# Group 3 Banking System

Software Requirements Specification

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 09/27/2025 | 1.0 | Completed 1.1, 1.4, 2.1, and 2.3  Transferred 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 | Finished 2.4  Added to 2.5 and 3.1 | Sydney Boone |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Use Case Specification Document – Step 2 in assignment description

UML Use Case Diagrams Document – Step 3 in assignment description

Class Diagrams – Step 5 in assignment description

Sequence Diagrams – Step 6 in assignment description

## 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 provide 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 be authenticated before receiving access to any part

## 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 all employees will

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

# 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 stores card numbers, account pins, balances, and transaction history

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:

3.1.2.1 This module will store account usernames and passwords for verification

3.1.2.2 This module will present a login screen allowing users to enter a pin

3.1.2.3 This module ensures that accounts may only be accessed by one program at a time

### Teller Module Requirements:

3.1.3.1 This module allows money to be deposited, withdrawn, or transferred between accounts SR10

### ATM Module Requirements:

3.1.4.1

### Logging Module Requirements:

3.1.5.1

### Fraud Prevention Module Requirements:

3.1.6.1

## External Interface Requirements

Provide module specific requirements as appropriate. SR10

Example:

3.2.1 SR9 SR1 The system must provide an interface to the University billing system administered by the Bursar’s office so that students can be automatically billed for the courses in which they have enrolled. The interface is to be in a comma-separated text file containing the following fields: student id, course id, term id, action. Where “action” is whether the student has added or dropped the course. The file will be exported nightly and will contain new transactions only.

## Internal Interface Requirements

Provide module specific requirements as appropriate. SR10

Example:

3.3.1 SR17 The system must process a data-feed from the grading system such that student grades are stored along with the historical student course enrolments. Data feed will be in the form of a comma-separated interface file that is exported from the grading system nightly.

3.3.2 SR24 The system must process a data-feed from the University billing system that contains new student records. The feed will be in the form of a comma-separated text file and will be exported from the billing system nightly with new student records. The fields included in the file are student name, student id, and student pin number.

# 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

## 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.

## 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.