# Group 3 Banking Application

Software Requirements Specification

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 9/22/2023 | 1.0 | Initial Version | All |
| 9/22/2023 | 1.1 | Brainstorm terms/abbreviations | Daniel Leal |
| 9/25/2023 | 1.2 | Started on system requirements | Daniel Leal |
| 10/2/2023 | 1.3 | Added requirements to modules | All |
| 10/3/2023 | 1.4 | Edited assumptions and constraints | All |
| 10/3/2023 | 1.5 | Transferred Document from Google docs to MS Word | David Kromka |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

**Acronyms/Abbreviations:**

* + 1. **ATM:** System that lets users do transactions with their bank account such as deposit, withdraw and see balance in their accounts.
    2. **GUI:** Graphical User Interface
    3. **Teller**: Bank worker who is able to open/freeze and modify user accounts based on their needs. \
    4. **User**: Bank customer who has an account or multiple accounts and is able to do transactions.
    5. **Deposit**: When a user puts money in a bank account.
    6. **Withdraw**: When a user takes out money from their account.
    7. **Freeze Accoun**t: when a bank account is put on “hold” in order to prevent any unwanted transaction from taking place.
    8. **Delete Account:** When an account is terminated and unlinked from a user.
    9. **Minimum balance**: MInimum balance an account has to have. Users may not be able to withdraw any money if minimum balance is reached.
    10. **Streamline system**: a system that avoids doing repetitive actions in order to have better performance and be more efficient when it comes to doing different processes or tasks.
    11. **Bank Statement**:  A list of information regarding users transactions, such as fees, deposits, withdrawals and  date of such transactions. As well as available balance in the user account.

## 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 software provides tellers with the ability to access a member’s account and provide banking services for them. This will include ATM software that allows members to access banking services without a teller. The software will have many essential features to handle modern-day banking needs. We will provide both members and tellers with a streamlined system for managing funds and accounts.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into 5 major modules: Server Module,  GUI module (User and Teller), User Account module, Bank Account module, Teller 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

* + 1. Since the GUIs must be able to run on any commonly used hardware, it must be implemented in pure Java.

## Assumptions and Dependencies

* + 1. **Assumptions:**
       1. It is assumed that the server’s Internet connection will never be faulty or terminate.
       2. It is assumed that the system will only operate within the United States and therefore only needs to support US dollars/cents as a unit of currency.
    2. **Dependencies:**
       1. The system will be dependent on the TCP/IP protocol for networking communications between the server and its various clients.

# Specific Requirements

## Functional Requirements

### Common Requirements:

* + - 1. System must be able to support storing the information of virtually unlimited tellers and end users.
      2. System must use the TCP/IP communication protocol for its networking.

### Server Module Requirements:

* + - 1. The server must have levels of authentication to distinguish between end users and bank tellers.
      2. The server must be able to hold a large number of unique bank users and associate them with their account(s).
      3. The server must store logs of the date, time, and associated actors when the following operations occur: a user logs in, a teller logs in, money is deposited or withdrawn, bank accounts/users are created/deleted, a user account is added/removed from a bank account.
      4. The server must be able to support a large number of users and tellers using the system interface at once.

### GUI Module Requirements:

* + - 1. ATM GUI must have a button for deposit, withdrawal, check balance, and an option to contact a teller.
      2. Contacting the teller at the ATM must allow users to request to transfer funds, freeze an account, create an account, or delete an account.
      3. Teller-side GUI must have a button for deposit, withdrawal, check balance, add account, freeze account, delete account, and transfer funds to another account. User Account Module Requirements:

### User Account Moule Requirements:

* + - 1. A user must be able to be associated to any number of bank accounts, including none.
      2. Users must have a unique user ID (username), and a password associated with this username.
      3. Users must only be able to interact with their account after having verified their username and password.
      4. Users must not have access to other users' accounts. Only the ones that are linked to their profile may be accessed unless using a joint account.
      5. Users must be able to deposit, withdraw, check balance, and transfer money.

### Bank Account Module Requirements:

* + - 1. Users must be prevented from withdrawing money if the withdrawal would leave their balance below the minimum allowed balance.

### Teller Module Requirements:

* + - 1. Teller must be able to create, freeze and delete user accounts.
      2. Every time a teller deposits, withdraws, checks balance, adds account, freezes account, deletes account and transfers funds between accounts, the action must be logged.
      3. A Teller must be able to create joint accounts by adding other verified users to existing accounts.
      4. There must be a system allowing tellers to receive requests from end users to perform account manipulation operations over the network.

## External Interface Requirements

* + 1. The bank system must have two different interfaces. One interface must be focused on user interaction with their bank account(s). This interface must let users deposit, withdraw money, and see their bank statement history.  The second interface must be focused on teller specifications. The system for the teller must be able to open, freeze and delete user accounts as well as do transactions just like a normal user account would.
    2. The bank system should have an initial prompt for users to put in their username and password in order to access their account. Once signed in, users can do different types of transactions as long as they don’t sign off or hit a system warning that avoids a certain task.
    3. The bank system should be able to verify users by verifying their User ID and pin number.

## Internal Interface Requirements

* + 1. Bank system must have a validation process in order to let users and tellers interact with an account. Based on their unique account number and pin.
    2. The Bank system should be able to keep track of the number of bank accounts in the system. The status of these accounts should also be able to be modified if any of the bank accounts changes its status (active/freeze/deleted).
    3. The System should be able to differentiate between a teller account and a customer account to place specific restrictions and/or give access to specific functions for each one.
    4. The System must separate the data from every user. Data for every specific user must not be accessible from other accounts. Only a teller's accounts should be able to access a user account.

# Non-Functional Requirements

## Security and Privacy Requirements

* + 1. Highly sensitive data like usernames and passwords should not be able to be accessed externally from the system.

## Environmental Requirements

* + 1. The bank system should be able to run at any time of the day.

## Performance Requirements

* + 1. System should have enough memory to store an unlimited amount of users accounts, tellers accounts, and user information
    2. The system should be able to verify the user's ID and password in a timely manner in order for the user to experience as little lag as possible while using the bank system.
    3. The system should be able to perform actions on any account in a timely manner to ensure that they experience minimal delay and risk.