**OBJECT ORIENTED PROGRAMMING WITH JAVA ASSIGNMENT 2025**

**Requirements Elicitation**

**Functional Requirements (What the system MUST DO):**

**Account Management:**

* + A bank teller (not the customer) must be able to open new accounts.
  + The system must support three account types: Savings, Investment, and Cheque.
  + Each new account must be opened with a minimum balance:
    - Savings Account: 50
    - Investment Account: 500
    - Cheque Account: 0
  + The system must prevent an account from being created if the initial deposit is below the minimum.
  + A customer (individual or company) can hold multiple accounts of different types.

**Customer Management:**

* + The system must capture customer details during the account opening process.
  + It must distinguish between Individual Customers and Company Customers.
    - Individuals require details like first name, surname, national ID.
    - Companies require details like company name, company registration number, date of incorporation.
  + All customers share common attributes like address and contact number.

**Transactions:**

* + A customer must be able to deposit funds into any of their accounts.
  + A customer must be able to withdraw funds from Savings and Cheque accounts.
  + Withdrawals must be blocked for Investment accounts.
  + The system must validate withdrawals and prevent them if there are insufficient funds.
  + The system must display an error message for invalid transactions (e.g., insufficient funds, withdrawal from investment account).

**Interest Calculation:**

* + The system must automatically calculate and apply monthly interest with no manual approval.
  + Interest rates must be applied as follows:
    - Individual Savings Account: 0.025%
    - Company Savings Account: 0.075%
  + The calculated interest must be credited to the account and recorded as a transaction.

**Transaction History & Reporting:**

* + The system must maintain a complete transaction history for every account.
  + Each transaction record must include:

**Date and time**

* + - Transaction type (e.g., Deposit, Withdrawal, Interest)
    - Amount
    - Resulting balance
  + A customer must be able to view their account balance and transaction history for any of their accounts.

**User Authentication:**

* + Users must log in to access the system.
  + Basic authentication is required (username/password).

**Non-Functional Requirements (How the system should PERFORM):**

**Usability:**

* + The user interface must be modern, intuitive, and user-friendly.
  + It should use intuitive colors and designs (e.g., green for positive actions like "Confirm", red for negative actions like "Cancel").
  + The layout must be logical and easy to navigate.

**Performance:**

* + The system should efficiently handle a small scale of at least 10-20 records per database table.
  + Support for a large number of concurrent users is not a primary requirement for this project.

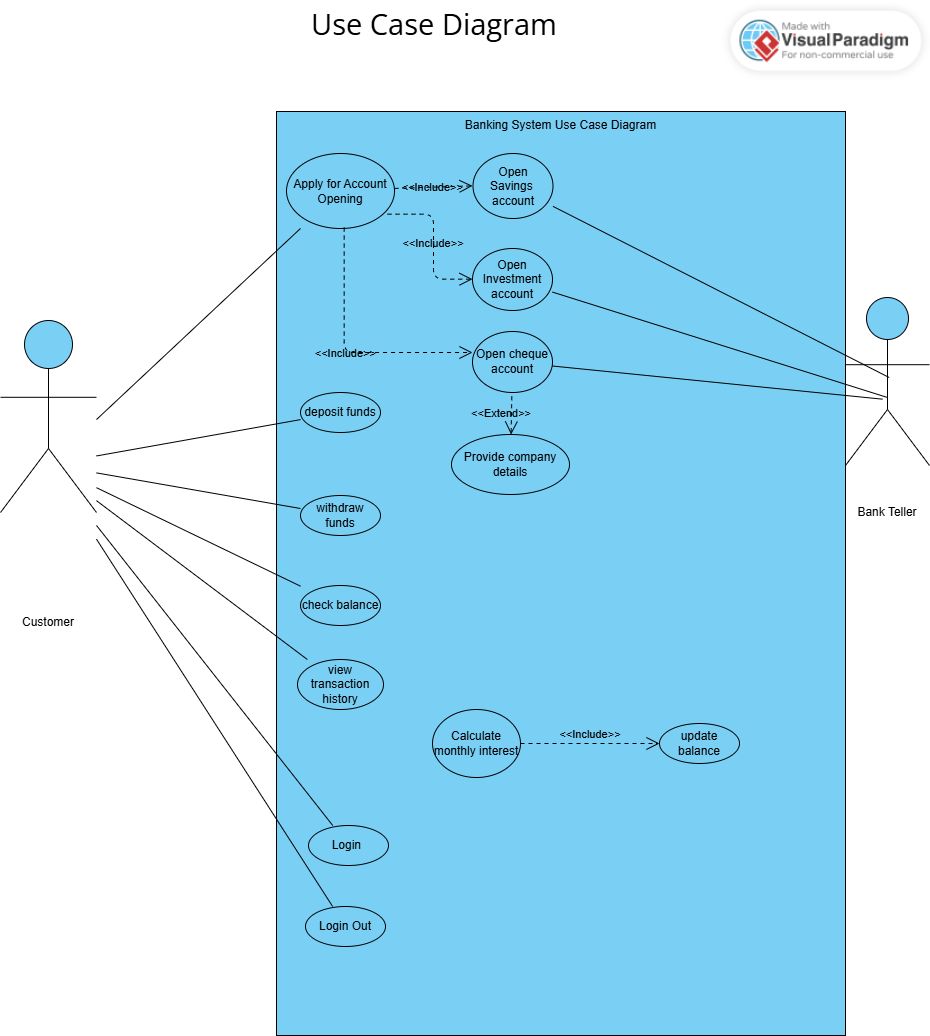
**Security:**

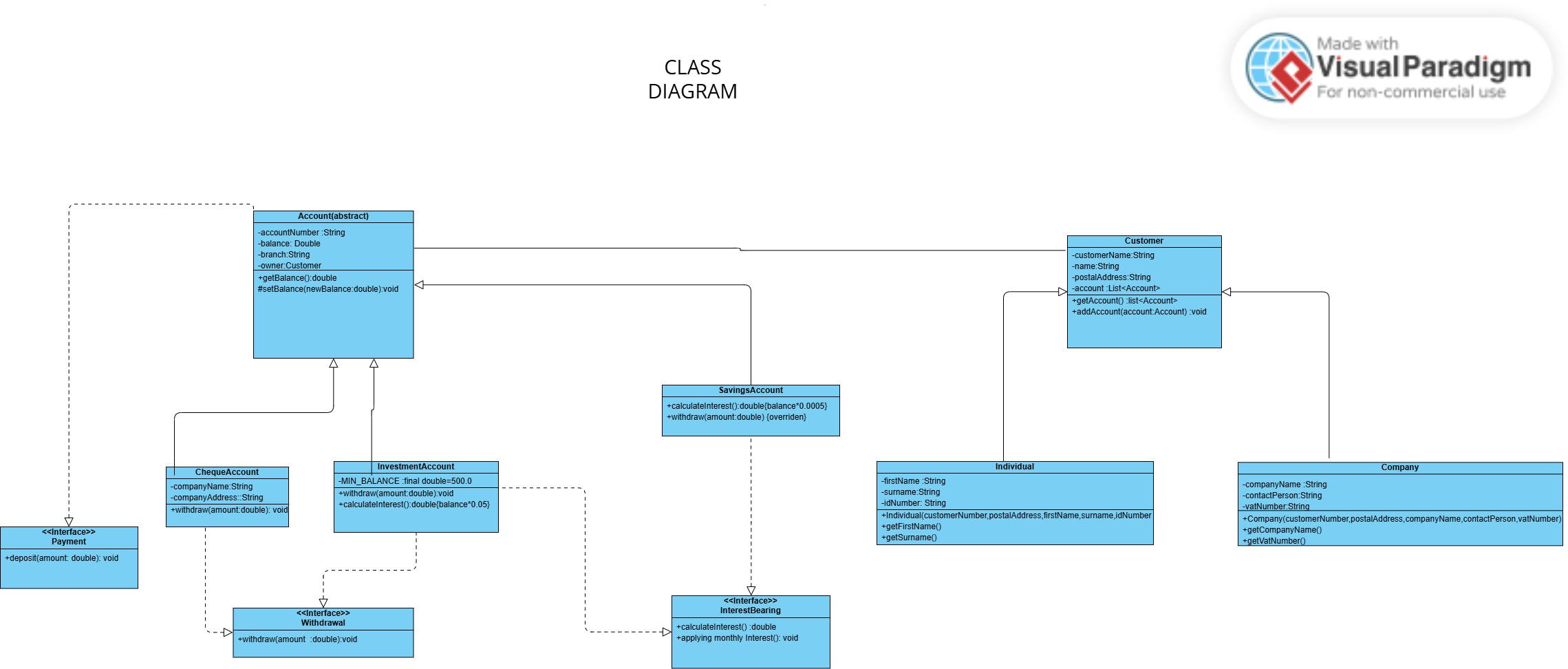
* + Basic user authentication (login) is required.
  + Advanced security features like account lockouts after failed attempts or encryption are explicitly out ofscope.

**Reliability & Automation:**

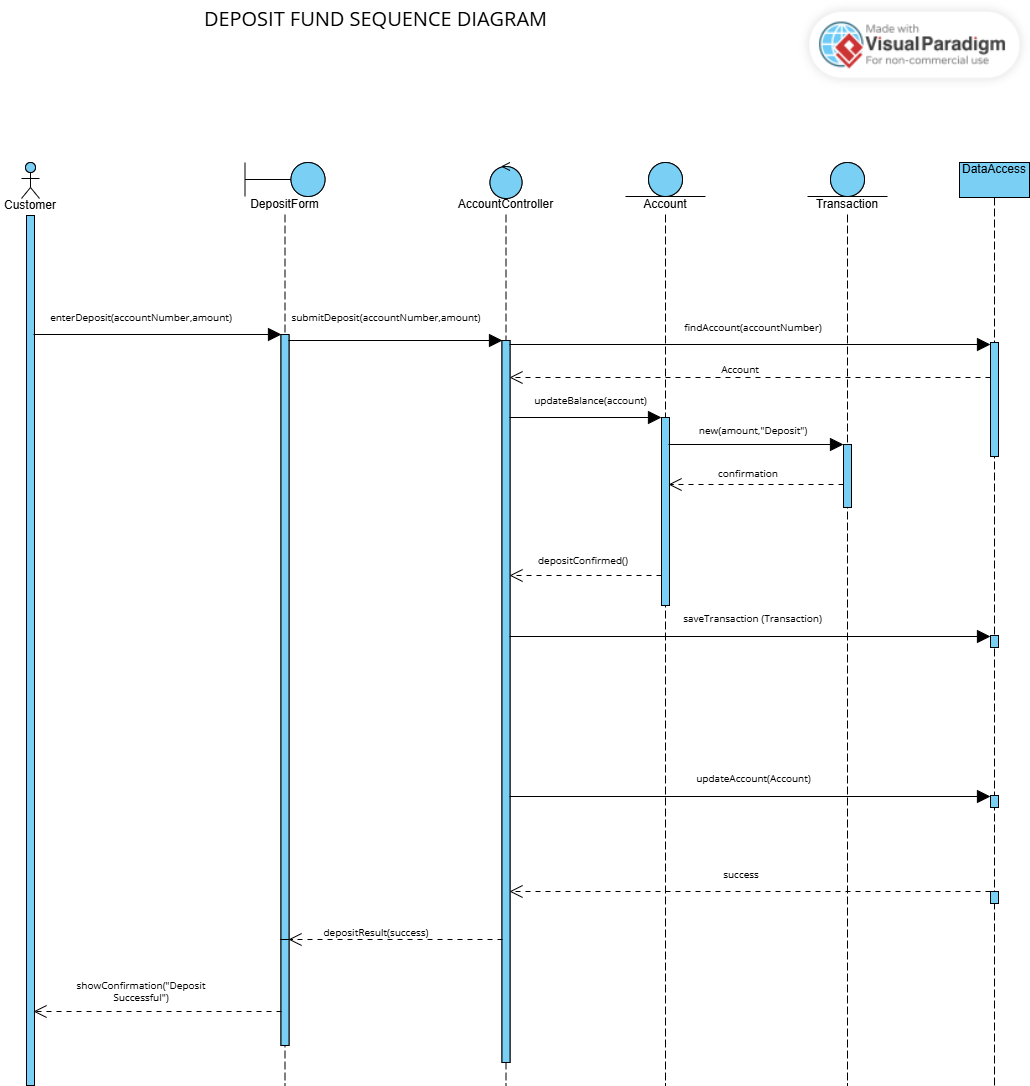
* + Key processes, like interest calculation, must be fully automated and reliable.
  + **Data Integrity:**
  + The system must accurately record all transactions and maintain correct account balances at all times.

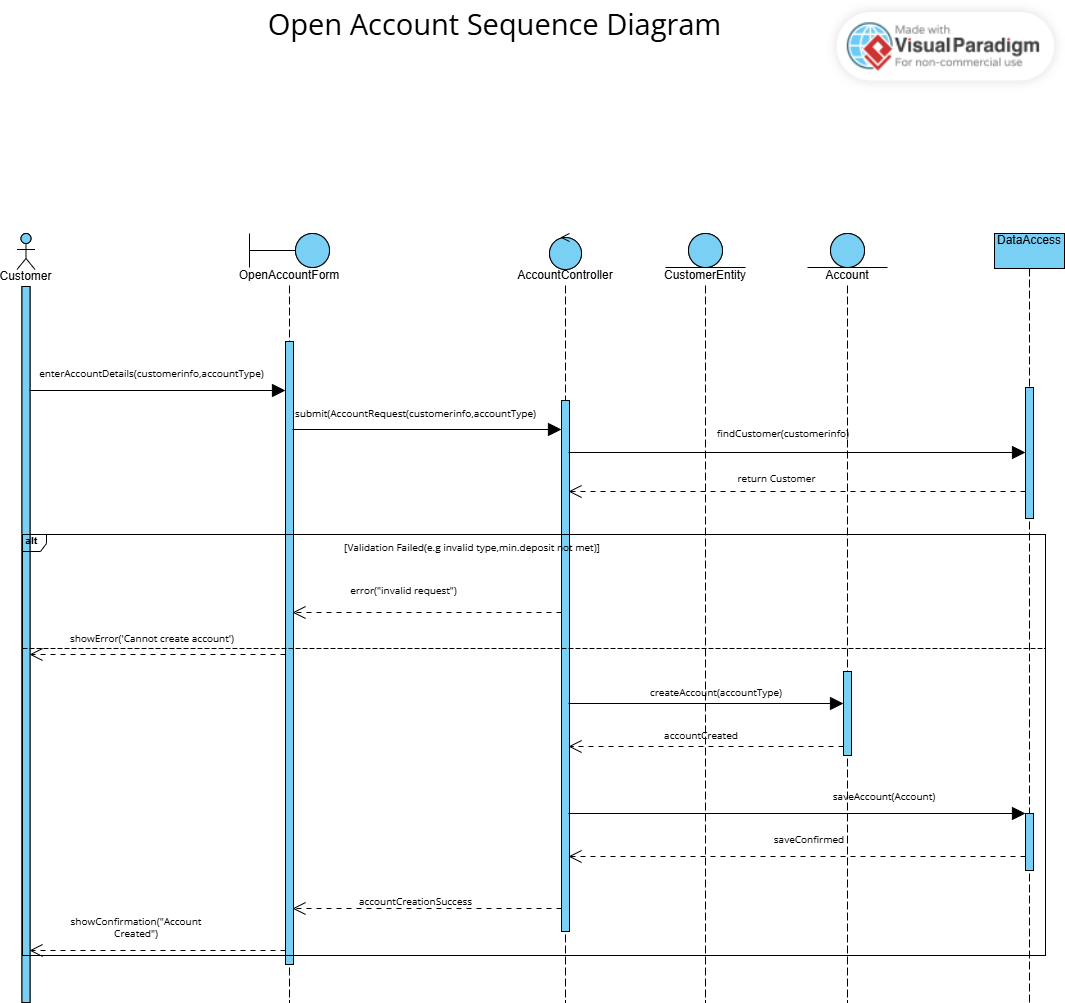
**Structural UML Modelling**

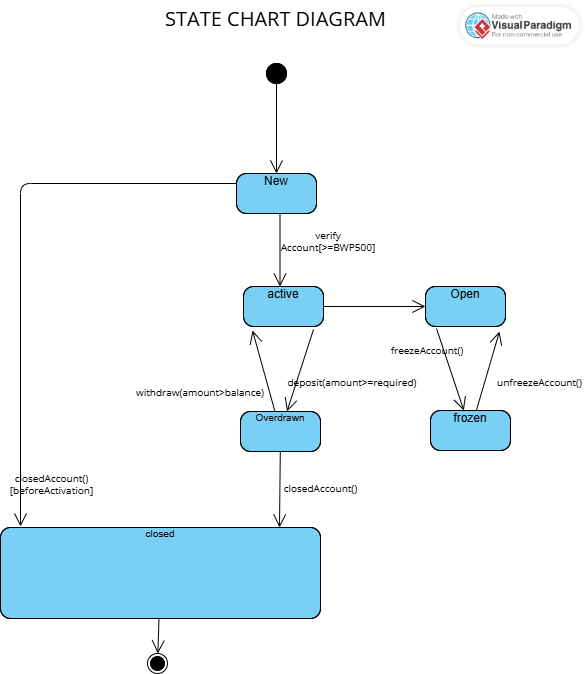
****

****

**Behavioural UML Modelling**

****

****

****

**INTERVIEW RECORD APPENDIX**

**Interviewer**: Oarabile Botho Zongwani, Student System Analyst

**Interviewee**: Mr Kentsenao Victor Baseki (LECTURER)

**Date**:18 September 2025

**Time**:10:30-12:20

**Purpose**: To elicit and clarify the core functional and non-functional requirements for the new Banking System

**Transcript Excerpt**

**FUNCTIONAL REQUIREMENTS**

1. Interest Calculation & Automation

* **Question:** Is the application of monthly interest fully automated, or does it require manual approval from a bank employee?
* **Elicited Requirement:** The system shall automatically calculate and apply monthly interest to savings and investment accounts without human intervention.

2. Transaction History for Interest

* **Question:** After the system automatically applies interest, is this recorded in the transaction history? If so, how is it displayed?
* **Elicited Requirement:** The system shall record the application of monthly interest as a credit transaction in the account's transaction history, showing the date, the interest amount credited, and the new balance.

3. Customer Information Attributes

* **Question:** Beyond basic details like name and address, what specific, critical information must we store for individual customers versus company customers?
* **Elicited Requirement:** The system shall capture:
  + For Individuals: National ID number, date of birth, and contact number.
  + For Companies: Company registration number, date of incorporation, and authorized signatories.

4. Unique Identification

* **Question**: What serves as the unique identifier for a company customer since they don't have a national ID like an individual?
* **Elicited Requirement:** A company's unique identifier shall be its company registration number (e.g., BW000123).

5. Login Security Rules

* **Question:** Should the system handle incorrect login attempts, for example, by locking the account after three failed tries?
* **Elicited Requirement**: (Scope Creep Warning) The system need not implement advanced login security features like attempt limits. The focus is on core banking functionality. *(This clarifies that it is* out *of scope)*.

6. Fund Transfer Functionality

* **Question:** Can a customer transfer funds between their own accounts?
* **Elicited Requirement:** (Scope Creep Warning) The system shall not include a funds transfer feature between accounts. *(This clarifies that it is* out *of scope)*.

7. Multi-User Concurrency

* **Question:** Does the system need to support multiple users (customers) banking simultaneously?
* **Elicited Requirement:** (Scope Creep Warning) The system is a demonstration model and does not need to be designed for multi-user concurrency or performance under load. *(This clarifies that it is* out *of scope)*.

8. Account Eligibility

* **Question:** Are there eligibility requirements, like age restrictions, for opening certain accounts (e.g., an investment account)?
* **Elicited Requirement:** The system shall not enforce eligibility checks beyond the minimum balance requirement. All customers can open any account type.

9. Error Handling for Transactions

* **Question**: If a transaction fails (e.g., a withdrawal due to insufficient funds), what should the system display to the user?
* **Elicited Requirement:** The system shall display a clear error message (e.g., "Insufficient funds") on the user interface. A failed transaction shall not be recorded in the transaction history.

10. Report Generation

* **Question:** Does the system need to generate any reports, like monthly statements?
* **Elicited Requirement**: The "View Transaction History" functionality shall serve as the account statement report. No separate report generation feature is required.

11. Account Closure

* **Question:** Can a customer close an account? If so, what happens to the remaining balance?
* **Elicited Requirement**: (Scope Creep Warning) The system shall not include an account closure feature. *(This clarifies that it is* out *of scope)*.

**NON-FUNCTIONAL REQUIREMENTS**

12. Currency and Limits

* **Question**: What currency should the system use (e.g., USD, BWP)? Are there limits on deposits or withdrawals for each account type?
* **Elicited Requirement:**
  + Usability/Flexibility: The system can use any currency denomination. The specific currency is not critical to the design.
  + Business Rules: The only limits are the minimum balances to open an account (Savings: 50, Investment: 500). There are no predefined limits on transaction amounts.

13. Look and Feel (Usability)

* **Question:** When you say the interface must represent "current technology," what does that mean? What are the expectations for the GUI design?
* **Elicited Requirement:**
  + Usability: The user interface shall be intuitive and user-friendly.
  + Aesthetics: It shall use modern design conventions (e.g., color-coded buttons: green for confirm, red for cancel) and not be a plain, gray interface. The design should be logical and visually coherent.

14. Performance & Scalability

* **Question:** (Implied from the multi-user question) How many users should the system support at once?
* **Elicited Requirement:**
  + Performance: The system is a prototype. Performance and scalability for a large number of concurrent users are not a concern for this project.

15. Data Integrity & Validation

* **Question**: (Implied from the error handling question) How should the system handle invalid data or transactions?
* **Elicited Requirement:**
  + Reliability: The system shall validate transactions (e.g., prevent overdrawal) and provide immediate feedback to the user.
  + Data Integrity: Failed transactions shall not corrupt account balances or history.

**In conclusion**

The interview with the Lecturer confirmed the critical nature of enforcing business rules (e.g., minimum balances, account type restrictions). It also highlighted key non-functional requirements, particularly concerning security, data integrity, and system performance.