**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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Mine Pump Control System (MPC).

## Scope

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

## Definitions, Acronyms, Abbreviations

List any acronyms, terms etc. that need to be defined.

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.

## 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 Mine Pump Control System (MPC), is designed to monitor and pump flood water out of mine shafts. As underground mining operations take place far below the water table, flooding into mine galleries and shafts is an ever-present danger.

# Overall Description

## Product Perspective

## Product Architecture

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

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.

Only the teller can open, close and edit accounts.

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

## Assumptions and Dependencies

List appropriate assumptions

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

It is assumed that the teller inputs every transaction correctly.

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

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

It is assumed that there’s more than one location.

It is assumed that tellers are given their login information from their employers.

# 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.

### User Interface Module Requirements:

### ATM Module Requirements:

* + - 1. The System must log out of account after a period of inactivity.

### Teller Module Requirements:

### Account Module Requirements:

### Payment Module Requirements:

### Record Keeping Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.

## External Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.2.1 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.

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

Example:

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

## 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.