April 3, 2020

Ransika Perera

500903101

Section 8

COE528 Project

Report

# UML Case Diagram

In my use case diagram, there are two primary actors and two secondary actors. The two primary actors are Manager and Customer which are both generalizations of a User of the program. The Customer also has three further generalizations depending on the level of customer, Silver, Gold or Platinum. The two secondary actors are the Bank and Database. The database helps verify logins for customers and the manager and also verifies whether customers do or don’t exist when the manager deletes or adds them. The bank verifies deposits, withdrawals, purchases and balance checking by the customer. When conducting an online purchase or withdrawing from the account the bank also verifies that there is enough funds in the account for the transaction. Finally, both the customer and manager have the option to logout once they have finished using the application.

# UML Class Diagram

In my class diagram, the highest-level class in the abstract User class which is the parent class to the Manager and Customer classes. The User class has the instance variables username, password and role, two methods login and logout, all of which are used by the children classes. The manager class has no instance variables and 2 methods addCustomer and deleteCustomer to add and remove customers from the database. The Manager class is also designed according to the singleton design pattern since there is only one manager, and this is done using the variable instance and getInstance method. The Customer class, on the other hand, has many more components. It has two instance variables bankAccount and level and four methods, deposit, withdraw, getBalance and onlinePurchase. The Customer class also has other variables and methods used in its design according to the state design pattern. Finally, the CustomerStates interface has an aggregation relationship with the Customer class and is implemented by three classes, Silver, Gold and Platinum. The multiplicity of the class diagram shows that in the application for User abstract class there is only ever one Manager instance but many Customer instances and for every customer, there are three different implementations of the CustomerState interface, one Silver, one Gold and one Platinum.

# Commented Class

The class that I have selected to address point number two of the assignment with regards to Javadoc commenting was the Manager class.

# State Design Pattern

As touched upon previously the parts of the UML class diagram that form the state design pattern is the CustomerState which is implemented by three classes Silver, Gold and Platinum. The CustomerState then forms an aggregation relationship with the Customer class to complete the state design pattern. In a more general sense, the Customer class is the context, the CustomerState interface is the state interface and the Silver, Gold and Platinum classes are the concrete states which implement the methods of the interface.

# References

No external references were used.