Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Object Oriented Eng Analysis Design
Course Number:	COE 528
Semester/Year	W2024

Instructor:	Boujemaa Guermazi

Assignment/Lab Title:	Term Project: Winter 2024

Submission Date:	March 25, 2024
Due Date:	March 25, 2024

Student's Last Name	**Student's First Name**	**Student #**	**Section**	**Signature**
Shanmuganathan	Theeshiikan	501157664	12	TA

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Overview:

The user class was selected to show the requirements mentioned in point 2.

Use Case Diagram:

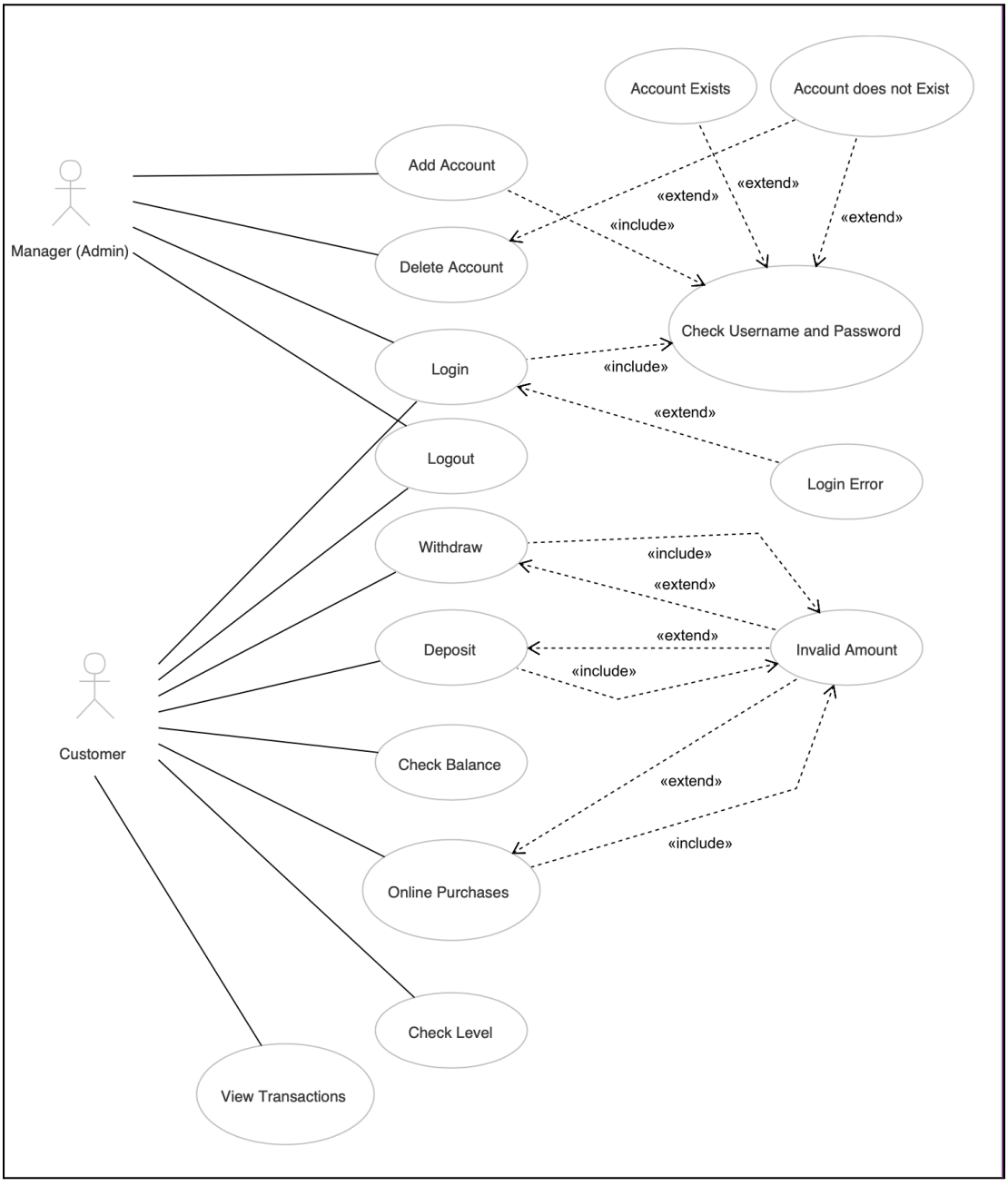

Figure 1: Use Case Diagram

Figure 1 represents the use case diagram from the bank account application should work. There are 2 actors (Customer and Manager). If a customer is logged in, they have a set of possible actions which include withdraw, deposit, online purchase, check balance, check level,

view transactions and finally logout. The first 3 actions have an exit condition of when an inclined amount is entered. Balance and level are always valid. If a manager is logged in, the possible action is add account, delete account and log out. The exit condition for adding an account is that the account username and password should only have letters or numbers and the account must not already exist. To delete an account the account wished to be deleted must exist. For both, logging out the exit condition is successfully clicking the button and log in the condition is to enter the correct username and password.

Use case diagram: Delete Account
Participating actor: Manager
Flow of Events: 1. The user must successfully be logged in as a manager 2. The using the drop down menu in the specified “Delete Customer” area, the correct username must be chosen to be deleted 3. A label will appear that says the Customer was successfully Deleted
Entry Condition: The user must be logged into as a Manager
Exit Condition: The customer that the manager wanted to delete was done so successfully.
Quality Requirements: The customer’s account must exist.

Class Diagram:

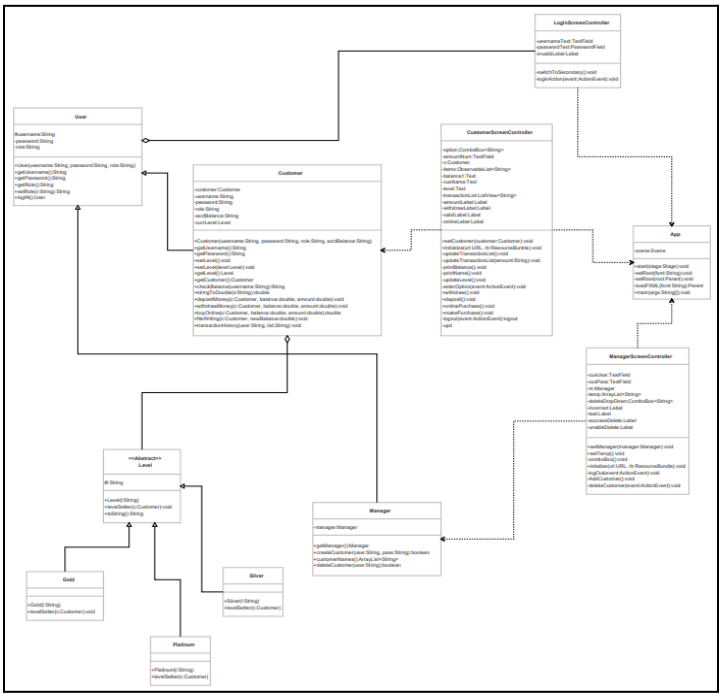

Figure 2: Class Diagram

Figure 2 is a class diagram that shows how the bank system created works and the relationship between the 11 different classes. The main method is held in the App class. There are 3 controllers that depend on this class which are the ManagerScreenController, CustomerScreenController and the LogInScreenController. These applications connect the rest of the code to the GUI. The Level class is abstract which has 3 child classes. These classes are Silver, Gold, and platinum. This is the state level class that shows the state design pattern. The user class makes a new object that can be either a Customer or Manager. The customer class extends the user class and carries out the functions that a customer can do. The methods in this class get called in the CustomerScreenController class. Likewise, the Manager class extends the user as well. The method in Manager carries out the functions that the manager can do and they are called in the ManagerScreenController Class. The methods in the User class get carried out in the LogInScreenController.

State Design Pattern:

The level class is the abstract method that implements the state diagram pattern. It has 3 child classes which are Gold, Silver and Platinum. These classes represent the different levels that a customer can have depending on their account balance. Once the balance changes, then the account level or state will change accordingly.

Resources:

https://stackoverflow.com/