

k. shabane

19/09/2025

19/09/2025

Year 2

**PROGRAMME: BSC COMPUTER SYSTEMS ENGINEERING**

**CSE202- OBJECT ORIENTED ANALYSIS & DESIGN WITH JAVA**

Cse24

cse24-011@thuto.bac.ac.bw

Kutlo Shabane

Cse24-011

**Part A: Requirements Elicitation**

**1.Functional Requirements**

The core functions that the Banking System must perform are as follows:

* **Open an Account & Create Customer Profile:** The system must allow a new user to become a customer by providing personal details (e.g., first name, surname, address) and setting up secure login credentials. This process simultaneously creates the customer's profile and their first bank account.
* **User Authentication (Login/Logout):** The system must verify a user's identity via username and password to grant access to their accounts. A secure logout function must terminate the session.
* **Open Additional Account Types:** The system must allow an authenticated customer to open new, additional accounts. The system must support three specific types:
  + **Savings Account:** Allows deposits and earns monthly interest. **Withdrawals are not permitted.**
  + **Investment Account:** Requires an initial deposit of BWP 500.00. Allows deposits, withdrawals, and earns a higher monthly interest rate than a Savings account.
  + **Cheque Account:** Allows both deposits and withdrawals. To open one, the customer must provide employment details (company name and address).
* **Deposit Funds:** The system must allow a customer to add money to any of their accounts (Savings, Investment, or Cheque), instantly updating the balance and recording the transaction.
* **Withdraw Funds:** The system must allow a customer to remove money from their **Investment** and **Cheque** accounts, provided sufficient funds are available. The system must prevent withdrawals from **Savings** accounts.
* **Apply Interest:** The system must automatically calculate and add monthly interest to eligible accounts. The business logic specifies **5% for Investment accounts** and **0.05% for Savings accounts**. Cheque accounts do not earn interest.
* **Provide Transaction History:** The system must provide a detailed view of all past transactions (deposits, withdrawals, interest payments) for any selected account, showing date, time, amount, and balance.
* **Support Multiple Accounts per Customer:** The system's design must inherently support a one-to-many relationship where a single customer profile can own multiple accounts of any type.

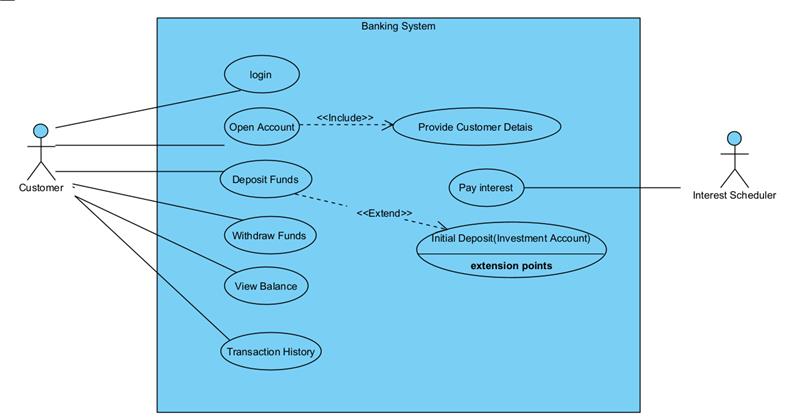
**2. Non-Functional Requirements**

The quality attributes and constraints the system must adhere to are:

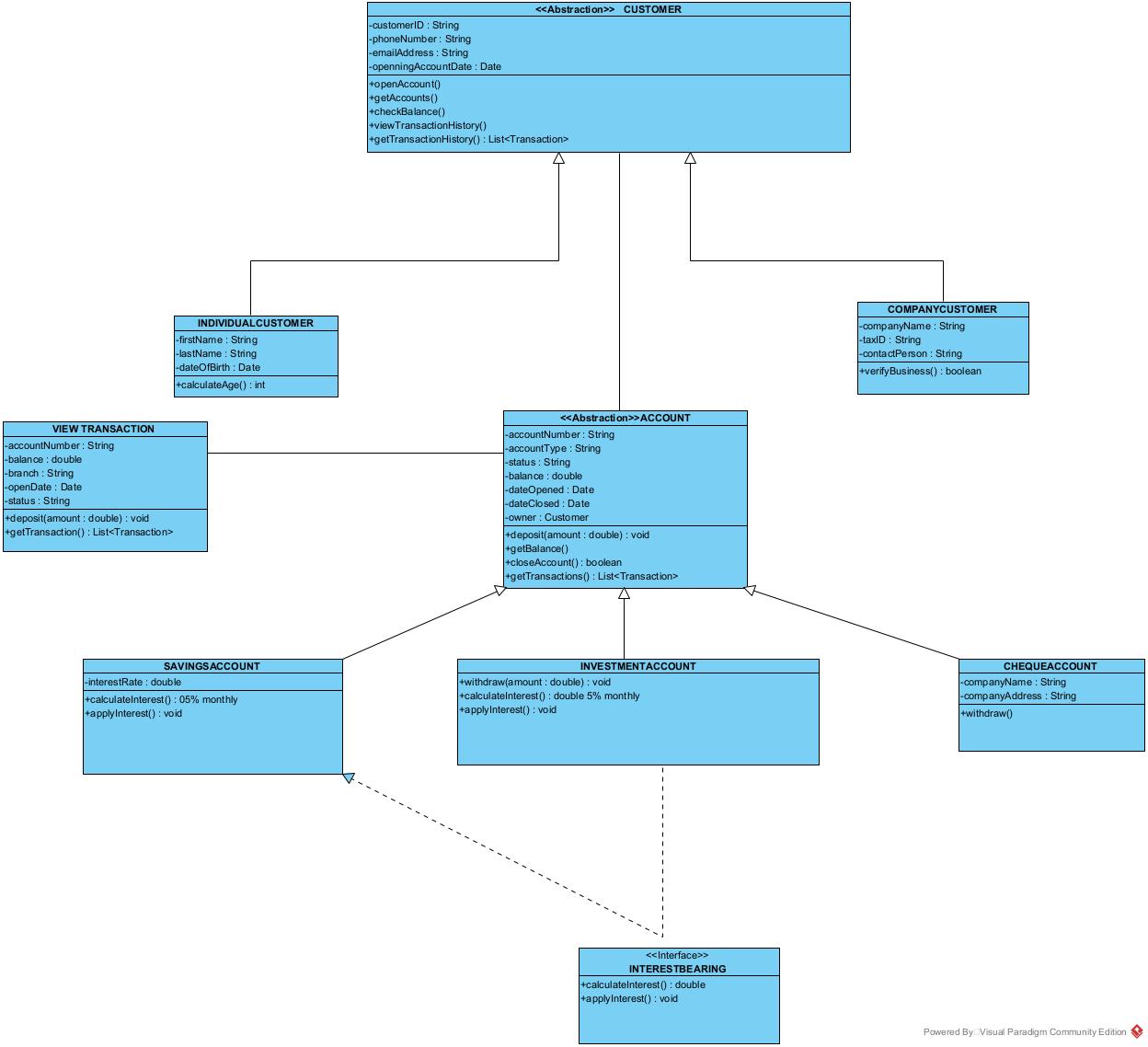
* **Security:** The system must implement strong password authentication. All sensitive data (e.g., passwords, account numbers, balances) must be encrypted both when stored in the database and during transmission to ensure customer data is protected.
* **Performance:** The system must be efficient and responsive. Key transactions such as deposits and withdrawals must execute and provide user confirmation in less than 2 seconds to ensure a seamless user experience.
* **Usability:** The system must feature a user-friendly graphical interface built with **JavaFX**. The design should be intuitive, with clear navigation and self-explanatory actions, making the system accessible to customers with varying levels of technical skill.
* **Scalability:** The system's architecture must be designed to support a growing number of concurrent users and a large volume of accounts without significant degradation in performance (e.g., slower response times).
* **Reliability:** The system must be stable and robust. This will be achieved through comprehensive exception handling to manage errors gracefully without crashing and by implementing a strategy for regular database backups to prevent data loss.

2. Structural UML Modelling

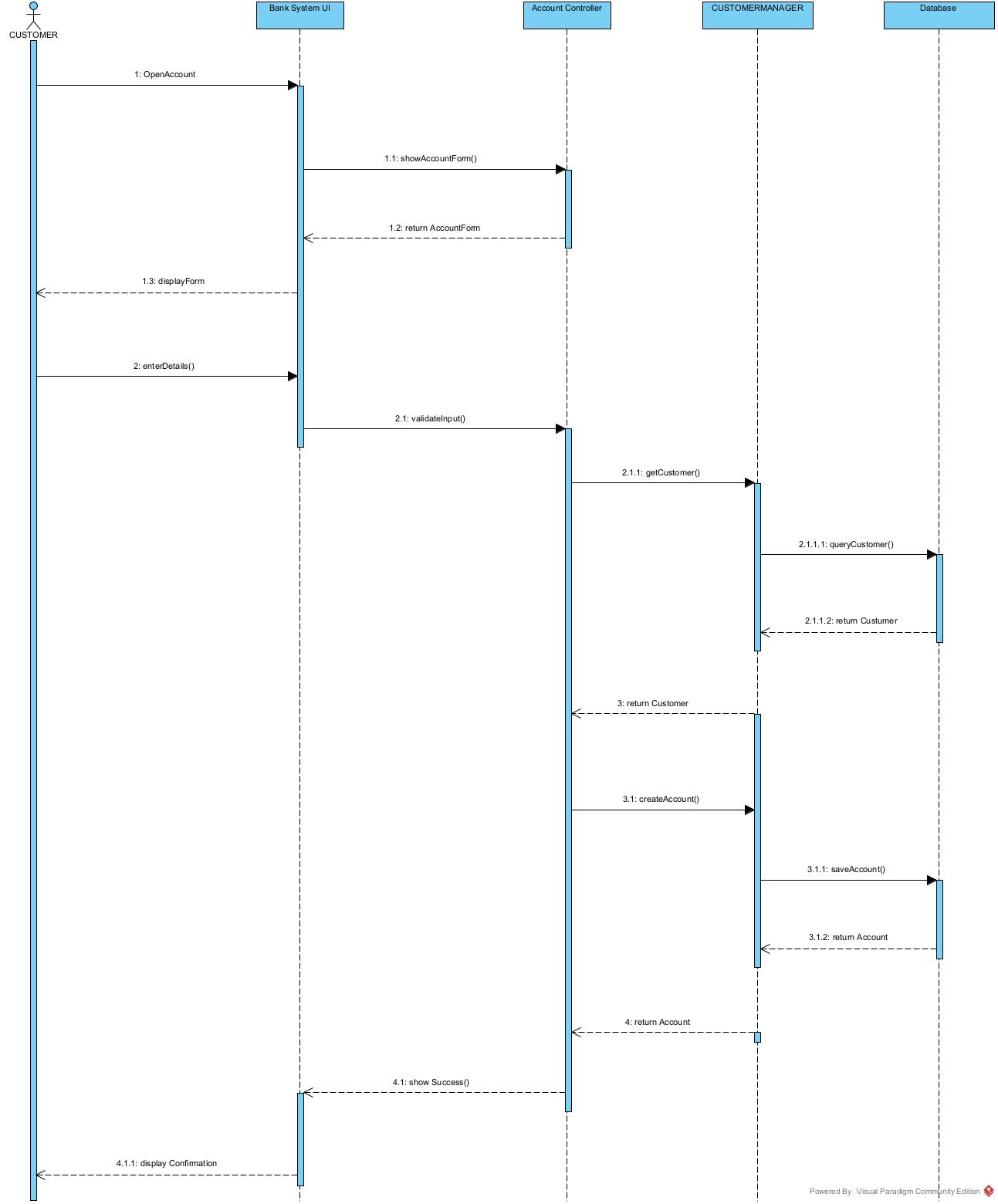
* 1. **SYSTEM USE CASE DIAGRAM**

****

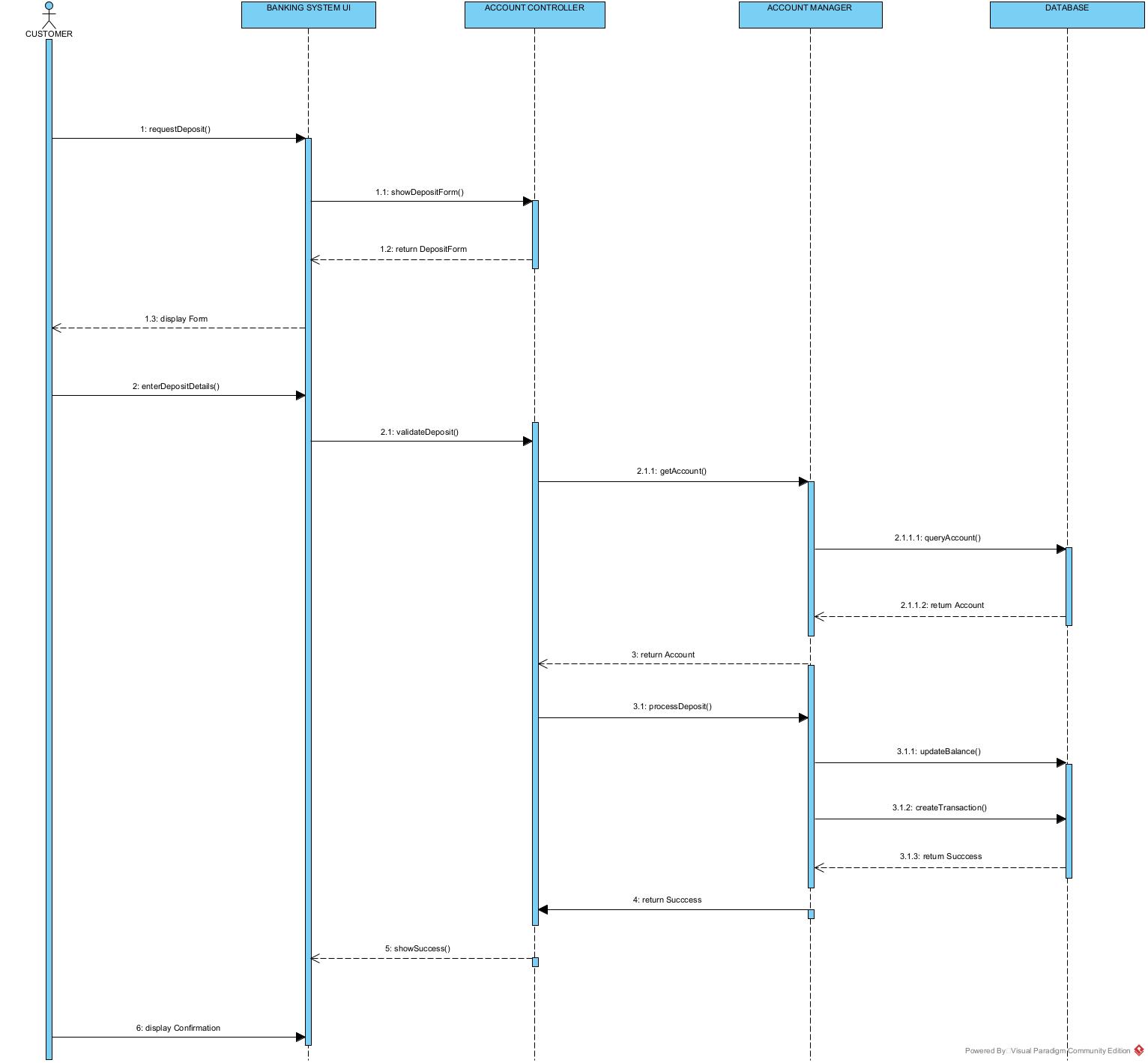
* 1. **CLASS DIAGRAM**

****

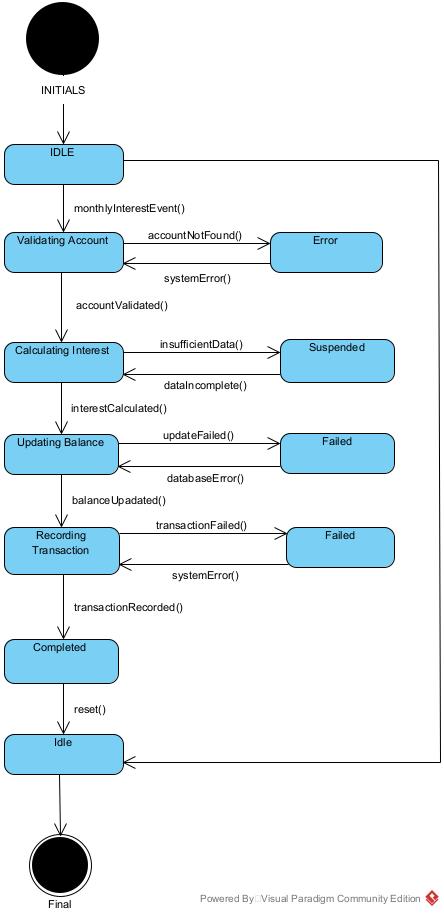
1. **BEHAVIOURAL UML MODELLING**
   1. **SEQUENCE DIAGRAMS**
   2. **OPEN ACCOUNT**

****

* 1. **DEPOSIT**



* 1. STATE DIAGRAM



**Appendix A – Interview Record**

**Project**: Banking System (OOAD Assignment)  
**Date of Interview**: 18 September 2025, 10:30 AM  
**Interview Type**: Group Interview  
**Interviewee**: Mr. Kentsenao Baseki   
**Interviewers**: Class Students

**QUESTIONS:**

1. What are the main services the system should provide?  
   **A:** The system should allow customers to register, login, open accounts, deposit funds, withdraw funds, and view transactions.
2. What security concerns should be addressed?  
   **A:** The system should have password protection, encryption of sensitive data, and prevent unauthorized access.
3. What types of accounts must be supported?  
   **A:** Savings, Investment, and Cheque accounts.
4. What are the expectations for performance?  
   **A:** Transactions should be processed within a few seconds, even when multiple users are online.
5. The interest rates were specified as 5% for Investment and 0.05% for Savings accounts, applied monthly.

|  |  |
| --- | --- |
| What types of accounts are supported? | Savings Account, Investment Account, Cheque Account. |
| Minimum deposit required? | Savings: **P50**, Investment: **P500**, Cheque: no minimum. |
| Who opens accounts? | **Bank teller** (not the customer). Customers provide details; teller captures them. |
| Can customers have multiple accounts? | Yes, but **not multiple of the same type**. A customer may hold a Savings + Investment + Cheque. |
| What customer details are required? | **Individual**: Name, Surname, National ID, Address, Date of Birth, Contact, Next of Kin.  **Company**: Company Name, Company Number, Date of Incorporation, Address, Contact, Signatories. |
| How is interest handled? | **Automated monthly**: applied to Savings (0.025) and Investment (0.075 for company accounts), credited to balance, and recorded in transaction history. |
| What can customers do with accounts? | Deposit funds, Withdraw funds (except investment withdrawals), View account balance, View transaction history. |
| What should transaction history include? | Date, Transaction Type (Deposit/Withdrawal/Interest), Amount, Credit/Debit, New Balance. |
| Should monthly statements be generated? | No separate reports — **transaction history serves as statement**. |
| Should failed login attempts be handled? | **Not required**. Focus is on OOP design, not advanced security. |
| What happens if a withdrawal exceeds balance? | System must display **error/insufficient funds message** and cancel the transaction. |