

# RIPHAH INTERNATIONAL UNIVERSITY



**Object Oriented Programming**  
**Spring 2023**

## **Bank Record System**

### **Project Team**

<b>Name of Students</b>	<b>Sap ID</b>	<b>Program</b>	<b>Valid Email Address</b>
Manahil Habib	47876	BSSE	<a href="mailto:47876@students.riphah.edu.pk">47876@students.riphah.edu.pk</a>
Nagarash Fateh	44815	BSSE	<a href="mailto:44815@students.riphah.edu.pk">44815@students.riphah.edu.pk</a>

**Date of Submission**  
6/4/2023

Table of Contents

Artifact # 1 Project Proposal..... 5

Artifact # 2 Screens..... 9

# **Artifact # 1**

## **Project Proposal**

# Introduction of the Project

**Project Title:** Bank Record System

## **Introduction:**

The Bank Record System is a web-based application designed to store the information of account type, account opening form, deposit, withdrawal, and searching the transaction information, transaction report, individual account opening form, and group account. Information about interest rates, statistical summaries of account types, and transaction reports are all shown in the records. This aids in offering the user a versatile solution. When compared to the current system, this one has a lower mistake probability. The system will be developed using Java.

## **Problem Statement:**

Create a Bank Record System to manage all bank records using java.

## **Proposed Solution:**

We will code Bank Record System in Java through Inheritance, Polymorphism, Abstraction and Encapsulation using Object Oriented Programming.

## **Scope of the Project:**

In order to keep track of everything in the bank, the banking system needs to maintain a few records, thus a software program is needed to make the task easier. For instance, the banking system is responsible for maintaining the value of INR and other currencies on a global scale.

## **Modules Description:**

1. Admin
2. Manager
3. Client

## **Modules Description:**

- **Admin:**
  1. Client Details (Search, Update and Delete)
  2. Registration (Manager and Client)
  3. Display Records
- **Manager:**
  1. Registration (Name, ID, Passcode)
  2. Create Accounts

3. Input Records (Create, Update, and Delete)
- **Client:**
  1. Registration (Name, ID, Passcode)
  2. View Account
  3. Contact Us
  4. Feedback

## **Inheritance**

### **Definition:**

Inheritance is the procedure or mechanism of acquiring all the properties and behavior of one class to another, i.e., acquiring the properties and behavior of a child class from the parent class.

### **Explanation:**

In a bank record system, inheritance can be utilized to create different types of bank accounts that share common functionalities. For instance, a "SavingsAccount" class and a "CheckingAccount" class can inherit from a common "BankAccount" class, which may contain common methods like deposit, withdraw, and getBalance, and then add their own specialized methods and data members.

## **Polymorphism**

### **Definition:**

The word polymorphism means having multiple forms. Polymorphism enables objects of different classes to be treated as if they were of the same class.

### **Explanation:**

In a bank record system, polymorphism can be applied to allow different types of bank accounts to be processed uniformly. For example, a "BankAccount" interface or an abstract class can define common methods like deposit, withdraw, and getBalance, which can be implemented differently in different account types. This way, different account objects can be treated uniformly, regardless of their specific types.

## **Abstraction**

### **Definition:**

Data Abstraction is technique whose feature provides us the capability of differentiating essential details that need to be displayed to the user.

### **Explanation:**

In a bank record system, abstraction can be utilized to create abstract classes or interfaces that define common methods and properties without providing any implementation. Concrete classes that inherit from these abstract classes or implement these interfaces can provide the actual implementation. This way, the complexities of

the underlying system can be abstracted, and a simplified interface can be exposed to the users of the bank record system.

## **Encapsulation**

### **Definition:**

Encapsulation gives us the ability to make variables of a class keep hidden from all other classes of that program or namespace.

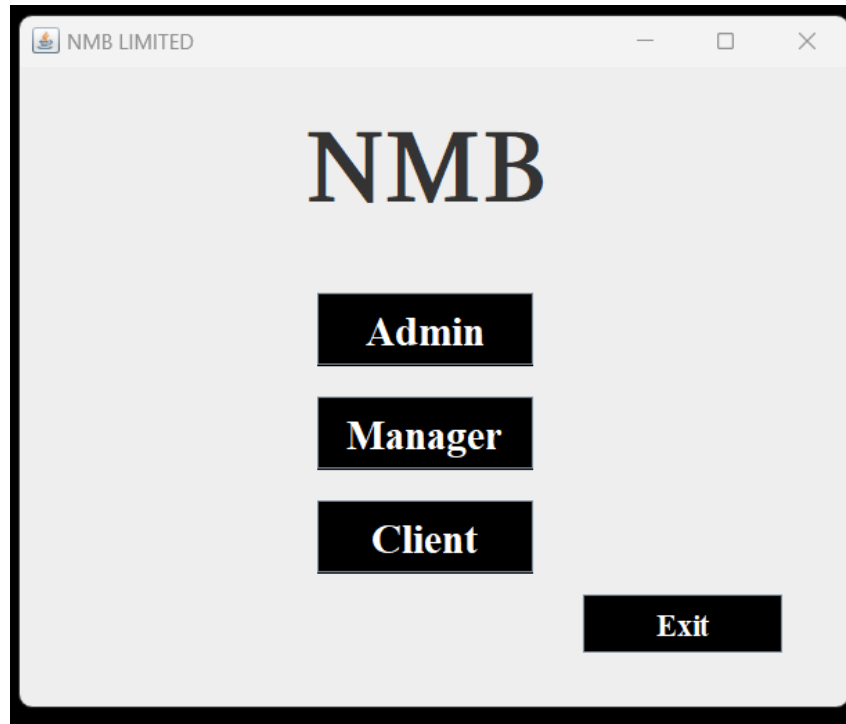
### **Explanation:**

In a bank record system, encapsulation can be implemented by using classes and objects to encapsulate data related to bank accounts, transactions, and customers. For example, a class "BankAccount" can have private data members such as account number, balance, and customer information, which can be accessed through methods like deposit, withdraw, and getBalance.

# **Artifact # 2**

## **Screens**

- Main Menu



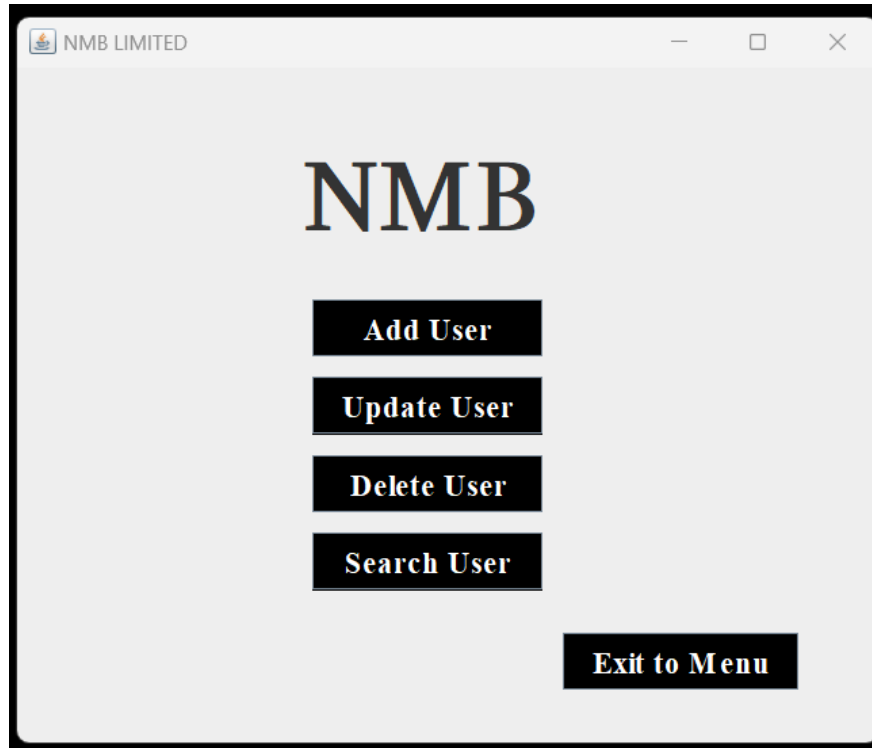
- Admin

Screen 01 (Reisteration)

A screenshot of a software application window titled "NMB LIMITED". The window has a light gray background and a black border. At the top center, the text "NMB" is displayed in a large, bold, serif font. Below this, there are two input fields. The first field is labeled "Card No." and the second field is labeled "Pin". Both labels are in a bold, serif font. Below the input fields, there are two black rectangular buttons with white text: "Sign up" and "Back".



Screen 02 (Admin Menu)



NMB LIMITED

# NMB

**Add User**

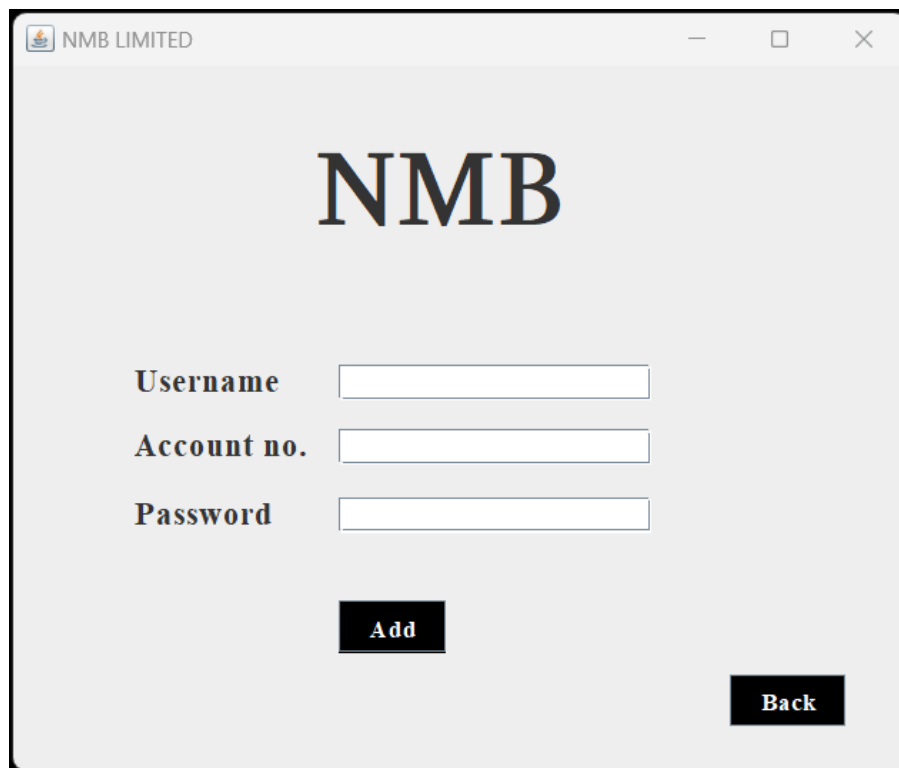
**Update User**

**Delete User**

**Search User**

**Exit to Menu**

Screen 03 (Add User)



NMB LIMITED

# NMB

**Username**

**Account no.**

**Password**

**Add**

**Back**

Screen 04 (Update User)



The screenshot shows a web application window titled "NMB LIMITED". The main heading is "NMB". Below it, there are three input fields labeled "Username", "Account no.", and "Password". A black button labeled "Update" is positioned below the input fields. A black button labeled "Back" is located in the bottom right corner.

NMB LIMITED

# NMB

Username

Account no.

Password

Update

Back

Screen 05 (Delete User)



The screenshot shows a web application window titled "NMB LIMITED". The main heading is "NMB". Below it, there are three input fields labeled "Username", "Account no.", and "Password". A black button labeled "Delete" is positioned below the input fields. A black button labeled "Back" is located in the bottom right corner.

NMB LIMITED

# NMB

Username

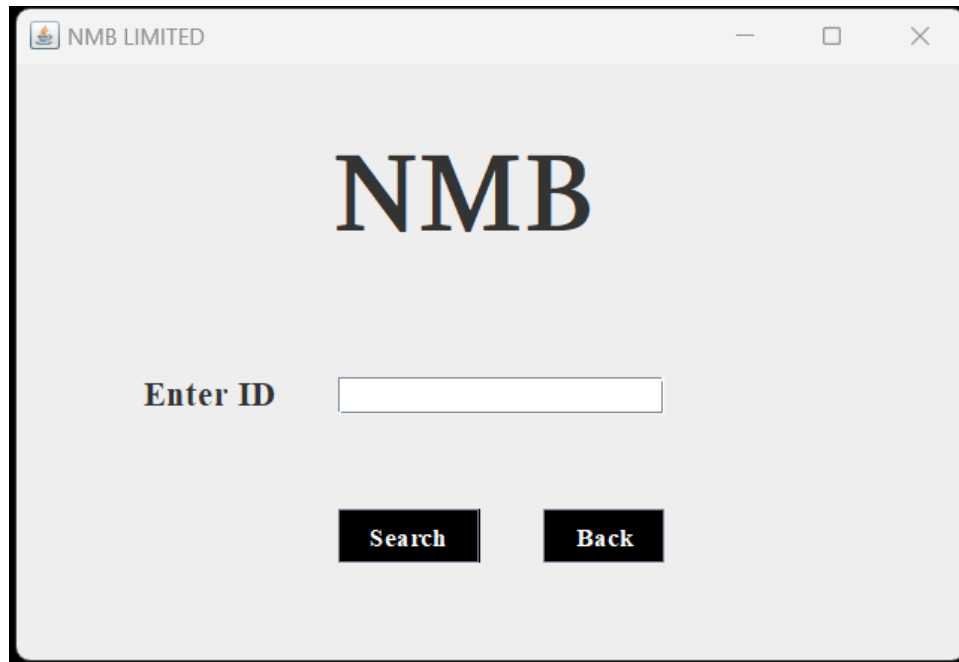
Account no.

Password

Delete

Back

Screen 06 (Search User)



The screenshot shows a web application window titled "NMB LIMITED". The main heading is "NMB". Below it, there is a label "Enter ID" followed by a text input field. At the bottom, there are two buttons: "Search" and "Back".

**NMB LIMITED**

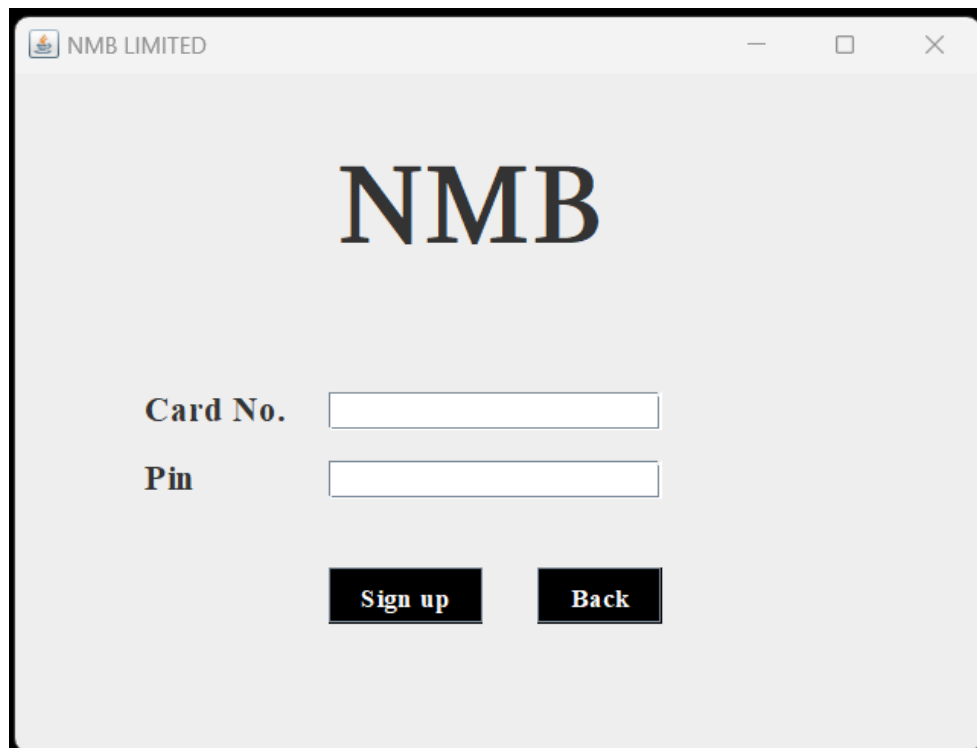
# NMB

Enter ID

**Search** **Back**

- Manager

Screen 01 (Registration)



The screenshot shows a web application window titled "NMB LIMITED". The main heading is "NMB". Below it, there are two labels: "Card No." and "Pin", each followed by a text input field. At the bottom, there are two buttons: "Sign up" and "Back".

**NMB LIMITED**

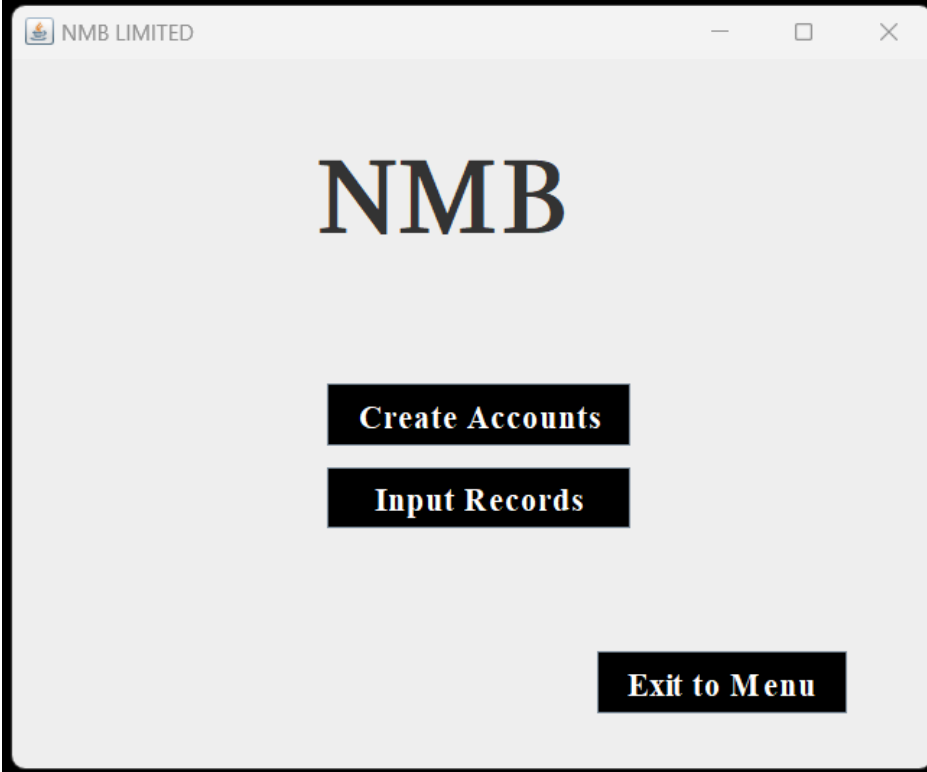
# NMB

Card No.

Pin

**Sign up** **Back**

Screen 02 (Manager Menu)



NMB LIMITED

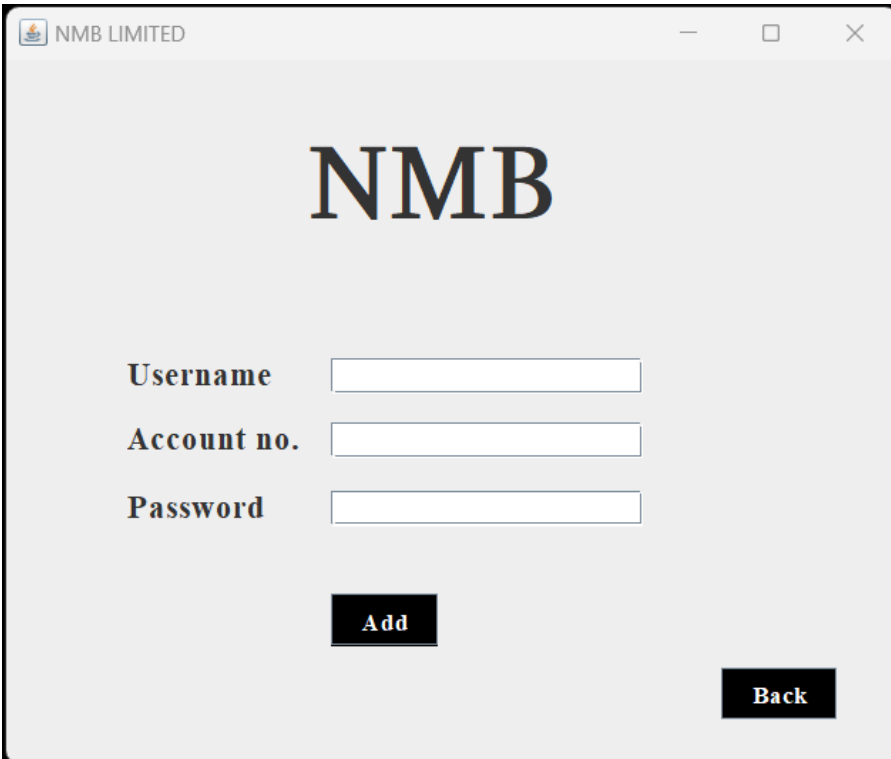
# NMB

**Create Accounts**

**Input Records**

**Exit to Menu**

Screen 03 (Create Accounts)



NMB LIMITED

# NMB

**Username**

**Account no.**

**Password**

**Add**

**Back**


Screen 04 (Input Records)

The screenshot shows a window titled "NMB LIMITED" with a light gray background. In the center, the "NMB" logo is displayed in a large, dark serif font. Below the logo, three black buttons with white text are stacked vertically: "Debit", "Deposit", and "Interest". In the bottom right corner, there is a black button with white text labeled "Back".

Screen 04 (Debit)

The screenshot shows a window titled "NMB LIMITED" with a light gray background. In the center, the "NMB" logo is displayed in a large, dark serif font. Below the logo, there are three input fields with labels to their left: "Account No.", "Holder Name", and "Debit Amount". Each label is in a bold, dark serif font, and each input field is a white rectangle with a thin gray border. Below the input fields, there are three black buttons with white text: "Done" is centered, "Check Balance" is on the left, and "Back" is on the right.

Screen 05 (Deposit)

NMB LIMITED

# NMB

**Account No.**


**Holder Name**

**Deposit Amount**

**Done**

**Check Balance** **Back**

Screen 06 (Interest)

NMB LIMITED

# NMB

**Account No.**

**Holder Name**

**Balance**

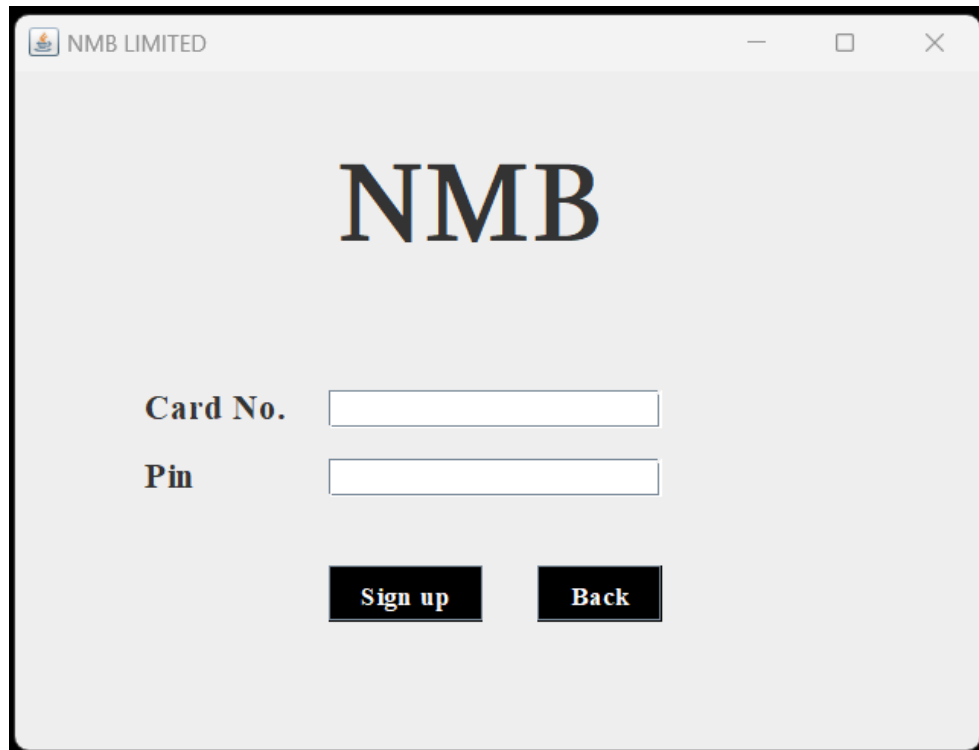
**Interest Rate**

**Done**

**Check Balance** **Back**

- Client

### Screen 01 (Registration)



The image shows a registration screen for NMB Limited. At the top left is the NMB Limited logo and name. The title "NMB" is centered in a large, bold, serif font. Below the title are two input fields: "Card No." and "Pin". Below these fields are two buttons: "Sign up" and "Back".

NMB LIMITED

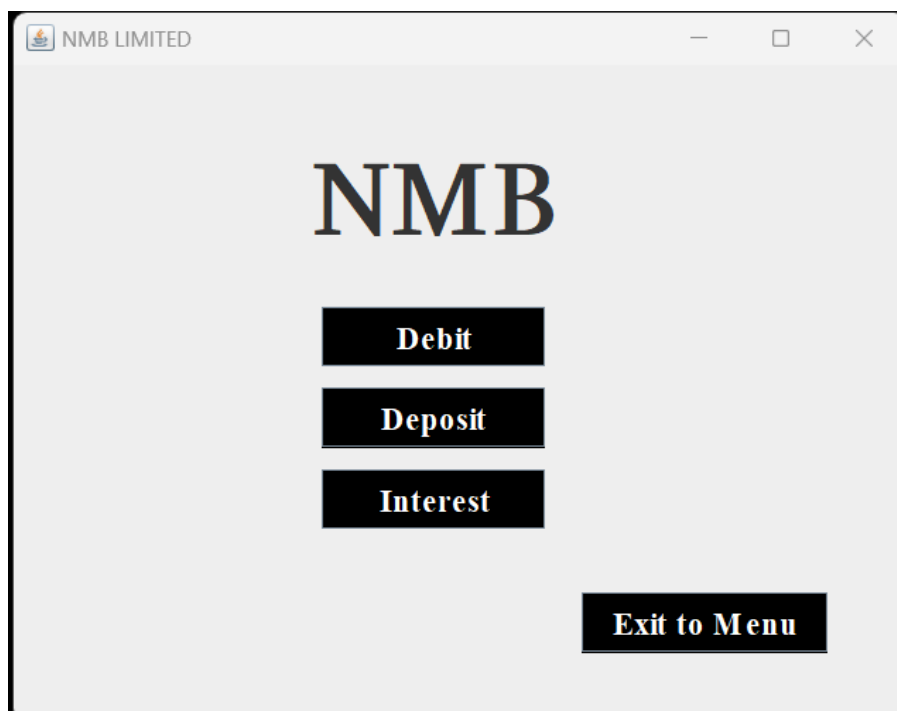
# NMB

Card No.

Pin

**Sign up** **Back**

### Screen 02 (Client Menu)



The image shows a client menu screen for NMB Limited. At the top left is the NMB Limited logo and name. The title "NMB" is centered in a large, bold, serif font. Below the title are three buttons: "Debit", "Deposit", and "Interest". At the bottom right is a button labeled "Exit to Menu".

NMB LIMITED

# NMB


**Debit**

**Deposit**

**Interest**

**Exit to Menu**

Screen 03 (Debit)

NMB LIMITED

# NMB

**Account No.**


**Holder Name**

**Debit Amount**

**Done**

**Check Balance** **Back**

Screen 04 (Deposit)

NMB LIMITED

# NMB

**Account No.**

**Holder Name**

**Deposit Amount**

**Done**

**Check Balance** **Back**



## Screen 05 (Interest)



A screenshot of a web application window titled "NMB LIMITED". The window displays the "NMB" logo at the top. Below the logo, there are four input fields with labels: "Account No.", "Holder Name", "Balance", and "Interest Rate". At the bottom of the form, there are three buttons: "Done", "Check Balance", and "Back".

**NMB**

Account No.

Holder Name

Balance

Interest Rate

**Done**

**Check Balance** **Back**

- Exiting from GUI

