# Group 3 Banking System

Software Requirements Specification

<https://github.com/syfabo/CS-401-Project>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 09/26/2025 | 1.0 | Completed 1.1, 1.4, 2.1, 2.2 and 2.3  Added Requirements to 3.1.1 | Sydney Boone |
| 09/27/2025 | 1.1 | Added 5 Constraints | Sydney Boone, Nidhi Prajapati, Gladwin Prabhu Rajkumar |
| 9/27/2025 | 1.2 | Added to 2.4, 2.5 and multiple sections in 3.1 | Sydney Boone |
| 9/29/2025 | 1.3 | Completed 1.3.3, 2.4, 2.5, 3.1.1 3.1.2, 3.1.5, 3.1.6, 3.2, and 3.3 Started 4.1 | Sydney Boone |
| 9/29/2025 | 1.4 | Completed 3.1.3, 3.1.4, and 1.3.4  Added to 3.1.5 | Nidhi Prajapati |
| 9/29/2025 | 1.6 | Completed 4.1, 4.2, 4.3, 1.3.1, and 1.3.2 | Gladwin Prabhu Rajkumar |
| 10/12/2025 | 1.7 | Created New Gannt Chart | Sydney Boone, Gladwin Prabhu Rajkumar |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose [4](#__RefHeading___Toc19440719)

1.1. Scope [4](#__RefHeading___Toc19440720)

1.2. Definitions, Acronyms, Abbreviations [4](#__RefHeading___Toc19440721)

1.3. References [4](#__RefHeading___Toc19440722)

1.4. Overview [4](#__RefHeading___Toc19440723)

2. Overall Description [5](#__RefHeading___Toc19440724)

2.1. Product Perspective [5](#__RefHeading___Toc19440725)

2.2. Product Architecture [5](#__RefHeading___Toc19440726)

2.3. Product Functionality/Features [5](#__RefHeading___Toc19440727)

2.4. Constraints [5](#__RefHeading___Toc19440728)

2.5. Assumptions and Dependencies [5](#__RefHeading___Toc19440729)

3. Specific Requirements [6](#__RefHeading___Toc19440730)

3.1. Functional Requirements [6](#__RefHeading___Toc19440731)

3.2. External Interface Requirements [6](#__RefHeading___Toc19440736)

3.3. Internal Interface Requirements [7](#__RefHeading___Toc19440737)

4. Non-Functional Requirements [8](#__RefHeading___Toc19440738)

4.1. Security and Privacy Requirements [8](#__RefHeading___Toc19440739)

4.2. Environmental Requirements [8](#__RefHeading___Toc19440740)

4.3. Performance Requirements [8](#__RefHeading___Toc19440741)

# Purpose

This document outlines the requirements for the Banking System.

## Scope

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

## Definitions, Acronyms, Abbreviations

No Definitions, Acronyms, or Abbreviations

## References

1.3.1 Use Case Specification – Pages 11-17 of this document

1.3.2 UML Use Case Diagrams – Page 18 of this document

1.3.3 Class Diagrams – Page 19 of this document

1.3.4 Sequence Diagrams – Pages 20 - 21 of this document

1.3.5 Gannt Chart – Page 22 of this document

## Overview

The Banking System is designed to provide banking services through an interface for authorized bank employees and a separate ATM application. Security and fraud prevention are ever-present concerns.

# Overall Description

## Product Perspective

The Banking System is an application that provides access to standard banking services for customers through both bank employees and ATM machines. The Banking System protects customers’ accounts from unauthorized access and minimizes fraud. The system is a java application that uses a client–server design over TCP/IP, and manages requests such as withdrawals, deposits, balance inquiries, and transfers. The system accommodates two types of users with different privileges and levels of access, authorized bank employees and ATM customers. The system also keeps a persistent log of user activity for record and fraud detection

## Product Architecture

The system will be organized into 5 major modules:

**(3.1.2)** The Authentication module

**(3.1.3)** The Teller module

**(3.1.4)** The ATM module

**(3.1.5)** The Logging module

**(3.1.6)** The Fraud Prevention 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)

* + 1. Authentication process for bank tellers and ATM customers
    2. Bank Teller application interface for authorized employees
    3. ATM application interface for self-serve customer usage
    4. Deposit and withdrawal of money, balance inquiries, and electronic transfers
    5. Logging of account activity and transaction history
    6. A server and client application that communicate over TCP/IP

## Constraints

2.4.1 The system must present a graphical interface

2.4.2 The system must be implemented in Java

2.4.3 The system must have a server and client application that operate using TCP/IP

2.4.4 The system must not have a web or HTML component

2.4.5 The system must not rely on any databases, libraries, or frameworks

2.4.6 The system must require users are authenticated before receiving access to any part

2.4.7 The system must be scalable to a very large number of users

## Assumptions and Dependencies

2.5.1 It is assumed that the system’s clock will be accurate and consistent to keep accurate records

2.5.2 It is assumed that all users of the system will be responsible with their credentials, logging in, and logging out as expected

2.5.3 It is assumed that the external card scanner reliably reads card numbers from customers’ physical cards

2.5.4 It is assumed that two accounts will never have the same account number

2.5.5 It is assumed that credit scores will automatically update accordingly

# Specific Requirements

## Functional Requirements

### Common Requirements:

3.1.1.1 SR9 The system has an interface for bank employees and another for ATM services

3.1.1.2 SR23 The system writes card numbers, account pins, balances, and transaction history to an external file

3.1.1.3 The system allows accounts to be opened or closed, and allows new people to be added to existing accounts

3.1.1.7 The system allows the creation of 3 types of accounts: checking, savings, and a line of credit

3.1.1.8 The system must generate a receipt/confirmation message after every transaction showing the transaction details (type, amount, date, and updated balance)

3.1.1.9 The system charges fees and provides interest appropriately

3.1.1.10 The system will give Bank employees access to change passwords, close accounts, or adjust credit limits.

### Authentication Module Requirements:

* + - 1. This module will store account usernames, passwords and pins, for verification
      2. This module will accept employees’ usernames and passwords and verify validity

3.1.2.3 This module will accept a pin associated with a customer’s profile and verify its validity

### Teller Module Requirements:

3.1.3.1 This module allows money to be deposited, withdrawn, or transferred between accounts

3.1.3.2 This module allows bank employees to manage customer accounts, including opening accounts, closing existing accounts, and adding new users to accounts.

3.1.3.3 This module will allow bank employees to change passwords or reset account credentials, such as passwords and account PINs.

3.1.3.4 This module will allow bank employees to change passwords or reset account PINs.

3.1.3.5 This module allows bank employees to set or adjust credit limits on line of credit accounts.

3.1.3.6 This module lets bank employees have access to view customer profiles, including balances and transaction history.

3.1.3.7 This module must display warnings when a transaction or account change exceeds configured thresholds.SR10

SR10

### ATM Module Requirements:

* + - 1. This module allows customers to perform self- service transactions, including deposits, withdrawals and transfers between their own accounts.
      2. This module allows customers to view account balances and transaction history for the past 30 days.
      3. This module must generate a receipt or confirmation message after every transaction showing transaction type, amount, date and updated balance.
      4. This module must log all login attempts, ATM transactions and account actions into the logging module.
      5. This module must terminate the session and log out the customer automatically after a period of inactivity.
      6. This module must display an error message and deny access after repeated failed login attempts.
      7. This module must prevent customers from making withdrawals if the account balance is insufficient.

### Logging Module Requirements:

3.1.5.1 This module must record all successful and failed logins.

3.1.5.2 This module must record all transaction performed through the Teller at the ATM modules, including deposits ,withdrawals, transfers and Line of Credit(LOC) borrowing/repayment.

3.1.5.3 This module should record all account management activities, including account creation , closure , passwords/PIN resets, changes of LOC.

3.1.5.4 This module should store logins in a secure and immutable format that prevents unauthorized modification or deletion.

3.1.5.5 This module must ensure all logs are timestamped, include the user ID or employee ID , and reference to affected account number.

3.1.5.6 This module must provide log data to the fraud prevention module for fraud analysis of suspicious activity.

### Fraud Prevention Module Requirements:

3.1.6.1 This module will make a note of suspicious activity as it is reported by the logging module

3.1.6.2 This module will flag a single profile being accessed at the same time from multiple programs as suspicious activity

3.1.6.3 The Fraud Prevention module will flag 3 or more consecutive failed login attempts as suspicious activity

3.1.6.4 The Fraud Prevention module will flag access from an ATM or Teller located more than 50 miles from the last transaction as suspicious activity

3.1.6.5 The Fraud Prevention module will flag transactions that exceed a set threshold amount as suspicious activity

3.1.6.6 The Fraud Prevention module will flag more than 3 transactions within a set time window as suspicious activity

## External Interface Requirements

3.2.1 The Authentication Module receives card information from an external card reader

3.2.2 The System will display a welcome screen where employees can enter credentials in the teller application

3.2.3 The system will display a screen where users can provide card information or username and enter a pin or password to verify identity in the ATM and teller applications

3.2.4 The system will display the customer profile, which will display all open accounts with account number and balance after validating credentials

3.2.5 The system will display a menu for each account on the profile with an option to make a transaction, open an account or close an account

3.2.6 The system will have an interface for transactions that accepts the amount and the location it will move to (deposit, withdraw, or transfer to another account)

3.2.7 The system will display a confirmation screen after any transaction/ account action

## Internal Interface Requirements

3.3.2 The Authentication Module communicates customer information to the Teller or ATM module to open the correct profile

3.3.3 The Teller and ATM module will report all transactions to the logging module for them to be saved

3.3.4 The logging module will communicate all information it receives with the Fraud Prevention module to monitor for suspicious activity

3.3.5 The Fraud Prevention module will send any suspicious log entries back to the logging module for them to be flagged

# Non-Functional Requirements

## Security and Privacy Requirements

4.1.1 The system must not grant access to any profiles or accounts without valid user authentication

4.1.2 The system must enforce role-based access so every user can only access relevant and appropriate customer data

4.1.3 The system must log all transactions and actions taken without leaking information to outside sources

## Environmental Requirements

Example:

4.2.1 SR20 System cannot require that any software other than a web browser be installed on user computers.

4.2.2 SR25 System must make use of the University’s existing Oracle 9i implementation for its database.

4.2.3 SR26 System must be deployed on existing Linux-based server infrastructure.

4.2.1 The system must run on Java SE with no external libraries, frameworks, or databases.

4.2.2 The system must not require a web browser

4.2.3 The system must work on any Mac or Windows operating system within the last 2 years with at least the latest secure Java SE

4.2.4 The system must be able to use both graphical and command line user interfaces.

## Performance Requirements

Example:

4.3.1 SR27 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.

4.3.1 The system must operate server and client requests securely through both regular and unexpected network delays.

4.3.2 The system must take no more than 30 seconds to load an accounts information

1.3.1 Use Case Specification

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | Customer Authentication |
| **Relevant Requirements** | 3.2.4 - The system will display the customer profile with account numbers and balances after validating credentials.  4.1.1 – The system must not grant access to any profiles or accounts without valid user authentication. |
| **Primary Actor** | Customer |
| **Pre-conditions** | * Customer has a valid bank card * ATM is connected to bank server |
| **Post-conditions** | * Customer profile is logged in * Customer can access their accounts through the ATM |
| **Basic Workflow** | 1. Customer inserts card with card number. 2. System prompts for PIN. 3. Customer enters PIN. 4. System validates PIN with stored credentials. 5. If valid, system grants access to ATM services. |
| **Alternative Workflow** | 1. Invalid PIN entered → system prompts for retry. 2. 3 consecutive invalid PINs → card locked |
| **Exceptions** | * ATM out of service. * Network error prevents validation. |
| **Related Use Cases** | Withdraw Cash, Deposit Funds, Balance Inquiry. |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Cash Withdrawal (ATM) |
| **Relevant Requirements** | 3.1.4.1 – ATM module allows customers to perform self-service transactions, including withdrawals.  3.1.4.7 – The system must prevent withdrawals if the account balance is insufficient. |
| **Primary Actor** | Customer |
| **Pre-conditions** | * Customer is authenticated * Account has sufficient balance (unless using LOC account). |
| **Post-conditions** | * Balance is updated. * Cash given to customer. * Transaction logged. |
| **Basic Workflow** | 1. Customer selects “Withdraw” option. 2. Customer enters amount. 3. System validates available funds. 4. System dispenses cash. 5. System updates account balance. 6. System prints receipt (if requested). |
| **Alternative Workflow** | 1. Invalid PIN entered → system prompts for retry. 2. 3 consecutive invalid PINs → card locked |
| **Exceptions** | * ATM out of service. * Network error prevents validation. |
| **Related Use Cases** | Withdraw Cash, Deposit Funds, Balance Inquiry. |

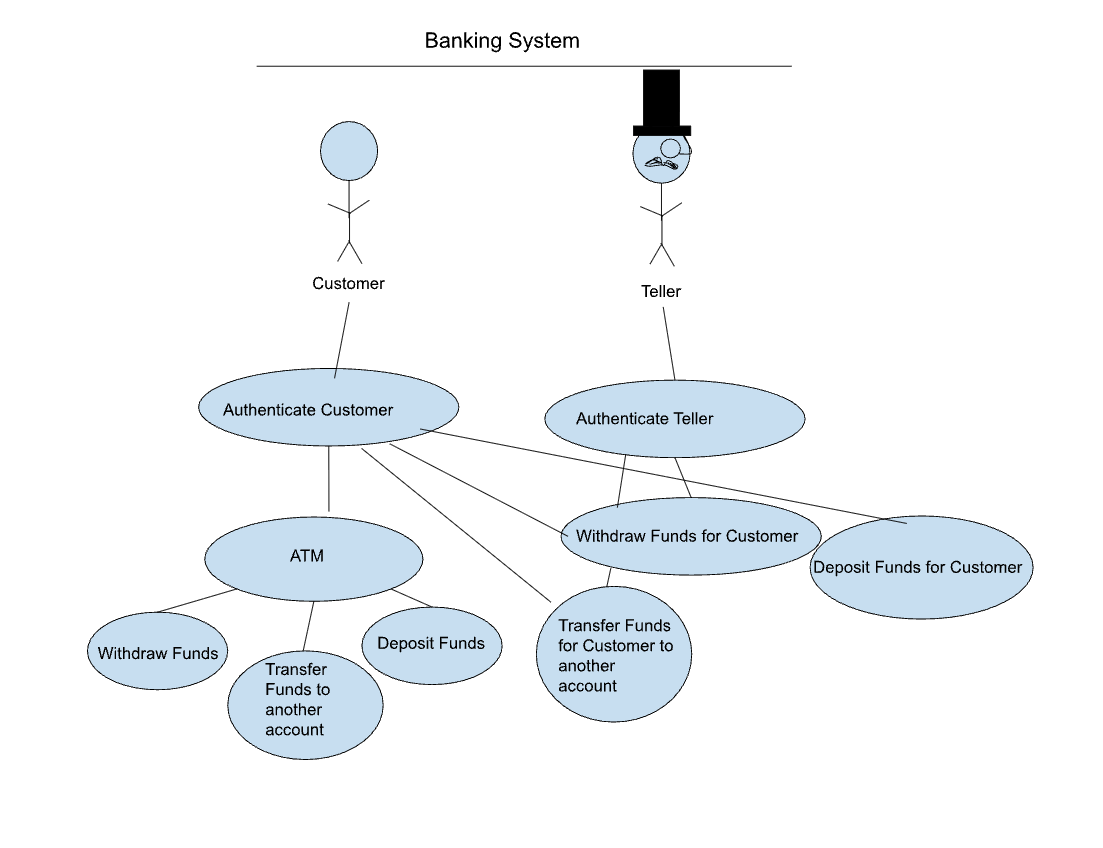
|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | Cash Deposit using ATM |
| **Relevant Requirements** | 3.2.6 – The system will have an interface for transactions that accepts the amount and the location it will move to. |
| **Primary Actor** | Customer |
| **Pre-conditions** | * Customer is authenticated. * ATM supports deposits. |
| **Post-conditions** | * Account balance is increased. * Transaction logged. |
| **Basic Workflow** | 1. Customer selects “Deposit” option. 2. Customer inserts cash/check. 3. System validates deposit (amount recognition). 4. System updates account balance. 5. System issues receipt. |
| **Alternative Workflow** | * Deposit rejected → system returns item and cancels transaction. |
| **Exceptions** | * ATM hardware error in accepting deposits. * Network/server error in balance update. |
| **Related Use Cases** | Withdraw Cash, Balance Inquiry. |

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | Employee Authentication (Teller) |
| **Relevant Requirements** | 4.1.1 – The system must not grant access to any profiles or accounts without valid user authentication. |
| **Primary Actor** | Bank Teller |
| **Pre-conditions** | * Teller has valid credentials issued by the bank. |
| **Post-conditions** | * Teller session is established. * Teller can access customer service functions. |
| **Basic Workflow** | 1. Teller enters username and password. 2. System validates credentials. 3. Teller is granted access to the teller GUI. |
| **Alternative Workflow** | * Invalid credentials entered → system prompts for retry. * 3 failed attempts → account locked. |
| **Exceptions** | * Log in error due to connectivity issues |
| **Related Use Cases** | Withdraw Cash, Deposit Funds, Balance Inquiry. |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | Transfer Funds to Another Account  (ATM) |
| **Relevant Requirements** | 3.1.4.1 – ATM module allows customers to perform self-service transactions, including transfers.  3.2.6 – Transaction interface accepts the amount and destination account.  4.1.3 – The system must log all transactions and actions taken. |
| **Primary Actor** | Customer |
| **Pre-conditions** | * Customer is authenticated with card and PIN. * Initial account has sufficient balance. * Destination account is valid. |
| **Post-conditions** | * Source account is debited. * Destination account is credited. * Transaction receipt is printed or displayed. |
| **Basic Workflow** | 1. Customer inserts Card and types in PIN 2. Customer selects “Transfer Funds” into ATM 3. ATM Prompts for Transfer Details including initial account, destination account, and amount 4. Customer enters details 5. System checks details 6. System withdraws from account 7. Transfer is logged and receipt is printed |
| **Alternative Workflow** | * Invalid PIN → new attempt made → 3 failures → card locked. * Insufficient funds → ATM cancels transfer and notifies customer. * Destination account invalid → ATM cancels request |
| **Exceptions** | * Networking issue * ATM has hardware glitch |
| **Related Use Cases** | Transfer Funds to Another Account (Teller) |

|  |  |
| --- | --- |
| **Use Case ID** | 6 |
| **Use Case Name** | Transfer Funds to Another Account  (Teller) |
| **Relevant Requirements** | 3.1.2.2 – The authentication module will accept employees’ usernames and passwords and verify validity.  4.1.1 – The system must not grant access to any profiles or accounts without valid user authentication.  4.1.3 – The system must log all transactions and actions taken. |
| **Primary Actor** | Teller |
| **Pre-conditions** | * Customer is authenticated with card and PIN. * Initial account has sufficient balance. * Destination account is valid. |
| **Post-conditions** | * Source account is debited. * Destination account is credited. * Transaction receipt is printed or displayed. |
| **Basic Workflow** | 1. Teller logs in to system 2. Customer requests transfer via system 3. Teller selects “transfer funds” 4. Teller enters transfer information like account with funds, transfer account, and dollar amount 5. System checks info 6. System transfers money 7. System gives confirmation to teller 8. Teller gives receipt to customer |
| **Alternative Workflow** | * Insufficient funds → system cancels transfer and the teller notifies the customer * Destination account not found → system cancels transfer and teller notifies customer. |
| **Exceptions** | * Network failure → Customer is notified * Teller system fails midway → System resends money back so it funds are not lost |
| **Related Use Cases** | Transfer Funds to Another Account (ATM) |

1.3.2 Use Case Diagrams



1.3.3 UML Class Diagram

A diagram of a company

AI-generated content may be incorrect.

1.3.4 Sequence Diagrams

**ATM and authenticator**

A diagram of a bank card

AI-generated content may be incorrect.

**Teller and authenticator**

A screenshot of a computer

AI-generated content may be incorrect.

1.3.4 Sequence Diagrams Cont.

**Logging Module Diagram**A diagram of a login

AI-generated content may be incorrect.

**Fraud Prevention**

A diagram of a computer program

AI-generated content may be incorrect.

Gannt Chart

