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**

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 banking 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 ID: 00190

Use Case Name: Managerial

Relevant Requirements: Bank ledger

Primary Actor: Manager

Pre-conditions:

* Correct branch of bank
* Log in with username and password
* Required level of authorization

Post-conditions:

* Confirm withdrawal of $10,000 or more
* Add note on reason of said withdrawal
* Review daily ledgers
* Can do everything an officer can

Basic Flow or Main Scenario:

1. Log into a workstation
2. Review ledgers
3. Add notes as needed
4. Save
5. Log out of workstation

Extensions or Alternate Flows:

* As an extension, managers are able to perform the same flow of tasks that an officer is able to
* Approval of withdraw of $10,000 or more
  1. A teller request a manager for approval
  2. A prompt is presented for manager to log in
  3. Log in is validated
  4. Manager add note as to reason for withdrawal
  5. Manager approve the withdraw
  6. Manager account is log out once the task has been “confirmed” (confirm prompt) as completed

Exceptions:

* Aside from $10,000+ approval and adding reason notes, cannot make changes to account info or balance, if it is currently being access by another workstation.
* Incorrect username and/or password
* Attempt log in at incorrect bank

Related Use Cases: 0020 (New Client Assistance), 0021 (Client Assistance)

Use Case ID: 00200

Use Case Name: New Client Assistance

Relevant Requirements: Client portion of the bank ledger

Primary Actor: Officer

Pre-conditions:

* Correct branch of bank
* New client ready to open a new account
* Log in with a username and password
* Required level of authorization

Post-conditions:

* Access to interface for adding and/or modifying clients information
* Can make deposit, withdrawal, or close accounts
* Can look at clients information and balance

Basic Flow or Main Scenario:

1. Client fill out application and seek officer for assistance
2. Officer log into a workstation
3. Officer process application
4. Make initial deposit
5. Officer saves application processed
6. Ledger is propagated with new client information
7. Officer prints required document to provide to client
8. Officer log out of workstation

Extensions or Alternate Flows:

* As an extension, an officer is able to perform the same flow of task a teller is able to.
* Closing accounts
  1. Log into a workstation
  2. Note reason for account closure
  3. Withdraw and empty client accounts
  4. Ledger is updated with account closed tag
  5. Print out receipt to provide account closure confirmation
  6. Officer log out of workstation

Exceptions:

* Cannot make changes to account info or balance, if it is currently being access by another workstation.
* Incorrect username and/or password
* Attempt log in at incorrect bank

Related Use Cases: 0021 (Client Assistance)

Use Case ID: 00210

Use Case Name: Client Assistance

Relevant Requirements: Client portion of the bank ledger

Primary Actor: Teller

Pre-conditions:

* Log in with username and password
* Appropriate level of authorization

Post-conditions:

* Able to view clients balance
* Able to assist client with deposit and withdraw
* Able to assist client with transferring of money

Basic Flow or Main Scenario:

1. Log into a workstation
2. Assist clients with deposit
3. Ledger is updated
4. Provide clients with balance and receipt
5. Log out of workstation

Extensions or Alternate Flows:

1. Log into a workstation
2. Assist clients with withdrawal
3. Ledger is updated
4. Provide clients with balance and receipt
5. Log out of workstation

Exceptions:

* Cannot make changes to account info or balance, if it is currently being access by another workstation.
* Incorrect username/password
* Attempt log in at incorrect bank

Related Use Cases: N/A

Use Case ID: 00300

Use Case Name: External Assistance

Relevant Requirements: Client portion of the bank ledger

Primary Actor: ATM

Pre-conditions:

* A card is present and inserted
* Prompt for access pin

Post-conditions:

* Access to check balance
* Access to deposit
* Access to withdraw

Basic Flow or Main Scenario:

1. Client card is inserted
2. Check for validity of card
3. Punch in pin
4. Check validity of pin
5. Access to user’s interface
6. Select Check balance
7. Client balance is shown
8. Provide receipt
9. Close client info
10. Return card

Extensions or Alternate Flows:

* Deposit
  1. Client card is inserted
  2. Check for validity of card
  3. Punch in pin
  4. Check validity of pin
  5. Access to user’s interface
  6. Deposit is selected
  7. Deposit hatch is open and ready for cash/check to be feed in
  8. Close deposit hatch
  9. Confirm amount to be deposited
  10. Confirmation received
  11. Update client only portion of bank ledger
  12. Prompt if client needs to do anything else
  13. Provide receipt
  14. Close client info
  15. Return card
* Withdraw
  1. Client card is inserted
  2. Check for validity of card
  3. Punch in pin
  4. Check validity of pin
  5. Access to user’s interface
  6. Withdraw is selected
  7. Prompt amount to be withdraw
  8. Confirm withdraw
  9. Update client-only portion of bank ledger
  10. Withdraw hatch is opened and cash is expended
  11. Prompt if client needs to do anything else
  12. Provide receipt
  13. Close client info
  14. Return card

Exceptions:

* Incorrect bank affiliation
* Card cannot be read
* Incorrect pin
* Insufficient funds
* Stolen card

Related Use Cases: 00400 (Client)

Use Case ID: 00400

Use Case Name: Client

Relevant Requirements: Client portion of bank ledger

Primary Actor: Client

Pre-conditions:

* Client has at least one account with the bank
* Client has at least one account number
* Client have card on hand
* Client knows their pin code

Post-conditions:

* Is able to make deposit/withdraw
* Is able to check balance
* Is able to transfer money to another bank or account

Basic Flow or Main Scenario:

1. Client goes to a teller for assistant
2. Client insert card or provide account number to teller
3. Client enters pin
4. A deposit/withdraw/transfer/check balance is made
5. Client bank ledger is updated

Extensions or Alternate Flows: N/A

Exceptions:

* Card cannot be read
* Incorrect pin
* Insufficient funds
* Stolen card

Related Use Cases: 00300(External Assistance)



Diagram

Description automatically generated

## Overview

The banking system is designed to keep a record of client information, assist new clients with opening new accounts, making deposits and withdrawals, and transferring funds. As money is a necessity, inaccurate records of funds would lead to legal issues, along with mistrust in our banking system.

# 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

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:

* + - 1. User ids and passwords are alphanumeric strings between 6 and 20 characters in length
      2. Each Client should have a unique bank id and account number, with clients being able to have multiple accounts and accounts able to have multiple users
      3. The system should be able to connect wirelessly across various banks and ATMs
      4. System should use USD as its currency
      5. Clients should be able to deposit and withdraw any amount of money

### ATM 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.

### Server 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.

### Internal System 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

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