THIRD YEAR B.TECH. SEMESTER MINI PROJECT REPORT

A report submitted in partial fulfilment of Computer Programming Course

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Enrollment Number	Name
BT20ECE011	Mundru Abhiram
BT20ECE016	Om Manoj Gupta
BT20ECE028	Jagriti Isha
BT20ECE075	Nekkanti Guna Sai Kiran
BT20ECE112	Pallavi

Under the

Supervision of

Mr.Gaurav Malode, (A.Assitance Professor)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR

(An Institution of National Importance by Act of Parliament)

Waranga, PO - Bori (Butibori), Nagpur (Rural), Nagpur – 441108

SMART BANK SPACE

Introduction :

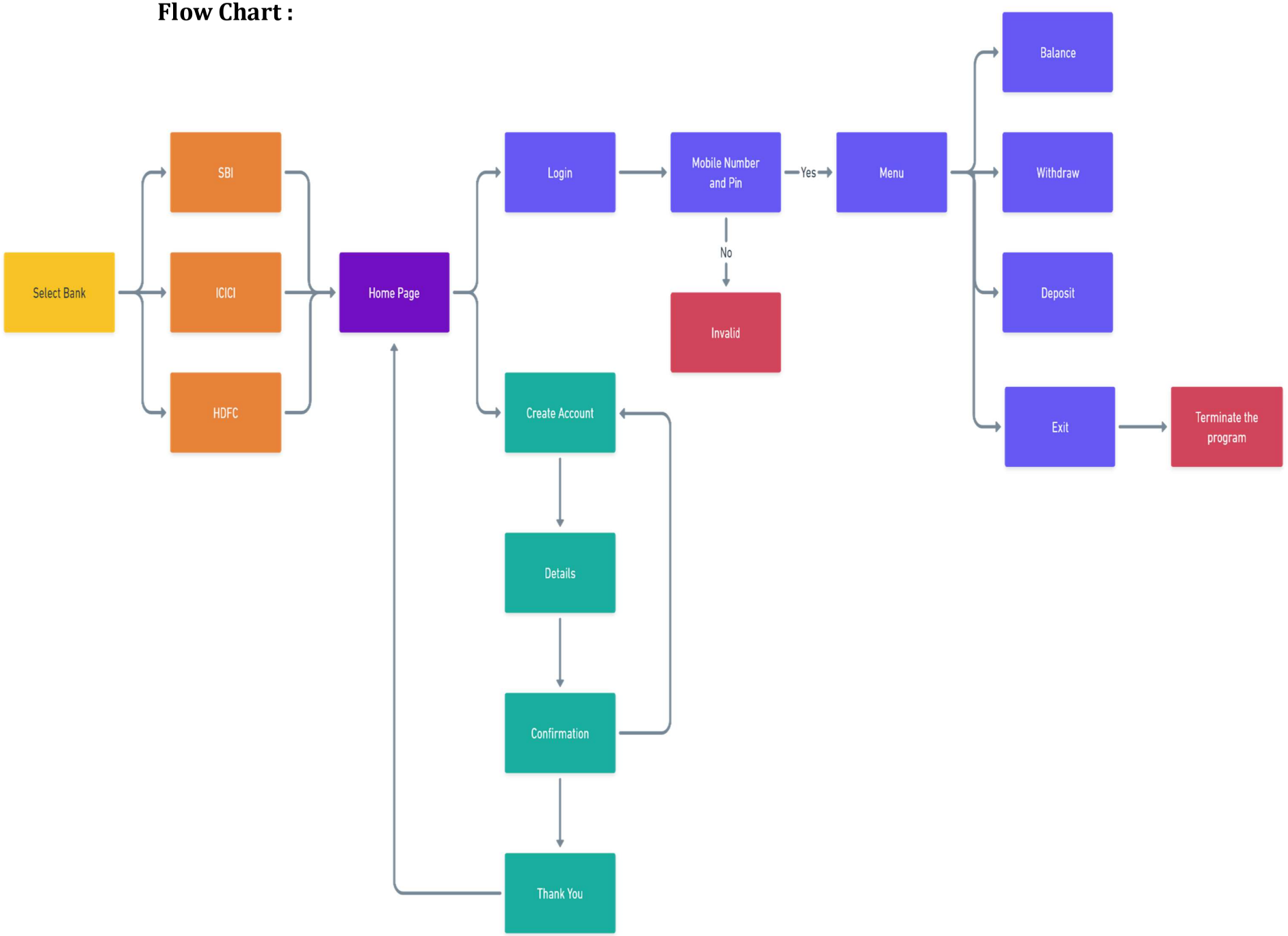
When the program is implemented, the user who uses this program will be able to see all the information and services provided. We have implemented Smart Bank Space where in we bring bank operations of different banks under one umbrella. The program is designed in such a way that the user has to card and pin number. Once verified, he is provided a menu and he/she had to enter the option provided in the menu. The operations include but not only limited to

1) Create Account
2) Login to existing account
3) Withdraw Money
4) Deposit Money
5) Check Balance

Technologies Used:

We have used Java to create a command line interface for user to interact. MySQL is used for database and SQL packages in Java such as Statement and ReturnSet are used to act as bridge between command line interface and database.

Flow Chart :

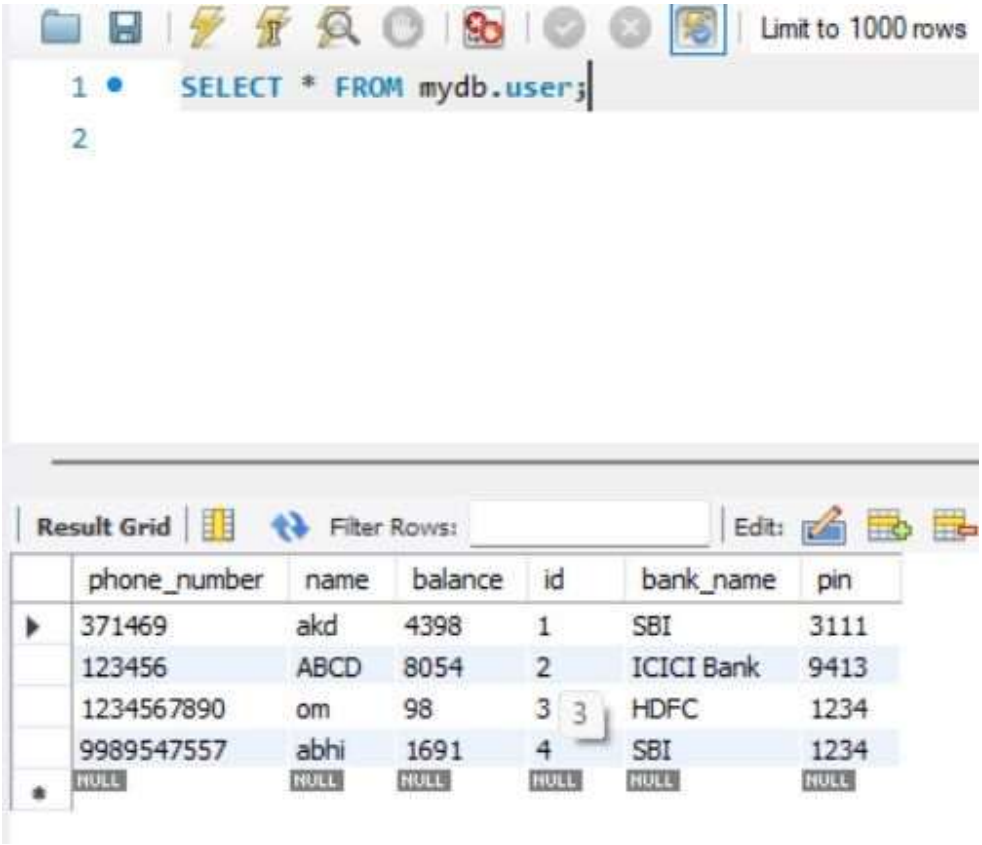

Select Bank
SBI
ICICI
HDFC
Home Page
Login
Mobile Number and Pin
Yes
No
Invalid
Menu
Balance
Withdraw
Deposit
Exit
Terminate the program
Create Account
Details
Confirmation
Thank You

Output :

- **Program Execution**

```
Select your bank
1. HDFC
2. ICICI Bank
3. SBI
1
Thanks for choosing HDFC
Smart Bank Space welcomes you

Enter your choice:
1. Login to your account
2. Open a new bank account
1
Enter your phone number:
9989547557
Enter your pin:
1234
Login successful

Enter your choice:
1. Check account balance
2. Withdraw money
3. Deposit money
4. Exit
1

Login successful
Your balance is: 195.0

Enter your choice:
1. Check account balance
2. Withdraw money
3. Deposit money
4. Exit
3

Enter the amount:
1000
Your balance is: 1195.0
Money deposited successfully
```

- **MySql Database**



phone_number	name	balance	id	bank_name	pin
371469	akd	4398	1	SBI	3111
123456	ABCD	8054	2	ICICI Bank	9413
1234567890	om	98	3	HDFC	1234
9989547557	abhi	1691	4	SBI	1234
NULL	NULL	NULL	NULL	NULL	NULL

Conclusion :

The following concepts have been implemented in our project.

1) Encapsulation
2) SQL connection
3) Exception Handling
4) Recursion
5) Access Modifiers
6) Polymorphism

Source Code:

```
package core;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

import database.db;

class sbs {
    private Connection conn = db.getConnection();
    private Scanner sc = new Scanner(System.in);
    int processingFee;
    String bank;

    public void selectBank() {
        System.out.println();
        Statement stmt = null;
        ResultSet rs = null;
        try {
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM banks");
            System.out.println("Select your bank");
            int index = 0;
            String bank_name[] = new String[3];
            while (rs.next()) {
                bank_name[index] = rs.getString("bank_name");
                System.out.println((index + 1) + ". " + bank_name[index]);
                index++;
            }
            int opt = sc.nextInt();

            switch (opt) {
                case 1:
                    processingFee = 2;
                    bank = bank_name[opt - 1];
                    break;
                case 2:
                    processingFee = 4;
                    bank = bank_name[opt - 1];
                    break;
                case 3:
```

```
                    processingFee = 5;
                    bank = bank_name[opt - 1];
                    break;
                default:
                    System.out.println("Enter a valid choice");
            }
            System.out.println("Thanks for choosing " + bank);
            homePage();
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }
    }

    public void createAccount() {
        System.out.println();
        String name;
        String phone;
        int pin;
        int balance = 0;
        System.out.println("Enter your name:");
        name = sc.next();
        System.out.println("Enter your phone number:");
        phone = sc.next();
        if (phone.length() != 10) {
            System.out.println("Enter a valid phone number");
            createAccount();
        }
        System.out.println("Choose your pin:");
        pin = sc.nextInt();
        if (pin < 1000 || pin > 9999) {
            System.out.println("Enter a valid pin");
            createAccount();
        }
        System.out.println("Do you want to deposit money? (y/n)");
        String opt = sc.next();
        if (opt.equals("y")) {
            System.out.println("Enter the amount to deposit");
            balance = sc.nextInt();
        }
        Statement stmt = null;
        System.out.println();
```

```
        System.out.println("Please check details and press 1 to confirm or 2 to
edit: ");
        System.out.println();

        System.out.println("Name: " + name + "\nPhone number: " + phone + "\nPin:
" + pin + "\nBalance: " + balance
                + "\nBank: " + bank);
        int choice = sc.nextInt();
        if (choice == 1) {
            try {
                stmt = conn.createStatement();
                stmt.execute("INSERT INTO user (name, phone_number, pin, balance,
bank_name) VALUES ('" + name + "', '" + phone
                        + "', '" + pin + "', '" + balance + "', '" + bank +
"')");
                System.out.println("Your account is created. \nPlease visit the
bank for verification.\nThank you");
            } catch (SQLException ex) {
                // handle any errors
                System.out.println("SQLException: " + ex.getMessage());
                System.out.println("SQLState: " + ex.getSQLState());
                System.out.println("VendorError: " + ex.getErrorCode());
            }
        } else {
            createAccount();
        }

        homePage();
    }

    public void homePage() {
        System.out.println("Smart Bank Space welcomes you");
        System.out.println();

        System.out.println("Enter your choice: ");
        System.out.println("1. Login to your account");
        System.out.println("2. Open a new bank account");
        int opt = sc.nextInt();
        switch (opt) {
            case 1:
                login();
                break;
            case 2:
                createAccount();
                break;
```

```
            default:
                System.out.println("Enter a valid choice");
        }
    }

    public void login() {
        System.out.println("Enter your phone number: ");
        String phone = sc.next();
        if (phone.length() != 10) {
            System.out.println("Enter a valid phone number");
            login();
        }
        System.out.println("Enter your pin: ");
        int enteredPin = sc.nextInt();
        if (enteredPin < 1000 || enteredPin > 9999) {
            System.out.println("Enter a valid pin");
            login();
        }
        Statement stmt = null;
        ResultSet rs = null;
        try {
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM user WHERE phone_number = " +
phone);
            if (rs.next()) {
                int pin = rs.getInt("pin");
                if (pin == enteredPin) {
                    System.out.println("Login successful");
                    menu(phone, pin);
                } else {
                    System.out.println("Incorrect pin");
                    login();
                }
            } else {
                System.out.println("Phone number not found");
                login();
            }
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }

    }
```

```
    public void menu(String phone, int pin) {
        System.out.println();
        System.out.println("Enter your choice: ");
        System.out.println("1. Check account balance ");
        System.out.println("2. Withdraw money");
        System.out.println("3. Deposit money ");
        System.out.println("4. Exit ");
        int opt = sc.nextInt();
        switch (opt) {
            case 1:
                checkBalance(phone, pin);
                break;
            case 2:
                withdraw(phone, pin);
                break;
            case 3:
                deposit(phone, pin);
                break;
            case 4:
                System.out.println("Thank you for using Smart Bank Space");
                System.exit(0);
                break;
            default:
                System.out.println("Enter a valid choice");
        }
    }

    public void checkBalance(String phone, int pin) {
        System.out.println();
        Statement stmt = null;
        ResultSet rs = null;
        try {
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM user WHERE phone_number = " +
phone);
            if (rs.next()) {
                int validPin = rs.getInt("pin");
                if (pin == validPin) {
                    System.out.println("Login successful");
                    float balance = rs.getFloat("balance");
                    System.out.println("Your balance is: " + balance);
                    menu(phone, pin);
                } else {
                    System.out.println("Incorrect pin");
```

```
                    login();
                }
            } else {
                System.out.println("Phone number not found");
                login();
            }
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }
        System.out.println();
        menu(phone, pin);
    }

    public void withdraw(String phone, int pin) {
        System.out.println();
        Statement stmt = null;
        ResultSet rs = null;

        System.out.println("Enter amount to withdraw, available denominations are
100, 500, 2000");
        float amount = sc.nextFloat();
        if (amount % 100 != 0) {
            System.out.println("Enter a valid amount");
            withdraw(phone, pin);
        }
        try {
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM user WHERE phone_number = " +
phone);
            if (rs.next()) {
                int validPin = rs.getInt("pin");
                if (pin == validPin) {
                    float balance = rs.getFloat("balance");
                    if (balance >= amount + processingFee) {
                        balance -= amount + processingFee;
                        stmt.execute("UPDATE user SET balance = " + balance + "
WHERE phone_number = " + phone);
                        System.out.println("Please collect your cash");
                        System.out.println("Your balance is: " + balance);
                    } else {
                        System.out.println("Insufficient balance");
                    }
```

```
                } else {
                    System.out.println("Incorrect pin");
                    login();
                }
            } else {
                System.out.println("Phone number not found");
                login();
            }
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }

        System.out.println();
        menu(phone, pin);
    }

    public void deposit(String phone, int pin) {
        System.out.println();
        System.out.println("Enter the amount: ");
        float amount = sc.nextFloat();
        Statement stmt = null;
        ResultSet rs = null;
        try {
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM user WHERE phone_number = " +
phone);
            if (rs.next()) {
                int validPin = rs.getInt("pin");
                if (pin == validPin) {
                    float balance = rs.getFloat("balance");
                    balance += amount;
                    stmt.execute("UPDATE user SET balance = " + balance + " WHERE
phone_number = " + phone);
                    System.out.println("Your balance is: " + balance);
                    System.out.println("Money deposited successfully");
                } else {
                    System.out.println("Incorrect pin");
                    login();
                }
            } else {
                System.out.println("Phone number not found");
                login();
```

```
            }
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }
        System.out.println();
        menu(phone, pin);
    }
}

public class BankingSystem {
    public static void main(String args[]) {
        sbs ob = new sbs();
        ob.selectBank();
    }
}
```