

**NANDHA ENGINEERING COLLEGE**

**ERODE-638052**

(Autonomous)

(Affiliated to Anna University, Chennai)



**DEPARTMENT  
OF  
COMPUTER SCIENCE AND ENGINEERING**

**BANK MANAGEMENT SYSTEM  
PBL REPORT  
(17CSC07-DATABASE MANAGEMENT SYSTEM)**

***Submitted by***

Pranav S	(21CS068)
Praveen N	(21CS069)
Sehshan Surya G	(21CS081)
Vignesh V	(21CS111)

in partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING  
IN  
COMPUTER SCIENCE AND ENGINEERING**

# **NANDHA ENGINEERING COLLEGE**

(Autonomous)

(Affiliated to Anna University, Chennai)

## **BONAFIDE CERTIFICATE**

Certified that this Report titled “BANK MANAGEMENT SYSTEM” is the bonafide work of **PRANAV.S (21CS068), PRAVEEN.N (21CS069), SEHSHAN SURYA.G (21CS081), VIGNESH.V (21CS111)** who carried out the work under my supervision.

### **Signature of the HOD**

Dr. D. VANATHI

Professor & Head

Department of CSE

Nandha Engineering College

Erode – 638052.

### **Signature of the Supervisor**

Ms. S. KAVITHA

Assistant Professor

Department of CSE

Nandha Engineering College

Erode – 638052.

Submitted for End Semester PBL Review Examination held on \_\_\_\_\_

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1	INTRODUCTION	5
2	SYSTEM SPECIFICATION	7
3	SOFTWARE DESCRIPTION	8
4	PROJECT DESCRIPTION	11
5	FUTURE ENHANCEMENT	13
6	CONCLUSION	14
7	APPENDIX	15
	7.1 SOURCE CODE	
	7.2 SCREEN SHOTS	
8	REFERENCE	60

## **ABSTRACT**

The Bank Management System is a comprehensive software solution designed to streamline and automate various banking operations, providing an efficient and secure platform for managing financial transactions, customer accounts, and administrative tasks. This project aims to enhance the overall efficiency, accuracy, and security of banking processes, thereby improving customer experience and organizational productivity.

The Bank Management System encompasses various modules that cover essential functionalities, including customer management, account management, transaction processing, loan management, and report generation. These modules are designed to integrate seamlessly, ensuring smooth and reliable operations within the bank.

The customer management module enables the bank to maintain accurate and up-to-date records of customers, including personal information, account details, and transaction history. It allows the creation, modification, and deletion of customer accounts, facilitating efficient customer relationship management.

The account management module provides comprehensive tools to manage different types of accounts offered by the bank, such as savings accounts, checking accounts, and fixed deposit accounts. It enables the bank to handle account openings, closures, balance inquiries, fund transfers, and other related transactions.

Overall, the Bank Management System project streamlines banking operations, improves customer service, enhances data accuracy, and facilitates effective decision-making within the bank. By automating key processes and ensuring data integrity and security, the system contributes to the efficient functioning of the bank and helps it meet the evolving needs of customers in the digital era.

# **CHAPTER 1**

## **INTRODUCTION**

The Bank Management System project, developed using Java programming language and MySQL database, aims to provide a robust and efficient solution for managing various banking operations. This project leverages the power of Java's object-oriented programming paradigm and the reliability of MySQL database to create a comprehensive software system that automates and streamlines the day-to-day activities of a bank.

The primary objective of the Bank Management System project is to enhance the overall efficiency and effectiveness of banking operations. It offers a user-friendly interface for both bank employees and customers, enabling them to perform various tasks seamlessly. The project incorporates various modules that handle essential functionalities such as customer management, account management, transaction processing, loan management, and report generation.

Java, being a versatile and platform-independent language, is an ideal choice for developing this project. It provides the necessary tools and libraries to build robust and scalable applications. The object-oriented nature of Java allows for the creation of modular and reusable code, making it easier to maintain and extend the system in the future.

MySQL, a widely-used relational database management system, is employed as the backend database for the Bank Management System project. It ensures the efficient storage and retrieval of data, providing a reliable and secure foundation for managing customer accounts, transaction records, loan information and other information can be stored and can be retrieved easily.

The Bank Management System project employs various Java libraries and frameworks, such as JavaFX for the graphical user interface (GUI) development and JDBC (Java Database Connectivity) for seamless communication between the Java application and the MySQL database.

The system's user interface offers a intuitive and user-friendly experience for bank employees to perform their tasks efficiently. It allows them to manage customer information, create and modify

accounts, process transactions, handle loan applications, and generate reports with ease. Customers can access their accounts, view balances, perform transactions, and access other banking services through a secure and intuitive web or desktop interface.

In conclusion, the Bank Management System project developed using Java and MySQL aims to automate and streamline banking operations, providing a comprehensive solution for managing customer accounts, transactions, loans, and generating reports. By leveraging the power of Java and MySQL, the project ensures reliability, scalability, and security while delivering an efficient and user-friendly banking system known as banking management system.

## **CHAPTER2**

### **SYSTEM SPECIFICATION**

#### **HARDWARE CONFIGURATION:**

System	:	HP
Processor	:	INTEL I3 11 <sup>TH</sup> GEN
RAM	:	8GB
Hard disk Capacity	:	1 TB

#### **SOFTWARE REQUIREMENTS:**

Operating System	:	Windows XP/Windows7/8/8.1/10/11
Front end	:	JAVA
Back end	:	MySql

## **CHAPTER3**

### **SOFTWARE DESCRIPTION**

#### **3.1. INTRODUCTION TO JAVA**

Java is a widely used, high-level programming language that was developed by Sun Microsystems (now owned by Oracle Corporation). It is known for its platform independence, meaning that Java programs can run on any device or operating system that has a Java Virtual Machine (JVM) installed. Java is extensively used for developing various types of software, ranging from desktop applications to enterprise-level web applications and mobile apps.

Here is a software description of Java:

**Object-Oriented Programming (OOP):** Java is based on the principles of OOP, which allows developers to create modular and reusable code. It supports concepts such as classes, objects, encapsulation, inheritance, and polymorphism, making it easier to design and implement complex software systems.

**Platform Independence:** Java programs are compiled into bytecode, which can run on any system with a JVM. This platform independence eliminates the need for developers to write different versions of their software for different platforms, reducing development time and effort.

**Rich Standard Library:** Java provides a vast standard library, known as the Java Class Library, which includes a wide range of pre-built classes and methods. This library offers ready-to-use functionality for common tasks such as input/output operations, networking, database connectivity, graphical user interface (GUI) development, and much more. It simplifies the development process and promotes code reusability.

**Memory Management:** Java incorporates automatic memory management through its garbage collection mechanism. Developers don't need to explicitly allocate or deallocate memory, as the JVM handles memory management. This feature reduces the risk of memory leaks and enhances the overall reliability and stability of Java applications.



**Exception Handling:** Java has a robust exception handling mechanism that enables developers to handle errors and exceptional situations gracefully. It allows the detection and recovery from unexpected errors, ensuring the stability of the software and providing a better user experience.

**Multi-threading:** Java supports multi-threading, which allows concurrent execution of multiple threads within a single program. This feature is beneficial for developing applications that require efficient utilization of system resources, such as parallel processing, handling multiple user requests simultaneously, and performing background tasks.

**Security:** Java has built-in security features that protect against various vulnerabilities, such as unauthorized access, data manipulation, and code injection. It provides a sandbox environment for executing untrusted code, ensuring the safety and integrity of the system.

**Community and Ecosystem:** Java has a large and active developer community, which contributes to the availability of numerous frameworks, libraries, and tools. These resources help developers in building robust and feature-rich applications more efficiently.

Overall, Java is a powerful and versatile programming language with a rich set of features and a vast ecosystem. Its platform independence, strong OOP support, extensive standard library, and emphasis on security make it an ideal choice for developing a wide range of software applications.

### **3.2. INTRODUCTION TO MySQL**

MySQL Workbench is the integrated environment for MySQL. It was developed by MySQL AB, and enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench is available in two editions, the regular free and open-source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

A command-line interface is a means of interacting with a computer program where the user issues commands to the program by typing in successive lines of text (command lines). MySQL ships with many command line tools, from which the main interface is the `mysql` client. MySQL Utilities is a set of utilities designed to perform common maintenance and

administrative tasks. Originally included as part of the MySQL Workbench, the utilities area stand-alone download available from Oracle.

Percona Toolkit is a cross-platform toolkit for MySQL, developed in Perl.[104] Percona Toolkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Percona Toolkit is included with several Linux distributions such as CentOS and Debian, and packages are available for Fedora and Ubuntu as well.

Percona Toolkit was originally developed as Maatkit, but as of late 2011, Maatkit is no longer developed. MySQL shell is a tool for interactive use and administration of the MySQL database. It supports JavaScript, Python or SQL modes and it can be used for administration and access purposes.

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for .NET/CLI Languages, and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion.

The `HTSQL`–URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs. Other drivers exist for languages

## **CHAPTER-4**

### **PROJECT DESCRIPTION**

#### **DESCRIPTION OF MODULES**

- Homepage
- Customer
- Account
- Customer Details
- Account
- Account Details
- Deposit
- Withdraw

1.Customer Management: This module allows the bank to maintain a centralized and up-to-date record of customer information. It includes features for creating, modifying, and deleting customer accounts, managing personal details, contact information, and transaction history.

2.Account Management: The account management module facilitates the efficient management of different types of accounts offered by the bank, such as savings accounts, checking accounts, and fixed deposit accounts. It enables bank employees to handle tasks such as account openings, closures, balance inquiries, fund transfers, and other related transactions.

3.Transaction Processing: This module handles the secure and accurate processing of various types of transactions, including deposits, withdrawals, fund transfers, and bill payments. It incorporates real-time transaction processing to ensure timely updates to customer accounts .

The Bank Management System project leverages the Java programming language for its development. Java's robustness, platform independence, and extensive libraries make it suitable for creating scalable and reliable banking applications. The project also utilizes a MySQL database to store and retrieve customer data, transaction records, loan information, and other vital banking data.

The system provides a user-friendly interface for both bank employees and customers. Bank employees can efficiently manage customer accounts, process transactions, handle loan applications, and generate reports through a secure and intuitive desktop or web interface. Customers can access their accounts, view balances, perform transactions, and access various banking services via a secure and user-friendly online interface.

Overall, the Bank Management System project streamlines banking operations, improves customer service, enhances data accuracy, and facilitates effective decision-making within the bank. It automates key processes, ensures secure transaction handling, and generates comprehensive reports, ultimately contributing to the efficient functioning of the bank and meeting the evolving needs of customers in the digital era.

## **CHAPTER 5**

### **SCOPE FOR FUTURE ENHANCEMENT**

The Bank Management System project offers various opportunities for future enhancements and improvements. Here are some potential areas for future development and expansion:

**Online Banking and Mobile Applications:** Adding online banking capabilities and developing mobile applications can provide customers with convenient access to their accounts and banking services. This can include features like account balance inquiries, fund transfers, bill payments, and transaction history. Implementing robust security measures and incorporating additional authentication methods can ensure the safety of online transactions.

**Integration with Payment Gateways:** Integrating the bank management system with popular payment gateways can enable customers to make online purchases and payments directly from their bank accounts. This integration can simplify the payment process for customers and expand the range of services offered by the bank.

**Fraud Detection and Prevention:** Implementing advanced fraud detection and prevention mechanisms can help safeguard the bank and its customers against fraudulent activities. Incorporating machine learning algorithms and anomaly detection techniques can enhance the system's ability to identify suspicious transactions, detect fraudulent patterns, and mitigate potential risks..

These future enhancements can further enhance the Bank Management System, improving customer satisfaction, operational efficiency, and competitive advantage for the bank. It is essential to prioritize and plan these enhancements based on the bank's specific needs, customer preferences, and industry trends.

## **CHAPTER 6**

### **CONCLUSION**

In conclusion, the Bank Management System project provides a comprehensive software solution for automating and streamlining banking operations. By leveraging Java programming language and MySQL database, the project offers a robust and user-friendly system for managing customer accounts, transactions, loans, and generating reports.

The project successfully achieves its objectives of enhancing operational efficiency, improving customer service, and ensuring data accuracy. It incorporates modules for customer management, account management, transaction processing, loan management, and report generation, covering the essential functionalities of a bank.

The use of Java enables platform independence, allowing the system to run on various devices and operating systems. The extensive Java Class Library provides pre-built classes and methods for common tasks, reducing development time and promoting code reusability. The integration with MySQL database ensures reliable data storage and retrieval, supporting efficient management of customer information, transaction records, and loan data.

The user-friendly interface of the system caters to both bank employees and customers, facilitating easy account management, transaction processing, and access to banking services. The project lays a strong foundation for potential future enhancements, such as online banking, mobile applications, advanced analytics, CRM integration, fraud detection, and regulatory compliance.

Overall, the Bank Management System project demonstrates the power of technology in improving banking operations. It enables banks to provide efficient services, maintain accurate records, make informed decisions, and adapt to the changing needs of customers. With continuous development and enhancement, the system has the potential to become a crucial tool in the modern banking industry, contributing to enhanced customer satisfaction and business success.

# CHAPTER7

## APPENDIX

### 7.1 SOURCE CODE

#### MAINMENU

```
/*  
  
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
  license  
  
* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template  
  
*/  
  
package hsbbank;  
  
  
  
/**  
 *  
 * @author Admin  
 */  
  
public class mainmenu extends javax.swing.JFrame {  
  
  
  
    /**  
    * Creates new form mainmenu  
    */  
  
    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() {

    jMenuBar2 = new javax.swing.JMenuBar();
    jMenu8 = new javax.swing.JMenu();    jMenu9
= new javax.swing.JMenu();
    jCheckBoxMenuItem1 = new javax.swing.JCheckBoxMenuItem();
    jDesktopPane1 = new javax.swing.JDesktopPane();    jMenuBar1 =
new javax.swing.JMenuBar();    jMenu1 = new
javax.swing.JMenu();    jMenuItem1 = new
javax.swing.JMenuItem();    jMenuItem8 = new
javax.swing.JMenuItem();    jMenuItem2 = new javax.swing.JMenu();
jMenuItem3 = new javax.swing.JMenuItem();    jMenuItem2 = new
javax.swing.JMenuItem();

    jMenu8.setText("File");
    jMenuBar2.add(jMenu8);

    jMenu9.setText("Edit");
    jMenuBar2.add(jMenu9);
    jCheckBoxMenuItem1.setSelected(true);

    jCheckBoxMenuItem1.setText("jCheckBoxMenuItem1");
```



```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jDesktopPane1.setPreferredSize(new java.awt.Dimension(350, 200));

javax.swing.GroupLayout    jDesktopPane1Layout    =    new
javax.swing.GroupLayout(jDesktopPane1);
jDesktopPane1.setLayout(jDesktopPane1Layout);
jDesktopPane1Layout.setHorizontalGroup(

jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 465, Short.MAX_VALUE)

);
jDesktopPane1Layout.setVerticalGroup(

jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 290, Short.MAX_VALUE)

);

jMenu1.setText("File");

jMenuItem1.setText("Customer");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jMenuItem1ActionPerformed(evt);
}
});
jMenu1.add(jMenuItem1);

jMenuItem8.setText("Account");

```

```

jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jMenuItem8ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem8);

jMenuBar1.add(jMenu1);

jMenu2.setText("Transaction");

jMenuItem3.setText("Withdraw");
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jMenuItem3ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem3);

jMenuItem2.setText("Deposit");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

```

```

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);    layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jDesktopPanel1,    javax.swing.GroupLayout.DEFAULT_SIZE,
465, Short.MAX_VALUE)
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jDesktopPanel1,    javax.swing.GroupLayout.DEFAULT_SIZE,
290, Short.MAX_VALUE)
    );

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

```

```

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:    Withdraw wc=new Withdraw();
jDesktopPanel1.add(wc);    wc.setVisible(true);

}

```

```

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
Customer cus = new Customer();    jDesktopPanel1.add(cus);
cus.setVisible(true);

// TODO add your handling code here:

}

```

```

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

    account ac = new account();

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

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:    Deposit ac = new Deposit();

jDesktopPane1.add(ac);    ac.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(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.JCheckBoxMenuItem jCheckBoxMenuItem1;  
private javax.swing.JDesktopPane jDesktopPane1; private  
javax.swing.JMenu jMenu1; private javax.swing.JMenu jMenu2;
```

```

private javax.swing.JMenu jMenuItem8;    private javax.swing.JMenu
jMenuItem9;    private javax.swing.JMenuBar jMenuItemBar1;    private
javax.swing.JMenuBar jMenuItemBar2;    private
javax.swing.JMenuItem jMenuItem1;    private
javax.swing.JMenuItem jMenuItem2;    private
javax.swing.JMenuItem jMenuItem3;    private
javax.swing.JMenuItem jMenuItem8;

    // End of variables declaration
}

```

## **CUSTOMER**

```

/*
 *      Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 *      Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit
this template
 */

```

```

package hsbbank;

```

```

import com.mysql.jdbc.Statement;
import java.awt.Label; import
java.sql.Connection; import
java.sql.PreparedStatement; import
java.sql.ResultSet; import
java.sql.SQLException; import
java.util.logging.Level; import
java.util.logging.Logger; import

```

```

java.sql.DriverManager; import
javax.swing.JOptionPane;

/**
 *
 * @author Admin
 */
public class Customer extends javax.swing.JInternalFrame {

    /**
     * Creates new form Customer
     */
    public Customer() {
        initComponents();
        autoID();    branch();
    }

    /**
     * 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() {

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    txtfname = new javax.swing.JTextField();
    txtlname = new javax.swing.JTextField();
    txtstreet = new javax.swing.JTextField(); txtcity
= new javax.swing.JTextField();    txtphone =
new javax.swing.JTextField();    jButton1 =
new javax.swing.JButton();    jButton2 = new
javax.swing.JButton();    jLabel9 = new
javax.swing.JLabel();    jComboBox1 = new
javax.swing.JComboBox<>();    jLabel11 =
new javax.swing.JLabel();

    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Customer"));

    jLabel2.setText("Customer No");

    jLabel3.setText("First Name");
```



```
jLabel4.setText("Last Name");
```

```
jLabel5.setText("Street");
```

```
jLabel6.setText("City");
```

```
jLabel7.setText("Branch");
```

```
jLabel8.setText("Phone");
```

```
txtfname.addActionListener(new java.awt.event.ActionListener() {  
public void actionPerformed(java.awt.event.ActionEvent evt) {  
txtfnameActionPerformed(evt);  
    } });  
txtcity.  
addAc  
tionLis  
tener(n  
ew  
java.a  
wt.eve  
nt.Acti  
onList  
ener()  
{  
public  
void  
action
```

Perfor

med(ja

va.awt

.event.

Action

Event

evt) {

txtcity

Action

Perfor

med(e

vt);

}

});

jButton1.setText("Add");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Cancel");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

```

jLabel9.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel9.setForeground(new java.awt.Color(0, 0, 204));    jLabel9.setText("jLabel9");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout); jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel1Layout.createSequentialGroup()

        .addGap(28, 28, 28)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addGroup(jPanel1Layout.createSequentialGroup()

                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                            .addGroup(jPanel1Layout.createSequentialGroup()

                                .addComponent(jLabel3)

                                .addComponent(jLabel8)

                                .addComponent(jLabel7)

                                .addComponent(jLabel6)

                                .addComponent(jLabel5)

                                .addComponent(jLabel4))

                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                                .addGroup(jPanel1Layout.createSequentialGroup()

                                    .addGap(0, 0, Short.MAX_VALUE)

                                    .addComponent(jLabel2)

                                    .addGap(63, 63, 63)))
                        )
                    )
                )
            )
        )
    )
)

```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addComponent(jLabel9)
```

```
    .addComponent(txtfname)
```

```
    .addComponent(txtlname)
```

```
    .addComponent(txtstreet)
```

```
    .addComponent(txtcity)
```

```
    .addComponent(txtphone)
```

```
    .addComponent(jComboBox1, 0, 135, Short.MAX_VALUE))
```

```
.addGap(67, 67, 67)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE, 96, Short.MAX_VALUE))
```

```
    .addGap(25, 25, 25))
```

```
);
```

```
jPanel1Layout.setVerticalGroup(
```

```
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(jPanel1Layout.createSequentialGroup())
```

```
    .addGap(17, 17, 17)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel2)
```

```
    .addComponent(jLabel9))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel3)
```

```
    .addComponent(txtfname, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,  
47, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(17, 17, 17)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel4)
```

```
    .addComponent(txtlname, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel5)
```

```
    .addComponent(txtstreet, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel6)
```

```
    .addComponent(txtcity, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,  
47, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```

        .addGap(17, 17, 17)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel7)

            .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel8)

            .addComponent(txtphone, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addContainerGap(47, Short.MAX_VALUE))

    );

    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
    jLabel1.setText("Customer");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);    layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(51, 51, 51)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel1))

            .addContainerGap(24, Short.MAX_VALUE))
    );

```

```

);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

```

```

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

```

```

Connection con;

PreparedStatement insert;

```

```

public void autoID()
{
try {
    Class.forName("com.mysql.jdbc.Driver");

    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
Statement s = (Statement) con.createStatement();

```

```

        ResultSet rs = s.executeQuery("select Max(cust_id) from customer");
rs.next();

        rs.getString("Max(cust_id)");
if(rs.getString("Max(cust_id)")==null)        {

            jLabel9.setText("CS001");

        }
else

        {

            long                                id                                =
Long.parseLong(rs.getString("Max(cust_id)").substring(2,rs.getString("Max(cust_id)").length()
));

            id++;

            jLabel9.setText("CS" + String.format("%03d", id));

        }
    } catch (ClassNotFoundException ex) {

        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {

        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```



```

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

}

public void branch() {
    try {
        Class.forName("com.mysql.jdbc.Driver");

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
insert = con.prepareStatement("select * from branch");        ResultSet rs =
insert.executeQuery();        jComboBox1.removeAllItems();

        while (rs.next())
        {
            jComboBox1.addItem(rs.getString(1));
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try {

    // TODO add your handling code here:

    String cust_id = jLabel9.getText();

    String firstname = txtfname.getText();

    String lastname = txtlname.getText();

    String street = txtstreet.getText();

    String city = txtcity.getText();

    String branch= jComboBox1.getSelectedItem().toString();

    String phone = txtphone.getText();


    Class.forName("com.mysql.jdbc.Driver");

    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");

    insert = con.prepareStatement("insert into
customer(cust_id,firstname,lastname,street,city,branch,phone)values(?,?,?,?,?,?,?)");

    insert.setString(1,cust_id);      insert.setString(2,firstname);
    insert.setString(3,lastname);      insert.setString(4,street);
    insert.setString(5,city);          insert.setString(6,branch);
    insert.setString(7,phone);          insert.executeUpdate();


    JOptionPane.showMessageDialog(this,"Record Added");


    txtfname.setText("");
    txtlname.setText("");
    txtstreet.setText("");      txtcity.setText("");

```

```

jComboBox1.setSelectedIndex(-1);

txtphone.setText("");      autoID();

    txtfname.requestFocus();

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);

    } catch (SQLException ex) {

        Logger.getLogger(Customer.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    setVisible(false);

dispose();

}

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

}

```

```

// Variables declaration - do not modify

private javax.swing.JButton jButton1; private
javax.swing.JButton jButton2;

private javax.swing.JComboBox<String> jComboBox1;

private javax.swing.JLabel jLabel1; private
javax.swing.JLabel jLabel2; private javax.swing.JLabel
jLabel3; private javax.swing.JLabel jLabel4; private
javax.swing.JLabel jLabel5; private javax.swing.JLabel
jLabel6; private javax.swing.JLabel jLabel7; private
javax.swing.JLabel jLabel8; private javax.swing.JLabel
jLabel9; private javax.swing.JPanel jPanel1; private
javax.swing.JTextField txtcity; private
javax.swing.JTextField txtfname; private
javax.swing.JTextField txtlname; private
javax.swing.JTextField txtphone; private
javax.swing.JTextField txtstreet;

// End of variables declaration
}

```

## ACCOUNT

```

package hsbbank;

import com.mysql.jdbc.Statement;
import java.awt.Label; import
java.sql.Connection; import
java.sql.PreparedStatement; import
java.sql.ResultSet; import
java.sql.SQLException; import
java.util.logging.Level; import

```

```
java.util.logging.Logger; import
java.sql.DriverManager; import
javax.swing.JOptionPane;
```

```
/**
```

```
*
```

```
* @author Admin
```

```
*/
```

```
public class account extends javax.swing.JInternalFrame {
```

```
/**
```

```
* Creates new form Customer
```

```
*/
```

```
public account() {
```

```
    initComponents();
```

```
    autoID();    branch();
```

```
}
```

```
/**
```

```
* 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() {

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();    jLabel3 =
new javax.swing.JLabel();    jLabel4 = new
javax.swing.JLabel();    jLabel5 = new
javax.swing.JLabel();    jLabel6 = new
javax.swing.JLabel();    txtcust_id = new
javax.swing.JTextField();    txtlname = new
javax.swing.JTextField();    txtbal = new
javax.swing.JTextField();    jButton1 = new
javax.swing.JButton();    jButton2 = new
javax.swing.JButton();    jLabel9 = new
javax.swing.JLabel();    txtaccount = new
javax.swing.JComboBox<>();    jButton3 = new
javax.swing.JButton();

    jLabel11 = new javax.swing.JLabel();

    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Account"));

    jLabel2.setText("Account No");

    jLabel3.setText("Customer id");

    jLabel4.setText("Customer name");

    jLabel5.setText("Account type");
```

```

jLabel6.setText("Balance");

    txtcust_id.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
    txtcust_idActionPerformed(evt);
        }
    });

    txtcust_id.addKeyListener(new java.awt.event.KeyAdapter() {
public void keyPressed(java.awt.event.KeyEvent evt) {
    txtcust_idKeyPressed(evt);
        }
    });

    txtbal.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
    txtbalActionPerformed(evt);
        }
    });

    jButton1.setText("Add");

    jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
    jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Cancel");

```

```

        jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);

        }

});

```

```

        jLabel9.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel9.setForeground(new java.awt.Color(0, 0, 204));        jLabel9.setText("jLabel9");

```

```

        txtaccount.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Savings", "Current", "Fix" }));

```

```

        jButton3.setText("Find");

        jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);

        }

});

```

```

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);        jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addGap(28, 28, 28)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel3)

```



```

        .addComponent(jLabel6)

        .addComponent(jLabel5)

        .addComponent(jLabel4)

        .addComponent(jLabel2))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

            .addComponent(jLabel9)

            .addComponent(txtcust_id)

            .addComponent(txtlname)

            .addComponent(txtaccount, 0, 135, Short.MAX_VALUE))

            .addComponent(txtbal, javax.swing.GroupLayout.PREFERRED_SIZE, 135,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(67, 67, 67))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

        .addGroup(

            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
96, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
96, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE,
96, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(43, 43, 43))

);

jPanel1Layout.setVerticalGroup(

```

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(17, 17, 17)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(jLabel9))
        .addGap(41, 41, 41)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(txtcust_id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE,
47, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(31, 31, 31)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
47, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtlname, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel4))
            .addGap(29, 29, 29)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jButton2,      javax.swing.GroupLayout.PREFERRED_SIZE,
47, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtaccount,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel5))

    .addGap(45, 45, 45)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)

    .addComponent(jLabel6)

    .addComponent(txtbal,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(158, Short.MAX_VALUE))

);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel1.setText("Account");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);    layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

        .addContainerGap(19, Short.MAX_VALUE)

        .addComponent(jLabel1)

        .addGap(480, 480, 480))

    .addGroup(layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(jPanel1,    javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addContainerGap()
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

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

```

```

Connection con;

```

```

PreparedStatement insert;

```

```

ResultSet rs;

```

```

public void autoID()
{
    try {
        Class.forName("com.mysql.jdbc.Driver");

        con

```

=

```

DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
Statement s = (Statement) con.createStatement();

        ResultSet rs = s.executeQuery("select  Max(account_no) from account");
rs.next();

        rs.getString("Max(account_no)");
if(rs.getString("Max(account_no)")==null)
    {
        jLabel9.setText("A0001");
    }
else
    {
        long id = Long.parseLong(rs.getString(1).substring(2,rs.getString(1).length()));
id++ ;

        jLabel9.setText("A" + String.format("%04d", id));
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
}
}

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

public void branch() {

```

```

try {
    Class.forName("com.mysql.jdbc.Driver");

    con
    DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
    insert = con.prepareStatement("select * from branch");

    ResultSet rs = insert.executeQuery();

    // jComboBox1.removeAllItems();

    while (rs.next())
    {
        //jComboBox1.addItem(rs.getString(2));
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
}

}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    // TODO add your handling code here:

    String account_no = jLabel9.getText();

    String customerid = txtcust_id.getText();

    String custname = txtlname.getText();

```

```

        String acctype= txtaccount.getSelectedItem().toString();
String bal = txtbal.getText();

        Class.forName("com.mysql.jdbc.Driver");

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");

        insert = con.prepareStatement("insert into
account(account_no,customerid,acctype,bal)values(?,?,?,?)");
insert.setString(1,account_no);        insert.setString(2,customerid);
insert.setString(3,acctype);        insert.setString(4,bal);        insert.executeUpdate();

        JOptionPane.showMessageDialog(this,"Account Created");

        txtcust_id.setText("");
txtlname.setText("");        txtbal.setText("");
        txtaccount.setSelectedIndex(-1);

        autoID();
        txtcust_id.requestFocus();

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);

    } catch (SQLException ex) {
Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

setVisible(false);
dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
try {
// TODO add your handling code here:

String cust_id = txtcust_id.getText();

Class.forName("com.mysql.jdbc.Driver");

con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
insert = con.prepareStatement("select * from customer where cust_id = ?" );
insert.setString(1, cust_id);      rs = insert.executeQuery();

if(rs.next()==false)
{
JOptionPane.showMessageDialog(this,"Customer No Not Found");
}
else
{

```



```

        String firstname = rs.getString("firstname");
txtlname.setText(firstname.trim());

    }

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {

        Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
    }

}

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

}

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

}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1; private
javax.swing.JLabel jLabel2; private
javax.swing.JLabel jLabel3; private
javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6; private
javax.swing.JLabel jLabel9; private
javax.swing.JPanel jPanel1;

```

```

    private javax.swing.JComboBox<String> txtaccount;
private javax.swing.JTextField txtbal;    private
javax.swing.JTextField txtcust_id;    private
javax.swing.JTextField txtlname;

    // End of variables declaration
}

```

## DEPOSIT

```

try {

    // TODO add your handling code here:


    con.setAutoCommit(false);

    String accno = txtacc.getText();

    String cust_id = jLabel13.getText();

    String firstname = jLabel14.getText();

    String lastname = jLabel15.getText();

    String date = jLabel16.getText();

    String balance = jLabel6.getText();

    String amount = txtamount.getText();


    Class.forName("com.mysql.jdbc.Driver");

    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");


    insert =    con.prepareStatement("insert into
deposit(acc_id,cust_id,date,balance,deposit)values(?,?,?,?);");

    insert.setString(1,accno);

insert.setString(2,cust_id);        insert.setString(3,date);

insert.setString(4,balance);

```

```

insert.setString(5,amount);
insert.executeUpdate();

        update = con.prepareStatement( "update Account set bal = bal+ ? where
account_no=?");
        update.setString(1, amount);
update.setString(2, accno);

        update.executeUpdate();

        JOptionPane.showMessageDialog(this,"Deposited Succesfully ");
con.commit();

    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {

try {
            con.rollback();
        } catch (SQLException ex1) {
            Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex1);
        }
        Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
    }

}

public void date()

```

```

{
    DateTimeFormatter dtd = DateTimeFormatter.ofPattern("yyyy/MM/dd");

    LocalDateTime now = LocalDateTime.now();

    String date = dtd.format(now);

    jLabel16.setText(date);

} try {
    // TODO add your handling code here:

    String accno = txtacc.getText();

    Class.forName("com.mysql.jdbc.Driver");

    con
    DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
    insert = con.prepareStatement("select      c.cust_id,c.firstname,c.lastname,a.bal      from
customer c, account a where c.cust_id = a.customerid and a.account_no = ?");
    insert.setString(1, accno);      rs = insert.executeQuery();      if(rs.next() == false)
    {
        JOptionPane.showMessageDialog(this,"Account not found");
    }
else
    {
        String id = rs.getString(1);

        String firstname = rs.getString(2);

        String lastname = rs.getString(3);

        String balance = rs.getString(4);

        jLabel13.setText(id.trim());

        jLabel14.setText(firstname.trim());

        jLabel15.setText(lastname.trim());

        jLabel6.setText(balance.trim());
    }
}

```

```

    }

    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
    }

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    setVisible(false);
dispose();
}

```

## **WITHDRAW**

```

try {
    // TODO add your handling code here:

    con.setAutoCommit(false);
    String accno = txtacc.getText();
    String cust_id = jLabel13.getText();
    String firstname = jLabel14.getText();

```

```

String lastname = jLabel15.getText();

String date = jLabel16.getText();

String balance = jLabel6.getText();

String amount = txtamount.getText();


Class.forName("com.mysql.jdbc.Driver");

con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");


insert = con.prepareStatement("insert into
withdraw(acc_id,cust_id,date,balance,withdraw)values(?,?,?,?,?)");

insert.setString(1,accno);      insert.setString(2,cust_id);
insert.setString(3,date);      insert.setString(4,balance);
insert.setString(5,amount);      insert.executeUpdate();


update = con.prepareStatement( "update Account set bal = bal- ? where account_no=?");
update.setString(1, amount);
update.setString(2, accno);

update.executeUpdate();


JOptionPane.showMessageDialog(this,"Withdrawn Succesfully ");
con.commit();


} catch (ClassNotFoundException ex) {

    Logger.getLogger(Withdraw.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {

```

```

try {

    con.rollback();

} catch (SQLException ex1) {

    Logger.getLogger(Withdraw.class.getName()).log(Level.SEVERE, null, ex1);

}

Logger.getLogger(Withdraw.class.getName()).log(Level.SEVERE, null, ex);

}

public void date()
{
    DateTimeFormatter dtd = DateTimeFormatter.ofPattern("yyyy/MM/dd");
    LocalDateTime now = LocalDateTime.now();
    String date = dtd.format(now);

    jLabel16.setText(date);

}

try {

    // TODO add your handling code here:

    String accno = txtacc.getText();

    Class.forName("com.mysql.jdbc.Driver");

    con
    DriverManager.getConnection("jdbc:mysql://localhost:3306/hsbank","root","Surya@2004!");
    insert = con.prepareStatement("select      c.cust_id,c.firstname,c.lastname,a.bal      from
customer c, account a where c.cust_id = a.customerid and a.account_no = ?");
    insert.setString(1, accno);      rs = insert.executeQuery();      if(rs.next() == false)

    {

        JOptionPane.showMessageDialog(this,"Account not found");
    }
}

```

```

        }
else
    {
        String id = rs.getString(1);
        String firstname = rs.getString(2);
        String lastname = rs.getString(3);
String balance = rs.getString(4);
jLabel13.setText(id.trim());
jLabel14.setText(firstname.trim());
jLabel15.setText(lastname.trim());
jLabel6.setText(balance.trim());
    }

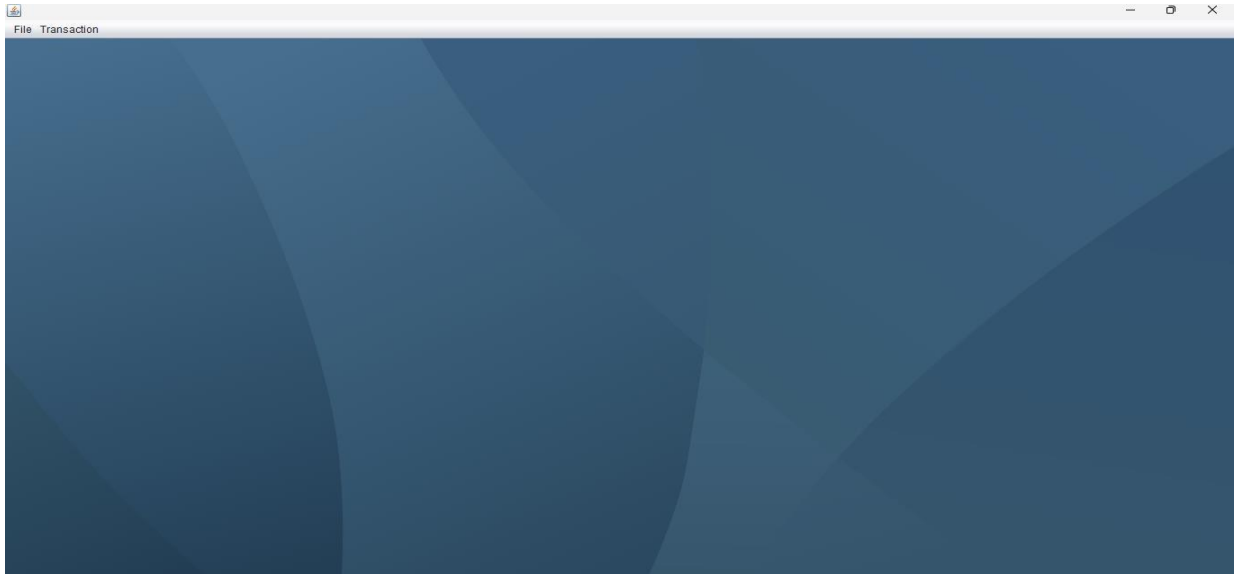
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Withdraw.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Withdraw.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

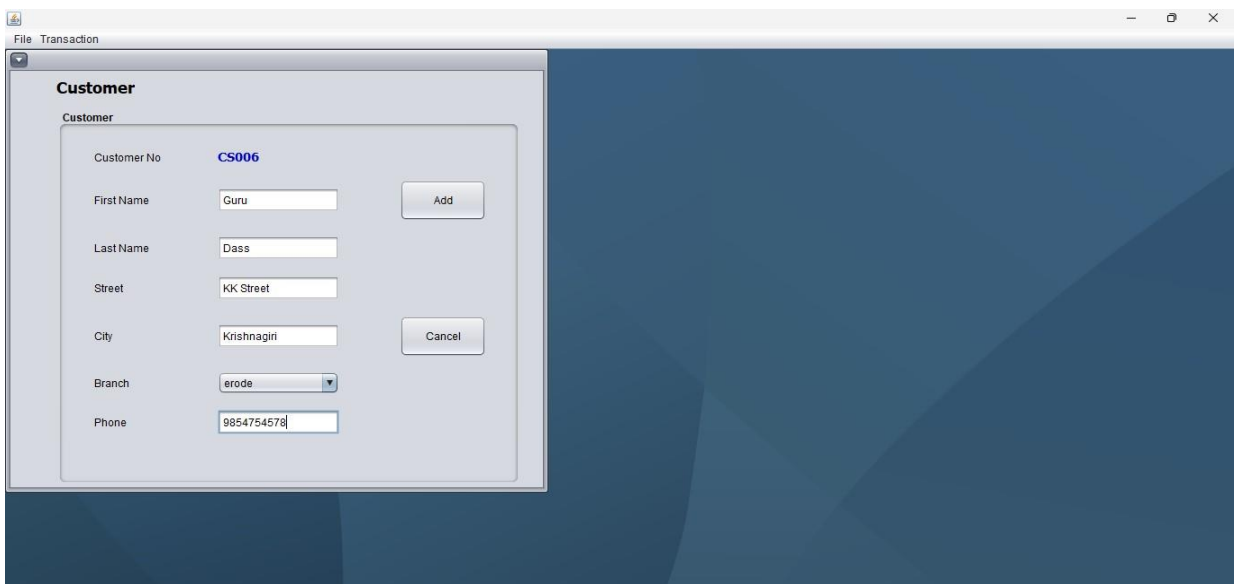


## 7.2 SCREENSHOTS

### HOME PAGE:



### CUSTOMER PAGE

A screenshot of a web application window titled "File Transaction". The window displays a dark blue background with a subtle geometric pattern. Overlaid on the left side is a "Customer" form. The form has a title bar "Customer" and a sub-header "Customer". It contains the following fields: "Customer No" (displaying "CS006" in blue), "First Name" (text input with "Guru"), "Last Name" (text input with "Dass"), "Street" (text input with "KK Street"), "City" (text input with "Krishnagiri"), "Branch" (dropdown menu with "erode" selected), and "Phone" (text input with "9954754576"). There are "Add" and "Cancel" buttons to the right of the form fields.

Customer	
Customer No	CS006
First Name	Guru
Last Name	Dass
Street	KK Street
City	Krishnagiri
Branch	erode
Phone	9954754576

## ACCOUNT PAGE

The screenshot shows a window titled "Transaction" with a menu bar containing "File" and "Transaction". Inside the window is a sub-window titled "Account". The "Account" sub-window has a title bar and a close button. It contains a form with the following fields and controls:

- Account No:** A0006
- Customer id:** CS005, with a "Find" button to its right.
- Customer name:** sehshan, with an "Add" button to its right.
- Account type:** Savings (selected in a dropdown menu), with a "Cancel" button to its right.
- Balance:** 10000

## DEPOSIT PAGE

The screenshot shows a window titled "Transaction" with a menu bar containing "File" and "Transaction". Inside the window is a sub-window titled "Deposit". The "Deposit" sub-window has a title bar and a close button. It contains a form with the following fields and controls:

- Account No:** A0001, with a "Find" button to its right.
- Balance:** 8900
- Deposit:** 1000 (displayed in a blue box)
- Customer No:** CS001
- Firstname:** Praveen
- Lastname:** S
- Date:** 2023/06/01
- Buttons:** OK and Cancel

## WITHDRAW

The screenshot shows a 'Transaction' window with a light blue background. The window has a title bar with 'File' and 'Transaction' menus. The main content area is divided into several sections:

- Account No:** A text box containing 'A0001' with a 'Find' button below it.
- Balance:** A label 'Balance' followed by the value '8900'.
- Deposit:** A label 'Deposit' followed by a blue box containing the value '1000'.
- Customer Information:** A table with the following data:

Customer No	CS001
Firstname	Praveen
Lastname	S
Date	2023/06/01
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

## **CHAPTER 8**

### **REFERENCES**

1. Choudhary, A., & Naidu, R. (2018). Banking management system using Java and MySQL. *International Journal of Computer Science and Information Technologies*, 9(2), 101-106.
2. Salaria, N. (2011). Banking management system: A framework. *International Journal of Computer Science and Information Security*, 9(6), 106-112.
3. <https://www.mysql.com/>
4. <https://www.java.com/en/>