**Progress Report**

**Finished implementation:**

* System Design & UML Class Diagram
* Version Control setup, local DB connection
* Testing CRUD operations on the database
* Creating necessary tables for classes
* React application setup (User login and registration page).

**Unfinished implementation:**

* Relationships between related tables
* Relationships between classes (composition, association, aggregation)
* Service setup for classes.
* Controllers/Routes for backend API.
* React views like deposit, withdraw and transfer page.
* Hooking up backend service to the frontend application.
* Testing backend, frontend, and deployment.

**Changes:**

We might have to drop implementing great ideas for the project that we originally planned to add, primarily due to time constraint on this project. All of us got busy with other coursework due to the end of semester and we now have about a week from 6th April till 13th April to completely focus on this project and create a working model of our project.

We will be making following changes to our application:

* Mapping inheritance to database tables is a challenge. We spent a full day researching and testing how an inheritance hierarchy can be stored in database tables, but we could not get it working. The entire idea of inheritance in our application may be scrapped, but we will continue to try.
* Implementing relationships between classes and tables, even without inheritance, has also been difficult, but we made progress, and it looks like we’ll be able to deliver.
* We might not be implementing the feature where customer has 2 types of bank accounts, Saving and Checking since this has to do with inheritance. This feature’s life depends on our progress with inheritance.
* We will not be implementing Customer Service Representative and Admin modules into our application due to time constraints.

**Research:**

Frontend application has routes which display views depending on where the user clicks on a webpage. Guarding routes on frontend application is important since we want to filter out those users who are not authorized to access certain routes that contain sensitive information. Upon researching for the same an implementation for router guards is in place on the frontend application to protect sensitive data by using browser’s local storage.

Comparing security aspects and looking for methods to guard the server against any Security breaches using Spring Boot Security. Methods for authentication/authorization, route guarding, database access guard, and user session sanitation to prevent reverse shells and cross-side scripting. Getting authentication and authorization proved simple, but sanitation, route guarding, and database guard are still in development.

Currently researching how to map an inheritance hierarchy into a series of related database tables. We’ve successfully managed to push an inheritance hierarchy into database tables but retrieving data from a database and storing it in an inheritance hierarchy is proving to be quite challenging.

**Implementation:**

We have 3 key classes implemented for the backend service: Bank Account, Transaction and User. These modules are the foundations of our backend application, and they are not fully implemented yet. We have a skeleton of these modules where attributes and necessary methods have been implemented, along with database tables unit test. For the front end, we have login, registration, deposit, and withdrawal views implemented, but they need to be updated to show appropriate data when the backend service gets hooked up. We also need to add some client-side validation onto our front-end application to avoid overloading backend server with trivial requests.

**Work Division:**

Work division for the backend service is equally divided among our group members. Omar is given the responsibility to implement Login/Registration & Security for a user, which acts as a middleware checkpoint before accessing any other routes or operations on the server. The user cannot access the server or any of its information without having an authenticated account, which must then be authorized through an email link. Philippe is working on Transaction module, whose prime responsibility is to record the movement of money between different accounts, as well as how the application will interact with the database. Soham is working on developing a skeleton for frontend application with React.js and Bank Account module for backend where the module will hold all data pertaining to a user’s bank account like account number & balance and implementations of withdrawal, and deposit functions.

As for the frontend application, we have not decided who will work on which part but as soon as our backend service is up and running, we will be moving onto developing frontend application where we will divide the work equally.

**Teamwork Experience:**

The three of us have a long-standing relationship that started in our first semester at Seneca. We feel comfortable expressing our ideas and doing live coding over a call. We chose each other as teammates because the three of us have shown interest in material beyond the scope of the courses, and we each pursue that material in our own way. We couldn’t ask for a better team.