My use case diagram features two users, a customer and a manager, I will reference them both as a user. A user can log in and log out. Logging in requires authentication from the program, which will compare the username and password with the corresponding username.txt file if the login is a customer, or with username: admin, and password: admin if it is a manager. A customer can also make a purchase, deposit, or withdraw money. The current balance in the customer’s account is displayed at the top of the screen. The manager can create a customer or delete a customer. Adding a customer will create a text file with the necessary information stored in it.

Use Case name: Login

Participating Actors: Manager and Customer

Flow of Events:

1. User sees the related GUI which prompts for username and password
2. User provides the necessary information
3. The information is authenticated
4. The user is lead to the next page

Entry Condition: Program is started

Exit Condition: User (Manager or Customer) provides correct login credentials and clicks ‘submit’

Quality Requirements: None

My class diagram shows all the controllers for all the different .fxml files, as well as the small framework which controls the switching of scenes in the window, to different .fxml files. The customer class is the container that holds the customer object and makes changes to the customer file when necessary. One of the fields of the customer class is a Tier object which represents the customer tier and calls the correct Tier class getFee() method when required. The class diagram also shows the relationships between the classes, and their dependencies. It shows the behaviours and properties of all classes. It describes the general structure and architecture of the program. It shows the starting point of the program which is the main class. The relationship between the Customer, Tier, Silver, Gold, and Platinum classes show a State Design Pattern. The Customer in this case is the context. The class I chose to implement point 2 is the Customer class.