Software Engineering Lab Project
Code Specification

Archisman Pathak
18CS30050

2020-05-23

1 Code Link

The entire code can be accessed from the following link :

`https://github.com/TheThinker01/Sangit`

The details of running the code are present in **README.pdf** submitted.

2 Front End Interface

The Following front end interfaces are present in Sangit:

1. This project uses HTML, CSS, Bootstrap, Javascript and JQuery to implement the front end interfaces.

2. **Homepage** : Home page houses the Popular songs ,User's Playlists, Song Queue, and links to raise issue,Admin Panel and Music Provider Interface

3. **My Playlists** : The logged in user's Playlists and also shared Playlists of other users are shown here

4. **My Queue** : The user's songs queue

5. **AJAX** - Asynchronous Javascript and XML has been used for transferring server data to client side without page reloads. This has been

used to fetch the Playlist's songs and for implementing song search functionality

6. **Music Player** - The Music Player interface present at the bottom of the homepage is built from scratch using JQuery and Web Audio API.

7. **Raise Issue** : A separate page to file complains to the admin as specified in the SRS.

8. **Navigation bar** : Has the buttons for searching songs and also auxiliary profile buttons.

9. **Admin Panel** : Only admin can access this site. This site contains information about all the data stored in the database using seperate webpages , only visible to the admin.

10. **Music Panel Interface** : Only accessible by the music providers. Here they can view all their songs and playlists that they have added and suitably modify them.

3 Backend Functionalities

The following Backend Functionalities have been implemented.

1. This project uses JAVA EE , Hibernate,Apache Tomcat 7 and MySQL as backend technology stack and maven has been used to manage dependencies and build the project.

2. For every frontend-interface mentioned above a corresponding servlet has been created which handles the **HttpRequest** and generates **HttpResponse**. For example there is a servlet for handling login data, one for creating users, one for adding music and so on.

3. MVC i.e. Model-View-Controller pattern has been strictly followed in the project. All access to the database is limited to the Data access objects (DAO) which have their own methods for querying and adding data.

4. The use of Hibernate Query Language(HQL) and hibernates session methods ensure all data operations are implemented safely and attacks like **SQL Injection is avoided**.

5. The JSP's form the view and have no direct access to the database. All access is routed through the controllers which in turn use the DAO's. This greatly enhances the security and stability of the application.

6. Finally the Servlets implemented serve as the controllers validating all user input and requests and providing the desired data.

4 Database and Networking Involved

1. 4 tables have been created in the database, namely - Music, Issues, Playlists and Users. A number of relationships exists between these tables which have been implemented by ManyToMany and ManyToOne mappings as appropriate.

2. Hibernate has been used as an Object Relational Mapping framework and MySql has been used as the database engine.

3. Sangit is a web application and hence inherently has networking involved. All the requests are sent to Servlets as HttpRequest objects and are processed accordingly by them finally returning HttpResponse objects.