Project: Design and Implementation of Secure Banking System (SBS)

Project Requirements:

1. Identify the possible threats of use cases for the banking system and analyze them in terms of threat name, security asset, description, vulnerability, and consequences. Define security measures mitigating the threats in terms of threat, security service, and mitigation. (The templates for analysis and security measures will be given in class.)

2. Develop the secure class model:
    a. Draw the classes to realize the use cases using the UML class notation and describe them shortly. Give a stereotype (role name) of each class using the class structuring criteria that classify the classes as an interface, entity, control, or application logic. Define the attributes of each entity class.
    b. Draw the security classes to realize the security measures defined in (1) using the UML class notation and describe them shortly.

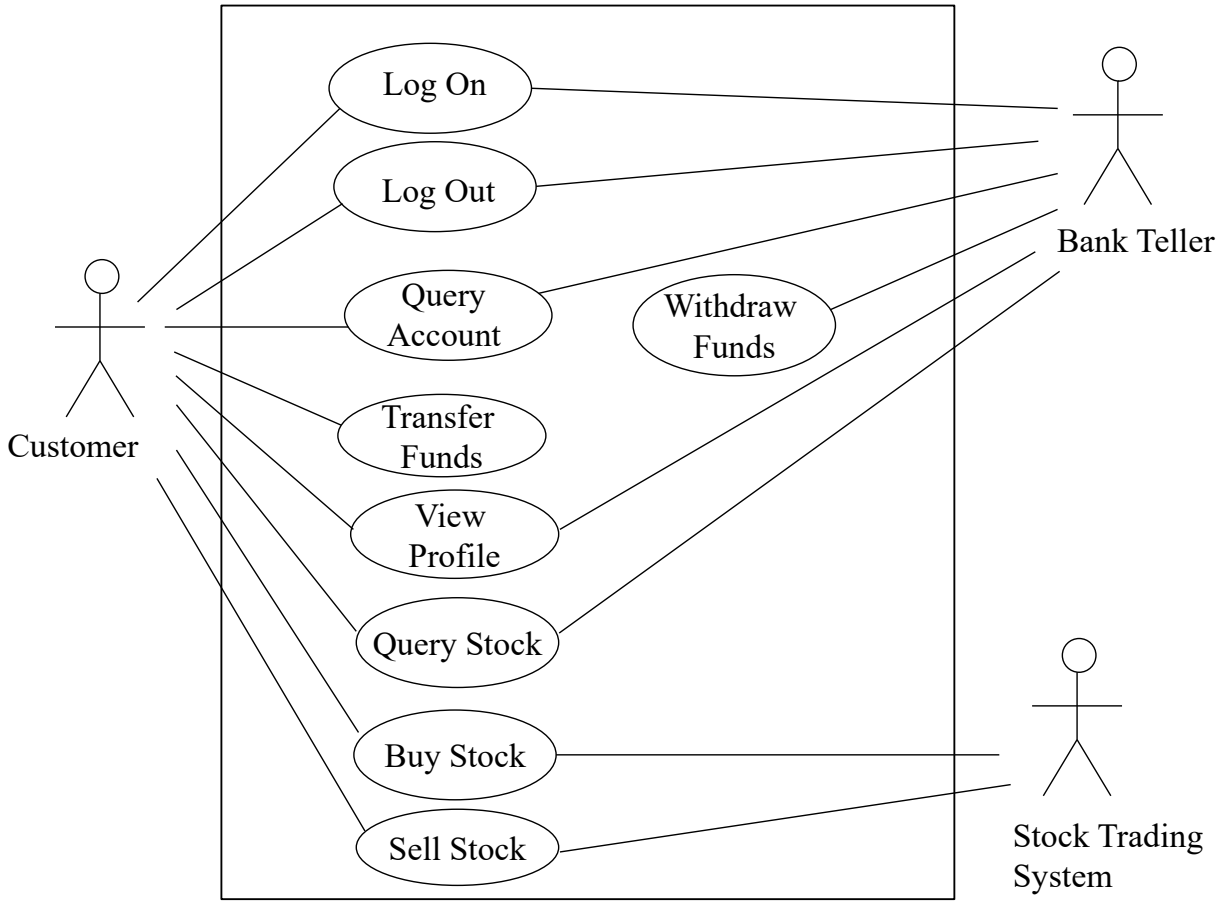
3. Develop the secure interaction model:
    a. Draw the communication or sequence diagrams for each use case. The interaction model must include the main sequence and alternative sequence in the use case description.
    b. Add the security objects to the communication or sequence diagrams (3a) to realize the security measures defined in (1).

4. **Only CS5332 Requirement** - Develop the secure software architecture:
    a. Draw the integrated communication diagram.
    b. Define subsystems on the integrated communication diagram.
    c. Draw a secure software architecture model that shows subsystems and their communication.

5. **Only CS4331 Requirement** - Implement the secure interaction model using Web or App-based or multi-threaded programs. The graphical user interface is not mandatory. Assume that there are four files (database tables) to store separately:
    a. Customer/bank teller ID/password
    b. Customer profile data
    c. Bank Account data
    d. Stock transaction data

Make your assumptions as necessary.

Use Case Model for Banking System



Use Case Description for Banking System

Use case name: Login
Summary: Customer (or Bank Teller) logs in the system.
Actor: Customer, Bank Teller
Precondition: The system is idle.
Main sequence:
1. Customer (or Bank Teller) selects login.
2. Customer (or Bank Teller) enters ID and password to the system.
3. System checks if the customer (or bank teller) ID and password are correct.
4. The system displays a welcome message if the ID and password are correct.

Alternative sequence:
- Step 3: If the ID or password is not correct, the system displays that the ID or password is incorrect.

Postcondition: Customer (or Bank Teller) has logged in the system.

Use case name: Log out
Summary: Customer (or Bank Teller) logs out system.
Actor: Customer, Bank Teller
Precondition: Customer (or Bank Teller) logged in the system.
Main sequence:
1. Customer (or Bank Teller) selects Log Out.
2. System prompts Customer (or Bank Teller) to verify log-out.
3. Customer (or Bank Teller) enters confirmation into the system.
4. System gets Customer (or Bank Teller) to log out if Customer (or Bank Teller) verifies log-out.

Alternative sequence:
- Step 2: If the Customer (or Bank Teller) does not confirm, the system does not log out.

Postcondition: The Customer (or Bank Teller) has logged out.

Use Case Name: Withdraw Funds
Summary: The Bank Teller withdraws a specific amount of funds from a valid customer bank account.
Actor: Bank Teller
Main sequence: Bank Teller has login the system.
Description:
1. Bank Teller selects Withdrawal, enters the amount, and selects the customer account number.
2. The system checks whether the customer has enough funds in the account.
3. If the account has enough funds, the system authorizes dispensing of cash.
4. Bank Teller confirms that he/she has handed over cash to the customer.
5. The system displays the system's initial menu.

Alternatives:
- Step 3: If the system determines that the customer account has insufficient funds, it displays an apology and terminates.

Postcondition: Bank Teller has withdrawn the funds.

Use Case Name: Query Account
Summary: Customer (or Bank Teller) views the balance of a valid account.
Actor: Customer or Bank Teller
Precondition: Customer or Bank Teller has login the system.
Main sequence:
1. Customer (or Bank Teller) selects Query and enters account number.
2. The system reads account balance.
3. The system displays the account balance if the account number is valid.

Alternatives:
Step 3: If the system determines that the account number is invalid, it displays an error message and terminates.
Postcondition: Customer or Bank Teller has looked at the account balance.
Use Case Name: Transfer Funds

Summary: Customer transfers funds from one valid bank account to another.
Actor: Customer
Precondition: Customer has login the system.
Main sequence:

1. Customer selects Transfer and enters `amount, from-account,` and `to-account`.
2. The system performs the transfer if the system determines that the customer has enough funds in the `from-account` and `to-account` is valid.

Alternatives:

- Step 2: If the system determines that the `from-account` number is invalid, it displays an error message and terminates.
- Step 2: If the system determines that the `to-account` number is invalid, it displays an error message and terminates.
- Step 2: If the system determines that there are insufficient funds in the customer's `from-account`, it displays an apology and terminates.

Postcondition: Customer funds have been transferred.

Use Case Name: View Profile
Summary: Customer (or Bank Teller) views his/her profile.
Actor: Customer or Bank Teller
Precondition: Customer (or Bank Teller) has login the system.
Main sequence:

1. Customer (or Bank Teller) selects View Profile (with customer name and SSN).
2. The system reads the customer's profile (name, SSN, address, phone, income, email).
3. The system displays the customer profile.

Alternatives: None
Postcondition: Customer (or Bank Teller) has looked at his/her profile.

Use Case Name: Buy Stock
Summary: Customer buys stocks using the funds in the bank account.
Actor: Customer, Stock Trading System
Precondition: Customer has login the system.
Main sequence:

1. Customer selects Buy Stock and enters the stock name, quantity, unit price, and account number to pay the stock.
2. The system checks whether the customer has enough funds to buy the stock in the account.
3. If the system determines that the account has enough funds, it submits the stock order to the stock trading system.
4. If the stock purchase order has been entered into a contract successfully, the system transfers the funds to buy the stock from the customer account to the stock trading system, and it receives the confirmation message from the stock trading system.
5. The system updates the customer's stock record with the stock name, quantity, buy unit price.
6. The system displays the completeness stock purchase to the customer.

Alternative sequence:

- Step 3: If the system determines that the account has insufficient funds, it displays an error message and terminates.
- Step 4: If the stock purchase order has not been entered into a contract, it displays a message and terminates.

Postcondition: Customer has bought stock.

Use Case Name: Sell Stock
Summary: The Customer sells the stock.
Actor: Customer, Stock Trading System
Precondition: Customer has login the system.
Main sequence:

1. Customer selects Sell Stock and enters the stock name, quantity, unit price, and account number to receives the funds for the stock sell.
2. The system checks whether the customer account is valid.
3. If the system determines that the account is valid, it submits the stock sell to the stock trading system.
4. If the stock sell is entered into a contract successfully, the system receives the stock sell funds from the stock trading system and transfers the funds to the customer bank account.
5. The system updates the customer's stock record with the stock name, quantity, sell unit price.
6. The system displays the complete stock sell to the customer.

Alternative sequence:

- Step 3: If the system determines that the account is invalid, it displays an error message and terminates.
- Step 4: If the stock sell is not entered into a contract, it displays a message and terminates.

Postcondition: Customer has sold stock.

Use Case Name: Query Stock
Summary: Customer (or Bank Teller) views the transactions of stock buy and sell.
Actor: Customer or Bank Teller
Precondition: Customer (or Bank Teller) has login the system.
Main sequence:

1. Customer (or Bank Teller) selects Query Stock (with customer name).
2. The system displays the customer's stock buy and sell transactions with buy or sell, stock name, quantity, and unit price.

Alternative sequence: None
Postcondition: Customer has looked at the transactions of stock buy/sell.