Assignment Cover Page

Part A: System Documentation

# 1. Requirements Elicitation

## 1.1. Functional Requirements

## 1.2. Non-Functional Requirements

# 2. Structural UML Modelling

## 2.1. System Use Case Diagram

## 2.2. Class diagram

# 3. Behavioural UML Modelling)

## 3.1. 'Login' & 'Withdraw' Sequence Diagrams

## 3.2. State Diagram

**Meeting record appendix**

# 1. Requirements Elicitation

## 1.1. Functional Requirements:

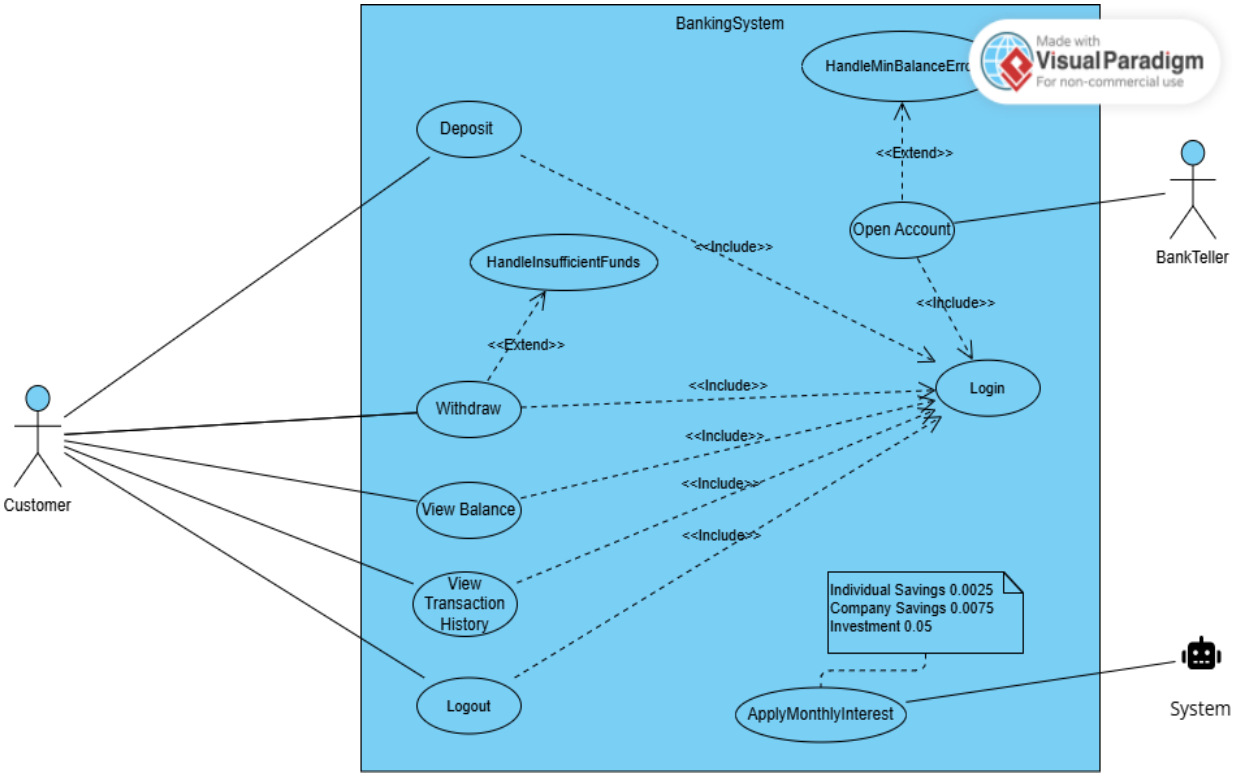
- The Bank(teller) shall Open accounts for Customers with account specific rules (P50 minimum for Savings, P500 minimum for investment).
- A customer shall be able to deposit, withdraw, and view account balances.
- One customer may hold multiple (types of) accounts (Savings, Cheque, Investment) – only one of each type.
- The System shall automatically credit interest monthly using specified rates.
- A customer shall be able to view a transaction history showing date, type (deposit/withdrawal/interest), debit/credit, amount and updated balance.

## 1.2. Non-Functional Requirements:

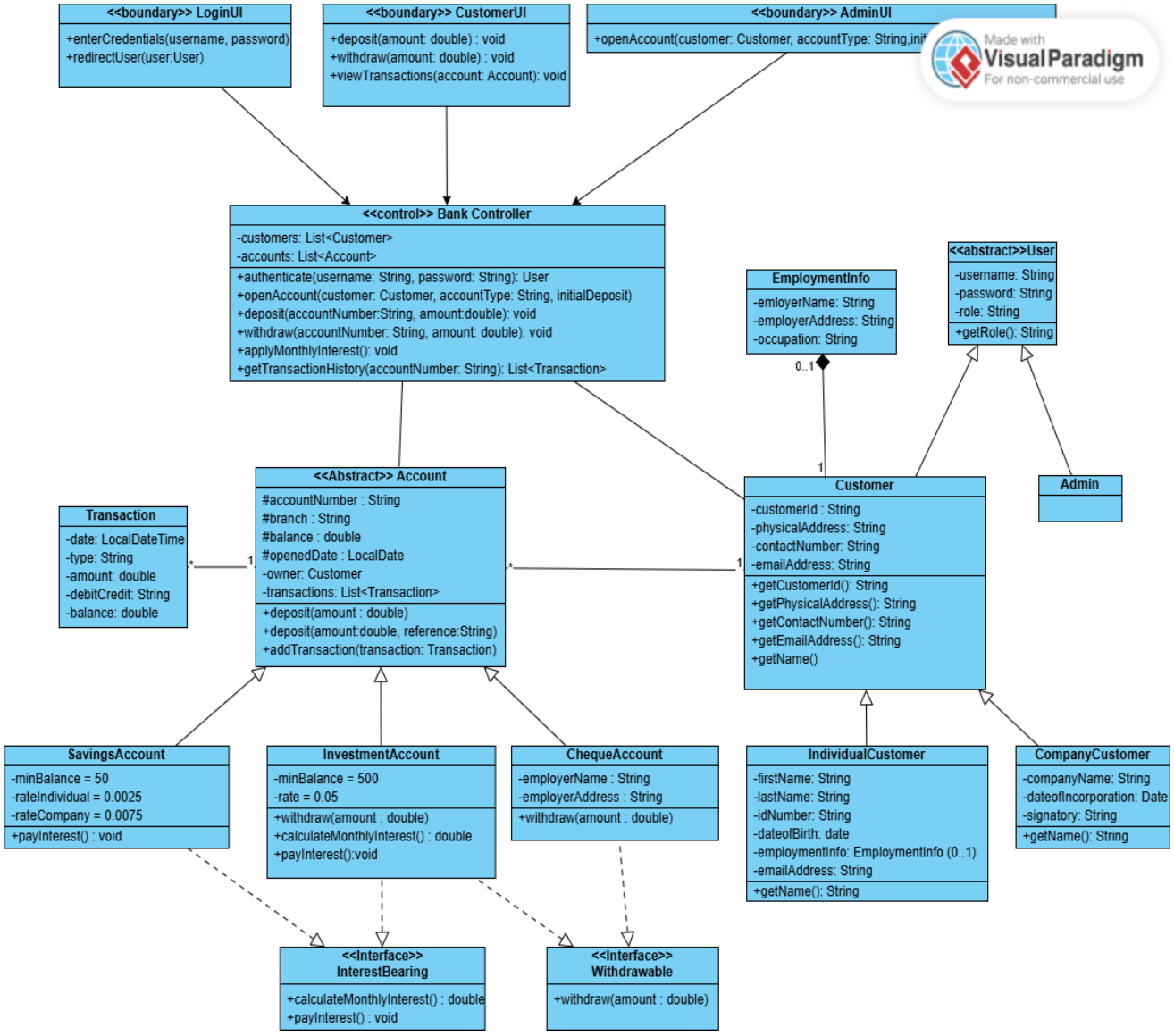
- The system shall authenticate Users before granting access.
- The system shall contain atleast 10 to 20 Records per Database table for testing.
- The system shall display alerts for failed transactions.
- The user interface shall be colourful, creative and intuitive, reflecting modern design trends.
- The system shall be designed using OOP principles to allow for future extension without modifying existing structures.

## 2. Structural UML Modelling

### 2.1 System Use Case Diagram

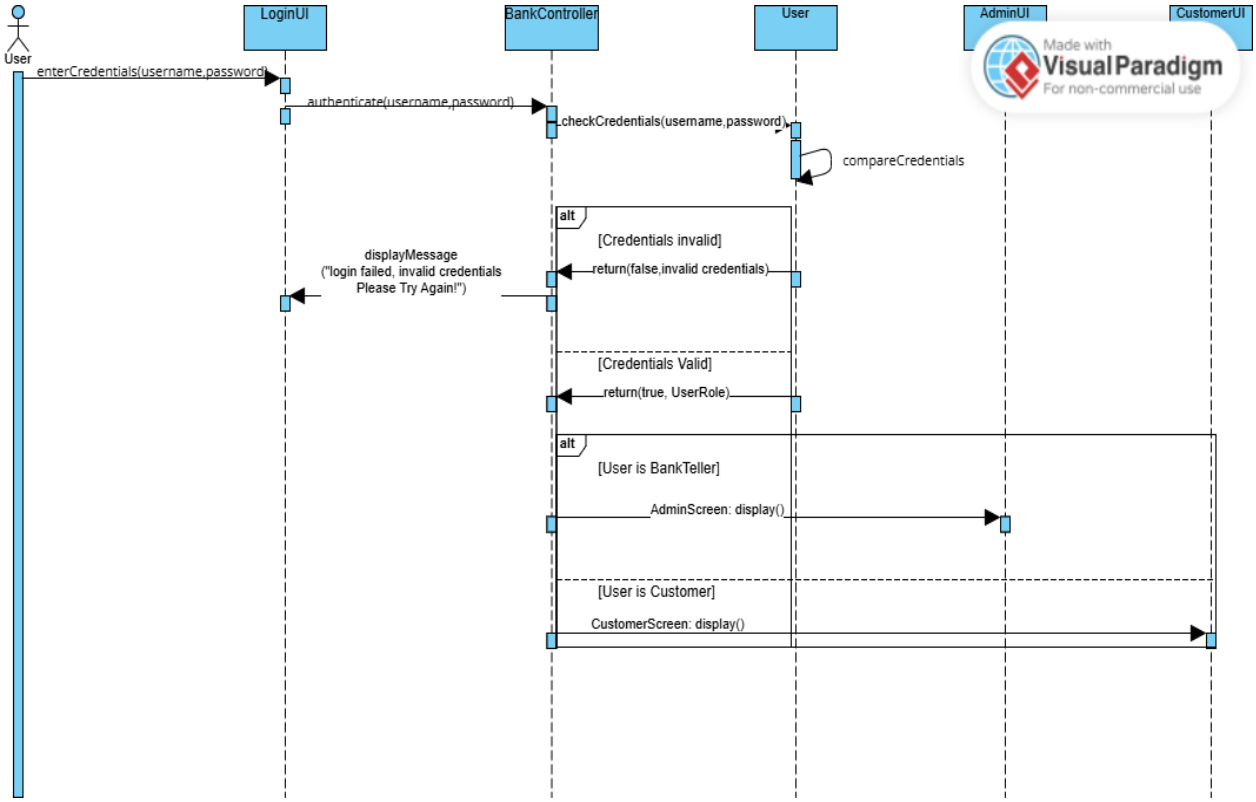


## 2.2 Class Diagram

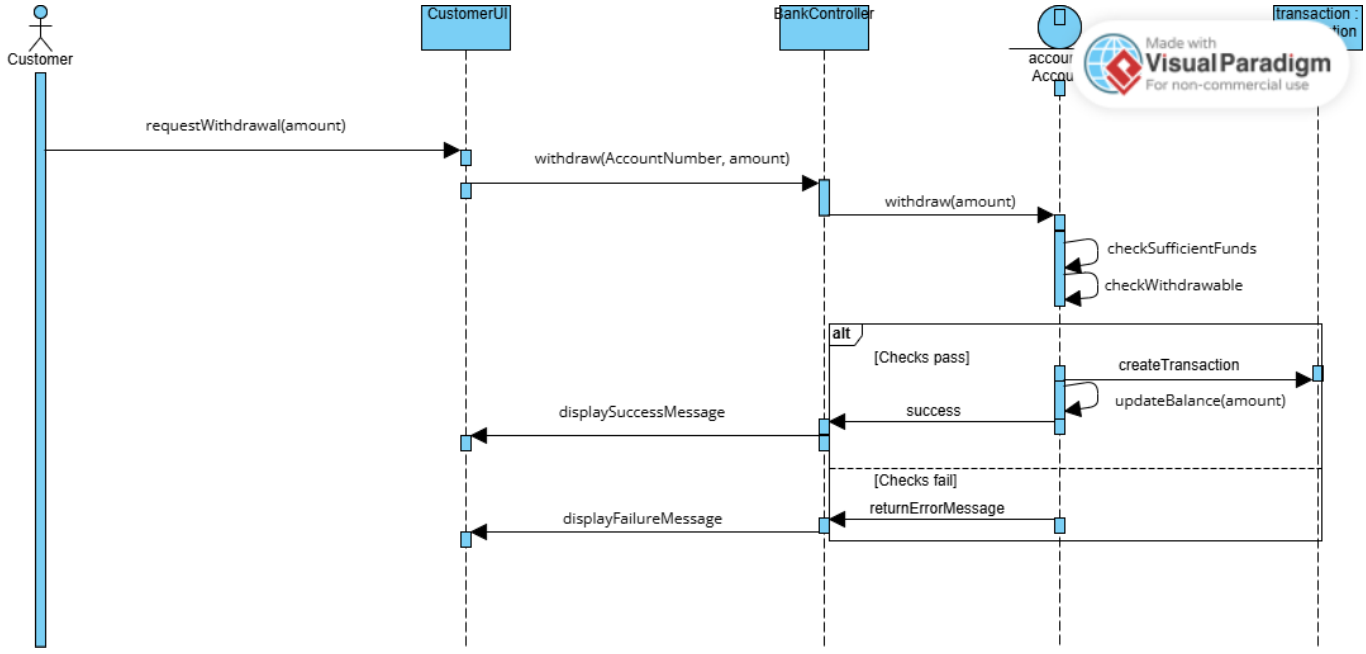


# 3. Behavioural UML Modelling

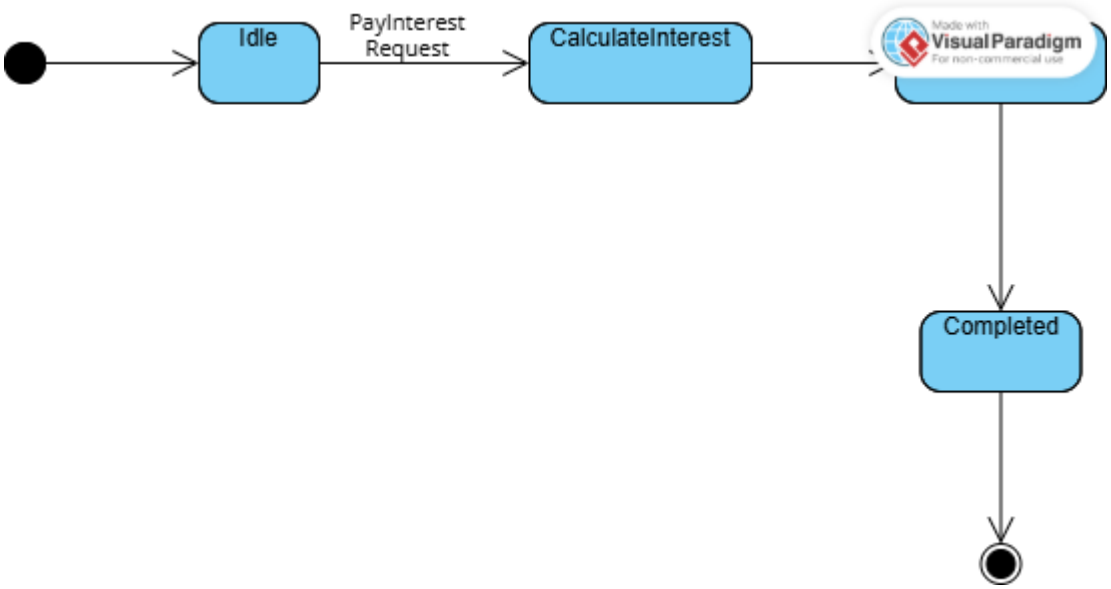
## 3.1 Login Sequence diagram




## 3.1 Withdraw Sequence Diagram



## 3.2 State Diagram



# Meeting Record Appendix

## Requirements Elicitation for Banking System

**Interview Date**: 18-September-2025, 1030hrs – 1216hrs CAT

**Mode:** Teams Virtual Meeting

**Interviewers:** CSE-24 Year 2 Students, 90 students in attendance.

**Interviewee:** Mr. Kentsenao Baseki, Client

## **General Overview** (Mr. Baseki):

The assignment scenario itself is quite general, so analysis is required as you design. For example, A customer, includes companies as well – the simple "first name, last name" given in the scenario would not be appropriate for a company.

Furthermore, an Investment Account must have a minimum balance of P500. This means that as the account is created, the system should throw an error if at creation, the balance is less than P500. Same applies to P50 for the Savings Account.

Monthly Interest for a Company Savings Account is 0.075%, and for an Individual Savings account, 0.025%. You will need to show Abstraction and Inheritance when applying these.

A Customer (company or individual) can have multiple accounts. This does not mean one customer can have many Savings accounts, for example. But it means one customer can have multiple (types) of accounts – A customer having a Savings Account, Investment Account, and a Cheque Account.

The system should "Open an Account", not register a customer. You will have to Capture customer details to open an account; however, a bank does not register customer. – Lest we have customers that are not tied to an account. And, the Bank opens the account, not the customer. The customer logins, views balance, deposits, withdraws.

A Transaction history should be visible for any account held. Details should include date, type (deposit/withdrawal/interest earned), amount, credit, debit, balance thereafter.

Non-Functional requirements include user authentication, 10-20 Records for each Database Table, and a UI Representative of the current trends of

technology - Colourful & Creative - instead of basic Grayscale – which would earn you less marks.

## Questions and Answers:

The following is a record of relevant questions posed by students, and answers given by the client, Mr. Baseki.

**Question - Olerato:** Should monthly interest be fully automated or require approval by admin?
**Answer – Mr. Baseki :** Fully automated.

**Question - Anita:** Should the monthly statement reflect interest?
**Answer – Mr. Baseki:** The value applied to the balance, should reflect on the statement as credited, and then the current balance updated.

**Question - Pitso:** What critical information needs to be collected for customers?
**Answer – Mr. Baseki:** As an analyst, you determine what is needed. You should have a General customer. Then differentiate as necessary with each customer, e.g. Next of Kin for an Individual vs Signatory for a Company, and Date of Birth for an Individual Account vs Date of Incorporation for a company.

**Question – Pitso :** Since individual customers have an ID number, what would be the Unique identification for company customers?
**Answer – Mr. Baseki:** A Database Schema might put them in different tables, but in general they're all customers. Joins. Modelling, Customer Id person National ID, Company – Company Number.

**Question - Felix:** Should an account lock a customer out after multiple login attempts?
**Answer – Mr. Baseki:** A key aspect of the assignment is to assess your ability to analyse and design a system using OOP. Functionalities are not much of an issue, but due diligence is required in the design of OO Models - Representing a true structure of a system - the features are not as important. The "Open for extension, closed for modification" Rules means the system can always be improved later. Therefore, the Application of OO Concepts and Principles is what matters most.

**Question – Alex:** Could we add the security feature as an added functionality?
**Answer – Mr. Baseki:** The Client's budget is 10 marks: Be weary of doing things that will not earn you marks. Doing only what is needed is a form of requirements-management.

**Question – Tlotlo:** Can customers transfer funds between their accounts?
**Answer – Mr. Baseki:** Agreed by Interviewers and Interviewee – Do not add such functionality.

**Question – Anita:** Does the system support banking at the same time?
**Answer – Mr. Baseki:** That would be performance and scalability. It would involve distribution, etc. Non-functional requirements are not much of an expectation here, but applying OOP concepts is. Let's not dwell much on those.

**Question – Pitso:** What kind of Currency should we use? Are there limits on deposits and withdrawals?
**Answer – Mr Baseki:** Anyone can decide the currency they want to use. Focus is not on nitty grities of the data. As long as the data types are representative of the object picked from requirements elaboration. Limits – P50 Savings, P500 Cheque Account, P0 – Cheque Account. Insufficient funds when customer tries to deduct more than limit. We don't want scope creep.

**Question – Tadiwanashe**: Should there be GUI displays for failed transactions?
**Answer – Mr. Baseki:** It's Given, that there should be an alert message on the screen.

**Question – Tadiwanashe:** Should the system generate reports?
**Answer – Mr. Baseki:** Transaction History is the report in this instance. Let's draw the line there.

**Question – Tumisang:** Can a customer close an account? If so, what happens to the balance?
**Answer – Mr. Baseki:** Let's not include that.