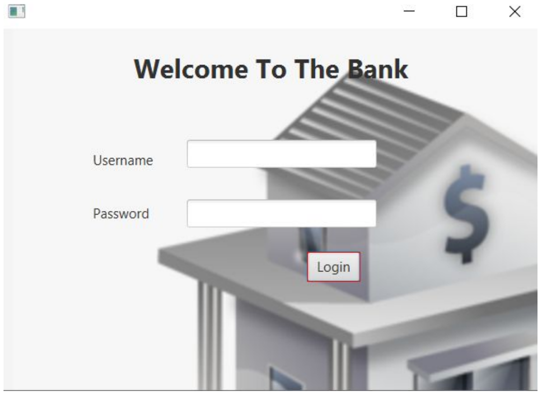
Daniel Sadig
500894225

COE 528 report



UML use case diagram

Use Case: Log in
Participating actors: Manager and Client
Flow Events

A Manager can log in to the manager class by entering their username and password, it is then validated. Then the manager can create a client by entering their username and password, the new user has an initial balance of 100 dollars. The manager can also delete a client, this removes the users file from the folder .Once a client is created they can login, their password and username must also be verified. When verified it goes to the client class which displays the client's Username, balance and status (silver, gold or platinum). The client can deposit money, they can also withdraw money and perform online purchases such that they have enough money in the bank.Also, for the online purchase there is a 50 dollar requirement, meaning all purchases must be greater than fifty dollars .Once, the customer logs out the changes made to their account are written and saved to the file. To exit the manager class you simply press the logout button and to exit the customer class you press the "Go Back" button.

UML Class Diagram

The **FXMLDocumentController** is the login page, if there is no client created then the manager can enter the **ManagerWindowController**. They do this by entering their username and password, if verified it will enter the **ManagerWindowController** which has a **ManagerAcount** which creates an array of **User**. In the **ManagerAcount** the manager can create a new **Silver** user or delete a user. The manager can then exit the **ManagerWindowController** through the logout function. The client has a login,

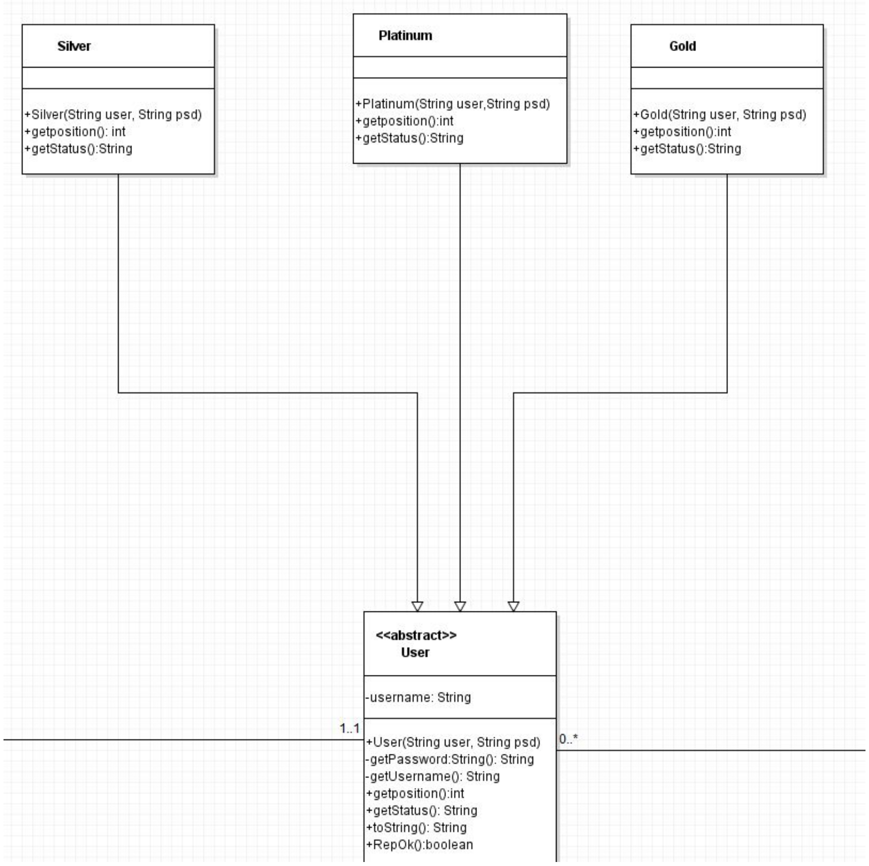
FXMLDocumentController , if they enter a valid username and password they go to the **ClientGUIController** which creates a **Silver, Gold,** or **Platinum** client, these classes depend on the **User** class. The **ClientGUIController** also has a **ClientAccountMain** this class allows a user to deposit, withdraw and perform online purchases. It also ensures that your bank is not negative, updates your status based on the funds in your account and ensures that an online purchase is not less than fifty dollars. The client can then exit the **ClientGUIController** by the “Go Back” button, which returns you to the login page.

Q: Mention the class that you have chosen to address point number 2

- I used the user class to write my abstraction function, rep invariant and the necessary clause

Q: Refer to the UML class diagram and indicate the part(s) from the state diagram

- My abstract class that gets extended is the **user** class. The concrete classes are Silver, Gold and Platinum. The ClientGUIController has a User that will cast a client to silver, gold or platinum based on the amount of money you have in your balance.This is done in the purchase function of the ClientGUIController .Also, based on the state you are in your online purchase fee will be different.


Silver
+Silver(String user, String psd)
+getposition(): int
+getStatus():String
Platinum
+Platinum(String user,String psd)
+getposition():int
+getStatus():String
Gold
+Gold(String user, String psd)
+getposition():int
+getStatus():String
<<abstract>>
User
-username: String
+User(String user, String psd)
-getPassword:String(): String
-getUsername(): String
+getposition():int
+getStatus(): String
+toString(): String
+RepOk():boolean
1..1
0..*