Lab Project Report

Rajveer Dhillon 500777569

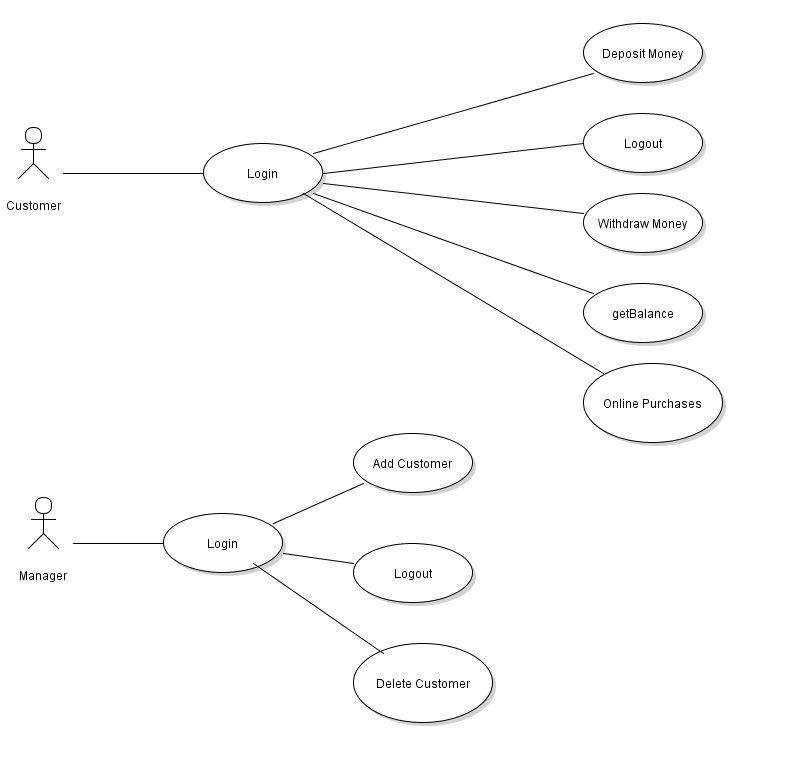
section 8.

Using the UML Diagram, the basic approach to the program can be illustrated. There will be two types of users: Customers and Managers. The customer can withdraw money, deposit money, purchase items online and logout. There are some restrictions to these methods, the user cannot withdraw beyond its balance and the purchase as well. The Manager can add customers and delete customers and can also logout.

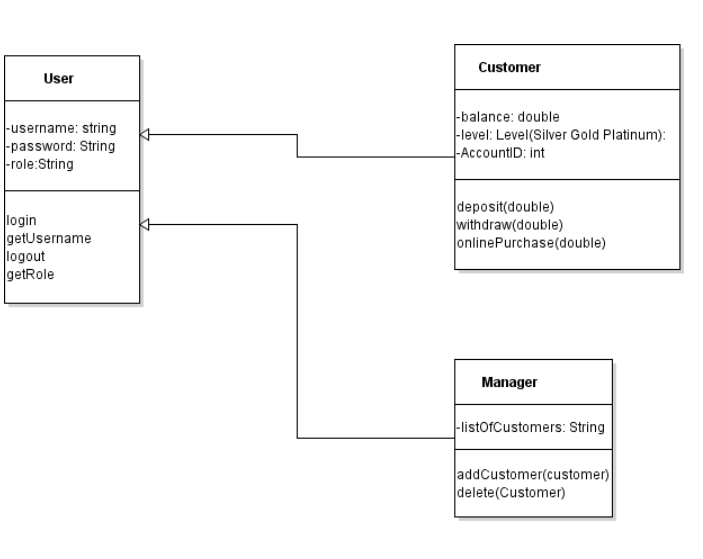
The Case Diagram goes more in-depth with the different aspects of the customer and manager. They both come from the same abstract class, User, as they are both users of the program. The user has the basics of the classes: username, password and role. While they both have different methods, in principle the program should address them as either the user or the manager as they are used in the login screen. Subsequently, the since the manager and customer classes use the same abstraction, their logins and logouts are similar meaning the difference in code is only the role they have (either a customer or manager) which is implemented in the program.

The State Diagram implements the levels the customer is at: Silver for when the balance is less than $10 000, Gold for when the balance is greater than $10 000 and less than $20 000 and Platinum when the balance is greater thn $20 000. The state(level) of the customer should change with every change to the balance, in a update method which is called for every transaction. The fee for purchasing online should also be changed for everytime the state(level) is switched. The state diagram was implemented in the code.

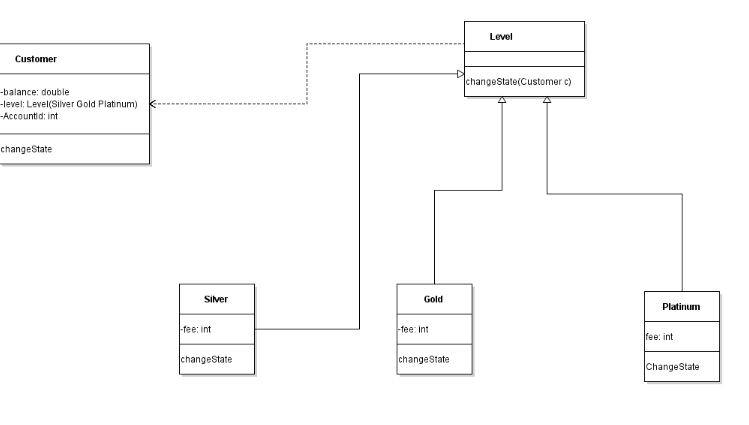
The class in which the second point was implemented in was the Customer Class(ContextCustomer). In the customer class, the customer had: username, password, balance, accountID. The username and password were immutable while the balance and future variables were. The Overview, Abstract Function, Rep Invariant and toString were implemented as well. The Abstract Function was implemented, outllining the change in state(level) from Silver, Gold or to Platinum. The Rep Invariant was implemented as three phases: if the deposit was less than 0 then false otherwise true, if the balance – withdraw is greater or equal than 0 then true otherwise false and if balance –(purchase + fee) is greater or equal than 0 then true otherwise false.



**Figure 1: UML Case Diagram.**

****

**Figure 2: Case Diagram.**

****

**Figure 3: State Diagram.**