# Banking System

Software Requirements Specification

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/27/2025 | 1.00 | Initial Version | Harven Dhanota |
| 03/01/2025 | 1.01 | Revision: Added Assumptions + Common, Teller, and ATM Module Requirements | Jaishnoor Kaur |
| 03/03/2025 | 1.02 | Revision: Added Network Module Requirements | Harven Dhanota |
| 03/03/2025 | 1.03 | Revision: Added Reqs | Jaishnoor Kaur |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

1. ATM: automated teller machine
2. Teller: an employee that works for the bank, at the front desk
3. Bank: the organization that uses this program
4. Customer: an individual that has an account with the bank and can perform transactions

## 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 assist customers and tellers with performing transactions. The automated teller machine (ATM) can assist customers with basic transactions, whereas the teller can assist customers with more advanced transactions and modifying other account information.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into three major modules: the ATM module, the Teller module, and the Central module.

Note: System architecture should follow standard OO design practices.

## 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

2.4.1.

List appropriate constraints.

Constraint example: SR7 Since users may use any web browser to access the system, no browser-specific code is to be used in the system.

## Assumptions and Dependencies

2.5.1. It is assumed that a teller has existing credentials to log in to the system using the teller module.

2.5.2. It is assumed that the teller has verified the identity of the customer before performing any actions on their account.

2.5.3. It is assumed that any account action performed by a teller is done with the proper authorization.

# Specific Requirements

## Functional Requirements

### Common Requirements:

3.1.1.1. The bank system supports two types of accounts: checking accounts and savings accounts.

3.1.1.2. An account can only be accessed from one interface at any given time.

3.1.1.3. All IDs are unique. For customers, this means preventing duplicate user details i.e., for a pre-existing user, link any new accounts to their pre-existing credentials

3.1.1.4 Any changes made in any module are synchronously reflected in the Central module which stores data.

3.1.1.5 Each account has a status: open or closed.

### ATM Module Requirements:

3.1.2.1. A customer can log in to their account through the ATM interface using their credentials. All actions can only be performed by a logged-in customer.

3.1.2.2. A customer can view all their accounts (including account balances) and transaction history using an ATM.

3.1.2.3. A customer can withdraw money from their account, up to an amount of $1,000.

3.1.2.4. A customer can deposit money into their customer’s account, up to an amount of $2,000. If their current account balance is less than $2,000, then they can only withdraw up to their current account balance.

3.1.2.5.

### Teller Module Requirements:

3.1.3.1. A teller can log in to the teller module using their bank-provided credentials.

3.1.3.2. The teller is able to withdraw money from a customer’s account, up to their current account balance.

3.1.3.3. The teller is able to deposit money into a customer’s account, with no limit on the amount getting deposited.

3.1.3.4. A teller can create a new account of any type and delete an account for a customer.

3.1.3.5 For each of the above actions, account creation history (for 3.1.3.4) or transaction history (for 3.1.3.2, 3.1.3.3) are updated in Central module.

3.1.3.6 The history can be viewed by the Teller but cannot be modified or deleted.

3.1.3.7

### Central Module Requirements:

3.1.4.1. The central server can store account information, including account status, creation/closure dates, customer log-in credentials, the type(s) of accounts the customer has, their current account balance(s), and a list of all people authorized to view the account(s).

3.1.4.2. The central server stores each customer’s transaction history, alongside a date/time stamp of when that transaction occurred. The name of the person that performed the transaction is also recorded, for customers that have joint accounts.

3.1.4.3. The network is able to handle connections between the central server and the client interface (whether that interface be through an ATM or through a Teller).

3.1.4.4.

## External Interface Requirements

3.2.1. The Teller and ATM Module have a GUI.

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

* + 1. The modules communicate with each other via an automatic network using TCP/IP.

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. A customer cannot view or modify information or accounts of any other customer.

Example:

4.1.1 The SR8 System must encrypt data being transmitted over the Internet.

## Environmental Requirements

4.2.1.

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

4.3.1.

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.