Software Requirements Specification

**Group 6**

**Daniel, Davinder, John, Juan, Skone**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| mm/dd/yyyy | 1.0 | Initial Version | Your Name |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 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

1.2.1 Business Day: A business Day is between 9:00AM - 5:00PM, excluding weekends and federal holidays.

## 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 3 major modules: the ATM module, the Database/Server module, and the Internal System module.

## Product Functionality/Features

The high-level features of the system are as follows: The database will communicate with two types of interfaces, an ATM for customer use and an internal system that is for teller use. The system will recognize two basic user types: Customer and employee, which can then have more unique children like teller or supervisor. For the customer user, the system will have several account types available such as checking, savings, or business. A customer may open as many accounts as they please. A customer will access the bank through either a teller or ATM. The teller will confirm identity by asking security questions. The ATM will confirm identity by using a debit card and pin. An ATM will have the following functions: Deposit, withdraw, and transfer money, check the available balance, and print account information. A teller will have the same functions however they may also open and close accounts. A higher-level employee account will be able to open and close employee accounts. The database will be able to sync up account information from across various banks to make sure all information is correct everywhere. The internal system for employees will ask for confirmation with a username and password.

## Constraints

Since users cannot access the bank online, all banking options must be available either by ATM or teller

Since the system syncs up all banks in the system together, a priority queue like structure must be used for transactions

The system must be written in Java

Since tellers can also be customers, another must be used to change a tellers account to avoid fraud

## Assumptions and Dependencies

A customer should never have to directly interface with the database

Only registered account holders will know an account’s pin and have the correct debit card

# Specific Requirements

## Functional Requirements

### Common Requirements:

Provide requirements that apply to all components as appropriate. SR10

Example:

3.1.1.1 SR9 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.

3.1.1.2 SR23 The system should provide HTML-based help pages on each screen that describe the purpose of each function within the system.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate. SR10

Example:

3.1.2.1 SR9 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.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate. SR10

Example:

3.1.2.1 SR9 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.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate. SR10

Example:

3.1.2.1 SR9 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. 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 portal designated to bank employees must remain separate and inaccessible to bank customers.

## Environmental Requirements

4.2.1 The system must be written in Java

4.2.2 The system must be compatible with current ATM models present at branch locations, as well as those in designated businesses.

## Performance Requirements

4.3.1 Upon the completion of a deposit or transfer (Direct Deposit, Cash Deposit, Transfer Between Accounts), funds must be available within the next business day.

4.3.2 Upon the completion of a check deposit, availability of those funds may be delayed by one more business day(s), depending on the amount being deposited.

4.3.3 The System must be completed and ready for operation by May 4, 2022