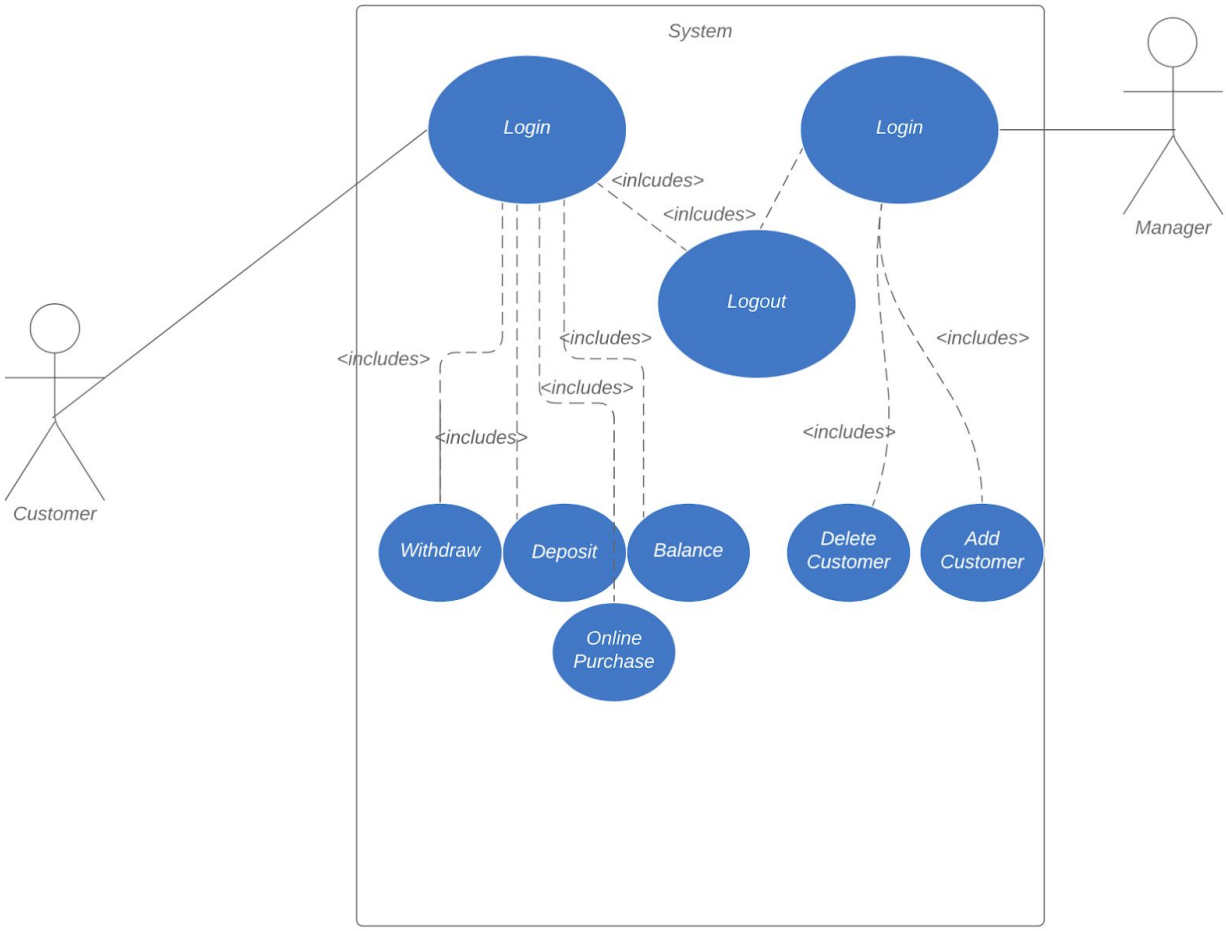
Bank App Project Report

Use Case Diagram

The following use case diagram outlines the source code for the application. It showcases 2 actors a customer and a manager. Both share 2 of the same use cases log out and log in. However the manager actor proceeds through the login case and goes to different use cases than the customer actor. The manager actor is able to add and delete customers while traversing through the login case. With both actors having the ability to log out. The customer is able to login and be authenticated through the manager actor. They can deposit, withdraw, purchase, logout, check balance and check their level.

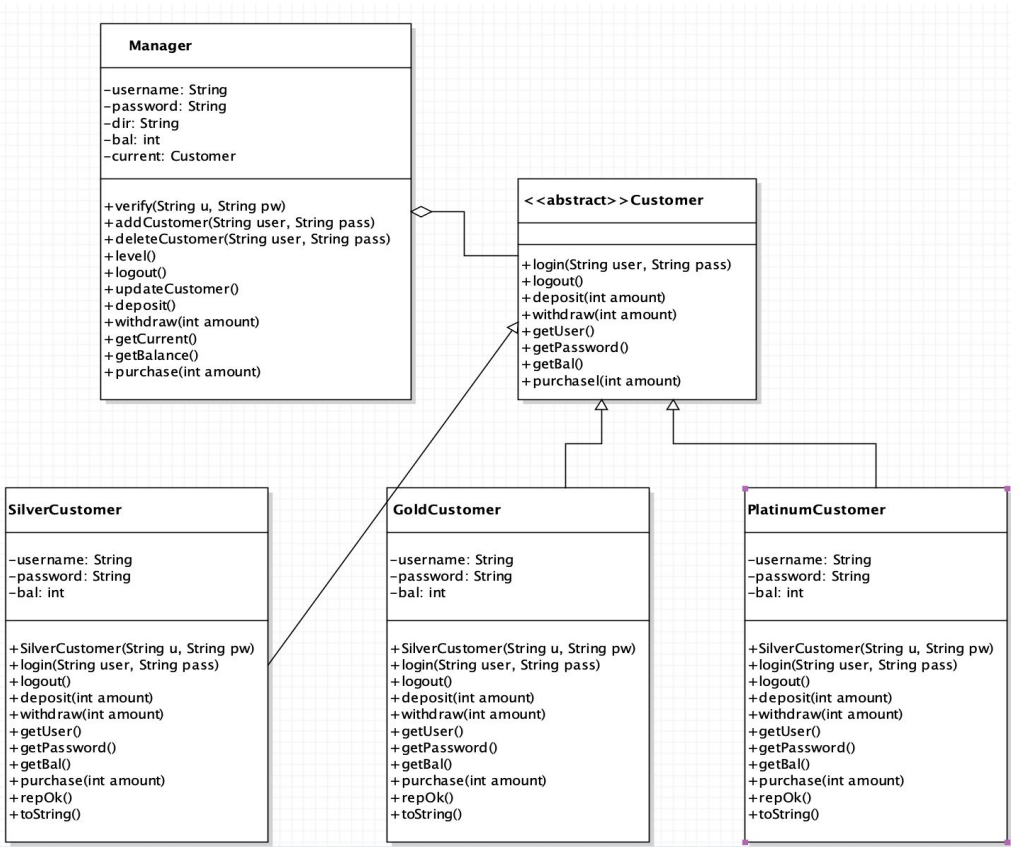


The add customer use case follows the following path:

1. Add Customer
2. Involves the Manager
3. The customer must have a first and last name and not already exist
4. The manager hits the back button or the customer is successfully added
5. Flow of events:
    a. Input first name
    b. Input last name
    c. Customer is added with a balance of 100

Class Diagram

The following class diagram outlines the class setup of the app. It displays all the classes except for the GUI class. The main class which the others extend from is the manager class. Firstly the customer class is an aggregation to the manager class. This is because in the manager class the customer class is referenced as a variable to be used in all the methods. From the customer class proceeds the 3 subtypes of customers, silver, gold, and platinum. The customer class is abstract therefore the silver, gold, platinum class implements all the abstract methods.



State Design

To apply the state pattern, I implemented it through an abstract customer class that was referenced in the manager class. The concrete classes of Silver, Gold, and Platinum implement the abstract methods found in the customer class. Inside the manager class it updates the account based on the balance so ultimately it controls the switching of states. Since there is 3 separate types of customers having an abstract overall customer class helps to create new customer objects.