**Software Requirements and Design Document**

**For**

**Group 3**

**(WeShed)**

Version 2.0

**Authors**:

Eliot S.

Noal G.

Rogelio L.

Steven K.

# **1.** **Overview**

Our project is a web app meant to be used to practice jazz music. The core functionality deals with displaying backing tracks and lead sheets meant to be used to play along with to practice improvisation. The remaining features we will implement include time spent on each track, as well as maintaining a timestamp of the last time each tune was played, so the program is able to recommend songs that the user has not played in a long time. Social and gamification elements like a friends list, achievements and challenges will encourage users to spend more time on the site, and more time practicing.

# **2.** **Functional Requirements**

High priority requirements:  
 Interface for displaying songs/tracks  
 Login/Authentication system  
 Time tracking on each song  
 Search bar for users

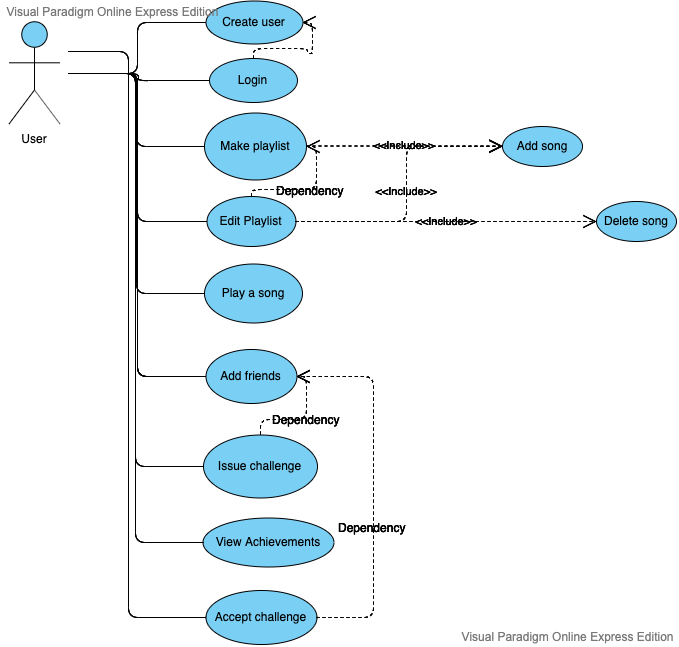
Medium Priority Requirements:  
 Recommendations based on timestamps  
 Friends list  
 Issue/Accept Challenges for time played  
 Notification System

Low Priority Requirements:  
 Streak Tracking

# **3.** **Non-functional Requirements**

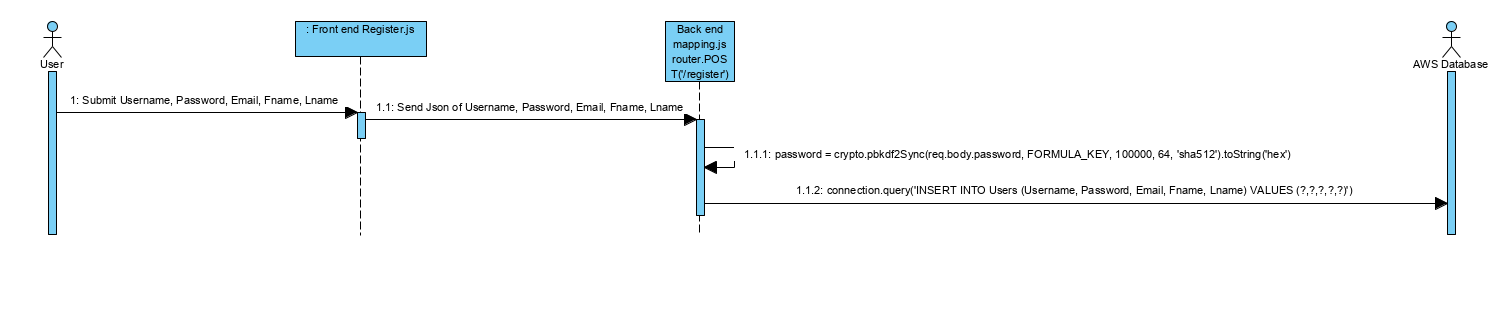
As a result of our personal user login-based system, our project will require some form of data security in order to ensure the safety of all users. Furthermore, general quality assurance is needed for the design, performance, and reliability of the website, both in terms of code and visual design, in order to provide a high-quality experience for the users.

# **4.** **Use Case Diagram**

**

# **5.** **Class Diagram and/or Sequence Diagrams**

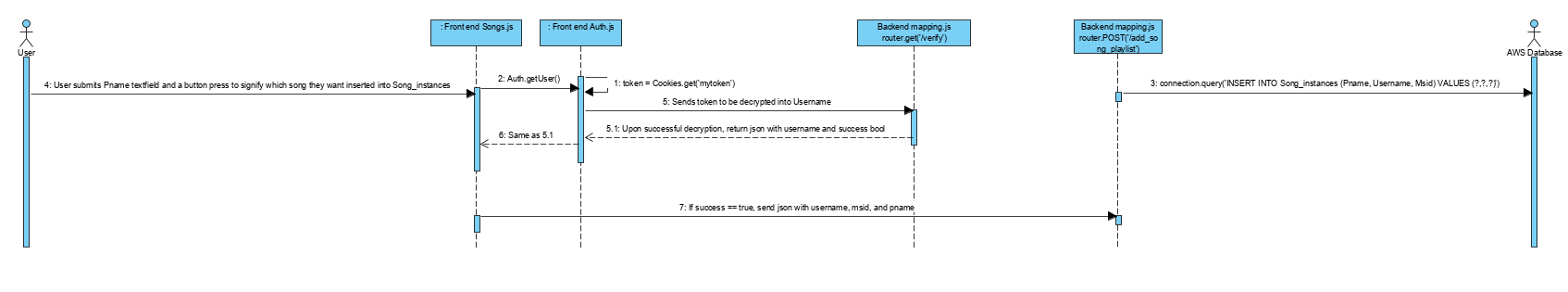
*Create User Sequence Diagram:*

**

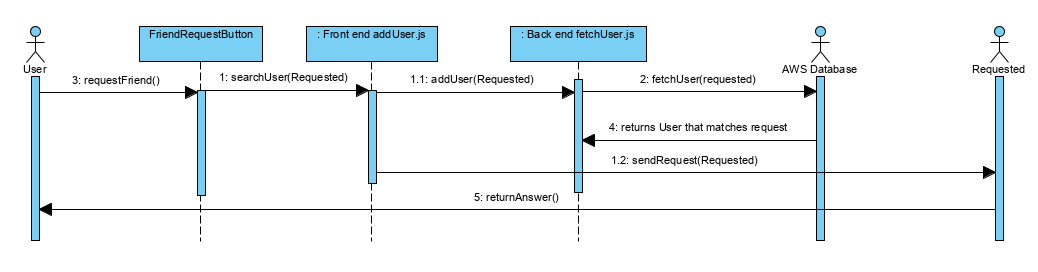
*Login Sequence Diagram:*

**

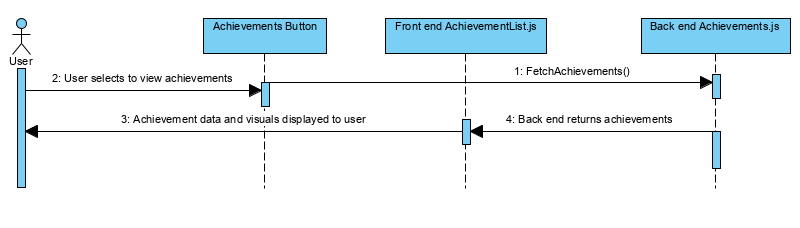
*Add Song Sequence Diagram:*

**

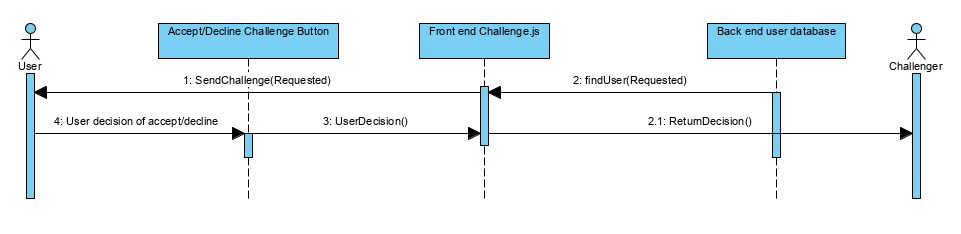
*Friend Request Sequence Diagram:*

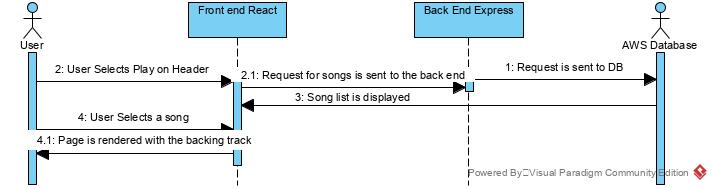
**

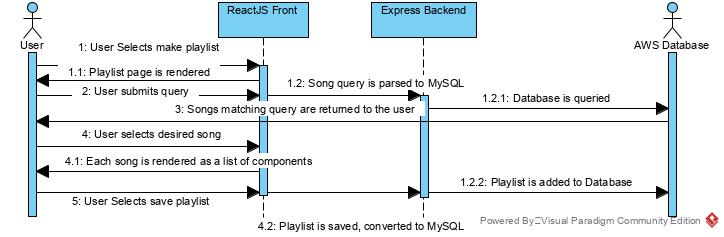
*View Achievements Sequence Diagram:*

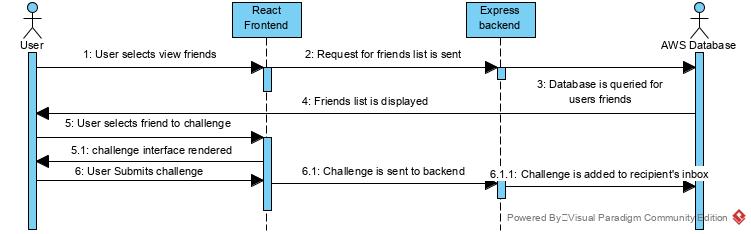
**

*Accept Challenge Sequence Diagram:*

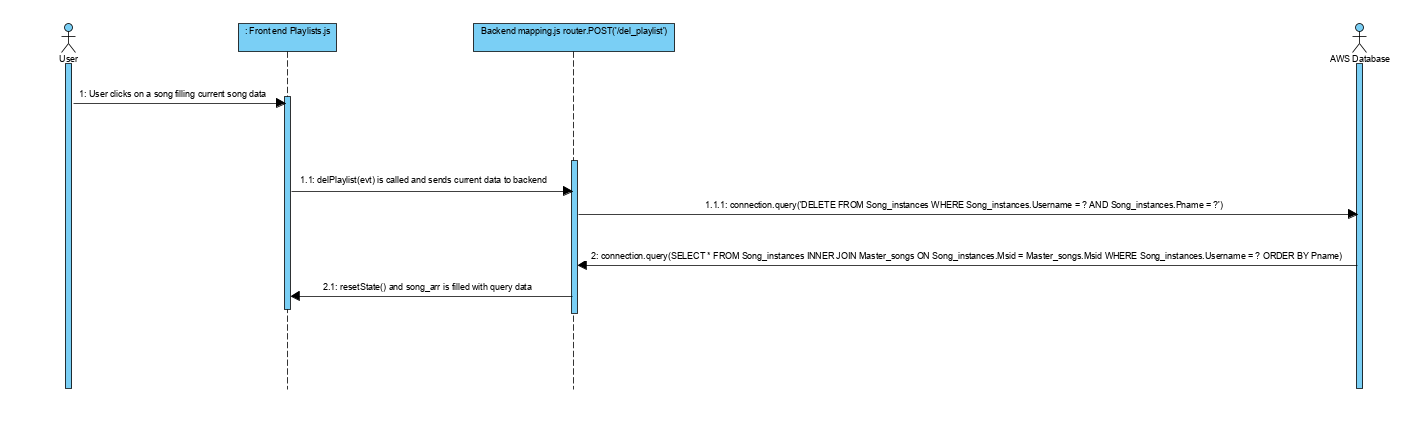
**

*Play Song Sequence Diagram:*

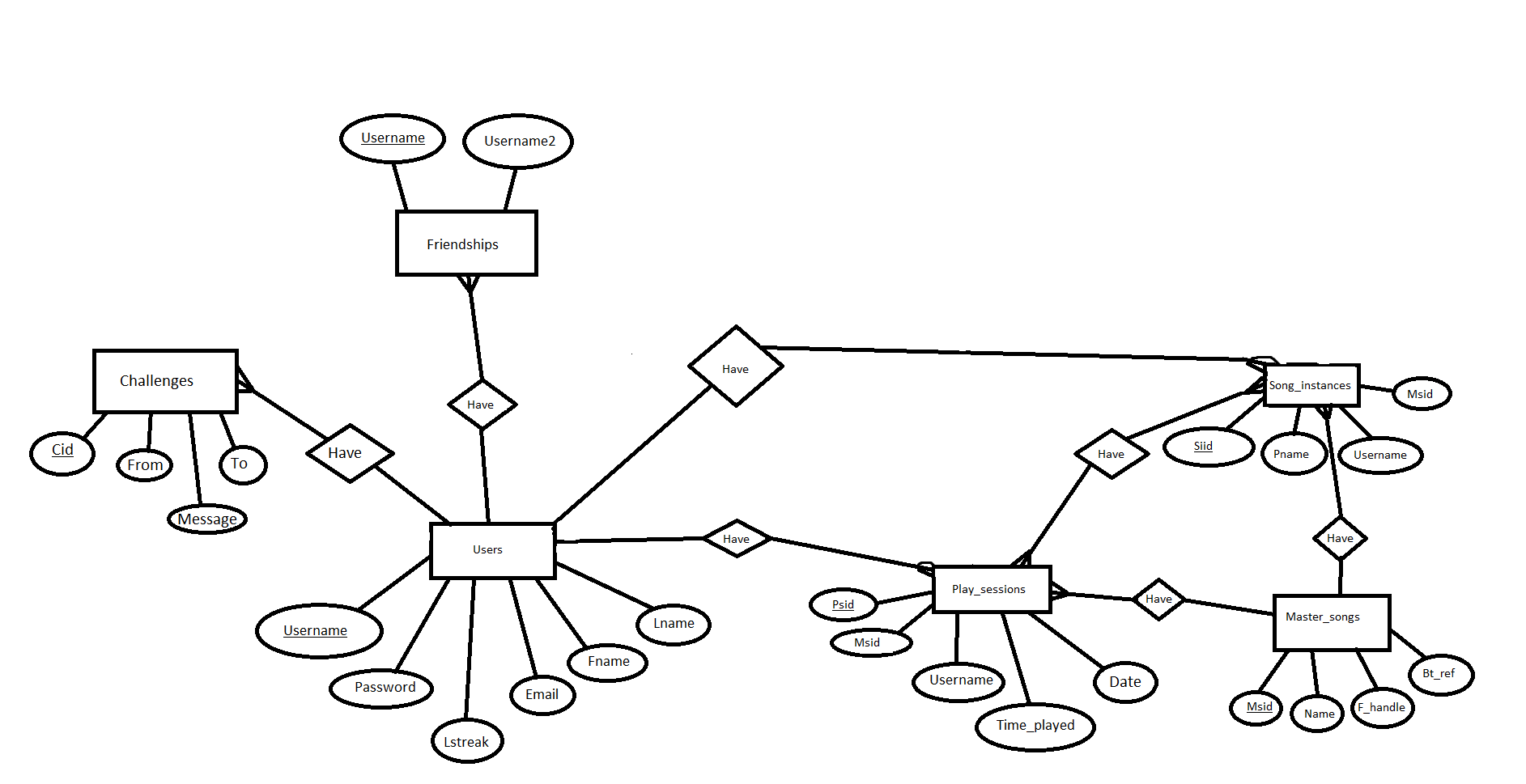
*Make Playlist Sequence Diagram:  
*

*Issue Challenge Sequence Diagram:  
*

*Delete Playlist Sequence Diagram:*

**

*Tentative increment 2 database schema:*

**

# **6.** **Operating Environment**

Our project will be a web application, with the front-end being run in the browser using React.js, and the backend running in a node.js runtime using Express.js and MySQL. For the purposes of this project, it will be assumed that both the front and back ends will be run locally, as opposed to being deployed on a web server.

# **7.** **Assumptions and Dependencies**

It is assumed that the front end, back end, and database will all run simultaneously. There are several dependencies in the project which could halt development entirely. This project depends on using npm start to run the front end and node app.js to run the back end. The back end makes queries or responds to front end requests and relies on an AWS database which is always running. If any one of these falters, the project will not operate as intended. If Cors(Cross origin resource sharing), mysql, jstoken, cookie-js, express, or react become deprecated or fail then the entire project will lose functionality. The mytoken hash in a cookie on localhost serves as a key to the entire website, so if cookies within browsers change during our progression, this could ruin our persistent login system and protected routes.