



**NAMIBIA UNIVERSITY**  
OF SCIENCE AND TECHNOLOGY

## **Topic: Phonebook Application**

<b>Student Name</b>	<b>Student Number</b>
Natanael N. Treves	224032143
Wapingena Muukua	223080071
Ally M. Kafidi	224088092
Iiyambo T. Nangombe	224049844
Simeon P. Penda	224013386
Uahenua P.M Hengari	224085050

DSA Project

Group: 20

## OVERVIEW

This Phonebook Application is designed for a Namibian telecommunications company to efficiently manage contacts. It allows users to insert, search, update, delete, and display contacts using basic linear data structures. The application focuses on providing a simple and user-friendly interface for managing contact information.

## FEATURES

**Insert Contact:** Add new contacts to the phonebook.

**Search Contact:** Find contacts by name.

**Update Contact:** Modify existing contact information.

**Delete Contact:** Remove contacts from the phonebook.

**Display Contacts:** List all contacts in the phonebook.

**Sort Contacts:** (Optional) Organize contacts alphabetically for quicker access.

**Efficiency Analysis:** Analyse the performance of the search function.

## MODULES

### 1. Insert Module

Function: InsertContact(String name, String phoneNumber) Purpose:

Adds a new contact to the phonebook.

### 2. Search Module

Function: SearchContact(String name)

Purpose: Searches for a contact by name and returns the contact details.

### **3. Update Module**

Function: UpdateContact(String name, String newPhoneNumber) Purpose:

Updates the phone number of an existing contact.

### **4. Delete Module**

Function: DeleteContact(String name)

Purpose: Deletes a contact from the phonebook.

### **5. Display Module**

Function: DisplayContacts()

Purpose: Displays all contacts currently stored in the phonebook.

## **PSEUDOCODE**

### **INSERT MODULE:**

Start

```
FUNCTION InsertContact(phonebook, name, phoneNumber)
    NEW_NODE = CREATE_NODE(name, phoneNumber)
    IF phonebook IS EMPTY THEN      phonebook =
    NEW_NODE
    ELSE
        CURRENT = phonebook
        WHILE CURRENT.next IS NOT NULL DO
            CURRENT = CURRENT.next
        END WHILE
        CURRENT.next = NEW_NODE
    END IF
END FUNCTION
```

End

### **SEARCH MODULE:**

Start

```
FUNCTION SearchContact(phonebook, name)
    CURRENT = phonebook
    WHILE CURRENT IS NOT NULL DO
        IF CURRENT.name == name THEN
            RETURN CURRENT.phoneNumber
        END IF
        CURRENT = CURRENT.next
    END WHILE
    RETURN "Contact not found"
END FUNCTION
```

End

### **DISPLAY MODULE:**

BEGIN

Prompt user for contact details  
Get Contacts

If (Contacts name and number not found) THEN  
 Display ‘No contacts available’  
END

ELSE If (contact name AND number found) THEN  
 Display ‘Contact Details found’

```
    END IF
  END IF
END
```

### **DELETE MODULE :**

START

DATA: contact name  
PHONE: contact phone number  
NEXT: pointer to next contact  
PREV: pointer to previous contact

CONSTRUCTOR Node(name, phone, email)  
DATA = name  
PHONE = phone  
NEXT = NULL  
PREV = NULL

FUNCTION deleteAtGivenPos(head)  
 TEMP = head  
 DISPLAY "Enter position where you want to delete"  
 GET position

```
    i = 1
    WHILE(i < position - 1)
    IF temp.next is null
      DISPLAY "Contact not found"
      RETURN head
    ENDIF
```

```
    TEMP = TEMP.NEXT
    i = i + 1
  ENDWHILE
```

```
  IF temp.next is null
    DISPLAY "Contact not found"
    RETURN head
  ENDIF
```

```
nextnode = temp.next
temp.next = nextnode.next
```

```
  IF nextnode.next is not null
    nextnode.next.prev = temp
  ENDIF
```

```
nextnode = free
DISPLAY "Contact deleted successfully"
```

RETURN head

head = deleteAtGivenPos(head)

END

### **UPDATE MODULE:**

Start

```
FUNCTION UpdateContact(phonebook, name, newContact)
  FOR EACH contact IN phonebook
    IF contact.name == name
      contact = newContact
RETURN "Contact updated"
END IF
END FOR
RETURN "Contact not found"
END FUNCTION
```

End

### **SORT MODULE:**

Start

```
FUNCTION SortContacts(phonebook)
  IF phonebook IS NULL OR phonebook.next IS NULL THEN
    RETURN phonebook
  END IF
  SPLIT_LINKED_LIST(phonebook, LEFT, RIGHT)
  LEFT = SortContacts(LEFT)
  RIGHT = SortContacts(RIGHT)
  RETURN MERGE(LEFT, RIGHT)
END FUNCTION
```

End

### **ANALYSE THE EFFICIENCY OF YOUR SEARCH ALGORITHM**

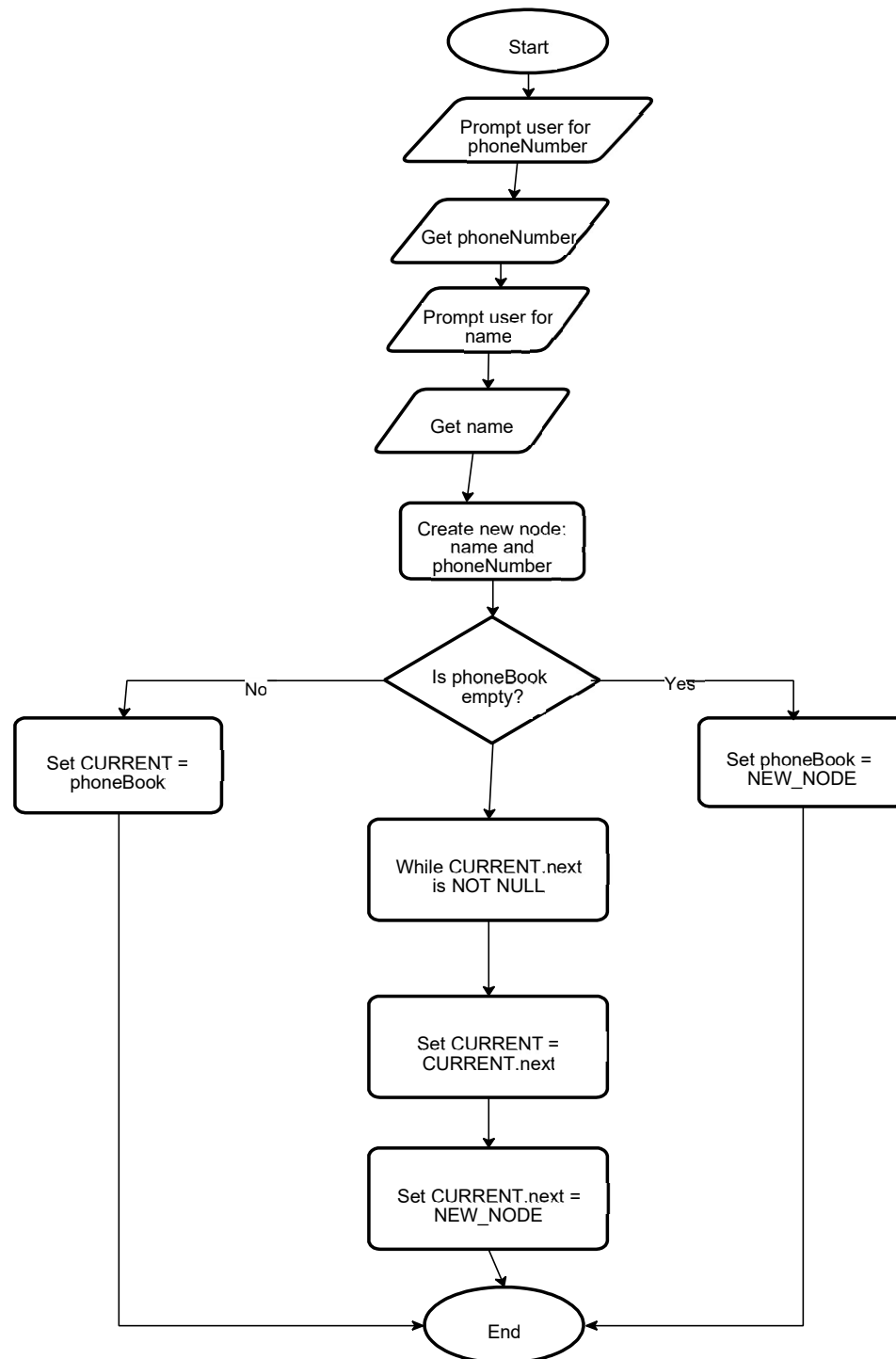
Start

```
FUNCTION AnalyzeSearchEfficiency(phonebook, name)
START_TIME = CURRENT_TIME()
result = SearchContact(phonebook, name)
END_TIME = CURRENT_TIME()
TIME_TAKEN = END_TIME - START_TIME
PRINT "Search result:", result
PRINT "Time taken:", TIME_TAKEN
END FUNCTION
```

End

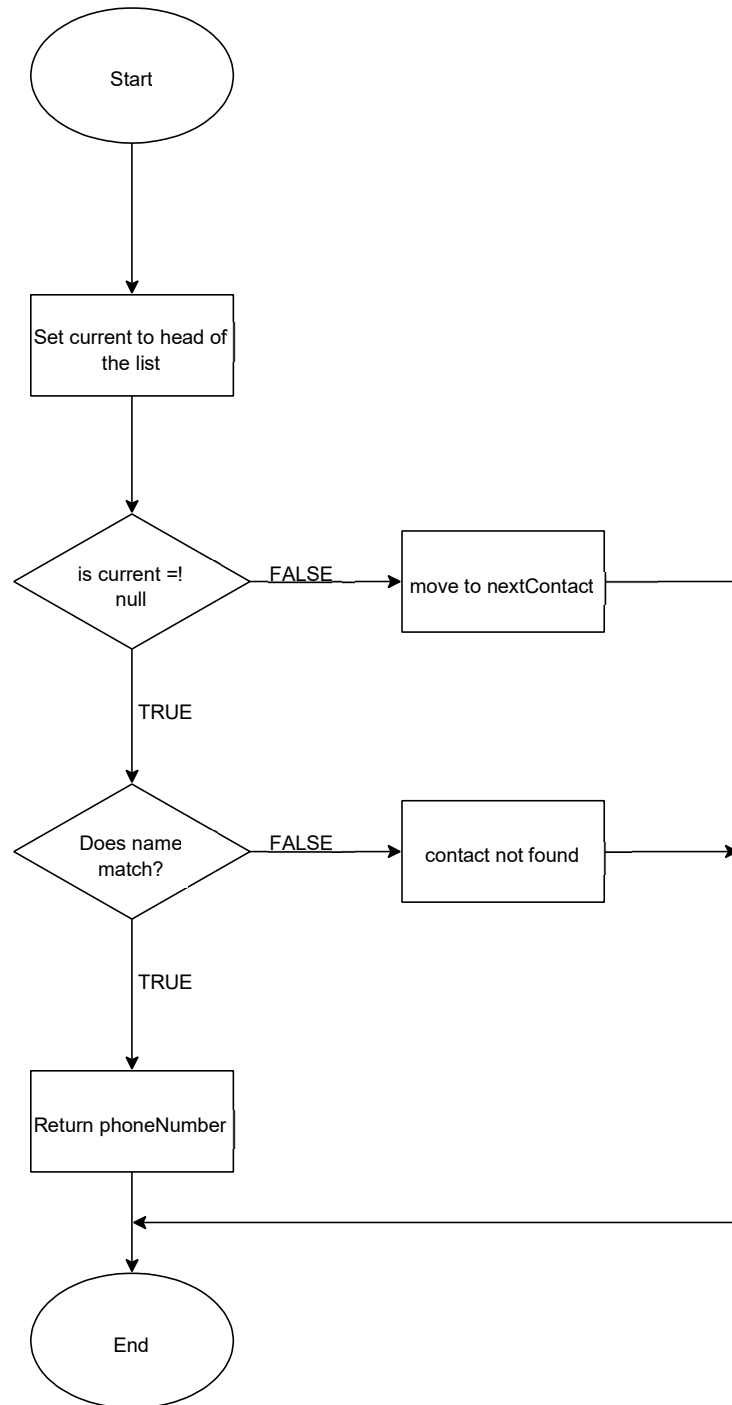
## FLOWCHARTS

### INSERT MODULE:

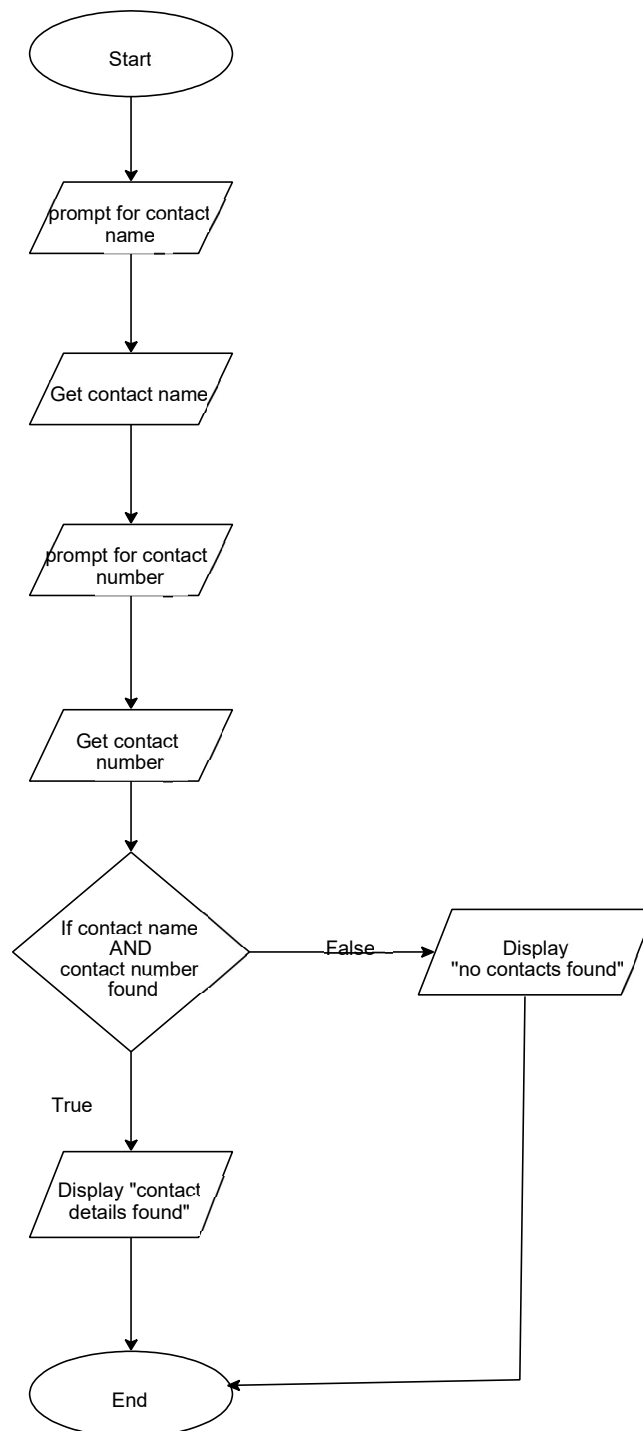




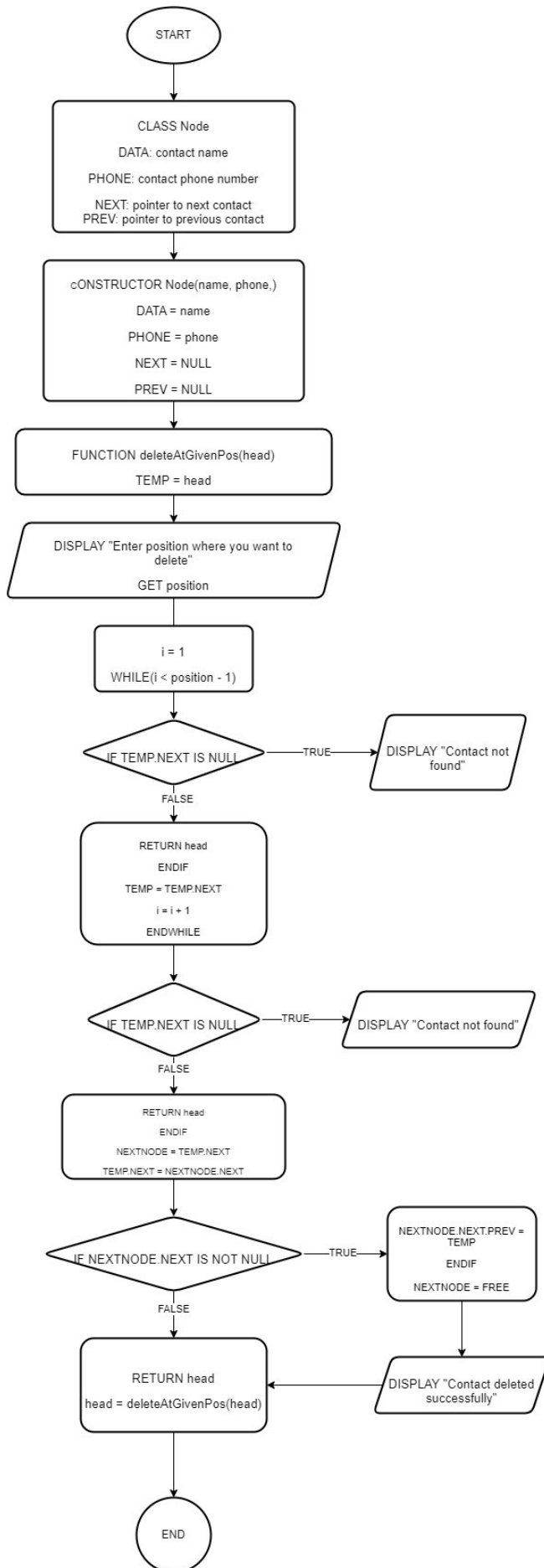
## SEARCH MODULE:



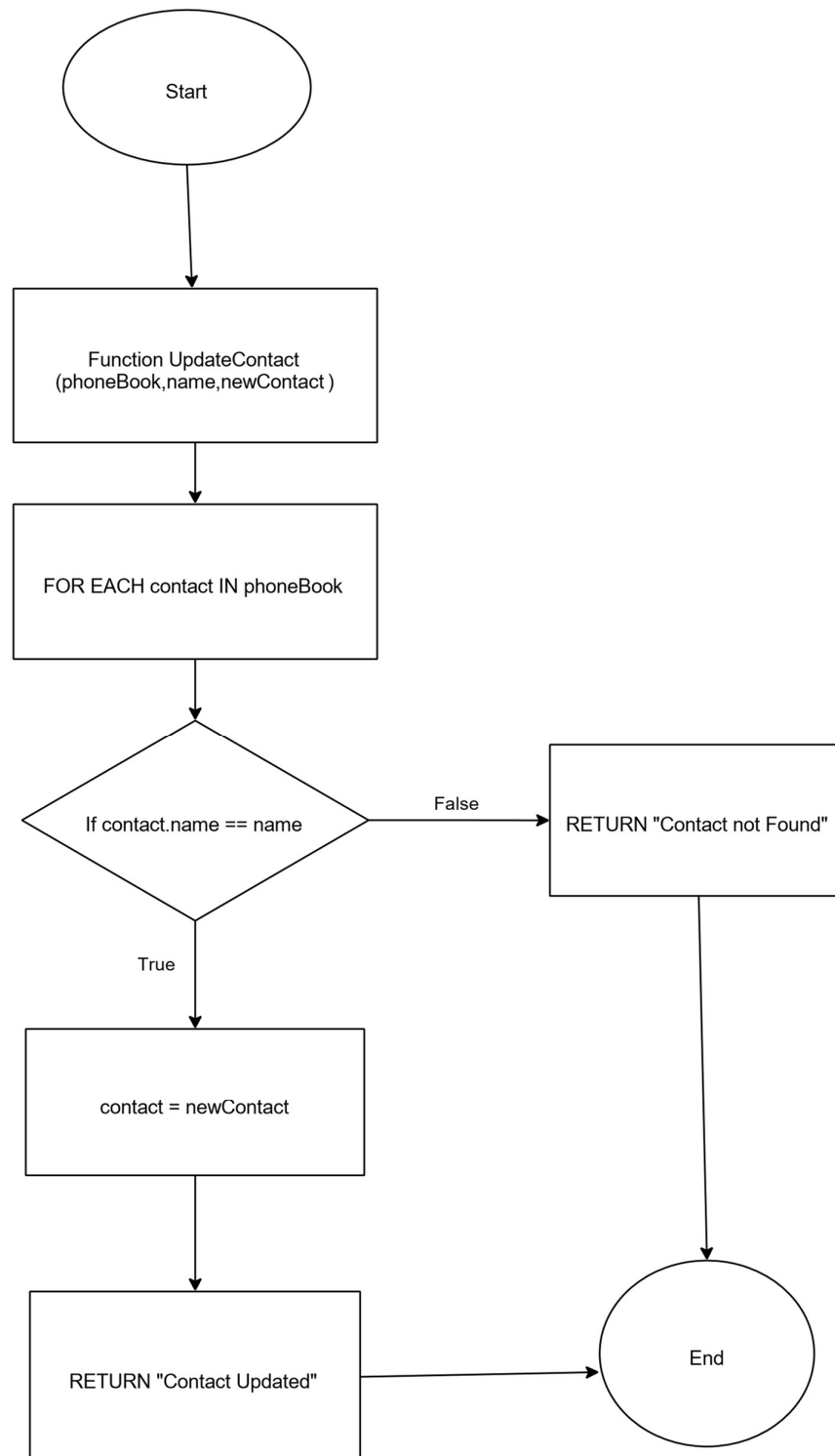
## DISPLAY MODULE:



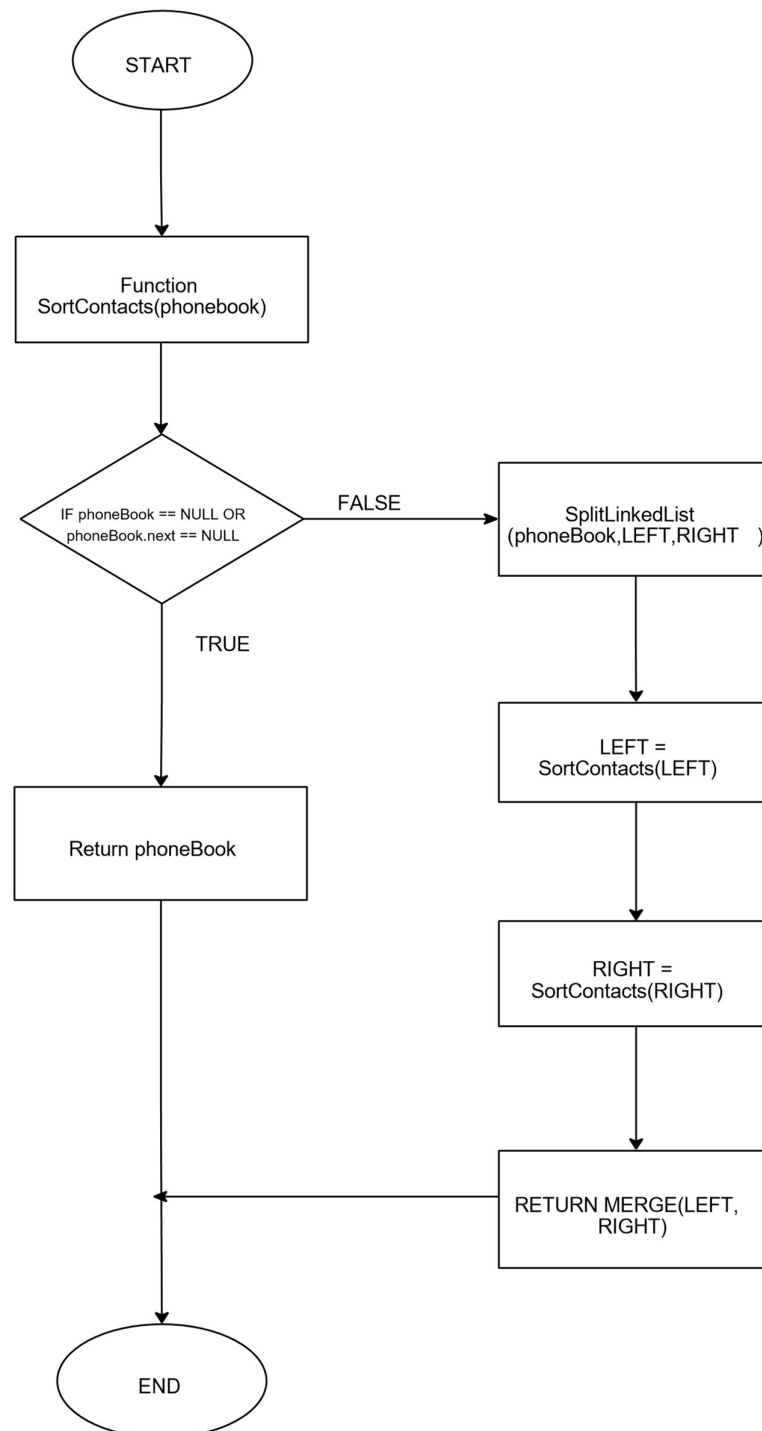
## DELETE MODULE:



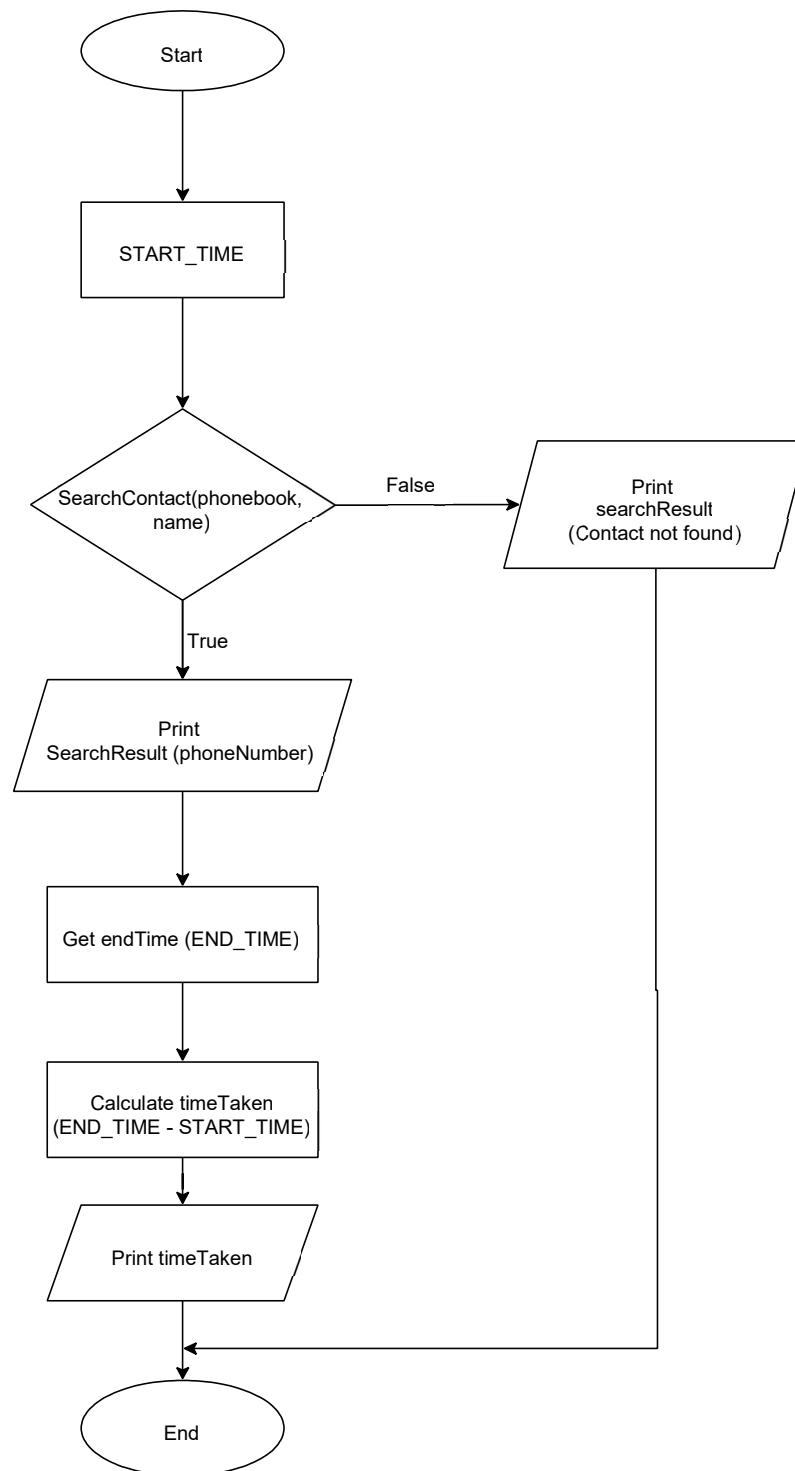
## UPDATE MODULE:



## SORT MODULE:



## ANALYSE THE EFFICIENCY OF YOUR SEARCH ALGORITHM



## Contributors

Student Name	Contribution
Uahenua P.M Hengari	Pseudocode and flowchart for Insert Contact.
Natanael N. Treves	Pseudocode and flowchart for Search Contact. Pseudocode and flowchart for analyse the efficiency of your search Algorithm.
Iiyambo T. Nangombe	Pseudocode and flowchart for Display Contacts.
Wapingena Muukua	Pseudocode and flowchart for Delete Contact
Ally M. Kafidi	Pseudocode and flowchart for Update Contact.
Simeon P. Penda	Pseudocode and flowchart for Sorting Contact.

## CODES

### Admin login code.

```
package phnbook;

import java.awt.Component;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```

import java.sql.ResultSetMetaData;
import javax.swing.JOptionPane;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author KTCC
 */
public class Admin extends javax.swing.JFrame {

    /**
     * Creates new form Admin
     */
    public Admin() {
        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() {

        jLabel3 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
    }

```



```

jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jPanel2 = new javax.swing.JPanel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
cnCancel = new javax.swing.JButton();
txtusername = new javax.swing.JTextField();
txtpass = new javax.swing.JPasswordField();
cnLog = new javax.swing.JButton();

jLabel3.setText("jLabel3");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(153, 153, 153));
jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel1.setText("Admin Login ");

jLabel2.setText("Please enter your login details!!");

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(177, 177, 177)
            .addComponent(jLabel1)
        )
    );

```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())

        .addContainerGap(139, Short.MAX_VALUE)

        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 175,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(134, 134, 134))

    );

    jPanel1Layout.setVerticalGroup(

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

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addContainerGap()

        .addComponent(jLabel1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel2)

        .addContainerGap(10, Short.MAX_VALUE))

    );

    jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0), 2));

    jLabel4.setText("Username");

    jLabel5.setText("Password");

    cnCancel.setBackground(new java.awt.Color(102, 102, 102));

    cnCancel.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

    cnCancel.setText("Cancel");

    cnCancel.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.Bevel
Border.RAISED));

```

```

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

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

cnLog.setBackground(new java.awt.Color(102, 102, 102));
cnLog.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
cnLog.setText("log in");

cnLog.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBo
rder.RAISED));

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

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(22, 22, 22)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(68, 68, 68)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

    .addComponent(txtusername)

    .addComponent(txtpass, javax.swing.GroupLayout.DEFAULT_SIZE, 176,
Short.MAX_VALUE))

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    .addComponent(cnCancel, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(cnLog, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addContainerGap())

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup())

.addGap(60, 60, 60)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel4)

```

```

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

        .addGap(60, 60, 60)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel5)

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

        .addGap(31, 133, Short.MAX_VALUE))

    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(cnLog, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(cnCancel, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap())

    );

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

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

        .addGroup(

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

            .addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

```

```

);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        );

pack();

setLocationRelativeTo(null);
} // </editor-fold>

```

```

private void cnCancelActionPerformed(java.awt.event.ActionEvent evt) {

    Component frame = new JFrame("Exit");

    if (JOptionPane.showConfirmDialog(frame, "Confirm if you want to exit","Contact
Management",
        JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
    {
        System.exit(0);
    }
}

```

```

private void cnLogActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // Get the username and password from your input fields

    String username = txtusername.getText(); // Assume this is your username input field

```

```
String passwordInput = new String(txtpass.getPassword()); // Assume this is your password  
input field
```

```
// Database connection details
```

```
String url = "jdbc:mysql://localhost:3306/phonebook";
```

```
String user = "root";
```

```
String password = "6161586";
```

```
Connection conn = null;
```

```
PreparedStatement pstmt = null;
```

```
ResultSet rs = null;
```

```
try {
```

```
    // Establish the connection
```

```
    conn = DriverManager.getConnection(url, user, password);
```

```
    // Prepare the SQL query
```

```
    String sql = "SELECT * FROM admin WHERE user_name = ? AND password = ?";
```

```
    pstmt = conn.prepareStatement(sql);
```

```
    pstmt.setString(1, username);
```

```
    pstmt.setString(2, passwordInput);
```

```
    // Execute the query
```

```
    rs = pstmt.executeQuery();
```

```
    // Check if a result was returned
```

```
    if (rs.next()) {
```

```
        // Login successful, proceed to the next JFrame
```

```
        new contact().setVisible(true); // Assuming ContactFrame is your next JFrame
```

```
        this.dispose(); // Close the current JFrame if needed
```

```
    } else {
```

```

        // Show an error message if login fails

        JOptionPane.showMessageDialog(this, "Invalid username or password.", "Login
Failed", JOptionPane.ERROR_MESSAGE);

    }

    } catch (SQLException e) {

        e.printStackTrace();

        JOptionPane.showMessageDialog(this, "Database connection error.", "Error",
JOptionPane.ERROR_MESSAGE);

    } finally {

        // Close resources

        try {

            if (rs != null) rs.close();

            if (pstmt != null) pstmt.close();

            if (conn != null) conn.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

}

```

```

private void txtusernameActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

/**

 * @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(Admin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (InstantiationException ex) {
```

```
        java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
        } catch (IllegalAccessException ex) {
```

```
            java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
                java.util.logging.Logger.getLogger(Admin.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 Admin().setVisible(true);
```

```

    }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton cnCancel;
private javax.swing.JButton cnLog;
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.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPasswordField txtpass;
private javax.swing.JTextField txtusername;
// End of variables declaration
}

```

## ConnectionToDB

```

package phnbook;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class connectionToDB {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/connector"; // Change
    database name, if needed

    private static final String DB_USER = "root"; // Your MySQL username

```

```

private static final String DB_PASSWORD = "6161586"; // Your MySQL password

// Method to establish the connection
public static Connection getConnection() {
    Connection conn = null;

    try {
        // Register the MySQL driver (optional for modern versions of JDBC)
        Class.forName("com.mysql.cj.jdbc.Driver");    // MySQL 8 and later use
        `com.mysql.cj.jdbc.Driver`

        // Attempt to establish a connection
        conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        System.out.println("Connection to database established successfully.");
    } catch (ClassNotFoundException e) {
        System.err.println("MySQL JDBC Driver not found.");
        e.printStackTrace();
    } catch (SQLException e) {
        System.err.println("Failed to connect to the database.");
        e.printStackTrace();
    }

    return conn;
}
}

```

## Contact Management System code (Inserting, Deleting, Updating, Sorting, Displaying and Searching )

```
package phnbook;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import javax.swing.JOptionPane;
import java.sql.SQLException;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author KTCC
 */
public class contact extends javax.swing.JFrame {

    private static final String username = "root";
    private static final String password = "6161586";
    private static final String dataConn = "jdbc:mysql://localhost:3306/connector";

    Connection sqlConn = null;
```

```

PreparedStatement pst = null;
ResultSet rs = null;
int q, i, id, deleteItem;
    private Object stmt;

    public contact() {
        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() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        cnAdd = new javax.swing.JButton();
        cnUpdate = new javax.swing.JButton();
        cnDelete = new javax.swing.JButton();
        cnDisplay = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        cnCancel = new javax.swing.JButton();
        cnSort = new javax.swing.JButton();
    }

```

```

jPanel3 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
txtId = new javax.swing.JTextField();
txtlastName = new javax.swing.JTextField();
txtfirstName = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
txtphoneNumber = new javax.swing.JTextField();
cnReset = new javax.swing.JButton();
inputName = new javax.swing.JTextField();
cnSearch = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(153, 153, 153));
jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(51,
51, 51)));

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
jLabel1.setText("Contact management");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel1)
                .addComponent(jLabel2)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(txtId)
                .addComponent(txtlastName)
                .addComponent(txtfirstName)
                .addComponent(jLabel6)
                .addComponent(txtphoneNumber)
                .addComponent(cnReset)
                .addComponent(inputName)
                .addComponent(cnSearch)
                .addComponent(jScrollPane1)
                .addComponent(jTable1)
            )
        )
    );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jLabel4)
            .addComponent(txtId)
            .addComponent(txtlastName)
            .addComponent(txtfirstName)
            .addComponent(jLabel6)
            .addComponent(txtphoneNumber)
            .addComponent(cnReset)
            .addComponent(inputName)
            .addComponent(cnSearch)
            .addComponent(jScrollPane1)
            .addComponent(jTable1)
        )
    );

```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(jLabel1)

        .addGap(371, 371, 371))

    );

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

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(24, 24, 24)

            .addComponent(jLabel1)

            .addContainerGap(7, Short.MAX_VALUE))

    );

    jPanel2.setBackground(new java.awt.Color(204, 204, 204));

    jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0), 3));

    cnAdd.setBackground(new java.awt.Color(102, 102, 102));

    cnAdd.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

    cnAdd.setText("Add new");

    cnAdd.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBo
rder.RAISED));

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

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

            cnAddActionPerformed(evt);

        }

    });

    cnUpdate.setBackground(new java.awt.Color(102, 102, 102));

    cnUpdate.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

    cnUpdate.setText("Update");

```

```
cnUpdate.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
```

```
    cnUpdate.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            cnUpdateActionPerformed(evt);  
        }  
    });
```

```
cnDelete.setBackground(new java.awt.Color(102, 102, 102));  
cnDelete.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N  
cnDelete.setText("Delete");
```

```
cnDelete.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
```

```
    cnDelete.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            cnDeleteActionPerformed(evt);  
        }  
    });
```

```
cnDisplay.setBackground(new java.awt.Color(102, 102, 102));  
cnDisplay.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N  
cnDisplay.setText("Display");
```

```
cnDisplay.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
```

```
    cnDisplay.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            cnDisplayActionPerformed(evt);  
        }  
    });
```



```

jLabel5.setText("Sort: ASC");

cnCancel.setBackground(new java.awt.Color(102, 102, 102));
cnCancel.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
cnCancel.setText("Cancel");

cnCancel.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

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

cnSort.setBackground(new java.awt.Color(102, 102, 102));
cnSort.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
cnSort.setText("Sort");

cnSort.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

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

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel5)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cnCancel, javax.swing.GroupLayout.DEFAULT_SIZE, 100, false)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cnSort, javax.swing.GroupLayout.DEFAULT_SIZE, 100, false)
        )
);

```

```

        .addGap(17, 17, 17)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(cnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(cnCancel, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
                javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

            .addComponent(cnAdd, javax.swing.GroupLayout.DEFAULT_SIZE, 120,
                Short.MAX_VALUE)

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

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

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

            .addComponent(jLabel5))

        .addContainerGap(17, Short.MAX_VALUE))

    );

    jPanel2Layout.setVerticalGroup(

        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel2Layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(cnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(cnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(cnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE, 43,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(cnDisplay, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(jLabel5)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(cnSort, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(cnCancel, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap())

);

jPanel3.setBackground(new java.awt.Color(204, 204, 204));

jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0), 3));

jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel2.setText("Id");

jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel3.setText("First Name");

jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel4.setText("Last Name");

jLabel6.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel6.setText("Phone Number");

cnReset.setBackground(new java.awt.Color(102, 102, 102));
cnReset.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

```

```

cnReset.setText("Reset");

cnReset.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

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

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(31, 31, 31)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addComponent(jLabel6))
    .addGap(195, 195, 195)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)
    .addComponent(txtfirstName, javax.swing.GroupLayout.DEFAULT_SIZE, 365, Short.MAX_VALUE)

```

```

        .addComponent(txtlastName)
        .addComponent(txtId)
        .addComponent(txtphoneNumber))
        .addGap(75, 75, 75))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup())

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(cnReset)

        .addContainerGap())
    );

    jPanel3Layout.setVerticalGroup(
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup())
        .addContainerGap()
        .addComponent(cnReset)
        .addGap(1, 1, 1)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

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

            .addGap(39, 39, 39)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

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

            .addComponent(jLabel3))

            .addGap(36, 36, 36)

```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

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

```
    .addComponent(jLabel4))
```

```
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 37,  
Short.MAX_VALUE)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel6)
```

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

```
    .addGap(30, 30, 30))
```

```
);
```

```
cnSearch.setBackground(new java.awt.Color(153, 153, 153));
```

```
cnSearch.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
```

```
cnSearch.setText("Search");
```

```
cnSearch.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.Bevel  
Border.RAISED));
```

```
cnSearch.addActionListener(new java.awt.event.ActionListener() {
```

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

```
        cnSearchActionPerformed(evt);
```

```
    }
```

```
});
```

```
jTable1.setBackground(new java.awt.Color(204, 204, 204));
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {
```

```
        new Object [] {
```

```

    },
    new String [] {
        "Id", "First Name", "Last Name", "Phone Number"
    }
));
jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(cnSearch)
                        .addGap(18, 18, 18)
                        .addComponent(inputName, javax.swing.GroupLayout.PREFERRED_SIZE,
265, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(18, 18, 18)
                        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
            )
        )
);
layout.setVerticalGroup(

```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

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

                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(inputName,
                        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(cnSearch)))
                .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 238,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(7, Short.MAX_VALUE))
        );

    pack();

    setLocationRelativeTo(null);
} // </editor-fold>

private void cnDeleteActionPerformed(java.awt.event.ActionEvent evt) {

```



```

        int selectedRow = jTable1.getSelectedRow();
        Connection conn = null;
        if (selectedRow != -1) {
            // Get the ID of the selected row (assuming it's in the first column)
            DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
            Object id = model.getValueAt(selectedRow, 0); // Adjust the column index if needed

            try {
                // Establish database connection
                conn = connectionToDB.getConnection();

                // Prepare DELETE SQL query
                String sql = "DELETE FROM contact WHERE Id = ?";
                PreparedStatement pstmt = conn.prepareStatement(sql);
                pstmt.setObject(1, id);

                // Execute delete
                int deletedRows = pstmt.executeUpdate();

                if (deletedRows > 0) {
                    // Remove the row from the table model
                    model.removeRow(selectedRow);
                    JOptionPane.showMessageDialog(null, "Row deleted successfully.");
                } else {
                    JOptionPane.showMessageDialog(null, "Deletion failed or row not found in the
database.");
                }

                // Close the PreparedStatement
                pstmt.close();
            }
        }
    }
}

```

```

    } catch (SQLException e) {
        e.printStackTrace();

        JOptionPane.showMessageDialog(null, "An error occurred: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
    } finally {
        // Close the connection in the finally block
        try {
            if (conn != null) {
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
} else {
    JOptionPane.showMessageDialog(null, "Please select a row to delete.");
}

}

```

```

private void cnSearchActionPerformed(java.awt.event.ActionEvent evt) {
String searchQuery = inputName.getText();

```

```

    // Database connection

```

```

String url = "jdbc:mysql://localhost:3306/connector";

```

```

String user = "root";

```

```

String password = "6161586";

```

```

try (Connection conn = DriverManager.getConnection(url, user, password)) {

```

```
String sql = "SELECT * FROM contact WHERE Id LIKE ? OR firstName LIKE ?  
OR lastName LIKE ? OR phoneNumber LIKE ?";
```

```
try (PreparedStatement pstmt = conn.prepareStatement(sql)) {  
    pstmt.setString(1, "%" + searchQuery + "%");  
    pstmt.setString(2, "%" + searchQuery + "%");  
    pstmt.setString(3, "%" + searchQuery + "%");  
    pstmt.setString(4, "%" + searchQuery + "%");  
    ResultSet rs = pstmt.executeQuery();
```

```
    StringBuilder results = new StringBuilder();
```

```
    while (rs.next()) {  
        results  
            .append(", ID: ").append(rs.getString("Id"))  
            .append(", First Name: ").append(rs.getString("firstName"))  
            .append(", Last Name: ").append(rs.getString("lastName"))  
            .append(", Phone Number: ").append(rs.getString("phoneNumber"))  
            .append("\n");  
    }
```

```
    if (results.length() > 0) {  
        JOptionPane.showMessageDialog(this, results.toString());  
    } else {  
        JOptionPane.showMessageDialog(this, "No contacts found.");  
    }  
}
```

```
} catch (SQLException ex) {  
    ex.printStackTrace();  
    JOptionPane.showMessageDialog(this, "Error searching contact: " +  
ex.getMessage());  
}
```

```

    }

    private JFrame frame;

    private void cnCancelActionPerformed(java.awt.event.ActionEvent evt) {

        frame = new JFrame("Exit");

        if (JOptionPane.showConfirmDialog(frame, "Confirm if you want to exit", "Contact
Management",
            JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }
    }

    private void cnResetActionPerformed(java.awt.event.ActionEvent evt) {
        txtId.setText("");
        txtfirstName.setText("");
        txtlastName.setText("");
        txtphoneNumber.setText("");
        inputName.setText("");
    }

    private void cnAddActionPerformed(java.awt.event.ActionEvent evt) {

        String Id = txtId.getText();
        String firstName = txtfirstName.getText();
        String lastName = txtlastName.getText();
        String phoneNumber = txtphoneNumber.getText();

        // Database connection

        String url = "jdbc:mysql://localhost:3306/connector";
    }

```

```

String user = "root";
String password = "6161586";

try (java.sql.Connection conn = DriverManager.getConnection(url, user, password)) {
    String sql = "INSERT INTO contact (Id, firstName, lastName, phoneNumber)
VALUES (?, ?, ?, ?)";
    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setString(1, Id);
        pstmt.setString(2, firstName);
        pstmt.setString(3, lastName);
        pstmt.setString(4, phoneNumber);
        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this, "Contact saved successfully!");
    }
} catch (SQLException ex) {
    ex.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error saving contact: " + ex.getMessage());
}
}

```

```

private void cnDisplayActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    Connection conn = null;

```

```

    PreparedStatement pstmt = null;

```

```

    ResultSet rs = null;

```

```

    try {

```

```

        // Establish the connection

```

```

        conn = connectionToDB.getConnection();

```

```

        // SQL query to fetch job postings

```

```

String query = "SELECT * FROM contact";

// Prepare the statement
pstmt = conn.prepareStatement(query);

// Execute the query and get the result set
rs = pstmt.executeQuery();

// Get metadata for dynamically creating column names
ResultSetMetaData rsmd = rs.getMetaData();
int columnCount = rsmd.getColumnCount();

// Create an array for column names
String[] columnNames = new String[columnCount];
for (int i = 1; i <= columnCount; i++) {
    columnNames[i - 1] = rsmd.getColumnName(i);
}

// Create a DefaultTableModel to store the data
DefaultTableModel model = new DefaultTableModel(columnNames, 0);

// Set the model for the table (jTable1)
jTable1.setModel(model);

// Clear the table before adding new data
model.setRowCount(0);

// Iterate through the result set and add data to the table
while (rs.next()) {
    Object[] row = new Object[columnCount];

```

```

        for (int i = 1; i <= columnCount; i++) {
            row[i - 1] = rs.getObject(i); // Get the object from each column
        }

        // Add the row to the table model
        model.addRow(row);
    }

} catch (SQLException e) {
    // Show an error dialog if an exception occurs
    JOptionPane.showMessageDialog(this, "An error occurred: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
} finally {
    // Close the resources
    try {
        if (rs != null) rs.close();
        if (pstmt != null) pstmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

```

private void cnSortActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    Connection conn = null;

```

```

    PreparedStatement pstmt = null;

```

```

    ResultSet rs = null;

```

```

try {
    // Establish the connection
    conn = connectionToDB.getConnection();

    // SQL query to fetch job postings
    String query = "SELECT Id, firstName, lastName, phoneNumber FROM contact
ORDER BY firstName ASC";

    // Prepare the statement
    pstmt = conn.prepareStatement(query);

    // Execute the query and get the result set
    rs = pstmt.executeQuery();

    // Get metadata for dynamically creating column names
    ResultSetMetaData rsmd = rs.getMetaData();
    int columnCount = rsmd.getColumnCount();

    // Create an array for column names
    String[] columnNames = new String[columnCount];
    for (int i = 1; i <= columnCount; i++) {
        columnNames[i - 1] = rsmd.getColumnName(i);
    }

    // Create a DefaultTableModel to store the data
    DefaultTableModel model = new DefaultTableModel(columnNames, 0);

    // Set the model for the table (jTable1)
    jTable1.setModel(model);

    // Clear the table before adding new data

```



```

model.setRowCount(0);

// Iterate through the result set and add data to the table
while (rs.next()) {
    Object[] row = new Object[columnCount];

    for (int i = 1; i <= columnCount; i++) {
        row[i - 1] = rs.getObject(i); // Get the object from each column
    }

    // Add the row to the table model
    model.addRow(row);
}

} catch (SQLException e) {
    // Show an error dialog if an exception occurs
    JOptionPane.showMessageDialog(this, "An error occurred: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
} finally {
    // Close the resources
    try {
        if (rs != null) rs.close();
        if (pstmt != null) pstmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

}

```

```

private void cnUpdateActionPerformed(java.awt.event.ActionEvent evt) {

    String Id = txtId.getText();

    String firstName = txtfirstName.getText();

    String lastName = txtlastName.getText();

    String phoneNumber = txtphoneNumber.getText();


    // Database connection

    String url = "jdbc:mysql://localhost:3306/connector";

    String user = "root";

    String password = "6161586";


    try (java.sql.Connection conn = DriverManager.getConnection(url, user, password)) {

        // SQL query to update the contact

        String sql = "UPDATE contact SET  firstName = ?, lastName = ?, phoneNumber = ?
WHERE Id = ?";


        // Prepare the statement

        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {

            pstmt.setString(1, firstName); // Set the first name

            pstmt.setString(2, lastName); // Set the last name

            pstmt.setString(3, phoneNumber); // Set the phone number

            pstmt.setString(4, Id); // Set the ID for the WHERE clause


            // Execute the update

            int updatedRows = pstmt.executeUpdate();

            if (updatedRows > 0) {

                JOptionPane.showMessageDialog(this, "Contact updated successfully!");

            } else {

                JOptionPane.showMessageDialog(this, "No contact found with the given ID.");

            }

        }

    }
}

```

```

    }
} catch (SQLException ex) {
    ex.printStackTrace(); // Consider using a logging framework here
    JOptionPane.showMessageDialog(this, "Error updating contact: " + ex.getMessage());
}
}

/**
 * @param args the command line arguments
 */
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(contact.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

```

```
java.util.logging.Logger.getLogger(contact.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(contact.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(contact.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 contact().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton cnAdd;
```

```
private javax.swing.JButton cnCancel;
```

```
private javax.swing.JButton cnDelete;
```

```
private javax.swing.JButton cnDisplay;
```

```
private javax.swing.JButton cnReset;
```

```
private javax.swing.JButton cnSearch;
```

```
private javax.swing.JButton cnSort;
```

```
private javax.swing.JButton cnUpdate;
```

```
private javax.swing.JTextField inputName;
```

```
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.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
private javax.swing.JTextField txtId;  
private javax.swing.JTextField txtfirstName;  
private javax.swing.JTextField txtlastName;  
private javax.swing.JTextField txtphoneNumber;  
// End of variables declaration  
}
```