**Banking Application**

Group 3

Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 02/21/2025 | 1.0 | Initial Version | R.H |
| 02/25/2025 | 1.1 | Decided on Initial set of Modules | Group |
| 02/27/205 | 1.3 | Added Initial Requirements | Alexandra |
| 03/04/2025 | 1.5 | Fixed formatting; Added definitions; Added overview | Sophia |
| 03/04/2025 | 1.8 | Added UML Class Diagram | Matthew |
| 03/05/2025 | 2.0 | Added External Interface rq. ATM module rq. Payment module rq. And record keeping rq. | Ruba |
| 03/05/20215 | 2.1 | Added Performance Requirements | Ruba |
| 03/05/2025 | 2.5 | Added Use-Case UML | Phakin |
| 03/05/2025 | 2.6 | Added Sequence Diagram. | Phakin |
| 03/06/2025 | 2.8 | Updated Internal Interface Requirements, added Server Module Requirements, added security requirements | Sophia |
| 03/06/2025 | 3.0 | Added more Internal Interface requirements, Security and Privacy requirements and Environmental requirements. | Alexandra |
| 05/06/2025 | 3.5 | Deleted some requirements that were too much to implement. | Alexandra |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

**1.** **Purpose 4**

1.1. Scope 4

1.2. Definitions, Acronyms, Abbreviations 4

1.3. References 4

1.4. Overview 4

**2.** **Overall Description 5**

2.1. Product Perspective 5

2.2. Product Architecture 5

2.3. Product Functionality/Features 5

2.4. Constraints 5

2.5. Assumptions and Dependencies 5

**3.** **Specific Requirements 6**

3.1. Functional Requirements 6

3.2. External Interface Requirements 6

3.3. Internal Interface Requirements 7

**4.** **Non-Functional Requirements 8**

4.1. Security and Privacy Requirements 8

4.2. Environmental Requirements 8

4.3. Performance Requirements 8

# Purpose

This document outlines the requirements for the Banking Application

## Scope

This document will catalog the user, system, and hardware requirements for the Banking Application. It will not, however, document how these requirements will be implemented.

## Definitions, Acronyms, Abbreviations

**Client:** Person who benefits from the banking functions. Including depositing/withdrawing money, owning accounts of any amount.

**Teller:** The employee at a bank who has more permissions than a client. They can open/close accounts, edit accounts, deposit/withdraw money from a client’s account.

**ATM:** The interface where a client can perform transactions and see their account information.

**Deposit** – Adding funds to an account.

**Withdrawal** – Taking funds out of an account.

**Record** – Immutable logs of all transactions, actions performed by clients, and actions performed by Tellers

**Overdrawing –** Attempting to withdraw more funds than what is present in the account.

**Transaction** – An action that modifies the balance of an account.

**Fraud** – Race conditions; simultaneous access; out of order transactions.

## References

See Use Case Specification Document.

See UML Use Case Diagrams Document.

See Class Diagram Document

See Sequence Diagrams

## Overview

The Banking Application is designed to provide basic banking services across several different physical Bank locations. It operates on an autonomous network and supports simultaneous server access across all locations.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into 6 major modules: The Server Module, The ATM module, the Teller Interface module, the Account Module, the Payment Module, the Record Keeping module, and the Server Module.

## Product Functionality/Features

The high-level features of the system are as follows (see section 3 of this document for more detailed requirements that address these features):

## Constraints

Only Tellers can open, close, and edit accounts.

The application will only be accessible on devices available at the Bank locations.

Tellers have preexisting login credentials; A new user of type “Teller” cannot and will not be instantiated.

## Assumptions and Dependencies

It is assumed that Tellers will always input transactions correctly; there will never be the need to “reverse” a transaction due to accidental circumstances. So it is their responsibility to do so correctly.

It is assumed that Clients will always input ATM transactions correctly; there will never be the need to “reverse” a transaction due to accidental circumstances. So it is their responsibility to do so correctly.

It is assumed that the tellers are logging in from their work environment, device and location.

It is assumed that tellers will always log out after their work shifts.

It is assumed that there is more than one Bank location.

# Specific Requirements

## Functional Requirements

### Common Requirements:

### The system must prompt the user to log in as a Teller or a Client (ATM).

* + - 1. Users can access bank account information at any location of the same bank.
      2. Information is stored after the Client finishes each transaction and after tellers finish account transactions
      3. All banks operate on the same autonomous network.
      4. All information regarding transactions, user data, and teller data are stored on a centralized server.

### ATM Module Requirements:

* + - 1. The system should allow Clients to deposit or withdraw money, look at account balance and view transaction history.
      2. The system implements withdrawal and deposit limits on ATM transactions, also known as transactions, done from the Client interface rather than the teller interface.
      3. When Clients attempt to login the system checks to make sure the user has a profile in the database.
      4. The accounts that are shared between different Clients should allow both owners to make deposits and withdrawals from those accounts.
      5. Clients are only allowed to access their own accounts.
      6. Clients will be able to view their transaction history.
      7. ATMs are limited to handling deposits and withdrawals of up to $5000.

### Teller Module Requirements

* + - 1. Tellers can perform any function that an ATM can perform (but not vice versa).
      2. Tellers/Employees can access bank account information by using the client’s unique ID.
      3. Tellers can change client’s bank account information (i.e. Address, phone number, username, password, and name)

### Account Module Requirements:

* + - 1. Clients can have any number of saving and checking accounts
      2. Clients can share any number of banking accounts with other users
      3. Overdraft fees will be charged if the user attempts to withdraw more money than is available in their account.
      4. All accounts will have a balance.
      5. An account that is Closed will no longer be accessible by the Client.
      6. Accounts will be automatically closed by the Banking system if the balance reaches -$1000.
      7. Clients can choose to pay off their debt with Automated Recurring Payments.
      8. The client must specify which Checking account will be used to for Automated Recurring Payments.
      9. Only Checking Accounts can be used for Automated Recurring Payments.
      10. Credit Lines will only be given to Clients with a preexisting Checking account with a balance of at least $1000

### Payment Module Requirements:

* + - 1. The system should update the account balances after each payment.

### Record Keeping Module Requirements:

* + - 1. Logs all changes made to a client’s account by the teller. Denotes that the changes were made by the teller as well as the Employee ID of that teller.
      2. Logs all transactions completed by Client and denotes that a client made the transaction as well as the Bank ID of that Client.
      3. On all transactions the system will also log the time, date, type of transaction, the amount of transaction and on which account the transaction was made. Every transaction done on a credit line should also include the remaining available credit and the credit balance.
      4. In the case of Autopayments the transactions should be denoted as automatic and should denote both the credit account number and the associated account number.
      5. The system should log all instances of account closures and account openings. It should also log the creation of a new client profile and deletion of a client profile.

**3.1.7 Server Module Requirements**

3.1.7.1 The server must maintain persistent, immutable storage of all transaction records, account creation, and account deletion.

3.1.7.2 The server will store all information in plain text. There will be no encryption.

3.1.7.3 The server will be able to handle as many concurrent users as hardware will allow.

3.1.7.4 The server must authenticate users against the stored credentials database.

3.1.7.5 Connection timeouts will be implemented to manage inactive ATM sessions.

## External Interface Requirements

* + 1. The system will prompt the user to login by displaying two login buttons: One for the Client and another for the Teller.
    2. The system will prompt the Client to input their unique bank ID and password by displaying 2 typing spaces listed above each other. The top typing space will have the words “Bank ID” written above it and in between the two typing spaces and above the second typing space the words “Password” will be displayed.
    3. The system will prompt the Teller to input their employee ID and their password by displaying two typing spaces. The top typing space will have the words “Employee ID” written above it and in between the two typing spaces and above the second typing space the words “Password” will be displayed.
    4. The System will display an error message on the screen if the password doesn’t match the ID or vice versa.
    5. The System will display Two user profiles: Teller profile and Client profile. Both Profiles will display the Banks name at the top of the screen. In the left corner of the screen, the name of the user will be displayed and right under it will be the user’s ID. In the case of the Teller, the profile page will have two Buttons: “Search Client Profile” button and right below it a “Create New Client Profile” Button. In the Client’s case, all the client’s accounts will be displayed one on top of the other in order of which was made first.
    6. When the teller prompts the system by pressing “Search Client Profile” the system will display a typing space where the Teller can input the name of the Client or the Client’s Bank ID.
    7. When the teller prompts the system by pressing the “Create New Client Profile” button, they will be redirected to a screen displaying an automatically generated Bank ID, a typing space for the following: First Name, Last Name, Phone Number, Address, Password.
    8. Tellers should be able to see a list of matching names when they search for a name in the system. The tellers should be redirected to the client’s profile if they use the Clients Unique Bank ID. When on the client’s page the system should allow the Teller to see the client’s full information at the top of the screen (i.e. Full government name, address, phone number and Bank ID). The Teller should also be able to see the different account the client has on their Profile.
    9. The system will redirect clients to the appropriate account pages when they press the account graphics on the Clients Profile Screen. Each account screen will display the account number and the account balance. If the account is a credit line, it will display the available credit at the top and right below it the credit balance, at the bottom of the page it should say when the next billing cycle is. On a checking or savings account page the system should display the balance at the center of the screen and below it two buttons next to each other for “Deposits” and “Withdrawals”. The teller should also be able to see the same pages for each of the Client’s accounts after they found that client.
    10. In the case of accounts that are shared between clients, all clients sharing that account should be able to see the account on their profile page, with the words shared displayed next to the account number.
    11. If a client that is not in the banks Database attempts to login a message will be displayed that the Bank ID number does not exist.

## Internal Interface Requirements

3.3.1. TCP/IP will be the transport protocol for all client-server communication.

3.3.2. The system must establish a persistent connection between the client and server.

3.3.3. The system must handle multiple users at the same time without issues.

3.3.4. The server must process client requests and send appropriate responses.

3.3.5. The system must allow for future upgrades without disrupting existing

functionality.

3.3.6. Inactive sessions must be automatically closed after a certain period.

3.3.7. The system must store records of all internal transactions and operations for

tracking.

# Non-Functional Requirements

## Security and Privacy Requirements

* + 1. The system will not encrypt data being transmitted over the Internet
    2. Bank employees must login with credentials when accessing the Teller interface
    3. Clients must be able to login with credentials when accessing the ATM interface
    4. Username and Password validation must occur before any transaction processing.
    5. All financial transactions must be logged in the record keeping module to prevent fraud.
    6. Records must be retrievable by the Teller.
    7. Records must be immutable to prevent false data.

## Environmental Requirements

4.2.1. The application will be written in Java and will use Java sockets.

4.2.2. JDK 23 is required to run the application.

4.2.3. The system must use multi-threading to allow multiple users to perform

transactions simultaneously.

4.2.4. The system must support multiple bank locations.

4.2.5. The system must run on bank-approved devices.

## Performance Requirements

## System must render all UI pages in no more than 9 seconds for dynamic pages. Static pages (HTML-only) must be rendered in less than 3 seconds.

* + 1. High transaction processing speed with minimal downtime.
    2. Able to accommodate more than one user at a time and can handle a high volume of transactions accurately and efficiently.