**REQUIREMENTS ELICITATION**

|  |  |
| --- | --- |
| A picture of a winding road and trees  OOAD-JAVA ASSIGNMENT | KAGO KELEFE    CSE24-114  **COMPUTER SYSTEMS ENGINEERING** |

FUNCTIONAL REQUIREMENTS

This describes what the system must do, the behavior, the functions and operations the system must perform. They are expressed from the user perspective. The core functions of a banking system are as follows:

Account creation – the system will allow the bank employee or bank teller to open new account of the following type savings , Investment or cheque account for the customers and enforcing specific deposit for each account and the rules for that account opened. Rules like no withdrawal in the savings account and investment allow withdrawal.

Processing transactions- the system shall facilitate the depositing and withdrawal of funds. The system must allow an authorized user to initiate a deposit into an account by providing an account number, also the system must validate the deposit amount is a positive value greater than zero. And the system will update the balance by the deposited amount. On withdrawals user can withdraw money from the specified account but the system will validate if the account type allows withdrawal. Upon successful withdrawal the system must decrease the amount balance by amount withdrawn.

Interest calculations- the system shall automatically apply monthly interest at the defined rates to customer balances. The interest affects the savings account and investment account but the cheque account does not have interest and the system must skip the interest calculation process for the type of account. The calculated amount will be deposited to ye account balance.

NON FUNCTIONAL REQUIREMENTS

Describes how the system should behave and what the constraints it will operate on or under are. They define ye quality attribute of a system rather than its specific behavior. They ensure the system usable, reliable and effective

Security -The user account should be locked temporary after three attempts to login to prevent attacks and also there should be a maximum of eight characters which will be a combination of alphanumeric characters .lastly in the user interface or customer should hide sensitive information like account number like just revealing last four digits instead of full account number.

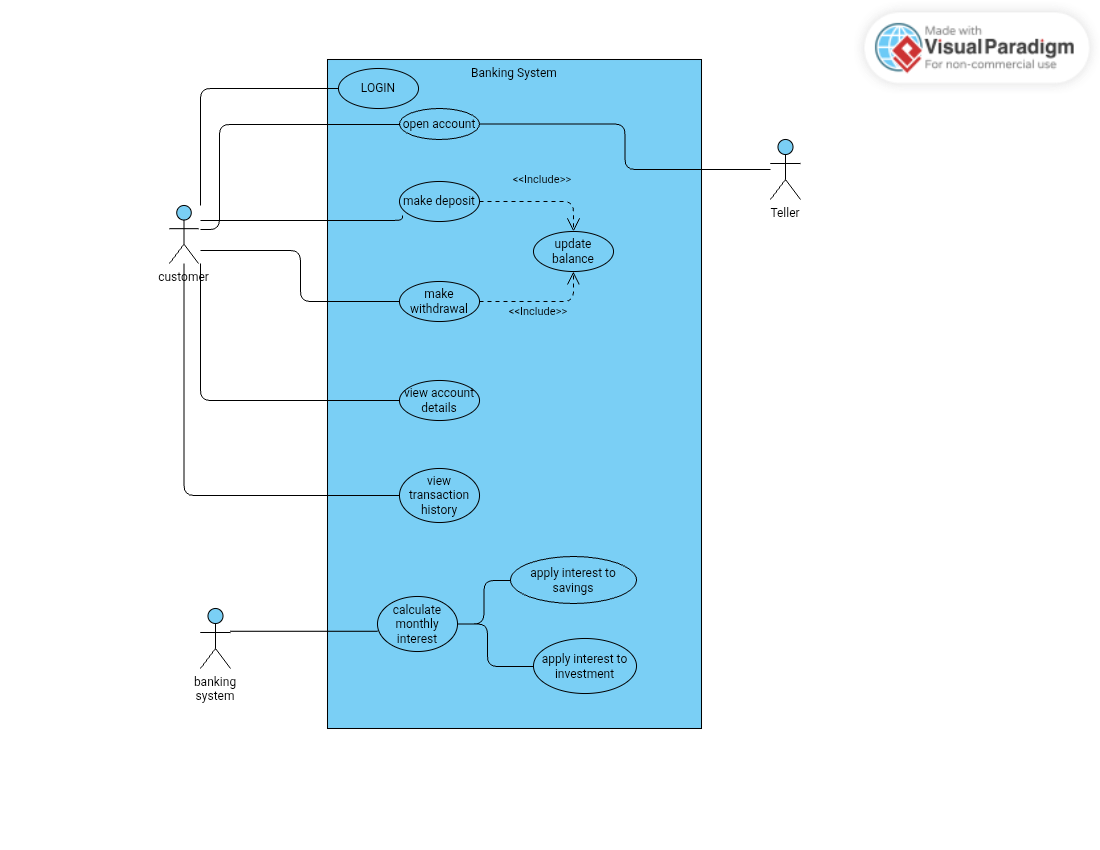
Performance – we should expect instantly response time of three milliseconds. This is important because there is more customers assisted and if the system is slow creates long wait times and queues and led to poor customer experience.

Usability and user experience – The primary users are the bank clerk or employees and also customers. The customers are people of varying degrees of computing ability so the system should be simple as possible. It should have clear labels and provide the necessary feedback which is understandable without confusing the customer .Also the system when performing withdrawal and deposit there should be a pop up message to say the withdrawal is successful or deposit is successful and they must be simple.

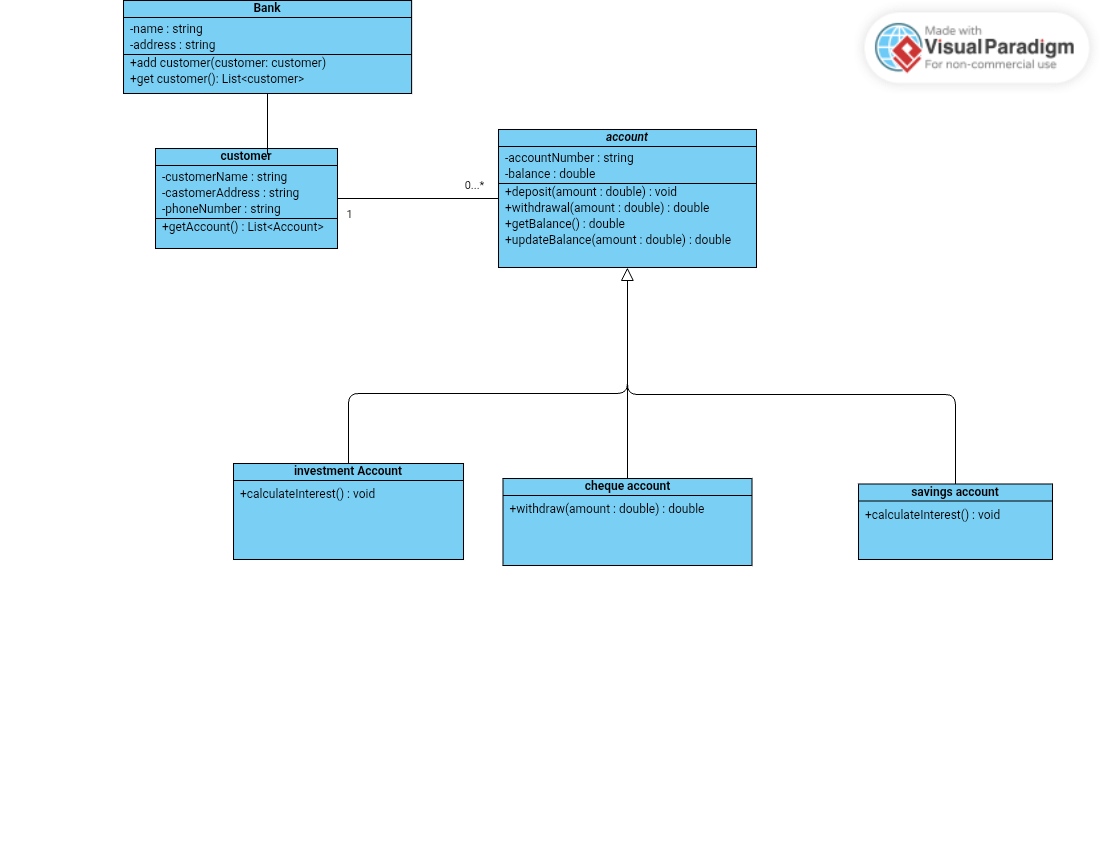
Reliability and availability – The system should be available 24 hours per day and with 9.9% uptime. Also if the transaction goes wrong halfway it should be reversed and the system must communicate the error and allow the user to retries.

**Structural UML Modelling**

Systems Use Case Diagram

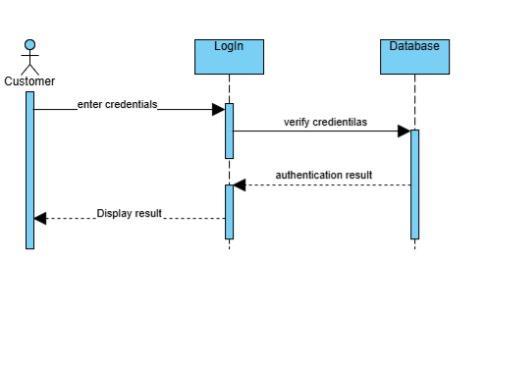


Class Diagram



**Behavioral UML Modelling**

Login Sequence Diagram



NAME OF INTERVIEWER: KAGO TLOTLO ANOLD KELEFE

NAME OF INTERVIWEE: THEMBA MOENG

DATE: 18 SEPTEMBER 2025

TIME: 09:34

1) How should users access this system? Should we use simple username and password authorization, or do we require a more secure two-factor authentication (2FA) method, such as using an authenticator app or an SMS text message?

**Answer;** the two-factor authentication would be desirable, but we will not be able to do it via the desktop application. It will be more doable if we had what?

Also the gateways communicating with what with the maybe e-mail also to send SMSs, so we’ll just say no. We’ll allow authentication for the usernames and passwords, and also having a password.

Policy that there should be a minimum of eight characters which will be a combination of alphanumeric characters and also the data within the database should be encrypted. So how you’ll find out the encryption algorithm applicable.

2) Should user accounts be temporarily locked after a certain number of failed login attempts (for example, 5 attempts) to prevent brute-force attacks? If so, for how long should the account be locked?”

**Answer;** you should basically have a maximum of three attempts. If you have the wrong password and username combination, the account can be locked.

3) How many users do you expect to be using the system simultaneously, particularly during peak hours? This will help us determine the load requirements for scaling the infrastructure.

**Answer;** I expect 50 or so users to be active on the system.

4) What is considered an acceptable response time for the system? For example, when a user clicks a button, should the result be displayed instantly or is a delay of 2-3 seconds acceptable?

**Answer;** we expect instantaneous response time of 3minseconds.

5) Who are the primary users, and what is their technical proficiency? Should the user interface be a simple, straightforward design, or does it need to support complex, long term risk analysis and data visualization?

**Answer;** so the primary users are the bank clerk or employees and also the customers. So obviously the bank customers are I think yeah bank customers are people of varying degrees of computing ability. So the system should be as simple as.

Possible with clear labels and also providing the necessary feedback which is understandable without confusing the custom. Also it should be simple enough that the person can use the system immediately.

Without any formal training.

6) How should users provide feedback on a successful action or transaction? Should the system use simple confirmation messages or is a more collaborative feedback system like a pop-up survey or a rating prompt required?

**Answer;** yes, basically like once you withdraw it could be a pop up message to say a withdrawal successful or deposit successful.

7) Is the application required to support multiple languages and regional settings (internationalization) or is English the only necessary language for the user base?

**Answer;** Just English

8) How should the system handle the unexpected termination of a process or transaction? If something goes wrong halfway through a critical operation, what mechanisms are in place to ensure data consistency and prevent corruption?

**Answer;**

9) What is the expected behavior if an operation fails? Should the system automatically roll back (undo) any partial changes to maintain a valid state, or is manual intervention required?

**Answer;** Reverse transaction