Assn4-erbdesc

Developers
Kaitlin Erb, Marc-André Descoteaux

Design Decisions
This project is built off **SpringMVC**. It uses a **hibernate** and **h2** to create a database of users. We relied on **SpringMVC** for its dependency injection feature, for Hub and Mediator. The structure of our webapp was built off of similar **SpringMVC** webapps; one of those structures was designed using the **Repository Pattern**. This pattern was used in tandem with the h2 database to add persistence and allow for centralized data access. Ideally, we'd want **Hub** and the **Devices** to also be their own repository, but were short on time.

Running The Repo
When running the app, SpringMVC finds the *main()* the application file with the decorator *@SpringBootApplication* and executes the code. Clone the repo and run app through the command: gradle bootRun

Once the application spits out log files and hits 75% EXECUTING, in your browser of choice, go to: localhost:8080

Users & H2 Database
To login as a user you can choose to login as an admin or as a guest. For admin login:

username: *admin*
password: *seng330*

For guest login:

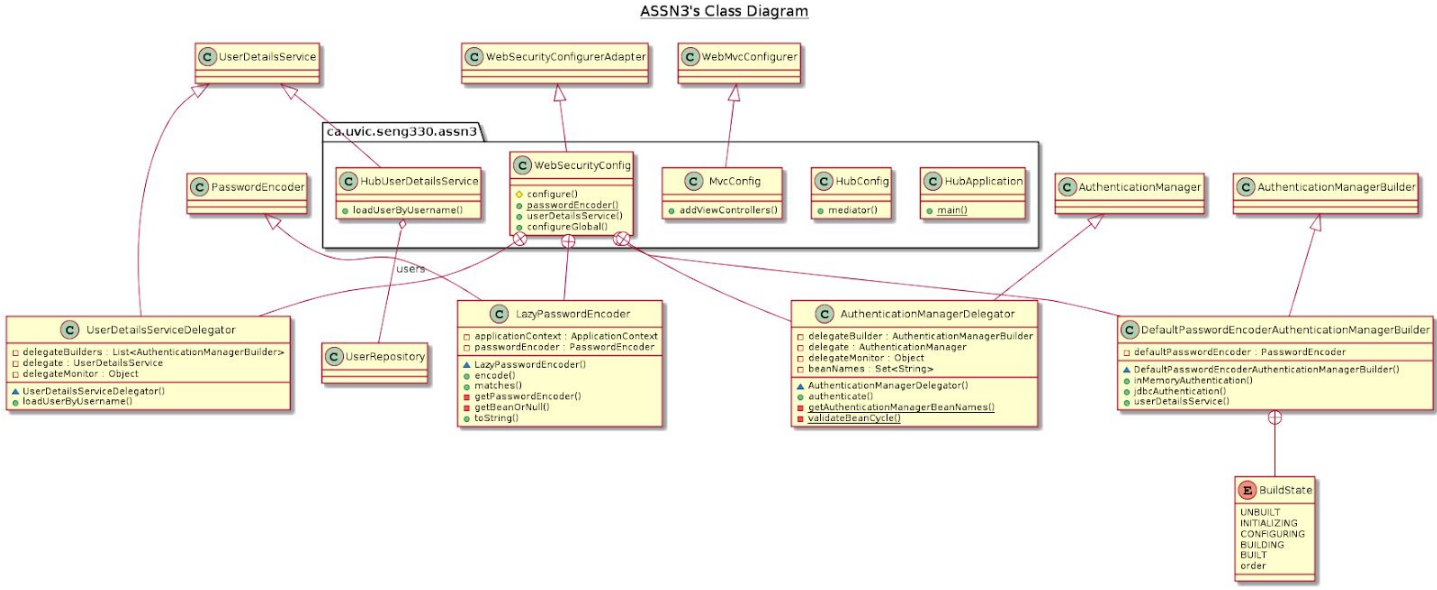
username: *guest*
password: *password*

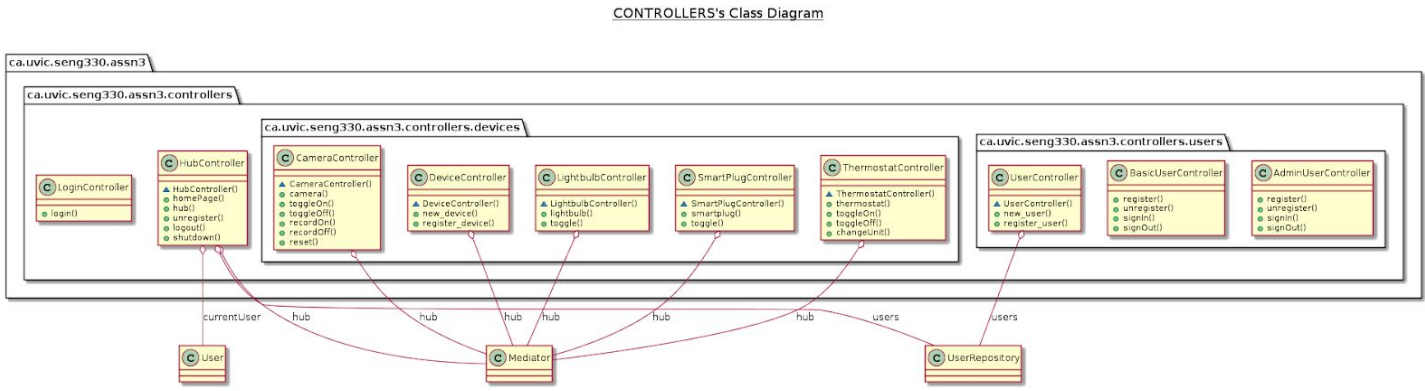
Any newly created users will be stored into the h2 database. Since we used Spring Security, user credentials will automatically be verified. If the username and password don't match up, then the person will be unable to enter the app. Login permissions are used - only logged in users are able to see certain pages.

To access the database, after starting the app, go to: localhost:8080/h2-console

UML Diagrams

Due to the project being so large, UML class diagrams have been segregated. All of the UML class diagrams can be found in a folder at the root of the repo, here: https://github.com/SENG330/assn4-erbdesc/tree/master/uml-class-diagrams





MODELS's Class Diagram