| **Software Requirement Specifications**  **Bank Management System**  **Version: 1.30**   | Project Code |  | | --- | --- | | Supervisor | Ms. Syeda Rubab Jaffar | | Co Supervisor |  | |  |  | | Project Team | Syed Yousif Ali Shah (22K-5174)  Huzaifa Kashif  (22K-5158)  Muhammad Hamza Hussain  (22K-5182) | | Submission Date | 23/03/2024 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

[Instructions]

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.
* MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

Document History

| **Version** | **Name of Person** | **Date** | **Description of change** |
| --- | --- | --- | --- |
| v1.00 | Huzaifa Kashif | 20/03/2024 | Document Created |
| v1.20 | Syed Yousif Ali Shah | 22/03/2024 | Added Use Case Diagram |
| v1.30 | Muhammad Hamza Hussain | 23/03/2024 | Modified Functional And Non-Functional Requirements |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

| **Name** | **Role** | |
| --- | --- | --- |
|  | | Supervisor |
|  | | Co- Supervisor |
|  | |  |

Document Sign-Off

| **Version** | **Sign-off Authority** | **Sign-off Date** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1.](#_30j0zll) Introduction 7

[1.1.](#_3znysh7) Purpose of Document 7

[1.2.](#_2et92p0) Intended Audience 7  
1.3 Abbreviations ………………………………………………………………………………………...7

[1.4.](#_tyjcwt) Document Convention 7

[2.](#_3dy6vkm) Overall System Description 8

[2.1.](#_1t3h5sf) Project Background 8

[2.2.](#_4d34og8) Project Scope 8

[2.3.](#_2s8eyo1) Not In Scope 8

[2.4.](#_17dp8vu) Project Objectives 8

[2.5.](#_3rdcrjn) Stakeholders 8

[2.6.](#_26in1rg) Operating Environment 8

[2.7.](#_35nkun2) System Constraints 8

[2.8.](#_1ksv4uv) Assumptions & Dependencies 8

[3.](#_2jxsxqh) External Interface Requirements 9

[3.1.](#_z337ya) Hardware Interfaces 9

[3.2.](#_3j2qqm3) Software Interfaces 9

[3.3.](#_1y810tw) Communications Interfaces 9

[4.](#_4i7ojhp) Functional Requirements 10

[4.1.](#_2xcytpi) Functional Hierarchy 10

[4.2.](#_1ci93xb) Use Cases 10

[4.2.1.](#_3whwml4) [Title of use case] 10

[5.](#_2bn6wsx) Non-functional Requirements 11

[5.1.](#_qsh70q) Performance Requirements 11

[5.2.](#_3as4poj) Safety Requirements 11

[5.3.](#_1pxezwc) Security Requirements 11

[5.4.](#_49x2ik5) User Documentation 11

[6.](#_147n2zr) References 12

[7.](#_3o7alnk) Appendices 13

1. **Introduction**

* 1. **Purpose of Document**

The purpose of this document is to present a detailed description of the Bank Management System.This document would explain each feature and requirement of the system, while also covering all the external interfaces, and non-functional requirements.

* 1. **Intended Audience**

This document and the system itself is intended for all the stakeholders and the developers involved in using, and creating the BS project. Eg; Bank customers, whose financial data security, experience, and service accessibility is of paramount importance, for the staff, as they would benefit from the improved automation, efficiency, and workflow, as well as for the executives and the management, as they would benefit from the overall impact of the system on the bank’s profitability, and strategic goals.

**1.3 Abbreviations**

| BMS | Bank Management System |
| --- | --- |
| BS | Banking System |
| UI | User Interface |
| OS | Operating System |
| HDD / SSD | Hard-Disk Drive / Solid-State Drive |
| GUI | Graphical User Interface |
| HTTP | Hypertext Transfer Protocol |

* 1. **Document Convention**

Font: Arial

Font Size: 11-12

1. **Overall System Description**
   1. **Project Background**

This Banking Management System (BMS) is developed to address the need for an efficient banking operations management which includes account creation, transactions, and account management. The system aims to streamline banking operations, thereby, improving customer service.

* 1. **Project Scope**

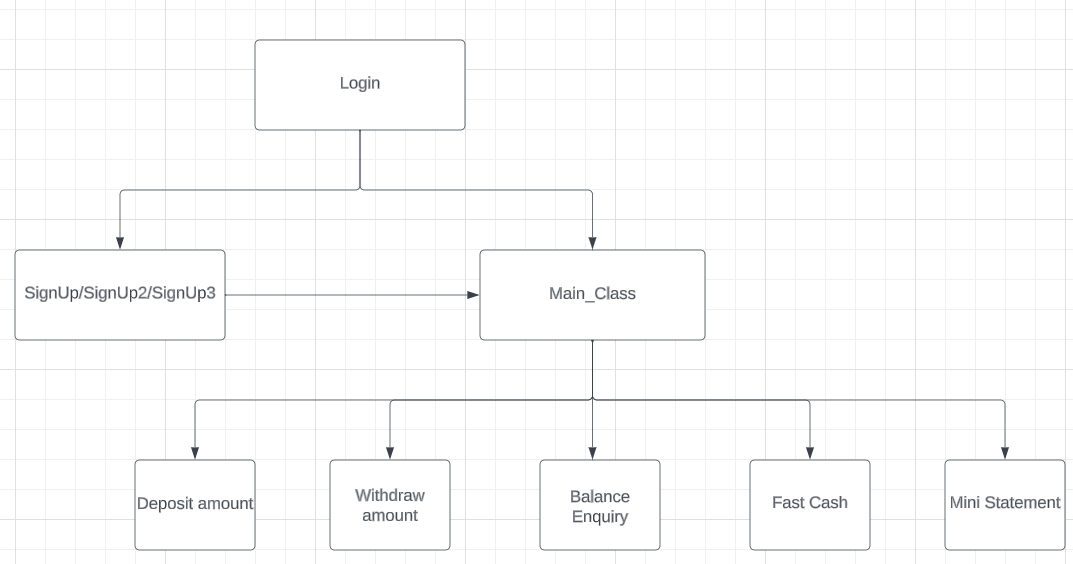
The purpose of this system is to automate various banking operations, customer onboarding, account management, transaction processing, and reporting, thereby significantly reducing manual work and paperwork, improving efficiency. This project encloses the development of a user-friendly GUI for banking operations using Java Swing, and gracefully integrates with MySQL database for storing account information.

* 1. **Not In Scope**

Currently, integration with external banking systems, and third party services is not within the scope of this system.

* 1. **Project Objectives**

Following is a class connection diagram that represents how each class interacts with the other, and the functionalities that the system provides:



* 1. **Stakeholders**
* **Administrators:** Administrators must possess high technical expertise to manage user accounts, system configuration, system database, and system security.They would also be responsible for system maintenance and updates.

* **Employees:** Bank employees would be the regular users of this system, responsible for handling day-to-day banking operations (eg; accounts creation confirmation, providing customer support, and executing transactions). They must possess moderate technical expertise for interacting, and navigating through the GUI components of the banking system.

* **End-Users:** End users of the banking system would have varying technical expertise. Hence, they would be responsible for engaging in account related activities, such as performing transactions (eg; deposition, withdrawal, fast cash withdrawal), account balance enquiry, and checking their mini-statement (activity log).
  1. **Operating Environment**

The system operates in a stable Java environment, and utilizes Java Swing for the GUI components.Whereas, the system will use Java Swing for GUI. It is compatible with MySQL databases for data storage, and data retrieval. Operating systems include Windows, Linux, and macOS, but the recommended operating system is Windows.

* 1. **System Constraints**

Appropriate constraints might include adherence to banking regulations, a properly configured MySQL database for data storage, as well as compatibility with Java Swing for GUI development.

* 1. **Assumptions & Dependencies**

**Assumptions:**

* We have utilized a stable and up-to-date Java version for our Bank Management System so that all the features that we have implemented are put in an efficient and effective manner.
* We have utilized a stable Java Swing version for our GUI components so that all our GUI components are well navigated and utilized to the fullest.
* We assume that we have utilized a stable MySQL database so that the information of the user or the user’s data is safeguarded by backing it up to the latest database and then retrieving the information back whenever required, especially when there is a chance of data deletion or corruption.

**Dependencies:**

* Our Banking Management System must align with the operational policies, security protocols, and service standards as specified by the customer to meet strategic goals.
* BMS must comply with the data protection laws, organizational and financial regulations, and industry standards to ensure legal and organizational integrity.
* BMS also depends on the user and staff training that is essential and helpful for the users to understand the features of BMS, troubleshooting and security protocols.

1. **External Interface Requirements**
   1. **Hardware Interfaces**

The system interfaces with the standard hardware components to ensure its compatibility with Java, Java Swing, and MySQL working environment. The standard hardware components include RAM for temporary data storage, HDD/SSD for lasting data storage, network interfaces, etc.

For interacting with the system, there should be a monitor screen for GUI and information display, mouse, and keyboard for interacting with the GUI components, and a continuous supply of energy (battery).

* 1. **Software Interfaces**

| Windows Operating System | We recommend Windows OS for better support, performance, and its user-friendliness. |
| --- | --- |
| MySQL | To save user account transactions, and other related account details and data, we have chosen MySQL Database. |
| Intellij Idea | We chose Intellij IDE for developing the banking system because of its user-friendly interface, built-in classes, interfaces and functions related to Java, and also due to its unique feature to autocorrect, import packages, and provide solutions to errors on its own. |
| Java Swing | We have utilized Java Swing for developing the GUI for the banking system. |

* 1. **Communications Interfaces**

The system communicates through HTTP, and other standard protocols. The system also utilizes simple electronic forms for the user to either create, or login to his or her bank account. Moreover, it ensures secure data transfer through encryption. Communication between the system, and MySQL database is established by utilizing Java Database Connectivity for data transmission and data retrieval.

1. **Functional Requirements**
   1. **Functional Hierarchy**
2. **Creation Of Account Using Sign-Up:**
3. **Description And Priority**

This feature of our BMS system will allow the users who do not currently have an existing account in the database to create it. It involves three main processes in which the user will provide personal, account and additional details. The priority of this feature is high, as it is a fundamental aspect of user registration and access to banking services.

1. **Stimulus/Response Sequences**

* **User Registration/personal information page:**
  + The user selects the option of "Sign Up" from the main menu.
  + A unique form number is assigned for the particular user.
  + The system redirects the user to the first page, asking the user to enter personal details such as name, father’s name, gender, email address, city, state, address date of birth, and marital status.
  + Then the user is asked to enter a 4 digit pin that he wants to use for his account.
* **Additional Information:**
  + The system redirects the user to the second page, where additional details are collected.
  + The second page asks the user to enter personal details such as: Religion, Category, Income, Qualification, Occupation, PAN number, senior citizen(yes/no) and existing account(yes/no)
  + the user clicks on the next button.
* **Account Details:**
  + The system redirects the user to the third page, where the user is asked to select the account type he wants to create.
  + A 16 digit card number is generated with only the last 4 digits visible and hidden pin number.
  + Then the user is asked to select the services he wants from our BMS system.
  + The user will have to click on the declaration box to confirm that the details he has entered are all correct else he will not be able to proceed.
  + After providing this information, the user submits the form. Upon clicking the submit button, an information box will appear showing the card number assigned to the user which is saved in the database and the pin number the user entered for his account.
* **Account Creation Confirmation:**
  + The system verifies all entered details for accuracy and completeness.
  + If all information is valid, the system creates a new account for the user and confirms successful account creation.
  + The user is then redirected to the login page to access their newly created account.

1. **Functional Requirements**

**REQ-1:**

The system should validate the uniqueness of the chosen pin number for the account.

**REQ-2:**

The system must make sure that the pin set for the account fulfills the complexity standards for enhanced security.

**REQ-3:**

The system should store the sensitive information provided by the user securely in the database.

**REQ-4:**

The system should not allow the user to proceed without filling the required information to ensure the complete data submission.

**REQ-5:**

The system should generate a unique form number upon each request of account creation.

**REQ-6:**

The system shall not proceed until and unless the user declares in the tick box that he has provided the correct information.

**REQ-7:**

The system should generate a unique card number that will only be visible after the user submits the last form of account details.

**REQ-8:**

The system must allow the user to select multiple options among the services he wants but only one option in the account type selection part.

**II. Sign-In To An Already Existing Account To Use The ATM:**

1. **Description And Priority**

This feature enables users to sign in to their already existing accounts when using the ATM. To proceed with ATM transactions, users must enter the correct Card Number and PIN number. The priority of this feature is critical, as it directly relates to the security and access control of user accounts.

1. **Stimulus/Response Sequences**
   * The user enters the ATM card in the machine.
   * The system asks the user to enter the Card number and then the PIN (personal identification number).
   * User selects the sign in the box.
   * The system verifies the entered card number and the pin against the record already present in the database of our BMS.
   * If the pin card number is found and the entered pin matches the card number stored, the user successfully signs in and is allowed to proceed with all the transaction and other processes.
2. **Functional Requirements**

**REQ-1:**

The system should compare and find the entered card number in the list of accounts registered in the database.

**REQ-2:**

The system must validate the pin against the card number entered securely with a limited amount of attempts allowed.

**REQ-3:**

Upon successful authentication, the system should grant access to the user's account and display available transaction options.

**REQ-4:**

The system should provide a clear error message in case of unsuccessful authentication and the user should be asked to reenter the information.

**REQ-5:**

The system should have a mechanism to handle and log security-related events, such as multiple failed login attempts, for auditing purposes.

**III. Types of accounts users can create:**

1. **Description and Priority**

This feature allows users to create different types of accounts based on their financial needs. Users should be able to choose from four types of accounts: Savings Account, Current Account, Fixed Deposit Account, and Recurring Deposit Account. The priority of this feature is high, as it provides users with flexibility in selecting the type of account that best suits their requirements.

1. **Stimulus/Response Sequences**

* .
  + The user is prompted to select the type of account he wants to create among four options during account creation, in page 3 of account details.
  + The system asks the user to select one option among the options including: Saving Account, Current Account, Fixed Deposit Account and Recurring Deposit Account
  + If the user selects the saving account option then the user will be asked to enter the amount to deposit if he wants to after the account has been created successfully.
  + If the user selects the "Current Account" option then the system collects necessary information for current accounts, such as business information.
  + If the user selects "Fixed Deposit Account," the system prompts the user to enter details such as deposit amount, tenure, and interest rate.
  + If the user selects "Recurring Deposit Account," the system collects information like monthly deposit amount, tenure, and interest rate.

1. **Functional Requirements**

**REQ-1:**

The system should provide a user friendly interface clearly showing the available account types.

**REQ-2:**

The system must validate that this field is not left empty to avoid incompleteness of data for account creation.

**REQ-3:**

The system should allow only one option to be selected.

**IV. Generating A 16 Digit Card Number, And Prompting User To Select A PIN:**

1. **Description And Priority**

This feature ensures that the user should be given a unique Card Number upon creating an account that should be of 16 digits and should only be visible after the submission of the third page and the user should be asked to enter the 4 digit PIN they want for their account in the first page of personal details during account creation. The priority of this feature is critical, as it involves security measures for user authentication.

1. **Stimulus/Response Sequences**

* .
  + The user is prompted to enter a 4 digit PIN for their account in the account creation form, page 1(personal details)
  + After the user submits the third page of account creation form, a dialogue box will appear that will generate and reveal the card number and show the pin code already set by the user.
  + After completing all required details on the subsequent pages, the user receives a confirmation message indicating that the account has been successfully created.

1. **Functional Requirements**

**REQ-1:**

The system should generate a unique 16 digit card number for the user.

**REQ-2:**

The system should prompt the user to enter a 4 digit PIN for an account that will meet the complexity standards.

**REQ-3:**

The system should add the new card number and PIN in the database securely.

**REQ-4:**

The system should reveal the card number only when the third page of form is submitted and the account is created successfully.

**V. Selection Of Services From The Account:**

1. **Description And Priority**

This feature allows users to specify the services they want associated with their account. The system prompts the user to select from a list of services, including ATM card, Internet Banking, Mobile Banking, Email Alerts, Cheque Book, and E-statement. The priority of this feature is high, as it tailors the account to the user's preferences and requirements..

1. **Stimulus/Response Sequences**
   * After filling the 2nd page of additional details, the user is redirected to the account details page where he is prompted to select the services he wants from the system from the given list.
   * System gives a list of services to select from including: ATM card, Internet Banking, Mobile Banking, Email Alerts, Cheque Book, and E-statement.
   * The user selects the services they want associated with their account needs.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear UI with proper representation of services offered by the bank.

**REQ-2:**

The system should display a list of available services including: ATM card, Internet Banking, Mobile Banking, Email Alerts, Cheque Book, and E-statement

**REQ-3:**

The system should allow the user to select multiple options from the provided list.

**REQ-4:**

The system should store the user's chosen services for future reference and account management.

**VI. Deposition Of Amount In The Account:**

1. **Description And Priority**

This feature allows users to deposit money into their accounts after successful account creation. The option to deposit money should be available both immediately after creating the account and at any time during the user's interactions with their account. The priority of this feature is high, as it is a fundamental aspect of banking services and account management.

1. **Stimulus/Response Sequences**

* .
  + After successfully creating the account, the system will prompt the user to make an initial deposit.
  + The user is directed to specify the amount of money to deposit.
  + The user will click on the deposit button available in the bottom left part.
  + the user can even select the back button if he wishes to go back to the main menu page.
  + The user can also access the option of depositing money during normal use of the account from the main menu.
  + The system will record the time and amount deposited and update the MINI statement that keeps the record.

1. **Functional Requirements**

**REQ-1:**

The system should provide the option of deposit in the main menu.

**REQ-2:**

The system should be able to specify the deposit amount.

**REQ-3:**

The system must validate the deposit information provided by the user, ensuring accuracy and completeness.

**REQ-4:**

The system should update the user's account balance in the database upon successful deposit,and provide a confirmation message.

**REQ-5:**

The system should generate a transaction record for each deposit, including details such as the deposit amount, date, and time and update the MINI statement to keep record of the deposit.

**REQ-6:**

The system should generate a proper error message if the deposition fails.

**REQ-7:**

The system should record time and money deposited to maintain the deposit history in the mini statement.

**REQ-8:**

The system should provide a back button to exit the deposit page.

**VII. Balance Enquiry Of Account:**

1. **Description And Priority**

This feature allows users to check the current balance of their account. The user can select the "Balance Enquiry" option to view the amount present in their account. The priority of this feature is high, as it provides users with real-time information about their financial status.

1. **Stimulus/Response Sequences**
   * The user will select the option of Balance Enquiry from the main menu.
   * The system then retrieves the current balance of the account from the database and displays it to the user.
   * The user may choose to go back to the main menu by clicking on the back button after viewing the current balance.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of balance enquiry in the main menu.

**REQ-2:**

The system should display the correct balance of the account to the user.

**REQ-3:**

The system should allow the user to navigate back to the main menu by clicking on the back button.

**REQ-4:**

The system should generate a proper error message if any issue is encountered in retrieving the balance information.

**VIII. Cash Withdrawal from ATM:**

1. **Description and Priority**

This feature allows users to withdraw cash from an ATM by specifying the amount they want to withdraw. The system should validate the withdrawal amount against the available balance in the user's account. If the user enters an amount greater than the available balance, the system should display a sorry message. The priority of this feature is high, as it is a fundamental aspect of ATM usage and account management.

1. **Stimulus/Response Sequences**
   * The user will select the option of “Cash Withdrawal” from the main menu.
   * The system will prompt the user to enter the amount of cash he wants to withdraw.
   * the user clicks on the “withdraw” button.
   * The system checks if the entered withdrawal amount is less than or equal to the available balance in the user's account.
   * If the amount entered by the user is less than the available balance then the system dispenses the requested cash, makes an update in the account balance in the database and will provide a confirmation message.
   * If the amount entered by the user is greater than the available balance then the system will display a message indicating insufficient balance.
   * The system will record the time and amount withdrawn and update the list in the MINI statement to keep the record of transactions.
   * The user may choose to go back to the main menu by clicking on the back button.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of “Cash withdrawal” in the main menu.

**REQ-2:**

The system should prompt the user to enter the amount he wants to withdraw.

**REQ-3:**

The system should allow the user to navigate back to the main menu by clicking on the back button.

**REQ-4:**

The system should display a message for the user that the maximum withdrawal possible is 10 000 Rs.

**REQ-5:**

The system must validate the entered amount against the account balance.

**REQ-6:**

The system should dispense the amount entered if it is valid.

**REQ-7:**

The system should update the account balance after the withdrawal and should display a confirmation message.

**REQ-8:**

The system should record the time and amount withdrawn and update the list in the MINI statement to keep the record of transactions.

**REQ-9:**

The system should generate a display message that the balance is not sufficient if the user enters more than the available balance.

**IX. Fast Cash for Transaction:**

1. **Description and Priority**

This feature enables users to quickly withdraw a predefined amount of cash using the "Fast Cash" option. Users are given a set of choices (e.g., 500, 1000, 2000, 5000, 10000) to select from, providing a convenient and efficient way to withdraw common cash denominations. The priority of this feature is high, as it enhances the user experience and streamlines the cash withdrawal process.

1. **Stimulus/Response Sequences**
   * The user will select the option of “Fast Cash” from the main menu.
   * The system will prompt the user to select from a list of predefined cash withdrawal amounts which will include: 500, 1000, 2000, 5000, 10000.
   * the user selects and then the system will validate that the selected amount is less than or equal to the available balance in the user’s account.
   * If the amount selected by the user is less than the available balance then the system dispenses the requested cash, makes an update in the account balance in the database and will provide a confirmation message.
   * The system should record the time and amount withdrawn and update the list in the MINI statement to keep the record of transactions.
   * If the amount entered by the user is greater than the available balance then the system will display a message indicating insufficient balance.
   * The user may choose to go back to the main menu by clicking on the back button.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of “Fast Cash” in the main menu.

**REQ-2:**

The system should prompt the user to select the amount he wants to withdraw from the given list.

**REQ-3:**

The system should allow the user to navigate back to the main menu by clicking on the back button.

**REQ-4:**

The system must validate the selected amount against the account balance.

**REQ-5:**

The system should dispense the amount selected if it is valid.

**REQ-6:**

The system should update the account balance after the withdrawal and should display a confirmation message.

**REQ-7:**

The system should record the time and amount withdrawn and update the list in the MINI statement to keep the record of transactions.

**REQ-8:**

The system should generate a display message that the balance is not sufficient if the user selects more than the available balance.

**X. Change of PIN:**

1. **Description and Priority**

This feature allows users to change the PIN of their account for enhanced security. The user initiates the process by signing in and selecting the "Change PIN" option. The system then guides the user through entering a new PIN, confirming it, and updating the database with the changes. The priority of this feature is high, as it contributes to the overall security of the user's account.

1. **Stimulus/Response Sequences**
   * The user signs in and selects the option of “PIN change” from the main menu.
   * The system will prompt the user to enter a new pin.
   * The system will ask the user to reenter the new PIN to ensure accuracy.
   * The user will click on the change button.
   * If the PIN entered and the reentered PIN match, the system will update the PIN information of the particular account in the database.
   * The system displays a confirmation message, indicating that the PIN change was successful.
   * The user may choose to go back to the main menu by clicking on the back button.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of “PIN change” in the main menu.

**REQ-2:**

The system should prompt the user to enter the new PIN

**REQ-3:**

The system should ask the user to re enter the PIN to confirm the change.

**REQ-4:**

The system should update the user's account in the database with the newly entered PIN after validating the entered and reentered PIN.

**REQ-5:**

The system should display an error message if the two entered PINs don't match.

**REQ-6:**

A confirmation message should be displayed to the user, indicating the successful change of the PIN.

**REQ-7:**

The system should allow the user to return to the main menu by clicking the Back button.

**XI. Mini Statement To Keep Record Of Cash Withdrawn And Deposited:**

1. **Description And Priority**

This feature allows users to request a MINI Statement to view a record of their cash deposits and withdrawals. The statement includes details such as the day, month, time (hours, minutes, seconds), year, whether it was a withdrawal or deposit, and the amount of cash involved. The priority of this feature is high, as it provides users with a convenient way to keep track of their recent financial transactions.

1. **Stimulus/Response Sequences**

* .
  + The user signs in and selects the option of “MINI statement” from the main menu.
  + The system retrieves and displays a slip showing records of the user's cash deposits and withdrawals.
  + Each entry in the statement includes the day, month, time (hours, minutes, seconds), year, transaction type (withdrawal or deposit), and the amount of cash involved.
  + The user may choose to go back to the main menu by clicking on the back button.

1. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of “MINI statement” in the main menu.

**REQ-2:**

The system should promptly retrieve and display a slip showing records of the user's recent cash deposits and withdrawals

**REQ-3:**

The system must give a statement that should include details such as the day, month, time (hours, minutes, seconds), year, transaction type (withdrawal or deposit), and the amount of cash involved.

**REQ-4:**

The system should organize the statement entries in chronological order, with the latest transactions appearing first.

**REQ-5:**

The system should allow the user to return to the main menu by clicking the Back button.

**XII. Exit to log out the account:**

1. **Description and Priority**

This feature provides users with the option to log out of their account by selecting the "Exit" button. Logging out is crucial for security and privacy, ensuring that unauthorized individuals cannot access the user's account. The priority of this feature is high, as it contributes to the overall security and user control of the system.

1. **Stimulus/Response Sequences**
   * The user signs in and selects the option of “Exit” from the main menu.
   * After logging out, the system returns to the login screen, requiring the user to sign in again to access their account.
2. **Functional Requirements**

**REQ-1:**

The system should provide a clear option of “Exit” in the main menu.

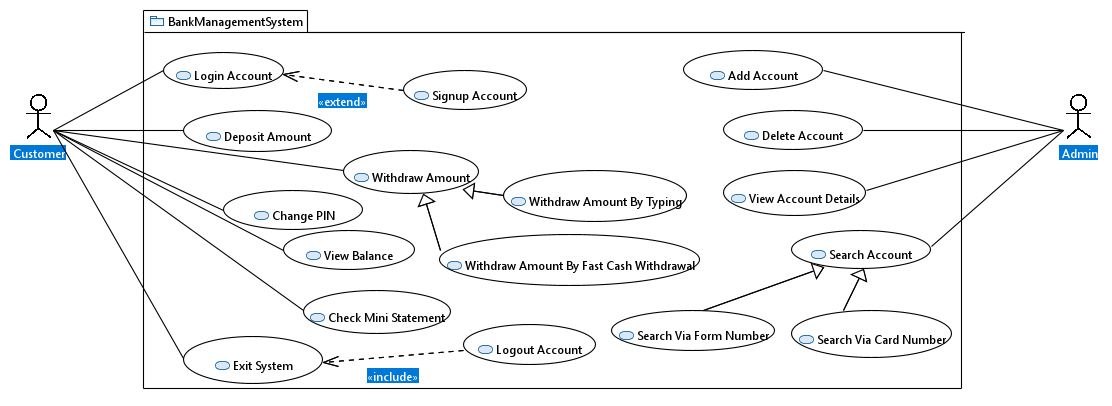
**REQ-2:**

The system should return to the login screen after logging out, requiring the user to sign in again to access their account

**REQ-3:**

The system should not allow the user to access any account-related functionalities after logging out until they successfully sign in again.

* 1. **Use Cases**

****

* + 1. **Login Account**

| **Use case Id: 001** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 001. | | |
| **Actors:**  Customer | | | | |
| **Feature:** Account Authentication | | | | |
| **Pre-condition:** | | - System is running.  - Customer already has a registered account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer enters the correct card number and PIN, and clicks on the “SIGN IN” button. | | | System validates credentials. System logs the customer into their account if all credentials are valid. |
| **Alternate Scenarios:** | | | | |
| **1a:** Customer enters the wrong card number and/or PIN. The system displays the error message, and redirects the customer back to the login/signup page. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The customer is successfully logged into their account. The customer gains access to their account. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.2. Signup Account**

| **Use case Id: 002** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 002 | | |
| **Actors:**  Customer | | | | |
| **Feature:** Account Registration | | | | |
| **Pre-condition:** | | - The system is running.  - The customer does not have an existing bank account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer clicks on the “SIGN UP” button. | | | System presents registration forms, prompting the user to enter personal details, additional details (occupation, education), and bank details. |
| **2.** | Customer fills out the registration forms, and clicks on the submit button. | | | The system creates a new account for the customer if all the details are valid and unique. |
| **Alternate Scenarios:**. | | | | |
| **1a:** Customer enters invalid details, and/or leaves field(s) empty. The system displays the appropriate error message.  **2a:** Customer clicks on the “Cancel” button. The system redirects the customer back to the Login/Signup Page. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Customer successfully created a new account. The customer can now login their account using their card number and PIN. | | | |
| **Use Case Cross referenced** | | | Login Account | |

**4.2.3. Deposit Amount**

| **Use case Id: 003** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 003 | | |
| **Actors:** Customer | | | | |
| **Feature:** Account Deposit | | | | |
| **Pre-condition:** | | - The customer is logged into their account.  - The customer has funds available for deposition. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer selects the “DEPOSIT” option from the main menu. | | | System prompts the customer to enter the amount they wish to deposit. |
| **2.** | Customer enters the amount to deposit and clicks on the “SUBMIT” button. | | | System validates the amount entered by the customer. |
| **Alternate Scenarios:** | | | | |
| **1a:** Customer does not have sufficient funds available to deposit. The system displays an appropriate error message. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Deposit transaction is successfully completed. The customer receives a confirmation message. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.4 Withdraw Amount By Typing**

| **Use case Id: 004** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 004 | | |
| **Actors:** Customer | | | | |
| **Feature:** Account Withdrawal | | | | |
| **Pre-condition:** | | - The customer is logged into their account.  - Sufficient balance is available in the account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer selects the “WITHDRAW” option from the main menu. | | | System prompts the customer to enter the amount they wish to withdraw. |
| **2.** | Customer enters the amount to withdraw and clicks on the “SUBMIT” button. | | | System validates the amount entered by the customer. |
| **Alternate Scenarios:** | | | | |
| **1a:** Customer enters the amount that is greater than the balance available in the bank account. The system displays an appropriate error message.  **2a:** Customer enters an amount that is greater than 10000. The system displays an appropriate error message. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The specified amount is successfully withdrawn from the customer’s account. The customer’s account balance is updated based on the amount withdrawn. | | | |
| **Use Case Cross referenced** | | | Withdraw Amount | |

**4.2.5. Withdraw Amount By Fast Cash Withdrawal**

| **Use case Id: 005** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 005 | | |
| **Actors:**  Customer | | | | |
| **Feature:** Account Withdrawal | | | | |
| **Pre-condition:** | | - The customer is logged into their account.  - Sufficient balance is available in the account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer selects the “FAST CASH” option from the main menu. | | | System prompts the customer to select the amount they wish to withdraw from a list of predefined amounts. |
| **2.** | Customer selects the amount to withdraw from a list of predefined amounts. | | | System validates the amount entered by the customer. |
| **Alternate Scenarios:** | | | | |
| **1a:** Customer selects the amount that is greater than the balance available in the bank account. The system displays an appropriate error message. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The specified amount is successfully withdrawn from the customer’s account. The customer’s account balance is updated based on the amount withdrawn. | | | |
| **Use Case Cross referenced** | | | Withdraw Amount | |

**4.2.6. Change PIN**

| **Use case Id: 006** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 006 | | |
| **Actors:** Customer | | | | |
| **Feature:** PIN Management | | | | |
| **Pre-condition:** | | - The customer is logged into their account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer selects the “CHANGE PIN” option from the main menu. | | | System prompts the customer to enter their current PIN for verification. |
| **2.** | Customer enters the current PIN, and then the desired new PIN, then clicks the “CONFIRM” button. | | | If the current PIN is verified, the system changes the PIN, and then displays a confirmation message. |
| **Alternate Scenarios:** | | | | |
| **1a:** The system enters the incorrect PIN. The system displays an error message and prompts the user to re-enter the current PIN. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Customer’s PIN is successfully changed. | | | |
| **Use Case Cross referenced** | | | none | |

**4.2.7. View Balance**

| **Use case Id: 007** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 007 | | |
| **Actors:** Customer | | | | |
| **Feature:** Account Balance Display | | | | |
| **Pre-condition:** | | - The Customer is logged into their account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer selects the “BALANCE ENQUIRY” button in the main menu. | | | System displays the balance available on the customer's bank account. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The customer views their account balance displayed by the system. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.8. Check Mini Statement**

| **Use case Id: 008** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 008 | | |
| **Actors:** Customer | | | | |
| **Feature:** Mini Statement Retrieval | | | | |
| **Pre-condition:** | | - The Customer is logged into their bank account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The customer selects the “MINI STATEMENT” button in the main menu. | | | System retrieves the customer account’s mini statement. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The customer views their mini statement displayed by the system. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.9 Exit Account**

| **Use case Id: 009** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 009 | | |
| **Actors:**  Customer | | | | |
| **Feature:** System Exit | | | | |
| **Pre-condition:** | | - The customer is logged into their account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The customer selects the “EXIT” button on the main menu | | | The system prompts the user to exit the system. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Customer successfully exits the system. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.10 Logout Account**

| **Use case Id: 010** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 010 | | |
| **Actors:**  Customer | | | | |
| **Feature:** Account Logout | | | | |
| **Pre-condition:** | | - The customer is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | The customer clicks on the “EXIT” button on the main menu. | | | The system prompts the user to exit the system, and then logs out the customer. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | The customer successfully exits, and logs out of their bank account. | | | |
| **Use Case Cross referenced** | | | Exit System | |

**4.2.11. Add Account**

| **Use case Id: 011** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 011 | | |
| **Actors:**  Admin | | | | |
| **Feature:** Account Creation | | | | |
| **Pre-condition:** | | - The admin is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin fills out the required fields (eg: Customer name, Account type etc) | | | The system creates a new account and stores it into the database. |
| **Alternate Scenarios:** | | | | |
| **1a:** Admin fills out non-unique credentials. The system database displays an error, prompting the admin to correct the input. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Admin successfully adds a new account to the system database. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.12. Delete Account**

| **Use case Id: 012** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 012 | | |
| **Actors:**  Admin | | | | |
| **Feature:** Account Deletion | | | | |
| **Pre-condition:** | | - Admin is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin enters the card number, or form number of the account, and then clicks on the “DELETE” button. | | | System verifies the entered account’s details, and deletes the account if the account exists, and is eligible for deletion. |
| **Alternate Scenarios:** | | | | |
| **1a:** Admin enters incorrect card number, or form number. The system gives an error, and redirects the admin to re-enter the correct card number, or form number. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Account is successfully deleted. Account is removed from the system, and is no longer accessible. | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.13. View Account Details**

| **Use case Id: 013** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 013 | | |
| **Actors:**  Admin | | | | |
| **Feature:** Account Details Viewing | | | | |
| **Pre-condition:** | | - The admin is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin runs the appropriate query in the database, which views multiple accounts. | | | The system database displays the registered accounts. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Admin views the account details. | | | |
| **Use Case Cross referenced** | | | None | |

**4.2.14. Search Account Via Form Number**

| **Use case Id: 014** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 014. | | |
| **Actors:**  Admin | | | | |
| **Feature:** Search Account | | | | |
| **Pre-condition:** | | - Admin is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin runs the appropriate query in the database, and enters the form number in the field. | | | System database displays the appropriate account with that form number. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Admin views the account details associated with the entered form number. | | | |
| **Use Case Cross referenced** | | | Search Account | |

**4.2.15. Search Account Via Card Number**

| **Use case Id: 015** | | | | |
| --- | --- | --- | --- | --- |
| **Use case Id:** | | 015 | | |
| **Actors:**  Admin | | | | |
| **Feature:** Search Account | | | | |
| **Pre-condition:** | | - Admin is logged into the system. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin runs the appropriate query in the database, and enters the card number in the field. | | | System database displays the appropriate account with that card number. |
| **Alternate Scenarios:** | | | | |
| None | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Admin views the account details associated with the entered form number. | | | |
| **Use Case Cross referenced** | | | Search Account | |

1. **Non-functional Requirements**
   1. **Performance Requirements**

* **Account-Based Transactions Response Time:** The system should respond within 3 seconds for routine transactions.
* **Simultaneous User Handling:** The system should be able to support at least 50 concurrent users without a significant loss in system performance.
* **Database Query Performance:** Database queries for routine operations, such as, balance enquiry, checking mini statement, should have a response time of maximum 2 seconds.
  1. **Safety Requirements**
* **Data Retrieval:** In case of loss of data, the system is able to retrieve user data from the database.

* **Confirmation Popups:** The system would provide confirmation popups, to ensure that the user becomes aware of the changes made by the user.

* **Data Integrity:** The system ensures user data integrity, thereby, preventing unauthorized access, and modifications.

* **Error Handling:** The system is able to handle errors in case of unexpected events.
  1. **Security Requirements**
* **Data Encryption:** All the communication between the system and the database, with the user is encrypted to safeguard personal information, and relevant account details.
* **User Authentication:** Accessing the account requires secure authentication mechanisms, through login/signup functions and modules.
  1. **User Documentation**

1. **User manuals:** Manuals that guide by providing step by step instructions to use the features of our BMS.
2. **Tutorials:** Interactive tutorials to help users in learning and mastering specific aspects of the BMS. Video demonstrations and guided exercises will be used to enhance user experience.
3. **Delivery formats:** Both digital and print formats will be used for user documentation. Digital formats may include pdf documents, online help and video tutorials. Print formats may include physical user manuals for those who prefer hard copies.
4. **Online help:** Interactive online help will be provided to all the users for the BMS interface that will be intended to be easily accessible. This will provide guidance for specific tasks and error resolution.

1. **References**

* **Project References:**

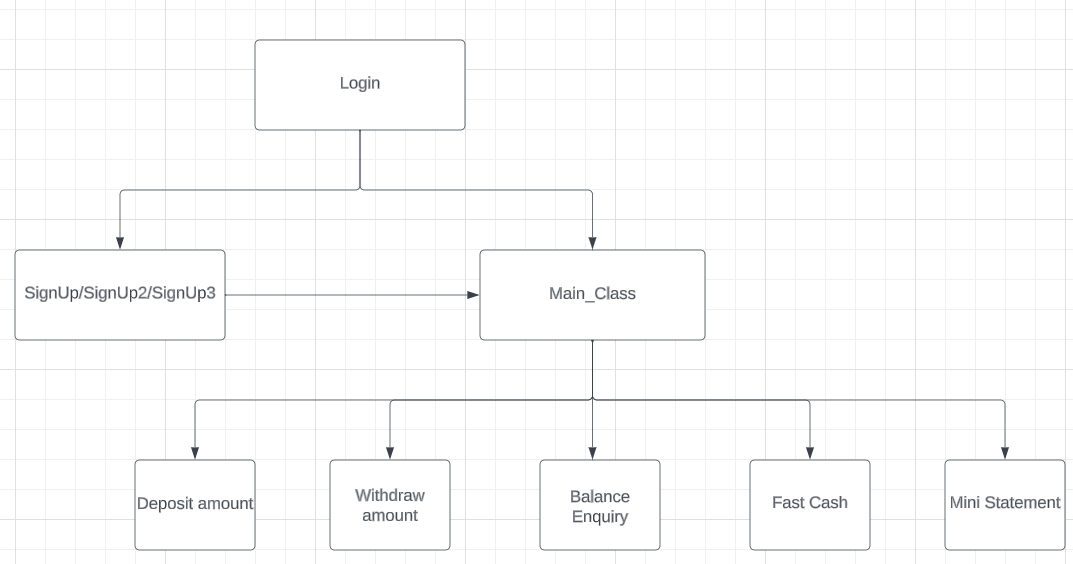
1. Code For Interview – <https://www.youtube.com/@codeforinterview>
2. Learning Never Ends – <https://www.youtube.com/@LearningNeverEnds786>

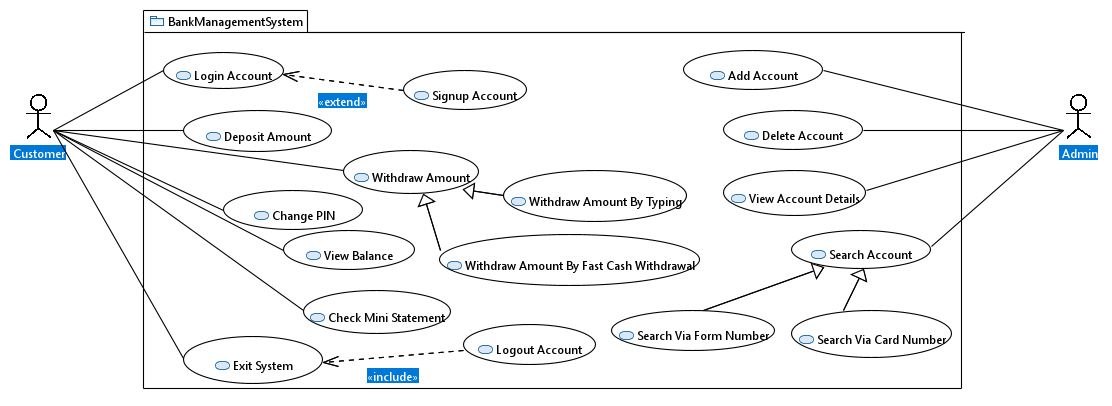
* **Documentation References:**

1. KrazyTech – https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database
2. **Appendices**

* **Appendix A: Glossary**

| BMS | Bank Management System |
| --- | --- |
| BS | Banking System |
| UI | User Interface |
| OS | Operating System |
| HDD / SSD | Hard-Disk Drive / Solid-State Drive |
| GUI | Graphical User Interface |
| HTTP | Hypertext Transfer Protocol |

* **Appendix B: Analysis Models**

****

* **Appendix C: To Be Determined List**
* Authenticating the regulatory standards that the system needs to comply with, as well as the specific requirements for each standard.
* Specifying the process for the integration of the bank management system with external existing banking infrastructure components.