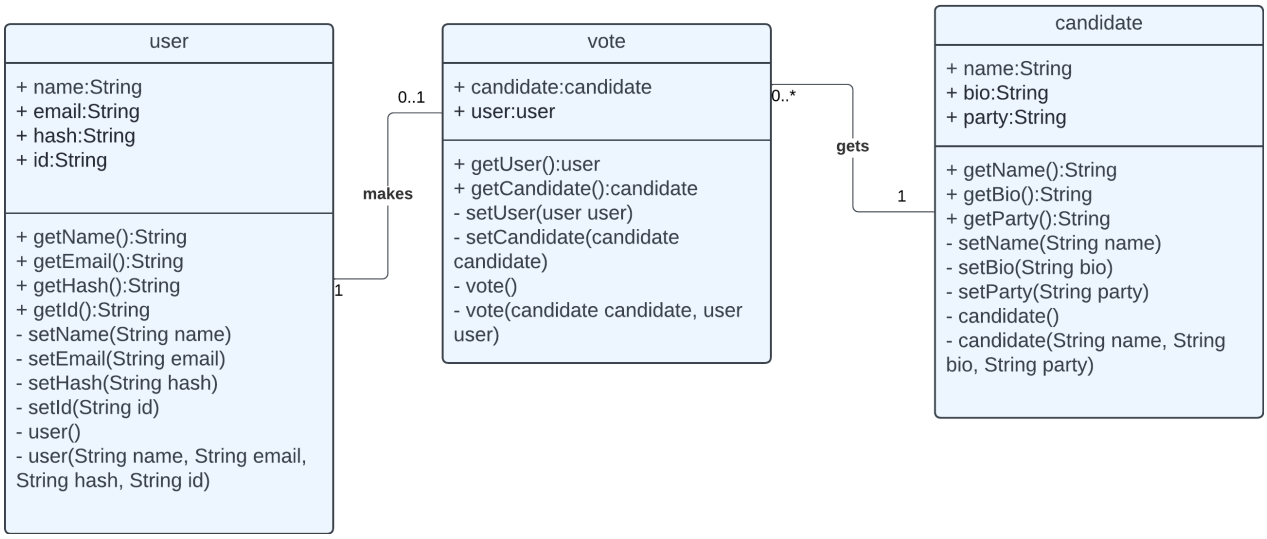
INF4027W Mini Coding Workship: E-Voting App
Author: Adrian Groening
ID: GRNADR009

Database
Name: Cloud Firestore (Firebase)
Description: The database is initialized at the start of the app, allowing for one and only one connection/ initialization to occur. This database is similar to SQL in the way that it makes reads and writes. Previously, the real-time database was used and as a result, a wide range of issues relating to threading and its asynchronous nature arose. The Firestore database fixed those issues because of its synchronous behavior. The database structure is simple and includes only 3 collections, consisting of a user, a vote, and a candidate. Bcrypt is used to encrypt the passwords and they are stored directly on the database as hash.



Structure
Framework: Springboot and Vaadin (Java)
Description: The application imitates a MVC structure as it includes model classes such as user, vote, and candidate, as well as view and controller components such as the home view, vote view, main layout, application class, and frontend files that include the baseline html and javascript.
Services
The application has 3 services consisting of the Firebase service, the password service, and the mail service. The Firebase service is responsible for making read and write operations to the database, fetching it when it is needed in the app, and writing to the collections once authorized changes are made. The password service uses the Bcrypt API to hash passwords once a user signs up successfully as well as to check to see if a user-entered password matches its hash. The mail service uses the mail check API through the Java net library to check whether an email is disposable or not.
Validation, Controls, & Security
The application includes login and sign-up forms that utilize all three of the services to add and authenticate users into the system.
Functionality
When the system runs, the user lands on the home page and can see the election results as they are happening, all without having to log in. When the user goes to the vote view, they will be prompted to sign up or log in. Once they are signed up, they are going to be taken back to the home view and can go over to the vote view to log in. When a user logs in they are greeted by the voting interface in which they can cash their vote. Once a user votes then they are notified that their vote has been cast.
Running the Application
mvn clean install
mvn spring-boot:run or mvnw for windows or ./mvnw for mac