Banking Software Application

Software Requirements Specification

Revision History

Date	Revision	Description	Author
02/08/2023	1.0	Initial Version	All group members
02/13/2023	1.1	Modules Modified, Description Update, etc.	All group members
02/28/2023	1.2	Phase 1 Complete Revision	All group members

Table of Contents

1. Purpose

This document outlines the requirements for Banking Software Application.

1.1. 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.

1.2. Definitions, Acronyms, Abbreviations

ATM - Automated Teller Machine. Is a specialized computer that allows you to complete bank transactions without the need to see a bank representative.

Ledger – A collection of accounts in which account transactions are recorded.

Teller - An employee of a bank whose responsibilities include the handling of customer cash and negotiable instruments.

GUI - Stand for Graphical User Interface and is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators such as primary notation, instead of text-based UIs, typed command labels or text navigation.

Java - A high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

HTML - HyperText Markup Language is the standard markup language for documents designed to be displayed in a web browser.

1.3. References

UML Use Case Diagram*

Sequence Diagram*

Class Diagram*

*see 5 below

1.4. Overview

This Banking Software Application is designed to provide multiple interfaces. One interface for customers to interact with an ATM system to perform withdrawals, deposits, and balance checks. Another interface for customers to interact with a bank employee to perform withdrawals, deposits, and account opening/closing. The software is required to have a server client to store bank account information, a ledger, authenticate transactions, and to pass information from bank employee to customer and vice-versa.

2. Overall Description

2.1. Product Perspective

2.2. Product Architecture

The system will be organized into 5 major modules: the ATM module, the Teller module, the Bank Server module, the Ledger module, and the Account module.

2.3. 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):

2.4. Constraints

It is assumed that access to the ATM module by a user can only be within one location at a time, not multiple locations simultaneously.

ATMs must always require online connections to the bank server.

2.5. Assumptions and Dependencies

Assumptions:

A customer should not be able to access the teller without a bank employee's account.

Users should not be able to access another bank account without the correct ID and PIN.

Dependencies:

Users can only access the ATM module by having an account

A user needs the aid of a bank teller to open/close their bank account.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Common Requirements:

3.1.1.1 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.

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

3.1.2. ATM Module Requirements:

3.1.2.1 ATM module should communicate with the bank server in order to process transactions.

3.1.2.2 The ATM module should communicate with the bank server in order to authenticate customer login.

3.1.2.3 The ATM module should provide an interface for the customers to use standard banking services.

3.1.2.4 Transactions made through the ATM will have an imposed limit on withdrawals and deposits to mitigate fraud.

3.1.2.5 After logging in a customer should be able to see all their bank accounts and choose which to access.

3.1.3. Teller Module Requirements:

3.1.3.1 The teller module should communicate with the bank server in order to process transactions.

3.1.3.2 The teller module should provide two interfaces: one for the employee, one for the customer.

3.1.3.3 The teller module should communicate with the bank server in order to authenticate bank employee login and authenticate customer login.

3.1.3.4 A customer should not be able to access the teller without a bank employee logged in, since the bank employee is required to finalize actions through the teller.

3.1.3.5 The customer must specify which bank account they wish to access.

3.1.4. Bank Server Module Requirements:

3.1.4.1 The bank server should communicate with the ATM and Teller modules in order to process transactions.

3.1.4.2 The bank server should perform transactional calculations requested by the ATM and Teller module.

3.1.4.3 The bank server should verify that the bank account has balance to perform the requested transaction.

3.1.4.4 The bank server should send information back to the teller and ATM modules if a transaction has been accepted or declined.

3.1.4.5 The bank server should send information back to the teller and ATM modules if login is successful or failed.

3.1.4.6 The bank server should ignore any transaction requests or account change requests from any teller or ATM user that hasn’t successfully logged in.

3.1.4.7 The bank server should use a getter to check whether or not the requested customer account is being currently accessed. If it is, then deny log in.

3.1.4.8 There should be a time delay to check if an account is trying to being accessed by two or more clients simultaneously, if this occurs, deny login.

3.1.5. Ledger Module Requirements:

3.1.5.1 Upon bank server handling a transaction, that information should be handled by the ledger to be logged as an entry.

3.1.5.2 Every transaction should be written to a text file including: transaction type (withdrawal/deposit), amount changed, who made the transaction (Bank Employee/ATM User).

3.1.6. Bank Account Module Requirements:

3.1.6.1 A bank account can only be created through a teller.

3.1.6.2 A bank account with zero balance must be closed after a certain amount of time.

3.1.6.3 All bank accounts must be associated with a customer.

3.1.6.4 Each bank account is given a unique numerical identifier.

3.1.6.5 A bank account can be one of two types: checking/savings.

3.1.6.6 A bank account can be assigned a nickname which will show up for the teller and at an ATM to make lookup easier.

3.1.7. Customer Account Module Requirements

3.1.7.1 Each customer will have a username, password, and have accounts associated with it.

3.1.7.2 Each customer can have one bank account or more.

3.1.7.3 Each customer should have a boolean flag indicating whether or not it is in use.

3.1.7.4 A customer can update their user account information through a teller.

3.1.8. Bank Employee Account Module Requirements

3.1.7.3 Each customer should have a boolean flag indicating whether or not it is in use.

3.1.7.4 A customer can update their user account information through a teller.

3.2. External Interface Requirements

3.2.1 The system must provide a GUI interface to the customer in order to access the ATM to login, check balances, and complete transactions.

3.2.2 The system must provide a GUI interface to the customer in order to access the customer portion of the teller module.

3.2.3 The system must provide a GUI interface to the bank employee in order to access the customer portion of the teller module.

3.3. Internal Interface Requirements

3.3.1 The system must process data from the bank server's handled transactions and send them to the ledger. The Ledger will write all this information to a text file for bookkeeping.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

4.1.1 Every user must have a unique login username.
4.1.2 A user account should not be able to be accessed in two different locations simultaneously
4.1.3 Add a time delay for logins to check if there are multiple clients trying to access the same account simultaneously
4.1.4 User account password/pin must be blurred or obfuscated when being entered.

4.2. Environmental Requirements

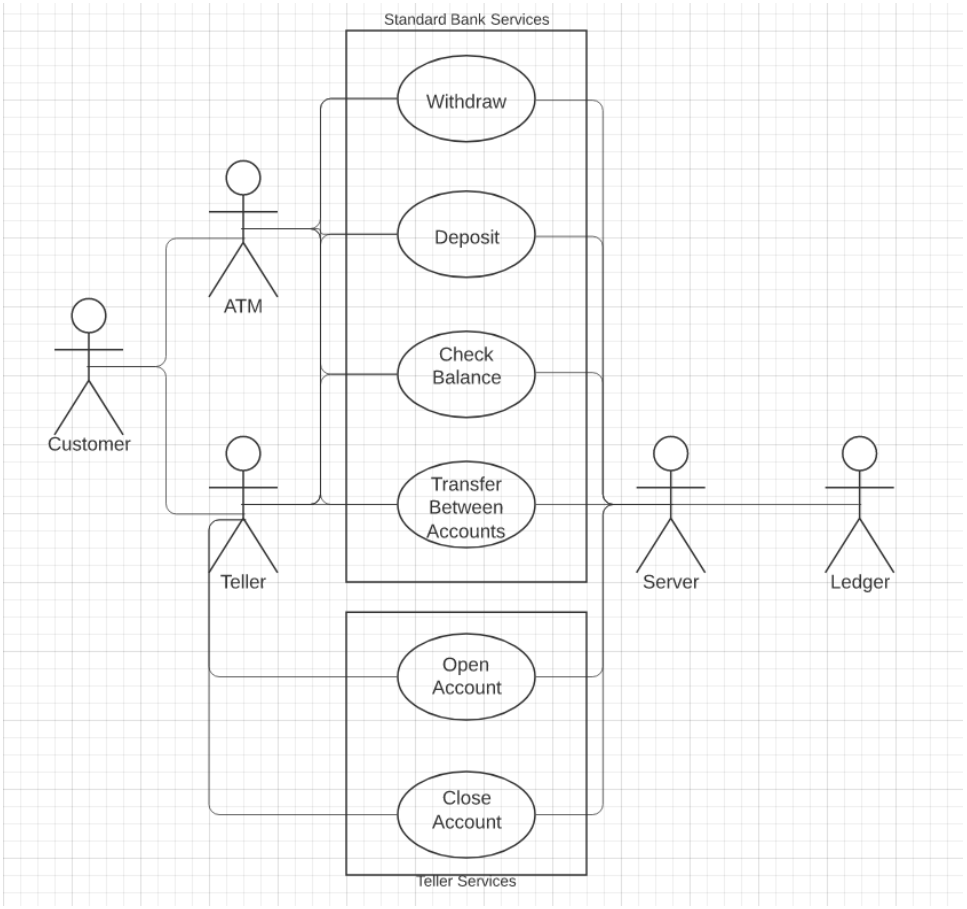
4.2.1 System must be created as a Java program.
4.2.2 System modules must communicate over TCP/IP.

4.3. Performance Requirements

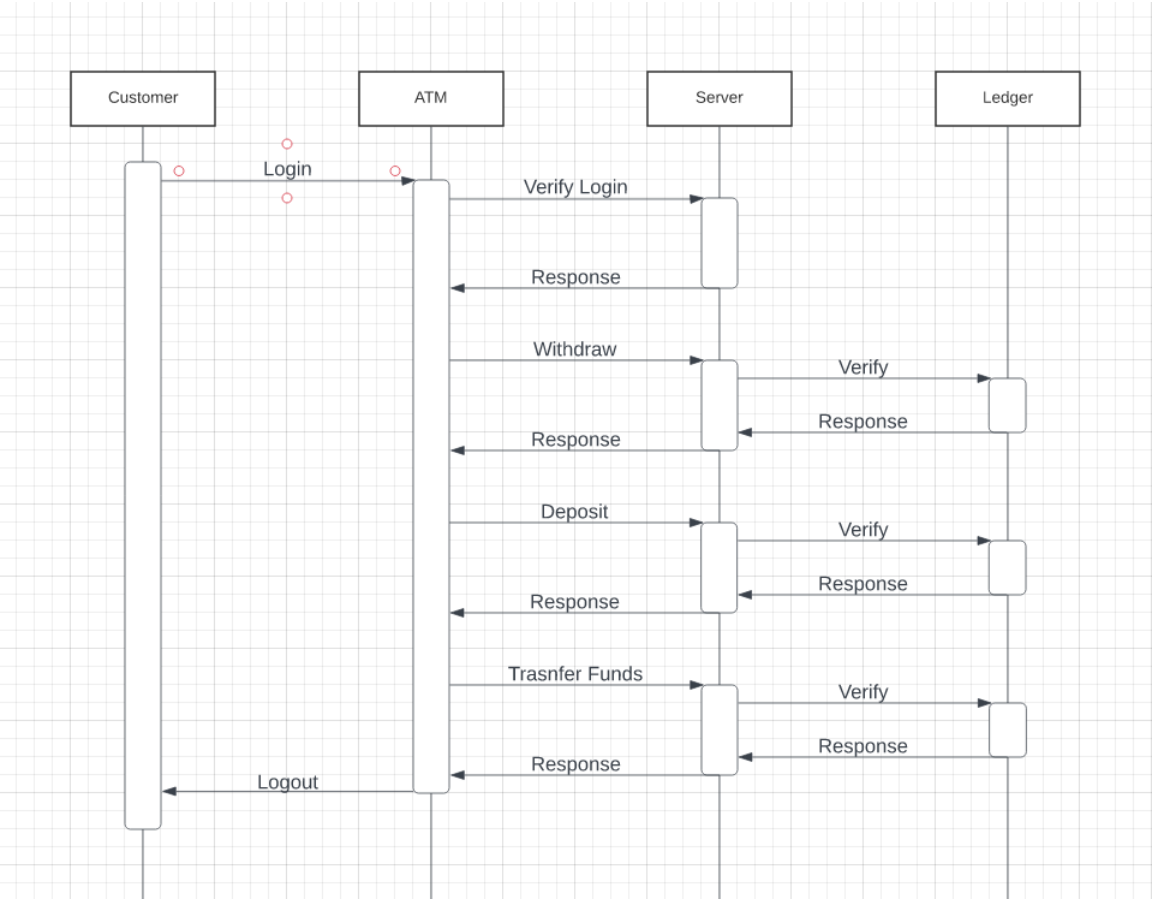
4.3.1 System must be able to process handling multiple clients (ATM Users, Teller Users) and requests. A multithreaded server is required.

5. UML Diagrams

5.1 Use Case Diagram



5.2 Sequence Diagram



5.3 Class Diagram