NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Topic: **Phonebook Application**

Student Name	Student Number
Simeon P. Penda	224013386

DSA Project Group: 20

Sort Module

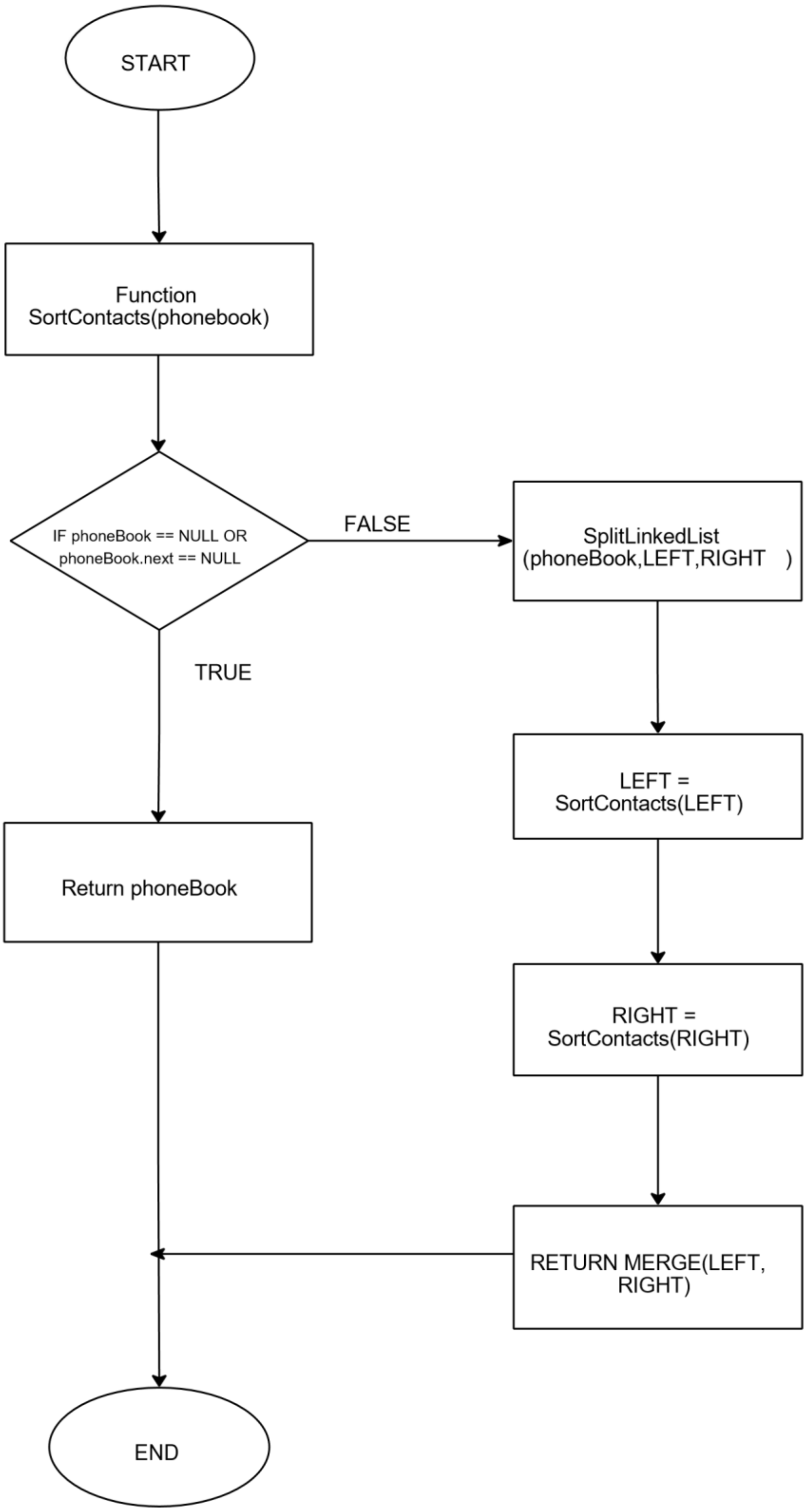
Function: SortContacts()

Purpose: sort all contacts alphabetically

Pseudocode

```
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
```

Flowchart


START
Function SortContacts(phonebook)
IF phoneBook == NULL OR phoneBook.next == NULL
FALSE
SplitLinkedList (phoneBook,LEFT,RIGHT)
TRUE
LEFT = SortContacts(LEFT)
Return phoneBook
RIGHT = SortContacts(RIGHT)
RETURN MERGE(LEFT, RIGHT)
END

Code

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

 Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

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


        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();

        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();
        }
    }

    }
```