
Software Requirements Document for BANK System

Author: Andrew Laird, John Chargo, Guillermo Hernandez, John Leacox

Version	Date	Author	Change
0.1	09/05/04	SM	Initial Document
0.2	02/01/06	JC	Wrote 1.1-1.3, 2 Overview-2.1.3, 2.3 Designed sequence diagrams for 2.2.5 - 2.2.8
0.3	02/01/06	AL	Designed 2.2.3 - 2.2.5 Designed sequence diagrams for 2.2.3 and 2.2.4
0.4	02/01/06	GH	Designed 2.2.1, 2.2.2, 2.2.6 – 2.2.8 Designed sequence diagrams for 2.2.1-2.2.2
0.5	02/02/06	JL	Wrote Feature Outline
0.6	02/06/06	AL	Wrote sections 3.3 and 3.5
0.7	02/06/06	GH	Wrote Sections 3.1 and 3.6
0.8	02/06/06	JL	Wrote sections 3.2
0.9	02/06/06	JC	Wrote Sections 3.1

Table of Contents

1	Introduction.....	3
1.1	Purpose	3
1.2	Definitions, acronymns, abbreviations	3
1.3	References	3
1.4	Overview	3
2	Overall Description.....	4
2.1	Product Perspective.....	4
2.2	Product functions	6
2.3	User characteristics	23
2.4	Constraints.....	23
2.5	Assumptions and Dependencies.....	23
3	Specific Requirements	24
3.1	External Interface Requirements.....	25
3.2	FEATURES.....	28
3.3	Performance requirements.....	29
3.4	Design Constraints.....	Error! Bookmark not defined.
3.5	Software System Attributes.....	29
3.6	Other Requirements	31
	[None]	31
	Appendix A: ATM Screen-flow Diagram	32
	Appendix B: ATM Screenshots	33
	Appendix C: Teller Screen-flow Diagram	34
	Appendix D: Teller Screenshots	35

Deleted: 1 . Introduction . 4¶
1.1 . Notations and Terms . 5¶
1.1.1 . Requirement indicators . 5¶
1.1.2 . Read/Write/Edit . 5¶
2 . Functional Requirements . 6¶
2.1 . Control Access Levels . 6¶
2.1.1 . Types of access levels . 6¶
2.2 . Administer System . 8¶
2.2.1 . Action selection . 8¶
2.2.2 . Setup system indicators . 8¶
2.2.3 . Create new modules . 8¶
2.2.4 . Setup a user – screens yet to be created . 10¶
2.2.5 . [JL, Mandatory] Setup defaults – Jim and Karen will develop screens . 10¶
2.2.6 . [JL, Mandatory] Setup automatic reporting – Jim and Karen will develop screens . 10¶
2.2.7 . [JL, Mandatory] Set up benchmark screens - Jim and Karen will develop screens . 10¶
2.2.8 . [JL, Mandatory] Set up audit capabilities/defect tracking - Jim and Karen will develop screen . 10¶
2.3 . Enter-Edit Data . 10¶
2.3.1 . Select indicator . 10¶
2.3.2 . Enter/View Monthly Data . 11¶
2.3.3 . Enter/View Case Data . 11¶
2.3.4 . Viewing tables, run charts, control charts, benchmark charts . 12¶
2.3.5 . Miscellaneous . 12¶
2.4 . Decision Support . 12¶
2.4.1 . Messages . 13¶
2.4.2 . Alert types . 14¶
2.4.3 . Alert setup . 14¶
2.4.4 . Miscellaneous . 15¶
2.5 . Generate Reports and Charts . 15¶
2.5.1 . Selection of report and general formatting . 15¶
2.5.2 . Standard Reports . 15¶
2.5.3 . Custom Reports . Error! Bookmark not defined.¶
2.5.4 . View saved templates/reports . 16¶
2.6 . Display Reports and Charts . 17¶
2.6.1 . Reports . 17¶
2.6.2 . Control Charts . 19¶
2.6.3 . Benchmark Charts . 20¶
2.6.4 . Miscellaneous . 20¶
2.7 . General Screen Formatting and Actions . 21¶
2.8 . Export/Import Features ** . 23¶
3 . Product Requirements . 24¶
3.1.1 . Database . 24¶
3.1.2 . Portability . 24¶
3.1.3 . Performance . 24¶
3.1.4 . Safety . 25¶
3.1.5 . Privacy . 25¶

1 Introduction

1.1 PURPOSE

The purpose of this SRS (Software Requirements Specification) document is to provide a detailed description of our software project, specifications, and goals. This document describes the features included that are included in our project, its user interface, and its functionality. The document describes how our team sees and understands the requirements of the software.

1.2 DEFINITIONS, ACRONYMS, ABBREVIATIONS

Term	Description
ATM	ATM is an acronym for 'automated teller machine'
C-Requirements	Customer-requirements – Requirements specified and easily understood by the customer
D-requirements	Developer-requirements – A detailed description of the product requirements as supplied to the developer
Data-flow	A diagram that shows the flow of data between processing elements
GBE	GBE is an acronym for our clients title "Greatest Bank Ever"
GUI	GUI is an acronym for 'Graphical User Interface'
SRS	SRS is an acronym for 'software requirements specification'
State-Transition	A diagram that shows the flow of the program from one status to another
Use-case	A diagram used to describe the way a customer interacts with the system

1.3 REFERENCES

Software Engineering –An Object-Oriented Perspective by Eric J. Braude

1.4 OVERVIEW

[OMIT]

2 Overall Description

The application is to be fully-functional bank software. It will consist of a few different modules. The first module is to be a software application that can be used by bank customers as in an ATM. This application will allow the user to deposit funds, withdraw funds, check balance, and transfer funds.

The second module is to be a software application that is used within the bank by the bank tellers and managers. This application will allow everything the ATM allows as well as some additional features. These features include: account creation, account deletion, loan approval, customer records, and reports.

Both pieces of software will be linked to a central bank server. This server will handle multiple threads and will therefore allow for simultaneous access of multiple users. It will provide for user authentication and will store all data.

2.1 PRODUCT PERSPECTIVE

The product is banking software similar in functionality to the software run in most banks. The reporting feature is also similar to the reports available in Intuit QuickBooks and Quicken.

2.1.1 Concept of Operations

The bank software will allow for two different methods of operation. The first method of operation will be through an ATM terminal. This operation is performed by the user. The user will be allowed to deposit funds, withdraw funds, check balance, and transfer funds. It will all be done through a simple, easy to use graphic user interface. It will implement the standard buttons found on a standard ATM machine.

The second part of the banking software is the bank operations software. It will run through a PC and will be manipulated by a bank teller or manager via keyboard and mouse. This will allow for a

different type of GUI. Instead of a series of consecutive screens operated by ATM buttons this software will all be organized around a menu bar. All functions will be accessible at any time through a menu bar at the top of the screen. These functions include: authenticating, depositing funds, withdrawing funds, checking balance, requesting account creation, authorizing account creation, requesting loans, authorizing loans, and generating reports. The users will be classified into two different categories: Bank managers and bank officers. This classification determines what access privileges the user has.

2.1.2 Major User Interfaces

A screen-flow diagram shows the flow of screens as the user interacts with the system. For the ATM screen-flow diagram please see Appendix A. For the teller software please see appendix C.

2.1.2.1 Example Screenshot and description

Screenshots are provided to demonstrate what screens the user will see. Rough diagrams of the ATM screens can be seen in Appendix B. Rough diagrams for the teller software can be seen in Appendix D.

2.1.3 Hardware Interfaces

The ATM system will operate using the standard hardware available in an ATM. This includes a screen, card reader, receipt printer, keypad, and a few simple input buttons.

The Teller software will operate using the standard I/O hardware available in a desktop PC. Hardware will run through the keyboard, mouse, monitor, and printer.

2.1.4 Software Interfaces

// example: CGI-URL or function signatures etc (OMIT for now).

2.1.5 Communication Interfaces

// example: modem etc (OMIT for now)

2.1.6 Memory Constraints

// RAM, and other storage constraints (OMIT for now)

2.1.7 Operations

// special operations (if any) (OMIT for now)

2.1.8 Site Adaptation Requirements

//ex: Japanese language etc (OMIT for now)

2.2 PRODUCT FUNCTIONS

// First draw the overall use-case diagram and give a brief description

// then – describe the major use-case

2.2.1 Deposit Money (Bank Customer- ATM System)

Description

The Deposit Money use-case for the customer is for the ATM System. Its primary function is to allow the user to deposit money into an account by means of the ATM System.

Actors

1. Bank Customer
2. ATM System

Main Flow

1. The user is presented with the main menu screen.
2. The user will request deposit option from the ATM by clicking on the deposit button of the main menu.
3. The user is presented with a screen that prompts the user for the account to where the money will be deposited and the amount of money to deposit.
4. The user enters the correct information and then clicks on the confirm button.
5. The ATM system sets updateBalanceNeeded on the users account.
6. The ATM system will prompt the user for another transaction. If the user input is yes then the ATM system will reset to main menu screen. If the user input is No then the system will logout the user and return to the login screen.

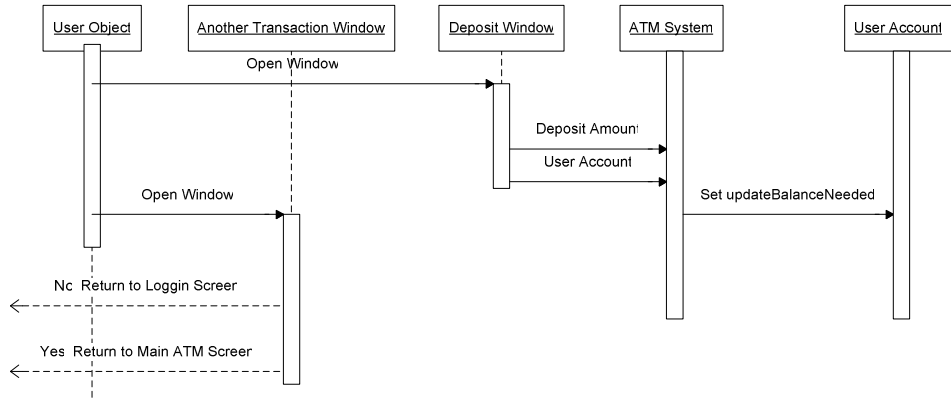
Pre-Conditions

1. The user has been authenticated properly.

Post-Conditions

1. UpdateBalanceNeeded flag has been set for the appropriate account.

Deposit Sequence Diagram



2.2.2 Transfer Funds (Bank Customer – ATM system)

Description

The Transfer Funds use-case for the customer is for the ATM System. Its primary function is to allow the user to transfer funds from one account to another.

Actors

1. Bank Customer
2. ATM System

Main Flow

1. The user is presented with the main menu screen.
2. The user requests transfer funds from ATM system main menu screen.
3. The user selects the accounts for outgoing funds and incoming funds.
4. The user enters the dollar amount of funds to transfer.
5. The ATM system queries the bank server to validate fund transfers vs account balances.
6. The ATM System queries the bank server to complete funds transfer.
7. The user is prompted for another transaction. If the user input is Yes then go to main menu screen. If the user input is No then go to login screen.

Alternate Flow

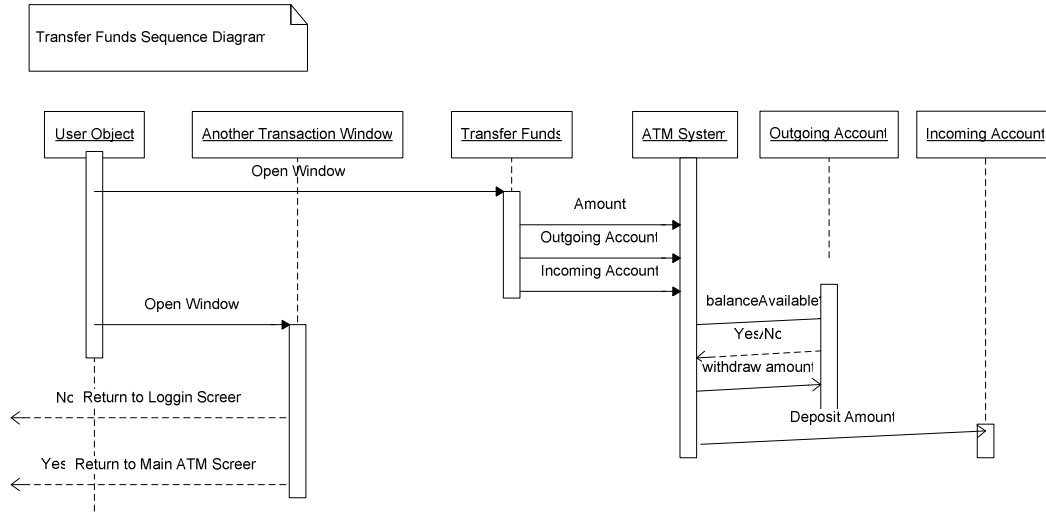
1. If the user doesn't have the necessary funds to complete a funds transfer then the user is notified.
2. The user is prompted for another transaction. If the user input is Yes then go to main menu screen. If the user input is No then go to login screen.

Pre-Condition

1. The user has been authenticated properly.

Post-Condition

1. The funds are transferred from the outgoing account into the incoming account.



2.2.3 Authenticate (Bank Customer-ATM System)

Description

The Authenticate use-case for the customer is for the ATM System. Its primary function is to ensure that the user is accessing their information and account and to ensure that no unauthorized user access the intended user's account.

Actors

1. Bank Customer
2. ATM System

Main Flow

1. System is at the welcome screen when user comes to the Authenticate Use-Case. The welcome screen has a sign in prompt.
2. The welcome screen will prompt the user for his / her account number and password.
3. The user enters his / her information
4. The information is validated against the bank database.
5. If the validation process returns true, then the user is granted access to the ATM System.

Alternate Flows

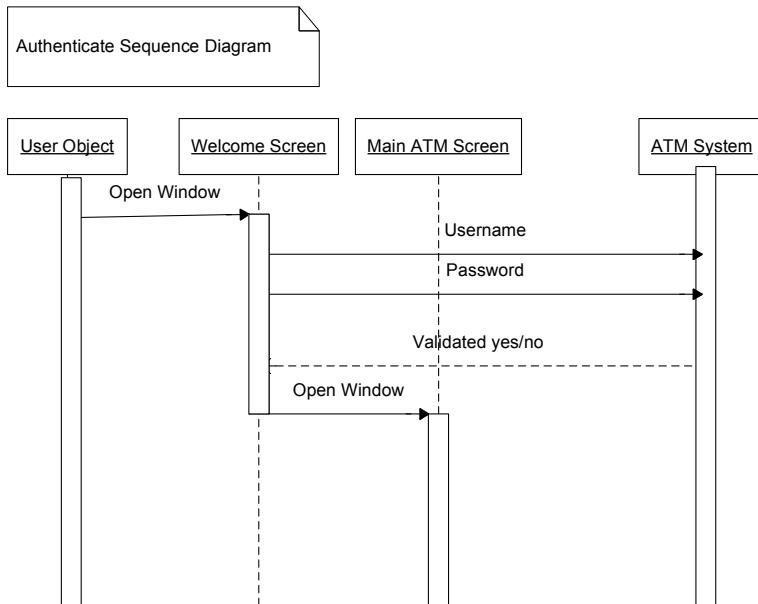
1. If the validation returns false, then the user is informed and the ATM returns to the welcome screen.
2. If the user enters their information incorrectly three times, then their account is locked and the Bank Officer must unlock the account before the user can use the ATM again.
3. Any time the user tries to access their account before it is unlocked they will receive a message telling them that their account is still locked.

Pre-conditions

1. The ATM System is at the welcome screen when the user approaches the ATM.

Post-condition

1. If the user is validated, the ATM System is logged in and ready for the user to access their account.
2. If the user was not validated, then the ATM System is at the welcome screen waiting for a new customer.



2.2.4 Withdraw Money (Bank Customer-ATM System)

Description

The Withdraw Money use-case is on the ATM System. Its primary function is to allow the user, once logged in, to remove money from the account of their choosing.

Actors

1. Bank Customer
2. ATM System

Main Flow

1. The user selects the *Withdraw* option from the main ATM Screen.
2. The user selects from a list of his / her accounts which one to withdraw from.
3. The user specifies in the amount box how much he / she wants to withdraw.
4. The ATM System verifies that the amount is in increments of \$10.
5. The ATM System verifies that the user has the amount available in their account.
6. The balance is subtracted from the user's account, and they are deposited the cash.
7. The *Another Transaction* screen will appear and the user will select yes to return to the main ATM Screen or no to exit the ATM System. The user will be logged out automatically.

Alternate Flows

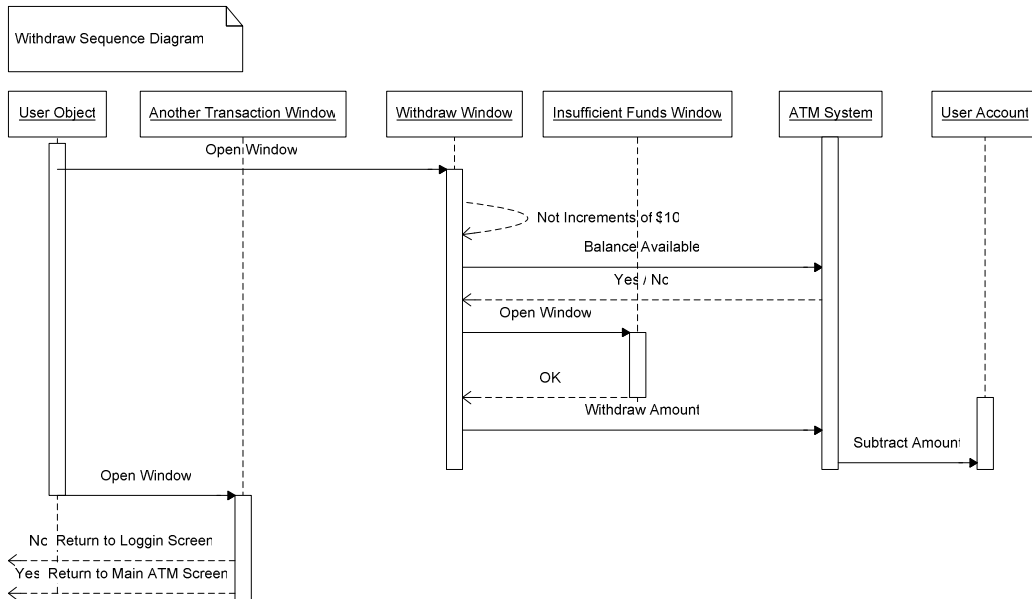
1. If the user does not enter an amount in increments of \$10 then the user will be informed that the amount must be in increments of \$10. The *Withdraw* screen will appear again for the user to retry their withdrawal.
2. If the desired balance is not available in the user's account then the *Insufficient Funds* window will appear to inform the user of the problem. The user can select *OK* and the system will return to the *Withdraw* screen.

Pre-conditions

1. The user has logged into their account.
2. The user is at the main ATM Screen.

Post-condition

1. The user has withdrawn the money from their account if they entered the correct amount and if their account had that amount in it.
2. The user sees the main ATM Screen if they selected to do another transaction, or the ATM System has logged the user out and the user sees the welcome screen.



2.2.5 Check Balance (Bank Customer-ATM System)

Description

The Check Balance use-case is on the ATM System. The primary purpose for this function is to allow the user to check the balance of their account at any ATM without having to go to the bank.

Actors

1. Bank Customer
2. ATM System

Main Flow

1. The user selects the *Check Balance* option from the main ATM Screen.
2. The user enters the account that they would like to know the balance of.
3. The ATM System runs the print function with the balance and account information on the slip.
4. The *Another Transaction* screen will appear and the user will select yes to return to the main ATM Screen or no to exit the ATM System. The user will be logged out automatically.

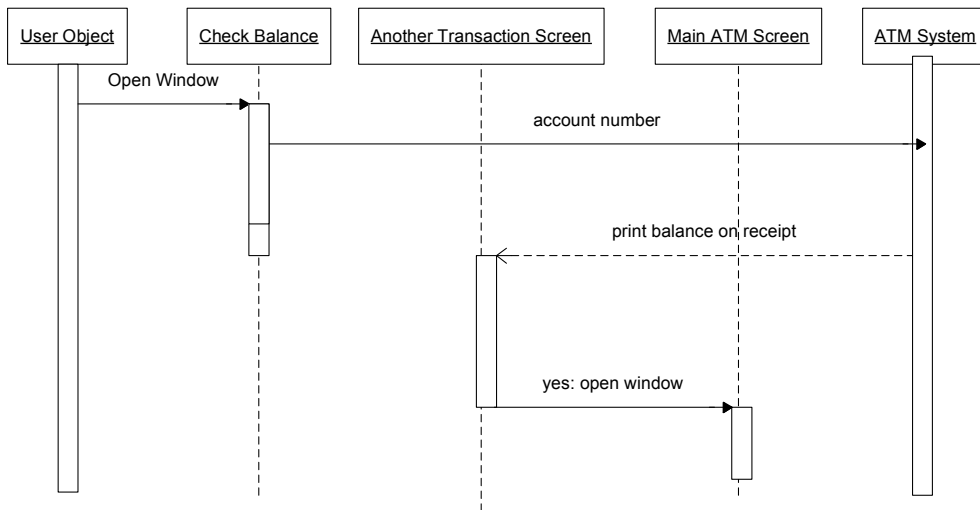
Pre-conditions

1. The user has logged into their account.
2. The user is at the main ATM Screen.

Post-condition

1. The user has a printed copy of his / her account's balance in his / her hand.
2. The user has logged of if he / she selected no to another transaction and is looking at the Welcome screen, or he / she is at the main ATM Screen if he / she selected yes to another transaction.

Check Ballance Sequence Diagram



2.2.6 Request Loans (Bank Officer-Bank Officer System)

Description

The Request Loans use-case is on the Bank Officer System. The primary purpose for this function is to allow the bank officer to request a loan on behalf of the bank customer.

Actors

1. Bank Officer
2. Bank Officer System

Main Flow

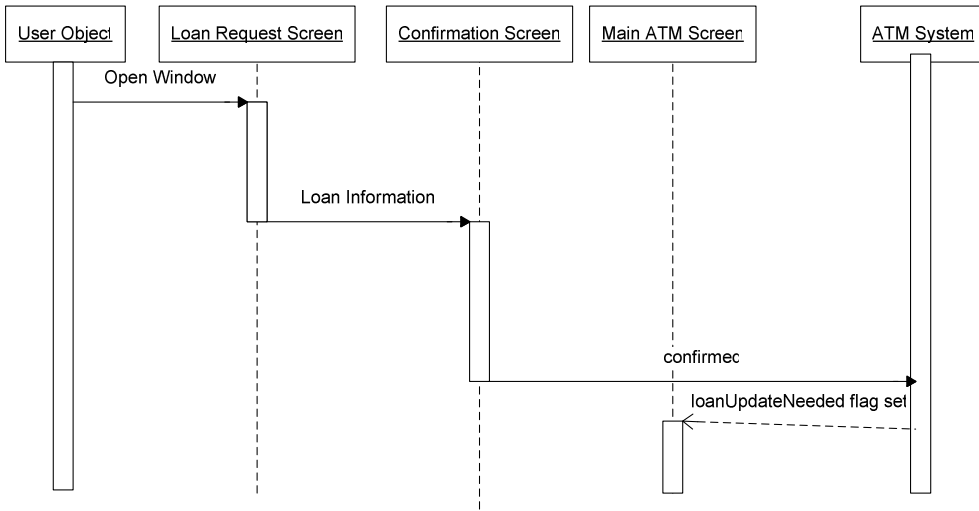
1. The user selects request loans option from main menu screen.
2. The user enters the loan information.
3. The user confirms the transaction.
4. The Bank Officer Systems sets flag loanUpdateNeeded on users account.
5. The Bank Officer System returns to main menu screen.

Pre-Conditions

1. The bank customer has a valid bank account.

Post-Conditions

1. The flag loanUpdateNeeded is set on the users account.



2.2.7 Request Opening/Closing of a Bank Customer Account (Bank Officer-Bank Officer System)

Description

The Request Opening/Closing of a Bank Customer Account use-case is on the Bank Officer System. The primary purpose for this function is to allow the bank officer to request the opening or closing of a bank customer account.

Actors

1. Bank Officer
2. Bank Officer System

Main Flow

1. The user selects the option for opening or closing from the main menu screen.
2. The user enters the customer information. If opening account enter new information. If closing account enter customer information.
3. The user confirms the action of either opening or closing the account.
4. The database is updated to reflect the new change.
5. The Bank Officer System returns to main menu screen.

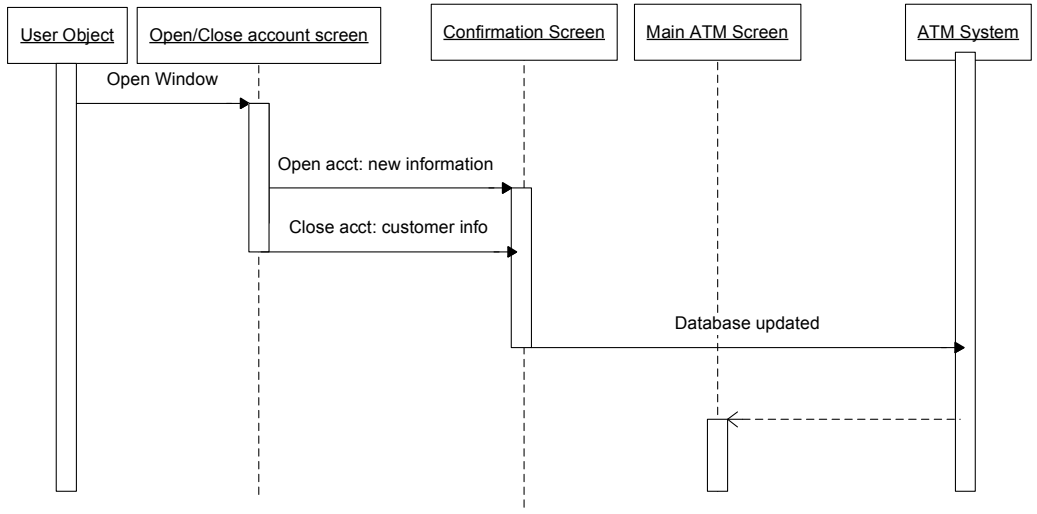
Pre-Condition

1. If closing an account the bank customer has a valid bank account to close.

Post-Condition

1. The database is updated to reflect the new changes.

Request Opening / Closing of a Bank Customer Account



2.2.8 Print Bank Reports (Bank Officer-Bank Officer System)

Description

The Print Bank Reports use-case is on the Bank Officer System. The primary purpose for this function is to allow the bank officer to request the opening or closing of a bank customer account.

Actors

1. Bank Officer
2. Bank Officer System

Main Flow

1. The user selects Bank Reports option from the main menu screen
2. Select print button
3. Confirm action
4. Print bank report
5. Bank Officer System returns to main menu screen

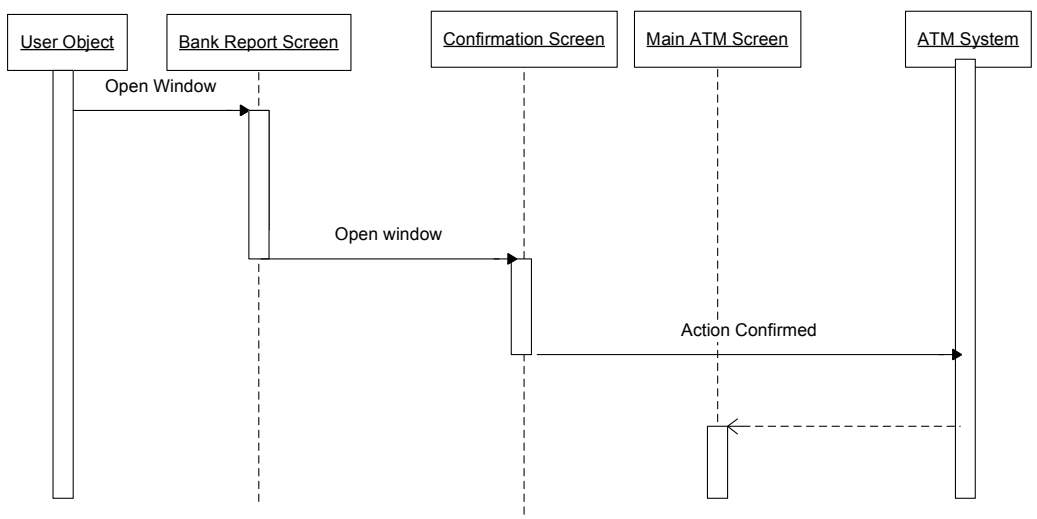
Pre-Condition

1. The bank officer is logged on to the Bank Officer System.

Post-Condition

2. The appropriate bank report was printed.

Print Report Sequence Diagram



2.3 USER CHARACTERISTICS

The typical bank customer will be a person, from the age of 10 and up. There will more than likely be a fairly equal distribution of males and females. The typical customer will probably use the ATM a couple of times a week. The typical customer might not know anything about computers, so their system needs to be very simple and easy to use. The typical customer will probably be a busy person; therefore, they will need to do their transactions as quickly and efficiently as possible.

The other user is a bank employee. The bank employee will be a different type of user. The bank employee is a fairly educated user, who is willing to sacrifice simplicity for functionality. They will use the software daily, for every transaction. This could quite possibly be 30-60 transactions per hour per employee. Due to this frequency of usage stability and speed of this software is incredibly important.

2.4 CONSTRAINTS

// all conditions that may limit design options

2.5 ASSUMPTIONS AND DEPENDENCIES

// hardware and software assumptions and dependencies

3 Specific Requirements

// Here you need to put in details (if any). Mark items [None] if you do not have any information.

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

3.1.1.1 Server

This server will have a very simple set of commands. It will run in a console. Upon initial server initialization the server will prompt the user for the port number and the name of the bank. The server will run until the user input a 'Q' to quit. When the application quits all the objects are saved. The server application will load the account and customer objects next time it boots if the same bank name is used. The server shall spawn separate threads for each incoming request.

3.1.1.2 ATM Machine

The ATM machine shall have a graphic user interface. There will be a series of screens. The first screen the user shall see is the login screen. This login screen will prompt the user for a username and password. This username and password will be sent to the server for authentication. At this point, the system routes to the correct bank server.

After the login is complete the user shall have the option to withdraw, deposit, transfer, or check balance. If the user chooses to withdraw funds they will be prompted to enter the amount to withdraw and the account they want to withdraw from. After the user enters this information the request is sent to the server. The server will process this request and either accept or deny it. If approved the server then processes the transaction and returns to the ATM system the information needed to print a receipt.

The deposit function is to work in a similar fashion. The user chooses what account to deposit to and the amount to deposit. The request is then sent to the server to either be accepted or denied. If approved the server processed the transaction and returns to the ATM system the information needed to print a receipt.

The check balance function also works similarly. The user chooses which account to get the balance of and then a request is sent to the server. The server returns to the ATM system the account balance.

Upon completion of the transaction the user will be prompted to perform another transaction. If they choose yes the process repeats. The ATM software shall also display errors to the user if there are insufficient funds or other problems. There will also be a help available to walk the user through the operating procedure of the bank system.

1. Teller Software

The teller software shall initialize with a login prompt. The user enters their username and password. This information shall be authenticated through the server. An error message shall be displayed if an incorrect username/password combination is entered.

The next screen will be a main screen. It will have a toolbar through which all operations are performed: File, Customers, Accounts, Loans, Reports, Manage, and Help

When the user clicks file they will have the option to logout or quit.

When the user chooses Customers from the toolbar they will have the option to: create new customer, find customer, and delete customer. The new user choice will open the new user screen. This screen will require the user to input a name, address, phone, etc of the user. An error message shall be displayed if any fields are left empty.

When the user confirms this information the client shall transmit it to the server. If the user chooses find user from the toolbar a find user screen appears. From this screen the software user can search customers by customer number or customer name. An error message will appear if the customer number or name is invalid/not found from the server. When the user is found the client retrieves the customer information from the server. This information includes a list of accounts, balances, and transactions. The bank manager shall be able to void transactions through this screen as well.

The user shall be able to delete a customer through the customer toolbar option. To do this the user inputs the customer number. If the user does not know the customer number they shall be able to look it up. Once a customer number is entered the client application requests customer information from the server. The user can then click delete or cancel. When delete is clicked the client sends a request to the server to delete the customer.

Under the accounts option in the toolbar will be buttons to a deposit, withdraw, new account request, close account request, authorize close account, authorize new account, and get balance.

When the user selects deposit they are prompted to input an account number. There shall be a find account number option to help if the account number is unknown. Once the account number is found the user inputs the deposit amount. This is then sent to the server for processing.

Similarly to the deposit screen, when the user selects deposit they are prompted to input the account number and the amount to withdraw. This information is sent to the server in the form of a request. If the server approves the user is returned to the main screen. If the server declines the withdraw the user will be shown an error.

The 'request new account' function allows a teller to request the creation of an account. The customer number is required as is the type of account and interest rate. If the user does not know the customer number they can discover it by clicking on the 'find user number button'. Once the required information is obtained a request is sent to the sever. All theses requests wait on a queue until the bank manager approves them.

The 'request account close' option works similarly to the 'request new account'. The user inputs in the account number and then confirms it with the server. There is an option to find account number if it is needed. The requests are all sent to a server queue until the bank manager approves.

The bank manager is required to authorize account creations or closings. They do this through a window that lists all available closings/openings. The user clicks which one to authorize. This request is sent to the server and recorded.

The 'get balance' function allows the user to get the balance of an account. It requires the user to input the account number. If the account number isn't known the user can look it up. Once the account number is entered a request is sent to the server for the balance. The server then returns the balance which is displayed onscreen.

The 'request new loan' and 'authorize loan' options function in the same way as the request and authorize new accounts options. The 'accept payment' option functions identically to the deposit function.

The user can then click on reports to find customer statements, or other bank accounting information. When 'customer reports' is selected the bank manager is shown a list of all the customers with an account balance that month. The user can then view the statements to be printed individually or print them all as one batch.

Under the Manage tab of the toolbar there are two options: new user and manage user. When 'new user' is chosen the bank manager can assign a username and password to the user. The manager also inputs the real and the role of the user. Once this information is entered the request is sent to the server to create the user.

When a manger selects the Manage user option they are shown the manage user screen. This requires the employee's username to be entered. This

username can be discovered through a 'find username' button. From this screen the manager has the option to change bank roles, change passwords, or delete user. Once the bank manager makes one of these selections the request is fulfilled by the server.

The last option on the toolbar is a help option. Under the help option the user can display a help file that contains help or click about. The about option tells about the software developers, copy writes, etc.

3.1.2 Hardware Interfaces

[None]

3.1.3 Software Interfaces

[None]

3.1.4 Communications Interfaces

The client and server will be communicating through the internet via a Transmission Control Protocol of the TCP/IP Suite. This protocol is particularly suited for this application because it is a connection oriented protocol that allows for an ordered and reliable delivery of packets. The use of a non connection oriented protocol would not be well suited for this application due to the complications of dropped packets and unordered delivery of packets.

3.2 FEATURES

3.2.1 Server Requirements

3.2.1.1 [Mandatory] The server shall Load data at startup

3.2.1.2 [Mandatory] The server shall save data at shutdown

3.2.1.3 [Preferred] The server shall save account and user data anytime it is modified (i.e. after a deposit, withdraw, creation of account, etc.)

3.2.1.4 [Mandatory] The server shall handle requests in a separate thread

3.2.1.5 [Mandatory] The server shall lock accounts/user to prevent errors due to simultaneous access

3.2.1.6 [Mandatory] The server shall properly handle client requests

3.2.1.7 [Mandatory] The server shall handle simultaneous access to join accounts

3.2.2 Client Requirements

3.2.2.1 [Mandatory] The client shall require user authentication in order to perform any operations

3.2.2.2 [Mandatory] The client shall send requests to the server to complete basic bank operations (withdraws, deposits, transfers, and balance checks)

3.2.2.3 [Mandatory] The client shall send requests to the server to complete user/account related operations (create/remove user, create/remove account)

3.2.2.4 [Optional] The client shall allow user passwords to be changed

3.2.2.5 [Mandatory] The client shall be able to print monthly/yearly reports of accounts

3.2.3 ATM Requirements

3.2.3.1 [Mandatory] The ATM shall present an error message if the user enters invalid login information and allow the user to attempt a login again

3.2.3.2 [Mandatory] The ATM shall allow customers to deposit/withdraw/transfer money and check balances of accounts

3.3 PERFORMANCE REQUIREMENTS

3.3.1 [Mandatory] The system shall run with multiple threads.

3.3.2 [Mandatory] The system shall be free of deadlock relating to the multiple threads.

3.3.3 [Preferred] The system shall be able to support ten users with no user experiencing more than a one second lag.

3.4 DESIGN CONSTRAINTS

The GBE Bank Software is to be implemented in Java.

The GBE Bank Software documentation shall use IEEE standard 830-1993.

Short timeline for project

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 Reliability

3.5.1.1 [Mandatory] The system shall save the data stored in the user data structures once every five minutes.

3.5.1.2 [Mandatory] The system shall save the data of an active user whenever that user has just completed a deposit transaction.

3.5.1.3 [Mandatory] The system shall save the data of an active user whenever that user has just completed a withdraw transaction.

3.5.1.4 [Mandatory] The system shall save the data of an active user whenever that user has just completed a transfer funds transaction.

3.5.1.5 [Mandatory] The system shall save the data of an active user whenever a new user is created.

3.5.1.6 [Mandatory] The system shall save the data of an active user whenever a new account is created.

3.5.1.7 [Mandatory] The system shall save the data of an active user whenever a user password is changed.

3.5.1.8 [Preferred] The system shall not save on a check balance transaction.

3.5.1.9 [Mandatory] The system shall reboot itself automatically if there is a power failure and restore all saved data to the user accounts.

3.5.2 Availability

The ATM system shall be available and running in a stable state at all times

The Bank Server shall be available and running in a stable state at all times.

The Bank Officer Software shall be available at all times.

3.5.3 Security

3.5.3.1 [Mandatory] The system shall require a password to allow a user to logon.

3.5.3.2 [Mandatory] The system shall require the user to know their own password and account number.

3.5.4 Maintainability

3.5.4.1 [Mandatory] The source code shall be available to the developers.

3.5.5 Portability

[None]

3.6 OTHER REQUIREMENTS

[NONE]

Appendix A: ATM Screen-flow Diagram

Appendix B: ATM Screenshots

Appendix D: Teller Screenshots