Appendix 1: Assignment submission cover sheet

**ASSIGNMENT SUBMISSION COVER SHEET**

|  |
| --- |
| Cse24-009 |

**Student Id:**

|  |
| --- |
| Kaelo Lwande Kgosidialwa |

**Student names:**

|  |
| --- |
| Cse24-009@thuto.bac.ac.bw |

**Student email**:

|  |
| --- |
|  |

**Cohort**:

|  |
| --- |
| SCENARIO: The BANKING System |

**Assignment title**:

|  |
| --- |
| 17 November 2025 |

**Date of submission**:

|  |
| --- |
| BSC COMPUTER SYSTEMS ENGINEERING |

**.**

**Programme of Study**:

|  |
| --- |
| 2025 |

**Year of Study**:

**Intellectual property statement**

By checking the box below, I certify that this assignment is my own work and is free from plagiarism. I understand that the assignment may be checked for plagiarism by electronic or other means and may be transferred and stored in a database for the purposes of data-matching to help detect plagiarism. The assignment has not previously been submitted for assessment in any other unit or to any other institution.

**have read and understood the Botswana Accountancy College plagiarism guidelines policy.**



Agree **Signature**…………………………………….

**Date**………………………………………….

12th September 2025

**BEGINNING OF ASSIGNMENT**

**PART I: REQUIREMENTS ELICITATION**

We are creating a banking application that allows people to create accounts that belong to different branches, with the ability to deposit and withdraw from that account. Using a java application, a customer should be able to make an account through a branch assistant utilizing the app to make one, choosing it to be one of the three types of accounts it can be, customers will also be able to specify things such as an initial deposit in created accounts (which is required by some account types), customers will then be given the credentials to later log into the application themselves such as their account number, password and others, there customers can then log into the application and view the balances of their accounts as well as make deposits and withdrawals if allowed to.

Branches are responsible for editing, opening and closing accounts for customers through the application, using the search function built into the application to search against a database of existing branches and customers that the specific branch has. Branches are managed by themselves, the branch can choose to edit its own name, its branch unique identifier, as well as its address. They can also choose to delete a branch if it ever closes down.

An account has three types it can be, An Investment, Cheque or Savings account, each type of account has different capabilities, the type of account is specified when an account is created. For example, a savings account does not allow withdrawals, and has interest added to it after a certain period. Accounts are managed by both the customer it is tied to as well as the branch it was made at, customers can only do limited actions like withdrawal and depositing, while more permanent features are allowed to be done by the branches such as closing an account or withdrawing from an account that usually does not allow that, this is to ensure that a customer has to be given permission to make these actions by the branch according to the branches own guidelines.

For the saving of the data, I opted to use an open-source database engine called SQLite which I integrated into my application. I tried to use file systems at first, but this became very complicated, and I wanted the ease of a proper database system.

If I were to walk you through how a new customer would create an account and deposit funds. Firstly the customer visits our branch to open an account, the assistant would then open our application and login as a branch, after successfully doing so, they are then given the option to create an account, the customer would then give their details, these details will be used to make a customer object that is saved to the branch database, then an account is created, the customer specifies what type of account they’d like it to be and give us the money for an initial deposit, then this account is saved to the branch database and saved to the customer object’s list of accounts.

If the branch wanted to query later, they’d simply type the name of the customer or the account number of a specific account in the search function and find the information they are looking for.

**INTERVIEW RECORD**

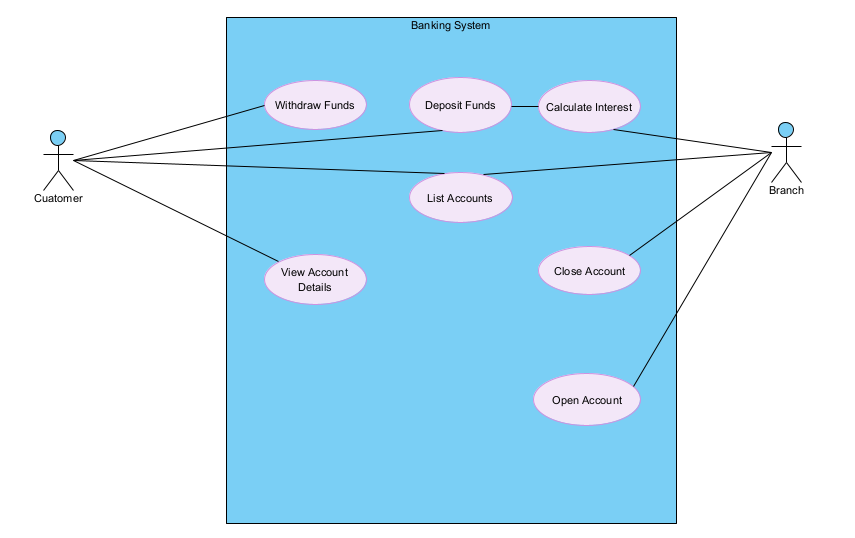
Interviewee: Mr. Themba Moeng

Interviewer: Kaelo Kgosidialwa

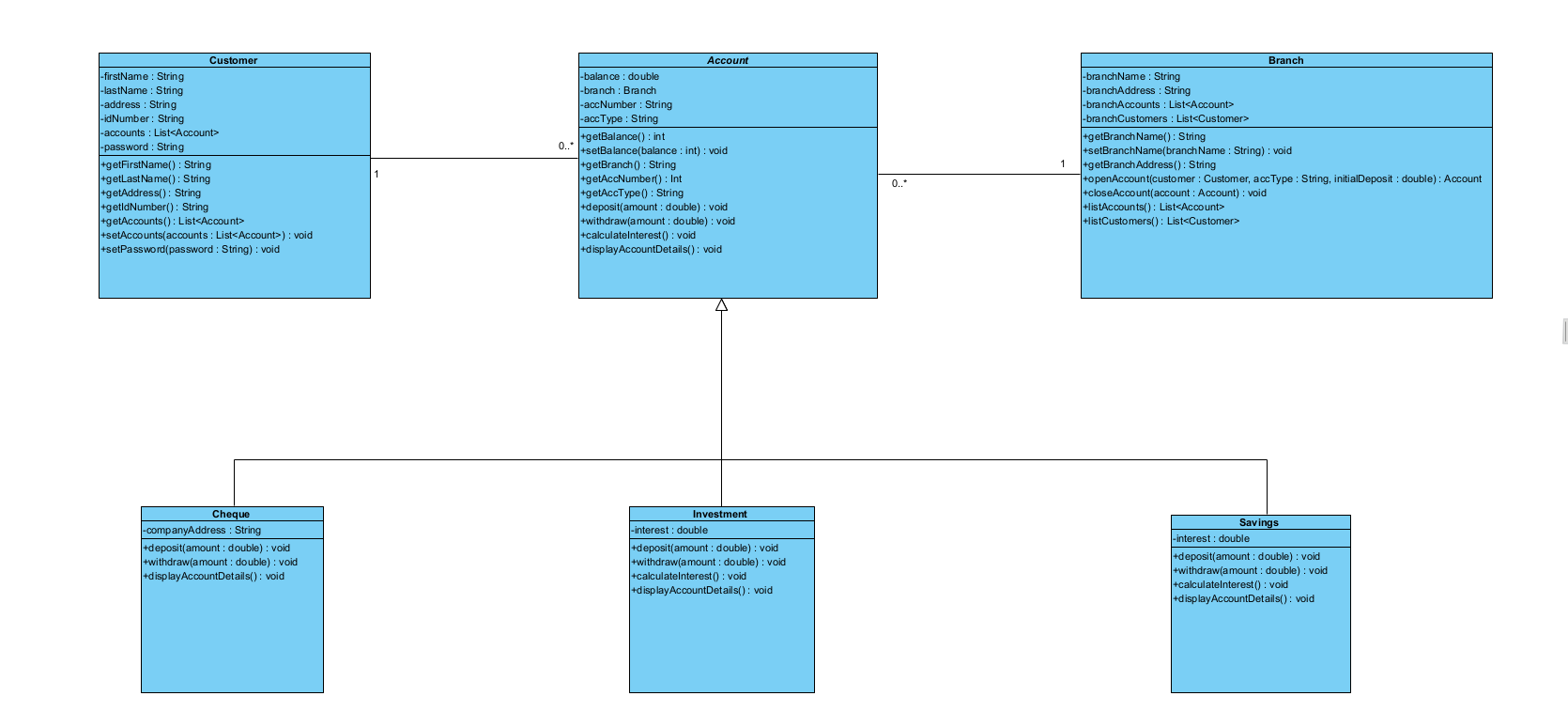
|  |  |
| --- | --- |
| **QUESTION** | **ANSWER** |
| 1. How do we show details about the transaction history from the customer’s side, their deposits and withdrawals and the branches side, the monthly payout of interest | You should actually show the transaction history in its own section in the customer menu, not just in the branch side, they must also be an option to select accounts with a certain account type and pay their calculated interest. |
| 2. Payouts (of interest) are made manually, not automatically? | Yes, the branch should have the control to pay accounts of a certain type depending on their interest rate of course. |
| 3. Should there be a feature to change account types after the account has been made | No it’s not really a requirement. We should really try as much as we can to minimize adding unnecessary features. |
| 4. Okay, but maybe we can add it by ourselves for our own testing | If you feel it is necessary to come up with a good final product by all means, then go for it. |

**PART II: STRUCTURAL UML MODELLING**

**USE CASE DIAGRAM**

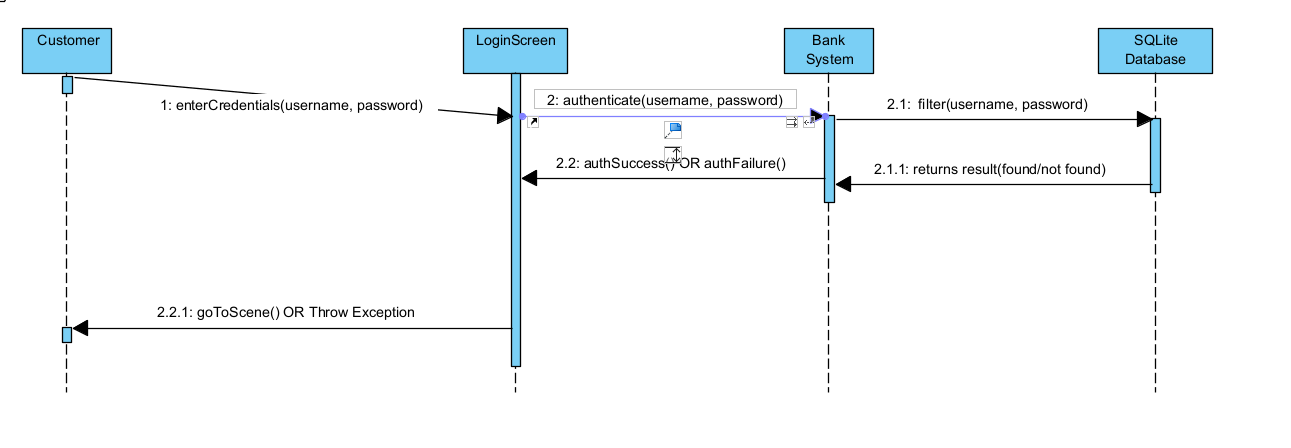


**CLASS DIAGRAM**

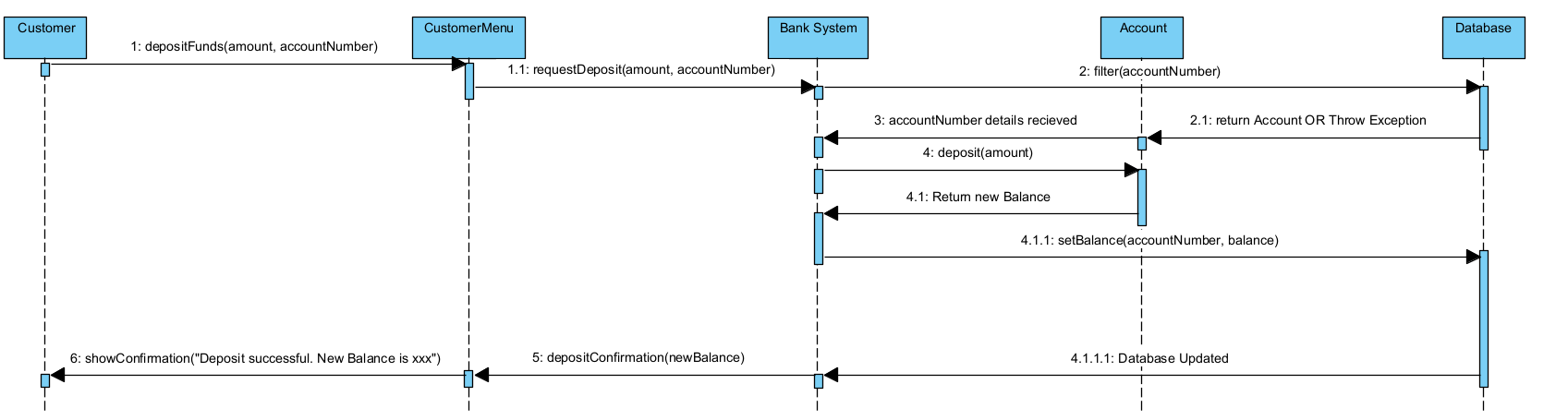


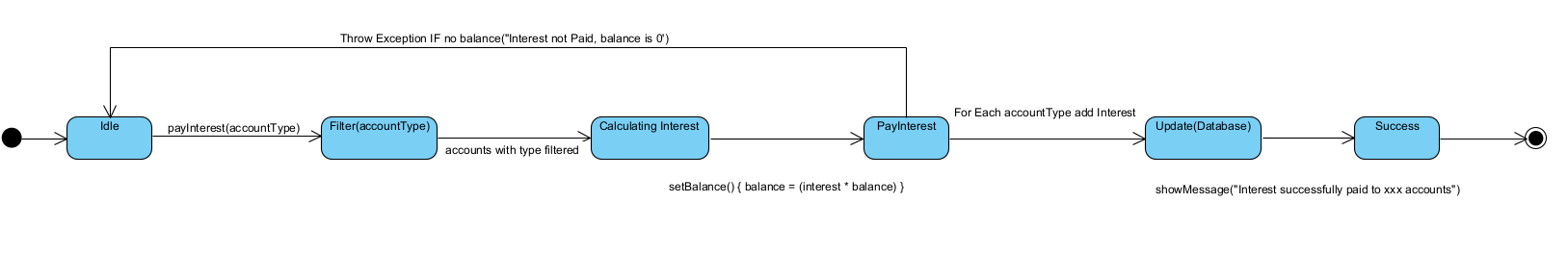
**PART III: BEHAVIOURAL UML MODELLING**

**SEQUENCE DIAGRAM SHOWING THE LOGIN USE CASE**

****

**SEQUENCE DIAGRAM SHOWING THE DEPOSIT FUNDS USE CASE**

****

**STATE DIAGRAM  
**