Name: Syed Yousuf

Student ID: 500833920

Section #:

1. Describing the Use case diagram:

Use case diagrams illustrate how the actors interact with the system. In this project, the two actors are Customer and Manager. Both Customer and Manager can login, but need to have their credentials validated before proceeding. After a successful login, the Customer can then proceed to perform any of the following actions: “withdraw money”, “deposit money”, “make purchase”, and “logout”. If the customer wants to make a purchase or withdraw money, the system will check if they have sufficient funds before it makes the transaction. When the manager successfully logs in, he/she has the option of: add customers, delete customers, and logout. When the manager deletes a customer, the system checks if the customer exists before deleting.

1. Describing the Class diagram:

A Class diagram shows all the basic classes in an application and how they interact with each other. The Manager class knows the username, password and role of the manager. It also knows if the manager has logged on successfully. The Manager class has all getter methods in order to retrieve the instance variables. The Manager class also has login and logout methods, so that the manager can login if the credentials are validated. Additionally, the Manager class has two other methods, addCustomer and removeCustomer method so that the manager can perform its main operations. Next is the Customer class, which handles the account balance and successful login of the customer. The Customer class also provides the essential methods so that the customer can do transactions, such as making purchases, depositing money, and withdrawing money. Each Customer object has one CustomerFile object, which has the username, password, and other information of the customer and handles input/output to the respective file (each customer has one file). Each Customer object also has one Level object from the Level class, which determines the level of the Customer.

1. The class that I have selected to address point number 2 is the Customer class. Thus all proper comments, abstract function, rep invariant, toString() method, repOk() method, ect is found in that class – Customer class.
2. On the UML class diagram, the parts that form the State design pattern is the Level abstract class and the three level types subclasses (silver, gold, and platinum). Each of these level types has a status and fee. The state design pattern is used here as the Customer can be any of these three level types at any given time.