## 1. Requirements Elicitation

### 1.1 Functional Requirements

The Banking System should provide the following functionality:

* Allowing new addition of customer profiles with personal information included
* Allowing users to open different types of accounts (Savings, Investment, Cheque)
* Depositing funds into any account type
* Only allow withdrawal from Investment and Cheque accounts (not Savings accounts)
* Calculate and apply interest rates monthly:
  + 5% for Investment accounts
  + 0.05% for Savings accounts
  + No interest in Cheque accounts
* Update account balances accurately after each transaction
* Display account information and balances for customers
* Enforce business rules:
  + Minimum deposit of BWP 500 for Investment accounts
  + Employment information required for Cheque accounts
  + No withdrawals from Savings accounts

### 1.2 Non-Functional Requirements

* **Security:** Sensitive customer data must be protected from unauthorized access
* **Performance:** The system should respond to user requests within 2 seconds
* **Usability:** The interface should be easy to use for both bank employees and customers
* **Reliability:** The system must maintain data integrity with no loss of transaction records
* **Availability:** The system should be available during banking hours (8am-5pm) with 99% uptime
* **Maintainability:** Code should be well-structured and documented for future enhancements

## 2. Structural UML Modeling

### 2.1 System Use Case Diagram

### 

### 2.2 Class Diagram

### 

**OOP Principles Demonstrated:**

* **Abstraction:** Account class is abstract with abstract calculateInterest() method
* **Inheritance:** SavingsAccount, InvestmentAccount, and ChequeAccount inherit from Account
* **Interface:** InterestBearing interface is implemented by interest-bearing accounts
* **Polymorphism:** Different account types implement calculateInterest() differently
* **Encapsulation:** All fields are private with public getter methods

## 3. Behavioral UML Modeling

### 3.1 Sequence Diagram - Deposit Funds

### 

### 3.2 Sequence Diagram – Login Process

### 

### 3.3 State Diagram - Account Object

### 

Banking System - Part A: System Documentation

Module: CSE202 - Object Oriented Analysis & Design with Java

Submitted: System 2025