Banking App

Jason Nguyen

Toronto Metropolitan University

COE528 - 011

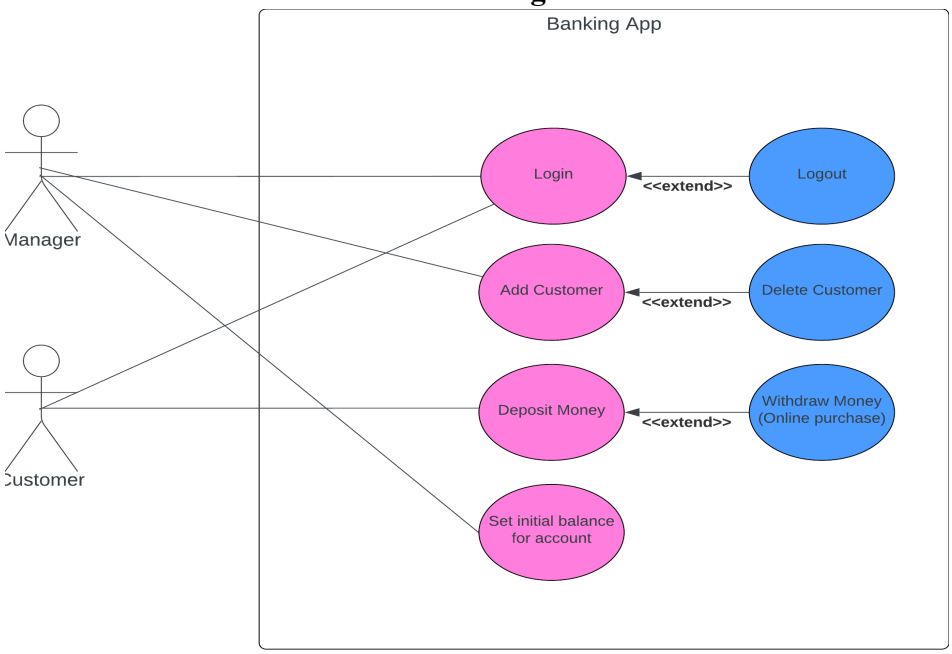
Prof. Boujemaa Guermazi

24th March 2024

Overview

The report for The Bank App Project, is presented herein. It consists of sections explaining different parts of the final project for the COE 528 course. In an overview, topics such as Use Case Diagram, Class Diagram, UML, and explanation of multiple implementations is discussed in detail. These diagrams is included as a picture, however, .PNG files of the diagrams will be submitted along as well. References in the report encompass sources whose ideas were utilized to develop the code and are listed at the end of the document.

Use Case Diagram



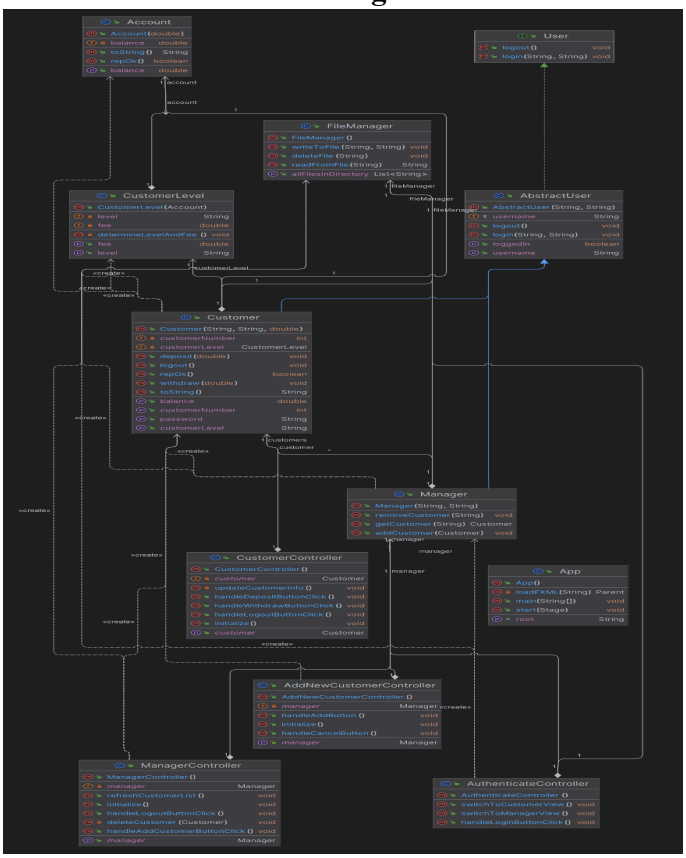
Desciption

The Use Case diagram shows the external actions during requirement gathering and analysis, capturing the key features from the user's perspective.The Bank Account Application uses a Use Case Diagram to illustrate the interactions between the Manager and the Customer. The Manager can log in, log out, add, and delete. The Customer can access banking functions. The system classifies customers into silver, gold, and platinum levels, with fees for online purchases. Authentication is essential for both the Manager and the Customer, and the diagram demonstrates the conditional logic for determining customer level and transaction fees.

Use Case Name	Banking App
Participating actors	Initiated by either the Customer or Manager
Flow of events	Customer/manager enters username and password.

	System validates the credentials against the stored information. If authentication is successful, the Customer/manager is logged into the system. If authentication fails, an error message is displayed. Customer/manager can then proceed to perform banking operations.
Entry condition	Either the Customer or the Manager attempts to log in
Exit condition	Customer/manager successfully logs in or receives an error message.
Quality requirements	Secure authentication process ensuring only valid customers can access their accounts.

Class Diagram



Description:

The class diagram shows the structure of the Bank Account Application, including the classes, their attributes, and how they relate to each other. One important class in this system is the "Customer" class. This class contains customer-specific details like username, password, role, bank account information, and account level. It also has methods for logging in, logging out, depositing, withdrawing (online purchase), and checking the account balance.

Selected Class for Point Number 2:

For the purpose of this report, the "Customer" class has been specifically selected to address point number 2. This class serves as a representation of individual bank customers and incorporates all functionalities pertinent to customer operations within the application. It's noteworthy that all classes within this project have been developed with the objective aligned with point number 2.

State Design Pattern in UML Class Diagram:

In the UML class diagram, the part that forms the State design pattern is the representation of the different customer levels (silver, gold, and platinum) as states. These states are linked to the "Customer" class, showing the behavior and features associated with each level. The customer's account balance determines their state, which dynamically changes and affects the fee calculation during online purchases.

References

Oracle. (n.d.). Reading, Writing, and Creating Files (The JavaTM Tutorials > Essential Java Classes > Basic I/O). Retrieved from https://docs.oracle.com/javase/tutorial/essential/io/file.html

CodeAhoy. (2022, May 22). File I/O in Java. Retrieved from https://codeahoy.com/learn/csfundamentals/ch5/#:~:text=File%20I%2FO%20in%20Java%201%20Reading%20characters%20from,to%20a%20text%20file%20. . .%204%20Summary%20

Stack Overflow. (n.d.). How to add dynamic columns and rows to TableView in java fxml. Retrieved from https://stackoverflow.com/questions/37559584/how-to-add-dynamic-columns-and-rows-to-table view-in-java-fxml