

Design and Implementation of Music Album Management System

Sachin Singh¹, Yash Rane¹, Tarang Ahuja¹, Dhanashree Toradmalle², Dhwaniket Kamble²

¹Student, Information Technology, Shah and Anchor Kuttchi Engineering College, Mumbai, Maharashtra, India

²Faculty, Information Technology, Shah and Anchor Kuttchi Engineering College, Mumbai, Maharashtra, India

ABSTRACT

Article Info

Volume 6, Issue 6

Page Number: 328-333

Publication Issue :

November-December-2020

Article History

Accepted : 15 Dec 2020

Published : 30 Dec 2020

Music Album Management System is a fully functional, flexible and convenient application and a user friendly interface which provides a good experience. This paper describes the phases of the design and development and emphasizes the system's functionality.

Keywords : Music Album Management System, Database, JDBC, Information Management

I. INTRODUCTION

The aim of this project is to develop a simple, easy to use and centralized music album management system. This application will store the information of customers, artists, songs, albums, movies, etc. The application is built for users to search songs and albums based on various searching parameters. Upon searching for a song the user will be provided with a non-audio based output such as artist name, album name, movie name, song length and other details.

In some of the preexisting systems, the database is maintained by the users, which results in unverified information to be added to the database, thus giving scope for errors. For these systems, the problem was tackled by giving all the privileges to admin for maintaining the database in place of users. Though, implementing this solution solves the previous problems, but created a new problem where the database didn't get updated regularly. To overcome

this challenge, our application consists of an admin-user coexistence where the user can suggest new songs and albums which can be updated into the database upon verification by the admin. The admin manages all the entries including customer details and can block a customer in case of any suspicious activity.

II. TECHNOLOGY STACK

A. Java Swing

Java Swing is a GUI toolkit which provides various widgets such as checkboxes, radio buttons, scroll bar menus, etc to build and design the webpage and to provide a simple interface to the user. The GUI is built in Java using Javax.swing.* package.

B. MySQL

MySQL is a relational database management system, which is a new generation of data management and analysis software owned by Oracle. It is a

multipurpose software which is used for data warehousing, logging applications and e-commerce.

C. JDBC

The connection between the GUI and database is done with the help of JDBC using which we can call SQL query into programs written in java language.

III. ARCHITECTURE OF DATABASE

A. Entity Relationship Diagram

The Entity Relationship Diagram describes the entities ADMIN (shown in Figure 1), CUSTOMER (shown in Figure 2), ALBUM (shown in Figure 3), MOVIE (shown in Figure 4), ARTIST (shown in Figure 5), YEARS (shown in Figure 6), SONG (shown in Figure 7) and PLAYLIST(weak entity) (shown in Figure 8).

Each entity has different attributes which are shown in circles. Also, every entity except PLAYLIST has a primary key which is underlined.

A CUSTOMER can search SONGS directly or using filters like MOVIE or ALBUM or YEARS or ARTIST. Also, a CUSTOMER can create its own PLAYLIST (not necessary hence, kept as a weak entity).

An ADMIN manages all the activities taking place in the database. The relationships like manages, has, access, creates and ARTIST-SONG connect the entities in a structured and simple manner.

Mapping cardinalities are explained below:

- One ADMIN can manage many CUSTOMER, many ALBUM, many MOVIE, many ARTIST, many YEARS and many SONG.
- One CUSTOMER can access many ALBUM, many MOVIE, many ARTIST, many YEARS and many SONG.
- Many PLAYLIST can be created by one CUSTOMER.

- Many SONG can be released in one YEARS..
- Many SONG can be a part of one ALBUM..
- Many SONG can belong to one MOVIE.
- Many ARTIST may have sung many SONG.

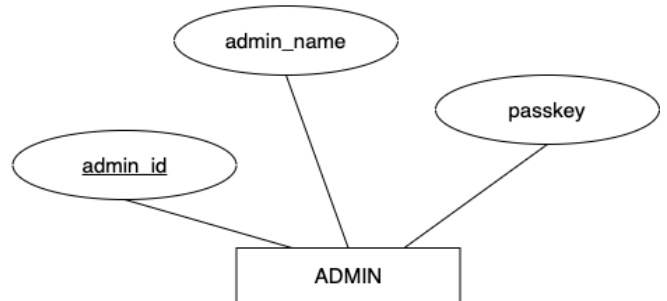


Figure 1. Admin Entity in E-R diagram

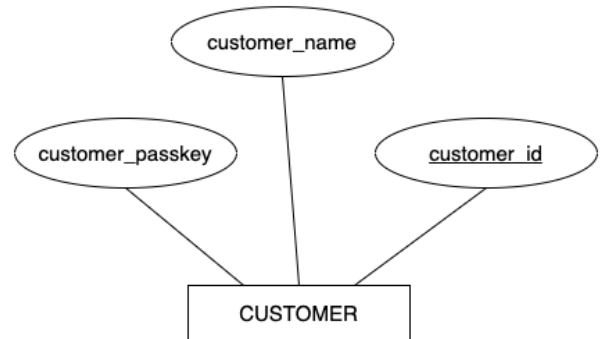


Figure 2. Customer Entity in E-R diagram

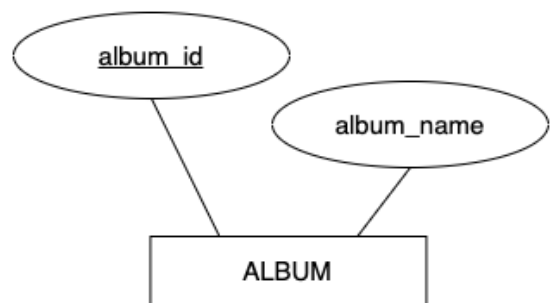


Figure 3. Album Entity in E-R diagram

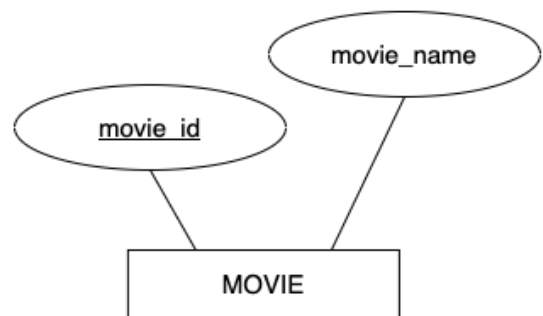


Figure 4. Movie Entity in E-R diagram

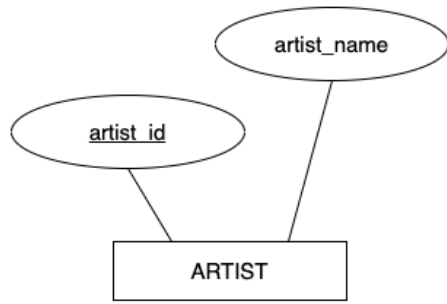


Figure 5. Artist Entity in E-R diagram

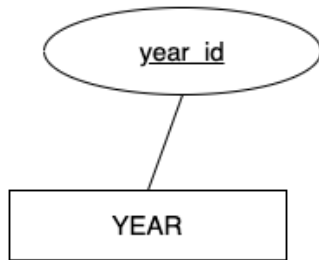


Figure 6. Year Entity in E-R diagram

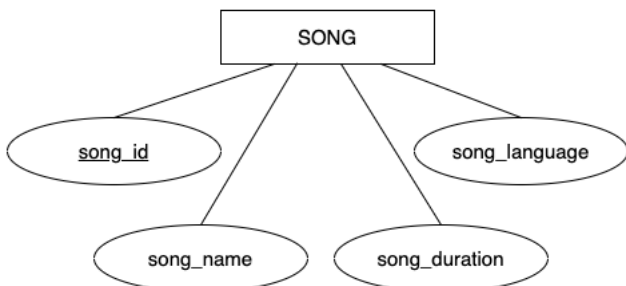


Figure 4. Song Entity in E-R diagram

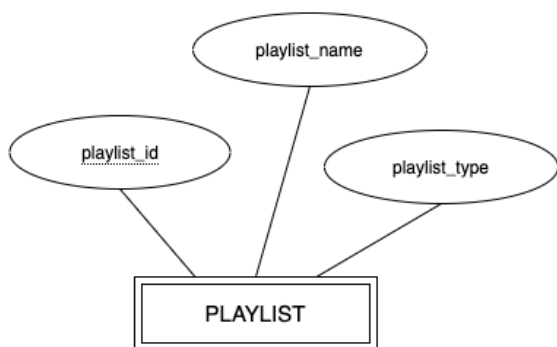


Figure 8. Playlist Entity in E-R diagram

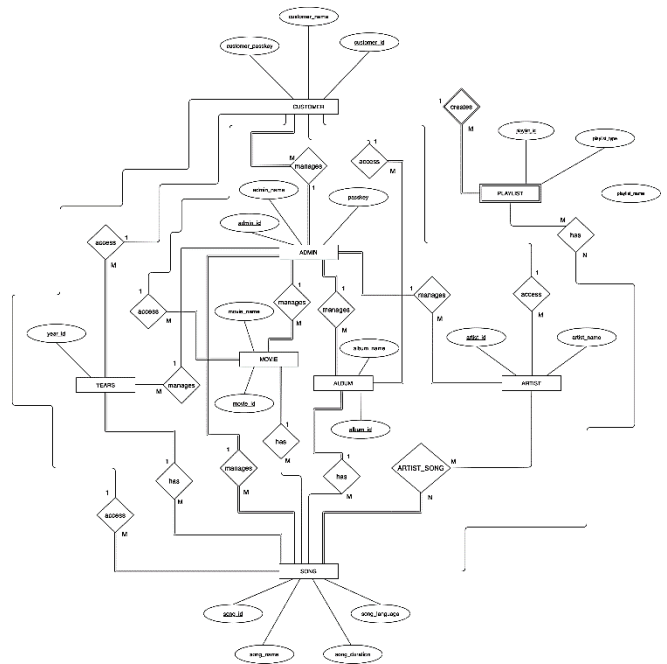


Figure 9. Entity Relationship Diagram

IV. Implementation of Music Album Management System

The implementation consists of the three following main components

A. Database

We created the database based on the entity relationship diagram of section III. The creation of the tables is made using MySQL.

We have created eight tables total, seven for the entities and one table ARTIST_SONG for the relation between entities ARTIST and SONG, since they have many to many relationships.

B. Java Database Connectivity

The connection between the GUI and database is done using JDBC. We have downloaded a jar file “mysql-connector-java-8.0.21.jar” and set the classpath in the reference libraries. This jar file consists of classes which help in connecting the application with MySQL workbench. Several DDL and DML commands can be performed using JDBC.

In our application we are using JDBC in registration of user, login of user, login of admin where the userId and password of the both user and admin is stored in database, which later help to access the main content. The song table in GUI is retrieved from the database using JDBC. Using JDBC, it is possible to access the Database and perform various DML commands such as create, update or delete.

C. GUI

1. Homepage: Homepage (shown in Figure 10) is the introductory page which gives the overview of the application. Several operations can be performed such as Admin login, User login and registration.

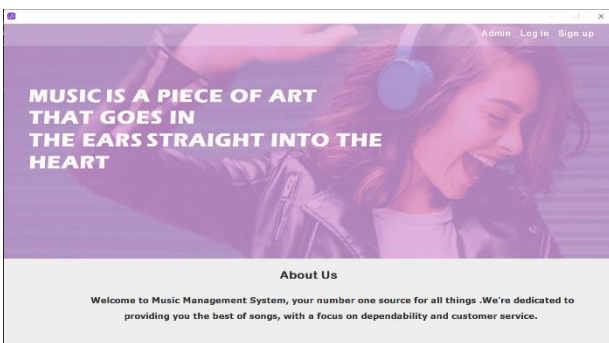


Figure 10. Home Page

2. Registration: Registration is the first step in application development to store the data of users. This is implemented using the home page window on clicking the sign up tab. On clicking on it, the Sign up window appears and the user can register by filling username, name and password.
3. Login (User): After registration users can login to the main content by using correct username and password. After login, the main window (shown in Figure 12) appears where users can search a song according to its choice. Searching can be done on the basis of artist, movies, year and album.

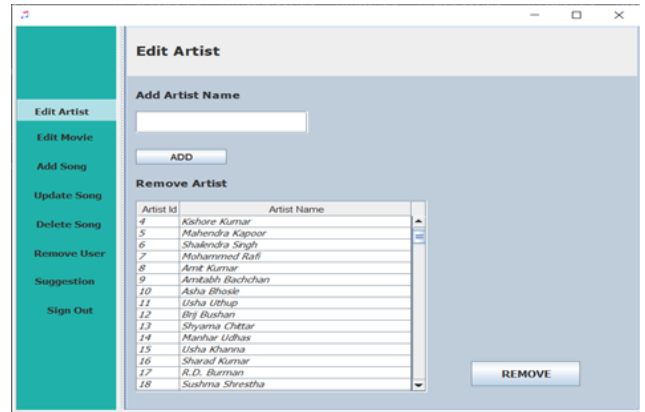


Figure 11. Admin Interface

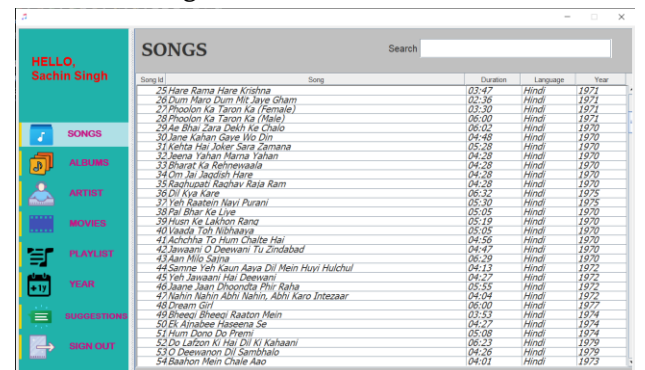


Figure 12. User Interface

4. Login (Admin): Admin interface (shown in Figure 11) consists of add song, edit movie, edit artist, update song, delete song, remove user and suggestion tabs. These help the admin to make changes in the database using GUI.

V. Use Cases

A. Suggestion Page

A user can suggest (Shown in Figure 13) admin to make certain changes in GUI. He/She can request admin to add song, update song, update artist and movie by giving suggestions in the suggestion tab and the same will be displayed to admin in admin interface (Shown in Figure 14).



Figure 13. User Suggestion Page

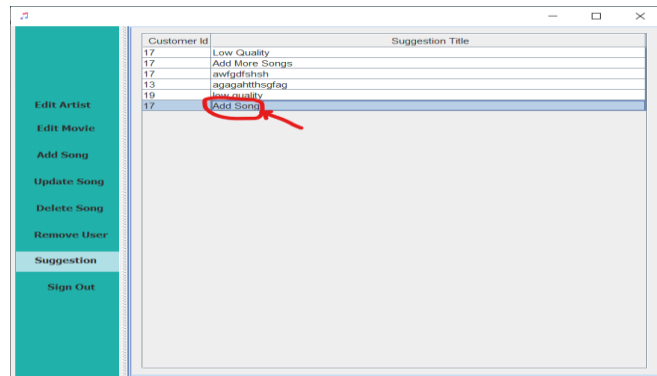


Figure 14. Admin Suggestion Interface

B. Adding Songs to playlist

A user can add songs to a playlist by clicking on “add to playlist” (Shown in figure 15) and the song is added to the playlist which can be accessed by clicking on the PLAYLIST tab. (Shown in figure 16).

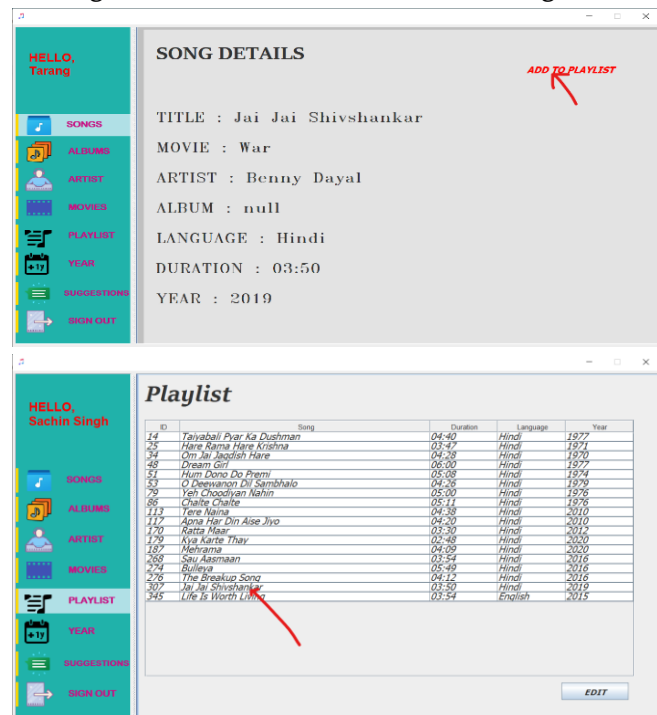


Figure 16. Playlist Tab

VI. CONCLUSION

As stated in the introduction the aim of our project was to solve the problems faced in the pre existing system. During the implementation, we successfully created a user-friendly application which can help users to search for songs and albums based on various searching parameters and also added a suggestion page which can help the admins to add songs and albums as per the users liking and maintain a database with higher authenticity.

VII. REFERENCES

- [1] Ruud Stegers, Peter Fekkes and Heiner Stuckenschmidt. MusiDB: A personalized search engine for music
- [2] Alexandros Nanopoulos, Dimitrios Rafailidis, Maria M. Ruxanda and Yannis Manolopoulos. Music search engines: Specifications and challenges
- [3] Maria M. Ruxanda, Alexandros Nanopoulos, Christian S. Jensen and Yannis Manolopoulos. RANKING MUSIC DATA BY RELEVANCE AND IMPORTANCE
- [4] Robert D. Taub, J. Alexander Cabanilla and George Tourtellot. Music Based Search Engine
- [5] F. B Abdullahi, J. C Kisha and T. Hassan. Design and Implementation of A Web Based Music Portal
- [6] TANG Yu-fang and ZHANG Yong-sheng. Design and Implementation of Student Information Management System Based on Web Services
- [7] David A.Goldberg, Ali Orooji. A Symmetrical Approach to Granting and Revoking Access rights in Database Management Systems
- [8] Parixit Dilip Parekh. Java Applet Based Database Management Interface
- [9] Ying Bai. JDBC API and JDBC Driver. In Practical database programming with Java.

- [10] M. Swain, J. A. Anderson, R. Korrapati, N.K. Swain. Database programming using Java

Cite this article as :

Sachin Singh, Yash Rane, Tarang Ahuja, Dhanashree Toradmalle, Dhwaniket Kamble, "Design and Implementation of Music Album Management System ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 6, pp. 328-333, November-December 2020.

Available at doi

: <https://doi.org/10.32628/CSEIT206653>

Journal URL : <http://ijsrcseit.com/CSEIT206653>