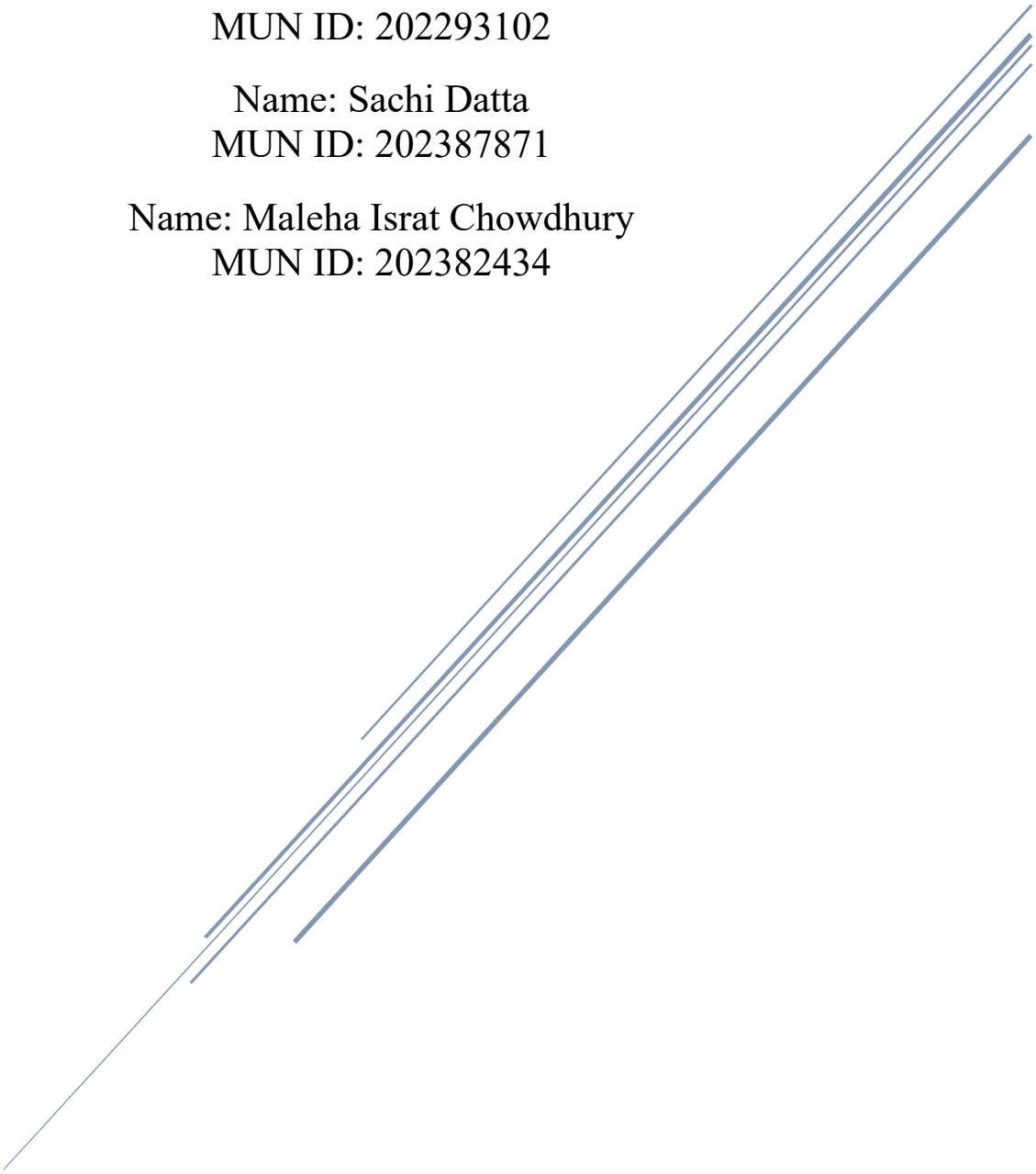
BASIC BANKING SYSTEM

Group 25

Name: Jakaria Hossain
MUN ID: 202293102

Name: Sachi Datta
MUN ID: 202387871

Name: Maleha Israt Chowdhury
MUN ID: 202382434



Memorial University of Newfoundland
ENGI – 9874 Software Design and Specification

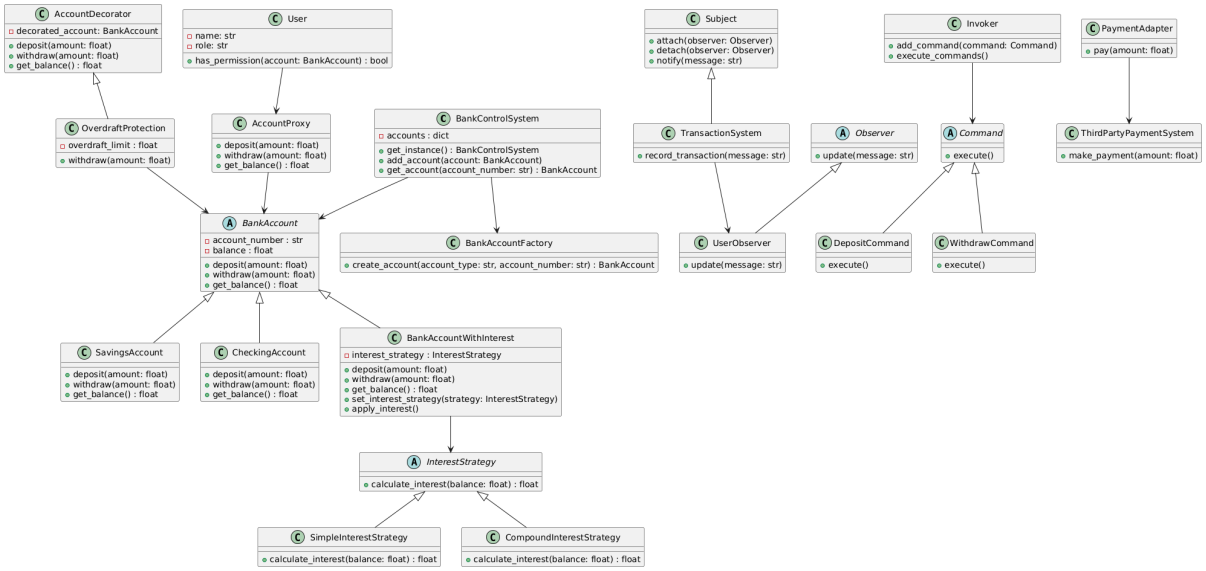
Problem Statement

The project addresses the problem of managing a banking system that supports multiple types of accounts (savings and checking) and various transactions (deposits and withdrawals). Additionally, it provides functionalities like interest application and overdraft protection. The system is designed to be modular and flexible, allowing for easy expansion and maintenance. Key requirements include:

- Ensuring only one instance of the banking system exists (Singleton Pattern).
- Dynamically applying different interest calculation strategies (Strategy Pattern).
- Notifying users of transactions (Observer Pattern).
- Adapting to third-party payment systems (Adapter Pattern).
- Controlling access to accounts based on user roles (Proxy Pattern).

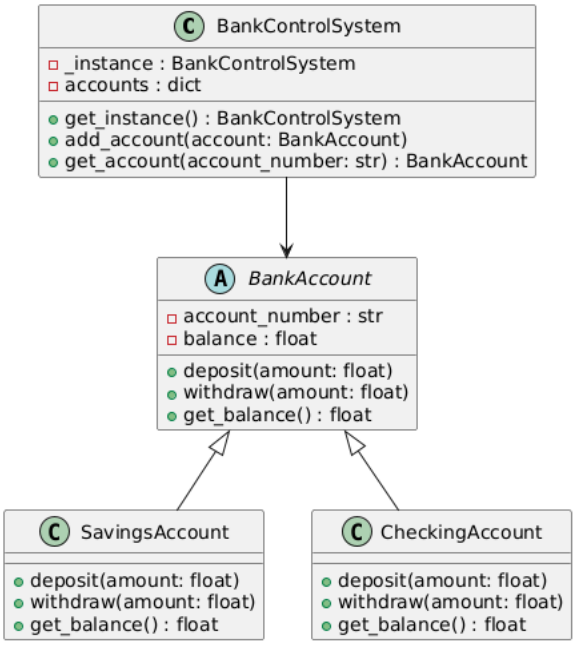
UML Class Diagram for the Entire Code

The following UML class diagram captures the overall structure of the system, showing how different components interact with each other.



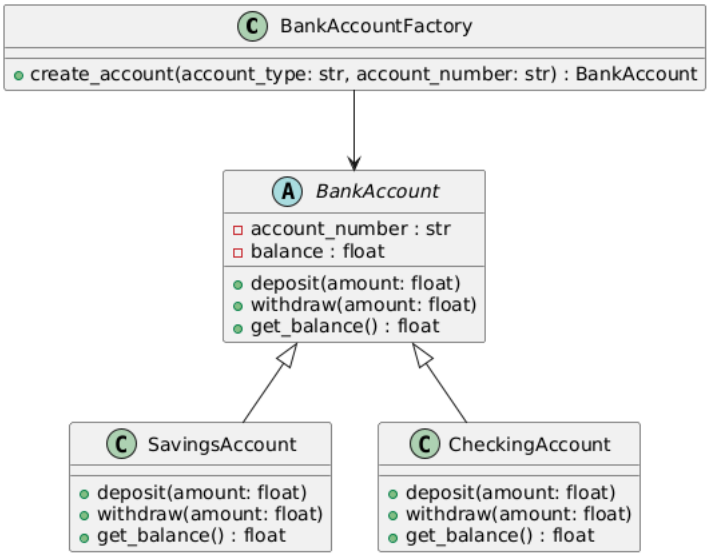
Singleton Pattern

Purpose: Ensures that a class has only one instance and provides a global point of access to it.



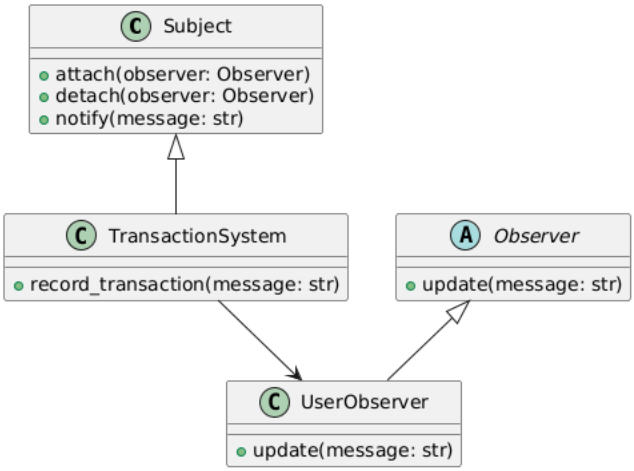
Factory Pattern

Purpose: Creates objects without specifying the exact class of object that will be created.



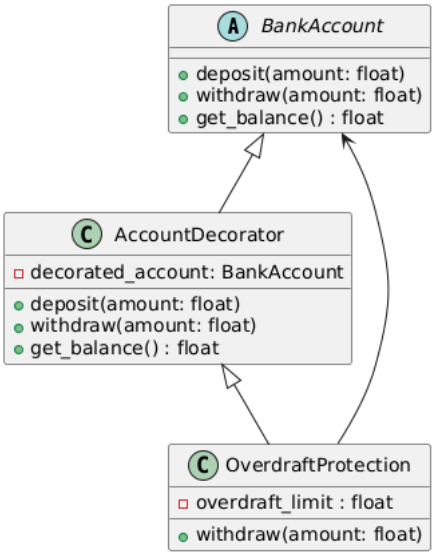
Observer Pattern

Purpose: Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.



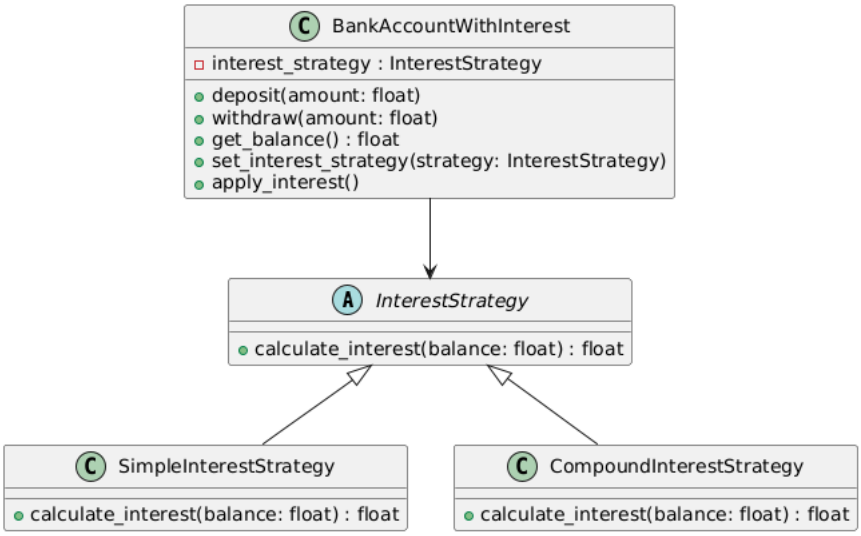
Decorator Pattern

Purpose: Adds behavior to individual objects dynamically without affecting the behavior of other objects from the same class.



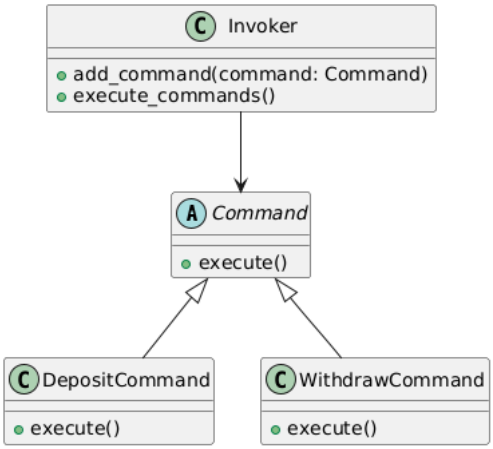
Strategy Pattern

Purpose: Defines a family of algorithms, encapsulates each one, and makes them interchangeable.



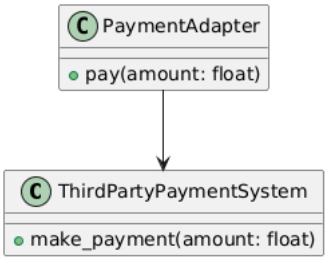
Command Pattern

Purpose: Encapsulates a request as an object, allowing for parameterization of clients with queues, requests, and operations.



Adapter Pattern

Purpose: Allows incompatible interfaces to work together.
the main system expects.



Proxy Pattern

Purpose: Provides a surrogate or placeholder for another object to control access to it.