Low Level Design

ATM Interface in Java (Console Based Application)

|  |  |
| --- | --- |
| Written By | Sidduganesh Musa |
| Document Version | 0.1 |
| Last Revised Date | 28/07/2023 |

**DOCUMENT CONTROL**

**Change Record :**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.1 | 28/07/2023 | Sidduganesh Musa | Introduction, Architecture, Architecture Description,  Test Cases. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reviews :**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Reviewer | Comments |
|  |  |  |  |
|  |  |  |  |

**Approval Status :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Review Date | Reviewed By | Approved By | Comments |
|  |  |  |  |  |

Contents

[**1.Introduction** 4](#_Toc141613357)

[1.1. What is Low-Level design document? 4](#_Toc141613358)

[1.2. Scope 4](#_Toc141613359)

[**2.Architecture** 5](#_Toc141613360)

[**3.Architecture Description** 6](#_Toc141613361)

[3.1 User Inputs: 6](#_Toc141613362)

[3.2 Authentication: 6](#_Toc141613363)

[3.3 Menu: 6](#_Toc141613364)

[3.4 Operations: 7](#_Toc141613365)

[**4.Test Cases:** 8](#_Toc141613366)

# **1.Introduction**

## 1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for ATM Interface in Java(Console Based Application). LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## 1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

# **2.Architecture**

START

USER INPUTS

AUTHENTICATION

MENU

OPERATIONS

END

# **3.Architecture Description**

## 3.1 User Inputs:

In the ATM interface, the user provides various inputs through the console-based application.

These inputs include:

Account Number: The unique identification number associated with the user's bank account.

PIN (Personal Identification Number): A secret numeric code used for user authentication and account access.

Transaction Choices: The user selects from a menu of options that include viewing transactions, making deposits, withdrawals, checking account balance, transferring funds, and quitting the application.

Transaction Amount: When performing deposit, withdrawal, or fund transfer operations, the user enters the specific amount involved in the transaction.

Destination Account (for Transfer): In the fund transfer operation, the user specifies the destination account number to transfer funds.

## 3.2 Authentication:

After the user enters their Account Number and PIN, the ATM interface verifies the provided credentials for user authentication. The authentication process involves comparing the entered Account Number and PIN with the stored values in the system. If the credentials match an existing account, the user is granted access to perform various banking operations. Otherwise, an error message is displayed, and the user is prompted to re-enter the credentials.

## 3.3 Menu:

Once the user is successfully authenticated, they are presented with a menu that displays the available banking operations they can perform.

The menu typically includes the following options:

Show Transactions: View the transaction history for the user's account.

Deposit: Deposit funds into the user's account.

Withdraw: Withdraw cash from the user's account.

Show Balance: Check the current account balance.

Transfer: Transfer funds from the user's account to another account.

Quit: Exit the ATM interface and end the session.

The user can select an option by entering the corresponding choice using the console input.

## 3.4 Operations:

Based on the user's menu selection, the ATM interface executes the chosen operation.

Each operation corresponds to a specific action:

Show Transactions: The ATM interface retrieves the transaction history associated with the user's account and displays it on the console.

Deposit: The user is prompted to enter the amount they wish to deposit. The ATM interface updates the account balance accordingly and records the deposit transaction in the transaction history.

Withdraw: The user is prompted to enter the amount they wish to withdraw. The ATM interface validates the withdrawal amount against the account balance to ensure sufficient funds are available. If the withdrawal is valid, the account balance is updated, and the withdrawal transaction is recorded.

Show Balance: The ATM interface fetches the account balance and displays it to the user on the console.

Transfer: The user is prompted to enter the amount they want to transfer and the destination account number. The ATM interface verifies the transfer amount and destination account validity, updates both the sender and receiver's account balances, and records the transfer transactions for both accounts.

Quit: The user chooses to end the session, and the ATM interface saves transaction history and account data, displays a farewell message, and terminates the application.

Throughout the operations, the ATM interface handles user input validation and error handling, ensuring a smooth and secure user experience during the banking transactions.

# **4.Test Cases:**

Test Case 1: Valid Authentication

Input: User Id: "user123", PIN: "1234"

Expected Output: Successfully authenticated and presented with the Main Menu options.

Test Case 2: Invalid Authentication

Input: User Id: "54321", PIN: "1234"

Expected Output: Authentication failed. Display an error message and prompt for credentials again.

Test Case 3: Show Transactions

Input: Select "Show Transactions" option from the Main Menu

Expected Output: Display the transaction history for the user's account.

Test Case 4: Deposit

Input: Select "Deposit" option from the Main Menu, Deposit Amount: 1000

Expected Output: Account balance updated by adding 1000, and the deposit transaction recorded.

Test Case 5: Withdraw

Input: Select "Withdraw" option from the Main Menu, Withdrawal Amount: 500

Expected Output: Account balance updated by deducting 500, and the withdrawal transaction recorded.

Test Case 6: Show Balance

Input: Select "Show Balance" option from the Main Menu

Expected Output: Display the current account balance.

Test Case 7: Transfer (Valid)

Input: Select "Transfer" option from the Main Menu, Transfer Amount: 300, Destination Account: "AC002"

Expected Output: Both sender and receiver account balances updated accordingly, and transfer transactions recorded for both accounts.

Test Case 8: Transfer (Insufficient Funds)

Input: Select "Transfer" option from the Main Menu, Transfer Amount: 1000, Destination Account: "AC003"

Expected Output: Display an error message indicating insufficient funds in the sender's account.

Test Case 9: Invalid Option

Input: Select an invalid option (e.g., "9") from the Main Menu

Expected Output: Display an error message indicating an invalid choice and prompt the user to select a valid option.

Test Case 10: Quit

Input: Select "Quit" option from the Main Menu

Expected Output: Save transaction history and account data, display a farewell message, and terminate the application.

Test Case 11: Stress Test

Input: Perform multiple consecutive operations (e.g., deposits, withdrawals, transfers) in rapid succession to test application responsiveness and stability.

Expected Output: The application should handle all operations without errors or crashes and maintain data consistency.

These test cases cover various scenarios, including valid and invalid authentication, different banking operations, error handling, and application termination. By running these test cases, you can ensure the ATM Interface in Java (Console-based-Application) functions correctly and delivers an efficient and secure banking experience to users.