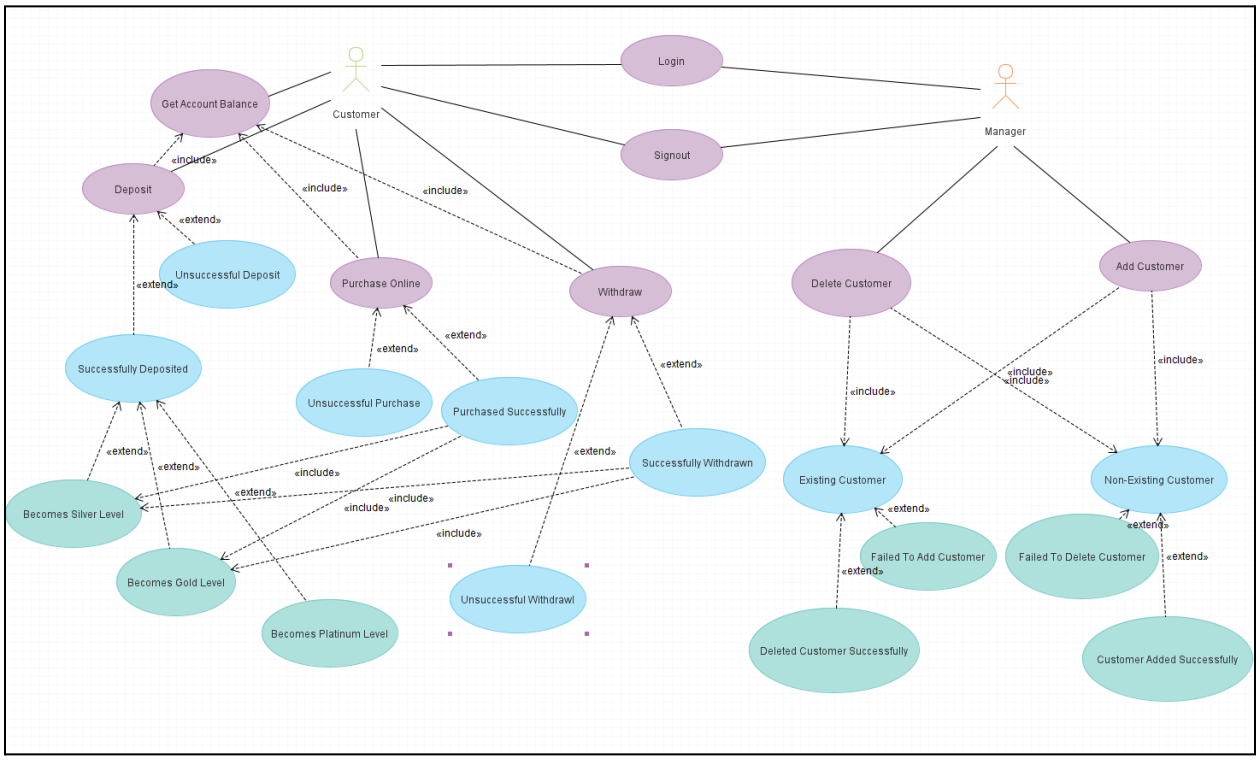
Sheldon Cerejo | 501173492 | Section 01

Final Report

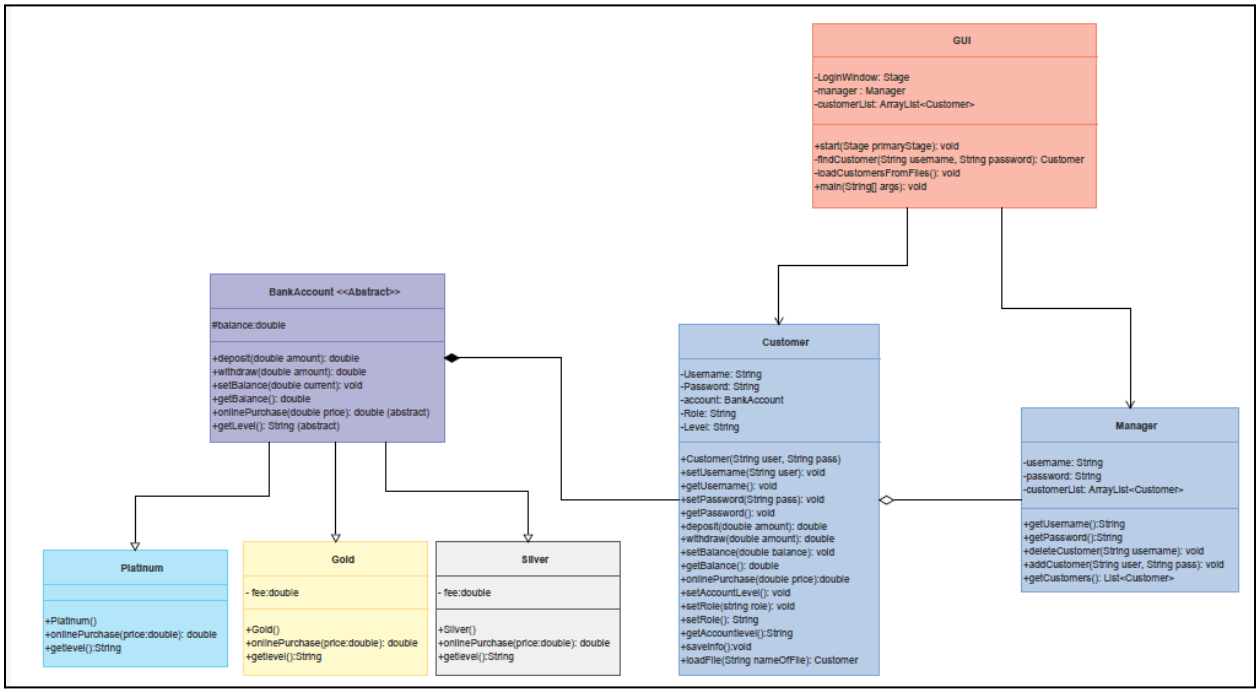


Use-Case Diagram

The use case diagram I have created contains two actors representing the two possible roles of the user. The user has the capability of logging in and signing out which is why they are shared cases between the Manager and Customer. The Manager has the ability to add or delete customers based on whether they exist or not. The customer has a lot more operations they can perform. The Customer can view their account balance, deposit, withdraw or purchase an item online. The account balance is displayed to the Customer directly and will constantly update when any other transactions are performed. Money can be deposited into the account directly if the amount is greater than $0. The Customer's level may upgrade to Gold or Platinum if the amount is sufficient enough to go over $10000 or $20000. Likewise, money can be withdrawn from the account and if the amount is sufficient, can degrade the account to Gold or Silver. A user can make an online purchase provided they have enough money within their account to do so, based on the account level the purchase will have an associated fee, $20 for silver, $10 for gold and $0 for platinum. Much like withdrawing money this can degrade accounts to gold or silver and a call to that is included when a purchase or withdrawal is successful.

Use Case Description

Use Case Name	Purchase Online
Participating Actors	Customer
Flow of Events	1. Customer makes online purchase. 2. System asks for confirmation. 3. Customer confirms payment. 4. Systems acknowledges payment.
Entry Conditions	Customer is logged into bank system.
Exit Conditions	Money is deducted from the account and payment has been acknowledged.
Quality Requirements	Customer deposits $9900+ for discount on purchase.



Class Diagram

This class diagram consists of seven classes, the six main source classes and the GUI class. Which is responsible for how the user interacts with the class information and how the classes interact with each other. The GUI is associated with Manager and Customer which are the two main classes. These two classes have an aggregation relationship since the Manager class can exist without the customer, however the opposite can not happen. The customer class has the methods that a customer can use such as deposit and withdraw but is ultimately composed of the

BankAccount abstract class which holds the customer information that is displayed to the user such as balance and account level. This abstract class is also broken down into several subclasses pertaining to each of the account levels since each of these subclasses contain the same types of data inherited from BankAccount but all behave differently depending on which one it is.

State Design Pattern

BankAccount is the class that implements the state-design pattern. Customer becomes the context with the abstract class BankAccount being the state with each level class forming concrete states Silver, Gold and Platinum.