yourPlaylistsButton;
    private static User user;
    private SaveFileCache data;

    @FXML
    ImageView a1, a2 ,a3 ,a4, a5, a6, g1, g2 ,g3 ,g4 ,g5, g6, s1 ,s2, s3, s4 ,s5 ,s6;

    public  void loadWindow(User user, String password){
        this.user = user;
        try {
            data = new SaveFileCache(user.getEmail(),password);
            mainPageWindow = new Stage();
            Main.window.close();
            Parent mainPage = FXMLLoader.load(LoginBox.class.getResource("/Ui/MainPage/MainScreen.fxml"));
            mainPageWindow.initModality(Modality.APPLICATION_MODAL);
            scene = new Scene(mainPage);
            mainPageWindow.setScene(scene);
            userProfileButton = (Button)scene.lookup("#userProfileButton");
            userProfilePic = (ImageView)scene.lookup("#userProfilePic");
            homeButton = (Button)scene.lookup("#homeButton");
            searchButton = (Button)scene.lookup("#searchButton");
            yourLibraryButton = (Button)scene.lookup("#yourLibraryButton");
            addPlaylistButton = (Button)scene.lookup("#addPlaylistButton");
            yourPlaylistsButton = (Button)scene.lookup("#yourPlaylistsButton");
            setCredentials();
            mainPageWindow.setTitle("Profile");
            mainPageWindow.setResizable(false);
            mainPageWindow.show();
            mainPageWindow.setOnCloseRequest(e->{
                try {
                    throw new SaveStateException();
```

Fig 6.1.7 Main Screen Controller

These snippets of code responsible for Home Screen interface like buttons, image etc.

LoginSignupController.java

```
package Ui.SignupLoginMain;
import Core.User;

public class LoginSignupController {

    public Stage window;

    public  Button mainlogin;
    public Button mainsignup;
    public TextField signupname;
    public TextField signupusername;
    public TextField signupemail;
    public TextField signuppassword;
    public TextField signuppassword1;
    public MenuButton signupmonth;
    public TextField signupyear;
    public TextField signupdate;
    public TextField signupcountry;
    public Button signupbutton;
    public RadioButton gendermale;
    public RadioButton genderfemale;
    public RadioButton genderother;
    public Text a1;
    public Text a2;
    public Text a3;
    public Text a4;
    public Text a5;
    public Text a6;
    public Text a7;
    public Text a8;
    public Button adminButton;
    public  void handlelogin(){
        try{
            LoginBox.getLogin();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }

    public void handleAdminButton(){
        try{
            MasterController mc = new MasterController();
```

Fig 6.1.8 Login Signup Controller

In Login Signup Controller, all the details required while login and signup are representing here.

6.2 SCREEN SHOTS

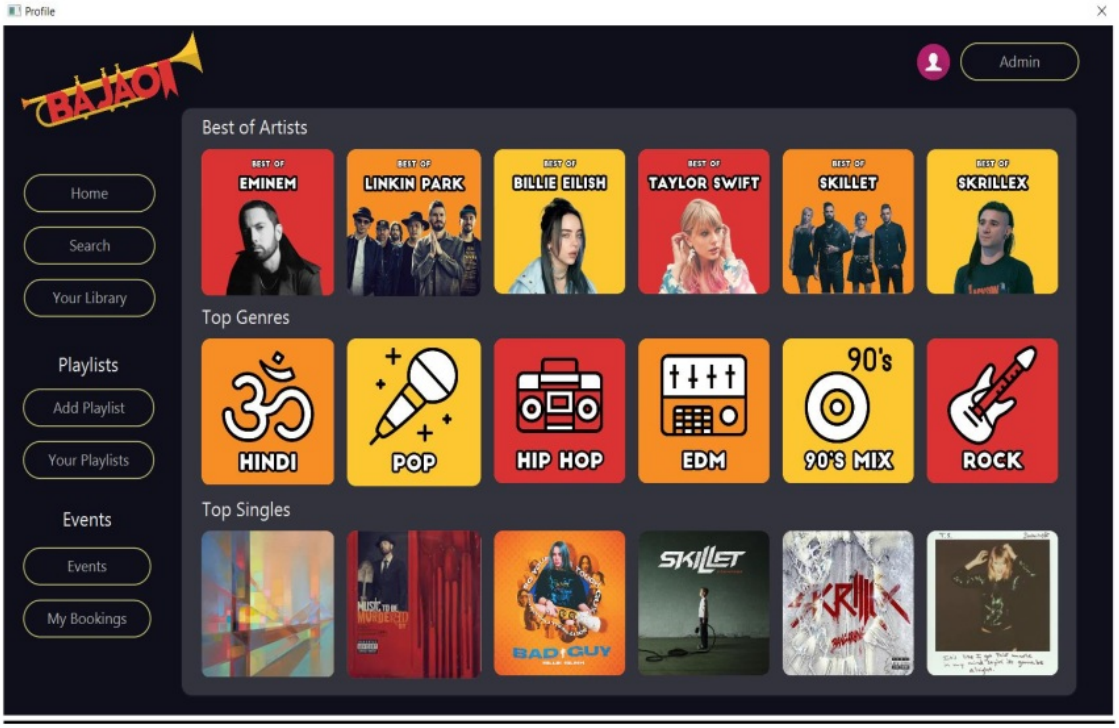


Fig. 6.2.1 Homepage

(In Homepage, there are search options to search your favorite songs also there is playlist option you can create playlist and add songs and there are also event options you can book musical event or concert by using this application. You can edit your admin details in the admin option in top right corner. And there is also best of artists, Top Genres and Top singles in Home Screen)

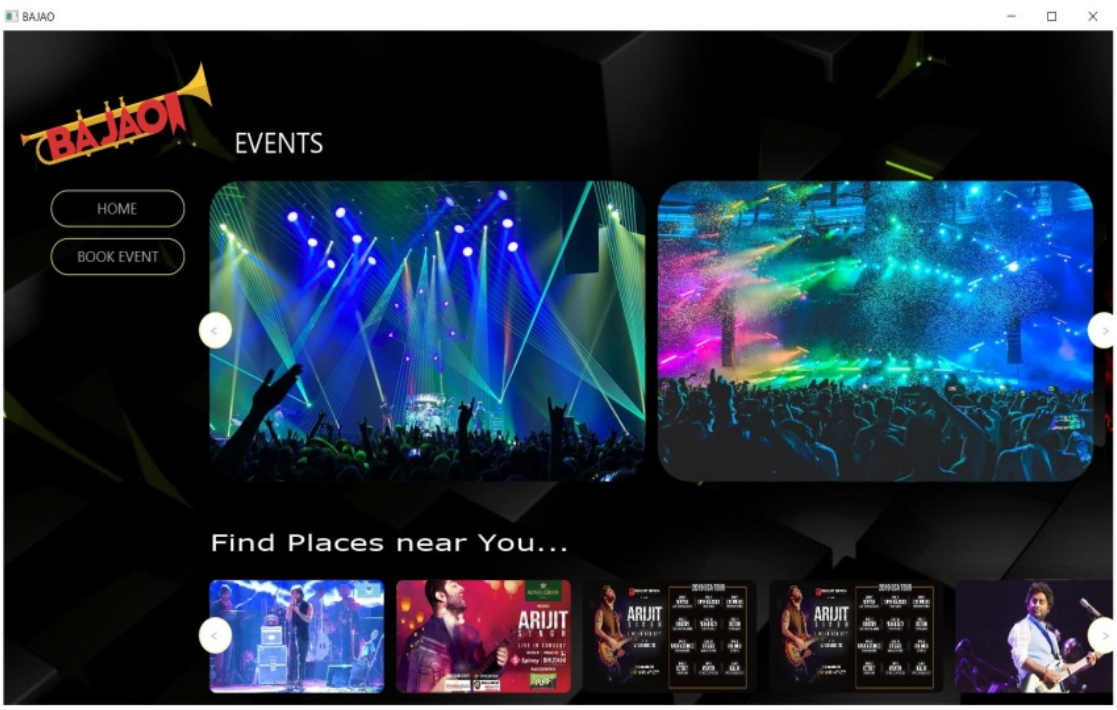


Fig 6.2.2 Events

(Events like music concert etc. you can book any musical event using this application and it also show some music concert happening in near future)

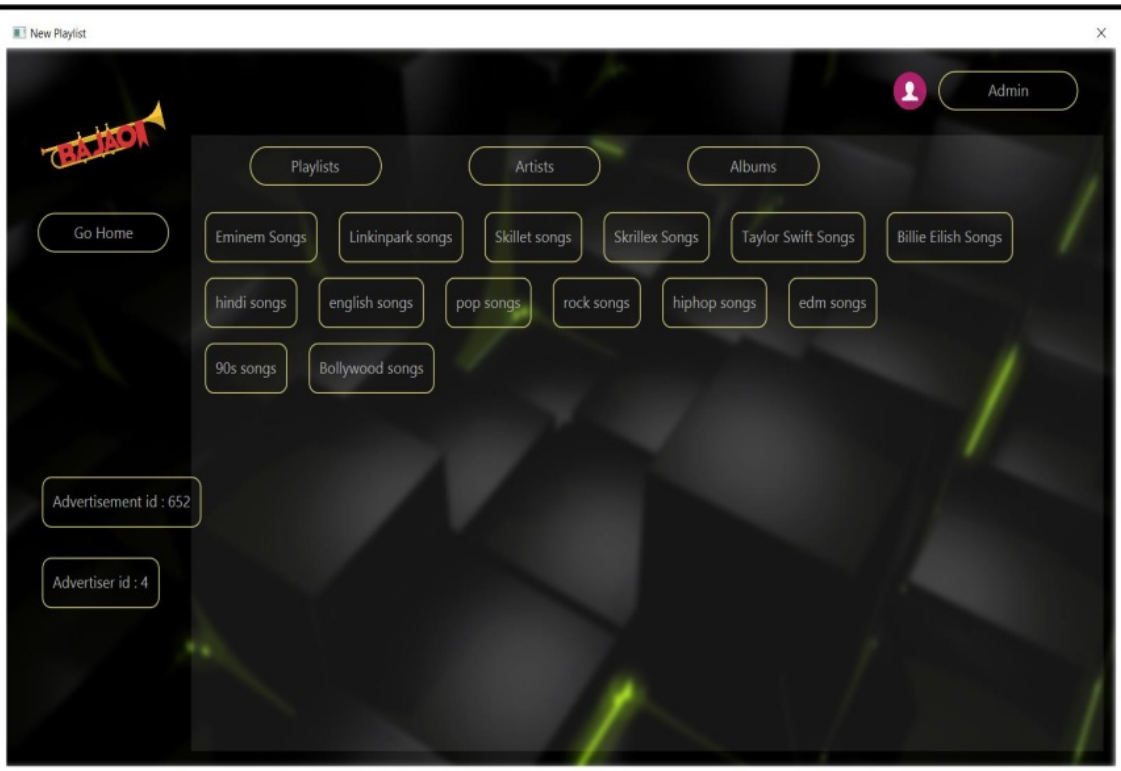


Fig. 6.2.3 Playlists

(You can create playlists and add songs to it. While creating playlists you can give any name like your favorite artist name ex. Eminem Songs etc. also you can advertise your product in this application. Admin can give permission by giving you Advertisement id and Advertiser id)

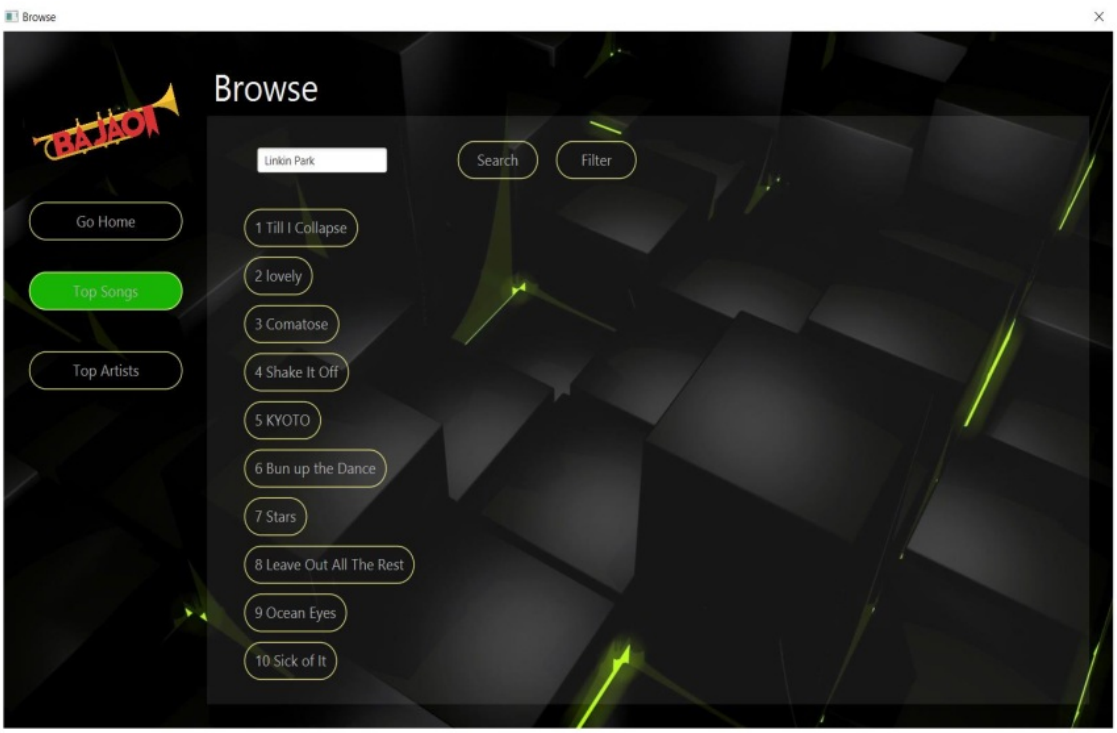


Fig. 6.2.4 Top Songs

(In Top Songs option you can search any songs using any artists name or songs name and there is also a filter using this filter you can search your songs like filter by songs, artists, Genres, Playlists)

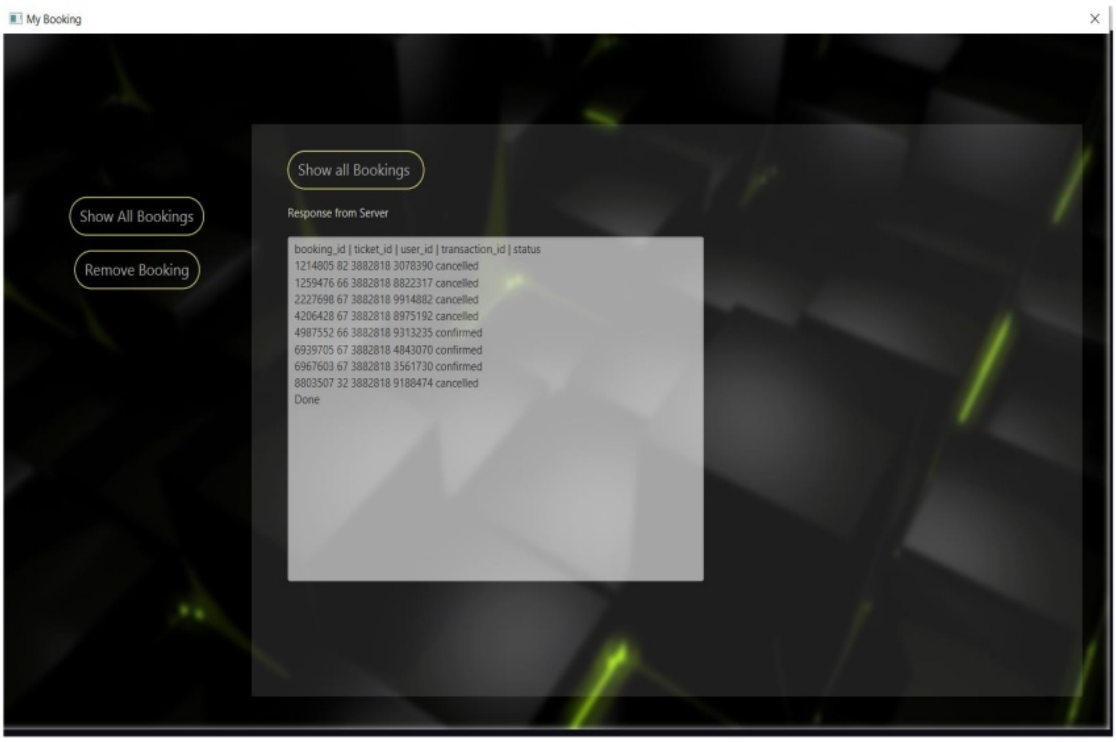


Fig. 6.2.5 Booking Details

(If user book any event using this application then the details store like booking_id, ticket_id, user_id, transaction_id and status for example if user succesfully booked a ticket then the details store in the backend like booking_id – 1214805, ticket_id – 82, user_id – 3882818, transaction_id – 3078390 and status – confirmed or cancelled)

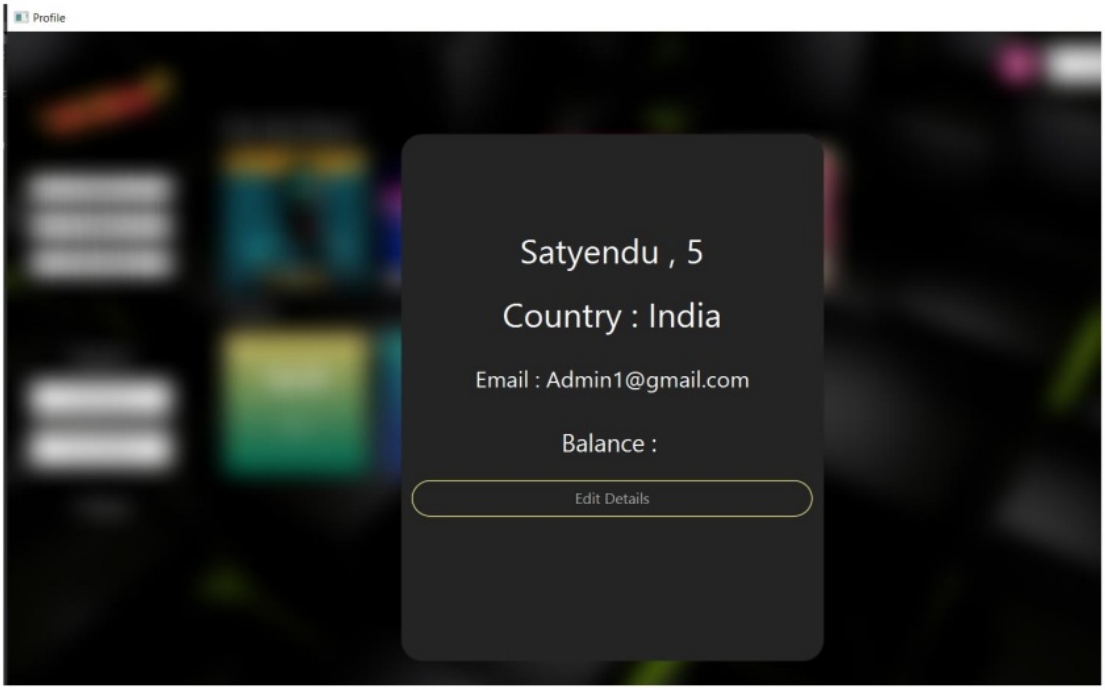


Fig 6.2.6 Edit Users Details

(Here ADMIN can edit the details like country, name, admin number, email, how much balance left etc.)

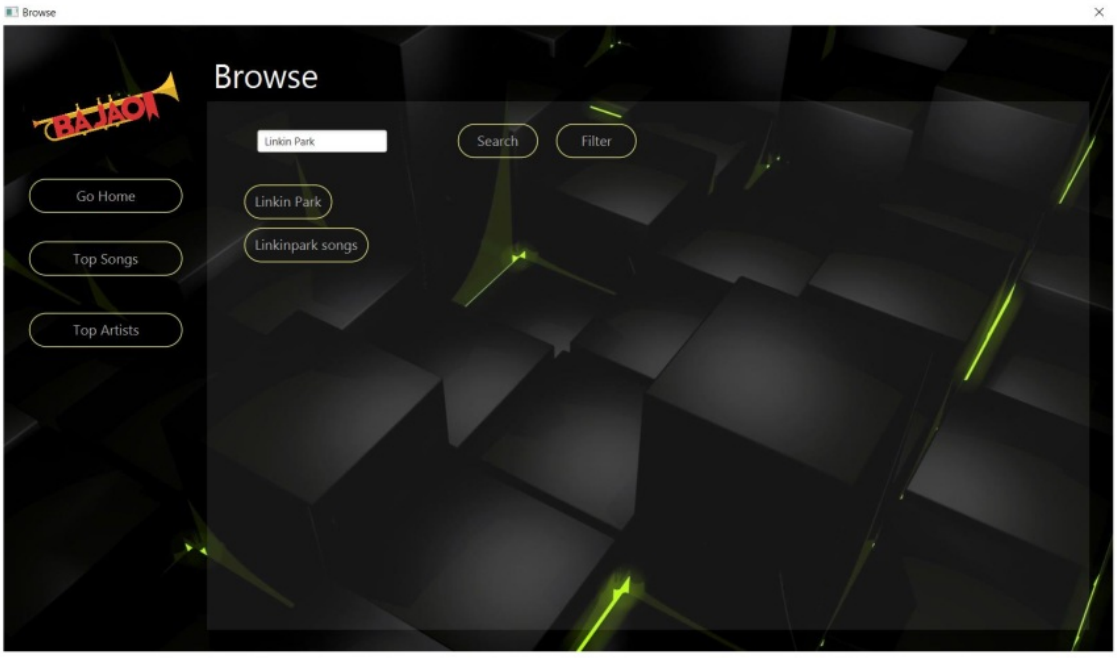


Fig 6.2.7 Search

(In Search box you can search your favorite songs or artists name like in the above screen shot you can see I'm searching an artist name i.e. Linkon Park then in result it showing me all the linkon Park songs)

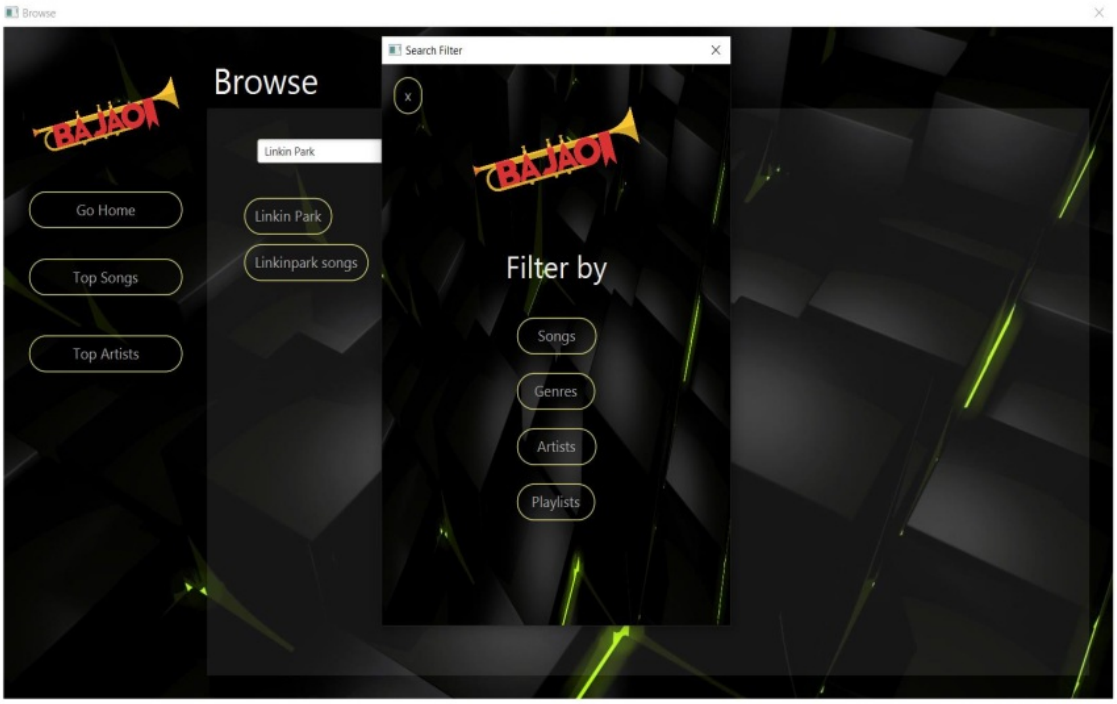


Fig 6.2.8 Filter

(Here You can filter your search by songs, Genres, Artists, Playlists)

CHAPTER 7

SYSTEM TESTING

Types of testing: -

1. Unit Testing

It focuses on smallest unit of software package style. during this we have a tendency to take a look at a personal unit or cluster of interconnected units. it's usually done by computer user by exploitation sample input and observant its corresponding outputs.

2. Integration Testing

The objective is to require unit tested parts and build a program structure that has been determined intentionally. Integration testing is testing within which a gaggle of parts ar combined to supply output.

(a) Black Box Testing: it's used for validation. during this we have a tendency to ignore internal operating mechanism and target what's the output?

(b) White Box Testing: it's used for verification. during this we have a tendency to focus om internal mechanism i.e. however the output is achieved?

3. Regression Testing

Every time new module is additional results in changes in program. this sort of testing makes positive that whole part works properly even when adding parts to the entire program.

4. Alpha Testing

This is a kind of validation testing. it's a kind of acceptance testing that is finished before the merchandise is discharged to customers. it's usually done by QA folks.

5. Beta Testing

The test is conducted at one or additional client sites by the end-user of the software package. This version is discharged for the restricted variety of users for testing in real time surroundings.

6. System Testing

In this software package is tested such it works fine for various OS. it's coated underneath the recording equipment testing technique. during this we have a tendency to simply target needed input and output while not that specialize in internal operating.

In this we've security testing, recovery testing, stress testing and performance testing.

7. Stress Testing

In this we have a tendency to provide unfavorable conditions to the system and check however they perform in those condition.

8. Performance Testing

It is designed to check the run-time performance of software package inside the context of Associate in Nursing integrated system. it's wont to take a look at speed and effectiveness of program.

Chapter 8

RESULTS AND DISCUSSIONS

8.1 Conclusions

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+ Share Preference configuration file. The system realized the music player programming.

8.2 Future enhancements

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.

Music Streaming and Event Booking Application

ORIGINALITY REPORT

0%	%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Sozer, . "Delivering Effective Training", DNA Analysis for Missing Person Identification in Mass Fatalities, 2014. Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		