Banking Software System

Software Requirements Specification

Revision History

Date	Revision	Description	Author
mm/dd/yyyy	X.X	Initial Version	Your Name
09/24/2024	1.0	- Separated GUI requirement	Nicholas Manha
09/25/2024	1.1	- Added terms to be defined for SRS 1.2, - defined Banking Software System, Customer, Teller, Automated Teller System, Bank's Services, Fraud - Added 2.3 Product Functionality/Features	Edgar Romero
09/25/2024	1.2	- Added Constraints for 2.4 - Added Security and Privacy Requirements for 4.1	Kyle Lam
09/26/2024	1.3	- Added 4.2 Environmental Requirements - Added 3.1.1 Common Requirements - Added 2.5 Assumptions and Dependencies	Nicholas Manha
09/27/2024	1.4	- Added 10 Module requirements Across 3.1.2, 3.1.3, 3.1.4 - Added 3.3 and 8 specific Internal Interface Requirements across each Module in 3.3	Edgar Romero
09/28/2024	1.5	- Added 2.1 Product Perspective - Added 3.2.1 - 3.2.5 External Interface Requirements - Added 4.2.3 - 4.2.4 Environmental Requirements	Nicholas Ferreira
09/28/2024	1.6	- Added 4.3 Performance Requirements, Defined terms in 1.2: Account Number, Checking Account, Savings Account, Balance, PIN, Account Recovery.	Edgar Romero
09/29/2024	1.7	- Added 2.5.1, 2.5.2, 2.5.3 of the 2.5 Assumptions - Added 1.2 Definitions of Deposit, Withdraw, Transfer, Account Freeze, and GUI	Kyle Lam
09/29/2024	1.8	- Added Use Case 1, 2	Edgar Romero
09/29/2024	1.9	- Combined 2.5 requirements - Added Use cases 3, 4, 5 - Added common requirement 3.1.1.6	Nicholas Manha

09/30/2024	1.10	- Transferred over all existing work into final SRS	Edgar Romero
10/01/2024	1.11	- Added Use case 10	Nicholas Manha
10/01/2024	1.12	- Added 1.4 Overview - Added Use case 6 and 9	Kyle Lam
10/01/2024	1.121	- Added Use Case 7	Nicholas Ferreira
10/01/2024	1.13	- Added Use case 8, - Added definition Receipt, - Updated Log Requirements in 3.1.5 and 3.3.4 to reflect added Receipt definition - Added Related Use Cases for Use Case 1-4, reformatted Use Cases to be more consistent.	Edgar Romero
10/03/2024	1.14	- Added requirements 3.1.3.5, 3.1.4.4, 3.1.4.5 - Filled in related use cases for use cases 10, 3, 4, 5 - Filled in relevant requirements for use cases 10, 3, 4, 5 - Defined TCP/IP, Port, Server, Client - Added use case diagram - Added class diagrams	Nicholas Manha
10/03/2024	1.15	- Add some class attributes and methods for ATM, Log, and Customer classes	Kyle Lam
10/28/2024	1.16	- Added common requirement 3.1.1.7	Nicholas Manha

Table of Contents

1. Purpose

This document outlines the requirements for the Banking Software System..

1.1. Scope

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

1.2. Definitions, Acronyms, Abbreviations

1.2.1 Banking Software System: The Banking Software System, or BSS, refers to the system of programs which make up the application that allows bank employees and customers to engage with the bank and all the services it provides including deposits, withdrawals, money transfers, account freezing, account recovery, and more.
1.2.2 Customer: A customer refers to the person who owns or will sign up for a bank account(s), and utilizes the bank's services through a Teller or through the ATM system.
1.2.3 Bank Teller: A Bank Teller, or Teller, is an employee of the bank that facilitates account transactions for a Customer that the Customer cannot do themselves on an ATM.
1.2.4 Automated Teller Machine: An Automated Teller Machine, or ATM, is a computer that provides a GUI for a Customer to complete quick transactions with their bank accounts which don't require a Teller.
1.2.5 Bank's Services: This term refers to every service the Bank provides for managing the Customers' bank accounts and the money coming to and from the bank accounts, as performed by a Teller or via ATM.
1.2.6 Account Number: A unique identifier associated with a Customer's Account.
1.2.7 Checking Account: An Account made for Customers to conveniently manage and access their money.
1.2.8 Balance: The current amount of money stored within a given Account.
1.2.9 Savings Account: An Account made for Customers to primarily store money.
1.2.10 PIN: Refers to the sequence of numbers a Customer uses to verify they have access to an account.
1.2.11 Fraud: Fraud refers to any illegitimate addition or subtraction of money from an Account through any of the Bank's Services using an ATM, a Bank Teller, or through any unidentified exploits within the BSS' Server-Client infrastructure.
1.2.12 Account Recovery: Account Recovery is a process a Bank Teller and Customer go through to regain access to a Bank Account that a Customer may have forgotten their PIN or Account Number for.
1.2.13 Transaction: The completion of any action related to managing the money in an Account or managing aspect of an Account itself are referred to as Transactions.
1.2.14 Sequential Transactions:
1.2.15 Deposit: The action of adding funds to a customer's bank account, typically through an ATM or by a bank employee. A deposit increases the balance of the account.
1.2.16 Withdraw: The action of removing funds from a customer's bank account which decreases the amount of the account's balance. It can be done by the ATM or bank teller.
1.2.17 Transfer: The action of moving funds from one bank account to another and can occur between a customer's own accounts or between accounts owned by different customers

1.2.18 Account Freeze: When a bank account is placed under a temporary restriction to prevent any transactions (like withdrawals, deposits, or transfers). Only bank tellers can initiate or stop an account freeze.
1.2.19 GUI: The interface that allows users to interact with the banking system visually and uses graphical elements like buttons, windows, and forms.
1.2.20 TCP/IP: Transmission Control Protocol/Internet Protocol
1.2.21 Port: A number to uniquely distinguish multiple connections to a single IP.
1.2.22 Server Application (Server): A computer that provides resources to other computers
1.2.23 Client Application (Client): A computer that receives the resources given by a server
1.2.24 Receipt: A Receipt is the message printed to the screen/given to a customer by a Bank Teller after the Customer performs or requests a transaction be completed from their account.

1.3. References

Use Case Specification Document –

UML Use Case Diagrams Document –

Class Diagrams –

Sequence Diagrams –

1.4. Overview

The Banking Software System is designed to facilitate secure and efficient interactions between customers and bank tellers. It provides ATM services with services to minimize fraud. From these services and roles, there will be two main interfaces of an ATM interface for the customer and a Teller interface for bank employees. The BSS will be using a Client-Server architecture over TCP/IP.

2. Overall Description

2.1. Product Perspective

This Banking System Software is designed as a standalone Bank Employee and ATM services application. This application comprises a client and server component communicating over TCP/IP.

2.2. Product Architecture

2.2.1 System Interfaces:

This application follows a client-server architecture. Users interact with the **client-side application** either through a bank teller(operated by a bank employee) or the ATM service(used by customers). Regardless of the interaction method, user actions are processed by the server-side application, which handles requests and other banking operations.

2.2.2 User Interfaces:

This banking software is split into two user interfaces. The first is the **Bank Employee Interface** which allows authorized staff members additional privileges. This includes actions like creating a user's PIN and linking different accounts. The customer and employee will use the same GUI, but certain actions will be hidden/unhidden depending on privileges. The second is the **ATM interface** which allows Customers to perform basic transactions, such as withdrawals and deposits, without a human teller.

2.2.3 Hardware/Software Interfaces:

The client-side application will run on standard desktops used by employees at the bank as well as standard ATMs. The server-side application will run on the bank's internal server hardware infrastructure.

Additionally, The Banking Software System will be organized into 4 major modules: the GUI module, the User module, the Account module, and the Log 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):

2.3.1 The BSS will include a client application that the Customer can use to access the Bank's Services through an ATM or Bank Teller.
2.3.2 The BSS will include a Server application that communicates with the Client application using TCP/IP.
2.3.3 The BSS will provide a GUI for both the Customer to use the ATM and a Bank Teller to help a Customer.
2.3.4 A Customer on the Client can use the ATM to withdraw, deposit, view, and transfer money.
2.3.5 A Customer on the Client can communicate with a Bank Teller to open a new bank account.
2.3.6 A Customer on the Client can communicate with a Teller to change their PIN.

2.3.7 A Customer on the Client can communicate with a Teller to recover their Account Number
2.3.8 A Customer on the Client can communicate with a Teller to initiate an Account Freeze in the case of suspected Fraud.
2.3.9 A Customer on the Client can communicate with a Teller, who can perform any action the Customer can perform themselves using the ATM.
2.3.10 The BSS will identify accounts via an Account Number.
2.3.11 Each bank account will be associated with a unique Account Number.
2.3.12 A Customer can own one or more accounts.
2.3.13 A Customer can access any account they have the Account number and PIN for.
2.3.14 Every action and transaction done by a Customer or Teller will have confirmation of completion in the form of a Receipt after the Customer/Teller finishes using the Bank's Services.
2.3.15 Every action involving a Bank Account will strictly occur in a sequential fashion to ensure instances of Fraud are avoided.

2.4. Constraints

2.4.1 The system cannot use a database, frameworks, or other external technologies and must store data in a text file.

2.4.2 The system must operate using TCP/IP for communication between the Client and Server applications. No other communication protocols may be used.

2.4.3 Server-Client communication must occur using ports above 1024, because lower ports are reserved for other services.

2.4.4 Encryption can't be used for transmitting sensitive information like account numbers, PINs, or transaction details between the Client and Server. Security must rely on other ways.

2.4.5 The user interfaces for the Teller and ATM must be built using Java's GUI libraries

2.5. Assumptions and Dependencies

2.5.1 It is assumed that customers are communicating with the teller to perform actions they request.

2.5.2 Money that is deposited by a customer is assumed to be real tender.

2.5.3 There must be at least one trained teller capable of operating the employee duties.

2.5.4 The system relies on TCP/IP communication for all transactions so it is assumed that both the client and server applications will be deployed in spots with good, stable network connections.

2.5.5 It is assumed that only real humans are using the application and that there are no bots.

2.5.6 The user running the software must have an operating system that can run Java programs.

2.5.7 It is assumed that all users accessing the system are given correct credentials and proper logins and that security policies are followed by all personnel.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Common Requirements:

3.1.1.1 Make account interactions sequential

3.1.1.2 Customer login capability with Account ID and pin number

3.1.1.3 All elements/features must be written in java and follow the standards of object oriented programming.

3.1.1.4 All network messages must utilize TCP/IP with a server client build pattern.

3.1.1.5 Any data that cannot be kept in an object must be stored in a text file.

3.1.1.6 If a customer wants to be added to an existing account, the owner of the account must consent to adding this new customer.

3.1.1.7 Each customer on an account gets their own unique pin (unique to that account).

3.1.2. GUI Module Requirements:

3.1.2.1 There will be one GUI for each of the different types of users of the BSS; Customers and Bank Tellers.

3.1.2.2 The GUI will provide the interface for which a Customer manages the money in their Bank Accounts through an ATM.

3.1.2.3 The GUI will allow the Bank Teller to manage the money in a Customer's Account on behalf of a Customer, as well as manage properties of the account itself.

3.1.3. User Module Requirements:

3.1.3.1 There will be two kinds of users of the BSS, Customers and Bank Tellers.

3.1.3.2 Customers can only check their Balance, initiate a money Transfer, Deposit and Withdraw money from their Bank Accounts.

3.1.3.3 Tellers have more permissions within the BSS to do more than a Customer could on their own at an ATM. These services they can provide on behalf of a Customer include Freezing an Account, Resetting a PIN, Recovering an Account.

3.1.4. Account Module Requirements:

3.1.4.1 There will be two different types of Accounts available to Customers: Checking Accounts and Savings Accounts.

3.1.4.2 Each Account in the BSS will have a unique Account Number and PIN and can only be accessed using these credentials.

3.1.4.3 An Account can only be created by a Bank Teller.

3.1.4.4 A Customer Account can be Frozen by a Bank Teller at any time.

3.1.4.5 Each Account ID must be unique and linked to only one Account

3.1.5. Log Module Requirements:

3.1.5.1 Logs will record the date and time of the associated service performed.

3.1.5.2 Logs will record who the service was performed for, who initiated the Service,

3.1.5.3 A receipt can be made from the information provided on a log created to document any transaction or service, and will contain a date, time, the amount of money in the transaction, specify the Transaction, and the Accounts involved in the transaction.

3.1.3.4 Each Log will have a unique ID.

3.1.3.5 Only a Teller is able to read Logs and they can be accessed at any time.

3.2. External Interface Requirements

3.2.1 The BSS must support standard desktop hardware used at the bank.

3.2.2 The physical ATM must communicate with the bank's server infrastructure.

3.2.3 The BSS must be supported by the bank's operating system (both desktop and ATM).

3.2.4. Every Transaction and Account action that occurs as a result of a Customer using the ATM, or a Bank Teller aiding a Customer will be logged and stored into a text file, with a timestamp of when the event occurred.

3.3. Internal Interface Requirements

3.3.1 GUI Module Requirements

3.3.1.1 The BSS will operate using a Server-Client design pattern.

3.3.1.2 The Server-Client communication with an ATM or Bank Teller will occur synchronously.

3.3.1.3 The Bank Teller and Customer's GUI will not interact with each other.

3.3.2 User Module Requirements

3.3.2.1 The Customer will not be able to change properties of a Bank Account, such as the PIN used to access it or whether an Account Freeze is placed on it or not, without the help of a Bank Teller; these services are available to every customer, but the ATM where Customers can self-help will not allow for anything beyond simple transactions and checking the balance of a Bank Account.

3.3.3 Account Module Requirements

3.3.3.1 The ATM will allow the Customer to interface with all of their Bank Accounts, or any Bank Account they have the credentials for, but only one Account per session.

3.3.3.2 If multiple Customers are given access to the same Bank Account, they may be logged into the same Account via ATM or Teller at the same time, however the Sequential Transaction feature will not allow for the Customers to initiate any action simultaneously.

3.3.4 Log Module Requirements

3.3.4.1 Every Transaction and Action will show a confirmation of its completion after it is complete via a notification to the Customer or Bank Teller from their respective user interfaces. In the case of a Customer at an ATM, a receipt will be printed.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

4.1.1 All interactions with bank accounts (like deposits, withdrawals, transfers) must occur sequentially to prevent inconsistencies and reduce the risk of fraud. No simultaneous transactions on the same account are allowed.
4.1.2 Users (bank tellers / customers) must authenticate using a secure login. Passwords and PINs will be checked against stored credentials, and role based access must make sure that only authorized users access specific functions (like only tellers can open accounts).
4.1.3 The system must automatically log out inactive users after a defined period (like 10 minutes of inactivity) to prevent unauthorized access to an account if left unattended.
4.1.4 All transactions and changes to accounts (like PIN updates, withdrawals) must require user confirmation before they are finalized. This ensures that the user intentionally performs these actions.
4.1.5 All significant actions or Bank Services provided (account creation, deposits, withdrawals, PIN changes, logins) must be logged with timestamp, user ID, and details of the action. These logs will be used for checking and will be stored in a secure spot that is only accessible by authorized employees/personnel.

4.2. Environmental Requirements

4.2.1 The system must be able to run on any operating system.

4.2.2 The system mustn't require any 3rd party software in order to run.

4.2.3 The BSS must be available to users during typical business hours.

4.2.4 The BSS should store server-side logs if environmental disasters destroy local copies.

4.3. Performance Requirements

4.3.1 GUI load times must be near instantaneous to provide a seamless user experience.
4.3.2 There will be no noticeable delay between the Server and Client, and the outcomes from any requested Bank Services will be immediately reflected in the Customer's Accounts.
4.3.3 The hardware included in the ATMs and desktops used for the Client-side application of the BSS, and the machines comprising the Bank's internal infrastructure which host the Server-side application must be relatively modern, having been manufactured within the last 12 years, to insure the BSS is able to run efficiently

Banking System Use Case Specification Document

Use Case ID: 1
Use Case Name: Teller Creates an Account
Relevant Requirements: 3.1.4.3

Primary Actor: Bank Teller

Pre-conditions:
1. A Customer must request from a Teller that a Bank Account be made for them.

Post-conditions:
1. The newly made Bank Account will be accessible by the Customer and a Teller via PIN and Account Number.
2. This event has the date and time recorded in a log.

Basic Flow or Main Scenario:
1. A Bank Teller asks the Customer what kind of Account they would like to establish with the Bank.
2. The Bank Teller creates a new Checking Account.
3. A PIN number is created for the Account.
4. The Bank Teller gives the Customer their Account number and PIN number.
5. The Customer gets a receipt detailing the date and time of their Account's creation.

Extensions or Alternate Flows:

Alternate Flow 1:
1. A Bank Teller asks the Customer what kind of Account they would like to establish with the Bank.
2. The Bank Teller creates a new Savings Account.
3. A PIN number is created for the Account.
4. The Bank Teller gives the Customer their Account number and PIN number.
5. The Customer gets a receipt detailing the date and time of their Account's creation.

Exceptions:

N/A

Related Use Cases:
1. Use Case 7: Logging Transactions
2. Use Case 8: Print Receipt

Use Case ID: 2
Use Case Name: Customer manages money at ATM
Relevant Requirements: 3.1.4.3, 3.3.3

Primary Actor: Customer

Pre-conditions:
1. There must exist an Account for a Customer to access and manage.
2. The Customer must have money in their Account to manage, or on hand to deposit into their Account.

Post-conditions:

1. A receipt of the completed Transactions and changes in the total Balance of the Account accessed.
2. This event has the date and time recorded in a log.

Basic Flow or Main Scenario:

1. A Customer requests to make a Deposit at an ATM.
2. The ATM prompts the user for an Account Number and PIN for access to an Account.
3. The Customer enters their credentials.
4. The Customer deposits money into an account.
5. A Receipt with the date, time and amount deposited is printed for the Customer.

Extensions or Alternate Flows:

Alternate Flow 1:

1. A Customer requests to make a Withdrawal at an ATM.
2. The ATM prompts the user for an Account Number and PIN for access to an Account.
3. The Custo
4. mer enters their credentials.
5. The Customer withdraws money from their account in an amount that does not exceed their Balance.
6. A Receipt with the date, time and amount withdrawn is printed for the Customer.

Alternate Flow 2:

1. A Customer requests to make a Transfer at an ATM.
2. The ATM prompts the user for an Account Number and PIN for access to an Account to Transfer money from.
3. The Customer enters their credentials.
4. The Customer then enters the amount of money they would like to transfer into another Account.
5. The ATM prompts the user for an Account Number.
6. The transaction occurs and a receipt with the date, time, Account Numbers and the amount transferred is printed for the Customer.

Exceptions:

1. A Customer cannot attempt to Transfer or Withdraw more money than their total Account Balance

Related Use Cases:

1. Use Case 4: Customer Transfers money between Accounts
2. Use Case 6: Check Account Balance
3. Use Case 7: Logging Transactions
4. Use Case 8: Print Receipt

Use Case ID: 3
Use Case Name: Bank Teller adds Customer to Account
Relevant Requirements: 3.1.1.6, 3.1.3.3

Primary Actor: Teller

Pre-conditions:

1. There must be an Account for the teller to add the Customer to.

Post-conditions:

1. The Customer is added to said Account.
2. The Customer is given his own login for the Account and can.
   interact with it.
3. A log detailing the addition of a Customer to an Account is created.

Basic Flow or Main Scenario:

1. Customer requests that the teller add them to an Account.
2. A prompt is given to the Account owner asking if they want this Customer allowed on their Account.
3. The Account owner agrees.
4. That Customer is allowed access to said Account.
5. Login credentials are given to said Customer.
6. The event that the Customer was added to another Account is logged.

Extensions or Alternate Flows:

7. Customer requests that the Teller add them to an Account.
8. A prompt is given to the account owner asking if they want this Customer allowed on their Account
9. The Account owner says no
10. That Customer is rejected from being added to that Account.

Exceptions:

1. The requested Account does not exist.
2. The owner of the Account does not consent to adding the new Customer

Related Use Cases:

1. Use Case 8: Print Receipt

Use Case ID: 4
Use Case Name: Customer Transfers money between Accounts
Relevant Requirements: 3.1.2.2, 3.1.2.3, 3.1.3.2, 3.1.3.3

Primary Actor: Customer

Pre-conditions:

1. A Customer must have an Account with money, and a target Account to send money to.

Post-conditions:

1. Money is transferred from the Customer's Account to the target Account.
2. A log detailing this transfer of money is created.

Basic Flow or Main Scenario:

1. Customer logs into their account
2. Customer begins a transfer operation using ATM GUI
3. ATM prompts customer with target account
4. Customer enters target account ID
5. The ATM prompts a Customer with what Account they want to send money from
6. A Customer chooses a Checking Account
7. ATM prompts Customers with how much money they wish to send
8. Money is sent from the customers account to the target Account
9. A receipt with the date, time and amount Transferred is printed for the Customer

Extensions or Alternate Flows:

1. Customer requests a Teller Transfer money from a source Account to a target Account.
2. Teller accesses funds from source Account.
3. Teller Transfers requested an amount of money to target Account.

Exceptions:

1. Insufficient funds in sender Account.
2. Target account does not exist.

Related Use Cases:

1. Use Case 2: Customer manages money at ATM
2. Use Case 7: Logging Transactions

3. Use Case 8: Print Receipt

Use Case ID: 5
Use Case Name: An Account is accessed using PIN and account number
Relevant Requirements: 3.1.1.2, 3.1.4.2

Primary Actor: Customer

Pre-conditions:
1. The account that is being accessed must exist.

Post-conditions:
1. The customer is logged into their account and is allowed to manage their funds.

Basic Flow or Main Scenario:
1. The Customer enters their PIN and Account Number.
2. ATM logs them into their account
3. A receipt of any transactions performed on the Customer's Account is created.

Extensions or Alternate Flows:
1. Teller is given a Customers PIN and Account Number.
2. Teller logs into the Customer's Account.
3. A receipt detailing every one of the Bank Services performed by the Bank Teller is created.

Exceptions:
1. Credentials entered are not associated with any Account
2. Credentials entered do not permit the customer to access the Account associated with said credentials.

Related Use Cases:
1. Use Case 2: Customer manages money at ATM
2. Use Case 6: Check Account Balance

Use Case ID: 6
Use Case Name: Check Account Balance
Relevant Requirements: 3.1.2.2, 3.1.3.2

Primary Actor: Customer

Pre-conditions: The customer must be logged in with a valid account and PIN.

Post-conditions: The current account balance is displayed, and no further changes are made.

Basic Flow or Main Scenario:
1. Customer logs into account
2. Customer selects the option to check balance on ATM interface
3. ATM displays the customer's current balance
4. Customer views the account balance
5. ATM offers the Customer the option to perform another action or to log out.

6. Customer logs out and ATM ends the session

Extensions or Alternate Flows:
Alternate Flow 1:
1. Customer logs in with invalid credentials
2. ATM sends prompt informing Customer that credentials are invalid
3. Customer is prompted to enter credentials again or stop
4. This flow continues until Customer enters valid credentials or choses to stop

Exceptions:
N/A

Related Use Cases:
1. Use case 2: Customer manages money at ATM
2. Use case 4: Customer Transfers money between Accounts
3. Use case 5: An Account is accessed using PIN and account number

---

Use Case ID: 7
Use Case Name: Logging Transactions
Relevant Requirements: 3.1.1.5, 3.1.5.1, 3.1.5.2, 3.1.5.3

Primary Actor: Log file

Pre-conditions:
1. A valid action was taken in the banking system, whether it's withdrawing, depositing, or transferring.

Post-conditions:
1. The log file will have the recent transaction added.

Basic Flow or Main Scenario:

1. The customer initiates a banking action such as withdrawing, depositing, or transferring with a teller.
2. The teller performs the corresponding response.
3. Once the action is performed, the logging file will add the date, time, the amount of money in the transaction, specify the transaction itself, and the accounts involved in the transaction, and a unique id.

Extensions or Alternate Flows:
Alternate Flow 1 (ATM transaction)
1. The customer initiates a banking action such as withdrawing, depositing, or transferring with the teller with an ATM.
2. The ATM performs the corresponding response.
3. Once the action is performed, the logging file will add the date, time, the amount of money in the transaction, specify the transaction itself, and the accounts involved in the transaction, and a unique id.

Exceptions:
1. Text file not found

Related Use Cases:
1. Use case 8: Print Receipt
2. Use case 9: View Transaction Logs

Use Case ID: 8
Use Case Name: Print Receipt
Relevant Requirements: 3.1.5.3, 3.3.4

Primary Actor: User

Pre-conditions:
1. There must exist an Account for a Customer to access and perform a service on.
2. A Bank Service is performed for a Customer

Post-conditions:
1. A change in the Customer's Account occurs.
2. A log is created detailing the transaction.

Basic Flow or Main Scenario:
1. A Customer is prompted for their credentials to access an Account at the ATM.
2. A Customer logs into their Account at an ATM.
3. The Customer performs a Transaction at the ATM.
4. A Receipt is printed for the user detailing the transaction.

Extensions or Alternate Flows:
Alternate Flow 1:
1. A Customer requests one of the Bank's services through a Teller.
2. The Teller performs the Bank Service.
3. A Receipt is printed for the Customer detailing the transaction.

Exceptions:
1. A receipt will not be printed if the user is only checking their Account Balance.

Related Use Cases:
1. Use case 7: Logging Transactions

Use Case ID: 9
Use Case Name: View Transaction Logs
Relevant Requirements: 3.1.5.1, 3.1.3.5

Primary Actor: Teller

Pre-conditions:
1. The Customer or Teller is authenticated and the system has stored a log of transactions in a text file associated with the account

Post-conditions:
1. The Transaction is retrieved from the text file and displayed to the user and no changes were made to the Account or Log file.

Basic Flow or Main Scenario:
1. Customer or Teller picks the view transaction log option on their respective GUI
2. Text File Log System retrieves the relevant log file, reads the data, and returns it to the GUI
3. GUI displays the transaction logs for the user.
4. ATM/GUI asks user to perform another option

5. User chooses to log out and session ends

Extensions or Alternate Flows:
Alternate Flow 1: No transaction history
1. Customer or Teller picks the view transaction log option on their respective GUI
2. Text file retrieves the log file but can not find any records.
3. Text file will return a message that says the file is empty
4. GUI will display that message
5. User chooses to log out and the session ends.

Exceptions:
N/A
Related Use Cases:
1. Use case 7: Logging Transactions

Use Case ID: 10
Use Case Name: Teller Freezes Account
Relevant Requirements: 3.1.4.4

Primary Actor: Teller

Pre-conditions:
1. The Account that is being frozen must exist.

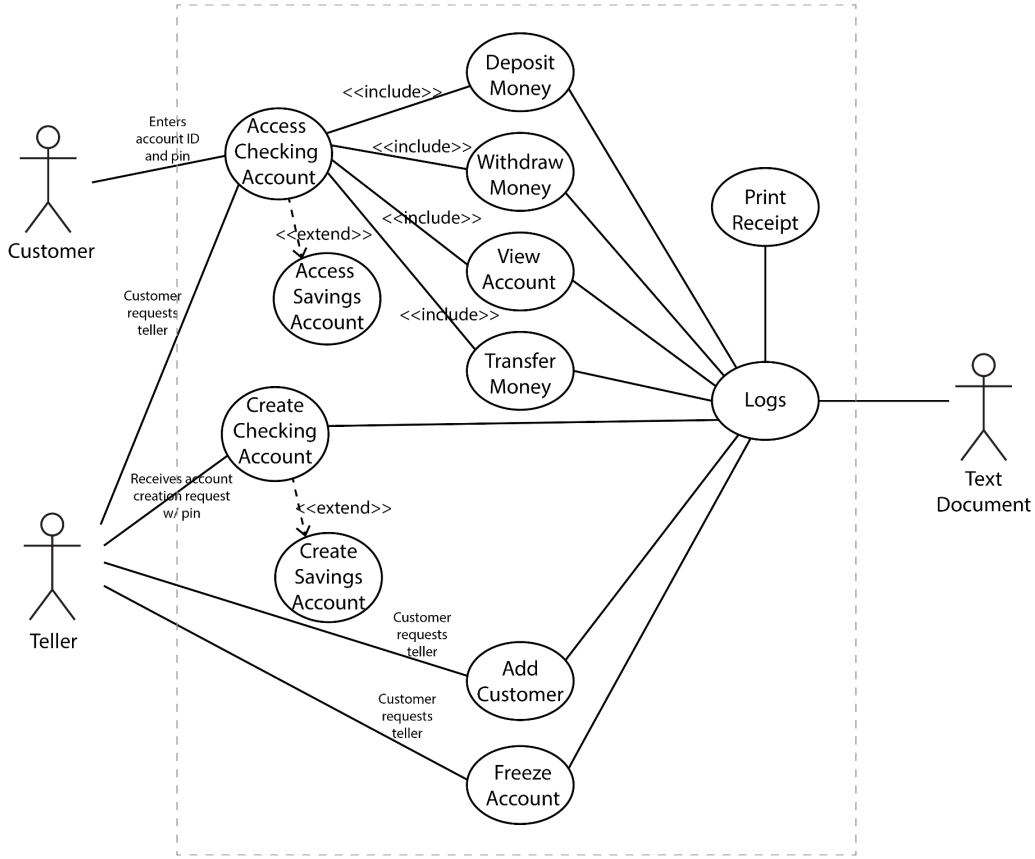
Post-conditions:
1. The target Account is flagged as Frozen, meaning Bank Services cannot be performed on that account other than viewing it.

Basic Flow or Main Scenario:
1. Teller receives Account ID
2. Teller flags account as Frozen

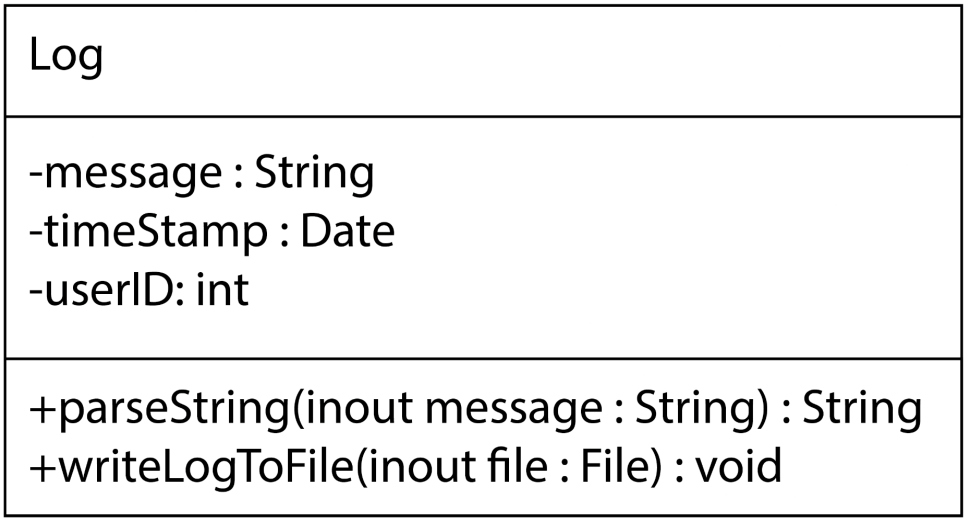
Extensions or Alternate Flows:
N/A
Exceptions:
1. Account doesn't exist

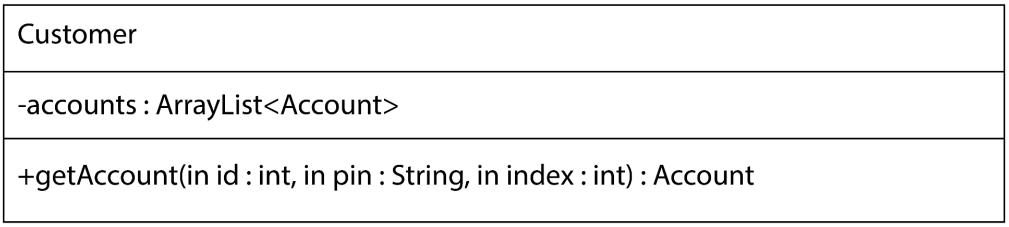
Related Use Cases:
1. Use case 1: Teller Creates an Account
2. Use Case 7: Logging Transactions

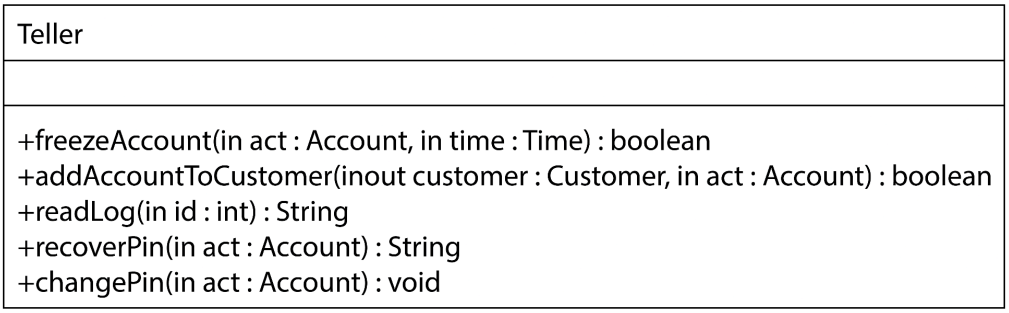
Bank Software System Use Case Diagram

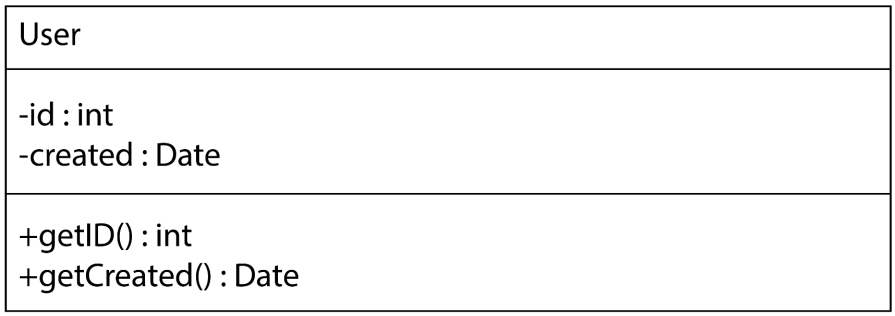


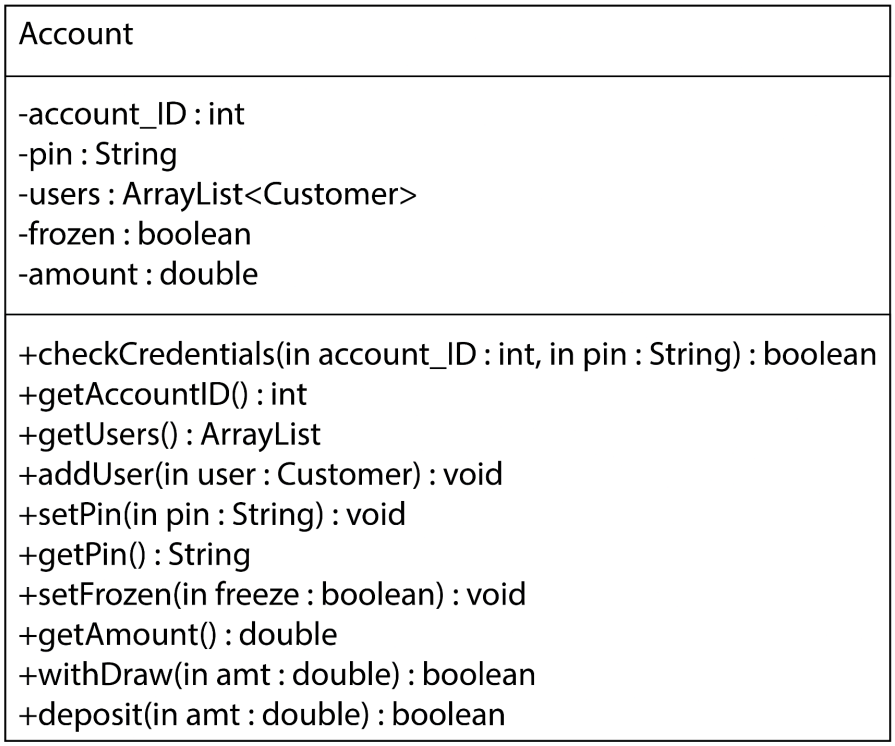
Bank Software System Class Diagrams

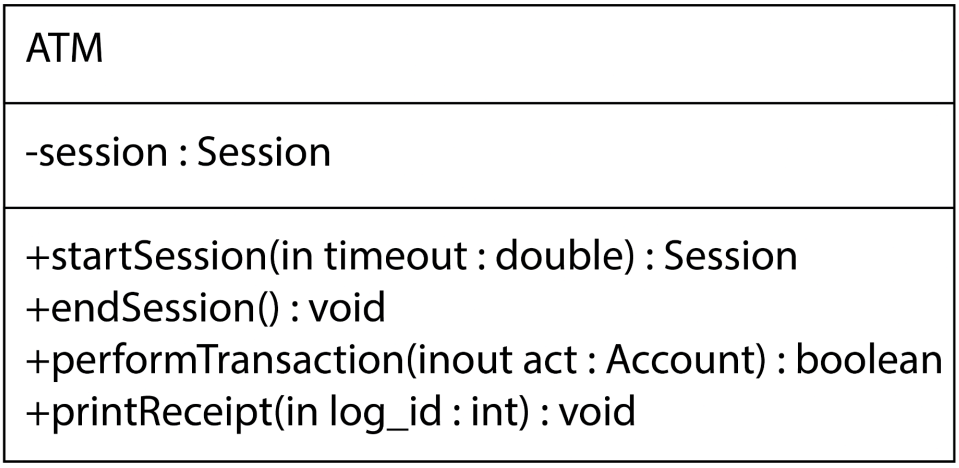







User
-id : int
-created : Date
+getID() : int
+getCreated() : Date


Account
-account_ID : int
-pin : String
-users : ArrayList<Customer>
-frozen : boolean
-amount : double
+checkCredentials(in account_ID : int, in pin : String) : boolean
+getAccountID() : int
+getUsers() : ArrayList
+addUser(in user : Customer) : void
+setPin(in pin : String) : void
+getPin() : String
+setFrozen(in freeze : boolean) : void
+getAmount() : double
+withDraw(in amt : double) : boolean
+deposit(in amt : double) : boolean


ATM
-session : Session
+startSession(in timeout : double) : Session
+endSession() : void
+performTransaction(inout act : Account) : boolean
+printReceipt(in log_id : int) : void