**‘Use Case Document**

Group 3 - Banking

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 2/14/25 | 1.0 | We wrote UC-001-3 | Team (no sophia :() |
| 2/24/25 | 1.5 | Added Use Cases for ATM transactions | Matthew Zhang |
| 2/24/25 | 1.8 | Started Use Cases 007 – 012 | Group Work Meeting |
| 2/25/25 | 2.0 | Finished Use Case 007-015 | Group Work (No Phakin) |
| 2/25/25 | 2.2 | Completed UCs: 007-009 and 011-013 | Matthew Zhang |
| 2/27/25 | 2.3 | Edited UC-013 and added UC-014 | Matthew Zhang |
| 03/05/2025 | 2.5 | Changed Wording and finished UC-004-006 and UC-016 | Ruba Hagog |
| 4/9/2025 | 2.7 | Added use cases UC-020, UC-021, UC-022 | Alexandra Molchanenko |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Use Case ID:** UC-001

**Use Case Name:** Login

**Relevant Requirements:** See SRS 3.1

**Primary Actor:** Employees and Clients

**Pre-conditions:** Establish a connection with the server.

**Post-conditions:** User gains access to the Employee System. {This describes the state of the system following the successful completion of this use. Effects on other systems and actors may also be described.}

**Basic Flow or Main Scenario:**

1. Establish connection with the server
2. System prompt user to log in as client or employee
3. User initiates an action by choosing to log in as an employee.
4. System responds by prompting the user to input login information.
5. User responds by inputting login information
6. System responds by communicating login information to the server.
7. Server responds by validating user information
8. If the user's login information is valid, the server communicated the validation with the system
9. The system responds by granting the user access to the employee interface.

**Extensions or Alternate Flows:**

1. Alternative Flow - Client Login
   1. User initiates an action by choosing to login as a client.
   2. System responds by prompting the user to input login information.
   3. User responds by inputting login information
   4. System responds by communicating login information to the server.
   5. Server responds by validating user information.
   6. If the user’s information is valid, the server communicates validation to the system.
   7. The system responds by granting user access to the client interface.
2. Alternative Flow - Failed Login
   1. If the user’s login information is invalid, the system responds by displaying a login error message.
   2. The user responds by attempting to login again.

**Exceptions:**

1. Users may not attempt login more than 3 times after the first failed attempt; so as to avoid server overload.
2. Users may close or lose connection to the server before entering login information.

**Related Use Cases:** UC-002. UC-003

**Use Case ID:** UC-002.

**Use Case Name:** View Profile - Client

**Relevant Requirements:** See SRS 3.1

**Primary Actor:** Client.

**Pre-conditions:** UC-001 is executed successfully.

**Post-conditions:** Clients can view all their different accounts, name and information

**Basic Flow or Main Scenario:**

The system redirects users to a profile page. The user is able to see all their accounts and is able to click on the different accounts. The user can also see their name and their information.

**Extensions or Alternate Flows:**

UC-004, UC-001.

**Exception:**

1. User loses connection or closes the device.

**Related Use Cases:** UC-004

**Use Case ID:** UC-003.

**Use Case Name:** View Profile - Teller

**Relevant Requirements:** See SRS 3.1

**Primary Actor:** Teller.

**Pre-conditions:** UC-001 is executed successfully.

**Post-conditions:** The Teller is able to view their name and bank location information and scroll through different employee functions.

**Basic Flow or Main Scenario:**

The system redirects the employee to their profile page. The employee is able to see their name and the bank location as well as a find client function.

**Extensions or Alternate Flows:**

UC-005.

**Exceptions:**

* + - 1. Employee closes the device or loses connection.

**Related Use Cases:** UC-003

**Use Case ID**: UC-004

**Use Case Name**: View Account-Client

**Primary Actor:** Client.

**Relevant Requirements:** See SRS 3.1.2

**Precondition:** the Client has logged in and has access to their client profile.

**Postcondition:** the client can view the account type, account balance, credit available and credit balance.

**Basic Flow or Main Scenario:**

**Use Case ID**: UC-005

**Use Case Name**: Find Client.

**Relevant** **Requirement**: See SRS

**Use Case ID**: UC-004

**Use Case Name**: View Account-ATM

**Relevant Requirements:** See SRS

**Primary Actor:** Client/User

**Use Case ID:** UC-005

**Use Case Name:** Deposit Cash - ATM

**Relevant Requirements:** See SRS

**Primary Actor:** Client/User

**Pre-conditions:** Client has logged into a bank account on an ATM using login credentials.

**Post-conditions:** Client has deposited a set amount of money into a bank account and is returned to the main ATM interface.

**Basic Flow or Main Scenario:**

1. ATM prompts user to deposit cash
2. User deposits set number of cash
3. ATM brings user back to main ATM menu

**Extensions or Alternate Flows:**

1. User exits without depositing
   1. Return to main menu

**Exceptions:**

1. ATM loses connection with user (either from time out or error)

**Related Use Cases:**

**Use Case ID:** UC-006

**Use Case Name:** Withdraw Cash - ATM

**Relevant Requirements:** See SRS

**Primary Actor:** Client/User

**Pre-conditions:** Client has logged into a bank account on an ATM using login credentials.

**Post-conditions:** Client has withdrawn a set amount of money into a bank account and is returned to the main ATM interface.

**Basic Flow or Main Scenario:**

1. ATM prompts user amount of cash to withdraw
2. User withdraws set number of cash
3. ATM brings user back to main ATM menu

**Extensions or Alternate Flows:**

1. User exits without withdrawing anything
2. Return back to main menu
3. User has insufficient funds to withdraw
   1. Presents warning to user and continues to prompt for withdrawal amount

**Exceptions:**

1. ATM loses connection with user (either from time out or error)

**Related Use Cases:**

**Use Case ID:** UC-007

**Use Case Name:** Find Client Account - Teller

**Relevant Requirements:** See SRS

**Primary Actor**: Teller

**Pre-conditions:** Teller is on the Teller interface and selects an option to find a client’s account.

**Post-conditions:** Teller brought to a menu that displays client’s bank information and a selection of actions that could be performed on the account.

**Basic Flow or Main Scenario:**

1. Teller on the main Teller interface and selects option to find client’s account
2. System asks for Teller to input client’s ID
3. After inputting ID, system will locate the corresponding client bank account linked to ID and send that client’s bank information to the teller’s server
4. Teller brought to another interface which displays client’s bank information and several options that could be performed on client’s bank account

**Extensions or Alternate Flows:**

1. An account does not exist with that client ID
   * 1. Teller prompted with message that the account does not exist and is brought back to main Teller interface

**Exceptions:**

**Related Use Cases:** UC-003

**Use Case ID**: UC-008

**Use Case Name**: View Client Account - Teller

**Relevant Requirements:** See SRS 3.1.2 and 3.1.4

**Primary Actor:** Teller/User

**Pre-conditions:** Teller has selected a function to find a client account and input valid information.

**Post-conditions:** Tellers will be able to perform a number of actions on a client's bank account.

**Basic Flow or Main Scenario:**

1. Teller selects client bank account using client’s ID
2. Teller server sends request to central server to transmit the bank details corresponding to client’s ID
3. Central server sends client’s bank details to Teller server
4. Teller is able to view client’s bank account information and several actions in a menu

**Extensions or Alternate Flows:**

1. Teller is presented with several actions that could be performed on the View Client menu
   * 1. Withdraw an amount of cash
     2. Deposit an amount of cash
     3. Delete account
     4. Change account information
     5. Quit and return to the Teller interface

**Exceptions:**

1. Client account does not exist in which no menu would appear and the Teller would return to main Teller interface

**Related Use Cases:** UC-007

**Use Case ID:** UC-009 (M.Z)

**Use Case Name:** Withdraw Cash - Teller

**Relevant Requirements:** See SRS 3.1.2 and 3.1.4

**Primary Actor:** Teller/User

**Pre-conditions:** Teller is on the menu which displays a client’s account and selected option to withdraw cash

**Post-conditions:** A set amount of money has been withdrawn from the client’s bank account

**Basic Flow or Main Scenario:**

1. Teller selects function to withdraw money from client’s bank account
2. System prompts Teller to input amount of money to be withdrawn
3. Teller inputs amount of money to be withdrawn from account
4. System records that input, checks to see if it is possible to withdraw, then withdraws that amount of money
5. This is recorded and immediately stored in a log
6. Account balance is updated
7. Teller is returned to the menu that views client bank information and list of functions that can be performed.

**Extensions or Alternate Flows:**

1. Client changes mind and does not want Teller to withdraw anything from bank account
2. Teller exits from system prompt and returns to View Client Account menu

**Exceptions:**

1. The withdrawal amount exceeds the amount of money inside the bank account
   * 1. System presents error message to Teller and transaction does not go through
     2. System returns to prompting for amount of money to withdraw

**Related Use Cases:** UC-008

**Use Case ID:** UC-010 (R.H)

**Use Case Name:** Deposit Cash - Teller

**Relevant Requirements:** See SRS 3.1.2 and 3.1.4

**Primary Actor**: Teller/User

**Pre-conditions:** Teller has logged into as a Teller of the bank and created a client profile.

**Post-conditions:** Teller deposits money into the Client’s intended bank account.

**Basic Flow or Main Scenario:**

1. Teller prompts the system by clicking on the intended bank account.
2. System responds by shifting the teller to the account page.
3. Teller prompts the system by choosing to deposit money.
4. The system responds prompting the teller to input the deposit amount.
5. Teller responds by inputting the amount and pressing enter.
6. The system responds by displaying a message asking the teller if the amount if correct
7. The teller responds by accepting the amount.
8. System responds by updating the account balance.

**Extensions or Alternate Flows:**

1. The Teller responds to the system by rejecting the amount inputting.
2. The system responds by taking the teller back to the deposit page.
3. The teller responds by inputting the amount again.
4. System responds by displaying a message asking if the amount inputting is correct
5. The teller responds by accepting
6. The system responds by updating the account balance.

**Exceptions:**

**Related Use Cases:** UC-008

**Use Case ID:** UC-011 (M.Z)

**Use Case Name:** Close account - Teller

**Relevant Requirements:** See SRS 3.1.2 and SRS 3.1.4

**Primary Actor:** Teller

**Pre-conditions:** Teller is viewing client bank information and selects option to close client’s account

**Post-conditions:** Client’s account is deleted, any remaining funds is removed, and client will not be able to access account again

**Basic Flow or Main Scenario:**

1. Teller selects option to close client’s bank account
2. System prompts Teller with confirmation page
3. Once Teller confirms deletion, client’s account will closed
4. Teller and client will not be able to access the account anymore

**Extensions or Alternate Flows:**

1. Client changes mind before Teller confirms account closure
   * 1. Teller exits from account closure page and returns to view of client’s bank account details and list of functions that can be performed

**Exceptions:**

**Related Use Cases:** UC-008

**Use Case ID:** UC-012 (M.Z)

**Use Case Name:** Edit Client’s Bank Information - Teller

**Relevant Requirements:** See SRS 3.1.2 and SRS 3.1.4

**Primary Actor:** Teller

**Pre-conditions:** Teller is viewing client bank information and selects option to edit the bank information client’s account

**Post-conditions:** Client’s account has its bank information changed.

**Basic Flow or Main Scenario:**

1. Teller selects option to edit client’s bank information
2. System prompts Teller with list of bank information they can change:
   * 1. Username
     2. Password
     3. Address\*
     4. Phone\*
     5. Legal name\*
3. System makes changes to the client’s bank account information and saves it to the central server
4. Teller returns to display of client’s bank account information and list of actions that could be performed

**Extensions or Alternate Flows:**

1. Client changes mind before Teller confirms account closure
2. Teller exits from edit account page and returns to view of client’s bank account details and list of functions that can be performed

**Exceptions:**

1. Invalid password or username
   1. Systems prompts Teller with an error message
   2. System will continue to prompt for valid password or username

**Related Use Cases:** UC-008

**Use Case ID:** UC-013 (MZ)

**Use Case Name:** Create New Client Profile

**Relevant Requirements:** See SRS 3.1.2 and 3.1.4

**Primary Actor**: Teller

**Pre-conditions:** Teller is on the main Teller interface and selects option to create new client account profile for the client

**Post-conditions:** A new client account profile is created and is added to server records

**Basic Flow or Main Scenario:**

1. Teller selects option to create new client account profile
2. System prompts Teller to add new username for account
3. System prompts Teller to add new password for account
4. System prompts Teller to add personal information of client
   * 1. Address, Phone number, and legal name
5. System gives Teller ID of account
6. System creates a new client account profile and saves that profile to the central server
7. Teller is notified of account profile creation and returned to main Teller interface

**Extensions or Alternate Flows:**

1. Client changes mind on account profile creation at any step before account creation
   * 1. Teller can exit from account profile creation menu back to main Teller interface

**Exceptions:**

1. Invalid password or username
   1. Systems prompts Teller with an error message
   2. System will continue to prompt for valid password or username
2. Invalid personal information (like phone number is not in a proper format)
   * 1. Systems prompts Teller with an error message
     2. System will continue to prompt for valid personal information

**Related Use Cases:** UC-003

**Use Case ID:** UC-014 (MZ)

**Use Case Name:** Create New Bank Account

**Relevant Requirements:** See SRS 3.1.2 and 3.1.4

**Primary Actor**: Teller

**Pre-conditions:** Teller is on the main Teller interface and selects option to create new client bank account for the client

**Post-conditions:** A new client bank account is created and is linked to the client’s profile. The bank account is recorded in the server storage.

**Basic Flow or Main Scenario:**

1. Teller selects option to create new bank account
2. System prompts Teller for the client’s profile ID, so system can connect new bank account to client’s profile
3. System prompts Teller for type of bank account:
   * 1. Checking
     2. Savings
     3. Line of credit
4. Teller chooses one of those bank accounts and confirms, allowing a new bank account to be created
5. System provides a unique ID for the bank account
6. Bank account saved in central server storage

**Extensions or Alternate Flows:**

1. Teller attempts to create a new line of credit for client when client already has a line of credit
   * 1. Error message appears as only one line of credit is allowed per client profile
2. Client changes mind before Teller confirms bank account creation.
   * 1. Teller can exit menu and return to main Teller interface

**Exceptions:**

1. Invalid profile ID
   * 1. Client does not have a profile in the bank’s system
     2. Cannot create new bank accounts until client profile and profile ID is created

**Related Use Cases:** UC-003

**Use Case ID:** UC-015 (P.S)

**Use Case Name:** Make a Payment- ATM

**Relevant Requirements:** See SRS

**Primary Actor**: Client/User

**Basic Flow or Main Scenario:**

* 1. Client prompts system by clicking “make a payment” button.
  2. System responds by redirecting the user to a page with the credit balance. The page allows the user to either pay full balance or a certain amount of the credit balance
  3. The client prompts the system by clicking pay full balance.
  4. The system responds by taking the balance from the checking account linked to the client’s credit line.

**Extensions or Alternate Flows:**

* 1. The client prompts the system by choosing to make pay part of the balance
  2. The system prompts the client by asking them to put the amount of money they wish to go towards the credit balance.
  3. Client responds by putting it in the desired amount
  4. The system responds by deducting that amount from the checking account and reducing the credit balance by that amount.

If Checking account linked to client goes to negative: See UC-018

**Use Case ID:** UC-016 (S.B)

**Use Case Name:** Automatic Billing.

**Relevant Requirements:** See SRS

**Primary Actor**: automated by Bank

**Basic Flow or Main Scenario:**

1. The balance is due for a credit card bill.
2. The customer has agreed to allow automatic, recurrent billing for their credit card.
3. The Bank will automatically withdraw the minimum payment amount from a preexisting, pre-approved checking account (Pre-approved meaning that the designated checking account was chosen for automatic payments.)

**Extensions or Alternate Flows:**

* 1. Not enough funds in the checking account for minimum payment– take out minimum payment amount from checking account, put checking account into negative balance, charge Overdraft Fee to client.

**Exceptions:**

**Related Use Cases:**

**Use Case ID:** UC-017 (A.M.)

**Use Case Name:** Log Information.

**Relevant Requirements:** See SRS

**Primary Actor**: Client/User

**Basic Flow or Main Scenario:**

1. A change occurs in any field of the clients’ account
2. The change is noted inside of the system
3. When the user is done with the specific page or logs out, the changes are saved in a log

**Extensions or Alternate Flows:**

1. The Teller changes a client’s account information
   1. The changes are automatically saved to a log
2. The Teller withdrawals or deposits clients’ money
   1. The changes are automatically saved to a log
3. The client uses the ATM to withdraw or deposit money
   1. The changes are automatically saved to a log

**Use Case ID:** UC-018 (SB)

**Use Case Name:** Accumulate Fees

**Relevant Requirements:** See SRS

**Primary Actor**: Automated by Bank

**Basic Flow or Main Scenario:**

1. A client’s account has raised a flag that tells the Bank system that Debt has accumulated, those being:
   * 1. Overdraft fee: A client’s checking account has gone into the negatives.
     2. Overdue Balance fee: The bill for a credit card has not been paid.
2. The Bank System will assess which type of fee to charge.
3. The Bank System will add the flat rate to a client’s record

**Extensions or Alternate Flows:**

**Exceptions:**

**Related Use Cases:**

**Use Case ID:** UC-019

**Use Case Name:** Close Account-Automatic

**Relevant Requirements:** See SRS

**Primary Actor**: Automated by Bank

**Basic Flow or Main Scenario:**

1. A client’s account has accumulated too much debt and has not had any payments made toward the debt.
2. The system logs all client information, including debt and the last payment made towards balance.
3. The system closes the account.
4. Goes to debt collections (tsktsktsktsk)

**Extensions or Alternate Flows:**

* 1. The system is prompted by that record breaking debt and call the police!

**Exceptions:**

**Related Use Cases:**

**Use Case ID: UC-020** **Use Case Name:** Process Server Message  
 **Relevant Requirements:** Internal Interface Requirements, Server Module  
 **Primary Actor:** Server  
 **Basic Flow or Main Scenario:** Server receives a message from a connected client or teller  
 Server checks the message type (LOGIN, LOGOUT, or TRANSACTION\_RECORD)  
 Server runs the correct method based on the type  
 Server processes the request and sends back a response  
 The action is saved in a log

**Extensions or Alternate Flows:** If the message type is invalid, server sends an error message  
 If the message is broken or unreadable, server skips it and logs the issue

**Exceptions:** If the server thread crashes, it restarts and keeps listening

**Related Use Cases:** UC-001, UC-005, UC-006, UC-009, UC-010, UC-017

**Use Case ID: UC-021** **Use Case Name:** Log Login Attempt  
 **Relevant Requirements:** SRS 3.1.6, Security Requirements   
 **Primary Actor:** Server  
 **Basic Flow or Main Scenario:** Server gets a login request from a client or teller  
 Server saves the login info:

* Bank ID or Employee ID
* Time and Date
* IP address (if available)
* Whether login was successful or not  
   Server continues with login process  
   Server stores this info in a text log

**Extensions or Alternate Flows:** If there are 3 failed login tries, server may block more attempts

**Exceptions:** If IP can’t be found, log it as “Unknown”

**Related Use Cases:** UC-001, UC-020, UC-017

**Use Case ID: UC-022** **Use Case Name:** Log Account Access Event  
 **Relevant Requirements:** SRS 3.1.6  
 **Primary Actor:** Server  
 **Basic Flow or Main Scenario:** A client or teller opens a bank account view  
 Server checks if they’re allowed to access it  
 Server logs the following:

* Who accessed it (Bank ID or Employee ID)
* The account number
* Time and Date
* That it was a “view only” access  
   Server then shows the account data

**Extensions or Alternate Flows:** If someone tries to view an account they shouldn’t, the system blocks it and logs the attempt

**Exceptions:** If the connection is lost halfway, server logs a partial access

**Related Use Cases:** UC-002, UC-003, UC-008, UC-017