Login System Design Document

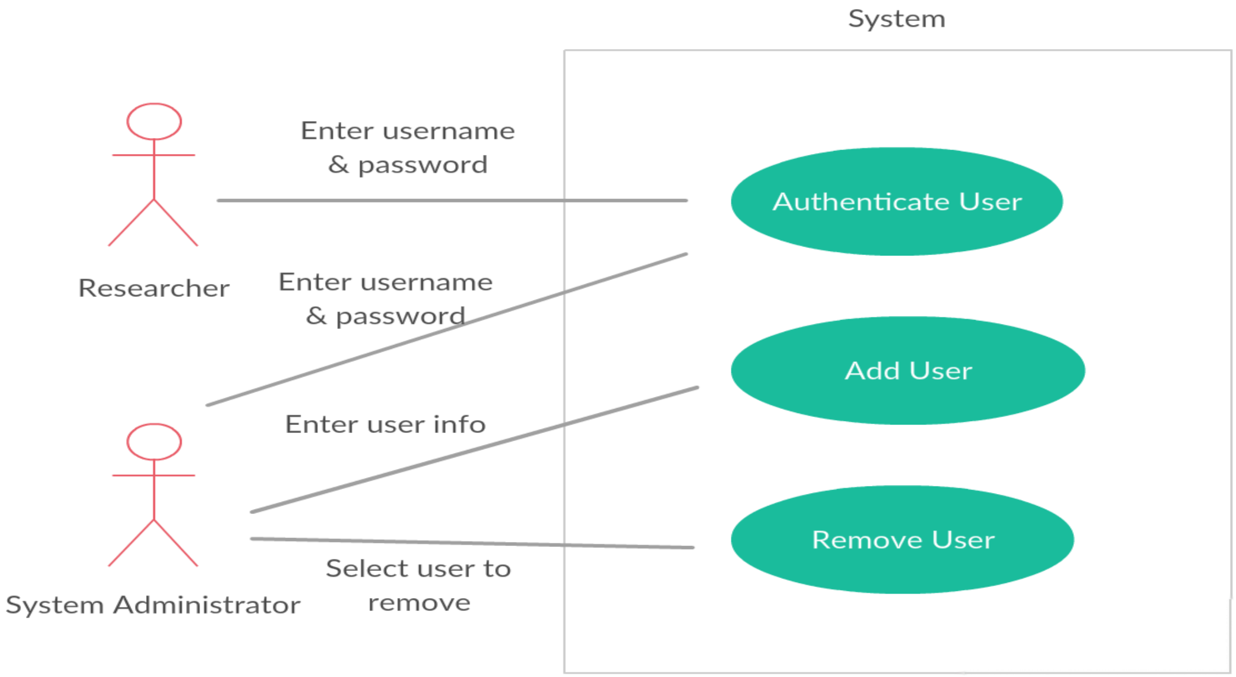
Purpose:

The purpose of this system is to provide an access control mechanism for the application.

System Requirements:

1. **User Model**
   a. **Username**
   b. **Name**
   c. **Email Address**
   d. **Password**
   e. **Admin User**
2. **Database**
   a. **Hash Password before storing**
   b. **Reliable Connection**
3. **User Authentication**
   a. **Login**
   b. **Logout**
4. **User Management**
   a. **Add user (admin only)**
   b. **Delete user (admin only)**
   c. **Change User Model**
5. **Session Management**
   a. **Authenticated Session**

Use Case Diagram for System:



System Technologies:

1. MEAN Stack

   The MEAN stack is a free and open-source, user-friendly full-stack JavaScript based framework for building dynamic websites and web applications. MEAN stack makes use of MongoDB, Express.js, Angular 4, and Node.js in this project.

   a. **MongoDB** is the Database that stores user models.

   b. **Express.js** is the back-end web framework. Hosts Rest API interface to MongoDB.

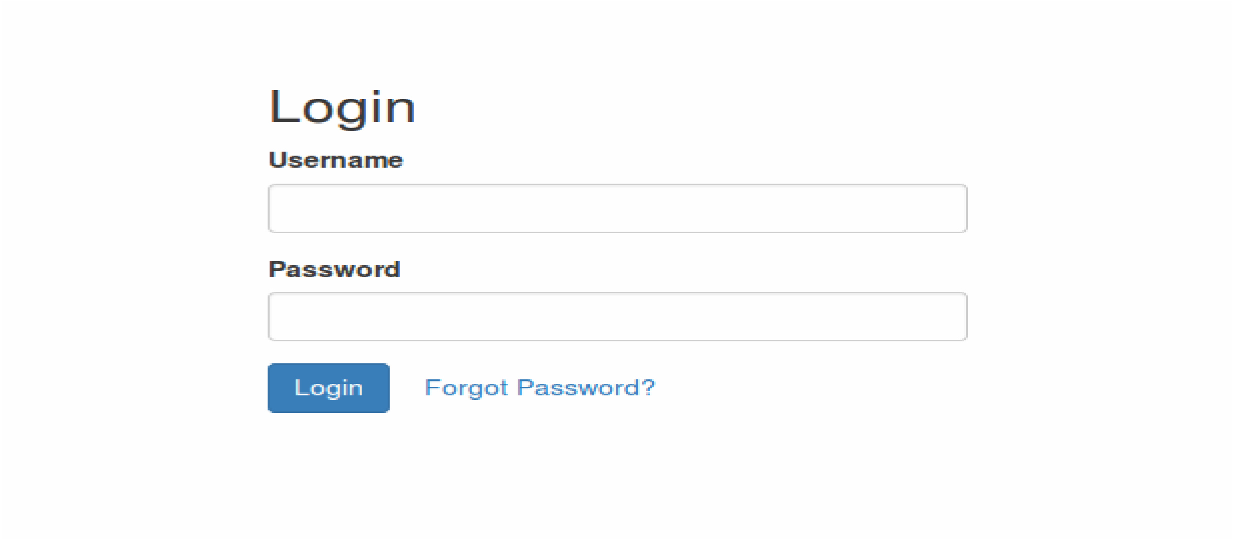
   c. **Angular 4** is the front-end framework, to implement the User Interface. Uses principles of MVC and is a Single page style application. **Uses Typescript, a syntactical superset of JavaScript.**

   d. **Node.js** is the server and web application platform.

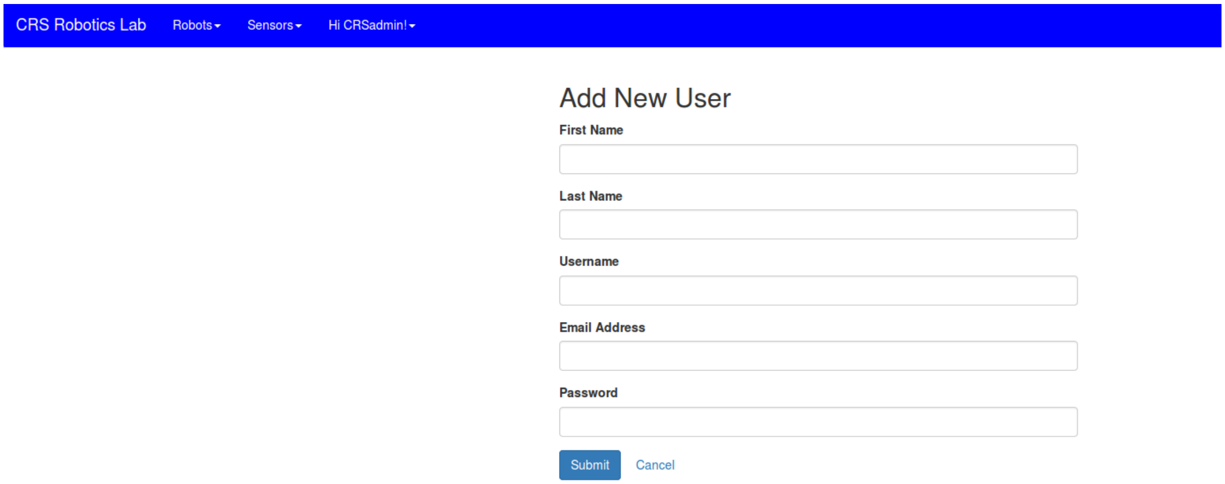
System Description:

1. Angular UI

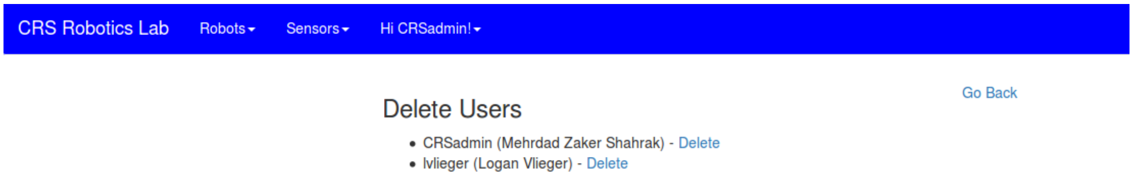
a. Login Page



b. Add User Page



c. Delete User Page



Delete Users

- CRSadmin (Mehrdad Zaker Shahrak) - Delete
- lvlieger (Logan Vlieger) - Delete

2. Angular Services

a. User service

The user service contains a standard set of CRUD methods for managing users via HTTP methods leveraging the REST API in Express.

b. Authentication service

The authentication service is used to login and logout of the application, to login it posts the users credentials to the API and checks the response for a JWT token, if there is one it means authentication was successful so the user details including the token are added to local storage.

3. Express Server Services

a. User controller

The Express.js users controller defines and handles all routes / endpoints for the Rest API that relate to users, this includes authentication, registration and standard CRUD operations.

b. User service

The MEAN Stack 2 user service is responsible for all database (MongoDB) interaction and core business logic related to user authentication, registration and management. **Calls bcrypt.js to hash user password before sending to MongoDB**.

Configuration:

User Model:

In the /_models directory contains a file called user.ts. It describes an object with the required fields for the user model, username, email, password, name and privilege, which is null, besides for the admin. Upon authentication, a JWT added to the model to be stored in the DB.

Admin User:

In the /app directory contains a file called admin.config.ts. It describes a user model with an additional field, *privilege*, set to true. This field allows the Angular UI controller to know what view to display. This admin user is automatically registered upon first initialization of the web app.

***** Admin user must be removed from Database and application closed before updating admin user, then application must be restarted for changes to take effect*****