**Banking System**

Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 06/21/2022 | 1.0 | Initial Version | Group 3 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 System. It will explain the purpose and features of the system, what the system will do, and the interface of the system. This document is intended for the client of the 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

Account Teller - Bank Staff that provides information to the user. Also able to open and close accounts for the user

User - The person who needs the system to do its task. An account holder or bank visitor.

Database - All the information monitored and collected by the banking system

## 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 will provide an interface for authorized banking employees in order to assist customers. There will also be an ATM system designed to perform most standard ATM functions.

# Overall Description

## Product Perspective

The Banking System is a Java Application with a GUI that operates with a server and client side application. The system interfaced with two other systems: a bank teller system and a user system. The system provides a secure environment for all financial transactions and for storing and retrieving user data

## Product Architecture

The system will be organized into two major modules: the bank employee module, the ATM 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):

Account holders will be able to view transaction history, deposit and withdraw.

Employees, bank tellers, will be able to do what account holders can do on their behalf as well as administrative actions such as open/close accounts and add account holders to joint accounts.

Bank information such as account credentials and account information (balance, transaction history, etc.) will be stored on a text file that will act as the “database” for the bank.

## Constraints

List appropriate constraints.

Constraint example: 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

List appropriate assumptions

Assumption Example: It is assumed that the maximum number of users at a given time is 15,000.

# Specific Requirements

## Functional Requirements

### Common Requirements:

3.1.1.1 Ids will be unique, no two accounts can have the same id, case sensitive alphanumeric strings 8 characters in length.

3.1.1.2 Passwords will be case sensitive alphanumeric strings between 6 and 20 characters in length.

### Employee Module Requirements:

3.1.2.1 Employees will log into their bank teller application using their own personal account id and password.

3.1.2.2 Employees should be able to use the customer’s id and pin to view their account.

3.1.2.3 Employees should be able to deposit and withdraw, without over withdrawing, from the user’s account on behalf of the account holder.

3.1.2.4 Employees will be able to add other bank members to a joint account.

3.1.2.5 Employees should be able to open, close, and freeze customer accounts.

3.1.3.6 Employees should be able to transfer money between customer accounts

### ATM Module Requirements:

3.1.3.1 Users should be allowed to log in using their issued id and pin.

3.1.3.2 Users should be able to deposit money into their account.

3.1.3.3 Users should be able to withdraw money from their account while not being able to over withdraw from their account.

3.1.3.4 Users should be able to transfer money between their accounts.

## External Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.2.1 The system must provide an interface to the banking system administered by the bank teller so that customers can check their account balance, open/close/freeze accounts, add/remove customers to/from a joint account, make deposits and withdrawals, and transfer money between accounts. The interface is to be in a comma-separated text file containing the following fields: customer id, customer account, action. Where “action” is what action the customer has taken. The file will be exported nightly and will contain all transactions.

3.2.2 The system must provide an interface to the banking system administered by an ATM so that customers can check their account balance, make deposits and withdrawals, and transfer money between accounts. The interface is to be in a comma-separated text file containing the following fields: customer id, customer account, action. Where “action” is what action the customer has taken. The file will be exported nightly and will contain all transactions.

## Internal Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.3.1 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 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 Account holder information such as id, pin, and account transactions should be

encrypted.

## Environmental Requirements

Example:

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

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

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

## Performance Requirements

Example:

4.3.1 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.