

# A PROJECT REPORT ON

## "BANK MANAGEMENT SYSTEM"

( A SOFTWARE DEVELOPED USING JAVA & MYSQL )



Under The guidance of  
PARTHA PRATIM BEZBARUAH (HoD. Computer Science)

**SUBMITTED BY:-**

NAME:- REKIBUR UDDIN

CLASS:- BCA 5TH SEMISTER

ROLL NO:- 01

REGISTRATION NO:- 22070556 OF 2023-2024

YEAR:- 2024-2025



Department of Computer Science  
**BAOSI BANIKANTA KAKATI COLLEGE**

**Nagaon, Barpeta (Assam)**

**PIN- 781311**

---

**Ref:.....**

**Date:.....**

From:

Mr. Partha Pratim Bezbaruah, HoD of Computer Science

**CERTIFICATE**

This is to certify that the Project Report entitled "**BANK MANAGEMENT SYSTEM**" Of "**B.B.K. COLLEGE, NAGAON**" has prepared and completed successfully by "**REKIBUR UDDIN**" under my supervision and guidance. The project work done by his a dissertation only and cannot be produced for any other purpose. His performance during that period has been found satisfactory.

**Place: NAGAON**

**Place: .....**

**(PARTHA PRATIM BEZBARUAH)**

**HoD, of Computer Science**

**BBK College, Nagaon**

## **CERTIFICATE FROM THE PROJECT GUIDE**

This is to certify that the project/Dissertation entitled, **BANK MANAGEMENT SYSTEM** is a bona fide done by **REKIBUR UDDIN** (BCA, ROLL NO- 01) in partial fulfillment of BCA examination and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

**Signature of Guide**

## **ABSTRACT**

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project we tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also, to enable the user's work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manual systems, which are overcome by this software. This project is developed using JAVA language and MYSQL use for database connection. The system design is then implemented with MYSQL, JAVA forms, and JDBC. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating while the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

## INDEX

<b>Sr. No.</b>	<b>Topic</b>	<b>Page No.</b>
1	Introduction	7
2	Problem Statement	8
3	System Design and Requirements - Architectural Diagram/ block diagram - Flow chart - Front end/Back end technology	9-11
4	Source Code	12-18
5	Experimental Results - GUI	19-26
6	Conclusion and Future Scope - Conclusion - Future Scope	27
	References	28

## LIST OF FIGURES

<b>Sr. No.</b>	<b>Name of the Figure</b>	<b>Page No.</b>
1	System Design	9
2	Flowchart	10
3	Login Page	19
4	Main Menu	20
5	Add Customer	21
6	Add Account	21
7	Transaction <ul style="list-style-type: none"><li>• Withdraw page</li><li>• Deposit Page</li></ul>	22-23
8	Transfer Page	24
9	Report	25
10	Balance	25
11	Admin	26

# **Chapter 1**

## **Introduction**

- Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money.
- Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly.
- And of course, it encourages management committee in taking some needed decision for future enhancement of the bank. Now a days, managing a bank is tedious job upto certain limit. So a bank management system that reduces the work is essential. Also today's world is a genuine computer world and is getting faster and faster day-by-day.
- Thus, considering above necessities, the system for bank management has became necessary which would be useful in managing the bank more efficiently.

## **Chapter 2**

### **Problem Statement**

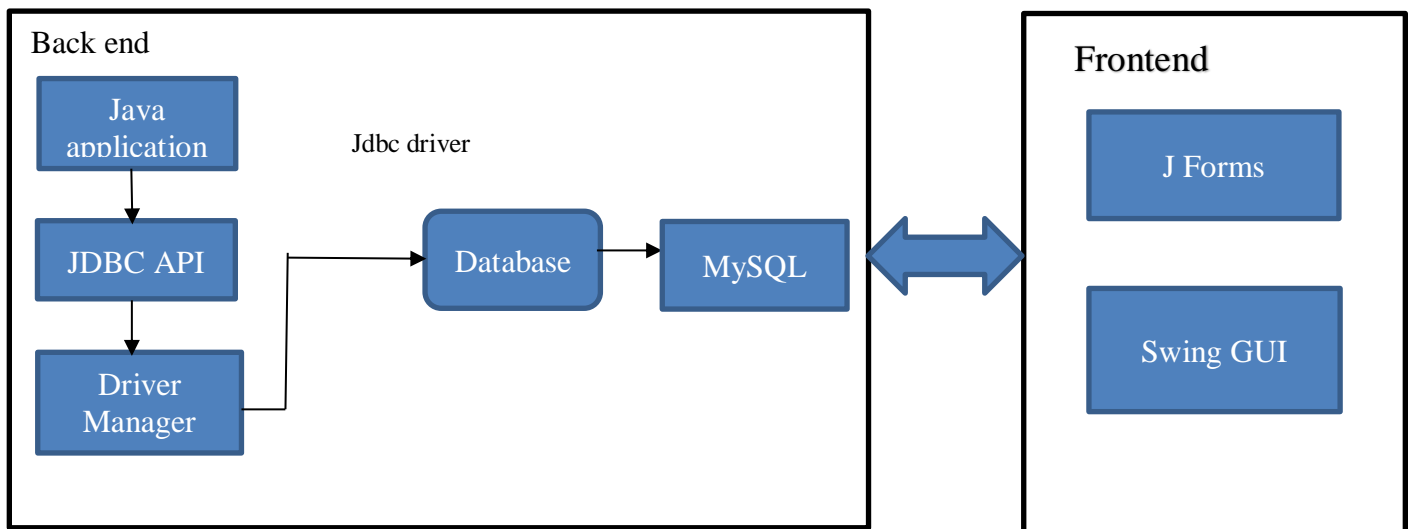
- The Bank Management system consists of the bank administrator and the customer. The administrator will need to create a new account for the customer by logging into their existing account.
- For creating a new account on the system the administrator will need to enter details of customer like first name, last name, address, contact no and will add the branch where the account has been created.
- The system executes operations like creating new account, checking the balance of the account, withdrawing, depositing, money transfers between two accounts and viewing transaction history.



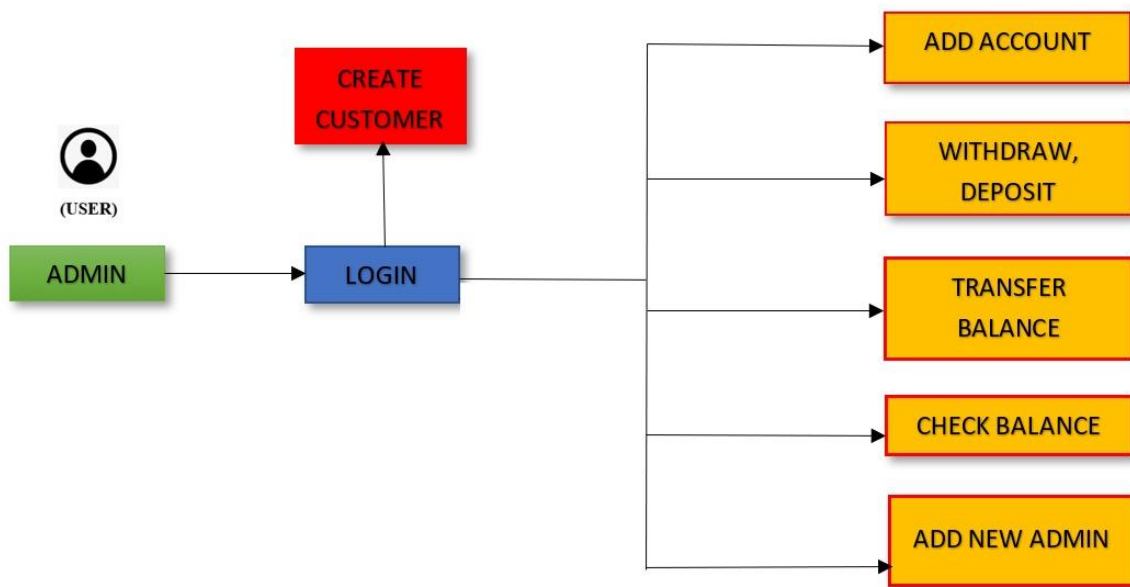
## Chapter 3

### System Design

#### 3.1 Architectural Diagram/ block diagram



### 3.2 Flow chart



# **System Requirements:**

## **Developer Requirements:**

- Interpreters: Java and Java Database Connectivity(JDBC).
- Software: Netbeans, Xampp.
- Databases: MySQL.
- Hardware: 10GB storage, 4gb ram 4.1.2

# Chapter 4

## Source code

### Main menu:

```
package bank;

import java.awt.Desktop;
import java.awt.PopupMenu;
import static java.awt.SystemColor.desktop;
import javax.swing.JDesktopPane;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
/**
 *
 * @author parth
 */
public class mainmenu extends javax.swing.JFrame {

    private PopupMenu internalFrame;

    public mainmenu() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jMenuItem1 = new javax.swing.JMenuItem();
        jDesktopPane1 = new javax.swing.JDesktopPane();
        JMenuBar1 = new javax.swing.JMenuBar();
        JMenu1 = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();
        jMenuItem3 = new javax.swing.JMenuItem();
        JMenu2 = new javax.swing.JMenu();
    }
}
```

```

jMenuItem4 = new javax.swing.JMenuItem();
jMenuItem5 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
jMenuItem6 = new javax.swing.JMenuItem();
jMenu4 = new javax.swing.JMenu();
jMenuItem7 = new javax.swing.JMenuItem();
jMenu5 = new javax.swing.JMenu();
jMenuItem8 = new javax.swing.JMenuItem();
jMenu6 = new javax.swing.JMenu();
jMenuItem9 = new javax.swing.JMenuItem();

jMenuItem1.setText("jMenuItem1");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setPreferredSize(new java.awt.Dimension(1366, 768));

javax.swing.GroupLayout jDesktopPanelLayout = new javax.swing.GroupLayout(jDesktopPanel);
jDesktopPanel.setLayout(jDesktopPanelLayout);
jDesktopPanelLayout.setHorizontalGroup(
    jDesktopPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 823, Short.MAX_VALUE)
);
jDesktopPanelLayout.setVerticalGroup(
    jDesktopPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 764, Short.MAX_VALUE)
);

jMenu1.setText("File");

jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_BACK_
SPACE, 0));
jMenuItem2.setText("Customer");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem2);

jMenuItem3.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_B,
java.awt.event.InputEvent.SHIFT_DOWN_MASK));
jMenuItem3.setText("Account");
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem3);

```

```

jMenuBar1.add(jMenu1);

jMenu2.setText("Transaction");

jMenuItem4.setText("Withdraw");
jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem4ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem4);

jMenuItem5.setText("Deposit");
jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem5);

jMenuBar1.add(jMenu2);

jMenu3.setText("Transfer");
jMenu3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
    }
});

jMenuItem6.setText("AccountToAccount");
jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem6ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem6);

jMenuBar1.add(jMenu3);

jMenu4.setText("Report");

jMenuItem7.setText("Customer Report");
jMenuItem7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem7ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem7);

jMenuBar1.add(jMenu4);

```

```

jMenu5.setText("Balance");

jMenuItem8.setText("Balance Check");
jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem8ActionPerformed(evt);
    }
});
jMenu5.add(jMenuItem8);

jMenuBar1.add(jMenu5);

jMenu6.setText("Admin");

jMenuItem9.setText("Account Creation");
jMenuItem9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem9ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem9);

jMenuBar1.add(jMenu6);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jDesktopPane1)
                    .addGap(69, 69, 69))
                .add(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jDesktopPane1)
                    .addContainerGap()))
            .addContainerGap())
);

pack();
} // </editor-fold>

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    customer cus = new customer();

```

```

jDesktopPane1.add(cus);
cus.setVisible(true);

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    account cus = new account();
    jDesktopPane1.add(cus);
    cus.setVisible(true);

}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    withdraw cus = new withdraw();
    jDesktopPane1.add(cus);
    cus.setVisible(true);

}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    deposit cus = new deposit();
    jDesktopPane1.add(cus);
    cus.setVisible(true);

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:


}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    transfer cus1 = new transfer();
    jDesktopPane1.add(cus1);
    cus1.setVisible(true);

}

```



```

private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cusreport cus1 = new cusreport();
    jDesktopPane1.add(cus1);
    cus1.setVisible(true);

}

private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    balance cus1 = new balance();
    jDesktopPane1.add(cus1);
    cus1.setVisible(true);

}

private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    user1 cus1 = new user1();
    jDesktopPane1.add(cus1);
    cus1.setVisible(true);

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
    catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.Level.SEVERE, null, ex);
    }
    catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.Level.SEVERE, null, ex);
    }
    catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.Level.SEVERE, null, ex);
    }
    finally {
        Main window = new Main();
        window.setVisible(true);
    }
}

```

```

    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(mainmenu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(mainmenu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(mainmenu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(mainmenu.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new mainmenu().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JDesktopPane jDesktopPanel1;
private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenu jMenuItem3;
private javax.swing.JMenu jMenuItem4;
private javax.swing.JMenu jMenuItem5;
private javax.swing.JMenu jMenuItem6;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem5;
private javax.swing.JMenuItem jMenuItem6;
private javax.swing.JMenuItem jMenuItem7;
private javax.swing.JMenuItem jMenuItem8;
private javax.swing.JMenuItem jMenuItem9;
// End of variables declaration

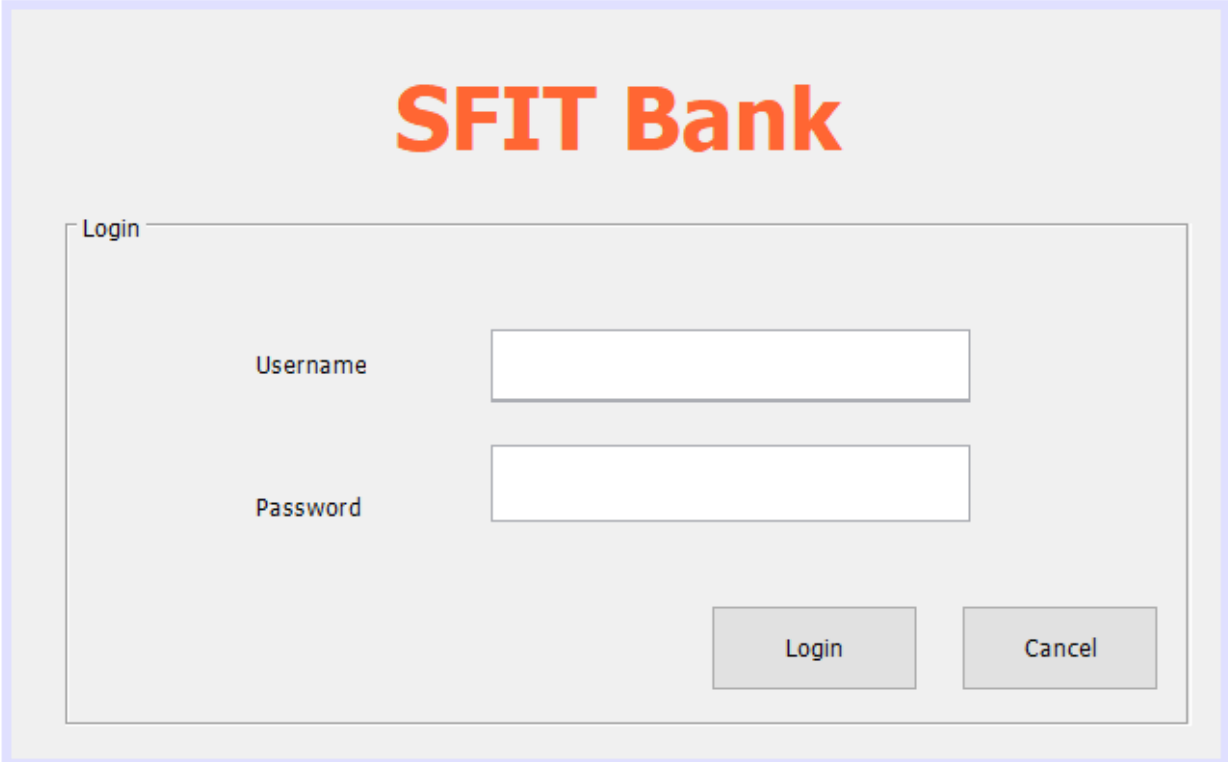
```

## Chapter 5

### Experimental Results

#### 5.1 GUI

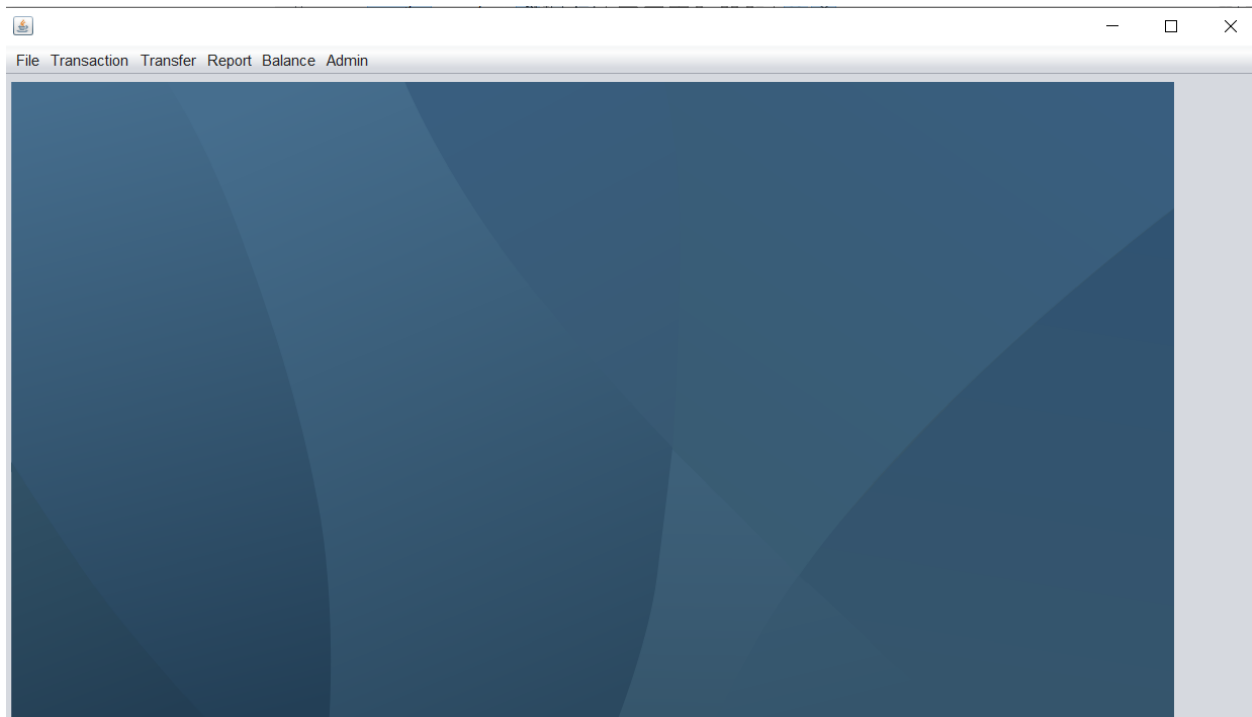
**Login Page:** Administrator(user) have to insert his credentials to access the system. All the data of the administrator is already stored in the database. If the credentials does not match, the user does not have access to the system.



The image shows a login page for SFIT Bank. At the top, the text "SFIT Bank" is displayed in a large, bold, orange font. Below this, there is a light gray rectangular box with a thin black border. Inside this box, the word "Login" is written in a small, dark font at the top left. Below "Login", there are two labels: "Username" and "Password", each followed by a white rectangular input field. At the bottom right of the box, there are two buttons: "Login" and "Cancel", both with a light gray background and a thin black border.

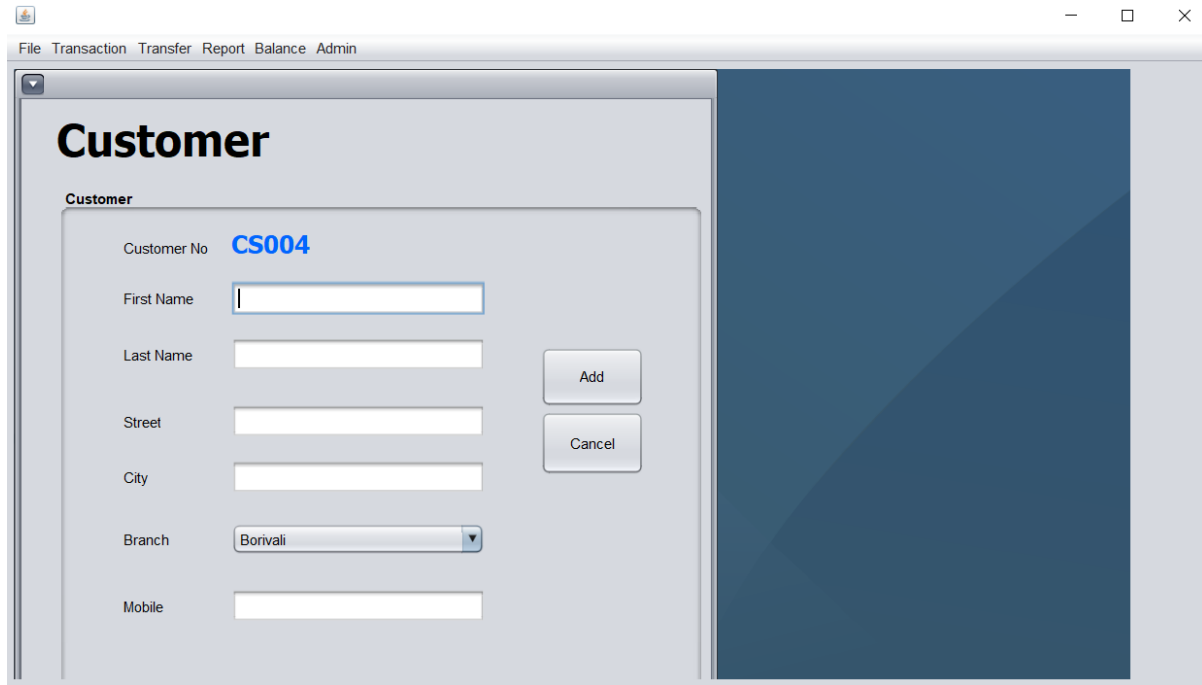
Login Page

**Main Menu:** Through this page the user can add a customer, create account, deposit, withdraw, transfer, report, balance check and can add new admin.



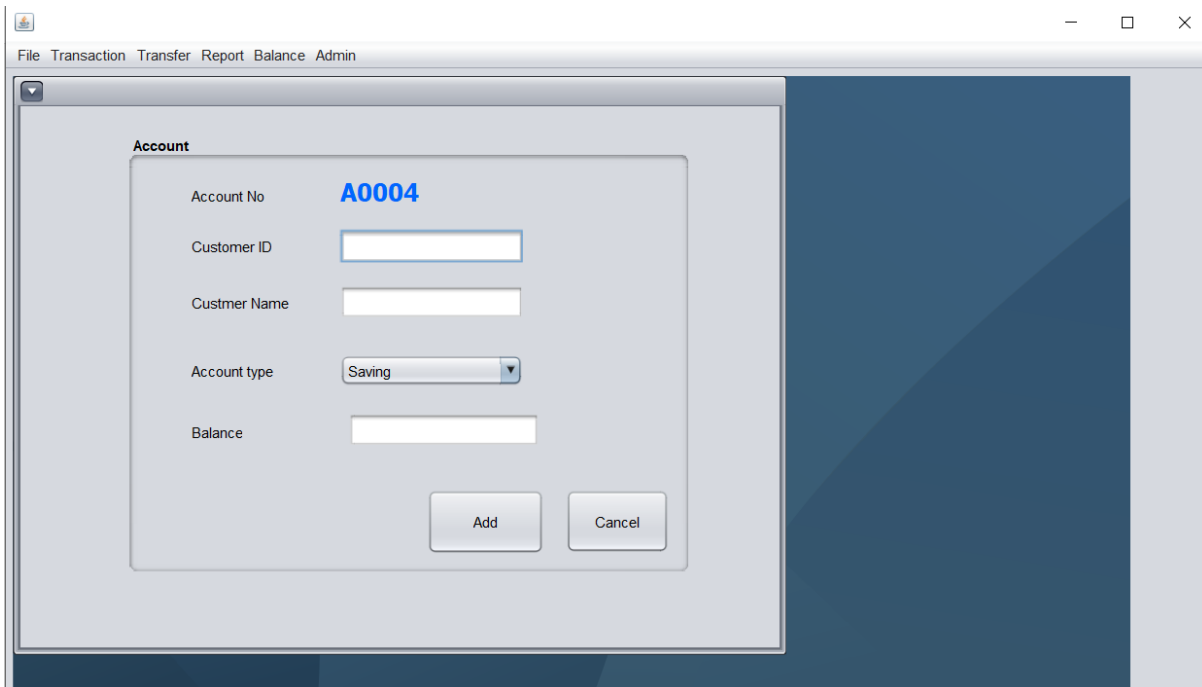
Main Menu

Addition of new customer and then creating account for the customer.



The screenshot shows a software window titled 'Customer' with a menu bar (File, Transaction, Transfer, Report, Balance, Admin). The form contains the following fields: 'Customer No' (pre-filled with 'CS004'), 'First Name' (empty text box), 'Last Name' (empty text box), 'Street' (empty text box), 'City' (empty text box), 'Branch' (dropdown menu with 'Borivali' selected), and 'Mobile' (empty text box). To the right of the form are 'Add' and 'Cancel' buttons.

ADD Customer



The screenshot shows a software window titled 'Account' with a menu bar (File, Transaction, Transfer, Report, Balance, Admin). The form contains the following fields: 'Account No' (pre-filled with 'A0004'), 'Customer ID' (empty text box), 'Customer Name' (empty text box), 'Account type' (dropdown menu with 'Saving' selected), and 'Balance' (empty text box). At the bottom of the form are 'Add' and 'Cancel' buttons.

ADD account

**Transaction:** The user can perform two actions i.e. withdrawing and depositing on his account.

On loading of withdraw/deposit page, the user has to enter the account no. If the account no details is stored in the database, It will be reflected on this page.

File Transaction Transfer Report Balance Admin

**Account No**

**Enter the Account No**

Find

Customer ID **jLabel7**

Firstname

Lastname

Date **2020/12/07**

**Balance**

**Withdraw**

Withdraw page

The image shows a Java Swing window titled "Deposit Page" with a menu bar containing "File", "Transaction", "Transfer", "Report", "Balance", and "Admin". The window has a light gray background and a dark blue sidebar on the right. A modal dialog box is open in the center, titled "Account No". Inside the dialog, there is a section titled "Enter the Account No" with a text input field and a "Find" button. Below this, there are labels for "Customer ID", "Firstname", "Lastname", and "Date". The "Customer ID" label is followed by a red text "jLabel7". The "Date" label is followed by the text "2020/12/07". To the right of the input fields, there is a "Balance" label and a large blue "Balance" text. Below the "Balance" label, there is a "Deposit" label and a red rectangular area. At the bottom of the dialog, there are "OK" and "Cancel" buttons.

File Transaction Transfer Report Balance Admin

Account No

Enter the Account No

Find

Balance

Balance

Deposit

Customer ID **jLabel7**

Firstname

Lastname

Date **2020/12/07**

OK Cancel

Deposit Page

**Transfer:** On this page, the user can transfer money from one account to another account. On loading, the user has to enter the account no's from which he wishes to transfer money.

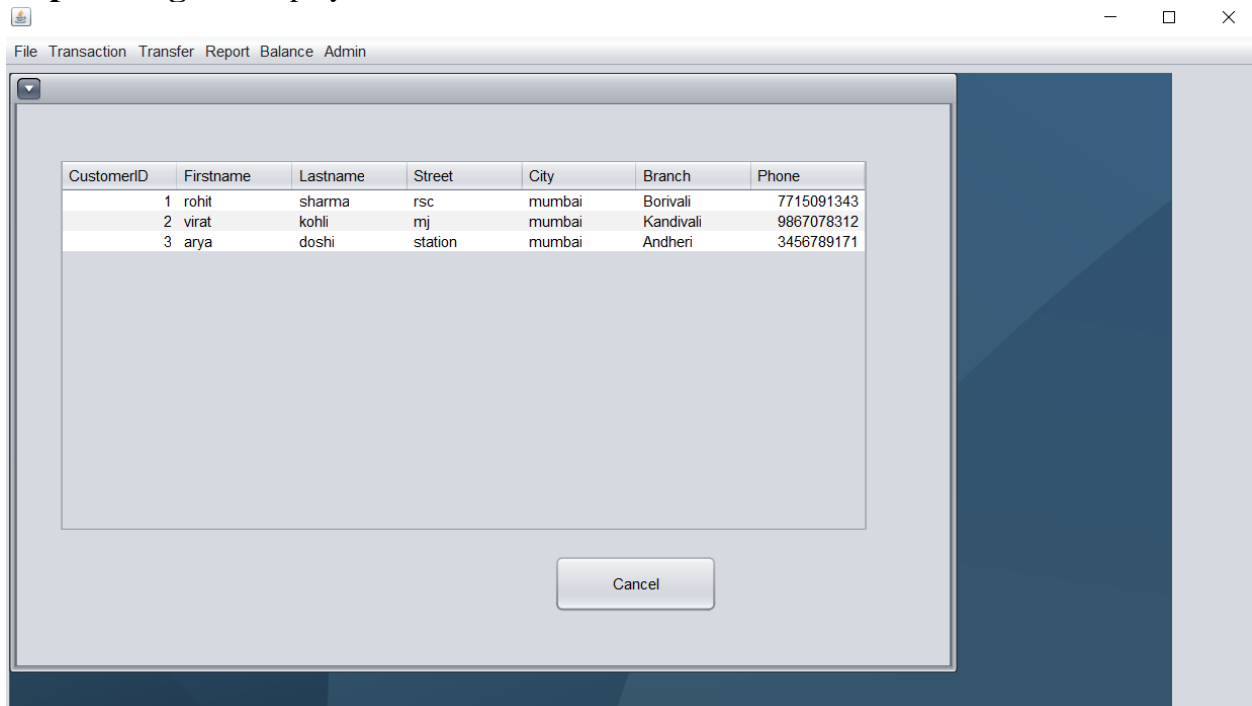
The screenshot shows a web application window with a menu bar containing 'File', 'Transaction', 'Transfer', 'Report', 'Balance', and 'Admin'. A modal dialog box titled 'Account' is displayed in the foreground. The dialog box contains four input fields with corresponding buttons:

Field Label	Input Field	Button
From Account No	<input type="text"/>	Find
Balance	<input type="text"/>	Transfer
To Account No	<input type="text"/>	Cancel
Amount	<input type="text"/>	

Transfer Page

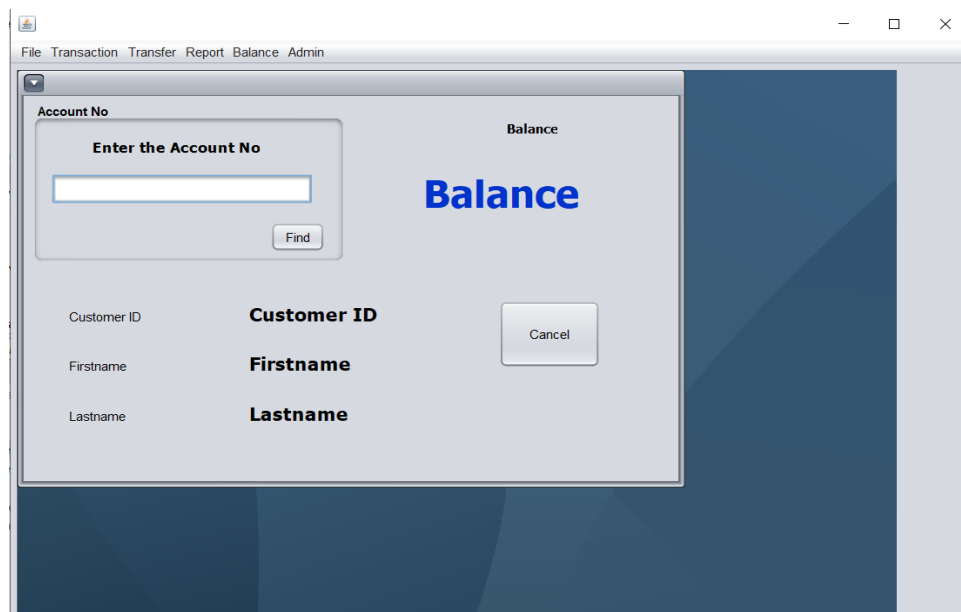


**Report Page:** It displays all the details of the customer to the administrator.



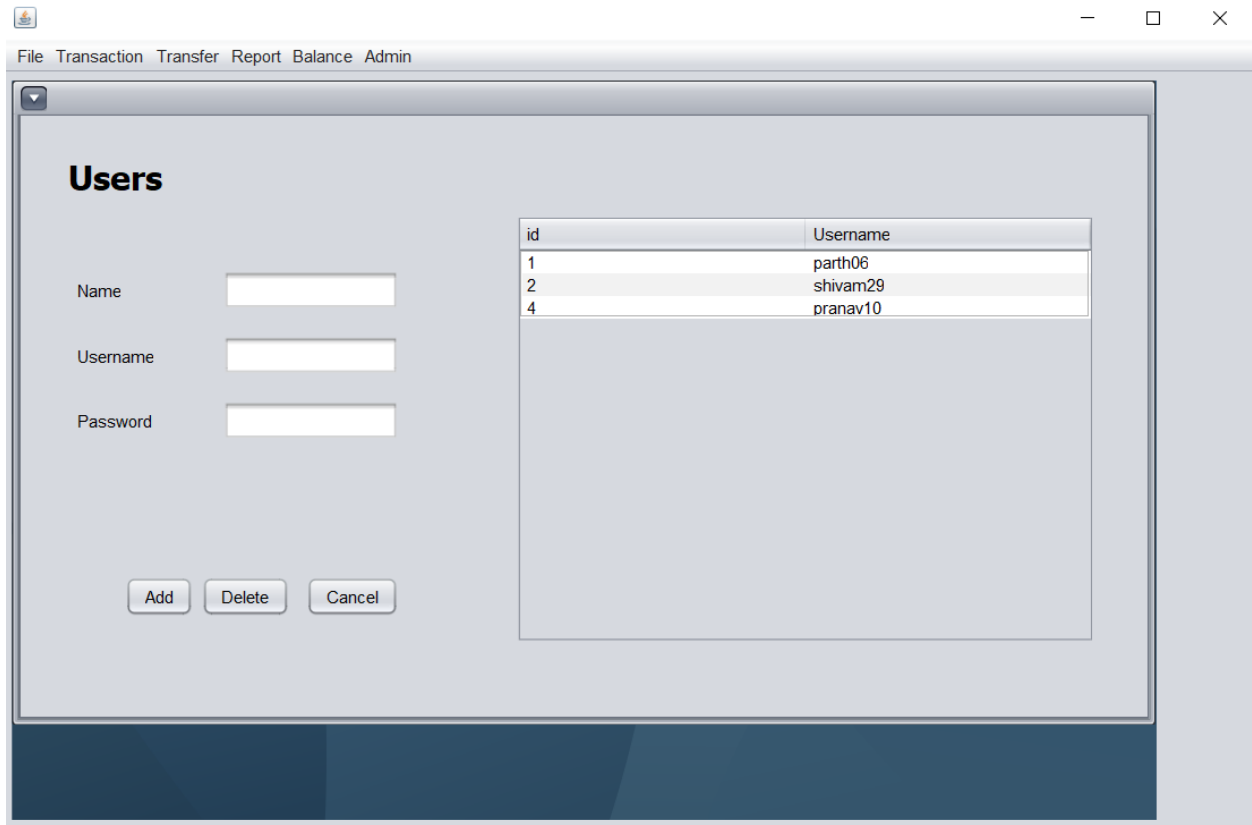
Report Page

**Balance:** It displays the account balance of the selected customer.



Balance page

**Admin:** It allows the existing administrator to create new administrators and delete existing administrator.



The screenshot displays a web application window titled "Admin Page". The window has a menu bar with options: File, Transaction, Transfer, Report, Balance, and Admin. The main content area is titled "Users" and contains a form for adding or deleting users. The form includes three input fields: "Name", "Username", and "Password". Below these fields are three buttons: "Add", "Delete", and "Cancel". To the right of the form is a table listing existing users.

id	Username
1	parth06
2	shivam29
4	pranav10

Admin Page

## **Chapter 6**

### **Conclusion and Future Scope**

#### **6.1 Conclusion**

Bank management system is a virtualization of transactions in banking system. The banking system are used manual working but when we used bank management system it is totally virtualization process which avoid manual process and converts it in automatic process. Bank management system is saving the time with accuracy than bank manual system.

#### **6.2 Future Scope**

- This project aspires to be a simulation of the Banking system for banks and also reduces the human errors caused by employees or the customer itself.
- If coupled with appropriate hardware this system can be turned into an ATM software.

## References

- <https://netbeans.org/kb/index.html>
- <https://www.javatpoint.com/online-banking-project>
- <https://projectsgeek.com/2016/02/complete-banking-system-java-project.html#:~:text=Proposed%20System,valid%20user%20id%20and%20password.>
- <https://www.apachefriends.org/index.html>