

Live-Sync

(A Real Time Music Synchronization platform)

End-Semester Project BE-III (Semester-6)

Advanced Java Technology

Students:

(name:prn)

1. Bhadrika Raval : 8021004342
2. Mihir Bhavsar : 8022057037
3. Harshvardhan Shinde : 8021079203

Teacher : Prof. Dr. Mamta Padole

Google Drive link:

Live-Sync

(Music Collaborator – Real Time Music Synchronization)

Project Description:

The Collaborative Music Jam is a web-based application that enables users to create and join real-time collaborative music sessions. This platform allows multiple users to synchronize their music playback, control the playback together, chat with each other, and create a shared listening experience. The primary goal of this project is to provide a seamless and engaging platform for music enthusiasts to discover, enjoy, and interact with music in a social and collaborative manner.

Playback Controls: Participants in a session can control various aspects of the playback, such as play/pause, volume adjustment, seeking within the track, and voting on the next song to be played.

Real-time Collaboration: Users can create or join collaborative sessions, where their music playback is synchronized across all participants. Everyone in the session has control over the playback, enabling a shared listening experience.

Technology Used:

Spring Boot:

This is used for the backend development. Spring Boot simplifies the process of creating stand-alone, production-grade Spring-based applications.

In the backend server application to handle HTTP requests, manage sessions, and interact with the database.

Database Configuration:

Here the non sql database -Mongodb Atlas (cloud mongodb) used here for the data storage.

JavaScript, HTML, CSS:

These are used for frontend development. JavaScript provides interactivity to web pages, HTML structures the content, and CSS styles the presentation.

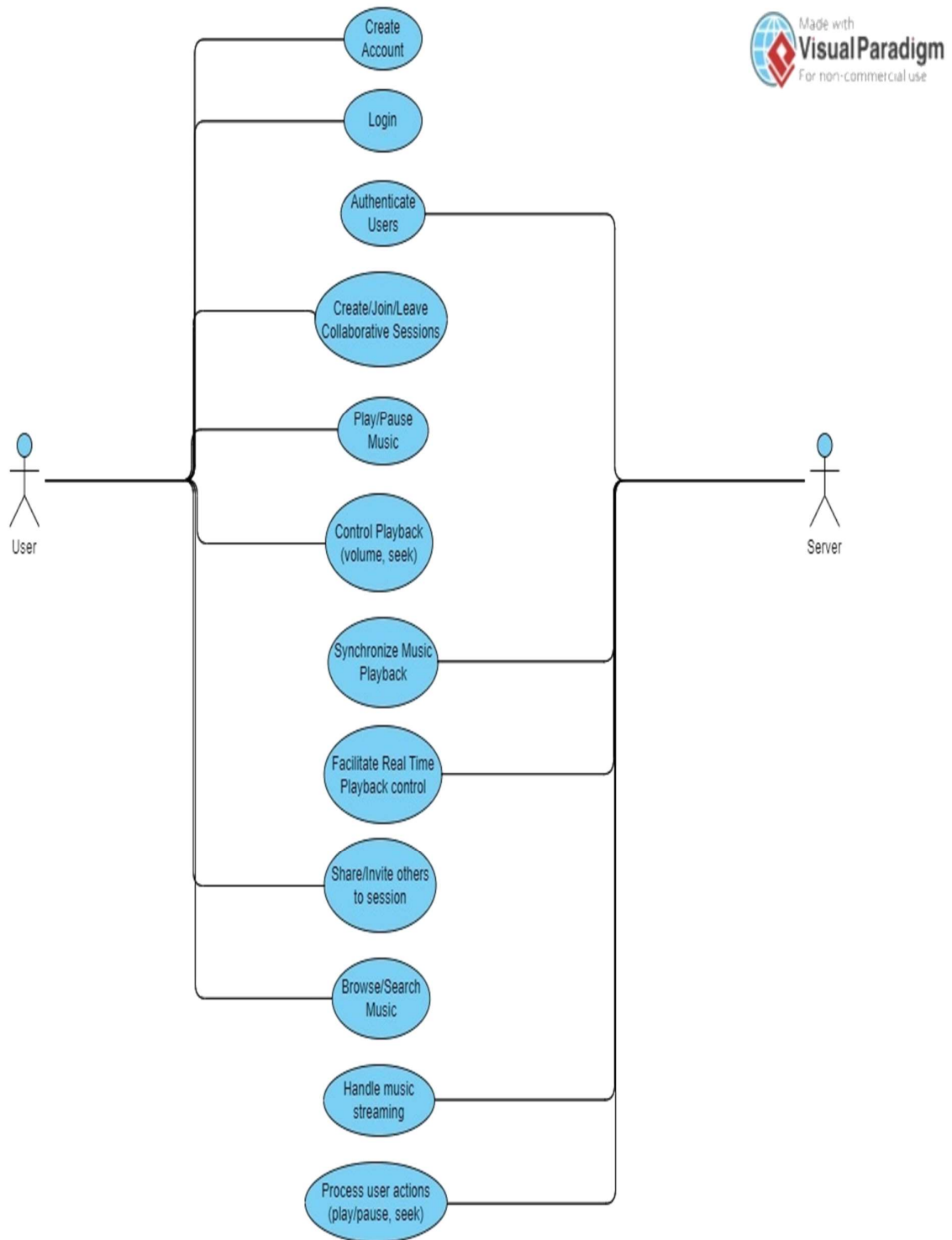
In the frontend web application to create the user interface, handle user interactions, and style the application.

TCP/IP Communication:

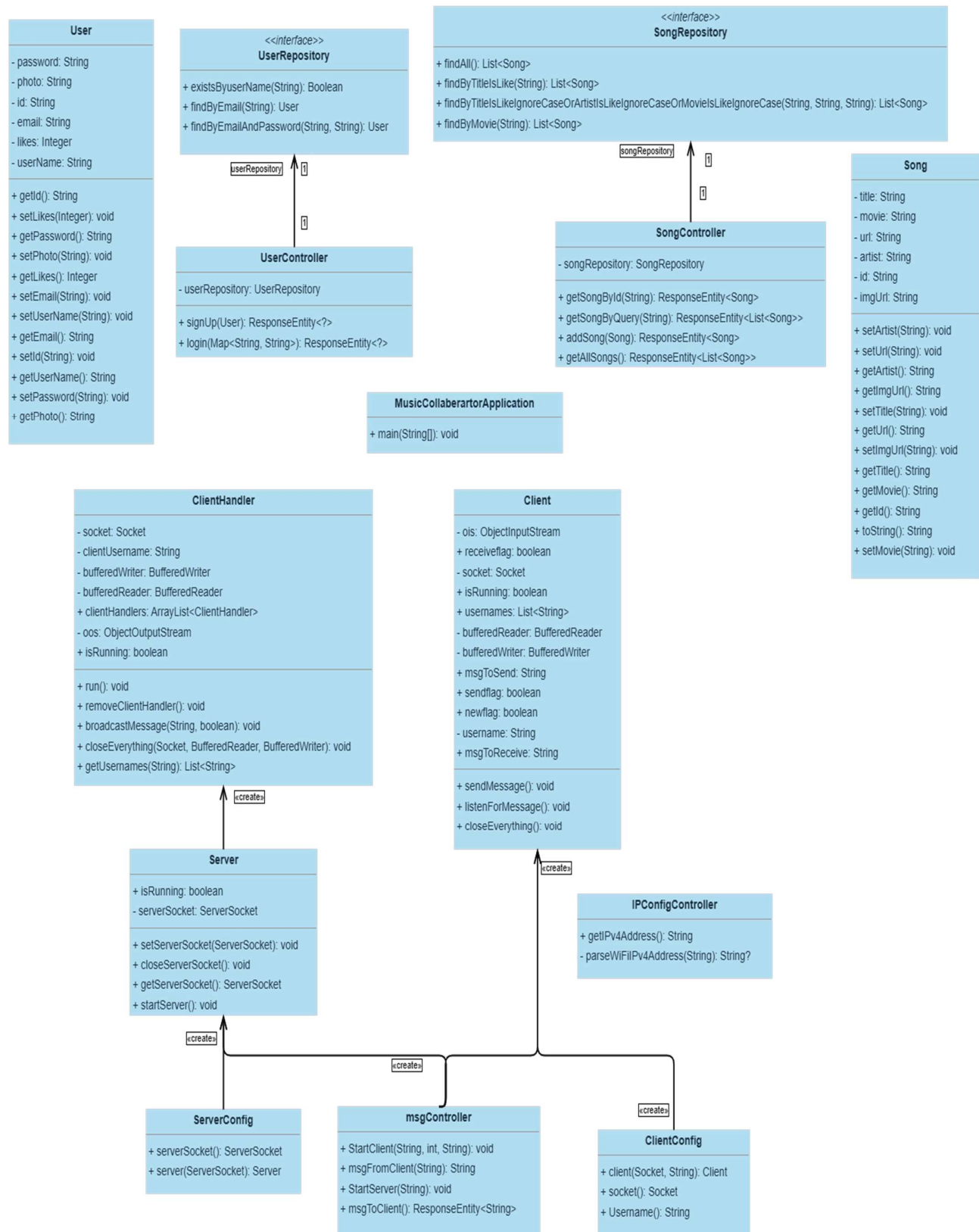
This is used for network communication between the client and server. TCP/IP is a set of protocols that govern how data is transmitted over the internet.

Where it's used: In the backend server application to communicate with clients over the network using TCP/IP sockets. This can be implemented using the Java java.net package.

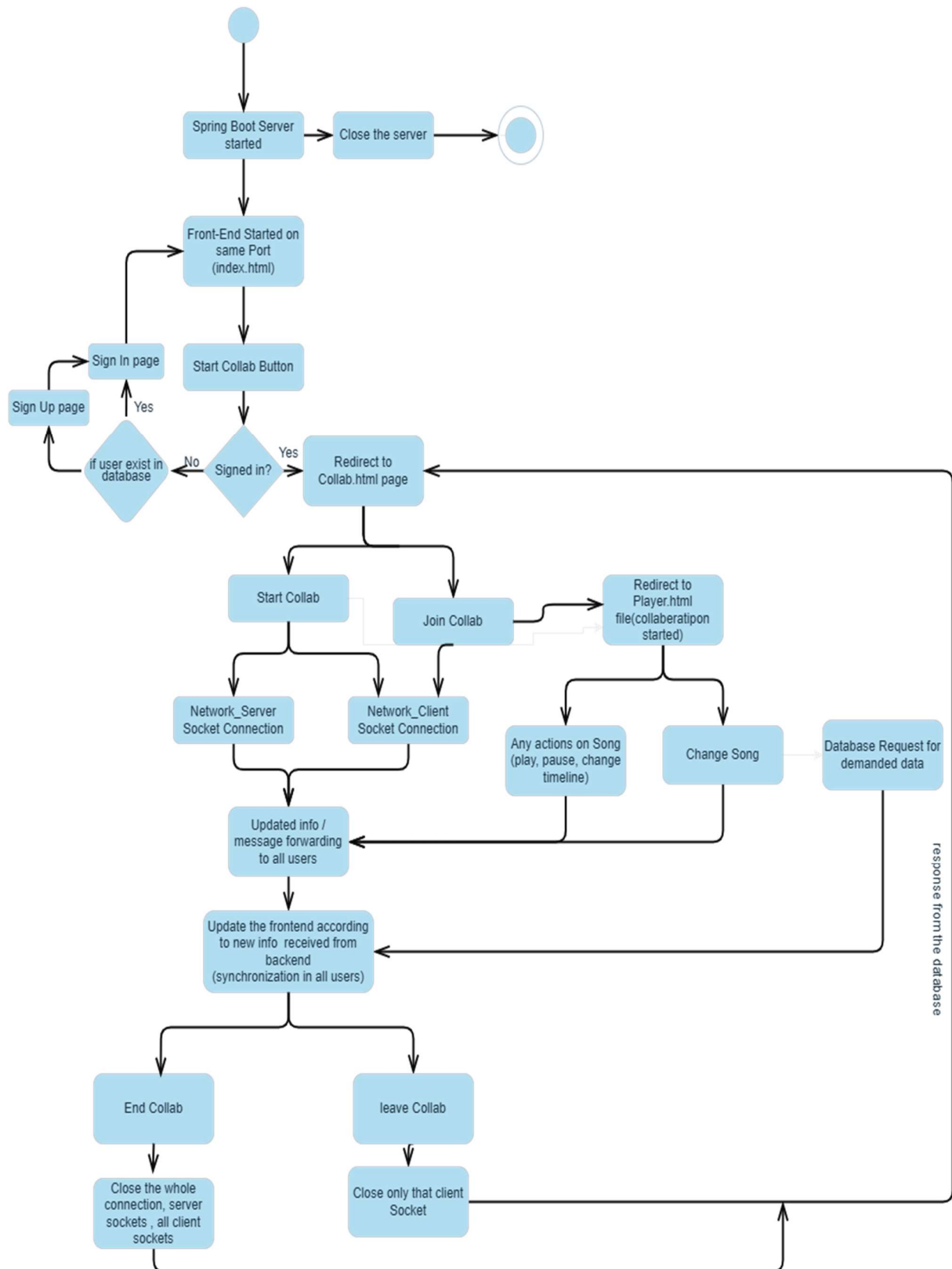
Use-Case Diagram:



Class Diagram:



Activity Diagram:



Implementation Details:

1. Backend Development with Spring Boot:

- Utilized Spring Boot framework for building the backend server application.
- Implemented RESTful APIs to handle client requests for user authentication, music collaboration sessions, and song management.
- Leveraged Spring Data JPA for seamless interaction with the underlying database, ensuring efficient data persistence and retrieval.
- Ensured scalability and robustness of the backend application architecture to support concurrent user interactions and collaborative music sessions.

2. Frontend Development using JavaScript, HTML, and CSS:

- Designed a responsive and intuitive user interface using HTML for structure, CSS for styling, and JavaScript for dynamic behavior.
- Implemented client-side functionalities for user authentication, session management, and real-time music collaboration.
- Utilized AJAX requests to interact with backend APIs asynchronously, ensuring smooth user experience and seamless data exchange between frontend and backend components.
- Ensured cross-browser compatibility and accessibility to accommodate a wide range of user devices and preferences.

3. Network Communication using TCP/IP and Java `java.net` package:

- Established TCP/IP communication channels between the frontend client and backend server for real-time data exchange.
- Implemented socket programming using the Java `java.net` package to enable bidirectional communication between clients and the server.
- Ensured reliable and efficient data transmission over the network, minimizing latency and optimizing performance for collaborative music sessions.
- Implemented error handling mechanisms to gracefully handle network disruptions and ensure seamless operation under varying network conditions.

Test Use Cases:

1. User Authentication:

- Test Case 1: Verify that existing users can successfully sign in with valid credentials.
- Test Case 2: Validate that new users can register/sign up with unique usernames and passwords.
- Test Case 3: Ensure that invalid credentials result in appropriate error messages and prevent unauthorized access.

2. Collaborative Music Sessions:

- Test Case 4: Confirm that users which are joining the collaboration are authenticated
- Test Case 5: Confirm that users can initiate/join collaborative music sessions without errors.
- Test Case 6: Validate real-time synchronization of music playback and control actions across all participants.
- Test Case 7: Verify that users can search for songs and play them synchronously within the collaborative session.
- Test Case 8: Verify that users can enter the session at the mid of any session and can leave also.

3. Error Handling and Resilience:

- Test Case 9: Simulate network disruptions and validate the application's behavior in handling connection timeouts and reconnection attempts.
- Test Case 10: Ensure graceful degradation of functionality under degraded network conditions, such as packet loss or high latency.
- Test Case 11: Verify that error messages are displayed appropriately for failed network operations or backend service errors.

4. Cross-Browser Compatibility and Accessibility:

- Test Case 12: Validate the application's appearance and functionality across different web browsers (e.g., Chrome, Firefox, Safari).

User Manual:

Sign Up:

Visit the Live-Sync website and click on the "Sign Up" button. Follow the prompts to create your account by providing the necessary information, such as your email address and a secure password.

Login:

Once you have created your account, you can log in to the application using your email and password.

Creating a Collaborative Session:

Start a New Session: After logging in, you will be directed to the main dashboard. Click on the "Start Collaboration" button to initiate a new collaborative session.

Invite Friends: To invite others to your session, click on the "copy button" button. You can share the session link with your friends via email, messaging apps, or social media platforms.

Joining a Collaborative Session

Join a Session: Click on the "Join" button next to the session you wish to join. you will need to enter the session invite link or code provided by the host.

Session Interface

Once you have joined a collaborative session, you will be presented with the session interface, which includes the following components:

Music Player: The music player displays the currently playing track and allows you to control the playback using the play/pause, volume, and seek controls.

Participants List: The participants list displays the names or usernames of all users currently in the session.

Music Library: You can search, browse, and add songs to the session queue.

Session Controls

Play/Pause: Use the play/pause button to control the music playback for the entire session.

Volume: Adjust the volume using the volume slider or buttons.

Seek: You can seek forward or backward within the currently playing track using the seek bar or buttons.

Future Add-Ons to Enhance User Experience:

Chat and Communication: During a collaborative session, users can engage in real-time text-based chat, facilitating discussions, sharing thoughts, and enhancing the overall social experience.

User Profiles and Playlists: Users can create personalized profiles, build their own playlists, and share them with others within the application.

Cross-platform Compatibility: The application will be designed to work seamlessly across various devices and platforms, including desktops, laptops, tablets, and smartphones.

Session Settings: Provide a name for your session and optionally set a description. You can also configure session settings, such as making it private or public, setting a maximum number of participants, and enabling voting rules.

Add to Queue: From the music library, you can add songs to the session queue by clicking the "Add to Queue" button next to the desired track.