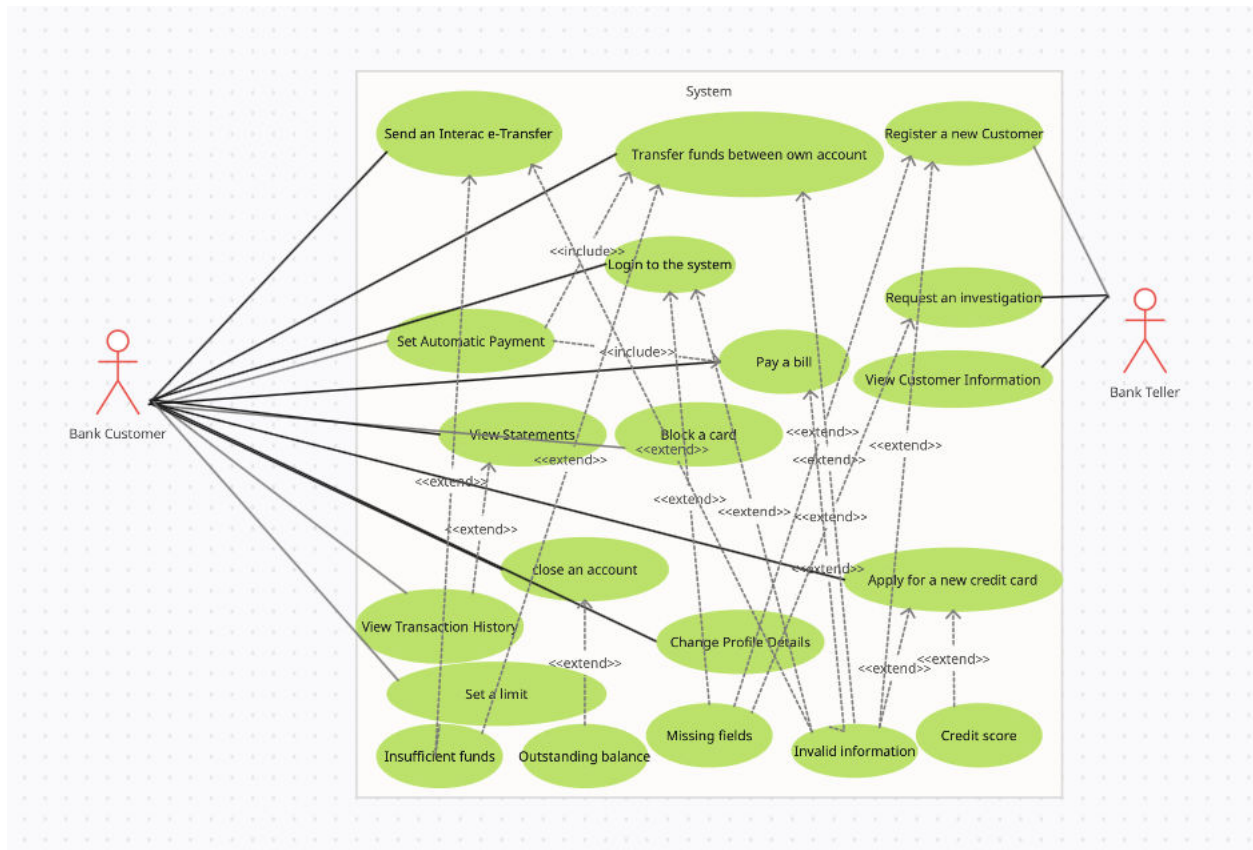


Use Case Diagram:



Use Case Scenarios:

Register a new bank customer

- Brief Description
 - This use case describes the process of registering a new bank customer to the banking system
- Actors
 - Bank Teller
- Triggers
 - System Administrator clicks on “Register Account” button on Bank Admin Interface
- Flow of Events
 - Basic Flow
 1. Use case begins when the Bank Teller clicks on “Create Account Button” on the Bank Admin’s side on the system.
 2. System prompts with a page to fill in details about the new customer such as their name, address, contact information, and other identification information.
 3. The Bank teller clicks on ‘Register Account’ button
 4. The system verifies the information, branches to “Data Encryption USe Case” and saves it in the database, and generates a unique account number and a one-time password for the new customer.
 5. The system displays a success message on successful registration of the customer.
- Alternative Flows and/or Subflows
 - Use case displays an error message if any required fields are left blanked or invalid characters has been entered and prompt to the user(Bank Admin) to correctly enter the information
- Special requirements
 - A valid Government issued photo ID and other basic information like SIN, TIN, and contact information.
- Preconditions
 - The bank teller is logged into the system
- Postconditions
 - - Success Postcondition
 - i. The bank account is created successfully and unique account number and one-time password is generated
 - Failure Postcondition
 - i. The use case will prompt the user to re-enter the information again
- Extension Points
 - None

Log in to the system

- Brief Description
 - The use case describes the process of logging in to the bank account. This use case appears when the user opens the mobile banking application. The application displays a login screen with the username and password
- Actors
 - Bank Customer
- Triggers
 - Bank Customer clicks on the “Login” button after entering their username and passwords
- Flow of Events
 - Basic Flow
 1. Use case begins when customer enters their username and passwords and click on the login button
 2. The system verifies the customer’s login credentials
 3. The system allows access to the customer’s account if the credentials are valid
 4. The system displays the account dashboard with account types followed by the balance in it along with other available options.
- Alternative Flows and/or Subflows
 - Invalid credentials
 - i. If the customer’s login credentials are incorrect, the system displays an error message and prompts the customer to re-enter their credentials
 - ii. If the customer enters incorrect username or password multiple times, the system blocks the log in trials for 5 minutes and prompts the customer to click on ‘forget passwords’ to set a new password or contact the bank’s customer service for assistance
 - Insufficient funds
 - i. The use case ends with a pop up stating that the customer does not have enough funds.
- Special requirements
 - Username and passwords
- Preconditions
 - The customer has an active bank account
- Postconditions
 -
 - Success Postcondition
 - i. The system displays the account dashboard with account balance in each account and other options.
 - Failure Postcondition
 - i. The system prompts the user to enter login credentials again up to a certain time.
- Extension Points
 - None

Apply for a new credit account

- Brief Description
 - This use case describes customers filling out an online application form with their personal information such as name, address, date of birth and also other financial and employment necessity information while applying for a new credit account.
- Actors
 - Bank Customer
- Triggers
 - Customers click on the “submit application” button with their consent approved in the registration page for opening a credit account.
- Flow of Events
 - The user selects "Open a new credit account" from the account actions menu
 - The user enters their personal and financial details as well as the desired credit limit into an online application form.
 - The system verifies the customer's information and runs a credit check to check the user's financial strength for the desired credit limit.
 - Once approved, a credit account is set and the user receives a confirmation email followed by security encryption and privacy protocols in securing the account.
 -
- Alternative Flows and/or Subflows
 - Low credit score
 - i. Use case ends with the system displaying a message if the user's financial strength is not strong enough for the credit limit they have chosen.
 - ii.
 - Use case displays an error message if the information provided by the user to fill up the registration form seems incorrect(fake) or similar to another user.
- Special requirements
 - Amount of credit required
- Preconditions
 - The customer has enough funds in their account to set up a credit account for the particular chosen credit limit.
- Postconditions
 - The customer is maintaining active credit transactions
- Success Postcondition
 - i. The credit account is created successfully and is linked to their online banking account
 - ii. Has a unique credit card is assigned to the account to be delivered to them in person.
- Failure Postcondition
 - i. The customer has been rejected to create a credit account due to their poor credit history, financial strength, or other factors.

- Extension Points
 - None

Close an account

- Brief Description
 - The use case describes the process of canceling the bank account. The use case will appear when the customer requests for the account closure through the banking application.
- Actors
 - Bank teller
 - Bank customer
- Triggers
 - The customer clicks on "Close account" button on the account dashboard and the system administrator clicks on "Close User account" on Bank Admin Interface
- Flow of Events
 - Basic Flow
 1. Use case begins when the Customer applies for "Close Account" on the account dashboard.
 2. The bank teller will access the Bank Admin Interface to initiate the closure of the account
 3. The "Close User Account" button on the bank admin interface will let the bank teller delete all of the saved transactions and the client private information
 4. Once the account is closed an automated email will be sent out to the customer
 5. The customer will lose all access to the bank account after the closure
- Alternative Flows and/or Subflows
 - Alternative flow (account has funds)
 1. The use cases branch to the alternative flow when the user has funds in the account during the closing request
 2. Request will not be completed if there is a negative balance in the chequing/savings account
 3. The system will deny the request if there is an overdue balance on the user's credit card
 4. All of the failures will pop up a "Request Denied" message which will lead to the issues
 5. The admin interface will provide the user with the ability to view the customer's details and obtain a report
 6. The account closure will be successful once the customer has resolved the objections
- Special requirements
 - Double authorization from the system administrator
- Preconditions
 - The client has to request closure
 - Account needs to have zero balance and the overdue bills need to be paid in full
- Postconditions
 - Success Postcondition
 1. Client will get a full account report to their designated email
- Failure Postcondition

- An unsuccessful message will be sent to the client's preferred email and phone number via the banking application.
- Extension Points
 - None

Send an e-Transfer

- Brief Description
 - The use case describes the process of sending a money from the user's account to an account that is saved (or will be saved) in the contact list of the user. This use case appears when the user presses e-Transfer button on the account actions menu. The action asks for an amount of money to send and a contact registered with an email or a/ phone number to send the money to
- Actors
 - Bank Customer
 - Contact that receives the money
- Triggers
 - The user clicks on “Send Money” button on the account actions menu
- Flow of Events
 - User case begins when the customer clicks on “e-Transfer ” button on the dashboard in their account
 - System prompts with a page to fill in fields including the amount of transactions, account type to send money from, and name and contact information of the receiver if the receiver is not a saved contact. If the receiver is in the saved contact list, it will ask the customer to select the receiver from the saved contact list.
 - Customer clicks on the “Complete Transfer” button
 - The system verifies that the amount entered, and receiver’s information is valid and the user has sufficient funds to make the transfer
 - If the transfer is successful, the system deducts the transferred amount from the sending account and sends it to the receiving account.
 - The system displays a success message and shows new account balances.
- Alternative Flows and/or Subflows
 - Incorrect information
 - i. The use case ends with the system displaying an error message if the amount is entered incorrectly or the receiver’s information is incorrect.
 - Insufficient Funds
 - i. The use case ends with a message stating that there’s not enough funds and prompts the user to re-enter the amount.
- Special requirements
 - Amount of money and email or phone number of the contact
- Preconditions
 - The customer must be logged in

- Postconditions
 -
- Success Postcondition
 - The system displays a brief report of the transfer statement including the information of the receiver and the sent amount of money
 -
- Failure Postcondition
 - the system prompts the user to re-enter the information
- Extension Points

Transfer funds between own accounts

- Brief Description
 - This use case describes the process of transferring funds between own accounts
- Actors
 - Bank Customer
- Triggers
 - Customer Clicks on the “Complete Transfer” button on the “Transfer Fund” page of their account
- Flow of Events
 - User case begins when the customer clicks on “Transfer Funds” button on the dashboard in their account
 - System prompts with a page to fill in multiple fields including account types from which the fund would be transferred to the account type receiving the fund and the amount of funds.
 - Customer clicks on the “Complete Transfer” button
 - The system verifies that the amount entered is valid and the user has sufficient funds to make the transfer
 - If the transfer is successful, the system deducts the transferred amount from the sending account and adds it to the receiving account.
 - The system displays a success message and shows new account balances.
- Alternative Flows and/or Subflows
 - Use case displays an error message if the amount is entered incorrectly or if there’s not enough fund and prompts the user to re-enter the amount
- Special requirements
 - Amount of money and account types to transfer funds from and to
- Preconditions
 - The customer must be logged in
- Postconditions
 -
 - Success Postcondition
 - i. The system displays a success message and new account balances.

- Failure Postcondition
 - i. The system prompts the user to enter the amount again
- Extension Points

Pay a bill

- Brief Description
 - This use case describes the process of sending money from the user's account to the billing company. This use case appears on clicking "Pay Bill" button on the dashboard
- Actors
 - Bank Customer
 - Billing Company
- Triggers
 - Customer clicks on the "Complete Payment" button in the "Pay Bill" page of their account
- Flow of Events
 - User case begins when the customer clicks on "Pay Bill" button on the dashboard in their account
 - System prompts with a page to fill in multiple fields including name of the billing company, account number of the billing company, payment amount, and user's account type for sending the money.
 - Customer clicks on the "Complete Payment" button.
 - The system verifies that the bill information and amount entered is valid, the user has sufficient funds to make the payment
 - If the transfer is successful, the system deducts the payment amount from the user's account and send it to the billing company
 - The system displays a success message and shows the new account balance.
- Alternative Flows and/or Subflows
 - Use case displays an error message if any required field is left blank, bill information or amount is entered incorrectly or if there's not enough fund and prompts the user to fill in the fields correctly
- Special requirements
 - Name and account no of the billing company
- Preconditions
 - The customer must be logged in the system
- Postconditions
 -
- Success Postcondition
 - The system displays a brief report of the transfer statement including the information of the billing company and the sent amount of money
- Failure Postcondition

- The system prompts the user to re-enter the information
- Extension Points

Set automatic payments

- Brief Description
 - This use case describes the process of setting autopayments on a scheduled date to pay a recurring bill such as a credit card, utility bill or mortgage
- Actors
 - Bank Customer
- Triggers
 - The customer clicks on "Scheduled payment" on user interface
- Flow of Events
 - Case starts when the client logs into the account and accesses the menu from the dashboard
 - Menu will have the option to pay bills and it will require billing information
 - Client will have to add payee by entering company name, account number and zip code
 - After adding the payee to the list the option for making the recurring payment will appear
 - The payment will require the amount, the account the money will be deducted from and setting up the delivery dates
- Alternative Flows and/or Subflows (Incorrect information)
 - The use case branches to an alternative flow when the user inputs incorrect information (Wrong payee name or account number)
 - The input box will pop up a message saying incorrect information and it will eventually fail the transaction
 - Use case will have an "OK" button which will lead back to the payee information page
- Special requirements
 - None
- Preconditions
 - Need to add the payee information
- Postconditions
 - Success Postcondition
 1. The application will show a message if the payment is successful
- Failure Postcondition
 - It will take back to the system which will require the user input after it shows an error
- Extension Points
 - None

Set a daily / monthly limit on an account

- **Brief Description**
 - This use case indicates a temporary daily or monthly limit on the number of transactions clients can perform on their savings or checking accounts.
- **Actors**
 - Bank customers
- **Triggers**
 - The customer selects the "Daily Transaction Limits" on the account summary page
- **Flow of Events**
 - Customer clicks on the Summary page from the user dashboard
 - A user prompt will ask to "Choose Limits" for the account
 - The option will contain ATM withdrawal, purchase or transfer limits for the account
 - The prompt will ask for the expiry date
 - "Done" button will succeed the process
- **Alternative Flows and/or Subflows**
 - None
- **Special requirements**
 - None
- **Preconditions**
 - The client is logged in
 - Putting proper information
- **Postconditions**
 - Success Postcondition
 - 1. The application will show a message if the limit has been set
- **Failure Postcondition**
 - It will keep showing error messages if the information is incorrect
 - The prompt page will require correct information afterwards
- **Extension Points**
 - None

Block a card

- **Brief Description**
 - This use case describes the operation of blocking all card operations for a selected card.
- **Actors**
 - The client
- **Triggers**

- The client selects a specific card in the provided menu and confirms the blockage of the card.
- Flow of Events
 - Normal flow
 - i. The client clicks on the “Privacy and Security” button, located next to the client’s name on top of the screen.
 - ii. From the provided page with Privacy and Security options, the client selects “Block a card”.
 - iii. The system shows a list of cards that the client has.
 - iv. The client selects a card they want to block.
 - v. The system provides a message with a list of consequences this will have, with buttons “Lock the card”, “Select a different card” and “Cancel”.
 - vi. The client clicks on “Block the card”.
 - vii. The banking system blocks the corresponding card.
 - viii. The use case ends with a pop up message indicating that the card has been successfully blocked, which also provides bank contact information to further investigate the issue leading to the card blocking (stealing, etc.). The client is returned to the “Privacy and Security” section.
 - Alternative Flow (Cancel the operation)
 - i. The use case branches to the alternative flow when the client selects “Cancel” (Normal flow, v).
 - ii. The use case ends with the client returned to the “Privacy and Security” section of the system.
- Special requirements
 - None
- Preconditions
 - The client is logged in.
 - The client has to have an existing physical or virtual card linked to one of the accounts.
- Postconditions
 - Success Postcondition
 - i. xviii. The card selected before is locked.
 - ii. The client is shown the “Privacy and Security” section of the banking system.
 - Failure Postcondition
 - i. The client is returned to the “Privacy and Security” section of the banking system.
- Extension Points
 - None

View transaction history

- Brief Description

- This use case describes the process of a client accessing transaction history.
- Actors
 - The client
- Triggers
 - The client clicks on the corresponding account.
- Flow of Events
 - Normal flow
 - i. The use case starts with the client being on the main page, where he clicks on one of the accounts they want to see the history of.
 - ii. The use case ends with the page with the corresponding account and its history shown.
 - Alternative Flows and/or Subflows
 - i. xxiii. None
- Special requirements
 - None
- Preconditions
 - The client is logged in.
 - The client has an account (chequing, savings, credit).
- Postconditions
 - Success Postcondition
 - i. xxiv. The transaction history is shown to the client.
 - Failure Postcondition
 - i.
- Extension Points
 - The client can select a transaction to see detailed information about it, such as exact time of the transaction, location, etc.

View statements

- Brief Description
 - This use case describes the process of a client accessing a specific bank statement. The statements are categorized by account (chequing, credit, savings) and period of time.
- Actors
 - Client
- Triggers
 - The client clicks on the “View Statement” button after specifying the account and period.
- Flow of Events
 - Normal Flow
 - i. xxvi. The client clicks on the corresponding account on the main page of the banking system.

- ii. xxvii.The client clicks on the “View Statements” button located above the history of transactions.
 - iii. xxviii.The system provides a list of periods for which the bank statements exist.
 - iv. xxix.The user clicks on the “View Statement” button next to the necessary period.
 - v. The use case ends with the statement being shown.
- Alternative Flows and/or Subflows
 - i. xxxi.None
- Special requirements
 - None
- Preconditions
 - The client is logged in
- Postconditions
 - Success Postcondition
 - i. The bank statement is shown to the client.
- Extension Points
 - The statement is available for download.

Change profile details

- Brief Description
 - The use case describes the process of a client accessing the profile information section and changing the section corresponding to the needs.
- Actors
 - Client
- Triggers
 - The client clicks on the “pencil” icon next to one of the sections of personal information, while being inside the “Personal Information” section of the banking system.
- Flow of Events
 - Normal flow
 - i. xxxiii.The use case starts, with the client logged in to the system, clicking on the “Personal Information” button located next to the client’s name at the top of the page.
 - ii. xxxiv.The client clicks on the pencil icon next to the information they would like to change (e-mail, address).
 - iii. xxxv.The system asks to input the updated information.
 - iv. xxxvi.The client inputs the information and clicks on “Update”.
 - v. xxxvii.The system makes updates to the information in the database. If the updated information is to be manually checked, it is sent to the corresponding department (address or name change).

- vi. xxxviii. The use case ends with a return to the page with personal information and a pop up saying that the information has been updated.
- Alternative Flows and/or Subflows
 - Alternative flow (incorrect information provided)
 - i. xxxix. The use case branches to the alternative flow when the user provides incorrect information (the information doesn't match the regex, etc.) (Normal flow, iv).
 - ii. The system shows a pop up saying that the information provided is incorrect and gives a reason (doesn't match the format, too many symbols, etc.)
 - iii. The use case ends with the user clicking on "Okay" on the pop up and being returned to the form to input information again.
- Special requirements
 - None
- Preconditions
 - The client is logged in.
- Postconditions
 -
 - Success Postcondition
 - i. The personal information page is shown. The information is either immediately updated, or sent to the bank for a manual check.
 - Failure Postcondition
 - i. The system is waiting for user interaction. The form to provide updated information is shown.
- Extension Points
 - None

View customer information

- Brief Description
 - This use case describes the process of checking a client account information.
- Actors
 - Bank teller
- Triggers
 - Bank teller clicks on the "Find a Client" button on the main page of the visual interface.
- Flow of Events
 - Normal flow
 - i. Bank teller clicks on the corresponding button on the main page of the visual interface.
 - ii. A prompt asks to input the search query. The options are: a) search by account number (unique customer ID); b) search by personal information (client name, phone number, e-mail address).

- iii. Bank teller inputs the information (either a or b).
 - iv. xlvii.The system shows the list of the corresponding to the search query accounts.
 - v. xlviii.The bank teller selects the account from the search results and clicks on “Manage Client”.
 - vi. The use case ends with the client account management page being shown, with the system adding the action of access to the account to the access history.
- Alternative Flows and/or Subflows
 - Alternative flow (client not found)
 - i. The use case branches to the alternative flow when there are no search results (normal flow, iii).
 - ii. A pop up dialog appears saying “No search results”.
 - iii. The use case ends with the bank teller clicking on “Okay” on the pop up. The system returns the bank teller to the search form (Normal flow, ii).
- Special requirements
 - None
- Preconditions
 - The bank teller is logged in.
 - The customer exists in the banking system.
- Postconditions
 -
 - Success Postcondition
 - i. System waiting for interaction. The client management page shown to the bank teller and a corresponding element added to the client account access history.
 - Failure Postcondition
 - i. System waiting for interaction. The search form is shown.
- Extension Points
 - None

Request an investigation on behalf of the customer

- Brief Description
 - This use case describes the process of requesting an investigation on one of the customer’s transactions. This use case is called from the use case “View Customer Information”
- Actors
 - Bank Teller
- Triggers

- Bank Teller clicks on the “Submit Investigation Form” button on one of the transactions in the Transaction History page of a customer.
- Flow of Events
 - Basic flow
 - i. Use case begins when the Bank Teller clicks on “Request Investigation”.
 - ii. System prompts with a page with multiple fields to fill in details regarding the investigation request.
 - iii. Bank teller clicks on “Submit Investigation Request”.
 - iv. System saves the request and sends it to the investigation department.
 - v. Use case ends with a pop-up dialog showing the request reference number and a success message, with the application form being closed. Return to the transaction history of the client.
 - Alternative Flow (Insufficient Information)
 - i. Use case branches to the “Insufficient Information” use case when at least one of the mandatory fields in the form was not filled in and the bank teller clicks on “Submit Request” (Basic flow, iii).
 - ii. A pop up dialog appears with a message saying “Insufficient Information. Please fill in all the missing fields marked red.
 - iii. The use case ends with clicking “Okay” on the pop up, and returns to the application, with fields required now marked red.
- Special requirements
 - None
- Preconditions
 - The bank teller is logged in.
 - The customer has a transaction recent enough to do the investigation.
- Postconditions
 - Success Postcondition
 - i. System waiting for user interaction. Transaction history shown. The application is sent to the department.
 - Failure Postcondition
 - i. Transaction history shown. No changes done to the system or reports sent.
- Extension Points
 - None

Trigger: what will start the use case

Actors: that participate in this use case

Entry conditions: that must hold in order for the use case to begin at all

Success guarantee: if the use case falls all the way to the end through the main flow of events, it is a success – the postconditions of which define

what it actually means

Minimum guarantee: if the use case ends via ANY of the alternative flows, it did not succeed, and failure postconditions define what should hold in that case (at least)

Steps in the main (basic) and alternative (exceptional) flows are often numbered to facilitate searching and cross-referencing