S.P.MANDALI’ S

**RAMNARAIN RUIA AUTONOMOUS COLLEGE**

Matunga, Mumbai, - 400019

Project Report

On

“Online Music Library Android Application”

Under the Guidance of

“Prof. Rasika Mundhe”

Project By

“Hemantkumar Vijaykumar Harijan”

TYBSC

Department of Computer Science

2017-18

Acknowledgment

* I Mr.Hemantkumar Vijaykumar Harijan, student of Ramnarain Ruia Autonomous College acknowledges my sincere gratitude to all those who helped me in my project work of the T.Y.Bsc(Computer Science).
* I would like to thank my project guide, Prof. Rasika Mundhe to help me to go ahead with my project and appreciating my work at every stage.
* I would like to thank all my teachers for their suggestion and idea throughout the development of this project.
* Thanks to entire non-teaching staff for complying with all needs and requests and troubleshooting technical problems.
* Convey my sincere thanks to all my classmates, seniors, friends for always being there, boosting my confidence and for their timely help.

Sincere Thanks From,

Mr. Hemantkumar Vijaykumar Harijan

Index

|  |  |  |
| --- | --- | --- |
| Sr No | Contents | Page No. |
|  |  |  |
|  | **Preliminary Investigation** |  |
| 1 | Current System | 4 |
| 2 | Limitations of Current System | 4 |
| 3 | Proposed System | 4 |
| 4 | Advantages of Proposed System | 4 |
| 5 | Feasibility | 5 |
| 6 | Stakeholders | 4 |
| 7 | Project Requirements | 5 |
| 8 | Software Requirements | 5 |
| 9 | Technologies Used | 5 |
| 10 | Gantt Chart | 6 |
|  |  |  |
|  | **System Analysis and Design** |  |
| 11 | Event Table | 10 |
| 12 | Class Diagram | 11 |
| 13 | Use Case Diagram | 12 |
| 14 | Use Case Scenario | 13 |
| 15 | Sequence Diagram | 19 |
| 16 | Activity Diagram | 24 |
| 17 | State Chart Diagram | 29 |
|  |  |  |
|  | **System Coding** |  |
| 18 | List of Tables with Attributes and Constraints | 31 |
| 19 | Screen Layouts | 32 |
|  |  |  |
|  | **System Implementation and Coding** |  |
| 20 | Client Side Code | 35 |
| 21 | Server Side Services | 123 |
|  |  |  |
| 22 | **Future Enhancements** | 134 |
| 23 | **References** | 136 |

**Preliminary Investigation**

Description of current system

* Current system is the world's largest music and audio streaming platform with a buzzing community of artists and musicians constantly uploading new music.
* Current system is where you can find the next big artists alongside chart-topping albums, live sets, and mixes for every occasion.
* It describes enormous catalogue lets you discover amazing hip hop, electronic, rock, classical, jazz, podcasts, audiobooks sports and much more.
* Get suggested tracks based on your likes and listening habits.
* Create playlist for parties, workouts, on the way to work, or whatever mood you're in.

Drawbacks of the current system

* We cannot use the music for the mass production or reproduction purpose, Because of the copywrite rules.

The proposed system

* The proposed system is to give a platform were users can upload their own creations and sell it at any cost they want.
* The other user can use the uploaded music for their own purpose without any copywrite issue.

Advantages

* Open source.
* No issue related to ownership of the audio file.

Stake Holders

Users

* Users can view their account, upload, download, sell & buy audio files.
* Explore new instrumental music in various genre.

Feasibility

Technical:

* The project is feasible as there isn't any difficulty to get required resources.

Economically:

* The project does not need much money for development.

Operational:

* The System required minimum knowledge of the mobile phones.
* The system can be easily used by the system.

Project Requirements

Hardware Requirement

* Minimum 4GB RAM.
* Intel Dual core 2.2 GHz processor.
* At least 1 GB secondary storage.

Software Requirements

* OS - Windows 7,8,8.1&10.
* Android Studio 3.0 with JAVA JDK 1.8.0\_131.
* Mysql, sql front.
* TOMCAT Server.

Gantt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Phase Title | June | | | | | July | | | | | | August | | | | | September | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| Project Approval |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Preliminary Investigation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System Analysis and Design | | | | | | | | | | | | | | | | | | | | | |
| Event Table |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Class Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Use Case Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Use Case Specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Sequence Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Activity Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| State Chart Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Phase Title | October | | | | | November | | | | | December | | | | | Januray | | | | | February | | | |
|  | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 |
| Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Actual Time of Completion |  | Expected Time of Completion |  | Break |

**System Analysis**

EVENT TABLE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | User Registration | Sign up | User | Register  New Account | Account created | System |
| 2 | Login User | Request to login | User | Verify Authentication | Login | System |
| 3 | Search Music | Search requested music | User | Music Searched | Display | System |
| 4 | Play audio file | Request to Play | User | Audio file Running | Audio file played successfully | System |
| 5 | Upload audio file | Select audio file | User | Uploading | Upload successful | System |
| 6 | Download audio file | Select audio file | User | Downloading | Download successful | System |
| 7 | Remove audio file | Select audio  File | User | Remove in process | Remove  Successful | System |
| 8 | Delete Account | Request to delete account | User | Delete | Account  Deleted | System |
| 9 | Feedback | Request for feedback form | User | Display  Feedback form | Feedback  Details | User |

Use Case Scenario

* Use case name: To Register new account.

Actor: User.

Flow of events:

1)The use case begins with a mandatory Registration form for all the new users.

2)The system displays a registration form.

3)The users enter the details in the Registration/Signup form.

It includes: Name, Age, User\_id &Password.

4)The user creates a new Account.

Pre-conditions:

The Entered details should not match with the other existing user.

Post-conditions:

The user successfully creates a new Account.

* Use case name: Login Account.

Actor: User.

Flow of events:

1)The system displays the login form.

2)The user enters the login details includes:

User\_id & passwords.

3)The system authenticates the entered details if the details are correct else authentication fail.

Pre-condition: User should have an account in the system.

Post-condition: The user successfully login into the Account.

* Use case name: Search.

Actor: User.

Flow of Events:

1)The user searches a music.

2)The system displays or give option for type of music, the user wants to search.

3)The user enters the name and type of music.

4)The system displays the results.

Pre-condition: The user must be login to his/her account.

Post-condition: The system successfully displays the data.

* Use case name: Play Audio.

Actor: User.

Flow of Events:

1)The user chooses the play option.

2)The system loads the music.

3)The system plays the music.

Pre-condition:

The user should have login into his/her account.

Post-condition:

The system plays the music successfully.

* Use case name: Download Audio.

Actor: User.

Flow of Events:

1)The user chooses the download option.

2)If the music audio is free, the user chooses the download option, else the user chooses the Buy option.

Pre-condition: The user should have login into his/her account.

Post-condition: The user download the audio successfully.

* Use case name: Upload Audio.

Actor: User.

Flow of Events:

* The user chooses the upload option.
* The user set the price for music audio or set it free.
* The system uploads the music audio.

Pre-condition: The user must be login into his/her account.

Post-condition: The system successfully uploads the audio.

* Use case name: Remove Audio

Actor: User.

Flow of Events:

1)The user chooses the remove option.

2)The System approve the audio from the account.

3)The system removes the audio from the account.

Pre-condition: The user must be login into his/her account.

Post-condition: Audio removed successfully.

* Use case name: Delete account.

Actor: User.

Flow of Events:

1)The user chooses the delete account option.

2)The user requests the system.

3)The system approves the request.

4)The account gets deleted.

Pre-condition: The user should have an account.

Post-condition: The system delete the account successfully.

9.Use case name: Feedback.

Actor: User.

Flow of Events:

1)The user chooses the feedback option.

2)The user fills the feedback form.

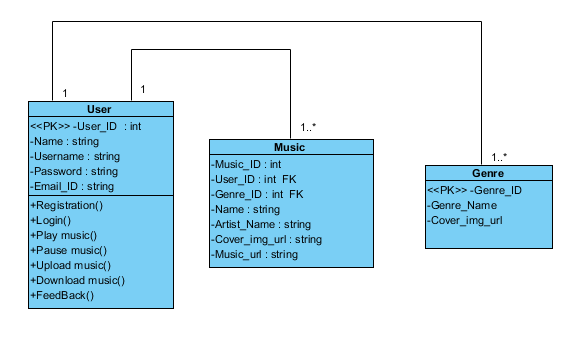
3)The system uploads the feedback form.

Pre-condition: The user must be login into his/her account.

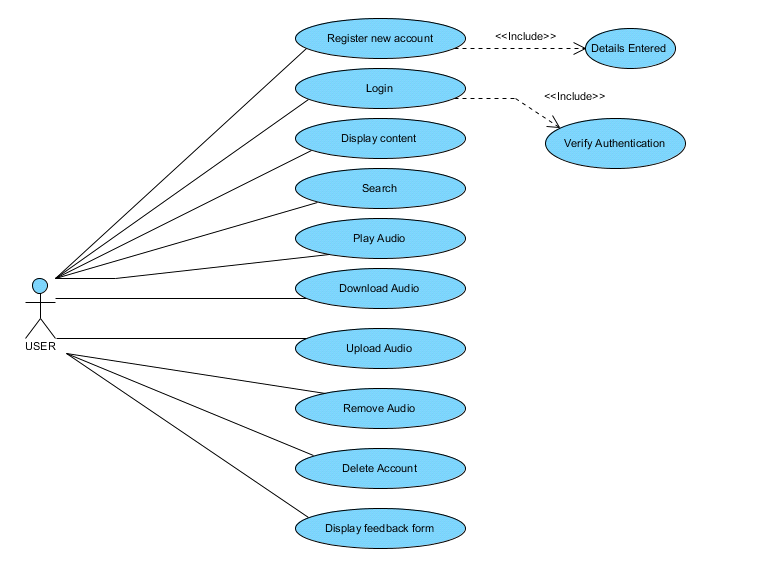
Post-condition: System sends feedback form successfully.

Class diagram

USER

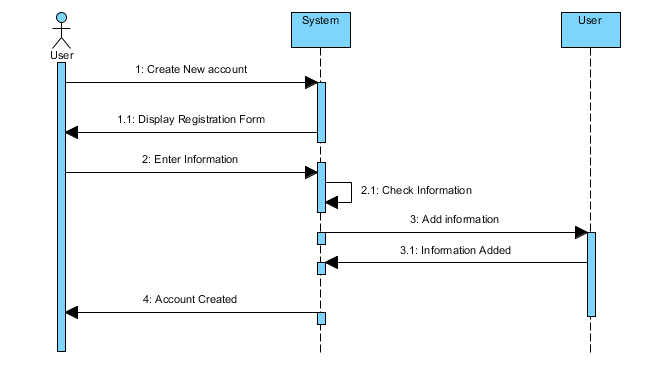


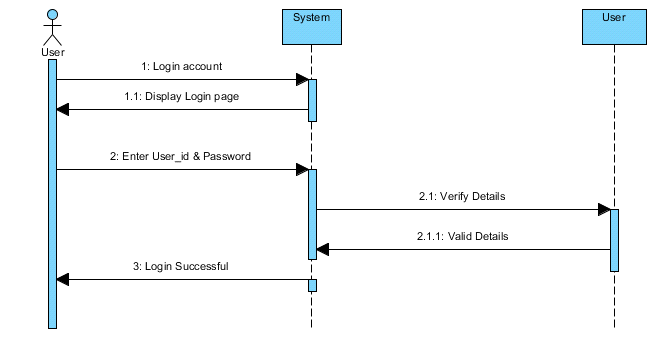
USE CASE DIAGRAM



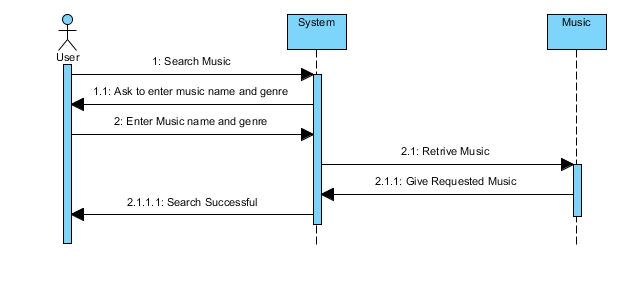
Sequence diagram

1 Register New account

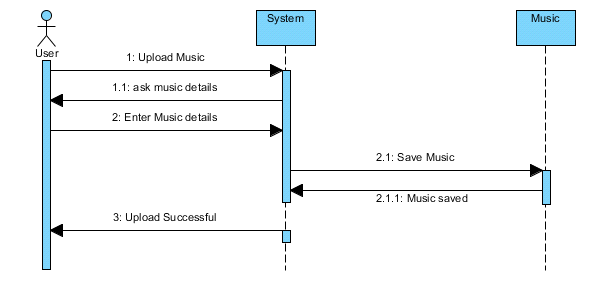


2 Login Account

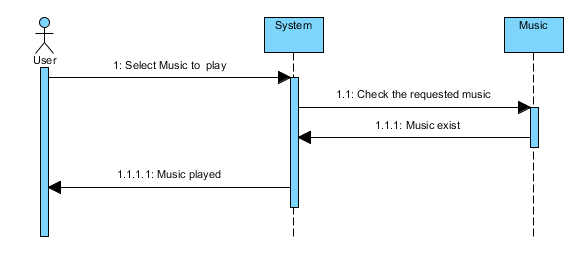
3 Search



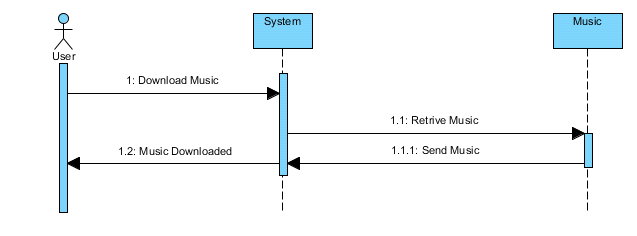
4 Upload



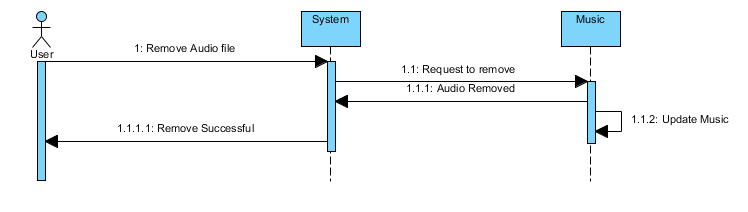
5 Play



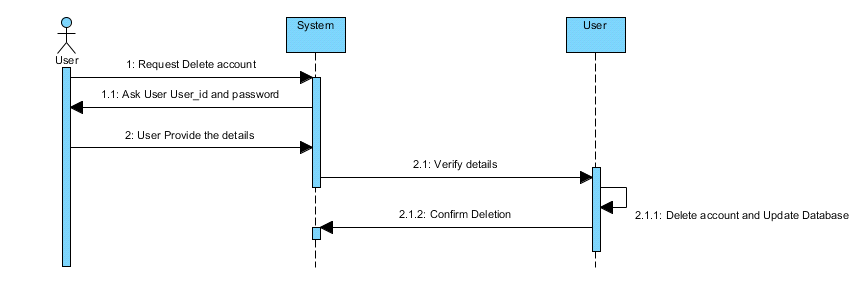
6 Download



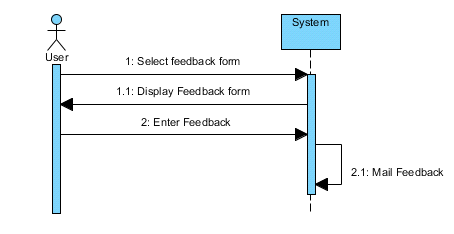
7 Remove



8 Delete Account

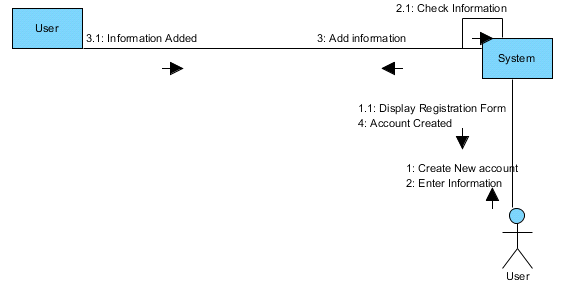


9 Feedback

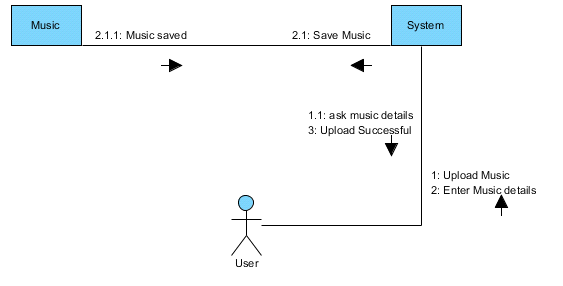


Collaboration Diagram

Create New Account

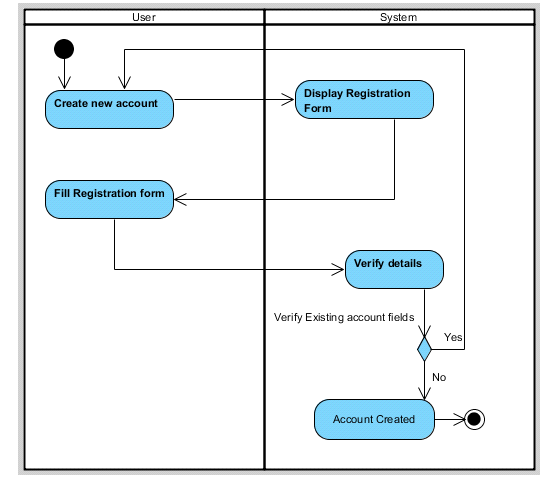


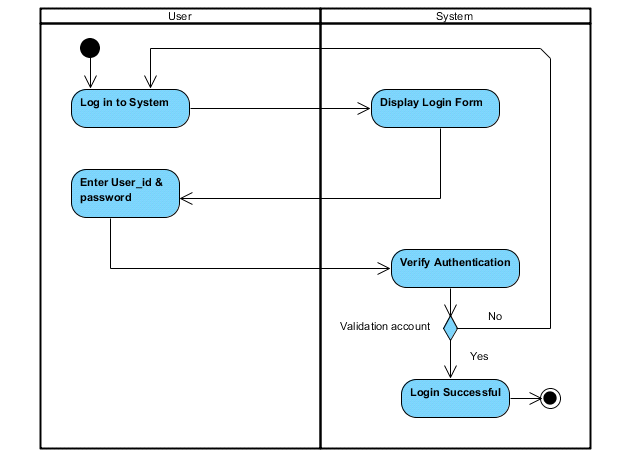
Upload Music



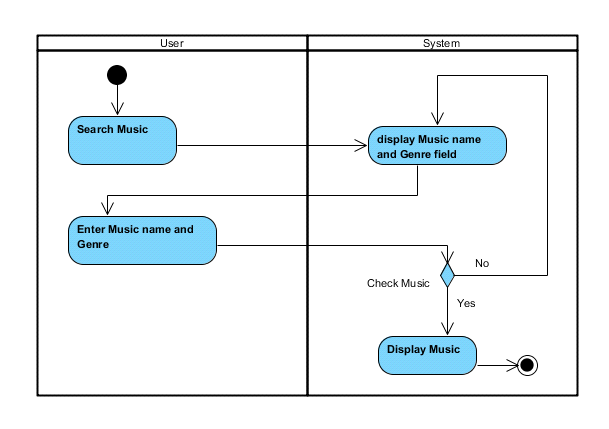
Activity Diagram

1 New Account

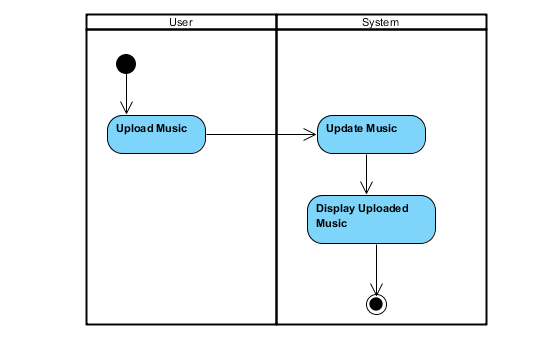


2 Login Account

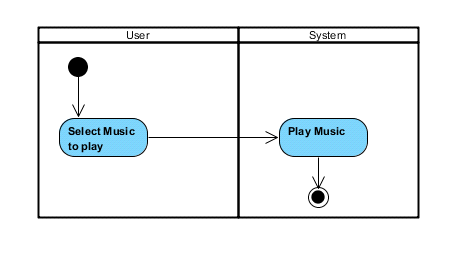
3 Search



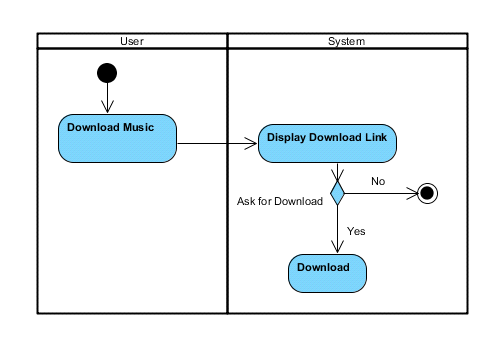
4 Upload



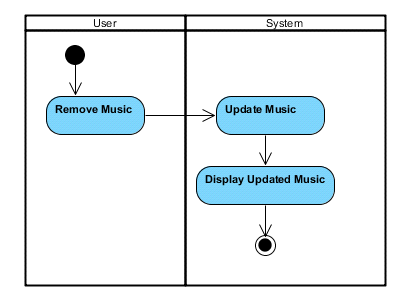
5 Play



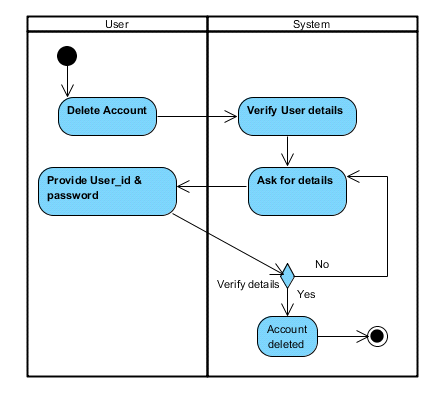
6 Download



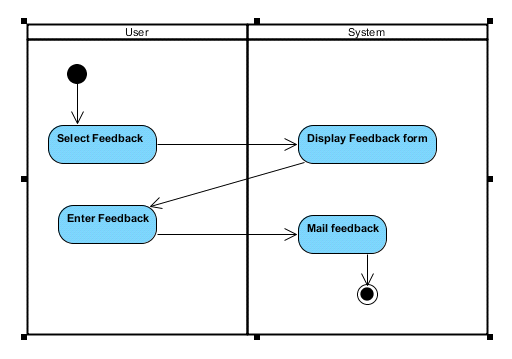
7 Remove



8 Delete account

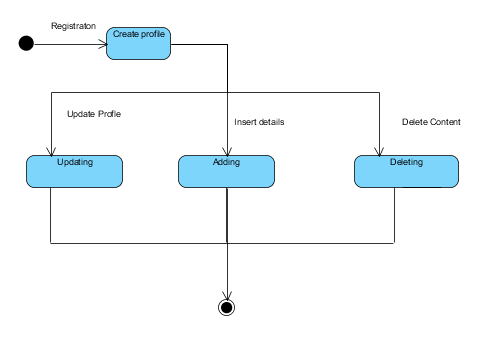


9 Feedback

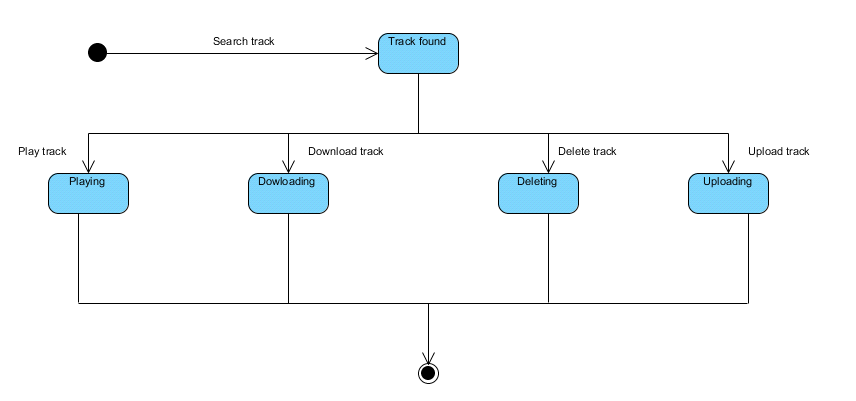


State Chart Diagram

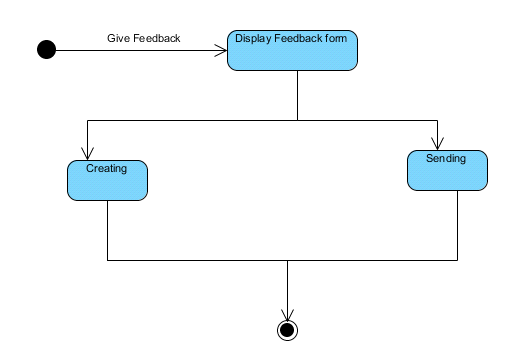
User



Music



Feedback



**System Coding**

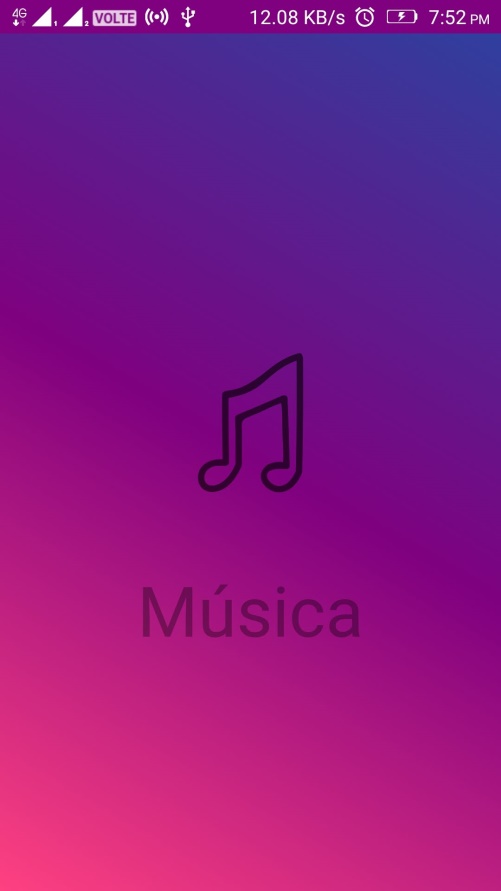
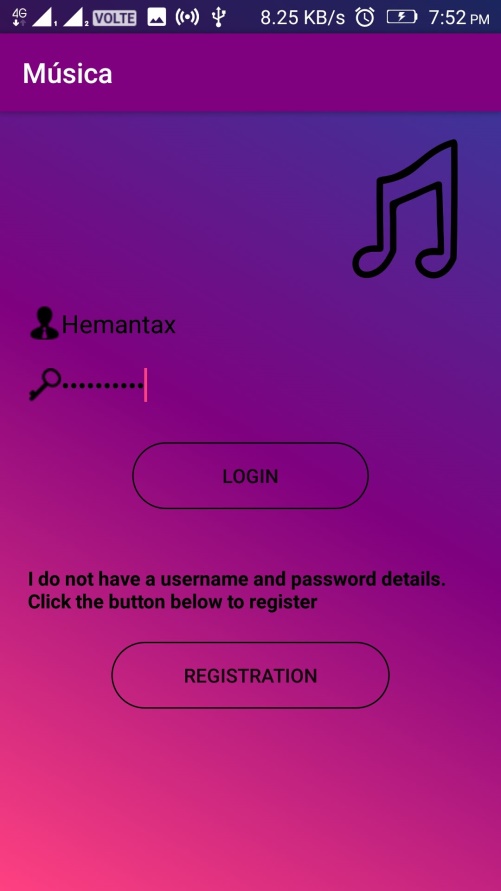
List of Tables with Attributes and Constraints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE NAME | FIELD NAME | DATATYPE | SIZE | CONSTRAINT |
| USER | USER\_ID | INT | 10 | PRIMARYKEY/AUTOINCREMENT/  NOT NULL |
|  | NAME | VARCHAR | 20 | NOT NULL |
|  | USERNAME | VARCHAR | 10 | UNIQUE/NOT NULL |
|  | PASSWORD | VARCHAR | 10 | NOT NULL |
|  | EMAIL\_ID | VARCHAR | 30 | UNIQUE/NOT NULL |

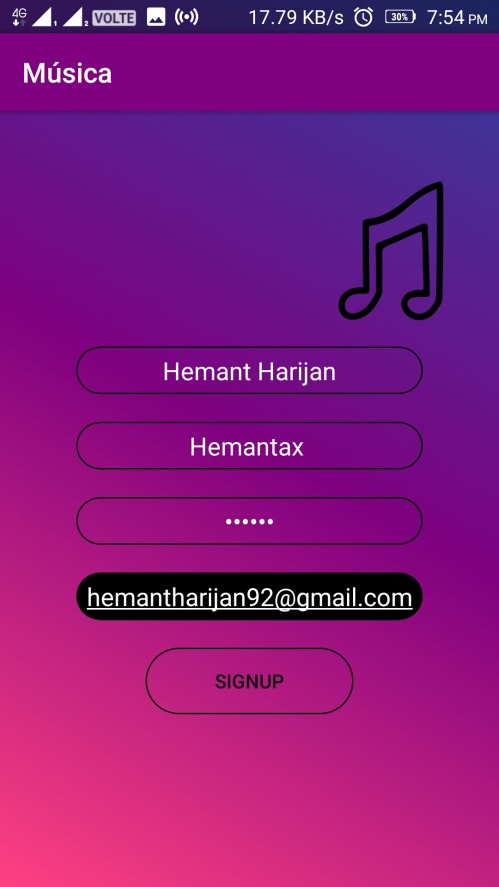
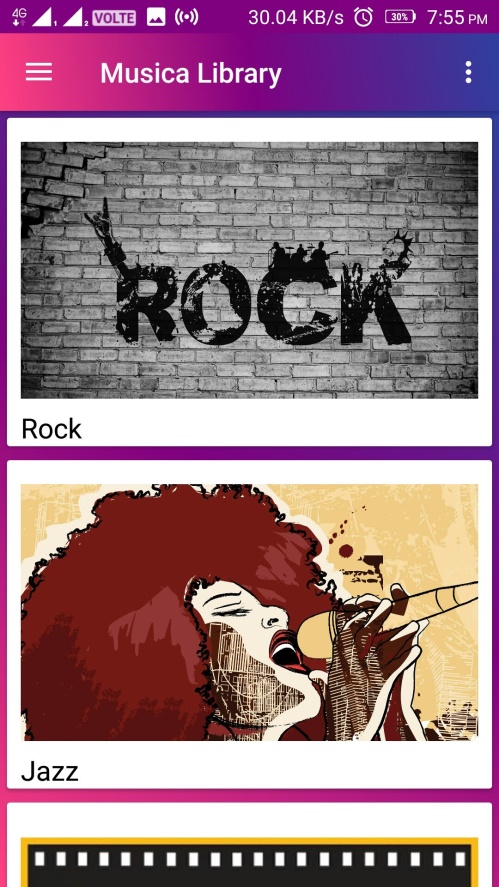
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE NAME | FIELD NAME | DATATYPE | SIZE | CONSTRAINT |
| GENRE | GENRE\_ID | INT | 11 | PRIMARYKEY/AUTOINCREMENT/  NOT NULL |
|  | GENRE\_NAME | VARCHAR | 255 | NOT NULL |
|  | IMG\_URL | VARCHAR | 225 | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE NAME | FIELD NAME | DATATYPE | `SIZE | CONSTRAINT |
| MUSIC | ID | INT | 11 | PRIMARYKEY/AUTOINCREMENT/ NOT NULL |
|  | USER\_ID | INT | 11 | FOREIGN KEY/NULL |
|  | GENRE\_ID | INT | 11 | FOREIGN KEY/NULL |
|  | MUSIC NAME | VARCHAR | 255 | NULL |
|  | ARTIST NAME | VARCHAR | 255 | NULL |
|  | COVER\_IMG\_URL | VARCHAR | 255 | NULL |
|  | MUSIC\_URL | VARCHAR | 255 | NULL |

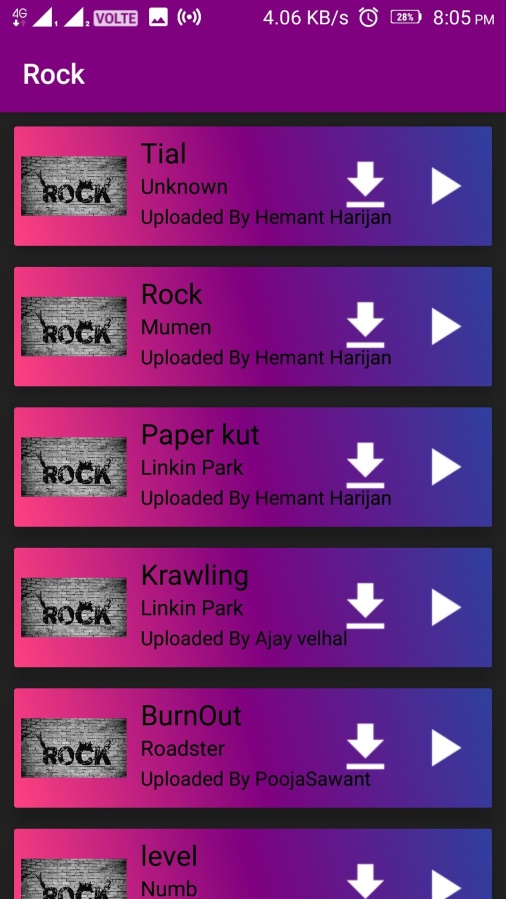
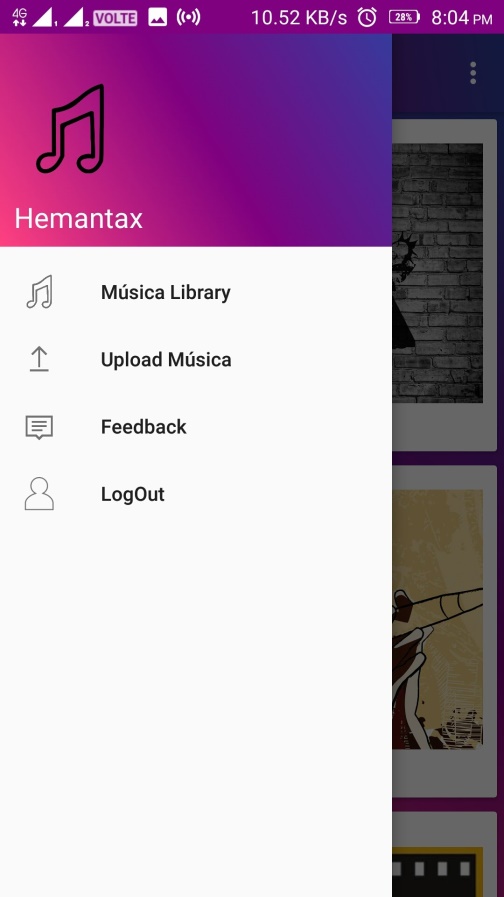
Screen Layout

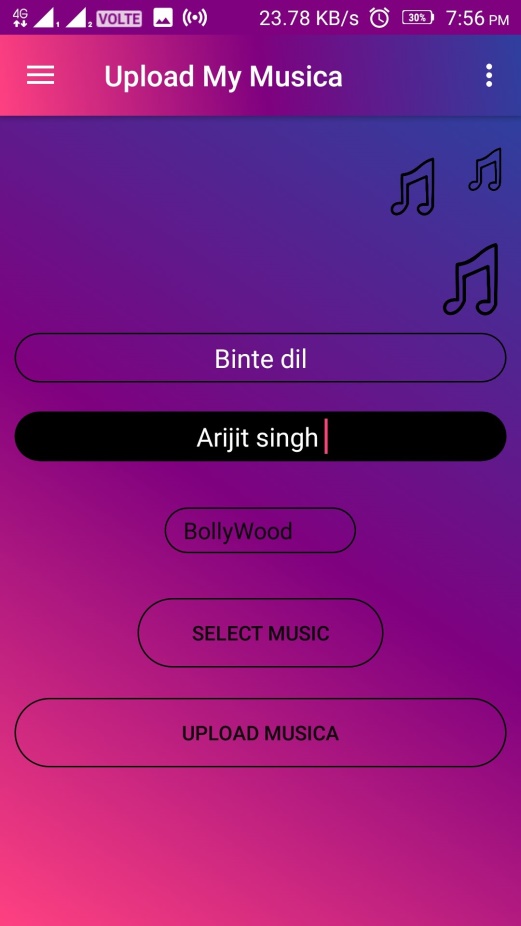
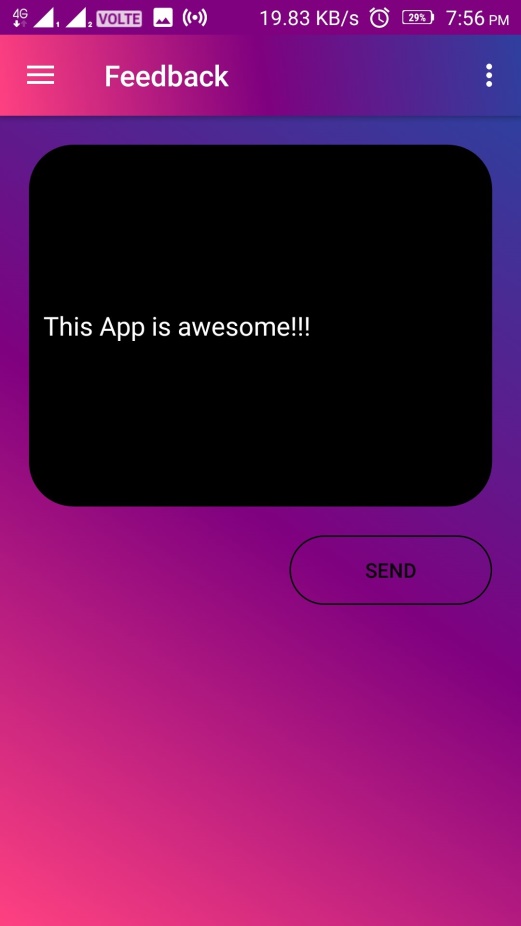
Splash Screen Activity Login Activity

Registration Activity Library Fragment

Playlist Activity Navigation Drawer

Upload Fragment Feedback

**System Implementation and Coding**

Server side codes

JSP Codes

Login.jsp

<%@ page import="java.sql.\*, org.json.\*" %>

<%

String username,password;

try{

username=request.getParameter("username");

password=request.getParameter("password");

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/musica","root","root");

PreparedStatement stmt = con.prepareStatement("select User\_ID from user where User\_Name=? and Password=?");

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet result = stmt.executeQuery();

JSONObject res = new JSONObject();

if (result.next()) {

out.println("1");

}

else {

out.println("0");

}

}

catch(Exception e){

out.println(e.toString());

}

%>

Insertdata.jsp

<%@ page import="java.sql.\*, org.json.\*" %>

<%

String name,username,password,email;

try{

name=request.getParameter("name");

username=request.getParameter("username");

password=request.getParameter("password");

email = request.getParameter("email");

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/musica","root","root");

PreparedStatement stmt = con.prepareStatement("insert into user(Name,User\_Name,Password,Email\_ID) values(?,?,?,?)");

stmt.setString(1,name);

stmt.setString(2,username);

stmt.setString(3,password);

stmt.setString(4,email);

int result = stmt.executeUpdate();

JSONObject res = new JSONObject();

res.put("result",result);

out.println(res.get("result"));

}

catch(Exception e){

out.println(e.toString());

}

%>

getgenre.jsp

<%@ page import="java.sql.\*, org.json.\*" %>

<%

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/musica","root","root");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("Select \* from genre");

JSONArray emp\_list = new JSONArray();

while(rs.next())

{

JSONObject emp = new JSONObject();

emp.put("id",rs.getInt("id"));

emp.put("genre\_name",rs.getString("genre\_name"));

emp.put("Img\_url",rs.getString("Img\_url"));

emp\_list.put(emp);

}

out.println(emp\_list);

}

catch(Exception e)

{

out.println(e.toString());

}

%>

getMusic.jsp

<%@ page import="java.sql.\*, org.json.\*" %>

<%

String genre\_id;

int genreid;

try{

genre\_id=request.getParameter("genre\_id");

genreid=Integer.parseInt(genre\_id);

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/musica","root","root");

PreparedStatement stmt = con.prepareStatement("select m.id, m.name,m.artist\_name,m.cover\_img\_url,m.music\_url,u.name from music m,user u where u.user\_id=m.user\_id and genre\_id=?;");

stmt.setInt(1,genreid);

ResultSet rs = stmt.executeQuery();

JSONArray songList = new JSONArray();

while(rs.next()){

JSONObject song = new JSONObject();

song.put("id",rs.getInt(1));

song.put("name",rs.getString(2));

song.put("artist\_name",rs.getString(3));

song.put("cover",rs.getString(4));

song.put("url",rs.getString(5));

song.put("uploaded",rs.getString(6));

songList.put(song);

}

out.println(songList);

}

catch(Exception e){

out.println(e.toString());

}

%>

UploadServlet.java

import java.io.\*;

import java.util.\*;

import java.sql.\*;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.fileupload.FileItem;

import org.apache.commons.fileupload.FileUploadException;

import org.apache.commons.fileupload.disk.DiskFileItemFactory;

import org.apache.commons.fileupload.servlet.ServletFileUpload;

import org.apache.commons.io.output.\*;

import org.json.\*;

public class UploadServlet extends HttpServlet {

private boolean isMultipart;

private String filePath;

//private int maxFileSize =50 \* 50 \* 1024;

// private int maxMemSize = 50\* 4 \* 1024;

private File file ;

private String user\_name,music\_name, artist\_name, genre,coverurl;

private int genreid, user\_id;

public void init( ){

// Get the file location where it would be stored.

filePath = getServletContext().getInitParameter("file-upload");

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, java.io.IOException {

// Check that we have a file upload request

isMultipart = ServletFileUpload.isMultipartContent(request);

response.setContentType("text/html");

java.io.PrintWriter out = response.getWriter( );

System.out.println("Servlet Called");

if( !isMultipart ) {

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet upload</title>");

out.println("</head>");

out.println("<body>");

out.println("<p>No file uploaded</p>");

out.println("</body>");

out.println("</html>");

return;

}

DiskFileItemFactory factory = new DiskFileItemFactory();

// maximum size that will be stored in memory

//factory.setSizeThreshold(maxMemSize);

// Location to save data that is larger than maxMemSize.

//factory.setRepository(new File("c:\\temp"));

// Create a new file upload handler

ServletFileUpload upload = new ServletFileUpload(factory);

try {

// Parse the request to get file items.

List fileItems = upload.parseRequest(request);

System.out.println(fileItems.toString());

// Process the uploaded file items

Iterator i = fileItems.iterator();

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet upload</title>");

out.println("</head>");

out.println("<body>");

while ( i.hasNext () ) {

FileItem fi = (FileItem)i.next();

System.out.println(fi.toString());

if ( !fi.isFormField () ) {

// Get the uploaded file parameters

String fieldName = fi.getFieldName();

String fileName = fi.getName();

String contentType = fi.getContentType();

boolean isInMemory = fi.isInMemory();

long sizeInBytes = fi.getSize();

// Write the file

if(fieldName.equals("file")){

if( fileName.lastIndexOf("\\") >= 0 ) {

file = new File( filePath + fileName.substring( fileName.lastIndexOf("\\"))) ;

} else {

file = new File( filePath + fileName.substring(fileName.lastIndexOf("\\")+1)) ;

}

System.out.println(file.toString() );

fi.write( file ) ;

}else if(fieldName.equals("username")){

user\_name = fi.getName();

}else if(fieldName.equals("musicname")){

music\_name =fi.getName();

}else if(fieldName.equals("artistname")){

artist\_name =fi.getName();

}else if(fieldName.equals("genre")){

genre =fi.getName();

}

System.out.println("Uploaded Filename: " + fileName + "<br>");

}

}

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/musica","root","root");

PreparedStatement stmt = con.prepareStatement("select g.id, g.Img\_url,u.User\_ID from genre g, user u where g.genre\_name=? and u.User\_Name=? ");

stmt.setString(1,genre);

stmt.setString(2,user\_name);

ResultSet rs = stmt.executeQuery();

if(rs.next())

{

genreid = rs.getInt("id");

coverurl=rs.getString("Img\_url");

user\_id=rs.getInt("User\_ID");

PreparedStatement stmt1 = con.prepareStatement("insert into music(user\_id,genre\_id,name,artist\_name,cover\_img\_url,Music\_url) values(?,?,?,?,?,?);");

stmt1.setInt(1,user\_id);

stmt1.setInt(2,genreid);

stmt1.setString(3,music\_name);

stmt1.setString(4,artist\_name);

stmt1.setString(5,coverurl);

stmt1.setString(6,"music/"+music\_name+".mp3");

int result = stmt1.executeUpdate();

JSONObject res = new JSONObject();

res.put("result",result);

out.println(res.get("result"));

}

} catch(Exception ex) {

ex.printStackTrace();

}

}

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, java.io.IOException {

throw new ServletException("GET method used with " +

getClass( ).getName( )+": POST method required.");

}

}

Web.xml

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1"

metadata-complete="true">

<display-name>Welcome to Tomcat</display-name>

<description>

Welcome to Tomcat

</description>

<context-param>

<description>Location to store uploaded file</description>

<param-name>file-upload</param-name>

<param-value>

C:\apache-tomcat-8.5.24\webapps\Music\music\

</param-value>

</context-param>

<servlet>

<servlet-name>UploadServlet</servlet-name>

<servlet-class>UploadServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>UploadServlet</servlet-name>

<url-pattern>/UploadServlet</url-pattern>

</servlet-mapping>

</web-app>

Client Side code

XML CODE

Resources Package

Anim package

Fade\_in.xml

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:shareInterpolator="true"

android:duration="5000">

<alpha

android:fromAlpha="0"

android:toAlpha="1">

</alpha>

</set>

Drawable package

myButton.xml

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android">

<item android:state\_pressed="true" >

<shape android:shape="rectangle" >

<corners android:radius="30dip" />

<stroke android:width="1dip" android:color="#000" />

</shape>

</item>

<item android:state\_focused="true">

<shape android:shape="rectangle" >

<corners android:radius="30dip" />

<stroke android:width="1dip" android:color="#000" />

<solid android:color="#000"/>

</shape>

</item>

<item >

<shape android:shape="rectangle" >

<corners android:radius="30dip" />

<stroke android:width="1dip" android:color="#000" />

</shape>

</item>

</selector>

Violet\_gradient.xml

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android">

<gradient

android:angle="45"

android:endColor="@color/colorPrimaryDark"

android:centerColor="@color/colorPrimary"

android:startColor="@color/colorAccent"

android:type="linear"/>

</shape>

Menu Package  
activity\_main\_drawer.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

tools:showIn="navigation\_view>

<group android:checkableBehavior="single >

<item

android:id="@+id/Library"

android:icon="@drawable/note1"

android:title="Música Library"

/>

<item

android:id="@+id/upload"

android:icon="@drawable/upload1"

android:title="Upload Música" />

<item

android:id="@+id/Feedback"

android:icon="@drawable/feedback\_icon"

android:title="Feedback" />

<item

android:id="@+id/Logout"

android:icon="@drawable/usericon"

android:title="LogOut"/>

</group> </menu>

Values Package

colors.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#800080</color>

<color name="colorPrimaryDark">#303F9F</color>

<color name="colorAccent">#FF4081</color>

<color name="transparent">@android:color/transparent</color>

</resources>

dimens.xml

<resources>

<dimen name="activity\_horizontal\_margin">16dp</dimen>

<dimen name="activity\_vertical\_margin">16dp</dimen>

<dimen name="nav\_header\_vertical\_spacing">8dp</dimen>

<dimen name="nav\_header\_height">176dp</dimen>

<dimen name="fab\_margin">16dp</dimen>

</resources>

string.xml

<resources>

<string name="app\_name">Música</string>

<string name="navigation\_drawer\_open">Open navigation drawer</string>

<string name="navigation\_drawer\_close">Close navigation drawer</string>

<string name="action\_settings">Settings</string>

<string name="genre">Choose a Genre</string>

<string-array name="genre\_list">

<item>Rock</item>

<item>Jazz</item>

<item>BollyWood</item>

<item>EDM</item>

<item>HipHop</item>

<item>Trance</item>

<item>Drums and Base</item>

<item>Electric</item>

<item>Fulte</item>

<item>GospelMusic</item>

<item>Metal</item>

<item>Chelo</item>

</string-array>

<string name="hello\_blank\_fragment">Hello blank fragment</string>

</resources>

style.xml

<resources>

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimary</item>

<item name="colorAccent">@color/colorAccent</item>

</style>

<style name="AppTheme.ActionBar.Transparent" parent="AppTheme">

<item name="windowActionBar">false</item>

<item name="windowNoTitle">true</item>

<item name="android:windowContentOverlay">@null</item>

<item name="windowActionBarOverlay">true</item>

<item name="colorPrimary">@android:color/transparent</item>

</style>

<style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

<style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />

</resources>

Layouts

activity\_login.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"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:tools="http://schemas.android.com/tools"

tools:context=".activity.LoginActivity"

android:background="@drawable/voilet\_gradient"

android:fitsSystemWindows="true">

<RelativeLayout

android:layout\_marginTop="120dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:id="@+id/linear1">

<EditText

android:id="@+id/etUser"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColorHint="#000"

android:layout\_marginEnd="20dp"

android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

android:elevation="10dp"

android:background="@android:color/transparent"

android:hint="Username"

android:gravity="center"

android:textColor="#000"

android:drawableLeft="@drawable/users"/>

<EditText

android:id="@+id/etPass"

android:textColorHint="#000"

android:background="@android:color/transparent"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="20dp"

android:layout\_marginTop="20dp"

android:gravity="center"

android:layout\_marginStart="20dp"

android:inputType="textPassword"

android:layout\_below="@+id/etUser"

android:hint="Password"

android:textColor="#000"

android:drawableLeft="@drawable/security"/>

<Button

android:id="@+id/btnLogin"

android:layout\_width="170dp"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/etPass"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="29dp"

android:background="@drawable/mybutton"

android:elevation="50dp"

android:text="LOGIN"/>

</RelativeLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/linear1"

android:orientation="vertical"

android:layout\_marginTop="20dp"

android:id="@+id/linearLayout">

<TextView

android:layout\_marginTop="20dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="I do not have a username and password details. Click the button below to register "

android:layout\_marginEnd="20dp"

android:textColor="#000"

android:layout\_marginStart="20dp"

android:layout\_marginBottom="20dp"

android:textStyle="bold"/>

<Button

android:id="@+id/btnReg"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/linearLayout"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="18dp"

android:background="@drawable/mybutton"

android:elevation="50dp"

android:text="Registration"

android:layout\_gravity="center"/>

</LinearLayout>

<ImageView

android:id="@+id/imageView2"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_alignParentEnd="true"

android:layout\_alignParentTop="true"

android:layout\_marginEnd="19dp"

android:layout\_marginTop="20dp"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_alignTop="@+id/imageView2"

android:layout\_marginEnd="51dp"

android:layout\_toStartOf="@+id/imageView2"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView4"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignBottom="@+id/imageView2"

android:layout\_alignParentStart="true"

android:layout\_marginBottom="15dp"

android:layout\_marginStart="30dp"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView5"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignStart="@+id/imageView4"

android:layout\_below="@+id/linearLayout"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView6"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:layout\_alignParentBottom="true"

android:layout\_alignParentEnd="true"

android:layout\_marginBottom="17dp"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView8"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_below="@+id/imageView5"

android:layout\_toEndOf="@+id/imageView5"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView9"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_above="@+id/imageView4"

android:layout\_alignEnd="@+id/imageView8"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView10"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignParentStart="true"

android:layout\_alignTop="@+id/imageView3"

android:layout\_marginStart="14dp"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView11"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_alignBottom="@+id/imageView4"

android:layout\_toStartOf="@+id/imageView2"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView12"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_alignStart="@+id/imageView11"

android:layout\_alignTop="@+id/imageView2"

android:layout\_marginStart="19dp"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView13"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_above="@+id/imageView8"

android:layout\_alignStart="@+id/imageView6"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView14"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_alignBottom="@+id/imageView6"

android:layout\_alignStart="@+id/imageView10"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView15"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignBottom="@+id/imageView8"

android:layout\_centerHorizontal="true"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView16"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_above="@+id/imageView8"

android:layout\_alignStart="@+id/imageView3"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView17"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignStart="@+id/imageView9"

android:layout\_alignTop="@+id/imageView13"

app:srcCompat="@drawable/star" />

</RelativeLayout>

activity\_registration.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"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:tools="http://schemas.android.com/tools"

tools:context=".activity.RegistrationActivity"

android:background="@drawable/voilet\_gradient”>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_marginTop="250dp">

<EditText

android:id="@+id/etName"

android:hint="name"

android:layout\_width="250dp"

android:layout\_gravity="center"

android:layout\_height="wrap\_content"

android:gravity="center"

android:padding="5dp"

android:textColor="#fff"

android:textColorHint="#000"

android:background="@drawable/mybutton"/>

<EditText

android:hint="username"

android:id="@+id/etUsername"

android:layout\_marginTop="20dp"

android:layout\_width="250dp"

android:layout\_gravity="center"

android:layout\_height="wrap\_content"

android:gravity="center"

android:textColor="#fff"

android:textColorHint="#000"

android:padding="5dp"

android:background="@drawable/mybutton"/>

<EditText

android:hint="password"

android:id="@+id/etPassword"

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:inputType="textPassword"

android:layout\_marginTop="20dp"

android:textColor="#fff"

android:textColorHint="#000"

android:gravity="center"

android:layout\_gravity="center"

android:background="@drawable/mybutton"

android:padding="5dp"/>

<EditText

android:hint="email"

android:id="@+id/etEmail"

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:inputType="textEmailAddress"

android:layout\_marginTop="20dp"

android:gravity="center"

android:textColor="#fff"

android:textColorHint="#000"

android:layout\_gravity="center"

android:background="@drawable/mybutton"

android:padding="5dp"/>

<Button

android:id="@+id/btnSignUp"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:text="Signup"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:layout\_marginBottom="20dp"

android:background="@drawable/mybutton"/>

</LinearLayout>

<ImageView

android:id="@+id/imageView7"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignParentEnd="true"

android:layout\_alignParentTop="true"

android:layout\_marginEnd="29dp"

android:layout\_marginTop="171dp"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView18"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_alignBottom="@+id/imageView7"

android:layout\_marginBottom="38dp"

android:layout\_marginEnd="15dp"

android:layout\_toStartOf="@+id/imageView7"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView19"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_alignBottom="@+id/imageView18"

android:layout\_marginBottom="48dp"

android:layout\_marginEnd="17dp"

android:layout\_toStartOf="@+id/imageView18"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView20"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_alignTop="@+id/imageView7"

android:layout\_toStartOf="@+id/imageView28"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView21"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_above="@+id/imageView20"

android:layout\_marginBottom="17dp"

android:layout\_toEndOf="@+id/imageView18"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView22"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_above="@+id/imageView19"

android:layout\_toEndOf="@+id/imageView21"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView23"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_above="@+id/imageView22"

android:layout\_marginStart="17dp"

android:layout\_toEndOf="@+id/imageView19"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView24"

android:layout\_width="35dp"

android:layout\_height="35dp"

android:layout\_alignEnd="@+id/imageView18"

android:layout\_below="@+id/imageView23"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView25"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_below="@+id/imageView23"

android:layout\_toEndOf="@+id/imageView27"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView26"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignEnd="@+id/imageView23"

android:layout\_below="@+id/imageView23"

android:layout\_marginTop="16dp"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView27"

android:layout\_width="45dp"

android:layout\_height="45dp"

android:layout\_above="@+id/imageView19"

android:layout\_toStartOf="@+id/imageView19"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView28"

android:layout\_width="45dp"

android:layout\_height="45dp"

android:layout\_alignTop="@+id/imageView23"

android:layout\_toStartOf="@+id/imageView26"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView29"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_alignTop="@+id/imageView18"

android:layout\_toStartOf="@+id/imageView22"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView30"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_below="@+id/imageView7"

android:layout\_toEndOf="@+id/imageView22"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView31"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_alignParentEnd="true"

android:layout\_below="@+id/imageView29"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView32"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignTop="@+id/imageView29"

android:layout\_toEndOf="@+id/imageView30"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView33"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignStart="@+id/imageView31"

android:layout\_below="@+id/imageView24"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView34"

android:layout\_width="35dp"

android:layout\_height="35dp"

android:layout\_alignBottom="@+id/imageView28"

android:layout\_toEndOf="@+id/imageView24"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView35"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_alignBottom="@+id/imageView18"

android:layout\_alignParentEnd="true"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView36"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_above="@+id/imageView35"

android:layout\_toEndOf="@+id/imageView19"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView37"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignStart="@+id/imageView32"

android:layout\_alignTop="@+id/imageView34"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView38"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignTop="@+id/imageView23"

android:layout\_toStartOf="@+id/imageView34"

app:srcCompat="@drawable/star" />

<ImageView

android:id="@+id/imageView39"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_alignEnd="@+id/imageView22"

android:layout\_alignTop="@+id/imageView38"

app:srcCompat="@drawable/star" />

</RelativeLayout>

Activity\_Splashscreen.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout 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="@drawable/voilet\_gradient"

tools:context="com.hemantax.musica.activity.SplashScreen">

<ImageView

android:id="@+id/imageView40"

android:layout\_width="100dp"

android:layout\_height="0dp"

android:layout\_marginBottom="26dp"

android:layout\_marginTop="205dp"

app:layout\_constraintBottom\_toTopOf="@+id/textView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/note1" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="169dp"

android:gravity="center"

android:text="Música"

android:textSize="50dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView40" />

</android.support.constraint.ConstraintLayout>

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fitsSystemWindows="true"

tools:openDrawer="start” >

<include

layout="@layout/app\_bar\_main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

<android.support.design.widget.NavigationView

android:id="@+id/nav\_view"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:fitsSystemWindows="true"

app:headerLayout="@layout/nav\_header\_main"

app:menu="@menu/activity\_main\_drawer" />

</android.support.v4.widget.DrawerLayout>

Activity\_Playlist.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout 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="com.hemantax.musica.activity.PlaylistActivity"

android:background="#222222">

<android.support.v7.widget.RecyclerView

android:id="@+id/songRv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

</android.support.constraint.ConstraintLayout>

App\_bar\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout 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="com.hemantax.musica.activity.MainActivity"

android:fitsSystemWindows="true">

<android.support.design.widget.AppBarLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/AppTheme.AppBarOverlay" >

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="@drawable/voilet\_gradient"

app:titleTextColor="#fff"

app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>

<include layout="@layout/content\_main"

/>

</android.support.design.widget.CoordinatorLayout>

Containt\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

tools:context="com.hemantax.musica.activity.MainActivity"

tools:showIn="@layout/app\_bar\_main"

android:id="@+id/fragment\_container"

android:background="@drawable/voilet\_gradient">

</FrameLayout>

Fragment\_Feedback.xml

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.hemantax.musica.fragment.FeedbackFragment">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/voilet\_gradient">

<EditText

android:id="@+id/etfeedback"

android:layout\_width="match\_parent"

android:layout\_height="250dp"

android:hint="FeedBack"

android:padding="10dp"

android:layout\_marginTop="20dp"

android:layout\_marginStart="20dp"

android:layout\_marginEnd="20dp"

android:textColor="#fff"

android:textColorHint="#000"

android:inputType="textMultiLine"

android:background="@drawable/mybutton” />

<Button

android:id="@+id/btnfeedbacksend"

android:layout\_below="@+id/etfeedback"

android:layout\_width="140dp"

android:layout\_height="wrap\_content"

android:text="Send"

android:layout\_marginTop="20dp"

android:layout\_marginStart="20dp"

android:layout\_marginEnd="20dp"

android:background="@drawable/mybutton"

android:layout\_alignParentEnd="true"/>

</RelativeLayout>

</FrameLayout>

Fragment\_Library.xml

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.hemantax.musica.fragment.LibraryFragment">

<android.support.v7.widget.RecyclerView

android:id="@+id/rvGenre"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</FrameLayout>

Fragment\_Upload.xml

<FrameLayout 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">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/voilet\_gradient">

<EditText

android:id="@+id/Mname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="10dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="150dp"

android:background="@drawable/mybutton"

android:hint="Music Name"

android:gravity="center"

android:textColor="#fff"

android:padding="5dp"

android:textColorHint="#000" />

<EditText

android:padding="5dp"

android:id="@+id/MArtist"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/Mname"

android:layout\_marginEnd="10dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="20dp"

android:gravity="center"

android:background="@drawable/mybutton"

android:hint="Artist Name"

android:textColor="#fff"

android:textColorHint="#000" />

<Spinner

android:id="@+id/MGenre"

android:padding="5dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/MArtist"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="32dp"

android:background="@drawable/mybutton"

android:entries="@array/genre\_list"

android:gravity="center"

android:prompt="@string/genre">

</Spinner>

<Button

android:id="@+id/Mpicker"

android:layout\_width="170dp"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/MGenre"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="31dp"

android:background="@drawable/mybutton"

android:padding="10dp"

android:text="select music" />

<Button

android:id="@+id/btnuploadmusic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/MGenre"

android:layout\_marginEnd="10dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="100dp"

android:text="Upload Musica"

android:background="@drawable/mybutton"

android:bottomRightRadius="30dp"

android:bottomLeftRadius="30dp"

android:topLeftRadius="30dp"

android:topRightRadius="30dp” />

<ImageView

android:id="@+id/imageView46"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignParentTop="true"

android:layout\_marginTop="29dp"

android:layout\_toEndOf="@+id/Mpicker"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView47"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignEnd="@+id/Mname"

android:layout\_below="@+id/imageView46"

android:layout\_marginTop="19dp"

app:srcCompat="@drawable/note1" />

<ImageView

android:id="@+id/imageView48"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_above="@+id/imageView47"

android:layout\_alignEnd="@+id/imageView47"

android:layout\_marginBottom="17dp"

app:srcCompat="@drawable/note1" />

</RelativeLayout>

</FrameLayout>

Item\_genre.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:elevation="10dp"

android:layout\_marginBottom="5dp"

android:layout\_marginTop="5dp"

android:layout\_marginEnd="5dp"

android:layout\_marginStart="5dp">

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical”>

<ImageView

android:id="@+id/imgGenre"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_marginEnd="10dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp” />

<TextView

android:id="@+id/tvGenreName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="10dp"

android:layout\_marginStart="10dp"

android:hint="Rock"

android:textSize="20sp"

android:textColor="#000000” />

</LinearLayout>

</android.support.v7.widget.CardView>

Item\_song.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_marginStart="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="5dp"

android:elevation="10dp"

app:cardElevation="10dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal"

android:background="@drawable/voilet\_gradient"

android:paddingBottom="5dp">

<ImageView

android:id="@+id/imgCover"

android:layout\_width="75dp"

android:layout\_height="75dp"

android:layout\_marginEnd="5dp"

android:layout\_marginStart="5dp"

android:layout\_marginTop="5dp"/>

<ImageView

android:id="@+id/btnDownload"

android:src="@drawable/ic\_download"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_centerVertical="true"

android:layout\_toStartOf="@+id/btnPlayPause" />

<ImageView

android:id="@+id/btnPlayPause"

android:src="@drawable/ic\_play"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignParentEnd="true"

android:layout\_centerVertical="true"

android:layout\_alignTop="@+id/btnDownload"

android:layout\_marginStart="5dp"

android:layout\_marginEnd="10dp" />

<LinearLayout

android:id="@+id/details"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_marginEnd="5dp"

android:layout\_marginTop="5dp"

android:layout\_toRightOf="@id/imgCover">

<TextView

android:id="@+id/tvSongName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Tial"

android:textColor="#000000"

android:textSize="20sp"

android:paddingStart="5dp" />

<TextView

android:id="@+id/tvArtist"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Unknown"

android:textSize="15sp"

android:textColor="#000000"

android:paddingLeft="5dp"/>

<TextView

android:id="@+id/tvUploaded"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Uploaded By Hemant"

android:paddingLeft="5dp"

android:layout\_marginTop="2dp"

android:textColor="#000000"/>

</LinearLayout>

</RelativeLayout>

</android.support.v7.widget.CardView>

Nav\_header.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"

android:layout\_width="match\_parent"

android:layout\_height="@dimen/nav\_header\_height"

android:background="@drawable/voilet\_gradient"

android:theme="@style/ThemeOverlay.AppCompat.Dark">

<ImageView

android:id="@+id/imageView"

android:layout\_width="64dp"

android:layout\_height="87dp"

android:layout\_alignParentStart="true"

android:layout\_centerVertical="true"

android:layout\_marginStart="18dp"

android:paddingTop="@dimen/nav\_header\_vertical\_spacing"

app:srcCompat="@drawable/note1" />

<TextView

android:id="@+id/usernamedisplay"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_below="@+id/imageView"

android:padding="10dp"

android:textSize="20dp"

android:paddingTop="@dimen/nav\_header\_vertical\_spacing"

android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

</RelativeLayout>

JAVA CODE

Activity Package

LoginActiviy.java

package com.hemantax.musica.activity;

import android.content.Intent;

import android.os.Build;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.view.Window;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.utils.PrefManager;

import com.hemantax.musica.utils.RetrofitUtils;

import com.hemantax.musica.utils.RetrofitUtils.RetrofitService;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class LoginActivity extends AppCompatActivity {

@BindView(R.id.etUser)

EditText etUser;

@BindView(R.id.etPass)

EditText etPass;

RetrofitService service;

PrefManager prefManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

prefManager = new PrefManager(this);

if (!prefManager.isFirstTimeLaunch() && prefManager.isLoggedIn()) {

launchHomeScreen();

finish();

}

setContentView(R.layout.activity\_login);

ButterKnife.bind(this);

service = RetrofitUtils.getInstance();

setStatusBarTranslucent(true);

}

protected void setStatusBarTranslucent(boolean makeTranslucent) {

if (makeTranslucent) {

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS);

} else { getWindow().clearFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS);

}

}

@OnClick({R.id.btnLogin, R.id.btnReg})

public void onViewClicked(View view) {

switch (view.getId()) {

case R.id.btnLogin:

login();

break;

case R.id.btnReg:

register();

break;

}

}

private void register() {

Intent intent=new Intent(LoginActivity.this,RegistrationActivity.class);

startActivity(intent);

}

private void login() {

final String name = etUser.getText().toString();

String pass = etPass.getText().toString();

service.login(name,pass).enqueue(new Callback<String>() {

@Override

public void onResponse(Call<String> call, Response<String> response) {

if(response.isSuccessful()){

if (response.body().equals("1")){

prefManager.setFirstTimeLaunch(false);

prefManager.setLoggedIn(true);

prefManager.setUserame(name);

Intent intent = new Intent(LoginActivity.this,MainActivity.class);

startActivity(intent);

}else{

etPass.setText("");

Toast.makeText(LoginActivity.this, "Username or password is Invalid", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

public void onFailure(Call<String> call, Throwable t) {

Log.e("Error",t.toString());

Toast.makeText(LoginActivity.this, "Failed", Toast.LENGTH\_SHORT).show();

}

});

}

private void launchHomeScreen() {

prefManager.setFirstTimeLaunch(false);

startActivity(new Intent(LoginActivity.this, MainActivity.class));

finish();

}

@Override

public void onBackPressed() {

}

}

MainActivity.java

package com.hemantax.musica.activity;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.design.widget.NavigationView;

import android.support.v4.app.ActivityCompat;

import android.support.v4.app.FragmentTransaction;

import android.support.v4.view.GravityCompat;

import android.support.v4.widget.DrawerLayout;

import android.support.v7.app.ActionBarDrawerToggle;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.Toolbar;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.TextView;

import com.hemantax.musica.R;

import com.hemantax.musica.fragment.FeedbackFragment;

import com.hemantax.musica.fragment.LibraryFragment;

import com.hemantax.musica.fragment.UploadFragment;

import com.hemantax.musica.utils.PrefManager;

import butterknife.BindView;

import butterknife.ButterKnife;

public class MainActivity extends AppCompatActivity

implements NavigationView.OnNavigationItemSelectedListener {

@BindView(R.id.toolbar)

Toolbar toolbar;

@BindView(R.id.nav\_view)

NavigationView navigationView;

@BindView(R.id.drawer\_layout)

DrawerLayout drawer;

TextView txtusername;

PrefManager prefManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ButterKnife.bind(this);

setSupportActionBar(toolbar);

prefManager = new PrefManager(this);

txtusername = navigationView.getHeaderView(0).findViewById(R.id.usernamedisplay);

txtusername.setText(prefManager.getUsername());

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(

this, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);

drawer.addDrawerListener(toggle);

toggle.syncState();

navigationView.setNavigationItemSelectedListener(this);

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.replace(R.id.fragment\_container, new LibraryFragment());

ft.commit();

getSupportActionBar().setTitle("Musica Library");

if(ActivityCompat.checkSelfPermission(this,Manifest.permission.READ\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED

){

requestPermission();

}

if (ActivityCompat.checkSelfPermission(this,Manifest.permission.WRITE\_EXTERNAL\_STORAGE)!=PackageManager.PERMISSION\_GRANTED){

requestPermission();

}

}

@Override

public void onBackPressed() {

if (drawer.isDrawerOpen(GravityCompat.START)) {

drawer.closeDrawer(GravityCompat.START);

} else {

super.onBackPressed();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@SuppressWarnings("StatementWithEmptyBody")

@Override

public boolean onNavigationItemSelected(MenuItem item) {

// Handle navigation view item clicks here.

int id = item.getItemId();

if (id == R.id.Library) {

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.replace(R.id.fragment\_container, new LibraryFragment());

ft.commit();

getSupportActionBar().setTitle("Musica Library");

} else if (id == R.id.upload) {

FragmentTransaction ftu =getSupportFragmentManager().beginTransaction();

ftu.replace(R.id.fragment\_container, new UploadFragment());

ftu.commit();

getSupportActionBar().setTitle("Upload My Musica");

} else if (id == R.id.Feedback) {

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.replace(R.id.fragment\_container, new FeedbackFragment());

ft.commit();

getSupportActionBar().setTitle("Feedback");

}else if(id == R.id.Logout) {

Intent intent = new Intent(MainActivity.this,LoginActivity.class);

prefManager.setLoggedIn(false);

startActivity(intent)

}

drawer.closeDrawer(GravityCompat.START);

return true;

}

public void requestPermission(){

ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE},1);

ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE},1);

}

}

PlayListActivity.java

package com.hemantax.musica.activity;

import android.content.ComponentName;

import android.content.Context;

import android.content.Intent;

import android.content.ServiceConnection;

import android.media.AudioManager;

import android.media.Image;

import android.media.MediaPlayer;

import android.os.AsyncTask;

import android.os.Bundle;

import android.os.Environment;

import android.os.IBinder;

import android.support.v4.media.session.PlaybackStateCompat;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.View;

import android.widget.ImageView;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.adapter.SongsRVAdapter;

import com.hemantax.musica.model.Song;

import com.hemantax.musica.services.MusicService;

import com.hemantax.musica.utils.RetrofitUtils;

import java.io.BufferedInputStream;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.URL;

import java.net.URLConnection;

import java.util.ArrayList;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.Unbinder;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class PlaylistActivity extends AppCompatActivity implements SongsRVAdapter.OnClickListener {

@BindView(R.id.songRv)

RecyclerView songRv;

RetrofitUtils.RetrofitService service;

List<Song> songList = new ArrayList<Song>();

SongsRVAdapter adapter;

private MusicService musicService;

private Intent playIntent;

private boolean musicBound = false;

private String TAG = this.getClass().getSimpleName();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_playlist);

ButterKnife.bind(this);

getSupportActionBar().setTitle(getIntent().getStringExtra("genre\_name"));

service = RetrofitUtils.getInstance();

int genre\_id = getIntent().getIntExtra("genre\_id", 1);

service.getMusic(genre\_id).enqueue(new Callback<List<Song>>() {

@Override

public void onResponse(Call<List<Song>> call, Response<List<Song>> response) {

songList = response.body();

setList(songList);

}

@Override

public void onFailure(Call<List<Song>> call, Throwable t) {

Toast.makeText(PlaylistActivity.this, "No Network", Toast.LENGTH\_SHORT).show();

}

});

adapter = new SongsRVAdapter(songList, this);

songRv.setLayoutManager(new LinearLayoutManager(PlaylistActivity.this));

songRv.setAdapter(adapter);

}

@Override

protected void onStart() {

super.onStart();

if (playIntent == null) {

Log.e("if", "entered");

playIntent = new Intent(this, MusicService.class);

startService(playIntent);

bindService(playIntent, musicConnection, Context.BIND\_AUTO\_CREATE);

}

}

@Override

protected void onStop() {

if (musicBound) {

unbindService(musicConnection);

musicBound = false;

}

super.onStop();

}

@Override

protected void onDestroy() {

musicService = null;

super.onDestroy();

}

public void setList(List<Song> songList) {

adapter = new SongsRVAdapter(songList, this);

songRv.setLayoutManager(new LinearLayoutManager(PlaylistActivity.this));

songRv.setAdapter(adapter);

musicService.setSongsList(songList);

}

@Override

public void onPlayPauseClicked(View v, int position, Song s) {

Log.e("PlaylistActivity", s.getUrl());

ImageView playPauseButton = (ImageView) v;

if (musicService.isPng()) {

playPauseButton.setImageResource(R.drawable.ic\_play);

musicService.pausePlayer();

} else {

playPauseButton.setImageResource(R.drawable.ic\_pause);

musicService.setSongPos(position);

musicService.playSong();

}

}

@Override

public void onDownloadClicked(View v, int position, Song s) {

service.getFile(RetrofitUtils.BASE\_URL + s.getUrl()).enqueue(new Callback<ResponseBody>() {

@Override

public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {

Log.d(TAG, response.message());

if(!response.isSuccessful()){

Log.e(TAG, "Something's gone wrong");

return;

}

DownloadFileAsyncTask downloadFileAsyncTask = new DownloadFileAsyncTask();

downloadFileAsyncTask.execute(response.body().byteStream());

}

@Override

public void onFailure(Call<ResponseBody> call, Throwable t) {

Log.e(TAG, t.getMessage());

}

});

}

public List<Song> getSongsList() {

return songList;

}

private ServiceConnection musicConnection = new ServiceConnection() {

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

MusicService.MusicBinder binder = (MusicService.MusicBinder) service;

Log.e("MusicService", "Instantiated");

musicService = binder.getService();

musicService.setSongsList(songList);

musicBound = true;

}

@Override

public void onServiceDisconnected(ComponentName name) {

musicBound = false;

}

};

private class DownloadFileAsyncTask extends AsyncTask<InputStream, Void, Boolean> {

final String appDirectoryName = getString(R.string.app\_name);

final File musicRoot = new File(Environment.getExternalStoragePublicDirectory(

Environment.DIRECTORY\_DOWNLOADS), appDirectoryName);

final String filename = "music.mp3";

@Override

protected Boolean doInBackground(InputStream... params) {

musicRoot.mkdirs();

InputStream inputStream = params[0];

File file = new File(musicRoot, filename);

OutputStream output = null;

try {

file.createNewFile();

output = new FileOutputStream(file);

byte[] buffer = new byte[1024]; // or other buffer size

int read;

Log.d(TAG, "Attempting to write to: " + musicRoot + "/" + filename);

while ((read = inputStream.read(buffer)) != -1) {

output.write(buffer, 0, read);

Log.v(TAG, "Writing to buffer to output stream.");

}

Log.d(TAG, "Flushing output stream.");

output.flush();

Log.d(TAG, "Output flushed.");

} catch (IOException e) {

Log.e(TAG, "IO Exception: " + e.getMessage());

e.printStackTrace();

return false;

} finally {

try {

if (output != null) {

output.close();

Log.d(TAG, "Output stream closed sucessfully.");

}

else{

Log.d(TAG, "Output stream is null");

}

} catch (IOException e){

Log.e(TAG, "Couldn't close output stream: " + e.getMessage());

e.printStackTrace();

return false;

}

}

return true;

}

@Override

protected void onPostExecute(Boolean result) {

super.onPostExecute(result);

Log.d(TAG, "Download success: " + result);

Toast.makeText(PlaylistActivity.this, "Download "+result, Toast.LENGTH\_SHORT).show();

}

}

}

RegisterActivity.java

package com.hemantax.musica.activity;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.utils.RetrofitUtils;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class RegistrationActivity extends AppCompatActivity {

@BindView(R.id.etName)

EditText etName;

@BindView(R.id.etUsername)

EditText etUsername;

@BindView(R.id.etPassword)

EditText etPassword;

@BindView(R.id.etEmail)

EditText etEmail;

RetrofitUtils.RetrofitService service;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_registration);

ButterKnife.bind(this);

service = RetrofitUtils.getInstance();

setStatusBarTranslucent(true);

}

protected void setStatusBarTranslucent(boolean makeTranslucent) {

if (makeTranslucent) {

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS);

} else {

getWindow().clearFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS);

}

}

@OnClick(R.id.btnSignUp)

public void onViewClicked() {

service.register(etName.getText().toString(),etUsername.getText().toString(),etPassword.getText().toString(),etEmail.getText().toString())

.enqueue(new Callback<String>() {

@Override

public void onResponse(Call<String> call, Response<String> response) {

String success=response.body().toString();

if(success.equals("1")){

Intent intent = new Intent(RegistrationActivity.this,LoginActivity.class);

startActivity(intent);

Toast.makeText(RegistrationActivity.this, "Registration successful", Toast.LENGTH\_SHORT).show();

}else{

etName.setText("");

etUsername.setText("");

etPassword.setText("");

etEmail.setText("");

Toast.makeText(RegistrationActivity.this, "Failed", Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onFailure(Call<String> call, Throwable t) {

Log.e("Error",t.toString());

Toast.makeText(RegistrationActivity.this, "Failed", Toast.LENGTH\_SHORT).show();

}

});

}

}

SplashScreen.java

package com.hemantax.musica.activity;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.widget.TextView;

import com.hemantax.musica.R;

import butterknife.BindView;

import butterknife.ButterKnife;

public class SplashScreen extends AppCompatActivity {

private static int splash\_screen=3000;

@BindView(R.id.imageView40)

ImageView imageView;

@BindView(R.id.textView)

TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splash\_screen);

ButterKnife.bind(this);

Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fade\_in);

imageView.startAnimation(animation);

textView.startAnimation(animation);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

Intent intent = new Intent(SplashScreen.this,LoginActivity.class);

startActivity(intent);

finish();

}

},splash\_screen);

}

}

Adapter Package

GenreRVAdapter.java

package com.hemantax.musica.adapter;

import android.content.Context;

import android.content.Intent;

import android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import com.hemantax.musica.R;

import com.hemantax.musica.activity.PlaylistActivity;

import com.hemantax.musica.model.Genre;

import com.hemantax.musica.utils.RetrofitUtils;

import com.squareup.picasso.Picasso;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

public class GenreRVAdapter extends RecyclerView.Adapter<GenreRVAdapter.MyViewHolder> {

List<Genre> genreList;

public GenreRVAdapter(List<Genre> genreList) {

this.genreList = genreList;

}

@Override

public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_genre, parent, false);

return new MyViewHolder(itemView);

}

@Override

public void onBindViewHolder(MyViewHolder holder, int position) {

Genre genre = genreList.get(position);

String url = RetrofitUtils.BASE\_URL + genre.getUrl();

holder.tvGenreName.setText(genre.getName());

Picasso.with(holder.ctx)

.load(url)

.into(holder.imgGenre);

}

@Override

public int getItemCount() {

return genreList.size();

}

public class MyViewHolder extends RecyclerView.ViewHolder {

@BindView(R.id.imgGenre)

ImageView imgGenre;

@BindView(R.id.tvGenreName)

TextView tvGenreName;

Context ctx;

public MyViewHolder(final View itemView) {

super(itemView);

ButterKnife.bind(this, itemView);

ctx = itemView.getContext();

itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

int position = getAdapterPosition();

Genre genre = genreList.get(position);

Intent i = new Intent(ctx, PlaylistActivity.class);

i.putExtra("genre\_name",genre.getName());

i.putExtra("genre\_id",genre.getId());

ctx.startActivity(i);

}

});

}

}

}

SongsRVAdapter.java

package com.hemantax.musica.adapter;

import android.app.Activity;

import android.content.Context;

import android.os.Environment;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.activity.PlaylistActivity;

import com.hemantax.musica.model.Song;

import com.hemantax.musica.utils.RetrofitUtils;

import com.squareup.picasso.Picasso;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class SongsRVAdapter extends RecyclerView.Adapter<SongsRVAdapter.MyViewHolder> {

RetrofitUtils.RetrofitService service;

List<Song> songList;

View itemView;

Context context;

private OnClickListener listener = null;

public SongsRVAdapter(List<Song> songList, OnClickListener listener) {

this.songList = songList;

this.listener = null;

this.listener = listener;

}

@Override

public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_song, parent, false);

return new MyViewHolder(itemView);

}

@Override

public void onBindViewHolder(MyViewHolder holder, int position) {

Song song = songList.get(position);

holder.tvSongName.setText(song.getName());

holder.tvArtist.setText(song.getArtist());

holder.tvUploaded.setText("Uploaded By " + song.getUploadedBy());

Picasso.with(holder.ctx).load(RetrofitUtils.BASE\_URL + song.getImageUrl()).into(holder.imgCover);

}

@Override

public int getItemCount() {

return songList.size();

}

public void setOnClickListener(OnClickListener listener) {

this.listener = null;

this.listener = listener;

}

public class MyViewHolder extends RecyclerView.ViewHolder {

@BindView(R.id.imgCover)

ImageView imgCover;

@BindView(R.id.btnDownload)

ImageView btnDownload;

@BindView(R.id.btnPlayPause)

ImageView btnPlayPause;

@BindView(R.id.tvSongName)

TextView tvSongName;

@BindView(R.id.tvArtist)

TextView tvArtist;

@BindView(R.id.tvUploaded)

TextView tvUploaded;

Context ctx;

public MyViewHolder(View itemView) {

super(itemView);

ButterKnife.bind(this, itemView);

ctx = itemView.getContext();

}

@OnClick({R.id.btnDownload, R.id.btnPlayPause})

public void onViewClicked(View view) {

int position = getAdapterPosition();

Song s = songList.get(position);

switch (view.getId()) {

case R.id.btnDownload:

listener.onDownloadClicked(view,position,s);

break;

case R.id.btnPlayPause:

listener.onPlayPauseClicked(view,position,s);

break;

}

}

}

public interface OnClickListener {

void onPlayPauseClicked(View v, int position, Song s);

void onDownloadClicked(View v, int position, Song s);

}

}

Fragment Package

FeedbackFragment.java

package com.hemantax.musica.fragment;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.Toast;

import com.hemantax.musica.R;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import butterknife.Unbinder;

public class FeedbackFragment extends Fragment {

@BindView(R.id.etfeedback)

EditText txtfeedback;

Unbinder unbinder;

public FeedbackFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_feedback, container, false);

unbinder = ButterKnife.bind(this, view);

return view;

}

@OnClick({R.id.btnfeedbacksend})

public void onViewClicked(final View view) {

String feedback = txtfeedback.getText().toString();

final Intent intent = new Intent(Intent.ACTION\_SEND);

intent.setData(Uri.parse("email"));

intent.setType("message/rfc822");

intent.putExtra(Intent.EXTRA\_EMAIL, new String[]{"hemantharijan92@gmail.com"});

intent.putExtra(Intent.EXTRA\_SUBJECT, "musica feedback");

intent.putExtra(Intent.EXTRA\_TEXT, feedback);

startActivity(Intent.createChooser(intent, "Select Application"));

Toast.makeText(getActivity(), "Feedback Send", Toast.LENGTH\_SHORT).show();

}

@Override

public void onDestroyView() {

super.onDestroyView();

unbinder.unbind();

}

}

LibraryFragment.java

package com.hemantax.musica.fragment;

import android.content.Context;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.activity.MainActivity;

import com.hemantax.musica.adapter.GenreRVAdapter;

import com.hemantax.musica.model.Genre;

import com.hemantax.musica.utils.RetrofitUtils;

import java.util.ArrayList;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.Unbinder;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class LibraryFragment extends Fragment {

@BindView(R.id.rvGenre)

RecyclerView rvGenre;

Unbinder unbinder;

Context ctx;

RetrofitUtils.RetrofitService service;

GenreRVAdapter adapter;

List<Genre> genreList = new ArrayList<Genre>();

public LibraryFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_library, container, false);

unbinder = ButterKnife.bind(this, view);

return view;

}

@Override

public void onActivityCreated(@Nullable Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

ctx = (AppCompatActivity)getActivity();

service = RetrofitUtils.getInstance();

service.getGenres().enqueue(new Callback<List<Genre>>() {

@Override

public void onResponse(Call<List<Genre>> call, Response<List<Genre>> response) {

Log.e("MainActivity","OnResponse");

genreList = response.body();

adapter = new GenreRVAdapter(genreList);

rvGenre.setLayoutManager(new LinearLayoutManager(ctx));

rvGenre.setAdapter(adapter);

}

@Override

public void onFailure(Call<List<Genre>> call, Throwable t) {

Log.e("MainActivity","OnFailure",t);

Toast.makeText(ctx, "No Network", Toast.LENGTH\_SHORT).show();

}

});

}

@Override

public void onDestroyView() {

super.onDestroyView();

unbinder.unbind();

}

}

UploadFragment.java

package com.hemantax.musica.fragment;

import android.app.ProgressDialog;

import android.content.CursorLoader;

import android.content.Intent;

import android.database.Cursor;

import android.net.Uri;

import android.os.Bundle;

import android.preference.MultiSelectListPreference;

import android.provider.MediaStore;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.Switch;

import android.widget.Toast;

import com.hemantax.musica.R;

import com.hemantax.musica.activity.MainActivity;

import com.hemantax.musica.activity.RegistrationActivity;

import com.hemantax.musica.utils.PrefManager;

import com.hemantax.musica.utils.RetrofitUtils;

import java.io.File;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import butterknife.Unbinder;

import okhttp3.MediaType;

import okhttp3.MultipartBody;

import okhttp3.RequestBody;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

import retrofit2.http.Multipart;

import static android.app.Activity.RESULT\_OK;

public class UploadFragment extends Fragment {

PrefManager prefManager;

@BindView(R.id.Mname)

EditText etMname;

@BindView(R.id.MArtist)

EditText etMartist;

@BindView(R.id.MGenre)

Spinner sMgenre;

Unbinder unbinder;

String Username;

RetrofitUtils.RetrofitService service;

Uri fileuri;

private ProgressDialog progress;

private String TAG = this.getClass().getSimpleName();

public UploadFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_upload, container, false);

unbinder = ButterKnife.bind(this,view);

return view;

}

@Override

public void onActivityCreated(@Nullable Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

prefManager=new PrefManager(getActivity());

Username = prefManager.getUsername();

service = RetrofitUtils.getInstance();

}

@OnClick({R.id.btnuploadmusic,R.id.Mpicker})

public void onViewClicked(final View view){

switch (view.getId()){

case R.id.btnuploadmusic:

String path = getAudioPath(fileuri);

File musicFile = new File(path);

Log.e(TAG,musicFile.toString());

MultipartBody.Part musicPart = MultipartBody.Part.createFormData("file",musicFile.getName(),RequestBody.create(MediaType.parse("audio/\*"),musicFile));

MultipartBody.Part username = MultipartBody.Part.createFormData("username",Username,RequestBody.create(MediaType.parse("text/plain"),Username));

MultipartBody.Part musicname = MultipartBody.Part.createFormData("musicname",etMname.getText().toString(),RequestBody.create(MediaType.parse("text/plain"),etMname.getText().toString()));

MultipartBody.Part artistname = MultipartBody.Part.createFormData("artistname",etMartist.getText().toString(),RequestBody.create(MediaType.parse("text/plain"),etMartist.getText().toString()));

MultipartBody.Part genre = MultipartBody.Part.createFormData("genre",sMgenre.getSelectedItem().toString(),RequestBody.create(MediaType.parse("text/plain"),sMgenre.getSelectedItem().toString()));

service.uploadMusic(musicPart,username,musicname,artistname,genre).enqueue(new Callback<ResponseBody>() {

@Override

public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {

Toast.makeText(getActivity(), "Upload Successful", Toast.LENGTH\_SHORT).show();

Log.e(TAG,response.body().toString());

UploadProgress(view);

progress.dismiss();

}

@Override

public void onFailure(Call<ResponseBody> call, Throwable t) {

Log.e(TAG,"Failure",t);

}

});

break;

case R.id.Mpicker:

Intent intent\_upload = new Intent();

intent\_upload.setType("audio/\*");

intent\_upload.setAction(Intent.ACTION\_GET\_CONTENT);

startActivityForResult(intent\_upload,1);

}

private String getAudioPath(Uri uri) {

String[] data = {MediaStore.Audio.Media.DATA};

CursorLoader loader = new CursorLoader(getActivity(), uri, data, null, null, null);

Cursor cursor = loader.loadInBackground();

int column\_index = cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DATA);

cursor.moveToFirst();

return cursor.getString(column\_index);

}

public void UploadProgress(View view){

progress=new ProgressDialog(getActivity());

progress.setMessage("Uploading Music");

progress.setProgressStyle(ProgressDialog.STYLE\_SPINNER);

progress.setIndeterminate(true);

progress.setProgress(0);

progress.show();

final int totalProgressTime = 100;

final Thread t = new Thread(){

@Override

public void run() {

int jumpTime =0;

while (jumpTime < totalProgressTime){

try{

sleep(200);

jumpTime+=5;

progress.setProgress(jumpTime);

}catch (InterruptedException e){

e.printStackTrace();

}

}

super.run();

}

};

t.start();

@Override

public void onActivityResult(int requestCode,int resultCode,Intent data){

super.onActivityResult(requestCode, resultCode, data);

if(requestCode == 1){

if(resultCode == RESULT\_OK){

//the selected audio.

fileuri = data.getData();

}

}

}

@Override

public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

}

@Override

public void onDestroy() {

super.onDestroy();

unbinder.unbind();

}

}

Model Package

Genre.java

package com.hemantax.musica.model;

import com.google.gson.annotations.SerializedName;

public class Genre {

@SerializedName("id") int id;

@SerializedName("genre\_name") String name;

@SerializedName("Img\_url") String url;

public Genre(int id, String name, String url) {

this.id = id;

this.name = name;

this.url = url;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

}

Songs.java

package com.hemantax.musica.model;

import com.google.gson.annotations.SerializedName;

public class Song {

@SerializedName("id") int id;

@SerializedName("name")String name;

@SerializedName("cover")String imageUrl;

@SerializedName("artist\_name")String artist;

@SerializedName("uploaded")String uploadedBy;

@SerializedName("url")String url;

public Song(int id, String name, String imageUrl, String artist, String uploadedBy,String url) {

this.id = id;

this.name = name;

this.imageUrl = imageUrl;

this.artist = artist;

this.uploadedBy = uploadedBy;

this.url = url;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getImageUrl() {

return imageUrl;

}

public void setImageUrl(String imageUrl) {

this.imageUrl = imageUrl;

}

public String getArtist() {

return artist;

}

public void setArtist(String artist) {

this.artist = artist;

}

public String getUploadedBy() {

return uploadedBy;

}

public void setUploadedBy(String uploadedBy) {

this.uploadedBy = uploadedBy;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

}

Upload.java

package com.hemantax.musica.model;

import com.google.gson.annotations.SerializedName;

public class upload {

@SerializedName("username") String username;

@SerializedName("Musicname")String Musicname;

@SerializedName("artist\_name")String artist\_name;

@SerializedName("genre")String genre;

public upload(String userid, String Musicname, String artist\_name, String genre) {

this.username = userid;

this.Musicname = Musicname;

this.artist\_name = artist\_name;

this.genre = genre;

}

public String getUsername() {

return username;

}

public void setUsername(String userid) {

this.username = userid;

}

public String getMusicname() {

return Musicname;

}

public void setMusicname(String musicname) {

Musicname = musicname;

}

public String getArtist\_name() {

return artist\_name;

}

public void setArtist\_name(String artist\_name) {

this.artist\_name = artist\_name;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

}

Service Package

MusicService.java

package com.hemantax.musica.services;

import android.app.Service;

import android.content.ContentUris;

import android.content.Intent;

import android.media.AudioManager;

import android.media.MediaPlayer;

import android.net.Uri;

import android.os.Binder;

import android.os.IBinder;

import android.os.PowerManager;

import android.provider.MediaStore;

import com.hemantax.musica.model.Song;

import com.hemantax.musica.utils.RetrofitUtils;

import java.io.IOException;

import java.util.List;

public class MusicService extends Service {

private final IBinder musicBind = new MusicBinder();

@Override

public IBinder onBind(Intent intent) {

return musicBind;

}

private MediaPlayer player;

private List<Song> songsList;

private int songPos;

private static MusicService mInstance = null;

Song playSong;

//public static final String BROADCAT\_ACTION = "musicChanged";

//Intent broadcastIntent;

public class MusicBinder extends Binder{

public MusicService getService(){

return MusicService.this;

}

}

@Override

public void onCreate() {

super.onCreate();

mInstance = this;

songPos = 0;

player = new MediaPlayer();

initMusicPlayer();

}

public void initMusicPlayer(){

player.setWakeMode(getApplicationContext(), PowerManager.PARTIAL\_WAKE\_LOCK);

player.setAudioStreamType(AudioManager.STREAM\_MUSIC);

player.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mediaPlayer) {

mediaPlayer.start();

}

});

player.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mediaPlayer) {

mediaPlayer.reset();

//playNext();

//sendBroadcast(broadcastIntent);

}

});

player.setOnErrorListener(new MediaPlayer.OnErrorListener() {

@Override

public boolean onError(MediaPlayer mediaPlayer, int i, int i1) {

mediaPlayer.reset();

return false;

}

});

}

public void setSongsList(List<Song> songsList) {

this.songsList = songsList;

}

public void setSongPos(int songPos) {

this.songPos = songPos;

}

@Override

public boolean onUnbind(Intent intent) {

// player.stop();

// player.release();

return false;

}

@Override

public void onDestroy() {

player.stop();

player.release();

super.onDestroy();

}

public void playSong(){

player.reset();

playSong = songsList.get(songPos);

String url = RetrofitUtils.BASE\_URL + playSong.getUrl();

try {

player.setDataSource(url);

} catch (IOException e) {

e.printStackTrace();

}

player.prepareAsync();

}

public int getPosn(){

return player.getCurrentPosition();

}

public int getDur(){

return player.getDuration();

}

public boolean isPng(){

return player.isPlaying();

}

public void pausePlayer(){

player.pause();

}

public void go(){

player.start();

}

public static MusicService getInstance(){

return mInstance;

}

public Song getPlaySong() {

return playSong;

}

}

Utils Package

PrefManager.java

package com.hemantax.musica.utils;

import android.content.Context;

import android.content.SharedPreferences;

public class PrefManager {

SharedPreferences sharedPreferences;

SharedPreferences.Editor editor;

Context context;

//Shared pref mode

int Private\_mode=0;

// Shared preferenes file name

private static final String pref\_name="welcome\_login";

private static final String is\_first\_time\_launch="IsFirstTimeLaunch";

private static final String is\_Logged\_in="IsLoggedIn";

private static final String userame = "Username";

private static final String userid = "Userid";

public PrefManager(Context context){

this.context=context;

sharedPreferences=context.getSharedPreferences(pref\_name,Private\_mode);

editor=sharedPreferences.edit();

}

public void setFirstTimeLaunch(boolean isFirstTime){

editor.putBoolean(is\_first\_time\_launch,isFirstTime);

editor.commit();

}

public boolean isFirstTimeLaunch(){

return sharedPreferences.getBoolean(is\_first\_time\_launch,true);

}

public void setLoggedIn(boolean islogin){

editor.putBoolean(is\_Logged\_in,islogin);

editor.commit();

}

public boolean isLoggedIn(){

return sharedPreferences.getBoolean(is\_Logged\_in,false);

}

public void setUserame(String Username){

editor.putString(userame,Username);

editor.commit();

}

public String getUsername(){

return sharedPreferences.getString(userame,"");

}

public void setUserid(String userid){

editor.putString(userid,userid);

}

public String getUserid()

{

return sharedPreferences.getString(userid,"");

}

}

RetrofitUtils.java

package com.hemantax.musica.utils;

import com.hemantax.musica.model.Genre;

import com.hemantax.musica.model.Song;

import java.util.List;

import okhttp3.MultipartBody;

import okhttp3.RequestBody;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.Retrofit;

import retrofit2.converter.gson.GsonConverterFactory;

import retrofit2.http.Field;

import retrofit2.http.FormUrlEncoded;

import retrofit2.http.GET;

import retrofit2.http.Multipart;

import retrofit2.http.POST;

import retrofit2.http.Part;

import retrofit2.http.Query;

import retrofit2.http.Streaming;

import retrofit2.http.Url;

public class RetrofitUtils {

public static RetrofitService instance;

public static String BASE\_URL = "http://192.168.43.237:8080/Music/";

public static RetrofitService getInstance(){

if (instance == null) {

instance = new Retrofit.Builder()

.baseUrl(BASE\_URL)

.addConverterFactory(GsonConverterFactory.create())

.build()

.create(RetrofitService.class);

}

return instance;

}

public interface RetrofitService{

@POST("insertData.jsp")

@FormUrlEncoded

Call<String> register(

@Field("name")String name,

@Field("username")String username,

@Field("password")String password,

@Field("email")String email

);

@POST("Login.jsp")

@FormUrlEncoded

Call<String>login(@Field("username")String username,

@Field("password")String password);

@POST("upload.jsp")

@FormUrlEncoded

Call<String>UploadMusicDetails(@Field("User\_name")String User\_name,

@Field("Music\_Name")String Music\_Name,

@Field("Artist\_Name")String Artist\_Name,

@Field("Genre")String Genre

);

@GET("getgenre.jsp")

Call<List<Genre>>getGenres();

@GET("getmusic.jsp")

Call<List<Song>>getMusic(@Query("genre\_id")int id);

@GET

@Streaming

Call<ResponseBody> getFile(@Url String url);

@Multipart

@POST("UploadServlet")

Call<ResponseBody>uploadMusic(

@Part MultipartBody.Part music,

@Part MultipartBody.Part username,

@Part MultipartBody.Part musicname,

@Part MultipartBody.Part artisname,

@Part MultipartBody.Part genre

); } }

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.hemantax.musica">

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".activity.MainActivity"

android:label="@string/app\_name"

android:theme="@style/AppTheme.ActionBar.Transparent" />

<activity android:name=".activity.LoginActivity">

</activity>

<activity android:name=".activity.RegistrationActivity" />

<activity android:name=".activity.PlaylistActivity" />

<service

android:name=".services.MusicService"

android:enabled="true"

android:exported="true"></service>

<activity android:name=".activity.SplashScreen"

android:theme="@style/AppTheme.ActionBar.Transparent">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

Build.gradle

apply plugin: 'com.android.application'

android {

compileSdkVersion 26

defaultConfig {

applicationId "com.hemantax.musica"

minSdkVersion 21

targetSdkVersion 26

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

allprojects {

repositories {

maven { url 'https://jitpack.io' }

}

}

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'com.android.support:appcompat-v7:26.1.0'

implementation 'com.android.support:support-v4:26.1.0'

implementation 'com.android.support:design:26.1.0'

implementation 'com.android.support:recyclerview-v7:26.1.0'

implementation 'com.android.support:cardview-v7:26.1.0'

implementation 'com.android.support.constraint:constraint-layout:1.0.2'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.1'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'

//retrofit

implementation 'com.squareup.retrofit2:retrofit:2.3.0'

implementation 'com.squareup.retrofit2:converter-gson:2.3.0'

implementation 'com.google.code.gson:gson:2.8.2'

//butterknife

implementation 'com.jakewharton:butterknife:8.8.1'

annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'

//picasso

implementation 'com.squareup.picasso:picasso:2.5.2'

}

**Future Enhancements:**

* Development of “Buy & Sell” music section where users can actually buy or sell their music.
* Implementation of Remove music & Delete Account feature.

**REFERANCES**

* Stackoverflow
* Android Hive
* Future Stud
* GitHub
* Android Docs
* The Android Arsenal