
Project Documentation

For

PAMUNUWILA HARDWARE

Version <1.1>

Prepared by

M.R.P.N.Tharuksha Rajapaksha 200019401866 nuwangularajapaksha@gmail.com 0713848592

Instructor: Miss. Narmada Harischandra

Course: PROFESSIONAL DIPLOMA IN SOFTWARE ENGINEERING

Batch: Griffins

Teaching Assistant: Miss. Narmada Harischandra

Date: 17th July 2021

Batch Name : Griffins

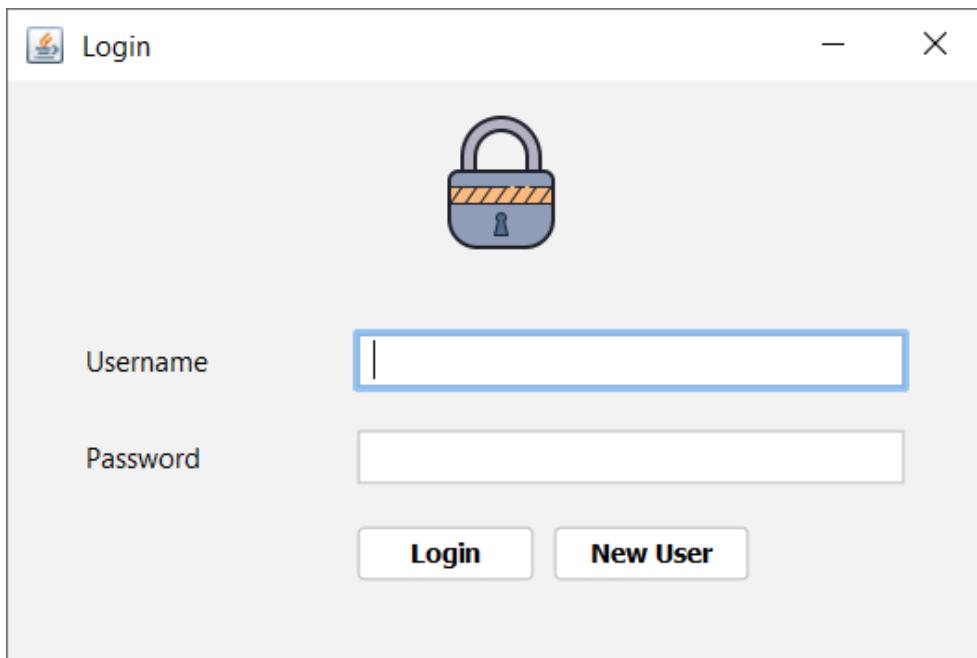
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Login Form

GUI Number : 01

Use case Name: User Management



01. Brief Description of the GUI:

The main objective of this interface is to check the username and password and give the user the main access to the system. It also provides access to register a first user (Admin) when logging in to the system for the first time.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|----------------|---|--|
| JPanel | To adjust alignments | Using JPanels makes it easy to adjust the alignments of components. Therefore all components are installed on JPanels. |
| JLabel | Display login icon & text | To add images and icons to make it easier for the user to identify this interface uniquely, to identify text fields, and to display text. |
| JTextField | Insert username | To enter the username |
| JPasswordField | Insert password | To enter the password |
| JButtons | Login to system and get access to create first user | After checking the username and password, the buttons are required to give the user access to the system and to give access to the system to register a new user when logging in to the system for the first time. These JButtons meet that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

LoginForm.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import model.LoginFormModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
 */
```

```
public class LoginForm extends javax.swing.JFrame {
```

```
/**  
 * Creates new form LoginForm  
 */  
  
private LoginFormModel loginFormModel;  
  
static Logger logger = Logger.getLogger(LoginForm.class.getName());  
  
  
int userNo;  
  
String usernameChecked;  
  
int jobroleNo;  
  
String jobroleName;  
  
  
public LoginForm() {  
    initComponents();  
  
    logger.debug("Access to LoginForm.java");  
  
  
    loginFormModel = new LoginFormModel();  
  
  
    this.addWindowListener(new WindowAdapter() {  
        @Override  
        public void windowClosing(WindowEvent e) {  
            Exit();  
        }  
  
    });  
  
  
    boolean result = loginFormModel.checkUsers();  
  
    if(result){  
        newUserButton.setEnabled(true);  
    }  
}
```

```
}
```

```
void Exit() {  
    int a = JOptionPane.showConfirmDialog(rootPane, "Do you Want to exit ?", "Message",  
    JOptionPane.YES_NO_OPTION);
```

```
    if (a == JOptionPane.YES_OPTION) {  
        logger.debug("Exit Program [END]");  
        System.exit(0);
```

```
}
```

```
}
```

```
/**
```

```
* 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();
```

```
    usernameField = new javax.swing.JTextField();
```

```
    passwordField = new javax.swing.JPasswordField();
```

```
    jLabel2 = new javax.swing.JLabel();
```

```
    jLabel3 = new javax.swing.JLabel();
```

```
    loginButton = new javax.swing.JButton();
```

```
    newUserButton = new javax.swing.JButton();
```

```
    messageLabel = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.DO NOTHING ON CLOSE);
setTitle("Login");
setResizable(false);

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-lock-100.png"))); // NOI18N

jLabel2.setText("Username");

jLabel3.setText("Password");

loginButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
loginButton.setText("Login");
loginButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginButtonActionPerformed(evt);
    }
});
```



```
newUserButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newUserButton.setText("New User");
newUserButton.setEnabled(false);
newUserButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newUserButtonActionPerformed(evt);
    }
});
```



```
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 126, Short.MAX_VALUE)
            .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18, Short.MAX_VALUE)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(usernameField)
                .addComponent(passwordField, javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(loginButton, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(newUserButton, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
                    javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap())
    );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE)
                .addComponent(usernameField, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE))
            .addGap(18, 18, 18)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(passwordField, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)
                .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE)))
        )
);
```

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(loginButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(newUserButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

messageLabel.setForeground(new java.awt.Color(255, 0, 0));
messageLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

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.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(216, 216, 216)
                .addComponent(jLabel1))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(36, 36, 36)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(messageLabel, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        .addGap(36, Short.MAX_VALUE))
    .addContainerGap(36, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
```

```

.addContainerGap()

.addComponent(jLabel1)

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

.addComponent(messageLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 16,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(0, 0, 0)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(35, Short.MAX_VALUE))

);

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)

);

layout.setVerticalGroup(

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

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

);

pack();

 setLocationRelativeTo(null);

}// </editor-fold>

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

// TODO add your handling code here:

String username = usernameField.getText();

String password = passwordField.getText();

```

```
if (username.equals("")) {  
    // JOptionPane.showConfirmDialog(rootPane, "Please Enter The Userame !", "Error", -1,  
    JOptionPane.ERROR_MESSAGE);  
  
    messageLabel.setText("Please Enter The Userame !");  
  
    logger.error("Error: Username not enterd");  
  
    usernameField.requestFocus(true);  
  
} else if (password.equals("")) {  
  
    // JOptionPane.showConfirmDialog(rootPane, "Please Enter The Password !", "Error", -1,  
    JOptionPane.ERROR_MESSAGE);  
  
    messageLabel.setText("Please Enter The Password !");  
  
    logger.error("Error: Password not enterd");  
  
    passwordField.requestFocus(true);  
  
} else {  
  
    Object[] userData = loginFormModel.getUserData(username, password);  
  
  
    int loginStatus = (int) userData[0];  
  
    userNo = (int) userData[1];  
  
    usernameChecked = (String) userData[2];  
  
    jobroleNo = (int) userData[3];  
  
    jobroleName = (String) userData[4];  
  
  
    if (loginStatus == 0) {  
  
        // JOptionPane.showConfirmDialog(rootPane, "Invalied Username !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        messageLabel.setText("Invalied Username !");  
  
        logger.error("Error: Invalied Username !");  
  
    } else if (loginStatus == 1) {  
  
        // JOptionPane.showConfirmDialog(rootPane, "Login Success", "Message", -1, 1);  
  
        new HomePage(userNo, usernameChecked, jobroleNo, jobroleName).setVisible(true);  
  
        logger.debug("Access to HomePage.java");  
  
        this.dispose();  
  
        logger.debug("LoginForm.java dispose");  
    }  
}
```

```
        } else if (loginStatus == 2) {  
  
            // JOptionPane.showConfirmDialog(rootPane, "Incorrect Username or Password !", "Error", -1,  
            JOptionPane.ERROR_MESSAGE);  
  
            messageLabel.setText("Incorrect Username or Password !");  
  
            logger.error("Error: Incorrect Username or Password !");  
  
        }  
  
    }  
  
}
```

```
private void newUserButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new FirstUserFormDialog(this, rootPaneCheckingEnabled).setVisible(true);  
  
    logger.debug("Access to FirstUserFormDialog.java");  
  
}
```

```
/**  
 * @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 {  
  
        UIManager.setLookAndFeel(new FlatIntelliJLaf());  
  
        UIManager.put("Table.showHorizontalLines", true);  
  
        UIManager.put("Table.showVerticalLines", true);  
  
        UIManager.put("ScrollBar.showButtons", true);  
  
    }
```

```

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//
//         if ("Windows".equals(info.getName())) {
//
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//             break;
//         }
//     }
// }

// } catch (ClassNotFoundException ex) {
//
//     java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (InstantiationException ex) {
//
//     java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (IllegalAccessException ex) {
//
//     java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//     java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// }

// }

//</editor-fold>

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new LoginForm().setVisible(true);
        logger.debug("Run PMUNUWILA HARDWARE Software Application");
    }
});

}

```

```
// Variables declaration - do not modify

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JButton loginButton;
private javax.swing.JLabel messageLabel;
private javax.swing.JButton newUserButton;
private javax.swing.JPasswordField passwordField;
private javax.swing.JTextField usernameField;

// End of variables declaration

}
```

LoginFormModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import org.apache.log4j.Logger;

/**
 *
```

```
* @author EZIO AUDITORY

*/
public class LoginFormModel {

    private Connection dbConnection;

    static Logger logger = Logger.getLogger(LoginFormModel.class.getName());

    public LoginFormModel() {
        dbConnection = database.dbConnection.getConnection();
    }

    public Object[] getUserData(String username, String password) {
        String query = "SELECT * FROM user INNER JOIN jobrole ON user.jobrole_no = jobrole.jobrole_no WHERE user_username = ? AND user_status = ?";

        logger.debug("SQL query String: " + query);

        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            psm.setString(1, username);
            psm.setInt(2, 1);
            ResultSet rs = psm.executeQuery();
            logger.debug("Query execution done");

            Object[] rowData = null;
            int loggingStatus = 0;
            int userNo = 0;
            String usernameChecked = null;
            String passwordChecked = null;
            int jobroleNo = 0;
            String jobroleName = null;
        }
    }
}
```

```

while (rs.next()) {

    userNo = rs.getInt("user_no");

    usernameChecked = rs.getString("user_username");

    passwordChecked = rs.getString("user_password");

    jobroleNo = rs.getInt("user.jobrole_no");

    jobroleName = rs.getString("jobrole_name");




    logger.debug("Check validation of username & password");

    if (usernameChecked.equals(username) && passwordChecked.equals(password)) {

        logingStatus = 1;

    } else {

        logingStatus = 2;

    }

    logger.debug("Set login status");

}

rowData = new Object[]{logingStatus, userNo, usernameChecked, jobroleNo, jobroleName};

logger.debug("return rowData");

return rowData;

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}

public boolean checkUsers() {

String query = "SELECT count(user_no) FROM pamunuwila.hardware_db.user";

logger.debug("SQL query String: " + query);




boolean giveAccess = false;

```

```
try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    while (rs.next()) {

        int userCount = rs.getInt("count(user_no)");

        if (userCount == 0) {

            giveAccess = true;

            logger.debug("give access");

        }

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return give access");

return giveAccess;

}

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: First User Form

GUI Number : 02

Use case Name: User Management

03. Brief Description of the GUI:

The main objective of this interface is to give the first-time user of this system the opportunity to register as a first user (Admin).

04. Justification of GUI component usage:

| Component | Usage | Justification |
|----------------|--|---|
| JPanel | To adjust alignments & give a title to the component group | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert username & other user registration details | To enter the username & other user registration details |
| JPasswordField | Insert password | To enter the password |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register first user (Admin) & clear text fields | Buttons are required to check the validity of text fields, register the details of a new user and clear the values in the text fields. These JButtons meet that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



FirstUserFormDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import model.ComboBoxLoadModel;
import model.UserModel;
import org.apache.log4j.Logger;
```

```
/**
```

```
*  
* @author EZIO AUDITORY  
*/  
  
public class FirstUserFormDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form FirstUserFormDialog  
     */  
  
    private UserModel userModel;  
    private ComboBoxLoadModel comboBoxLoadModel;  
    static Logger logger = Logger.getLogger(FirstUserFormDialog.class.getName());  
  
    DefaultComboBoxModel districtComboBoxModel;  
    DefaultComboBoxModel provinceComboBoxModel;  
    DefaultComboBoxModel jobroleComboBoxModel;  
  
    public FirstUserFormDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to FirstUserFormDialog.java");  
  
        userModel = new UserModel();  
        comboBoxLoadModel = new ComboBoxLoadModel();  
  
        Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();  
        districtComboBoxModel = new DefaultComboBoxModel(vDistrict);  
        districtComboBox.setModel(districtComboBoxModel);  
        districtComboBox.setSelectedIndex(0);  
  
        Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();  
        provinceComboBoxModel = new DefaultComboBoxModel(vProvince);
```

```
provinceComboBox.setModel(provinceComboBoxModel);
provinceComboBox.setSelectedIndex(0);

Vector vCompany = comboBoxLoadModel.jobroleComboBoxLoad();
jobroleComboBoxModel = new DefaultComboBoxModel(vCompany);
jobroleComboBox.setModel(jobroleComboBoxModel);
jobroleComboBox.setSelectedIndex(0);

}

/***
 * 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() {

    jPanel2 = new javax.swing.JPanel();
    userNoField = new javax.swing.JTextField();
    nameField = new javax.swing.JTextField();
    nicField = new javax.swing.JTextField();
    emailField = new javax.swing.JTextField();
    contactNoField = new javax.swing.JTextField();
    addressField = new javax.swing.JTextField();
    cityField = new javax.swing.JTextField();
    districtComboBox = new javax.swing.JComboBox<>();
    provinceComboBox = new javax.swing.JComboBox<>();
    jobroleComboBox = new javax.swing.JComboBox<>();
    jLabel1 = new javax.swing.JLabel();
    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();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
registerButton = new javax.swing.JButton();
newButton = new javax.swing.JButton();
jSeparator3 = new javax.swing.JSeparator();
jLabel11 = new javax.swing.JLabel();
usernameField = new javax.swing.JTextField();
jLabel12 = new javax.swing.JLabel();
PasswordField = new javax.swing.JPasswordField();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Create Admin");
setResizable(false);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13)))); // NOI18N

userNoField.setEditable(false);

contactNoField.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        contactNoFieldKeyPressed(evt);
    }
})
```

```
});

jobroleComboBox.setEnabled(false);

jLabel1.setText("User No");

jLabel2.setText("Name");

jLabel3.setText("NIC");

jLabel4.setText("Email");

jLabel5.setText("Contact No");

jLabel6.setText("Address");

jLabel7.setText("City");

jLabel8.setText("District");

jLabel9.setText("Province");

jLabel10.setText("Job Role");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

registerButton.setText("Register");

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

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

        registerButtonActionPerformed(evt);

    }

});
```

```
newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
newButton.setText("New");  
newButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        newButtonActionPerformed(evt);  
    }  
});  
  
jLabel11.setText("Username");  
  
jLabel12.setText("Password");  
  
PasswordField.setText("1234");  
  
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()  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jSeparator3)  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addComponent(jPanel2Layout.createSequentialGroup()  
                    .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                        javax.swing.GroupLayout.PREFERRED_SIZE)  
                    .addGap(18, 18, 18)  
                    .addComponent(usernameField))  
                .addGroup(jPanel2Layout.createSequentialGroup()  
                    .addComponent(jPanel2Layout.createSequentialGroup()  
                        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                            javax.swing.GroupLayout.PREFERRED_SIZE)  
                        .addGap(18, 18, 18)
```

```
.addComponent(PasswordField))

.addComponent(jSeparator1)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jobroleComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addGroup(jPanel2Layout.createSequentialGroup()

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

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

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

.addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)
```

```
.addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(nicField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(userNoField))

.addComponent(newButton, javax.swing.GroupLayout.DEFAULT_SIZE, 399, Short.MAX_VALUE)

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

.addGap(0, 0, Short.MAX_VALUE))

.addContainerGap()

);

jPanel2Layout.setVerticalGroup()

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

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(30, 30, 30)

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

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(userNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nameField)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nicField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(emailField)
    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(contactNoField)
        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(addressField)
        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(cityField)
        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(districtComboBox)
        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(provinceComboBox)
        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false))
```

```
.addComponent(usernameField)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(jLabel12, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

.addComponent(PasswordField))

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

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

.addComponent(jobroleComboBox)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(21, 21, 21))

);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGap(0, 457, Short.MAX_VALUE)

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()
```

```

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 697, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addContainerGap()

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

.addContainerGap()))

);

pack();

 setLocationRelativeTo(null);

}// </editor-fold>

```

```

private void contactNoFieldKeyPressed(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String contactNo = contactNoField.getText();

int length = contactNo.length();

char c = evt.getKeyChar();

if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {

if (length < 10) {

contactNoField.setEditable(true);

} else {

contactNoField.setEditable(false);

}

} else {

if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==

```

```
KeyEvent.VK_DELETE) {  
  
    contactNoField.setEditable(true);  
  
} else {  
  
    contactNoField.setEditable(false);  
  
}  
  
}  
  
}  
  
  
private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    String name = nameField.getText();  
  
    String nic = nicField.getText();  
  
    String email = emailField.getText();  
  
    String contactNo = contactNoField.getText();  
  
    String address = addressField.getText();  
  
    String city = cityField.getText();  
  
    int distictNo = districtComboBox.getSelectedIndex() + 1;  
  
    int provinceNo = provinceComboBox.getSelectedIndex() + 1;  
  
    String username = usernameField.getText();  
  
    String password = PasswordField.getText();  
  
    int jobroleNo = jobroleComboBox.getSelectedIndex() + 1;  
  
  
    logger.debug("Check validations");  
  
    if (name.equals("")) {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        nameField.requestFocus(true);  
  
    } else if (contactNo.equals("")) {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        contactNoField.requestFocus(true);  
  
    } else if (city.equals("")) {  
  
    }
```

```

JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

cityField.requestFocus(true);

} else if (username.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Username !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    usernameField.requestFocus(true);

} else if (password.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Password !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    usernameField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}[@][a-zA-Z]{0,10}{.}[a-zA-Z]{0,5}$";

    Pattern p = Pattern.compile(PATTERN);

    Matcher m = p.matcher(email);

    if (!m.matches()) {

        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = userModel.registerUser(name, nic, email, contactNo, address, city, distictNo, provinceNo,
username, password, jobroleNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            logger.debug("Message: " + result);

        }else{

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

    }

} else {

    String result = userModel.registerUser(name, nic, email, contactNo, address, city, distictNo, provinceNo,
username, password, jobroleNo);

    if (result.equals("Success")) {


```

```
JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

logger.debug("Message: " + result);

this.dispose();

logger.debug("FirstUserFormDialog.java Dispose");

}else{

JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

}
```

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

// TODO add your handling code here:

clearFields();

}
```

```
private void clearFields() {

userNoField.setText("");

nameField.setText("");

nicField.setText("");

emailField.setText("");

contactNoField.setText("");

addressField.setText("");

cityField.setText("");

districtComboBox.setSelectedIndex(0);

provinceComboBox.setSelectedIndex(0);

usernameField.setText("");

PasswordField.setText("");

jobroleComboBox.setSelectedIndex(0);

registerButton.setEnabled(true);
```

```
logger.debug("clearFields");

}

/** 
 * @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 {
    UIManager.setLookAndFeel(new FlatIntelliJLaf());
    UIManager.put("Table.showHorizontalLines", true);
    UIManager.put("Table.showVerticalLines", true);
    UIManager.put("ScrollBar.showButtons", true);
    UIManager.put("ScrollBar.width", 16);
} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(FirstUserFormDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
```

```
//    } catch (InstantiationException ex) {
//
//        java.util.logging.Logger.getLogger(FirstUserFormDialog.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    } catch (IllegalAccessException ex) {
//
//        java.util.logging.Logger.getLogger(FirstUserFormDialog.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//        java.util.logging.Logger.getLogger(FirstUserFormDialog.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    }
//
//</editor-fold>

/*
 * Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        FirstUserFormDialog dialog = new FirstUserFormDialog(new javax.swing.JFrame(), true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {

                System.exit(0);

            }

        });

        dialog.setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JPasswordField PasswordField;
private javax.swing.JTextField addressField;
private javax.swing.JTextField cityField;
private javax.swing.JTextField contactNoField;
```

```
private javax.swing.JComboBox<String> districtComboBox;
private javax.swing.JTextField emailField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
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 jPanel2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JComboBox<String> jobroleComboBox;
private javax.swing.JTextField nameField;
private javax.swing.JButton newButton;
private javax.swing.JTextField nicField;
private javax.swing.JComboBox<String> provinceComboBox;
private javax.swing.JButton registerButton;
private javax.swing.JTextField userNoField;
private javax.swing.JTextField usernameField;
// End of variables declaration
}
```

Batch Name : Griffins

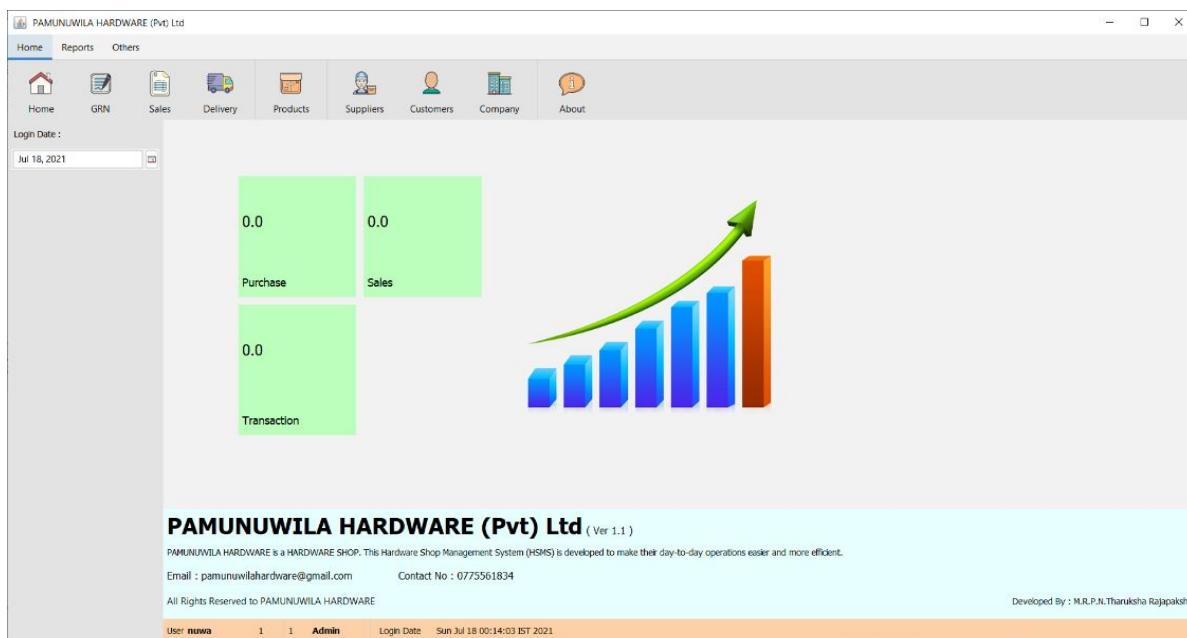
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Home Page

GUI Number : 03

Use case Name: User Management



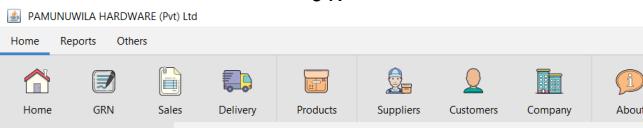
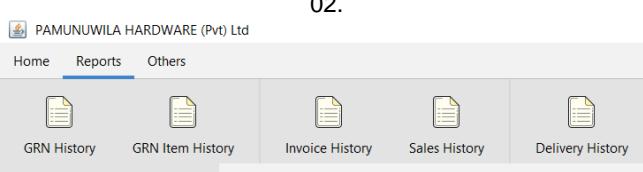
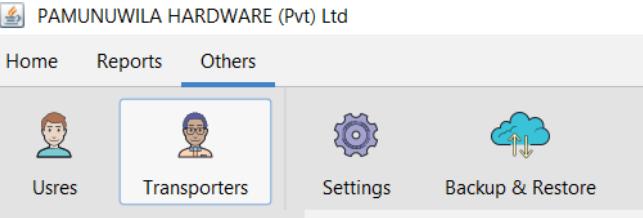
01. Brief Description of the GUI:

The First Interface that appears to anyone who logs in to this Software Application with a username and password. The main objective of this interface is to give access to other interfaces. Also displays login details, purchase details, sales details, and transaction details.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|--------------|--|---|
| JTabbedPane | Display custom menu bar | Need to make connections to other interfaces through the home page. So a unique menu bar is suitable for this system. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments and create design | Using JPanel makes it easy to adjust the alignments of components. It also has the ability to create different designs. Therefore all components are installed on JPanel. |
| JLabel | Display icons & text | Use images and icons to embellish the interface and display text. |
| JDateChooser | Display date | To display the current date |
| JButtons | Access to Other Interfaces | To connect with other interfaces on the system, you need buttons to access other interfaces. These JButtons meet that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---|---|
| Menu Creation | <p>01.</p>  | <p>To provide access to other interfaces on the system.</p> |
| | <p>02.</p>  | <p>To provide access to other interfaces on the system.</p> |
| | <p>03.</p>  | <p>To provide access to other interfaces on the system.</p> |
| Dialog Creation | <p>01.</p> | |
| | <p>02.</p> | |
| | <p>03.</p> | |
| Popup menu Creation | <p>01.</p> | |
| | <p>02.</p> | |
| | <p>03.</p> | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

HomePage.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import model.HomeModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
 */
```

```
public class HomePage extends javax.swing.JFrame {  
  
    /**  
     * Creates new form HomePage  
     */  
  
    private HomeModel homeModel;  
  
    static Logger logger = Logger.getLogger(HomePage.class.getName());  
  
  
    int userNoThis;  
  
    String usernameCheckedThis;  
  
    int jobroleNoThis;  
  
    String jobroleNameThis;  
  
  
    public HomePage(int userNo, String usernameChecked, int jobroleNo, String jobroleName) {  
        initComponents();  
  
        logger.debug("Access to HomePage.java");  
  
        setExtendedState(MAXIMIZED_BOTH);  
  
        homeModel = new HomeModel();  
  
  
        userNoThis = userNo;  
  
        usernameCheckedThis = usernameChecked;  
  
        jobroleNoThis = jobroleNo;  
  
        jobroleNameThis = jobroleName;  
  
  
        if (jobroleNoThis == 1 && jobroleNameThis.equals("Admin")) {  
            homeButton.setEnabled(true);  
  
            grnButton.setEnabled(true);  
  
            salesButton.setEnabled(true);  
  
            deliveryButton.setEnabled(true);  
  
            productButton.setEnabled(true);  
  
            supplierButton.setEnabled(true);  
        }  
    }  
}
```

```
customerButton.setEnabled(true);

companyButton.setEnabled(true);

aboutButton.setEnabled(true);

grnHistoryButton.setEnabled(true);

grnlItemHistoryButton.setEnabled(true);

invoiceHistoryButton.setEnabled(true);

salesHistoyButton.setEnabled(true);

deliveryHistoryButton.setEnabled(true);

userButton.setEnabled(true);

transporterButton.setEnabled(true);

settingsButton.setEnabled(true);

backupRestoreButton.setEnabled(true);

} else if (jobroleNoThis == 2 && jobroleNameThis.equals("Accountant")) {

homeButton.setEnabled(true);

grnButton.setEnabled(false);

salesButton.setEnabled(false);

deliveryButton.setEnabled(false);

productButton.setEnabled(false);

supplierButton.setEnabled(false);

customerButton.setEnabled(false);

companyButton.setEnabled(false);

aboutButton.setEnabled(true);

grnHistoryButton.setEnabled(true);

grnlItemHistoryButton.setEnabled(true);

invoiceHistoryButton.setEnabled(true);

salesHistoyButton.setEnabled(true);

deliveryHistoryButton.setEnabled(true);

userButton.setEnabled(false);

transporterButton.setEnabled(false);

settingsButton.setEnabled(false);

backupRestoreButton.setEnabled(false);
```

```
 } else if (jobroleNoThis == 3 && jobroleNameThis.equals("Cashier")) {  
    homeButton.setEnabled(true);  
    grnButton.setEnabled(true);  
    salesButton.setEnabled(true);  
    deliveryButton.setEnabled(true);  
    productButton.setEnabled(true);  
    supplierButton.setEnabled(true);  
    customerButton.setEnabled(true);  
    companyButton.setEnabled(true);  
    aboutButton.setEnabled(true);  
    grnHistoryButton.setEnabled(true);  
    grnlItemHistoryButton.setEnabled(true);  
    invoiceHistoryButton.setEnabled(true);  
    salesHistoyButton.setEnabled(true);  
    deliveryHistoryButton.setEnabled(true);  
    userButton.setEnabled(false);  
    transporterButton.setEnabled(true);  
    settingsButton.setEnabled(false);  
    backupRestoreButton.setEnabled(false);  
 } else if (jobroleNoThis == 4 && jobroleNameThis.equals("Employee")) {  
    homeButton.setEnabled(true);  
    grnButton.setEnabled(false);  
    salesButton.setEnabled(false);  
    deliveryButton.setEnabled(false);  
    productButton.setEnabled(true);  
    supplierButton.setEnabled(true);  
    customerButton.setEnabled(true);  
    companyButton.setEnabled(true);  
    aboutButton.setEnabled(true);  
    grnHistoryButton.setEnabled(false);  
    grnlItemHistoryButton.setEnabled(false);
```

```
invoiceHistoryButton.setEnabled(false);

salesHistoyButton.setEnabled(false);

deliveryHistoryButton.setEnabled(false);

userButton.setEnabled(false);

transporterButton.setEnabled(true);

settingsButton.setEnabled(false);

backupRestoreButton.setEnabled(false);

} else if (jobroleNoThis == 5 && jobroleNameThis.equals("Manager")) {

homeButton.setEnabled(true);

grnButton.setEnabled(true);

salesButton.setEnabled(true);

deliveryButton.setEnabled(true);

productButton.setEnabled(true);

supplierButton.setEnabled(true);

customerButton.setEnabled(true);

companyButton.setEnabled(true);

aboutButton.setEnabled(true);

grnHistoryButton.setEnabled(true);

grnlItemHistoryButton.setEnabled(true);

invoiceHistoryButton.setEnabled(true);

salesHistoyButton.setEnabled(true);

deliveryHistoryButton.setEnabled(true);

userButton.setEnabled(false);

transporterButton.setEnabled(true);

settingsButton.setEnabled(true);

backupRestoreButton.setEnabled(true);

} else {

homeButton.setEnabled(false);

grnButton.setEnabled(false);

salesButton.setEnabled(false);

deliveryButton.setEnabled(false);
```

```
productButton.setEnabled(false);
supplierButton.setEnabled(false);
customerButton.setEnabled(false);
companyButton.setEnabled(false);
aboutButton.setEnabled(false);
grnHistoryButton.setEnabled(false);
grnlItemHistoryButton.setEnabled(false);
invoiceHistoryButton.setEnabled(false);
salesHistoyButton.setEnabled(false);
deliveryHistoryButton.setEnabled(false);
userButton.setEnabled(false);
transporterButton.setEnabled(false);
settingsButton.setEnabled(false);
backupRestoreButton.setEnabled(false);

}

logger.debug("Give User Access");

userNoLabel.setText(String.valueOf(userNoThis));
usernameLabel.setText(usernameCheckedThis);
jobroleNoLabel.setText(String.valueOf(jobroleNoThis));
jobroleNameLabel.setText(jobroleNameThis);
datetimeLabel.setText(new Date().toString());
jDateChooser1.setDate(new Date());

Object[] purchaseAmount = homeModel.getPurchaseAmount();
purchaseAmountLabel.setText(new String(purchaseAmount[0] + ""));
logger.debug("Set Purchase Amount");

Object[] salesAmount = homeModel.getSalesAmount();
salesAmountLabel.setText(new String(salesAmount[0] + ""));
logger.debug("Set Sales Amount");
```

```
double transaction = (double) salesAmount[0] - (double) purchaseAmount[0];
transactionLabel.setText(String.valueOf(transaction));
logger.debug("Set Transaction");

this.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        Exit();
    }
});

private HomePage() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods,
choose Tools | Templates.
}

void Exit() {
    int a = JOptionPane.showConfirmDialog(rootPane, "Do you Want to exit?", "Message",
JOptionPane.YES_NO_OPTION);
    if (a == JOptionPane.YES_OPTION) {
        logger.debug("Exit Program [END]");
        System.exit(0);
    }
}

/**
 * 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() {  
  
    jTabbedPane1 = new javax.swing.JTabbedPane();  
  
    jPanel1 = new javax.swing.JPanel();  
  
    homeButton = new javax.swing.JButton();  
  
    grnButton = new javax.swing.JButton();  
  
    salesButton = new javax.swing.JButton();  
  
    deliveryButton = new javax.swing.JButton();  
  
    productButton = new javax.swing.JButton();  
  
    supplierButton = new javax.swing.JButton();  
  
    customerButton = new javax.swing.JButton();  
  
    companyButton = new javax.swing.JButton();  
  
    aboutButton = new javax.swing.JButton();  
  
    jSeparator1 = new javax.swing.JSeparator();  
  
    jSeparator2 = new javax.swing.JSeparator();  
  
    jSeparator3 = new javax.swing.JSeparator();  
  
    jPanel2 = new javax.swing.JPanel();  
  
    grnHistoryButton = new javax.swing.JButton();  
  
    grnItemHistoryButton = new javax.swing.JButton();  
  
    invoiceHistoryButton = new javax.swing.JButton();  
  
    salesHistoyButton = new javax.swing.JButton();  
  
    deliveryHistoryButton = new javax.swing.JButton();  
  
    jSeparator4 = new javax.swing.JSeparator();  
  
    jSeparator5 = new javax.swing.JSeparator();  
  
    jPanel3 = new javax.swing.JPanel();  
  
    userButton = new javax.swing.JButton();  
  
    transporterButton = new javax.swing.JButton();  
  
    settingsButton = new javax.swing.JButton();
```

```
jSeparator6 = new javax.swing.JSeparator();

backupRestoreButton = new javax.swing.JButton();

jPanel4 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jDateChooser1 = new com.toedter.calendar.JDateChooser();

jPanel5 = new javax.swing.JPanel();

jPanel7 = new javax.swing.JPanel();

purchaseAmountLabel = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jPanel9 = new javax.swing.JPanel();

transactionLabel = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jPanel10 = new javax.swing.JPanel();

salesAmountLabel = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

jPanel6 = 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();

jPanel8 = new javax.swing.JPanel();

jLabel7 = new javax.swing.JLabel();

usernameLabel = new javax.swing.JLabel();

userNoLabel = new javax.swing.JLabel();

jSeparator7 = new javax.swing.JSeparator();

jobroleNoLabel = new javax.swing.JLabel();

jobroleNameLabel = new javax.swing.JLabel();

jSeparator8 = new javax.swing.JSeparator();

jLabel8 = new javax.swing.JLabel();
```

```
datetimeLabel = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLOSE);

setTitle("PAMUNUWILA HARDWARE (Pvt) Ltd");

jPanel1.setBackground(new java.awt.Color(227, 227, 227));

homeButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-home-page-50.png"))); // NOI18N

homeButton.setText("Home");

homeButton.setContentAreaFilled(false);

homeButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

homeButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

homeButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        homeButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        homeButtonMouseExited(evt);

    }

});

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

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

        homeButtonActionPerformed(evt);

    }

});

grnButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-goodnotes-50.png"))); // NOI18N

grnButton.setText("GRN");
```

```
grnButton.setContentAreaFilled(false);

grnButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

grnButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

grnButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        grnButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        grnButtonMouseExited(evt);

    }

});;

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

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

        grnButtonActionPerformed(evt);

    }

});;

salesButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-invoice-50.png"))); // NOI18N

salesButton.setText("Sales");

salesButton.setContentAreaFilled(false);

salesButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

salesButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

salesButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        salesButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        salesButtonMouseExited(evt);

    }

});;
```

```
salesButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        salesButtonActionPerformed(evt);  
    }  
});  
  
deliveryButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-in-transit-50.png"))); // NOI18N  
deliveryButton.setText("Delivery");  
deliveryButton.setContentAreaFilled(false);  
deliveryButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);  
deliveryButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);  
deliveryButton.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseEntered(java.awt.event.MouseEvent evt) {  
        deliveryButtonMouseEntered(evt);  
    }  
    public void mouseExited(java.awt.event.MouseEvent evt) {  
        deliveryButtonMouseExited(evt);  
    }  
});  
deliveryButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        deliveryButtonActionPerformed(evt);  
    }  
});  
  
productButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-product-50.png"))); // NOI18N  
productButton.setText("Products");  
productButton.setContentAreaFilled(false);  
productButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);  
productButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
```

```
productButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        productButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        productButtonMouseExited(evt);
    }
});

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

supplierButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-supplier-50.png")));
// NOI18N
supplierButton.setText("Suppliers");
supplierButton.setContentAreaFilled(false);
supplierButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
supplierButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
supplierButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        supplierButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        supplierButtonMouseExited(evt);
    }
});
supplierButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        supplierButtonActionPerformed(evt);
    }
});
```

```
}

});

customerButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-customer-50.png"))); // NOI18N

customerButton.setText("Customers");

customerButton.setContentAreaFilled(false);

customerButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

customerButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

customerButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        customerButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        customerButtonMouseExited(evt);

    }

});

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

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

        customerButtonActionPerformed(evt);

    }

});

companyButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-company-50.png"))); // NOI18N

companyButton.setText("Company");

companyButton.setContentAreaFilled(false);

companyButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

companyButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

companyButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        companyButtonMouseEntered(evt);

    }

})
```

```
}

public void mouseExited(java.awt.event.MouseEvent evt) {
    companyButtonMouseExited(evt);
}

});

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

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

});

aboutButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-about-50.png"))); // NOI18N

aboutButton.setText("About");
aboutButton.setContentAreaFilled(false);
aboutButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
aboutButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
aboutButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {
        aboutButtonMouseEntered(evt);
    }

    public void mouseExited(java.awt.event.MouseEvent evt) {
        aboutButtonMouseExited(evt);
    }

});

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

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

});
```

```
jSeparator1.setOrientation(javax.swing.SwingConstants.VERTICAL);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jSeparator3.setOrientation(javax.swing.SwingConstants.VERTICAL);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(homeButton)
        .addGap(0, 0, 0)
        .addComponent(grnButton)
        .addGap(0, 0, 0)
        .addComponent(salesButton)
        .addGap(0, 0, 0)
        .addComponent(deliveryButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(productButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(supplierButton)
        .addGap(0, 0, 0)
        .addComponent(customerButton)
        .addGap(0, 0, 0)
    )
);
```

```
.addComponent(companyButton)

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(aboutButton)

.addContainerGap(1094, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup()

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

.addComponent(jSeparator1)

.addComponent(jSeparator2)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(grnButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

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

.addComponent(salesButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(deliveryButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(productButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(supplierButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(customerButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(companyButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)

.addComponent(aboutButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE))

.addContainerGap()

.addComponent(jSeparator3)

);

jTabbedPane1.addTab("Home", jPanel1);
```

```
jPanel2.setBackground(new java.awt.Color(227, 227, 227));  
  
grnHistoryButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-list-view-50.png"))); // NOI18N  
  
grnHistoryButton.setText("GRN History");  
  
grnHistoryButton.setContentAreaFilled(false);  
  
grnHistoryButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);  
  
grnHistoryButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);  
  
grnHistoryButton.addMouseListener(new java.awt.event.MouseAdapter() {  
  
    public void mouseEntered(java.awt.event.MouseEvent evt) {  
  
        grnHistoryButtonMouseEntered(evt);  
  
    }  
  
    public void mouseExited(java.awt.event.MouseEvent evt) {  
  
        grnHistoryButtonMouseExited(evt);  
  
    }  
  
});  
  
grnHistoryButton.addActionListener(new java.awt.event.ActionListener() {  
  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
        grnHistoryButtonActionPerformed(evt);  
  
    }  
  
});  
  
grnItemHistoryButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-list-view-50.png"))); // NOI18N  
  
grnItemHistoryButton.setText("GRN Item History");  
  
grnItemHistoryButton.setContentAreaFilled(false);  
  
grnItemHistoryButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);  
  
grnItemHistoryButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);  
  
grnItemHistoryButton.addMouseListener(new java.awt.event.MouseAdapter() {  
  
    public void mouseEntered(java.awt.event.MouseEvent evt) {  
  
        grnItemHistoryButtonMouseEntered(evt);  
  
    }  
  
});
```

```
}

public void mouseExited(java.awt.event.MouseEvent evt) {
    grnItemHistoryButtonMouseExited(evt);
}

});

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

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

});

invoiceHistoryButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-list-view-50.png"))); // NOI18N

invoiceHistoryButton.setText("Invoice History");
invoiceHistoryButton.setContentAreaFilled(false);
invoiceHistoryButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
invoiceHistoryButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
invoiceHistoryButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {
        invoiceHistoryButtonMouseEntered(evt);
    }

    public void mouseExited(java.awt.event.MouseEvent evt) {
        invoiceHistoryButtonMouseExited(evt);
    }

});

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

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

});

});
```

```
salesHistoyButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-list-view-50.png"))); // NOI18N

salesHistoyButton.setText("Sales History");

salesHistoyButton.setContentAreaFilled(false);

salesHistoyButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

salesHistoyButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

salesHistoyButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        salesHistoyButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        salesHistoyButtonMouseExited(evt);

    }

});

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

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

        salesHistoyButtonActionPerformed(evt);

    }

});

deliveryHistoryButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-list-view-50.png"))); // NOI18N

deliveryHistoryButton.setText("Delivery History");

deliveryHistoryButton.setContentAreaFilled(false);

deliveryHistoryButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

deliveryHistoryButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

deliveryHistoryButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        deliveryHistoryButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        deliveryHistoryButtonMouseExited(evt);

    }

});
```



```
.addComponent(deliveryHistoryButton)
.addContainerGap(1300, Short.MAX_VALUE))
);

jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jSeparator4)
.addGroup(jPanel2Layout.createSequentialGroup()
.addComponentGap()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(grnItemHistoryButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(grnHistoryButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)
.addComponent(invoiceHistoryButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(salesHistoryButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(deliveryHistoryButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addComponentGap())
.addComponent(jSeparator5)
);
```

```
jTabbedPane1.addTab("Reports", jPanel2);
```

```
jPanel3.setBackground(new java.awt.Color(227, 227, 227));
```

```
userButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-user-50.png"))); // NOI18N
userButton.setText("Usres");
userButton.setContentAreaFilled(false);
userButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
userButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
userButton.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
public void mouseEntered(java.awt.event.MouseEvent evt) {
    userButtonMouseEntered(evt);
}

public void mouseExited(java.awt.event.MouseEvent evt) {
    userButtonMouseExited(evt);
}

});

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

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

transporterButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-account-50.png"))); // NOI18N

transporterButton.setText("Transporters");
transporterButton.setContentAreaFilled(false);
transporterButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
transporterButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

transporterButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {
        transporterButtonMouseEntered(evt);
    }

    public void mouseExited(java.awt.event.MouseEvent evt) {
        transporterButtonMouseExited(evt);
    }
});

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

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

```
});

settingsButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-settings-50.png")));
// NOI18N

settingsButton.setText("Settings");
settingsButton.setContentAreaFilled(false);
settingsButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
settingsButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
settingsButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        settingsButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        settingsButtonMouseExited(evt);

    }

});;

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

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

        settingsButtonActionPerformed(evt);

    }

});;

jSeparator6.setOrientation(javax.swing.SwingConstants.VERTICAL);

backupRestoreButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/icons8-cloud-backup-restore-50.png")));
// NOI18N

backupRestoreButton.setText("Backup & Restore");
backupRestoreButton.setContentAreaFilled(false);
backupRestoreButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
backupRestoreButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
backupRestoreButton.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseEntered(java.awt.event.MouseEvent evt) {
```

```
        backupRestoreButtonMouseEntered(evt);

    }

    public void mouseExited(java.awt.event.MouseEvent evt) {

        backupRestoreButtonMouseExited(evt);

    }

});

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

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

        backupRestoreButtonActionPerformed(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()

        .addContainerGap()

        .addComponent(userButton)

        .addGap(0, 0, 0)

        .addComponent(transporterButton)

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

        .addComponent(jSeparator6, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(settingsButton)

        .addGap(0, 0, 0)

        .addComponent(backupRestoreButton)

        .addContainerGap(1481, Short.MAX_VALUE)

    );

jPanel3Layout.setVerticalGroup(
```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(transporterButton, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(userButton, javax.swing.GroupLayout.DEFAULT_SIZE, 80, Short.MAX_VALUE)
            .addComponent(settingsButton, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(backupRestoreButton, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    .addComponent(jSeparator6, javax.swing.GroupLayout.Alignment.TRAILING)
);
}

jTabbedPane1.addTab("Others", jPanel3);

jPanel4.setBackground(new java.awt.Color(227, 227, 227));

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel1.setText(" Login Date :");

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel4Layout.createSequentialGroup()
                    .addGap(14, 14, 14)
                    .addComponent(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel4Layout.createSequentialGroup()
                            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 236, Short.MAX_VALUE))
                        .addContainerGap())))
            .addGap(14, 14, 14)
        )
)
```

```
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jPanel7.setBackground(new java.awt.Color(188, 255, 188));

purchaseAmountLabel.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
purchaseAmountLabel.setText("0.00");
purchaseAmountLabel.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jLabel9.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel9.setText("Purchase");

javax.swing.GroupLayout jPanel7Layout = new javax.swing.GroupLayout(jPanel7);
jPanel7.setLayout(jPanel7Layout);
jPanel7Layout.setHorizontalGroup(
    jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel7Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel7Layout.createSequentialGroup()
            .addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(purchaseAmountLabel, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

```

```
.addContainerGap())
);

jPanel7Layout.setVerticalGroup(
    jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel7Layout.createSequentialGroup()
            .addGap(52, 52, 52)
            .addComponent(purchaseAmountLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 61, Short.MAX_VALUE)
            .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
                javax.swing.GroupLayout.PREFERRED_SIZE)
        )
        .addContainerGap()
);
}

jLabel10.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel10.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/bar-chart-grap-of-a-
function.png"))); // NOI18N

jPanel9.setBackground(new java.awt.Color(188, 255, 188));

transactionLabel.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
transactionLabel.setText("0.00");
transactionLabel.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jLabel13.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel13.setText("Transaction");

javax.swing.GroupLayout jPanel9Layout = new javax.swing.GroupLayout(jPanel9);
jPanel9.setLayout(jPanel9Layout);
jPanel9Layout.setHorizontalGroup(
    jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel9Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(transactionLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE)
        )
        .addGroup(jPanel9Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 180, javax.swing.GroupLayout.PREFERRED_SIZE)
        )
    )
);
```

```

.addContainerGap()

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

.addComponent(jLabel13, javax.swing.GroupLayout.DEFAULT_SIZE, 176, Short.MAX_VALUE))

.addContainerGap())

);

jPanel9Layout.setVerticalGroup(
jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel9Layout.createSequentialGroup()

.addGap(52, 52, 52)

.addComponent(transactionLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap())

);

jPanel10.setBackground(new java.awt.Color(188, 255, 188));

salesAmountLabel.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
salesAmountLabel.setText("0.00");
salesAmountLabel.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jLabel15.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel15.setText("Sales");

javax.swing.GroupLayout jPanel10Layout = new javax.swing.GroupLayout(jPanel10);
jPanel10.setLayout(jPanel10Layout);
jPanel10Layout.setHorizontalGroup(
jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
);
jPanel10Layout.setVerticalGroup(
jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
);

```

```
.addGroup(jPanel10Layout.createSequentialGroup()

    .addContainerGap()

    .addGroup(jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

        .addComponent(jLabel15, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    .addContainerGap())

);

jPanel10Layout.setVerticalGroup(

    jPanel10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel10Layout.createSequentialGroup()

        .addGap(52, 52, 52)

        .addComponent(salesAmountLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(jLabel15, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addContainerGap()

);

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel5Layout.createSequentialGroup()

        .addGap(120, 120, 120)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

            .addComponent(jPanel9, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(jPanel7, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

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


```

```

.addComponent(jPanel10, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 503,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(597, Short.MAX_VALUE)

);

jPanel5Layout.setVerticalGroup(
jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel5Layout.createSequentialGroup()

.addGap(90, 90, 90)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 413,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPanel7, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel10, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

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

.addComponent(jPanel9, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

.addContainerGap(117, Short.MAX_VALUE))

);

jPanel6.setBackground(new java.awt.Color(231, 255, 255));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
jLabel2.setText("PAMUNUWILA HARDWARE (Pvt) Ltd");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setText("PAMUNUWILA HARDWARE is a HARDWARE SHOP. This Hardware Shop Management System (HSMS) is developed to make their day-to-day operations easier and more efficient.");

```

```
jLabel4.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N  
jLabel4.setText("Email : pamunuwilahardware@gmail.com");  
  
jLabel5.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N  
jLabel5.setText("Contact No : 0775561834");  
  
jLabel6.setText("All Rights Reserved to PAMUNUWILA HARDWARE");  
  
jPanel8.setBackground(new java.awt.Color(252, 209, 167));  
  
jLabel7.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
jLabel7.setText("User");  
  
usernameLabel.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
  
userNoLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
  
jSeparator7.setOrientation(javax.swing.SwingConstants.VERTICAL);  
  
jobroleNoLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
  
jobrolenameLabel.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
  
jSeparator8.setOrientation(javax.swing.SwingConstants.VERTICAL);  
  
jLabel8.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
jLabel8.setText("Login Date");  
  
datetimeLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);
jPanel8.setLayout(jPanel8Layout);
jPanel8Layout.setHorizontalGroup(
    jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel8Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel7)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(usernameLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 107,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(userNoLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jSeparator7, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jobroleNoLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jobroleNameLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jSeparator8, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(datetimeLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 474,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
jPanel8Layout.setVerticalGroup(
```



```
.addContainerGap()

.addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addComponent(jLabel2)

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

.addComponent(jLabel12))

.addComponent(jLabel3)

.addGroup(jPanel6Layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(73, 73, 73)

.addComponent(jLabel5)))

.addGap(0, 558, Short.MAX_VALUE))

.addGroup(jPanel6Layout.createSequentialGroup()

.addComponent(jLabel6)

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

.addComponent(jLabel11))

.addContainerGap())

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

);

jPanel6Layout.setVerticalGroup(
jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addComponent(jLabel2)

.addComponent(jLabel12)

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

.addComponent(jLabel3)
```

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

.addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(12, 12, 12)

.addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(jLabel11))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18, Short.MAX_VALUE)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup()

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

.addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(0, 0, 0)

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

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

);

layout.setVerticalGroup()

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

.addGroup(layout.createSequentialGroup()
```

```
.addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 136,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(0, 0, 0)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGap(0, 0, 0)

.addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

);

pack();

 setLocationRelativeTo(null);

}// </editor-fold>
```

```
private void homeButtonMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

homeButton.setContentAreaFilled(true);

}
```

```
private void homeButtonMouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

homeButton.setContentAreaFilled(false);

}
```

```
private void grnButtonMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

grnButton.setContentAreaFilled(true);

}
```

```
private void grnButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    grnButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void salesButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    salesButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void salesButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    salesButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void deliveryButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    deliveryButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void deliveryButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    deliveryButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void productButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    productButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void productButtonMouseExited(java.awt.event.MouseEvent evt) {
```

```
// TODO add your handling code here:  
productButton.setContentAreaFilled(false);  
  
}  
  
  
private void supplierButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    supplierButton.setContentAreaFilled(true);  
  
}  
  
  
private void supplierButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    supplierButton.setContentAreaFilled(false);  
  
}  
  
  
private void customerButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    customerButton.setContentAreaFilled(true);  
  
}  
  
  
private void customerButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    customerButton.setContentAreaFilled(false);  
  
}  
  
  
private void companyButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    companyButton.setContentAreaFilled(true);  
  
}  
  
  
private void companyButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
}
```

```
companyButton.setContentAreaFilled(false);

}
```

```
private void aboutButtonMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

aboutButton.setContentAreaFilled(true);

}
```

```
private void aboutButtonMouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

aboutButton.setContentAreaFilled(false);

}
```

```
private void grnHistoryButtonMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

grnHistoryButton.setContentAreaFilled(true);

}
```

```
private void grnHistoryButtonMouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

grnHistoryButton.setContentAreaFilled(false);

}
```

```
private void grnItemHistoryButtonMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

grnItemHistoryButton.setContentAreaFilled(true);

}
```

```
private void grnItemHistoryButtonMouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

grnItemHistoryButton.setContentAreaFilled(false);
```

```
}
```

```
private void invoiceHistoryButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    invoiceHistoryButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void invoiceHistoryButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    invoiceHistoryButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void salesHistoyButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    salesHistoyButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void salesHistoyButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    salesHistoyButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void deliveryHistoryButtonMouseEntered(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    deliveryHistoryButton.setContentAreaFilled(true);  
}  
  
}
```

```
private void deliveryHistoryButtonMouseExited(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    deliveryHistoryButton.setContentAreaFilled(false);  
}  
  
}
```

```
private void userButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    userButton.setContentAreaFilled(true);  
  
}
```

```
private void userButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    userButton.setContentAreaFilled(false);  
  
}
```

```
private void transporterButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    transporterButton.setContentAreaFilled(true);  
  
}
```

```
private void transporterButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    transporterButton.setContentAreaFilled(false);  
  
}
```

```
private void settingsButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    settingsButton.setContentAreaFilled(true);  
  
}
```

```
private void settingsButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    settingsButton.setContentAreaFilled(false);  
  
}
```

```
private void homeButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new HomePage(userNoThis, usernameCheckedThis, jobroleNoThis, jobrolenameThis).setVisible(true);  
  
    this.dispose();  
  
}
```

```
private void grnButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new GrnManager().setVisible(true);  
  
}
```

```
private void salesButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new SalesManager().setVisible(true);  
  
}
```

```
private void deliveryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new DeliveryManager().setVisible(true);  
  
}
```

```
private void productButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new ProductManager().setVisible(true);  
  
}
```

```
private void supplierButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new SupplierManager().setVisible(true);  
  
}
```

```
private void customerButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new CustomerManager().setVisible(true);  
}  
  
}
```

```
private void companyButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new CompanyManager().setVisible(true);  
}  
  
}
```

```
private void aboutButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new About().setVisible(true);  
}  
  
}
```

```
private void grnHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new GenerateGrnHistoryDialog(this, rootPaneCheckingEnabled).setVisible(true);  
}  
  
}
```

```
private void grnItemHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new GenerateGrnItemHistoryDialog(this, rootPaneCheckingEnabled).setVisible(true);  
}  
  
}
```

```
private void invoiceHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new GenerateInvoiceHistoryDialog(this, rootPaneCheckingEnabled).setVisible(true);  
}  
  
}
```

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

```
// TODO add your handling code here:  
new GenerateSalesHistoryDialog(this, rootPaneCheckingEnabled).setVisible(true);  
}  
  
  
private void deliveryHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
    new GenerateDeliveryHistoryDialog(this, rootPaneCheckingEnabled).setVisible(true);  
}  
  
  
private void userButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
    new UserManager().setVisible(true);  
}  
  
  
private void transporterButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
    new TransporterManager().setVisible(true);  
}  
  
  
private void settingsButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
    //new ProductManager().setVisible(true);  
}  
  
  
private void backupRestoreButtonMouseEntered(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
    userButton.setContentAreaFilled(true);  
}  
  
  
private void backupRestoreButtonMouseExited(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
}
```

```
userButton.setContentAreaFilled(false);

}

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

    // TODO add your handling code here:

    new BackupAndRestore().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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
    // try {
    //     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
    //         if ("Windows".equals(info.getName())) {
    //             javax.swing.UIManager.setLookAndFeel(info.getClassName());
    //             break;
    //         }
    //     }
    // }
}
```

```
//      }

//      }

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

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

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

//  } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//      java.util.logging.Logger.getLogger(HomePage.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 HomePage().setVisible(true);
    }
});
```

```
}

// Variables declaration - do not modify

private javax.swing.JButton aboutButton;

private javax.swing.JButton backupRestoreButton;

private javax.swing.JButton companyButton;

private javax.swing.JButton customerButton;

private javax.swing.JLabel datetimelabel;

private javax.swing.JButton deliveryButton;

private javax.swing.JButton deliveryHistoryButton;

private javax.swing.JButton grnButton;

private javax.swing.JButton grnHistoryButton;
```

```
private javax.swing.JButton grnlItemHistoryButton;
private javax.swing.JButton homeButton;
private javax.swing.JButton invoiceHistoryButton;
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel15;
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.JPanel jPanel10;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel7;
private javax.swing.JPanel jPanel8;
private javax.swing.JPanel jPanel9;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
```

```

private javax.swing.JSeparator jSeparator4;
private javax.swing.JSeparator jSeparator5;
private javax.swing.JSeparator jSeparator6;
private javax.swing.JSeparator jSeparator7;
private javax.swing.JSeparator jSeparator8;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JLabel jobroleNameLabel;
private javax.swing.JLabel jobroleNoLabel;
private javax.swing.JButton productButton;
private javax.swing.JLabel purchaseAmountLabel;
private javax.swing.JLabel salesAmountLabel;
private javax.swing.JButton salesButton;
private javax.swing.JButton salesHistoryButton;
private javax.swing.JButton settingsButton;
private javax.swing.JButton supplierButton;
private javax.swing.JLabel transactionLabel;
private javax.swing.JButton transporterButton;
private javax.swing.JButton userButton;
private javax.swing.JLabel userNoLabel;
private javax.swing.JLabel usernameLabel;
// End of variables declaration
}


```

HomeModel.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package model;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Date;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class HomeModel {

    private Connection dbConnection;
    static Logger logger = Logger.getLogger(HomeModel.class.getName());

    public HomeModel() {
        dbConnection = database.dbConnection.getConnection();
    }

    public Object[] getPurchaseAmount(){
        String query = "SELECT SUM(grn_amount) FROM grn" ;
        logger.debug("SQL query String: " + query);

        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            ResultSet rs = psm.executeQuery();
            logger.debug("Query execution done");
            Object[] rowData = null;
            while (rs.next()) {
                double purchaseAmount = rs.getDouble("SUM(grn_amount)");
                rowData = new Object[]{purchaseAmount};
                logger.debug("return rowData");
            }
        } catch (SQLException e) {
            logger.error("Error executing SQL query: " + e.getMessage());
        }
    }
}
```

```
        return rowData;
    }

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
}

public Object[] getSalesAmount(){

String query = "SELECT SUM(customer_invoice_amount) FROM customer_invoice;" ;
logger.debug("SQL query String: " + query);

try {

PreparedStatement psm = dbConnection.prepareStatement(query);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

while (rs.next()) {

    double salesAmount = rs.getDouble("SUM(customer_invoice_amount)");

    logger.debug("return rowData");

    rowData = new Object[]{salesAmount};

    return rowData;
}

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
```


Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: GRN Manager Create GRN

GUI Number : 04

Use case Name: GRN Management

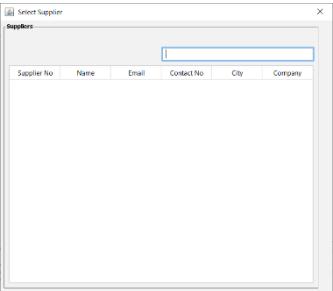
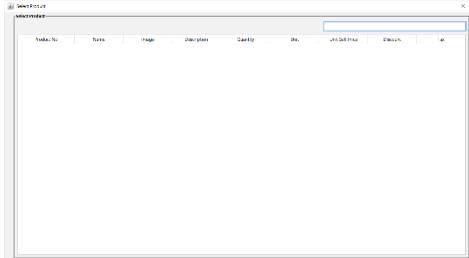
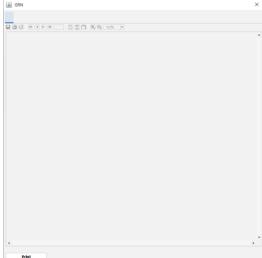
01. Brief Description of the GUI:

The main objective of this interface is to create GRN, save GRN and generate GRN reports.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|--------------|---|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the GRN Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Insert product details | One GRN can have multiple products. There should be space to add those products. That space is filled by this JTable (one product per table row can be inserted.) |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert GRN details | To enter or display the details needed to create a GRN. |
| JDateChooser | Display date | To display the current date |
| JButtons | Add table row, Delete table row, Save GRN, Clear text fields & Generate GRN reports | Buttons are needed to add rows to a table, delete rows from a table, save a GRN, clear text fields and generate GRN reports. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|--|---|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To enter the details of the supplier who is supplying the products. |
| | 02.  | To include details of supplied products. |
| | 03.  | To Generate GRN reports |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

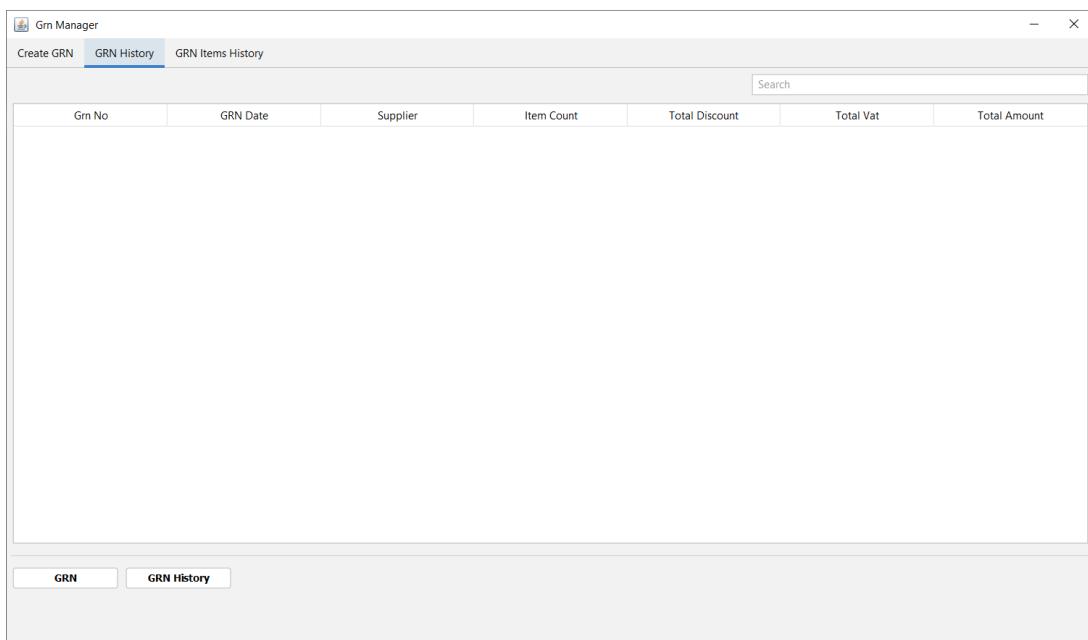
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: GRN Manager GRN History

GUI Number : 05

Use case Name: GRN Management



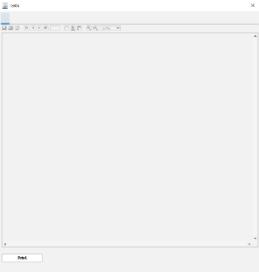
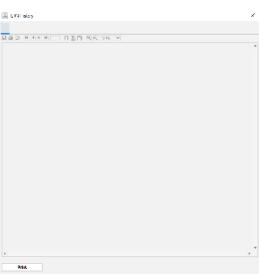
01. Brief Description of the GUI:

The main objective of this interface is to display the GRN history, search for the GRN, select a GRN and generate its GRN report and generate the GRN history report.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the GRN Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display the GRN history | During a search operation, the user can find nearly similar GRN lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |
| JButtons | Generate GRN reports & The GRN history report | Buttons are needed to generate a GRN selected from the table and to generate the GRN history. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---|---|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To generate a GRN selected from the table |
| | 02.  | To Generate the GRN history report |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

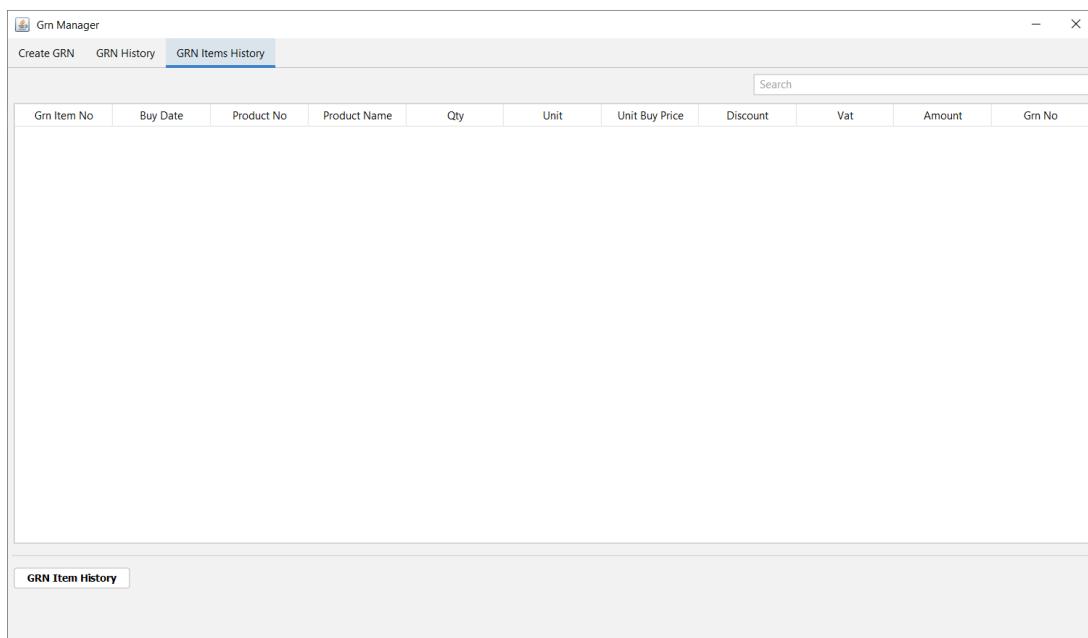
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: GRN Manager GRN Item History

GUI Number : 06

Use case Name: GRN Management



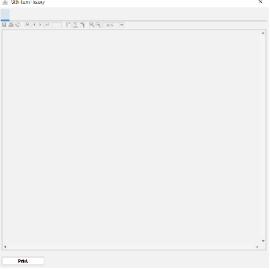
01. Brief Description of the GUI:

The main objective of this interface is to display the GRN item history, search for the GRN item and generate the GRN item history report.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--------------------------------------|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the GRN Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display the GRN item history | During a search operation, the user can find nearly similar GRN item lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |
| JButtons | Generate the GRN item history report | A button is required to generate the GRN item history. This JButton meets that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|--|---|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To Generate the GRN item history report |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

GrnManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.GrnModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GrnManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form GrnManager  
     */  
  
    private GrnModel grnModel;  
  
    private LoadSupplierDialog loadSupplierDialog;  
  
    private LoadProductDialog loadProductDialog;  
  
    static Logger logger = Logger.getLogger(GrnManager.class.getName());  
  
  
    DefaultTableModel grnTableModel;  
  
    DefaultTableModel grnTableModelFull;  
  
    DefaultTableModel grnItemTableModelFull;  
  
  
    int supplierNo;  
  
  
    public GrnManager() {  
        initComponents();  
  
        logger.debug("Access to GrnManager.java");  
  
  
        grnTableModel = (DefaultTableModel) grnTable.getModel();  
        grnTableModelFull = (DefaultTableModel) grnTableFull.getModel();  
        grnItemTableModelFull = (DefaultTableModel) grnItemTableFull.getModel();  
  
  
        grnModel = new GrnModel();  
        loadSupplierDialog = new LoadSupplierDialog(this, rootPaneCheckingEnabled);  
        loadProductDialog = new LoadProductDialog(this, rootPaneCheckingEnabled);  
  
  
        grnModel.loadGrnTableFull("", grnTableModelFull);  
        grnModel.loadGrnItemTableFull("", grnItemTableModelFull);
```

```
grnNoLoad();

grnDateChooser.setDate(new java.util.Date());

}

/**

 * 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() {

    jTabbedPane1 = new javax.swing.JTabbedPane();

    jPanel1 = new javax.swing.JPanel();

    jPanel4 = new javax.swing.JPanel();

    grnNoField = new javax.swing.JTextField();

    grnDateChooser = new com.toedter.calendar.JDateChooser();

    supplierNameField = new javax.swing.JTextField();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jScrollPane1 = new javax.swing.JScrollPane();

    grnTable = new javax.swing.JTable();

    jPanel5 = new javax.swing.JPanel();

    discountField = new javax.swing.JTextField();

    totalAmountField = new javax.swing.JTextField();

    netAmountField = new javax.swing.JTextField();
```

```
vatField = new javax.swing.JTextField();

itemCountField = new javax.swing.JTextField();

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();

jSeparator1 = new javax.swing.JSeparator();

jPanel6 = new javax.swing.JPanel();

addRowButton = new javax.swing.JButton();

deleteRowButton = new javax.swing.JButton();

saveGrnButton = new javax.swing.JButton();

newGrnButton = new javax.swing.JButton();

GrnButton = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

grnTableFull = new javax.swing.JTable();

jSeparator2 = new javax.swing.JSeparator();

grnButton1 = new javax.swing.JButton();

grnHistoryButton = new javax.swing.JButton();

jPanel3 = new javax.swing.JPanel();

keywordFieldFull1 = new javax.swing.JTextField();

jScrollPane4 = new javax.swing.JScrollPane();

grnItemTableFull = new javax.swing.JTable();

jSeparator3 = new javax.swing.JSeparator();

grnItemHistoryButton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Grn Manager");

setResizable(false);
```

```
grnNoField.setEnabled(false);

supplierNameField.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        supplierNameFieldMouseClicked(evt);
    }
});

jLabel1.setText("GRN No");

jLabel2.setText("Supplier");

jLabel3.setText("GRN Date");

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel4Layout.createSequentialGroup()
                .addGap(84, 84, 84)
                .addComponent(grnNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 229,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
            )
            .addComponent(jPanel4Layout.createSequentialGroup()
                .addGap(84, 84, 84)
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
            )
        )
    )
);
```

```

javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(grnDateChooser, javax.swing.GroupLayout.DEFAULT_SIZE, 277, Short.MAX_VALUE))

    .addComponent(supplierNameField))

    .addGap(654, 654, 654))

);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(grnNoField, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)
            .addComponent(grnDateChooser, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(supplierNameField, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE))
        .addContainerGap())
    );
}

grnTable.setFont(new java.awt.Font("Tahoma", 0, 15)); // NOI18N

grnTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
    },
    new String [] {
    }
}
);

```

```

    "", "Product", "Unit Sell Price", "Qty", "Unit", "Unit Buy Price", "Discount", "VAT", "Amount"
}

) {

Class[] types = new Class [] {
    java.lang.Object.class, java.lang.String.class, java.lang.Double.class, java.lang.Double.class,
    java.lang.String.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class
};

boolean[] canEdit = new boolean [] {
    false, false, true, true, false, true, true, true, false
};

public Class getColumnClass(int columnIndex) {
    return types [columnIndex];
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

grnTable.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
grnTable.setDragEnabled(true);
grnTable.setRowHeight(30);
grnTable.setSelectionBackground(new java.awt.Color(204, 204, 204));
grnTable.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        grnTableMouseClicked(evt);
    }
});

grnTable.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        grnTableKeyReleased(evt);
    }
});

```

```
}

});

jScrollPane1.setViewportView(grnTable);

if (grnTable.getColumnModel().getColumnCount() > 0) {

    grnTable.getColumnModel().getColumn(0).setMinWidth(25);

    grnTable.getColumnModel().getColumn(0).setMaxWidth(25);

}

discountField.setText("0.00");

discountField.setEnabled(false);

totalAmountField.setText("0.00");

totalAmountField.setEnabled(false);

netAmountField.setText("0.00");

netAmountField.setEnabled(false);

vatField.setText("0.00");

vatField.setEnabled(false);

itemCountField.setText("0");

itemCountField.setEnabled(false);

jLabel4.setText("Item Count");

jLabel5.setText("Discount");

jLabel6.setText("VAT");

jLabel7.setText("Total Amount");
```

```
jLabel8.setText("Net Amount");

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
jPanel5Layout.setHorizontalGroup(
jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel5Layout.createSequentialGroup()
.addContainerGap()
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
.addComponent(jLabel5, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jLabel4, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(18, 18, 18)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel5Layout.createSequentialGroup()
.addComponent(vatField, javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(discountField, javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 65, Short.MAX_VALUE)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(18, 18, 18)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(totalAmountField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 194, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(netAmountField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 194, javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
.addComponent(itemCountField))

.addContainerGap()

);

jPanel5Layout.setVerticalGroup

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(itemCountField, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(totalAmountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(discountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(vatField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(netAmountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

);
```

```
addRowButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
addRowButton.setText("Add Row");
addRowButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addRowButtonActionPerformed(evt);
    }
});
```

```
deleteRowButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
deleteRowButton.setText("Delete Row");
deleteRowButton.setEnabled(false);
deleteRowButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        deleteRowButtonActionPerformed(evt);
    }
});
```

```
saveGrnButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
saveGrnButton.setText("Save GRN");
saveGrnButton.setEnabled(false);
saveGrnButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        saveGrnButtonActionPerformed(evt);
    }
});
```

```
newGrnButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newGrnButton.setText("New");
newGrnButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newGrnButtonActionPerformed(evt);
    }
});
```

```
}

});

GrnButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
GrnButton.setText("GRN");
GrnButton.setEnabled(false);
GrnButton.addActionListener(new java.awt.event.ActionListener() {

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

        GrnButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
jPanel6.setLayout(jPanel6Layout);
jPanel6Layout.setHorizontalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(deleteRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(saveGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(newGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(GrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponentContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);
```

```
);

jPanel6Layout.setVerticalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(addRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(deleteRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(saveGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(newGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(GrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jScrollPane1)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel6Layout, javax.swing.GroupLayout.Alignment.TRAILING,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    )
);
```

```

    .addGap(18, 18, 18)

    .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addComponent(jSeparator1)

    .addContainerGap()

);

jPanel1Layout.setVerticalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 439,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap(48, Short.MAX_VALUE))

);

jTabbedPane1.addTab("Create GRN", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

```

```
public void focusGained(java.awt.event.FocusEvent evt) {
    keywordFieldFullFocusGained(evt);
}

public void focusLost(java.awt.event.FocusEvent evt) {
    keywordFieldFullFocusLost(evt);
}

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        keywordFieldFullKeyReleased(evt);
    }
});

grnTableFull.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "Grn No", "GRN Date", "Supplier", "Item Count", "Total Discount", "Total Vat", "Total Amount"
        }
    },
    new String [] {
        ""
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

grnTableFull.setRowHeight(25);
jScrollPane3.setViewportView(grnTableFull);
```

```
grnButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
grnButton1.setText("GRN");
grnButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        grnButton1ActionPerformed(evt);
    }
});
};

grnHistoryButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
grnHistoryButton.setText("GRN History");
grnHistoryButton.setOpaque(false);
grnHistoryButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        grnHistoryButtonActionPerformed(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()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jPanel2Layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jSeparator2))
        .addContainerGap())

```

```
.addGroup(jPanel2Layout.createSequentialGroup()

    .addComponent(grnButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(grnHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap()

);

jPanel2Layout.setVerticalGroup

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

.addGroup(jPanel2Layout.createSequentialGroup()

    .addContainerGap()

    .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 613,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

        .addComponent(grnButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(grnHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addContainerGap(76, Short.MAX_VALUE))

);

jTabbedPane1.addTab("GRN History", jPanel2);

keywordFieldFull1.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull1.setText("Search");
```

```

keywordFieldFull1.setToolTipText("Search");

keywordFieldFull1.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFull1FocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFull1FocusLost(evt);

    }

});

keywordFieldFull1.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFull1KeyReleased(evt);

    }

});

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

        new String [] {
            "Grn Item No", "Buy Date", "Product No", "Product Name", "Qty", "Unit", "Unit Buy Price", "Discount",
            "Vat", "Amount", "Grn No"
        }
    },
    new boolean[] canEdit = new boolean [] {
        false, true, false, false, false, false, false, false, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {

        return canEdit [columnIndex];
    }
}

```

```
});

grnItemTableFull.setRowHeight(25);

jScrollPane4.setViewportView(grnItemTableFull);

grnItemHistoryButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
grnItemHistoryButton.setText("GRN Item History");
grnItemHistoryButton.addActionListener(new java.awt.event.ActionListener() {

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

        grnItemHistoryButtonActionPerformed(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(0, 1003, Short.MAX_VALUE)
        .addComponent(keywordFieldFull1, javax.swing.GroupLayout.PREFERRED_SIZE, 469,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jSeparator3)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addComponent(grnItemHistoryButton)
        .addGap(0, 0, Short.MAX_VALUE)))
    .addGapContainerGap()
);
jPanel3Layout.setVerticalGroup(jPanel3Layout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)
);
```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(keywordFieldFull1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 613,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(grnItemHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(76, Short.MAX_VALUE))
);
```

```
jTabbedPane1.addTab("GRN Items History", jPanel3);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1501,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 835,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGap(0, 0, Short.MAX_VALUE))  
);  
  
pack();  
setLocationRelativeTo(null);  
}// </editor-fold>  
  
  
private void supplierNameFieldMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    loadSupplierDialog.setVisible(true);  
  
    Object[] supplierData = loadSupplierDialog.getSupplierData();  
    supplierNo = (int) supplierData[0];  
    supplierNameField.setText((String) supplierData[1]);  
    //System.out.println(supplierNo);  
  
}  
  
  
private void addRowButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (supplierNo != 0 && supplierNameField.getText().length() > 0) {  
        grnTableModel.addRow(new Object[]{0, null, 0.00, 0.00, "Number", 0.00, 0.00, 0.00, 0.00});  
        deleteRowButton.setEnabled(true);  
        saveGrnButton.setEnabled(true);  
        itemCountField.setText(String.valueOf(grnTableModel.getRowCount()));  
    } else {  
        loadSupplierDialog.setVisible(true);  
        Object[] supplierData = loadSupplierDialog.getSupplierData();  
        supplierNo = (int) supplierData[0];  
        supplierNameField.setText((String) supplierData[1]);  
    }  
}
```

```
}
```

```
private void deleteRowButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    int rowCount = grnTable.getRowCount();  
  
    int selectedRow = grnTable.getSelectedRow();  
  
    int selectedRowCount = grnTable.getSelectedRowCount();  
  
  
    if (selectedRowCount == 1) {  
        grnTableModel.removeRow(selectedRow);  
  
        if(rowCount == 1){  
            deleteRowButton.setEnabled(false);  
  
            saveGrnButton.setEnabled(false);  
  
        }  
    } else {  
        //if (rowCount == 0) {  
  
        // JOptionPane.showConfirmDialog(rootPane, "Table is Empty !", "Error", -1,  
        // JOptionPane.ERROR_MESSAGE);  
  
        // } else {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Select any Single Row !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        logger.error("Error: Please Select any Single Row !");  
  
        //}  
    }  
}
```

```
private void grnTableMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    int selectedRow = grnTable.getSelectedRow();  
  
    int selectedColumn = grnTable.getSelectedColumn();
```

```
if (selectedColumn == 1) {  
    loadProductDialog.setVisible(true);  
  
    Object[] productData = loadProductDialog.getProductData();  
  
    Object productNoo = productData[0];  
    String productName = (String) productData[1];  
    Object productQty = productData[2];  
    //int productUnitNo = (int) productData[3];  
    Object productUnitName = productData[4];  
    Object productUnitSellPrice = productData[5];  
    //Object productDiscount = productData[6];  
    //Object productTax = productData[7];  
  
    grnTableModel.setValueAt(productNoo, selectedRow, 0);  
    grnTableModel.setValueAt(productName, selectedRow, 1);  
    grnTableModel.setValueAt(productUnitSellPrice, selectedRow, 2);  
    grnTableModel.setValueAt(productQty, selectedRow, 3);  
    grnTableModel.setValueAt(productUnitName, selectedRow, 4);  
    grnTableModel.setValueAt(0.00, selectedRow, 5);  
    grnTableModel.setValueAt(0.00, selectedRow, 6);  
    grnTableModel.setValueAt(0.00, selectedRow, 7);  
    grnTableModel.setValueAt(0.00, selectedRow, 8);  
  
} else {  
  
    if ((int) grnTableModel.getValueAt(selectedRow, 0) == 0 || grnTableModel.getValueAt(selectedRow, 1) == null) {  
  
        loadProductDialog.setVisible(true);  
        Object[] productData = loadProductDialog.getProductData();  
    }  
}
```

```
Object productNoo = productData[0];

String productName = (String) productData[1];

Object productQty = productData[2];

//int productUnitNo = (int) productData[3];

Object productUnitName = productData[4];

Object productUnitSellPrice = productData[5];

//Object productDiscount = productData[6];

//Object productTax = productData[7];

grnTableModel.setValueAt(productNoo, selectedRow, 0);

grnTableModel.setValueAt(productName, selectedRow, 1);

grnTableModel.setValueAt(productUnitSellPrice, selectedRow, 2);

grnTableModel.setValueAt(productQty, selectedRow, 3);

grnTableModel.setValueAt(productUnitName, selectedRow, 4);

grnTableModel.setValueAt(0.00, selectedRow, 5);

grnTableModel.setValueAt(0.00, selectedRow, 6);

grnTableModel.setValueAt(0.00, selectedRow, 7);

grnTableModel.setValueAt(0.00, selectedRow, 8);

}

}

calculateValues();

}
```

```
private void grnTableKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

calculateValues();

}
```

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

// TODO add your handling code here:
```

```

int rowCount = grnTableModel.getRowCount();

boolean checkEmptyRows = false;

for (int i = 0; i < rowCount; i++) {

    int productNo = (int) grnTableModel.getValueAt(i, 0);

    String productName = (String) grnTableModel.getValueAt(i, 1);

    if (productNo == 0 || productName == null) {

        checkEmptyRows = true;

        break;

    }

}

if (checkEmptyRows) {

    JOptionPane.showConfirmDialog(rootPane, "There is some Empty Rows !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    logger.error("Error: There is some Empty Rows !");

} else {

    String grnNo = grnNoField.getText();

    Date grnDate = grnDateChooser.getDate();

    //supplierNo in hear

    String itemCount = itemCountField.getText();

    String totalDiscount = discountField.getText();

    String totalVat = vatField.getText();

    String totalAmount = totalAmountField.getText();

    String grnResult = grnModel.saveGrn(grnNo, grnDate, supplierNo, itemCount, totalDiscount, totalVat,
totalAmount);

    String grnItemResult = null;

    for (int i = 0; i < rowCount; i++) {

        int productNo = (int) grnTableModel.getValueAt(i, 0);

```

```

        double unitSellPrice = (double) grnTableModel.getValueAt(i, 2);

        double qty = (double) grnTableModel.getValueAt(i, 3);

        double unitBuyPrice = (double) grnTableModel.getValueAt(i, 5);

        double discount = (double) grnTableModel.getValueAt(i, 6);

        double vat = (double) grnTableModel.getValueAt(i, 7);

        double amount = (double) grnTableModel.getValueAt(i, 8);

    }

    grnItemResult = grnModel.saveGrnItem(productNo, unitSellPrice, qty, unitBuyPrice, discount, vat, amount,
    grnNo);

}

if (grnResult.equals("Success") && grnItemResult.equals("Success")) {

    JOptionPane.showConfirmDialog(rootPane, grnItemResult, "Message", -1, 1);

    GrnButton.setEnabled(true);

    addRowButton.setEnabled(false);

    deleteRowButton.setEnabled(false);

    saveGrnButton.setEnabled(false);

    grnModel.loadGrnTableFull("", grnTableModelFull);

    grnModel.loadGrnItemTableFull("", grnItemTableModelFull);

    logger.debug("Message: " + grnItemResult);

} else {

    JOptionPane.showConfirmDialog(rootPane, grnItemResult, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: " + grnItemResult);

}

}

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

    // TODO add your handling code here:
}

```

```
int rowCount = grnTableModel.getRowCount();

grnNoLoad();
supplierNo = 0;
supplierNameField.setText("");
grnDateChooser.setDate(new java.util.Date());

itemCountField.setText("0");
discountField.setText("0.00");
vatField.setText("0.00");
totalAmountField.setText("0.00");
netAmountField.setText("0.00");

addRowButton.setEnabled(true);
deleteRowButton.setEnabled(false);
saveGrnButton.setEnabled(false);
GrnButton.setEnabled(false);

for (int i = rowCount - 1; i >= 0; i--) {
    grnTableModel.removeRow(i);
}

logger.debug("clearFields");
}

private void GrnButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int grnNo = Integer.parseInt(grnNoField.getText());
    new GenerateGrnDialog(this, true, grnNo).setVisible(true);
}
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {  
  
    if (keywordFieldFull.getText().equals("Search")) {  
  
        keywordFieldFull.setText("");  
  
        keywordFieldFull.setForeground(new Color(0, 0, 0));  
  
    }  
  
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (keywordFieldFull.getText().equals("")) {  
  
        keywordFieldFull.setText("Search");  
  
        keywordFieldFull.setForeground(new Color(153, 153, 153));  
  
    }  
  
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keywordFull = keywordFieldFull.getText();  
  
    grnModel.loadGrnTableFull(keywordFull, grnTableModelFull);  
  
}
```

```
private void grnButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    int rowCount = grnTableFull.getRowCount();  
  
    int selectedRow = grnTableFull.getSelectedRow();  
  
    int selectedRowCount = grnTableFull.getSelectedRowCount();  
  
  
    if (selectedRowCount == 1) {  
  
        int grnNo = (int) grnTableModelFull.getValueAt(selectedRow, 0);  
  
        new GenerateGrnDialog(this, true, grnNo).setVisible(true);  
  
    } else {
```

```
if (rowCount == 0) {

    JOptionPane.showConfirmDialog(rootPane, "Table is Empty !", "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: Table is Empty !");

} else {

    JOptionPane.showConfirmDialog(rootPane, "Please Select any Single Row !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    logger.error("Error: Please Select any Single Row !");

}

}

}
```

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

// TODO add your handling code here:

new GenerateGrnHistoryDialog(this, true).setVisible(true);

}
```

```
private void keywordFieldFull1FocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordFieldFull1.getText().equals("Search")) {

    keywordFieldFull1.setText("");

    keywordFieldFull1.setForeground(new Color(0, 0, 0));

}

}
```

```
private void keywordFieldFull1FocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordFieldFull1.getText().equals("")) {

    keywordFieldFull1.setText("Search");

    keywordFieldFull1.setForeground(new Color(153, 153, 153));

}
```

```
    }

}

private void keywordFieldFull1KeyReleased(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    String keywordFull1 = keywordFieldFull1.getText();

    grnModel.loadGrnItemTableFull(keywordFull1, grnItemTableModelFull);

}
```

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

    // TODO add your handling code here:

    new GenerateGrnItemHistoryDialog(this, true).setVisible(true);

}
```

```
private void grnNoLoad() {

    int loadGrnNo = grnModel.getGrnNo();

    loadGrnNo++;

    grnNoField.setText(new String(loadGrnNo + ""));

}
```

```
private void calculateValues() {

    int selectedRow = grnTable.getSelectedRow();

    int rowCount = grnTableModel.getRowCount();

    double productQty = (double) grnTableModel.getValueAt(selectedRow, 3);

    double productUnitBuyPrice = (double) grnTableModel.getValueAt(selectedRow, 5);

    double productDiscount = (double) grnTableModel.getValueAt(selectedRow, 6);

    double productVat = (double) grnTableModel.getValueAt(selectedRow, 7);

    double productAmount = (productUnitBuyPrice - productDiscount + productVat) * productQty;

    grnTableModel.setValueAt(productAmount, selectedRow, 8);

}
```

```
itemCountField.setText(String.valueOf(rowCount));

double totalDiscount = 0;
double totalVat = 0;
double totalAmount = 0;

for (int i = 0; i < rowCount; i++) {
    double rowQty = (double) grnTableModel.getValueAt(i, 3);

    double rowDiscount = (double) grnTableModel.getValueAt(i, 6);
    totalDiscount += rowDiscount * rowQty;

    double rowVat = (double) grnTableModel.getValueAt(i, 7);
    totalVat += rowVat * rowQty;

    double rowAmount = (double) grnTableModel.getValueAt(i, 8);
    totalAmount += rowAmount;
}

discountField.setText(String.valueOf(totalDiscount));
vatField.setText(String.valueOf(totalVat));
totalAmountField.setText(String.valueOf(totalAmount));
netAmountField.setText(String.valueOf(totalAmount));

logger.debug("calculateValues");
}

/***
 * @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 {
    UIManager.setLookAndFeel(new FlatIntelliJLaf());
    UIManager.put("Table.showHorizontalLines", true);
    UIManager.put("Table.showVerticalLines", true);
    UIManager.put("ScrollBar.showButtons", true);
    UIManager.put("ScrollBar.width", 16);
} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(GrnManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(GrnManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(GrnManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(GrnManager.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 GrnManager().setVisible(true);
    }
});

}

// Variables declaration - do not modify

private javax.swing.JButton GrnButton;
private javax.swing.JButton addRowButton;
private javax.swing.JButton deleteRowButton;
private javax.swing.JTextField discountField;
private javax.swing.JButton grnButton1;
private com.toedter.calendar.JDateChooser grnDateChooser;
private javax.swing.JButton grnHistoryButton;
private javax.swing.JButton grnItemHistoryButton;
private javax.swing.JTable grnItemTableFull;
private javax.swing.JTextField grnNoField;
private javax.swing.JTable grnTable;
private javax.swing.JTable grnTableFull;
private javax.swing.JTextField itemCountField;
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.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTextField keywordFieldFull;
private javax.swing.JTextField keywordFieldFull1;
private javax.swing.JTextField netAmountField;
private javax.swing.JButton newGrnButton;
private javax.swing.JButton saveGrnButton;
private javax.swing.JTextField supplierNameField;
private javax.swing.JTextField totalAmountField;
private javax.swing.JTextField vatField;
// End of variables declaration
}
```

GrnModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package model;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Date;
import javax.swing.table.DefaultTableModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class GrnModel {

    private Connection dbConnection;
    static Logger logger = Logger.getLogger(GrnModel.class.getName());

    public GrnModel() {
        dbConnection = database.dbConnection.getConnection();
    }

    public int getGrnNo() {
        String query = "SELECT * FROM grn ORDER BY grn_no DESC LIMIT 1";
        logger.debug("SQL query String: " + query);

        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            ResultSet rs = psm.executeQuery();
            logger.debug("Query execution done");
            int rowData = 0;
            while (rs.next()) {

```

```

        int grnNoo = rs.getInt("grn_no");

        rowData = grnNoo;

        logger.debug("return rowData");

        return rowData;

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return 0");

return 0;

}

public String saveGrn(String grnNo, Date grnDate, int supplierNo, String itemCount, String totalDiscount, String
totalVat, String totalAmount) {

    String query = "INSERT INTO grn(grn_no, grn_date, supplier_no, grn_item_count, grn_discount, grn_vat,
grn_amount)"

        + "VALUE(?,?,?,?,?,?)";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, grnNo);

    psm.setDate(2, new java.sql.Date(grnDate.getTime()));

    psm.setInt(3, supplierNo);

    psm.setString(4, itemCount);

    psm.setString(5, totalDiscount);

    psm.setString(6, totalVat);

    psm.setString(7, totalAmount);

    psm.execute();

    logger.debug("Query execution done");

    logger.debug("return result");
}

```

```

        return "Success";

    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public String saveGrnItem(int productNo, double unitSellPrice, double qty, double unitBuyPrice, double discount,
double vat, double amount, String grnNo) {

    String query1 = "INSERT INTO grn_item(product_no, grn_item_quantity, grn_item_unit_buy_price,
grn_item_discount, grn_item_vat, grn_item_amount, grn_no) VALUE(?,?,?,?,?,?)";
    String query2 = "SELECT * FROM product WHERE product_no = ?";
    String query3 = "UPDATE product SET product_quantity = ?, product_unit_sell_price = ? WHERE product_no =
?";

    logger.debug("SQL query String: " + query1);
    logger.debug("SQL query String: " + query2);
    logger.debug("SQL query String: " + query3);

try {
    PreparedStatement psm = dbConnection.prepareStatement(query1);
    psm.setInt(1, productNo);
    psm.setDouble(2, qty);
    psm.setDouble(3, unitBuyPrice);
    psm.setDouble(4, discount);
    psm.setDouble(5, vat);
    psm.setDouble(6, amount);
    psm.setString(7, grnNo);
    psm.execute();
    logger.debug("Query1 execution done");
}

```

```

PreparedStatement psm2 = dbConnection.prepareStatement(query2);
psm2.setInt(1, productNo);

ResultSet rs = psm2.executeQuery();
logger.debug("Query2 execution done");

double currentQty = qty;

while (rs.next()) {

    currentQty = rs.getDouble("product_quantity") + currentQty;

}

PreparedStatement psm3 = dbConnection.prepareStatement(query3);
psm3.setDouble(1, currentQty);
psm3.setDouble(2, unitSellPrice);
psm3.setInt(3, productNo);
psm3.execute();

logger.debug("Query3 execution done");
logger.debug("return result");

return "Success";

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
    logger.error("return result");

    return "Error !";
}

}

public void loadGrnTableFull(String keywordFull, DefaultTableModel grnTableModelFull) {

String query = "SELECT * FROM grn INNER JOIN supplier "
+ "ON grn.supplier_no = supplier.supplier_no "
+ "WHERE grn_no LIKE ? OR grn_date LIKE ? OR supplier_name LIKE ? ORDER BY grn_no";

logger.debug("SQL query String: " + query);

try {

```

```

PreparedStatement psm = dbConnection.prepareStatement(query);

psm.setString(1, "%" + keywordFull + "%");
psm.setString(2, "%" + keywordFull + "%");
psm.setString(3, "%" + keywordFull + "%");

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

grnTableModelFull.setRowCount(0);

while (rs.next()) {

    int grnNo = rs.getInt("grn_no");

    Date date = rs.getDate("grn_date");

    String supplierName = rs.getString("supplier_name");

    int itemCount = rs.getInt("grn_item_count");

    double totalDiscount = rs.getDouble("grn_discount");

    double totalVat = rs.getDouble("grn_vat");

    double totalAmount = rs.getDouble("grn_amount");

    rowData = new Object[]{grnNo, date, supplierName, itemCount, totalDiscount, totalVat, totalAmount};

    grnTableModelFull.addRow(rowData);

    logger.debug("grnTableModelFull [addRow]");

}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

public void loadGrnItemTableFull(String keywordFull1, DefaultTableModel grnItemTableModelFull) {

String query = "SELECT * FROM grn_item INNER JOIN grn INNER JOIN product INNER JOIN unit "

+ "ON grn_item.grn_no = grn.grn_no AND grn_item.product_no = product.product_no AND
product.unit_no = unit.unit_no "

+ "WHERE grn_item_no LIKE ? OR grn_item.grn_no LIKE ? OR grn_date LIKE ? OR grn_item.product_no LIKE
? OR product_name LIKE ? ORDER BY grn_item_no";

```

```

logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keywordFull1 + "%");

    psm.setString(2, "%" + keywordFull1 + "%");

    psm.setString(3, "%" + keywordFull1 + "%");

    psm.setString(4, "%" + keywordFull1 + "%");

    psm.setString(5, "%" + keywordFull1 + "%");

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    grnItemTableModelFull.setRowCount(0);

    while (rs.next()) {

        int grnItemNo = rs.getInt("grn_item_no");

        Date buydate = rs.getDate("grn_date");

        int productNo = rs.getInt("product_no");

        String productName = rs.getString("product_name");

        double qty = rs.getDouble("grn_item_quantity");

        String unitName = rs.getString("unit_name");

        double unitBuyPrice = rs.getDouble("grn_item_unit_buy_price");

        double discount = rs.getDouble("grn_item_discount");

        double vat = rs.getDouble("grn_item_vat");

        double amount = rs.getDouble("grn_item_amount");

        int grnNo = rs.getInt("grn_no");

        rowData = new Object[]{grnItemNo, buydate, productNo, productName, qty, unitName, unitBuyPrice,
discount, vat, amount, grnNo};

        grnItemTableModelFull.addRow(rowData);

        logger.debug("grnItemTableModelFull [addRow]");

    }

} catch (Exception e) {

```

```
e.printStackTrace();  
logger.error("Exception: " + e.getMessage());  
}  
}  
  
}
```

Batch Name : Griffins

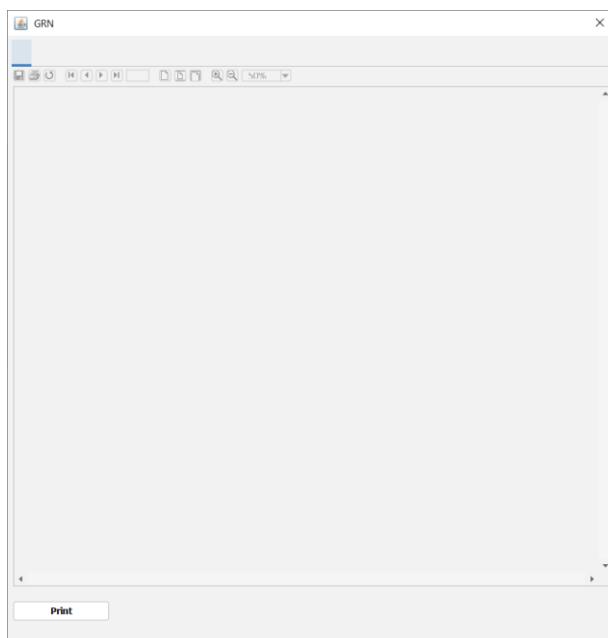
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate GRN Report

GUI Number : 07

Use case Name: GRN Management



01. Brief Description of the GUI:

The main objective of this interface is to generate, display and print GRN reports.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------------|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display GRN report | The GRN report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated GRN report | A button is required to print the generated GRN report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateGrnDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```

*/
public class GenerateGrnDialog extends javax.swing.JDialog {

    /**
     * Creates new form GenerateGrnDialog
     */
    static Logger logger = Logger.getLogger(GenerateGrnDialog.class.getName());

    String reportPath = "C:\\\\Users\\\\EZIO
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila_hardware_grn.jasper";

    Map m = new HashMap();
    int grnNumber;

    public GenerateGrnDialog(java.awt.Frame parent, boolean modal, int grnNo) {
        super(parent, modal);
        initComponents();
        logger.debug("Access to GenerateGrnDialog.java");

        grnNumber = grnNo;
        m.put("grn_no", grnNumber);

        try {
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
            JRViewer jv = new JRViewer(jp);
            jv.setZoomRatio((float) 0.5);
            jTabbedPane1.add(jv);
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("Exception: " + e.getMessage());
        }
    }
}

```

```
/**  
  
 * 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();  
  
    jTabbedPane1 = new javax.swing.JTabbedPane();  
  
    printButton = new javax.swing.JButton();  
  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
  
    setTitle("GRN");  
  
    setResizable(false);  
  
  
    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
  
    printButton.setText("Print");  
  
    printButton.addActionListener(new java.awt.event.ActionListener() {  
  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
            printButtonActionPerformed(evt);  
  
        }  
  
    });  
  
  
    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(0, 0, 0)  
            .addGroup(jPanel1Layout.createSequentialGroup()  
                .addGap(0, 0, 0)  
                .addContainerGap()  
            )  
        )  
    );  
}  
  
}
```

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

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel1Layout.createSequentialGroup()

        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap()

);

jPanel1Layout.setVerticalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addContainerGap(32, Short.MAX_VALUE))

);

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)

);

layout.setVerticalGroup

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

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

);

};
```

```

    pack();

    setLocationRelativeTo(null);

}// </editor-fold>

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

    // TODO add your handling code here:

    m.put("grn_no", grnNumber);

    try {

        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());

        JasperPrintManager.printReport(jp, true);

        logger.debug("Print Report");

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

    }

}

/** 

 * @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 {

        UIManager.setLookAndFeel(new FlatIntelliJLaf());

        UIManager.put("Table.showHorizontalLines", true);

        UIManager.put("Table.showVerticalLines", true);

    }

```



```
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        int grnNo = 0;
        GenerateGrnDialog dialog = new GenerateGrnDialog(new javax.swing.JFrame(), true, grnNo);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});

// Variables declaration - do not modify
private javax.swing.JPanel jPanel1;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JButton printButton;
// End of variables declaration
}
```

Batch Name : Griffins

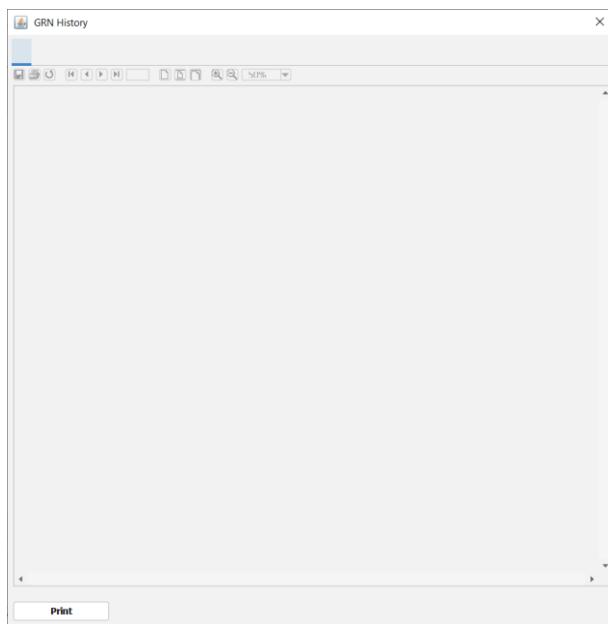
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate GRN History Report

GUI Number : 08

Use case Name: GRN Management



01. Brief Description of the GUI:

The main objective of this interface is to generate, display and print the GRN history report.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|------------------------------------|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display the GRN history report | The GRN history report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated GRN history report | A button is required to print the generated GRN history report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateGrnHistoryDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GenerateGrnHistoryDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form GenerateGrnHistoryDialog  
     */  
  
    static Logger logger = Logger.getLogger(GenerateGrnHistoryDialog.class.getName());  
  
    String reportPath = "C:\\\\Users\\\\EZIO  
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila.hardware_grn_history.jasper";  
  
    Map m = new HashMap();  
  
    public GenerateGrnHistoryDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to GenerateGrnHistoryDialog.java");  
  
        try {  
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());  
            JRViewer jv = new JRViewer(jp);  
            jv.setZoomRatio((float) 0.5);  
            jTabbedPane1.addTab(jv);  
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
    }  
  
    /**  
     * 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();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    printButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("GRN History");
    setResizable(false);

    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    printButton.setText("Print");
    printButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            printButtonActionPerformed(evt);
        }
    });
});

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(0, 0, 0)
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
);
```

```
.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

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)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
```

```
//</editor-fold>

private void printButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
        JasperPrintManager.printReport(jp, true);
        logger.debug("Print Report");
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
    }
}

/**
 * @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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
}
```

```
}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//         if ("Windows".equals(info.getName())) {

//             javax.swing.UIManager.setLookAndFeel(info.getClassName());

//             break;

//         }

//     }

// } catch (ClassNotFoundException ex) {

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

// } catch (InstantiationException ex) {

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

// } catch (IllegalAccessException ex) {

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

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

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

// }

//</editor-fold>

//</editor-fold>

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        GenerateGrnHistoryDialog dialog = new GenerateGrnHistoryDialog(new javax.swing.JFrame(), true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override
```

```
public void windowClosing(java.awt.event.WindowEvent e) {  
    System.exit(0);  
}  
});  
dialog.setVisible(true);  
}  
});  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel jPanel1;  
private javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JButton printButton;  
// End of variables declaration  
}
```

Batch Name : Griffins

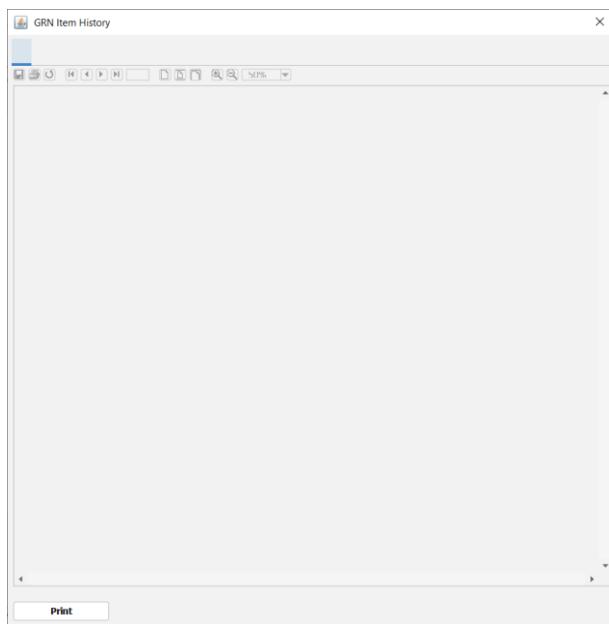
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate GRN Item History Report

GUI Number : 09

Use case Name: GRN Management



01. Brief Description of the GUI:

The main objective of this interface is to generate, display and print the GRN item history report.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display the GRN item history report | The GRN item history report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated GRN item history report | A button is required to print the generated GRN item history report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateGrnItemHistoryDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```

*/
public class GenerateGrnItemHistoryDialog extends javax.swing.JDialog {

    /**
     * Creates new form GenerateGrnItemHistoryDialog
     */
    static Logger logger = Logger.getLogger(GenerateGrnItemHistoryDialog.class.getName());

    String reportPath = "C:\\\\Users\\\\EZIO
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila.hardware_grn_item_history.jasper";
    Map m = new HashMap();

    public GenerateGrnItemHistoryDialog(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        logger.debug("Access to GenerateGrnItemHistoryDialog.java");

        try {
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
            JRViewer jv = new JRViewer(jp);
            jv.setZoomRatio((float) 0.5);
            jTabbedPane1.addTab(jv);
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("Exception: " + e.getMessage());
        }
    }

    /**
     * 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();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    printButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("GRN Item History");
    setResizable(false);

    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    printButton.setText("Print");
    printButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            printButtonActionPerformed(evt);
        }
    });
});

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(0, 0, 0)
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
);
```

```
.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

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)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
```

```
//</editor-fold>

private void printButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
        JasperPrintManager.printReport(jp, true);
        logger.debug("Print Report");
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
    }
}

/**
 * @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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
}
```

```

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(GenerateGrnItemHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(GenerateGrnItemHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(GenerateGrnItemHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(GenerateGrnItemHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// }
//</editor-fold>
```

```

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        GenerateGrnItemHistoryDialog dialog = new GenerateGrnItemHistoryDialog(new javax.swing.JFrame(),
true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {
```

```
        System.exit(0);

    }

});  
dialog.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JButton printButton;

// End of variables declaration

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Sales Manager Create Sale

GUI Number : 10

Use case Name: Sales Management

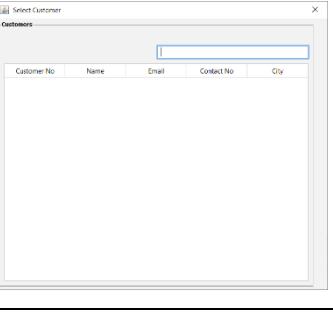
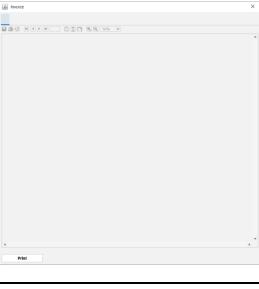
03. Brief Description of the GUI:

The main objective of this interface is to create sales, save sales and generate invoice reports.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|--------------|---|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Sales Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanels makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Insert product details | One sale can have multiple products. There should be space to add those products. That space is filled by this JTable (one product per table row can be inserted.) |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert sales details | To enter or display the details needed to create a sale. |
| JDateChooser | Display date | To display the current date |
| JButtons | Add table row, Delete table row, Save sales, Clear text fields & Generate invoice reports | Buttons are needed to add rows to a table, delete rows from a table, save a sale, clear text fields and generate invoice reports. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|--|--|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To enter the details of the customer who is buying the products. |
| | 02.  | To include details of bought products. |
| | 03.  | To Generate invoice reports |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

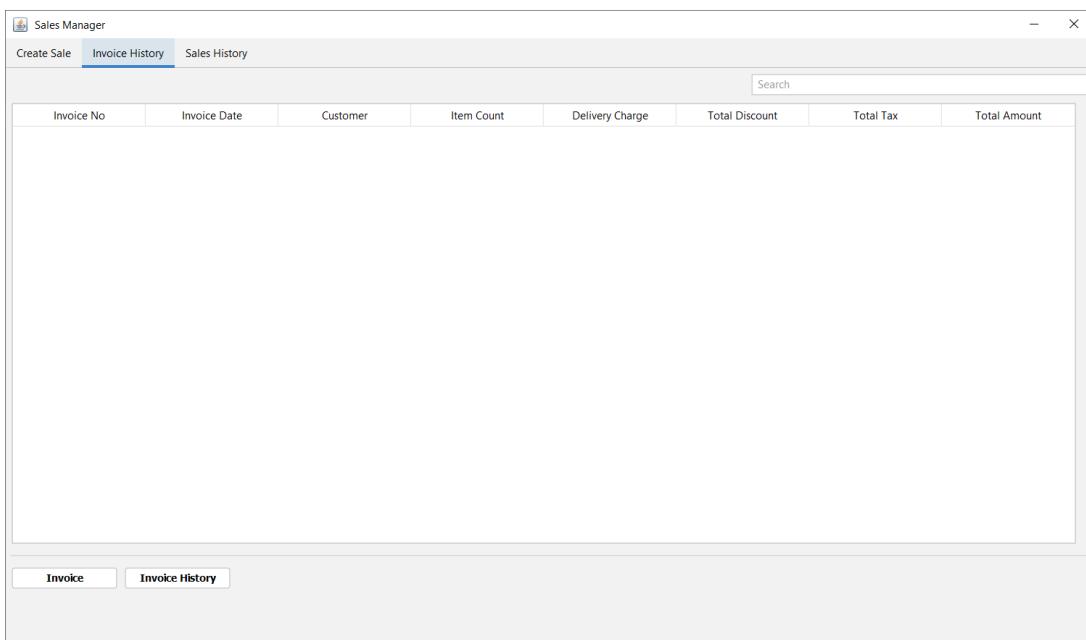
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Sales Manager Invoice History

GUI Number : 11

Use case Name: Sales Management



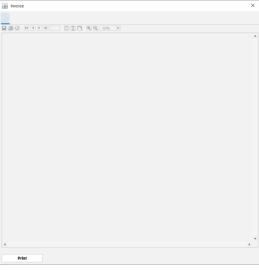
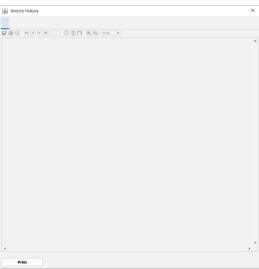
03. Brief Description of the GUI:

The main objective of this interface is to display the invoice history, search for the invoice, select an invoice and generate its invoice report and generate the invoice history report.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Sales Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display the invoice history | During a search operation, the user can find nearly similar invoice lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |
| JButtons | Generate invoice reports & the invoice history report | Buttons are needed to generate an invoice selected from the table and to generate the invoice history. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---|--|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To generate an invoice selected from the table |
| | 02.  | To Generate the invoice history report |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

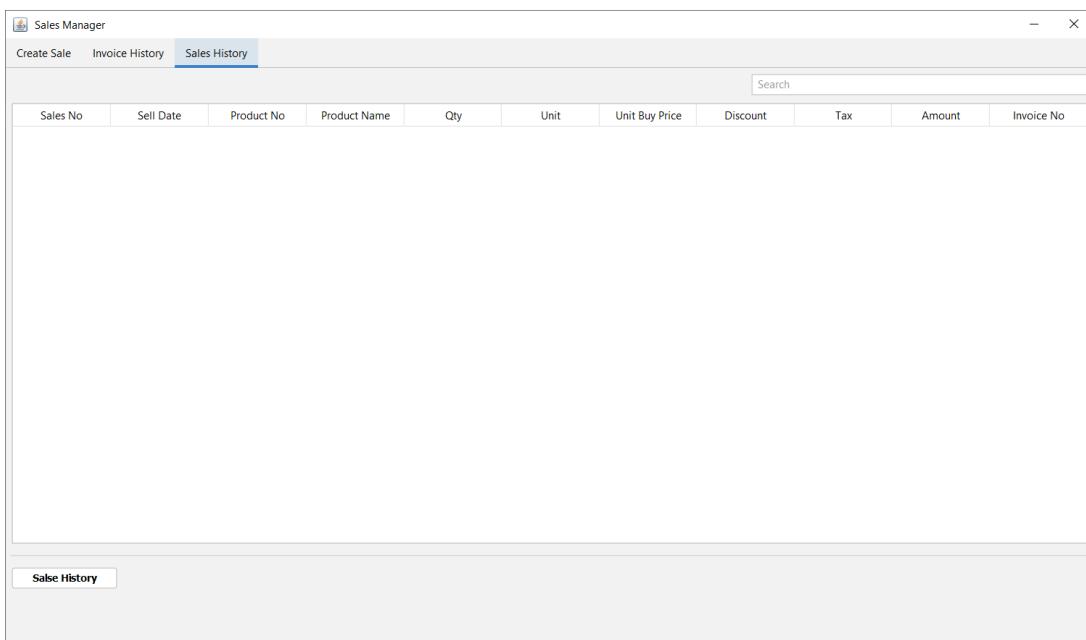
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Sales Manager Sales History

GUI Number : 12

Use case Name: Sales Management



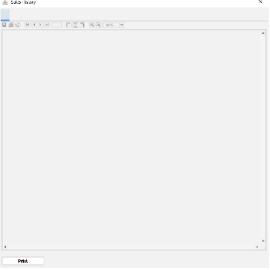
03. Brief Description of the GUI:

The main objective of this interface is to display the sales history, search for the sales and generate the sales history report.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|-----------------------------------|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Sales Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display the sales history | During a search operation, the user can find nearly similar sales lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |
| JButtons | Generate the sales history report | A button is required to generate the sales history. This JButton meets that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|--|--------------------------------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To Generate the sales history report |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



SalesManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.SalesModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

```
*/  
  
public class SalesManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form SalesManager  
     */  
  
    private SalesModel salesModel;  
  
    private LoadCustomerDialog loadCustomerDialog;  
  
    private LoadProductDialog loadProductDialog;  
  
    static Logger logger = Logger.getLogger(SalesManager.class.getName());  
  
  
    DefaultTableModel invoiceTableModel;  
  
    DefaultTableModel invoiceTableModelFull;  
  
    DefaultTableModel salesTableModelFull;  
  
  
    int customerNo;  
  
  
    public SalesManager() {  
        initComponents();  
  
        logger.debug("Access to SalesManager.java");  
  
  
        invoiceTableModel = (DefaultTableModel) invoiceTable.getModel();  
  
        invoiceTableModelFull = (DefaultTableModel) invoiceTableFull.getModel();  
  
        salesTableModelFull = (DefaultTableModel) salesTableFull.getModel();  
  
  
        salesModel = new SalesModel();  
  
        loadCustomerDialog = new LoadCustomerDialog(this, rootPaneCheckingEnabled);  
  
        loadProductDialog = new LoadProductDialog(this, rootPaneCheckingEnabled);  
  
  
        salesModel.loadInvoiceTableFull("", invoiceTableModelFull);  
  
        salesModel.loadSalesTableFull("", salesTableModelFull);
```

```
invoiceNoLoad();

invoiceDateChooser.setDate(new java.util.Date());

}

/**

 * 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() {

    jTabbedPane1 = new javax.swing.JTabbedPane();

    jPanel1 = new javax.swing.JPanel();

    jPanel4 = new javax.swing.JPanel();

    invoiceNoField = new javax.swing.JTextField();

    invoiceDateChooser = new com.toedter.calendar.JDateChooser();

    customerNameField = new javax.swing.JTextField();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jScrollPane1 = new javax.swing.JScrollPane();

    invoiceTable = new javax.swing.JTable();

    jPanel5 = new javax.swing.JPanel();

    discountField = new javax.swing.JTextField();

    totalAmountField = new javax.swing.JTextField();

    netAmountField = new javax.swing.JTextField();

    taxField = new javax.swing.JTextField();
}
```

```
itemCountField = new javax.swing.JTextField();

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();

jLabel9 = new javax.swing.JLabel();

deliveryChargeField = new javax.swing.JTextField();

jSeparator1 = new javax.swing.JSeparator();

jPanel6 = new javax.swing.JPanel();

addRowButton = new javax.swing.JButton();

deleteRowButton = new javax.swing.JButton();

saveInvoiceButton = new javax.swing.JButton();

newGrnButton = new javax.swing.JButton();

invoiceButton = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

invoiceTableFull = new javax.swing.JTable();

jSeparator2 = new javax.swing.JSeparator();

invoiceButton1 = new javax.swing.JButton();

invoiceHistoryButton = new javax.swing.JButton();

jPanel3 = new javax.swing.JPanel();

keywordFieldFull1 = new javax.swing.JTextField();

jScrollPane4 = new javax.swing.JScrollPane();

salesTableFull = new javax.swing.JTable();

jSeparator3 = new javax.swing.JSeparator();

salesHistoryButton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Sales Manager");
```

```
setResizable(false);

invoiceNoField.setEnabled(false);

customerNameField.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        customerNameFieldMouseClicked(evt);

    }

});

jLabel1.setText("Invoice No");

jLabel2.setText("Customer");

jLabel3.setText("Invoice Date");

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel4Layout.createSequentialGroup()

        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 90, Short.MAX_VALUE))

        .addGap(18, 18, 18)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addComponent(invoiceNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 229, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(84, 84, 84)


```

```

        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(invoiceDateChooser, javax.swing.GroupLayout.DEFAULT_SIZE, 277,
Short.MAX_VALUE))

        .addComponent(customerNameField))

        .addGap(654, 654, 654))

);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(invoiceNoField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

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

            .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(customerNameField, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE))

        .addContainerGap())
    );
}

invoiceTable.setFont(new java.awt.Font("Tahoma", 0, 15)); // NOI18N

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

},

},
```

```
new String [] {  
    "", "Product", "Qty", "Unit", "Unit Sell Price", "Discount", "Tax", "Amount"  
}  
){  
  
Class[] types = new Class [] {  
  
    java.lang.Object.class, java.lang.String.class, java.lang.Double.class, java.lang.String.class,  
    java.lang.Double.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class  
  
};  
  
boolean[] canEdit = new boolean [] {  
  
    false, false, true, false, true, true, false, false  
  
};  
  
  
public Class getColumnClass(int columnIndex) {  
  
    return types [columnIndex];  
  
}  
  
  
public boolean isCellEditable(int rowIndex, int columnIndex) {  
  
    return canEdit [columnIndex];  
  
}  
});  
  
invoiceTable.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));  
  
invoiceTable.setDragEnabled(true);  
  
invoiceTable.setRowHeight(30);  
  
invoiceTable.setSelectionBackground(new java.awt.Color(204, 204, 204));  
  
invoiceTable.addMouseListener(new java.awt.event.MouseAdapter() {  
  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
  
        invoiceTableMouseClicked(evt);  
  
    }  
});  
  
invoiceTable.addKeyListener(new java.awt.event.KeyAdapter() {  
  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
  
    }  
});
```

```
invoiceTableKeyReleased(evt);

}

});

jScrollPane1.setViewportView(invoiceTable);

if (invoiceTable.getColumnModel().getColumnCount() > 0) {

    invoiceTable.getColumnModel().getColumn(0).setMinWidth(25);

    invoiceTable.getColumnModel().getColumn(0).setMaxWidth(25);

}

discountField.setText("0.00");

discountField.setEnabled(false);

totalAmountField.setText("0.00");

totalAmountField.setEnabled(false);

netAmountField.setText("0.00");

netAmountField.setEnabled(false);

taxField.setText("0.00");

taxField.setEnabled(false);

itemCountField.setText("0");

itemCountField.setEnabled(false);

jLabel4.setText("Item Count");

jLabel5.setText("Discount");

jLabel6.setText("Tax");

jLabel7.setText("Total Amount");
```

```
jLabel8.setText("Net Amount");

jLabel9.setText("Delivery Charge");

deliveryChargeField.setText("0.00");
deliveryChargeField.setEnabled(false);
deliveryChargeField.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        deliveryChargeFieldKeyPressed(evt);
    }
    public void keyReleased(java.awt.event.KeyEvent evt) {
        deliveryChargeFieldKeyReleased(evt);
    }
});
javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
jPanel5Layout.setHorizontalGroup(
    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel5Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jLabel5, javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel4, javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
        .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(taxField, javax.swing.GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addGroup(jPanel5Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE)
                    .addComponent(jLabel9, javax.swing.GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE)))
            .addGroup(jPanel5Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(deliveryChargeField, javax.swing.GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE))))));
```

```
.addComponent(discountField, javax.swing.GroupLayout.DEFAULT_SIZE, 194, Short.MAX_VALUE)

.addComponent(itemCountField))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 65, Short.MAX_VALUE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(totalAmountField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 194, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(netAmountField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 194, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel5Layout.createSequentialGroup()

.addComponent(jLabel9, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)

.addGap(18, 18, 18)

.addComponent(deliveryChargeField, javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap())

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(deliveryChargeField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addComponent(itemCountField))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(totalAmountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(discountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(taxField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(netAmountField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

);

addRowButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

addRowButton.setText("Add Row");

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

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

        addRowButtonActionPerformed(evt);

    }

});

deleteRowButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
deleteRowButton.setText("Delete Row");
deleteRowButton.setEnabled(false);
deleteRowButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        deleteRowButtonActionPerformed(evt);
    }
});
```

```
saveInvoiceButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
saveInvoiceButton.setText("Save Invoice");
saveInvoiceButton.setEnabled(false);
saveInvoiceButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        saveInvoiceButtonActionPerformed(evt);
    }
});
```

```
newGrnButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newGrnButton.setText("New");
newGrnButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newGrnButtonActionPerformed(evt);
    }
});
```

```
invoiceButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
invoiceButton.setText("Invoice");
invoiceButton.setEnabled(false);
invoiceButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        invoiceButtonActionPerformed(evt);
    }
});
```

```
    }

});

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
jPanel6.setLayout(jPanel6Layout);
jPanel6Layout.setHorizontalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(addRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(deleteRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(saveInvoiceButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(newGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(invoiceButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(Short.MAX_VALUE)
    )
);

jPanel6Layout.setVerticalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(addRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(deleteRowButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        )
    )
);
```

```
.addComponent(saveInvoiceButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(newGrnButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(invoiceButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jScrollPane1)
.addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGap(18, 18, 18)
.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jSeparator1))
.addContainerGap())
);

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

```

```
.addContainerGap()

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 439,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap(48, Short.MAX_VALUE))

);

jTabbedPane1.addTab("Create Sale", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {
```

```

        keywordFieldFullKeyReleased(evt);

    }

});
```

invoiceTableFull.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
 "Invoice No", "Invoice Date", "Customer", "Item Count", "Delivery Charge", "Total Discount", "Total Tax",
 "Total Amount"
}
) {
 boolean[] canEdit = new boolean [] {
 false, false, false, false, true, false, false, false
 };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
 return canEdit [columnIndex];
}

});
invoiceTableFull.setRowHeight(25);
jScrollPane3.setViewportView(invoiceTableFull);

invoiceButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
invoiceButton1.setText("Invoice");
invoiceButton1.addActionListener(new java.awt.event.ActionListener() {
 public void actionPerformed(java.awt.event.ActionEvent evt) {
 invoiceButton1ActionPerformed(evt);
 }
});

```
invoiceHistoryButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
  
invoiceHistoryButton.setText("Invoice History");  
  
invoiceHistoryButton.setOpaque(false);  
  
invoiceHistoryButton.addActionListener(new java.awt.event.ActionListener() {  
  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
        invoiceHistoryButtonActionPerformed(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()  
  
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addGroup(jPanel2Layout.createSequentialGroup()  
  
.addGap(0, 0, Short.MAX_VALUE)  
  
.addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
  
.addComponent(jSeparator2)  
  
.addGroup(jPanel2Layout.createSequentialGroup()  
  
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 1472,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addGroup(jPanel2Layout.createSequentialGroup()  
  
.addComponent(invoiceButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 150,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(invoiceHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,  
javax.swing.GroupLayout.PREFERRED_SIZE)))  
  
.addGap(0, 0, Short.MAX_VALUE)))
```

```

    .addContainerGap())
};

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 613,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(invoiceButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(invoiceHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(76, Short.MAX_VALUE))
);

jTabbedPane1.addTab("Invoice History", jPanel2);

keywordFieldFull1.setForeground(new java.awt.Color(153, 153, 153));
keywordFieldFull1.setText("Search");
keywordFieldFull1.setToolTipText("Search");
keywordFieldFull1.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        keywordFieldFull1FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {

```

```

        keywordFieldFull1FocusLost(evt);

    }

});

keywordFieldFull1.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFull1KeyReleased(evt);

    }

});

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

        new String [] {
            "Sales No", "Sell Date", "Product No", "Product Name", "Qty", "Unit", "Unit Buy Price", "Discount", "Tax",
            "Amount", "Invoice No"
        }
    },
    new String [] {
        ""
    }
)) {
    boolean[] canEdit = new boolean [] {
        false, true, false, false, false, false, false, false, true, true
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

salesTableFull.setRowHeight(25);

jScrollPane4.setViewportView(salesTableFull);

salesHistoryButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
salesHistoryButton.setText("Salse History");

```

```
salesHistoryButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        salesHistoryButtonActionPerformed(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()  
        .addContainerGap()  
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jScrollPane4)  
            .addGroup(jPanel3Layout.createSequentialGroup()  
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addComponent(keywordFieldFull1, javax.swing.GroupLayout.PREFERRED_SIZE, 469,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
                .addComponent(jSeparator3)  
                .addGroup(jPanel3Layout.createSequentialGroup()  
                    .addComponent(salesHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,  
                        javax.swing.GroupLayout.PREFERRED_SIZE)  
                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                        .addGroup(jPanel3Layout.createSequentialGroup()  
                            .addGap(1322, 1322, 1322)))  
                    .addContainerGap())  
                );  
    jPanel3Layout.setVerticalGroup(  
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel3Layout.createSequentialGroup()  
            .addContainerGap()  
            .addComponent(keywordFieldFull1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    );  
});
```

```
.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 613,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(salesHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(76, Short.MAX_VALUE))

);

jTabbedPane1.addTab("Sales History", jPanel3);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1501,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE)
    )
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 835,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE)
    )
);

pack();

 setLocationRelativeTo(null);

}// </editor-fold>
```

```
private void customerNameFieldMouseClicked(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    loadCustomerDialog.setVisible(true);  
  
  
    Object[] customerData = loadCustomerDialog.getCustomerData();  
  
    customerNo = (int) customerData[0];  
  
    customerNameField.setText((String) customerData[1]);  
  
    //System.out.println(customerNo);  
  
}
```

```
private void invoiceTableMouseClicked(java.awt.event.MouseEvent evt) {  
  
    // TODO add your handling code here:  
  
    int selectedRow = invoiceTable.getSelectedRow();  
  
    int selectedColumn = invoiceTable.getSelectedColumn();  
  
  
    if (selectedColumn == 1) {  
  
        loadProductDialog.setVisible(true);  
  
  
        Object[] productData = loadProductDialog.getProductData();  
  
  
        Object productNoo = productData[0];  
  
        String productName = (String) productData[1];  
  
        Object productQty = productData[2];  
  
        //int productUnitNo = (int) productData[3];  
  
        Object productUnitName = productData[4];  
  
        Object productUnitSellPrice = productData[5];  
  
        Object productDiscount = productData[6];  
  
        Object productTax = productData[7];  
  
  
        invoiceTableModel.setValueAt(productNoo, selectedRow, 0);  
    }  
}
```

```
invoiceTableModel.setValueAt(productName, selectedRow, 1);

invoiceTableModel.setValueAt(productQty, selectedRow, 2);

invoiceTableModel.setValueAt(productUnitName, selectedRow, 3);

invoiceTableModel.setValueAt(productUnitSellPrice, selectedRow, 4);

invoiceTableModel.setValueAt(productDiscount, selectedRow, 5);

invoiceTableModel.setValueAt(productTax, selectedRow, 6);

invoiceTableModel.setValueAt(0.00, selectedRow, 7);

} else {

    if ((int) invoiceTableModel.getValueAt(selectedRow, 0) == 0 || invoiceTableModel.getValueAt(selectedRow, 1)
== null) {

        loadProductDialog.setVisible(true);

        Object[] productData = loadProductDialog.getProductData();

        Object productNoo = productData[0];

        String productName = (String) productData[1];

        Object productQty = productData[2];

        //int productUnitNo = (int) productData[3];

        Object productUnitName = productData[4];

        Object productUnitSellPrice = productData[5];

        Object productDiscount = productData[6];

        Object productTax = productData[7];

        invoiceTableModel.setValueAt(productNoo, selectedRow, 0);

        invoiceTableModel.setValueAt(productName, selectedRow, 1);

        invoiceTableModel.setValueAt(productQty, selectedRow, 2);

        invoiceTableModel.setValueAt(productUnitName, selectedRow, 3);

        invoiceTableModel.setValueAt(productUnitSellPrice, selectedRow, 4);

        invoiceTableModel.setValueAt(productDiscount, selectedRow, 5);

    }

}
```

```
        invoiceTableModel.setValueAt(productTax, selectedRow, 6);

        invoiceTableModel.setValueAt(0.00, selectedRow, 7);

    }

}

calculateValues();

}
```

```
private void invoiceTableKeyReleased(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    calculateValues();

}
```

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

    // TODO add your handling code here:

    if (customerNo != 0 && customerNameField.getText().length() > 0) {

        invoiceTableModel.addRow(new Object[]{0, null, 0.00, "Number", 0.00, 0.00, 0.00, 0.00});

        deleteRowButton.setEnabled(true);

        saveInvoiceButton.setEnabled(true);

        deliveryChargeField.setEnabled(true);

        itemCountField.setText(String.valueOf(invoiceTableModel.getRowCount()));

    } else {

        loadCustomerDialog.setVisible(true);

        Object[] customerData = loadCustomerDialog.getCustomerData();

        customerNo = (int) customerData[0];

        customerNameField.setText((String) customerData[1]);

    }

}
```

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

    // TODO add your handling code here:

    int rowCount = invoiceTable.getRowCount();
```

```

int selectedRow = invoiceTable.getSelectedRow();

int selectedRowCount = invoiceTable.getSelectedRowCount();

if (selectedRowCount == 1) {

    invoiceTableModel.removeRow(selectedRow);

    if(rowCount == 1){

        deleteRowButton.setEnabled(false);

        saveInvoiceButton.setEnabled(false);

    }

} else {

    //if (rowCount == 0) {

        // JOptionPane.showConfirmDialog(rootPane, "Table is Empty !", "Error", -1,
        //JOptionPane.ERROR_MESSAGE);

    //} else {

        JOptionPane.showConfirmDialog(rootPane, "Please Select any Single Row !", "Error", -1,
        JOptionPane.ERROR_MESSAGE);

        logger.error("Error: Please Select any Single Row !");

    //}

}

}

}

```

```

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

    // TODO add your handling code here:

    int rowCount = invoiceTableModel.getRowCount();

    boolean checkEmptyRows = false;

    for (int i = 0; i < rowCount; i++) {

        int productNo = (int) invoiceTableModel.getValueAt(i, 0);

        String productName = (String) invoiceTableModel.getValueAt(i, 1);

        if (productNo == 0 || productName == null) {

            checkEmptyRows = true;

            break;

```

```

}

}

if (checkEmptyRows) {
    JOptionPane.showConfirmDialog(rootPane, "There is some Empty Rows !", "Error", -1,
JOptionPane.ERROR_MESSAGE);
    logger.error("Error: There is some Empty Rows !");
} else {

    String invoiceNo = invoiceNoField.getText();
    Date invoiceDate = invoiceDateChooser.getDate();
    //customerNo in hear
    String itemCount = itemCountField.getText();
    String deliveryCharge = deliveryChargeField.getText();
    String totalDiscount = discountField.getText();
    String totalTax = taxField.getText();
    String totalAmount = totalAmountField.getText();

    String invoiceResult = salesModel.saveInvoice(invoiceNo, invoiceDate, customerNo, itemCount,
deliveryCharge, totalDiscount, totalTax, totalAmount);

    String salesResult = null;
    for (int i = 0; i < rowCount; i++) {
        int productNo = (int) invoiceTableModel.getValueAt(i, 0);
        double qty = (double) invoiceTableModel.getValueAt(i, 2);
        double unitSellPrice = (double) invoiceTableModel.getValueAt(i, 4);
        double discount = (double) invoiceTableModel.getValueAt(i, 5);
        double tax = (double) invoiceTableModel.getValueAt(i, 6);
        double amount = (double) invoiceTableModel.getValueAt(i, 7);

        salesResult = salesModel.saveSales(productNo, qty, unitSellPrice, discount, tax, amount, invoiceNo);
    }
}

```

```
if (invoiceResult.equals("Success") && salesResult.equals("Success")) {  
    JOptionPane.showConfirmDialog(rootPane, salesResult, "Message", -1, 1);  
    invoiceButton.setEnabled(true);  
    addRowButton.setEnabled(false);  
    deleteRowButton.setEnabled(false);  
    saveInvoiceButton.setEnabled(false);  
    salesModel.loadInvoiceTableFull("", invoiceTableModelFull);  
    salesModel.loadSalesTableFull("", salesTableModelFull);  
    logger.debug("Message: " + salesResult);  
}  
} else {  
    JOptionPane.showConfirmDialog(rootPane, salesResult, "Error", -1, JOptionPane.ERROR_MESSAGE);  
    logger.error("Error: " + salesResult);  
}  
}  
}  
}
```

```
private void newGrnButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int rowCount = invoiceTableModel.getRowCount();  
  
    invoiceNoLoad();  
    customerNo = 0;  
    customerNameField.setText("");  
    invoiceDateChooser.setDate(new java.util.Date());  
  
    itemCountField.setText("0");  
    deliveryChargeField.setText("0.00");  
    discountField.setText("0.00");  
    taxField.setText("0.00");
```

```
totalAmountField.setText("0.00");

netAmountField.setText("0.00");

addRowButton.setEnabled(true);

deleteRowButton.setEnabled(false);

saveInvoiceButton.setEnabled(false);

invoiceButton.setEnabled(false);

deliveryChargeField.setEnabled(false);

for (int i = rowCount - 1; i >= 0; i--) {
    invoiceTableModel.removeRow(i);
}

logger.debug("clearFields");
}

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

    // TODO add your handling code here:

    int invoiceNo = Integer.parseInt(invoiceNoField.getText());

    new GenerateInvoiceDialog(this, true, invoiceNo).setVisible(true);

}

private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {

    if (keywordFieldFull.getText().equals("Search")) {

        keywordFieldFull.setText("");
        keywordFieldFull.setForeground(new Color(0, 0, 0));

    }

}

private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

}
```

```

if (keywordFieldFull.getText().equals("")) {
    keywordFieldFull.setText("Search");
    keywordFieldFull.setForeground(new Color(153, 153, 153));
}
}

private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String keywordFull = keywordFieldFull.getText();
    salesModel.loadInvoiceTableFull(keywordFull, invoiceTableModelFull);
}

private void invoiceButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int rowCount = invoiceTableFull.getRowCount();
    int selectedRow = invoiceTableFull.getSelectedRow();
    int selectedRowCount = invoiceTableFull.getSelectedRowCount();

    if (selectedRowCount == 1) {
        int invoiceNo = (int) invoiceTableModelFull.getValueAt(selectedRow, 0);
        new GenerateInvoiceDialog(this, true, invoiceNo).setVisible(true);
    } else {
        if (rowCount == 0) {
            JOptionPane.showConfirmDialog(rootPane, "Table is Empty !", "Error", -1, JOptionPane.ERROR_MESSAGE);
            logger.error("Error: Table is Empty !");
        } else {
            JOptionPane.showConfirmDialog(rootPane, "Please Select any Single Row !", "Error", -1,
JOptionPane.ERROR_MESSAGE);
            logger.error("Error: Please Select any Single Row !");
        }
    }
}

```

```
}
```

```
private void invoiceHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    new GenerateInvoiceHistoryDialog(this, true).setVisible(true);  
}
```

```
private void keywordFieldFull1FocusGained(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
  
    if (keywordFieldFull1.getText().equals("Search")) {  
  
        keywordFieldFull1.setText("");  
  
        keywordFieldFull1.setForeground(new Color(0, 0, 0));  
    }  
}
```

```
private void keywordFieldFull1FocusLost(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
  
    if (keywordFieldFull1.getText().equals("")) {  
  
        keywordFieldFull1.setText("Search");  
  
        keywordFieldFull1.setForeground(new Color(153, 153, 153));  
    }  
}
```

```
private void keywordFieldFull1KeyReleased(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String keywordFull1 = keywordFieldFull1.getText();  
  
    salesModel.loadSalesTableFull(keywordFull1, salesTableModelFull);  
}
```

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

```
// TODO add your handling code here:  
  
new GenerateSalesHistoryDialog(this, true).setVisible(true);  
}  
  
  
private void deliveryChargeFieldKeyReleased(java.awt.event.KeyEvent evt) {  
  
// TODO add your handling code here:  
  
calculateValues();  
}  
  
  
private void deliveryChargeFieldKeyPressed(java.awt.event.KeyEvent evt) {  
  
// TODO add your handling code here:  
  
String delivery = deliveryChargeField.getText();  
  
char c = evt.getKeyChar();  
  
if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
  
deliveryChargeField.setEditable(true);  
}  
else {  
  
if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==  
KeyEvent.VK_DELETE) {  
  
deliveryChargeField.setEditable(true);  
}  
else {  
  
deliveryChargeField.setEditable(false);  
}  
}  
}  
}  
  
  
private void invoiceNoLoad() {  
  
int loadInvoiceNo = salesModel.getInvoiceNo();  
  
loadInvoiceNo++;  
  
invoiceNoField.setText(new String(loadInvoiceNo + ""));  
}
```

```
private void calculateValues() {  
  
    int selectedRow = invoiceTable.getSelectedRow();  
  
    int rowCount = invoiceTableModel.getRowCount();  
  
  
    double productQty = (double) invoiceTableModel.getValueAt(selectedRow, 2);  
  
    double productUnitSellPrice = (double) invoiceTableModel.getValueAt(selectedRow, 4);  
  
    double productDiscount = (double) invoiceTableModel.getValueAt(selectedRow, 5);  
  
    double productTax = (double) invoiceTableModel.getValueAt(selectedRow, 6);  
  
    double productAmount = (productUnitSellPrice - productDiscount + productTax) * productQty;  
  
  
    invoiceTableModel.setValueAt(productAmount, selectedRow, 7);  
  
  
    itemCountField.setText(String.valueOf(rowCount));  
  
  
    double deliveryCharge = Double.parseDouble(deliveryChargeField.getText());  
  
  
    double totalDiscount = 0;  
  
    double totalTax = 0;  
  
    double totalAmount = deliveryCharge;  
  
  
    for (int i = 0; i < rowCount; i++) {  
  
        double rowQty = (double) invoiceTableModel.getValueAt(i, 2);  
  
  
        double rowDiscount = (double) invoiceTableModel.getValueAt(i, 5);  
  
        totalDiscount += rowDiscount * rowQty;  
  
  
        double rowTax = (double) invoiceTableModel.getValueAt(i, 6);  
  
        totalTax += rowTax * rowQty;  
  
  
        double rowAmount = (double) invoiceTableModel.getValueAt(i, 7);  
  
        totalAmount += rowAmount;  
    }  
}
```

```

}

discountField.setText(String.valueOf(totalDiscount));
taxField.setText(String.valueOf(totalTax));
totalAmountField.setText(String.valueOf(totalAmount));
netAmountField.setText(String.valueOf(totalAmount));

logger.debug("calculateValues");
}

/** 
 * @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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
    // try {
    //     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
    //         if ("Windows".equals(info.getName())) {

```

```
//         javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//         break;
//
//     }
//
// } catch (ClassNotFoundException ex) {
//
//     java.util.logging.Logger.getLogger(SalesManager.class.getName()).log(java.util.logging.Level.SEVERE, null,
// ex);
//
// } catch (InstantiationException ex) {
//
//     java.util.logging.Logger.getLogger(SalesManager.class.getName()).log(java.util.logging.Level.SEVERE, null,
// ex);
//
// } catch (IllegalAccessException ex) {
//
//     java.util.logging.Logger.getLogger(SalesManager.class.getName()).log(java.util.logging.Level.SEVERE, null,
// ex);
//
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//     java.util.logging.Logger.getLogger(SalesManager.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 SalesManager().setVisible(true);
    }
});
```

```
}
// Variables declaration - do not modify
private javax.swing.JButton addRowButton;
private javax.swing.JTextField customerNameField;
private javax.swing.JButton deleteRowButton;
private javax.swing.JTextField deliveryChargeField;
private javax.swing.JTextField discountField;
```

```
private javax.swing.JButton invoiceButton;
private javax.swing.JButton invoiceButton1;
private com.toedter.calendar.JDateChooser invoiceDateChooser;
private javax.swing.JButton invoiceHistoryButton;
private javax.swing.JTextField invoiceNoField;
private javax.swing.JTable invoiceTable;
private javax.swing.JTable invoiceTableFull;
private javax.swing.JTextField itemCountField;
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.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTextField keywordFieldFull;
```

```
private javax.swing.JTextField keywordFieldFull1;  
private javax.swing.JTextField netAmountField;  
private javax.swing.JButton newGrnButton;  
private javax.swing.JButton salesHistoryButton;  
private javax.swing.JTable salesTableFull;  
private javax.swing.JButton saveInvoiceButton;  
private javax.swing.JTextField taxField;  
private javax.swing.JTextField totalAmountField;  
// End of variables declaration  
}
```

SalesModel.java

```
/*  
 * To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package model;
```

```
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.Date;  
import javax.swing.table.DefaultTableModel;  
import org.apache.log4j.Logger;
```

```
/**
```

```
*
```

```
* @author EZIO AUDITORY
```

```
*/
```

```
public class SalesModel {
```

```
private Connection dbConnection;

static Logger logger = Logger.getLogger(SalesModel.class.getName());

public SalesModel() {

    dbConnection = database.dbConnection.getConnection();

}

public int getInvoiceNo() {

    String query = "SELECT * FROM customer_invoice ORDER BY customer_invoice_no DESC LIMIT 1";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        ResultSet rs = psm.executeQuery();

        logger.debug("Query execution done");

        int rowData = 0;

        while (rs.next()) {

            int invoiceNo = rs.getInt("customer_invoice_no");

            rowData = invoiceNo;

            logger.debug("return rowData");

            return rowData;

        }

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

    }

    logger.debug("return 0");

    return 0;

}
```

```

public String saveInvoice(String invoiceNo, Date invoiceDate, int customerNo, String itemCount, String
deliveryCharge, String totalDiscount, String totalTax, String totalAmount) {

    String query = "INSERT INTO customer_invoice(customer_invoice_no, customer_invoice_date, customer_no,
customer_invoice_item_count, customer_invoice_delivery_charge, customer_invoice_discount,
customer_invoice_tax, customer_invoice_amount)"

    + "VALUE(?,?,?,?,?,?)";

logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, invoiceNo);

    psm.setDate(2, new java.sql.Date(invoiceDate.getTime()));

    psm.setInt(3, customerNo);

    psm.setString(4, itemCount);

    psm.setString(5, deliveryCharge);

    psm.setString(6, totalDiscount);

    psm.setString(7, totalTax);

    psm.setString(8, totalAmount);

    psm.execute();

    logger.debug("Query execution done");

    logger.debug("return result");

    return "Success";

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

    logger.error("return result");

    return "Error !";

}

}

public String saveSales(int productNo, double qty, double unitSellPrice, double discount, double tax, double
amount, String invoiceNo) {

String query1 = "INSERT INTO sales(product_no, sales_quantity, sales_unit_sell_price, sales_discount, sales_tax,

```

```
sales_amount, customer_invoice_no) VALUE(?,?,?,?,?,?)";  
  
String query2 = "SELECT * FROM product WHERE product_no = ?";  
  
String query3 = "UPDATE product SET product_quantity = ? WHERE product_no = ?";  
  
logger.debug("SQL query String: " + query1);  
  
logger.debug("SQL query String: " + query2);  
  
logger.debug("SQL query String: " + query3);  
  
  
try {  
  
    PreparedStatement psm = dbConnection.prepareStatement(query1);  
  
    psm.setInt(1, productNo);  
  
    psm.setDouble(2, qty);  
  
    psm.setDouble(3, unitSellPrice);  
  
    psm.setDouble(4, discount);  
  
    psm.setDouble(5, tax);  
  
    psm.setDouble(6, amount);  
  
    psm.setString(7, invoiceNo);  
  
    psm.execute();  
  
    logger.debug("Query1 execution done");  
  
  
    PreparedStatement psm2 = dbConnection.prepareStatement(query2);  
  
    psm2.setInt(1, productNo);  
  
    ResultSet rs = psm2.executeQuery();  
  
    logger.debug("Query2 execution done");  
  
    double currentQty = qty;  
  
    while (rs.next()) {  
  
        currentQty = rs.getDouble("product_quantity") - currentQty;  
  
    }  
  
    PreparedStatement psm3 = dbConnection.prepareStatement(query3);  
  
    psm3.setDouble(1, currentQty);  
  
    psm3.setInt(2, productNo);  
  
    psm3.execute();
```

```
logger.debug("Query3 execution done");

logger.debug("Query3 execution done");

logger.debug("return result");

return "Success";

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

    logger.error("return result");

    return "Error !";

}

}

public void loadInvoiceTableFull(String keywordFull, DefaultTableModel invoiceTableModelFull) {

    String query = "SELECT * FROM customer_invoice INNER JOIN customer "

        + "ON customer_invoice.customer_no = customer.customer_no "

        + "WHERE customer_invoice_no LIKE ? OR customer_invoice_date LIKE ? OR customer_name LIKE ? ORDER "
    BY customer_invoice_no";

    logger.debug("SQL query String: " + query);


```

```
try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keywordFull + "%");

    psm.setString(2, "%" + keywordFull + "%");

    psm.setString(3, "%" + keywordFull + "%");

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    invoiceTableModelFull.setRowCount(0);

    while (rs.next()) {

        int invoiceNo = rs.getInt("customer_invoice_no");

        Date date = rs.getDate("customer_invoice_date");


```

```

String customerName = rs.getString("customer_name");

int itemCount = rs.getInt("customer_invoice_item_count");

double deliveryCharge = rs.getDouble("customer_invoice_delivery_charge");

double totalDiscount = rs.getDouble("customer_invoice_discount");

double totalTax = rs.getDouble("customer_invoice_tax");

double totalAmount = rs.getDouble("customer_invoice_amount");

rowData = new Object[]{invoiceNo, date, customerName, itemCount, deliveryCharge, totalDiscount,
totalTax, totalAmount};

invoiceTableModelFull.addRow(rowData);

logger.debug("loadInvoiceTableFull [addRow]");

}

} catch (Exception e) {

e.printStackTrace();

logger.error("Exception: " + e.getMessage());

}

}

```

```

public void loadSalesTableFull(String keywordFull1, DefaultTableModel salesTableModelFull) {

String query = "SELECT * FROM sales INNER JOIN customer_invoice INNER JOIN product INNER JOIN unit "

+ "ON sales.customer_invoice_no = customer_invoice.customer_invoice_no AND sales.product_no = "
product.product_no AND product.unit_no = unit.unit_no "

+ "WHERE sales_no LIKE ? OR customer_invoice.customer_invoice_no LIKE ? OR customer_invoice_date "
LIKE ? OR sales.product_no LIKE ? OR product_name LIKE ? ORDER BY sales_no";

logger.debug("SQL query String: " + query);

```

```

try {

PreparedStatement psm = dbConnection.prepareStatement(query);

psm.setString(1, "%" + keywordFull1 + "%");

psm.setString(2, "%" + keywordFull1 + "%");

psm.setString(3, "%" + keywordFull1 + "%");

psm.setString(4, "%" + keywordFull1 + "%");

psm.setString(5, "%" + keywordFull1 + "%");

```

```

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

salesTableModelFull.setRowCount(0);

while (rs.next()) {

    int salesNo = rs.getInt("sales_no");

    Date sellDate = rs.getDate("customer_invoice_date");

    int productNo = rs.getInt("product_no");

    String productName = rs.getString("product_name");

    double qty = rs.getDouble("sales_quantity");

    String unitName = rs.getString("unit_name");

    double unitSellPrice = rs.getDouble("sales_unit_sell_price");

    double discount = rs.getDouble("sales_discount");

    double tax = rs.getDouble("sales_tax");

    double amount = rs.getDouble("sales_amount");

    int invoiceNo = rs.getInt("customer_invoice_no");

    rowData = new Object[]{salesNo, sellDate, productNo, productName, qty, unitName, unitSellPrice, discount,
    tax, amount, invoiceNo};

    salesTableModelFull.addRow(rowData);

    logger.debug("loadSalesTableFull [addRow]");

}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

public Object[] getSelectedInvoiceData(int invoiceNo) {

    String query = "SELECT * FROM customer_invoice WHERE customer_invoice_no = ? ORDER BY
customer_invoice_no";

    logger.debug("SQL query String: " + query);

```

```

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setInt(1, invoiceNo);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    while (rs.next()) {

        int invoiceNoo = rs.getInt("customer_invoice_no");

        double deliveryCharge = rs.getDouble("customer_invoice_delivery_charge");

        rowData = new Object[]{invoiceNoo, deliveryCharge};

        logger.debug("return rowData");

        return rowData;

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}

public void loadInvoiceTableFullForLoadInvoiceDialog(String keywordFull, DefaultTableModel
invoiceTableModelFull) {

    String query = "SELECT * FROM customer_invoice INNER JOIN customer "

        + "ON customer_invoice.customer_no = customer.customer_no "

        + "WHERE (customer_invoice_no LIKE ? OR customer_invoice_date LIKE ? OR customer_name LIKE ?) AND
customer_invoice_delivery_charge != ? ORDER BY customer_invoice_no";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keywordFull + "%");

```

```

psm.setString(2, "%" + keywordFull + "%");
psm.setString(3, "%" + keywordFull + "%");
psm.setInt(4, 0);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

invoiceTableModelFull.setRowCount(0);

while (rs.next()) {

    int invoiceNo = rs.getInt("customer_invoice_no");

    Date date = rs.getDate("customer_invoice_date");

    String customerName = rs.getString("customer_name");

    int itemCount = rs.getInt("customer_invoice_item_count");

    double deliveryCharge = rs.getDouble("customer_invoice_delivery_charge");

    double totalDiscount = rs.getDouble("customer_invoice_discount");

    double totalTax = rs.getDouble("customer_invoice_tax");

    double totalAmount = rs.getDouble("customer_invoice_amount");

    rowData = new Object[]{invoiceNo, date, customerName, itemCount, deliveryCharge, totalDiscount,
    totalTax, totalAmount};

    invoiceTableModelFull.addRow(rowData);

    logger.debug("loadInvoiceTableFullForLoadInvoiceDialog [addRow]");
}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());
}
}
}

```

Batch Name : Griffins

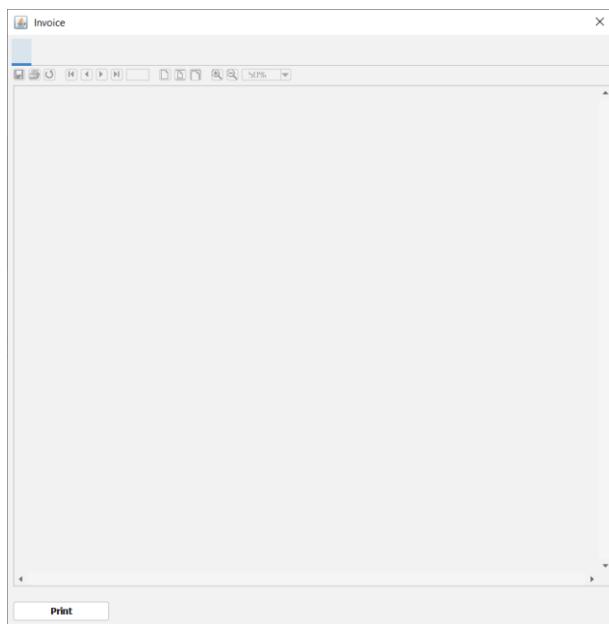
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate Invoice Report

GUI Number : 13

Use case Name: Sales Management



03. Brief Description of the GUI:

The main objective of this interface is to generate, display and print invoice reports.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--------------------------------|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display invoice report | The invoice report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated invoice report | A button is required to print the generated invoice report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateInvoiceDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```

*/
public class GenerateInvoiceDialog extends javax.swing.JDialog {

    /**
     * Creates new form GenerateInvoiceDialog
     */
    static Logger logger = Logger.getLogger(GenerateInvoiceDialog.class.getName());

    String reportPath = "C:\\\\Users\\\\EZIO
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila.hardware_invoice.jasper";

    Map m = new HashMap();
    int invoiceNumber;

    public GenerateInvoiceDialog(java.awt.Frame parent, boolean modal, int invoiceNo) {
        super(parent, modal);
        initComponents();
        logger.debug("Access to GenerateInvoiceDialog.java");

        invoiceNumber = invoiceNo;
        m.put("incoiceNo", invoiceNumber);

        try {
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
            JRViewer jv = new JRViewer(jp);
            jv.setZoomRatio((float) 0.5);
            jTabbedPane1.add(jv);
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("Exception: " + e.getMessage());
        }
    }
}

```

```
/**  
  
 * 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();  
  
    jTabbedPane1 = new javax.swing.JTabbedPane();  
  
    printButton = new javax.swing.JButton();  
  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
    setTitle("Invoice");  
    setResizable(false);  
  
  
    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
    printButton.setText("Print");  
    printButton.addActionListener(new java.awt.event.ActionListener() {  
  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
            printButtonActionPerformed(evt);  
  
        }  
  
    });  
  
  
    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
    jPanel1.setLayout(jPanel1Layout);  
    jPanel1Layout.setHorizontalGroup(  
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel1Layout.createSequentialGroup()  
            .addContainerGap()  
    );  
}
```

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

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel1Layout.createSequentialGroup()

        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap()

);

jPanel1Layout.setVerticalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

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)

);

layout.setVerticalGroup

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

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

);

};
```

```

    pack();

    setLocationRelativeTo(null);

}// </editor-fold>

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

    // TODO add your handling code here:

    m.put("incoiceNo", invoiceNumber);

    try {

        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());

        JasperPrintManager.printReport(jp, true);

        logger.debug("Print Report");

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

    }

}

}

/***
 * @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 {

        UIManager.setLookAndFeel(new FlatIntelliJLaf());

        UIManager.put("Table.showHorizontalLines", true);

        UIManager.put("Table.showVerticalLines", true);

    }

```

```
UIManager.put("ScrollBar.showButtons", true);

UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {

    System.err.println("Failed to initialize LaF");

}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//         if ("Windows".equals(info.getName())) {

//             javax.swing.UIManager.setLookAndFeel(info.getClassName());

//             break;

//         }

//     }

// } catch (ClassNotFoundException ex) {

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

// } catch (InstantiationException ex) {

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

// } catch (IllegalAccessException ex) {

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

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

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

// }

// //</editor-fold>

// //</editor-fold>

// //</editor-fold>

// //</editor-fold>
```

```
/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        int invoiceNo = 0;

        GenerateInvoiceDialog dialog = new GenerateInvoiceDialog(new javax.swing.JFrame(), true, invoiceNo);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {

                System.exit(0);

            }

        });

        dialog.setVisible(true);

    }

});

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JButton printButton;

// End of variables declaration

}
```

Batch Name : Griffins

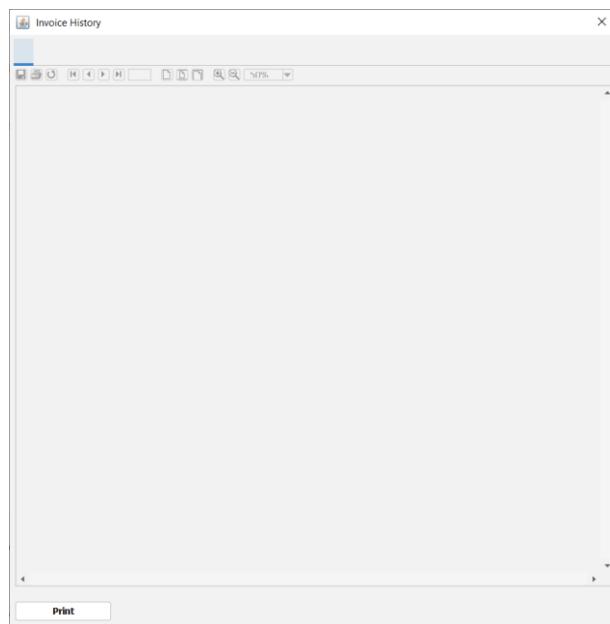
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate Invoice History Report

GUI Number : 14

Use case Name: Sales Management



03. Brief Description of the GUI:

The main objective of this interface is to generate, display and print the invoice history report.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display invoice history report | The invoice history report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated invoice history report | A button is required to print the generated invoice history report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateInvoiceHistoryDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GenerateInvoiceHistoryDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form GenerateInvoiceHistoryDialog  
     */  
  
    static Logger logger = Logger.getLogger(GenerateInvoiceHistoryDialog.class.getName());  
  
    String reportPath = "C:\\\\Users\\\\EZIO  
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila.hardware_invoice_history.jasper";  
  
    Map m = new HashMap();  
  
    public GenerateInvoiceHistoryDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to GenerateInvoiceHistoryDialog.java");  
  
        try {  
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());  
            JRViewer jv = new JRViewer(jp);  
            jv.setZoomRatio((float) 0.5);  
            jTabbedPane1.addTab(jv);  
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
    }  
  
    /**  
     * 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();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    printButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Invoice History");
    setResizable(false);

    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    printButton.setText("Print");
    printButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            printButtonActionPerformed(evt);
        }
    });
});

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(0, 0, 0)
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
);
```

```
.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

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)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
```

```

} // </editor-fold>

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

    // TODO add your handling code here:

    try {

        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());

        JasperPrintManager.printReport(jp, true);

        logger.debug("Print Report");

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

    }

}

/***
 * @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 {

        UIManager.setLookAndFeel(new FlatIntelliJLaf());

        UIManager.put("Table.showHorizontalLines", true);

        UIManager.put("Table.showVerticalLines", true);

        UIManager.put("ScrollBar.showButtons", true);

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {

        System.err.println("Failed to initialize LaF");

    }

}

```

```

}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//
//         if ("Windows".equals(info.getName())) {
//
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//             break;
//         }
//     }
//
// } catch (ClassNotFoundException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateInvoiceHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
// } catch (InstantiationException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateInvoiceHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
// } catch (IllegalAccessException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateInvoiceHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateInvoiceHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
// }

//</editor-fold>

//</editor-fold>

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        GenerateInvoiceHistoryDialog dialog = new GenerateInvoiceHistoryDialog(new javax.swing.JFrame(), true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

```

```
public void windowClosing(java.awt.event.WindowEvent e) {  
    System.exit(0);  
}  
});  
dialog.setVisible(true);  
}  
});  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel jPanel1;  
private javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JButton printButton;  
// End of variables declaration  
}
```

Batch Name : Griffins

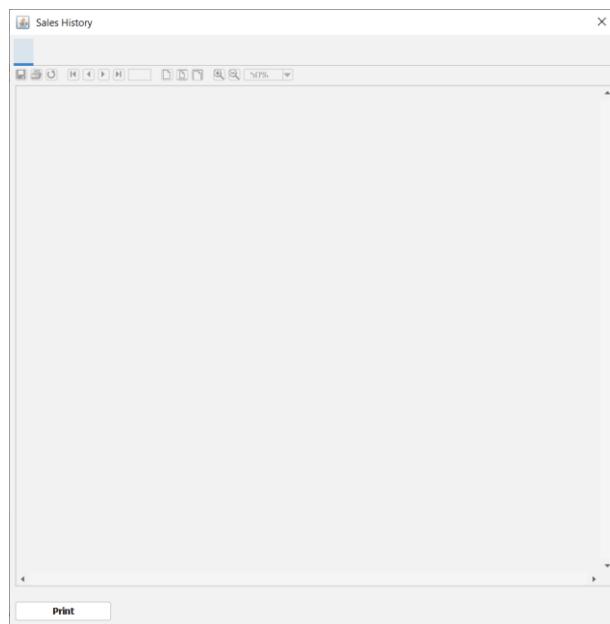
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate Sales History Report

GUI Number : 15

Use case Name: Sales Management



03. Brief Description of the GUI:

The main objective of this interface is to generate, display and print the sales history report.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--------------------------------------|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display sales history report | The Sales history report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated sales history report | A button is required to print the generated sales history report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateSalesHistoryDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GenerateSalesHistoryDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form GenerateSalesHistoryDialog  
     */  
  
    static Logger logger = Logger.getLogger(GenerateSalesHistoryDialog.class.getName());  
  
    String reportPath = "C:\\\\Users\\\\EZIO  
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila.hardware_sales_history.jasper";  
  
    Map m = new HashMap();  
  
    public GenerateSalesHistoryDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to GenerateSalesHistoryDialog.java");  
  
        try {  
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());  
            JRViewer jv = new JRViewer(jp);  
            jv.setZoomRatio((float) 0.5);  
            jTabbedPane1.addTab(jv);  
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
    }  
  
    /**  
     * 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();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    printButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Sales History");
    setResizable(false);

    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    printButton.setText("Print");
    printButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            printButtonActionPerformed(evt);
        }
    });
});

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(0, 0, 0)
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
);
```

```
.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

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)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
```

```
//</editor-fold>

private void printButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
        JasperPrintManager.printReport(jp, true);
        logger.debug("Print Report");
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
    }
}

/**
 * @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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
}
```

```

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(GenerateSalesHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(GenerateSalesHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(GenerateSalesHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(GenerateSalesHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// }

//</editor-fold>
```

```

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        GenerateSalesHistoryDialog dialog = new GenerateSalesHistoryDialog(new javax.swing.JFrame(), true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {
```

```
        System.exit(0);

    }

});  
dialog.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JButton printButton;

// End of variables declaration

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Delivery Manager Create Delivery

GUI Number : 16

Use case Name: Delivery Management

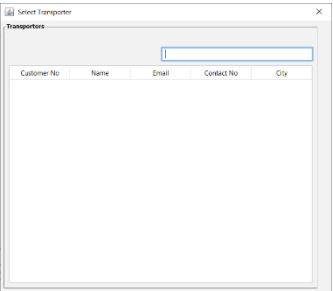
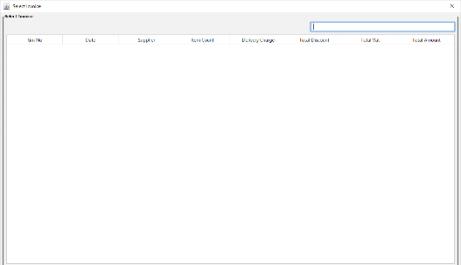
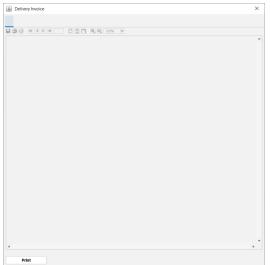
05. Brief Description of the GUI:

The main objective of this interface is to create delivery, save delivery, update delivery, remove delivery and generate delivery invoice reports.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Delivery Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display deliveries | During a search operation, the user can find nearly similar delivery lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert delivery details | To enter or display the details needed to create a delivery. |
| JButtons | Save delivery, Clear text fields & Generate delivery invoice reports | Buttons are needed to save delivery, update delivery, remove delivery, clear text fields and generate delivery invoice reports. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|--|---|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To enter the details of the transporter who is delivering the products. |
| | 02.  | To include the details of the invoice required for delivery. |
| | 03.  | To Generate delivery invoice reports |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

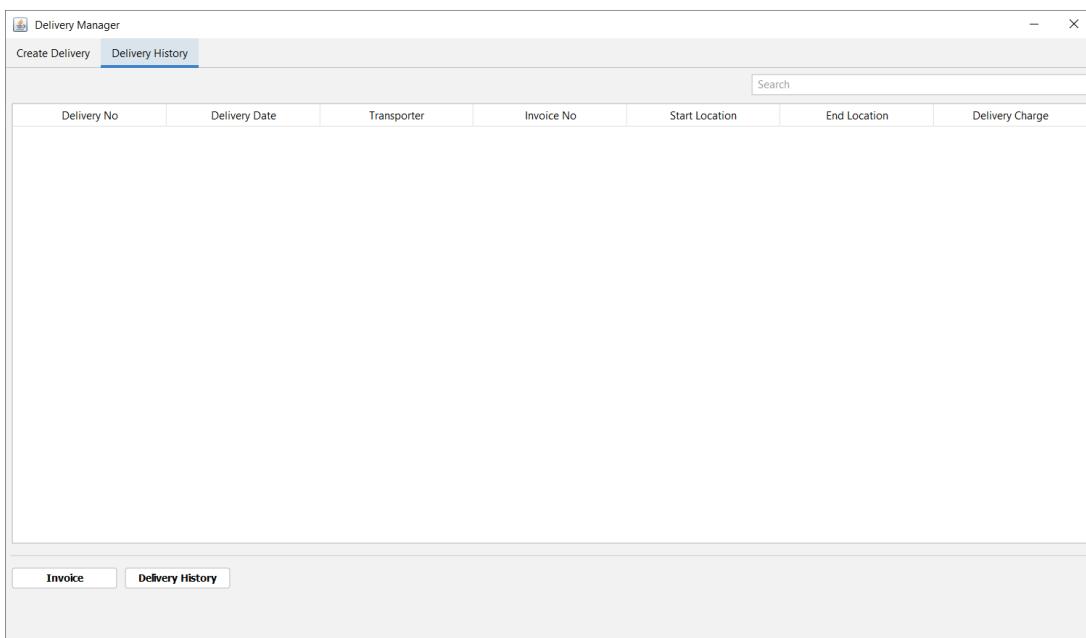
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Delivery Manager Delivery History

GUI Number : 17

Use case Name: Delivery Management



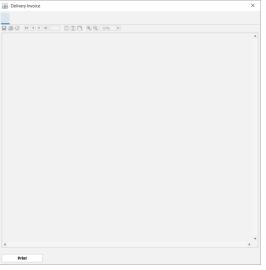
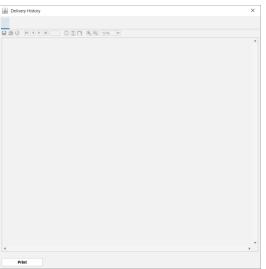
05. Brief Description of the GUI:

The main objective of this interface is to display the delivery invoice history, search for the delivery invoice, select a delivery invoice and generate its delivery invoice report and generate the delivery invoice history report.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Delivery Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display the delivery invoice history | During a search operation, the user can find nearly similar delivery lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |
| JButtons | Generate delivery invoice reports & the delivery invoice history report | Buttons are needed to generate delivery invoices selected from the table and to generate the delivery invoice history. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---|--|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01.  | To generate a delivery invoice selected from the table |
| | 02.  | To Generate the delivery invoice history report |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



DeliveryManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import static gui.CompanyManager.logger;
import java.awt.Color;
import java.awt.Image;
import java.awt.event.KeyEvent;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.DeliveryModel;
import org.apache.log4j.Logger;
/**
```

```
*  
* @author EZIO AUDITORY  
*/  
  
public class DeliveryManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form DeliveryManager  
     */  
  
    private DeliveryModel deliveryModel;  
    private LoadTransporterDialog loadTransporterDialog;  
    private LoadInvoiceDialog loadInvoiceDialog;  
    static Logger logger = Logger.getLogger(DeliveryManager.class.getName());  
  
    DefaultTableModel deliveryTableModel;  
    DefaultTableModel deliveryTableModelFull;  
  
    int transporterNo;  
  
    public DeliveryManager() {  
        initComponents();  
        logger.debug("Access to DeliveryManager.java");  
  
        deliveryTableModel = (DefaultTableModel) deliveryTable.getModel();  
        deliveryTableModelFull = (DefaultTableModel) deliveryTableFull.getModel();  
  
        deliveryModel = new DeliveryModel();  
        loadTransporterDialog = new LoadTransporterDialog(this, rootPaneCheckingEnabled);  
        loadInvoiceDialog = new LoadInvoiceDialog(this, rootPaneCheckingEnabled);  
  
        deliveryModel.loadDeliveryTable("", deliveryTableModel);  
        deliveryModel.loadDeliveryTableFull("", deliveryTableModelFull);  
    }  
}
```

```
deliveryNoLoad();

deliveryDateChooser.setDate(new java.util.Date());

}

/**

 * 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() {

    jTabbedPane1 = new javax.swing.JTabbedPane();

    jPanel1 = new javax.swing.JPanel();

    jPanel4 = new javax.swing.JPanel();

    deliveryNoField = new javax.swing.JTextField();

    transporterField = new javax.swing.JTextField();

    startLocationField = new javax.swing.JTextField();

    endLocationField = new javax.swing.JTextField();

    deliveryChargeField = new javax.swing.JTextField();

    jLabel1 = new javax.swing.JLabel();

    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();

    jLabel8 = new javax.swing.JLabel();
```

```
jSeparator1 = new javax.swing.JSeparator();

saveButton = new javax.swing.JButton();

updateButton = new javax.swing.JButton();

removeButton = new javax.swing.JButton();

newButton = new javax.swing.JButton();

jSeparator3 = new javax.swing.JSeparator();

jSeparator4 = new javax.swing.JSeparator();

deliveryDateChooser = new com.toedter.calendar.JDateChooser();

invoiceNoField = new javax.swing.JTextField();

invoiceButton = new javax.swing.JButton();

jSeparator2 = new javax.swing.JSeparator();

jPanel5 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

deliveryTable = new javax.swing.JTable();

keywordField = new javax.swing.JTextField();

jPanel2 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

deliveryTableFull = new javax.swing.JTable();

jSeparator5 = new javax.swing.JSeparator();

invoiceButton1 = new javax.swing.JButton();

deliveryHistoryButton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Delivery Manager");

setResizable(false);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N
```

```
deliveryNoField.setEditable(false);

transporterField.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        transporterFieldMouseClicked(evt);
    }
});

deliveryChargeField.setText("0.00");
deliveryChargeField.setEnabled(false);
deliveryChargeField.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        deliveryChargeFieldKeyPressed(evt);
    }
});

jLabel1.setText("Delivery No");

jLabel2.setText("Date");

jLabel3.setText("Transporter");

jLabel4.setText("Start Location");

jLabel5.setText("End Location");

jLabel6.setText("Delivery Charge");

jLabel8.setText("Invoice No");

saveButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
saveButton.setText("Save");

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

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

        saveButtonActionPerformed(evt);

    }

});
```

```
updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

updateButton.setText("Update");

updateButton.setEnabled(false);

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

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

        updateButtonActionPerformed(evt);

    }

});
```

```
removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

removeButton.setText("Remove");

removeButton.setEnabled(false);

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

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

        removeButtonActionPerformed(evt);

    }

});
```

```
newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

newButton.setText("New");

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

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

        newButtonActionPerformed(evt);

    }

});
```

```
});

invoiceNoField.setText("0");

invoiceNoField.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        invoiceNoFieldMouseClicked(evt);

    }

});

invoiceNoField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        invoiceNoFieldKeyPressed(evt);

    }

});

invoiceButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

invoiceButton.setText("Invoice");

invoiceButton.setEnabled(false);

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

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

        invoiceButtonActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel4Layout.createSequentialGroup()

        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jSeparator3)


```

```
.addGroup(jPanel4Layout.createSequentialGroup())

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(18, 18, 18)

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(deliveryNoField)

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

    .addGroup(jPanel4Layout.createSequentialGroup()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING

            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(transporterField)

            .addComponent(invoiceNoField))

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

        .addComponent(jSeparator4)

        .addComponent(jSeparator1)

        .addGroup(jPanel4Layout.createSequentialGroup()

            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING

                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)

            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING

                .addComponent(endLocationField)
```

```
.addComponent(deliveryChargeField))

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(startLocationField))

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(saveButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addContainerGap())

);

jPanel4Layout.setVerticalGroup()

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(38, 38, 38)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(deliveryNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

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

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

```
.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(transporterField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

.addComponent(invoiceNoField)

.addGap(18, 18, 18)

.addComponent(jSeparator4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(startLocationField)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(endLocationField)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(deliveryChargeField)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
.addComponent(saveButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(invoiceButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);
```

```
jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Deliveries", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N
```

```
deliveryTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Delivery No", "Date", "Transporter", "Invoice No", "Start Location", "End Location", "Delivery Charge"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false, true, true
};
});
```

```
public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

deliveryTable.setRowHeight(25);

deliveryTable.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        deliveryTableMouseClicked(evt);
    }
});

jScrollPane1.setViewportView(deliveryTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));
keywordField.setText("Search");
keywordField.setToolTipText("Search");
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        keywordFieldFocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        keywordFieldFocusLost(evt);
    }
});

keywordField.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        keywordFieldKeyReleased(evt);
    }
});

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
```

```
jPanel5Layout.setHorizontalGroup(  
    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel5Layout.createSequentialGroup()  
            .addContainerGap()  
            .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 929, Short.MAX_VALUE)  
                .addGroup(jPanel5Layout.createSequentialGroup()  
                    .addGap(0, 0, Short.MAX_VALUE)  
                    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 374,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
                .addContainerGap())  
        );  
  
    jPanel5Layout.setVerticalGroup(  
        jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addGroup(jPanel5Layout.createSequentialGroup()  
                .addGap(32, 32, 32)  
                .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 670, Short.MAX_VALUE)  
                .addContainerGap())  
        );  
  
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
jPanel1.setLayout(jPanel1Layout);  
jPanel1Layout.setHorizontalGroup(  
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel1Layout.createSequentialGroup()  
            .addContainerGap()  
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,  
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
        )  
);
```

```
.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addContainerGap()

);

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(jPanel2)

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap()

);

jTabbedPane1.addTab("Create Delivery", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});
```

```
keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        keywordFieldFullKeyReleased(evt);
    }
});

deliveryTableFull.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "Delivery No", "Delivery Date", "Transporter", "Invoice No", "Start Location", "End Location", "Delivery Charge"
        }
    },
    new boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false
    };
);

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

deliveryTableFull.setRowHeight(25);
jScrollPane3.setViewportView(deliveryTableFull);

invoiceButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
invoiceButton1.setText("Invoice");
invoiceButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        invoiceButton1ActionPerformed(evt);
    }
});
```

```
}

});

deliveryHistoryButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
deliveryHistoryButton.setText("Delivery History");
deliveryHistoryButton.setOpaque(false);
deliveryHistoryButton.addActionListener(new java.awt.event.ActionListener() {

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

        deliveryHistoryButtonActionPerformed(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(0, 0, Short.MAX_VALUE)
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jSeparator5)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(invoiceButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGapPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(deliveryHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGapGap(0, 0, Short.MAX_VALUE)))
);
```

```

    .addContainerGap()

);

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 613,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(invoiceButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(deliveryHistoryButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(76, Short.MAX_VALUE))
);

jTabbedPane1.addTab("Delivery History", jPanel2);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1501,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
);

```

```
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 835,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
);

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

```
private void saveButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String deliveryNo = deliveryNoField.getText();
    Date deliveryDate = deliveryDateChooser.getDate();
    //transporterNo
    String invoiceNo = invoiceNoField.getText();
    String startLocation = startLocationField.getText();
    String endLocation = endLocationField.getText();
    String deliveryCharge = deliveryChargeField.getText();

    logger.debug("Check validations");
    if (trnsporterNo == 0 || transporterField.getText().equals("")) {
        loadTransporterDialog.setVisible(true);
        Object[] transporterData = loadTransporterDialog.getTransporterData();
        trnsporterNo = (int) transporterData[0];
        transporterField.setText((String) transporterData[1]);
    } else if (invoiceNo.equals("0") || invoiceNo.equals("")) {
        loadInvoiceDialog.setVisible(true);
    }
}
```

```

Object[] invoiceData = loadInvoiceDialog.getInvoiceData();

invoiceNoField.setText(new String(invoiceData[0] + ""));
deliveryChargeField.setText(new String(invoiceData[1] + ""));

} else if (startLocation.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Start Location !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    startLocationField.requestFocus(true);

} else if (endLocation.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The End Location !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    endLocationField.requestFocus(true);

} else {

    String result = deliveryModel.saveDelivery(deliveryNo, deliveryDate, transporterNo, invoiceNo, startLocation,
endLocation, deliveryCharge);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        deliveryModel.loadDeliveryTable("", deliveryTableModel);

        deliveryModel.loadDeliveryTableFull("", deliveryTableModelFull);

        invoiceButton.setEnabled(true);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

```

```

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

// TODO add your handling code here:

String deliveryNo = deliveryNoField.getText();

Date deliveryDate = deliveryDateChooser.getDate();

//transporterNo

```

```

String invoiceNo = invoiceNoField.getText();

String startLocation = startLocationField.getText();

String endLocation = endLocationField.getText();

String deliveryCharge = deliveryChargeField.getText();

logger.debug("Check validations");

if (trnsporterNo == 0 || transporterField.getText().equals("")) {

    loadTransporterDialog.setVisible(true);

    Object[] transporterData = loadTransporterDialog.getTransporterData();

    trnsporterNo = (int) transporterData[0];

    transporterField.setText((String) transporterData[1]);

} else if (invoiceNo.equals("0") || invoiceNo.equals("")) {

    loadInvoiceDialog.setVisible(true);

    Object[] invoiceData = loadInvoiceDialog.getInvoiceData();

    invoiceNoField.setText(new String(invoiceData[0] + ""));

    deliveryChargeField.setText(new String(invoiceData[1] + ""));

} else if (startLocation.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Start Location !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    startLocationField.requestFocus(true);

} else if (endLocation.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The End Location !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    endLocationField.requestFocus(true);

} else {

    String result = deliveryModel.updateDelivery(deliveryNo, deliveryDate, trnsporterNo, invoiceNo,
startLocation, endLocation, deliveryCharge);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        deliveryModel.loadDeliveryTable("", deliveryTableModel);

        deliveryModel.loadDeliveryTableFull("", deliveryTableModelFull);

        invoiceButton.setEnabled(true);

    }
}

```

```
        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

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

    // TODO add your handling code here:

    String deliveryNo = deliveryNoField.getText();

    String result = deliveryModel.removeDelivery(deliveryNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        deliveryModel.loadDeliveryTable("", deliveryTableModel);

        deliveryModel.loadDeliveryTableFull("", deliveryTableModelFull);

        invoiceButton.setEnabled(true);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

    clearFields();

}

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

    // TODO add your handling code here:

    clearFields();

}
```

```
private void deliveryTableMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    clearFields();  
  
    int row = deliveryTable.getSelectedRow();  
  
    int deliveryNo = (int) deliveryTableModel.getValueAt(row, 0);  
  
    Object[] productData = deliveryModel.getSelectedDeliveryData(deliveryNo);  
  
  
    deliveryNoField.setText(new String(productData[0] + ""));  
    deliveryDateChooser.setDate((Date) productData[1]);  
    trnsporterNo = (int) productData[2];  
    transporterField.setText((String) productData[3]);  
    invoiceNoField.setText(new String(productData[4] + ""));  
    startLocationField.setText((String) productData[5]);  
    endLocationField.setText((String) productData[6]);  
    deliveryChargeField.setText(new String(productData[7] + ""));  
  
  
    if (deliveryNoField.getText().length() > 0) {  
        saveButton.setEnabled(false);  
        updateButton.setEnabled(true);  
        removeButton.setEnabled(true);  
        invoiceButton.setEnabled(true);  
    } else {  
        // registerButton.setEnabled(true);  
        // updateButton.setEnabled(false);  
        // removeButton.setEnabled(false);  
    }  
}
```

```
private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
  
    if (keywordField.getText().equals("Search")) {  
  
        keywordField.setText("");  
  
        keywordField.setForeground(new Color(0, 0, 0));  
  
    }  
  
}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
  
    if (keywordField.getText().equals("")) {  
  
        keywordField.setText("Search");  
  
        keywordField.setForeground(new Color(153, 153, 153));  
  
    }  
  
}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String keyword = keywordField.getText();  
  
    deliveryModel.loadDeliveryTable(keyword, deliveryTableModel);  
  
}
```

```
private void transporterFieldMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    loadTransporterDialog.setVisible(true);  
  
  
    Object[] transporterData = loadTransporterDialog.getTransporterData();  
  
    transporterNo = (int) transporterData[0];  
  
    transporterField.setText((String) transporterData[1]);  
  
}
```

```
private void invoiceNoFieldMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    loadInvoiceDialog.setVisible(true);  
  
    Object[] invoiceData = loadInvoiceDialog.getInvoiceData();  
  
    invoiceNoField.setText(new String(invoiceData[0] + ""));  
  
    deliveryChargeField.setText(new String(invoiceData[1] + ""));  
}
```

```
private void invoiceNoFieldKeyPressed(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String invoiceNo = invoiceNoField.getText();  
  
    char c = evt.getKeyChar();  
  
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
        invoiceNoField.setEditable(true);  
    } else {  
        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==  
KeyEvent.VK_DELETE) {  
            invoiceNoField.setEditable(true);  
        } else {  
            invoiceNoField.setEditable(false);  
        }  
    }  
}
```

```
private void deliveryChargeFieldKeyPressed(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String deliveryCharge = deliveryChargeField.getText();  
  
    char c = evt.getKeyChar();  
  
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
        deliveryChargeField.setEditable(true);  
    }
```

```
    } else {
        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==
KeyEvent.VK_DELETE) {
            deliveryChargeField.setEditable(true);
        } else {
            deliveryChargeField.setEditable(false);
        }
    }
}
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {
    if (keywordFieldFull.getText().equals("Search")) {
        keywordFieldFull.setText("");
        keywordFieldFull.setForeground(new Color(0, 0, 0));
    }
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordFieldFull.getText().equals("")) {
        keywordFieldFull.setText("Search");
        keywordFieldFull.setForeground(new Color(153, 153, 153));
    }
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String keywordFull = keywordFieldFull.getText();
    deliveryModel.loadDeliveryTableFull(keywordFull, deliveryTableModelFull);
}
```

```
private void invoiceButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    int deliveryNo = Integer.parseInt(deliveryNoField.getText());  
  
    new GenerateDeliveryInvoiceDialog(this, true, deliveryNo).setVisible(true);  
  
}
```

```
private void invoiceButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    int rowCount = deliveryTableFull.getRowCount();  
  
    int selectedRow = deliveryTableFull.getSelectedRow();  
  
    int selectedRowCount = deliveryTableFull.getSelectedRowCount();  
  
  
    if (selectedRowCount == 1) {  
  
        int deliveryNo = (int) deliveryTableModelFull.getValueAt(selectedRow, 0);  
  
        new GenerateDeliveryInvoiceDialog(this, true, deliveryNo).setVisible(true);  
  
    } else {  
  
        if (rowCount == 0) {  
  
            JOptionPane.showConfirmDialog(rootPane, "Table is Empty !", "Error", -1, JOptionPane.ERROR_MESSAGE);  
  
            logger.error("Error: Table is Empty !");  
  
        } else {  
  
            JOptionPane.showConfirmDialog(rootPane, "Please Select any Single Row !", "Error", -1,  
JOptionPane.ERROR_MESSAGE);  
  
            logger.error("Error: Please Select any Single Row !");  
  
        }  
  
    }  
  
}
```

```
private void deliveryHistoryButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    new GenerateDeliveryHistoryDialog(this, true).setVisible(true);
```

```
}
```

```
private void deliveryNoLoad() {  
    int loadDeliveryNo = deliveryModel.getDeliveryNo();  
    loadDeliveryNo++;  
    deliveryNoField.setText(new String(loadDeliveryNo + ""));  
}
```

```
private void clearFields() {  
    deliveryNoLoad();  
    deliveryDateChooser.setDate(new java.util.Date());  
    trnsporterNo = 0;  
    transporterField.setText("");  
    invoiceNoField.setText("0");  
    startLocationField.setText("");  
    endLocationField.setText("");  
    deliveryChargeField.setText("0.00");
```

```
    saveButton.setEnabled(true);  
    updateButton.setEnabled(false);  
    removeButton.setEnabled(false);  
    invoiceButton.setEnabled(false);  
  
    logger.debug("clearFields");  
}
```

```
/**  
 * @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 {

    UIManager.setLookAndFeel(new FlatIntelliJLaf());

    UIManager.put("Table.showHorizontalLines", true);

    UIManager.put("Table.showVerticalLines", true);

    UIManager.put("ScrollBar.showButtons", true);

    UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {

    System.err.println("Failed to initialize LaF");

}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//         if ("Windows".equals(info.getName())) {

//             javax.swing.UIManager.setLookAndFeel(info.getClassName());

//             break;

//         }

//     }

// } catch (ClassNotFoundException ex) {

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

// } catch (InstantiationException ex) {

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

// } catch (IllegalAccessException ex) {

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

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

//     java.util.logging.Logger.getLogger(DeliveryManager.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 DeliveryManager().setVisible(true);
    }
});

}

// Variables declaration - do not modify
private javax.swing.JTextField deliveryChargeField;
private com.toedter.calendar.JDateChooser deliveryDateChooser;
private javax.swing.JButton deliveryHistoryButton;
private javax.swing.JTextField deliveryNoField;
private javax.swing.JTable deliveryTable;
private javax.swing.JTable deliveryTableFull;
private javax.swing.JTextField endLocationField;
private javax.swing.JButton invoiceButton;
private javax.swing.JButton invoiceButton1;
private javax.swing.JTextField invoiceNoField;
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 jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel4;
```

```

private javax.swing.JPanel jPanel5;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator5;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JTextField keywordField;

private javax.swing.JTextField keywordFieldFull;

private javax.swing.JButton newButton;

private javax.swing.JButton removeButton;

private javax.swing.JButton saveButton;

private javax.swing.JTextField startLocationField;

private javax.swing.JTextField transporterField;

private javax.swing.JButton updateButton;

// End of variables declaration

}


```

DeliveryModel.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package model;


```

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


```

```
import java.util.Date;
import javax.swing.table.DefaultTableModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */

public class DeliveryModel {

    private Connection dbConnection;
    static Logger logger = Logger.getLogger(DeliveryModel.class.getName());

    public DeliveryModel() {
        dbConnection = database.dbConnection.getConnection();
    }

    public int getDeliveryNo() {
        String query = "SELECT * FROM delivery ORDER BY delivery_no DESC LIMIT 1";
        logger.debug("SQL query String: " + query);

        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            ResultSet rs = psm.executeQuery();
            logger.debug("Query execution done");

            int rowData = 0;
            while (rs.next()) {
                int grnNoo = rs.getInt("delivery_no");
                rowData = grnNoo;
                logger.debug("return rowData");
            }
            return rowData;
        }
    }
}
```

```
    }

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return 0");

return 0;

}
```

```
public String saveDelivery(String deliveryNo, Date deliveryDate, int transporterNo, String invoiceNo, String startLocation, String endLocation, String deliveryCharge) {
```

```
    String query = "INSERT INTO delivery(delivery_no, delivery_date, transporter_no, customer_invoice_no, delivery_start_location, delivery_end_location, delivery_charge, delivery_status)"
        + "VALUES(?,?,?,?,?,?)";
```

```
    logger.debug("SQL query String: " + query);
```

```
try {
```

```
    PreparedStatement psm = dbConnection.prepareStatement(query);
```

```
    psm.setString(1, deliveryNo);
```

```
    psm.setDate(2, new java.sql.Date(deliveryDate.getTime()));
```

```
    psm.setInt(3, transporterNo);
```

```
    psm.setString(4, invoiceNo);
```

```
    psm.setString(5, startLocation);
```

```
    psm.setString(6, endLocation);
```

```
    psm.setString(7, deliveryCharge);
```

```
    psm.setInt(8, 1);
```

```
    psm.execute();
```

```
    logger.debug("Query execution done");
```

```
    logger.debug("return result");
```

```
    return "Success";
```

```
} catch (Exception e) {
```

```
    e.printStackTrace();
```

```

        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public String updateDelivery(String deliveryNo, Date deliveryDate, int transporterNo, String invoiceNo, String startLocation, String endLocation, String deliveryCharge) {
    String query = "UPDATE delivery SET delivery_date = ?, transporter_no = ?, customer_invoice_no = ?, delivery_start_location = ?, delivery_end_location = ?, delivery_charge = ? "
        + "WHERE delivery_no = ?";

    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setDate(1, new java.sql.Date(deliveryDate.getTime()));
        psm.setInt(2, transporterNo);
        psm.setString(3, invoiceNo);
        psm.setString(4, startLocation);
        psm.setString(5, endLocation);
        psm.setString(6, deliveryCharge);
        psm.setString(7, deliveryNo);
        psm.execute();
        logger.debug("Query execution done");
        logger.debug("return result");
        return "Success";
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

```

```

}

public String removeDelivery(String deliveryNo) {
    String query = "UPDATE delivery SET delivery_status = ? WHERE delivery_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, 0);
        psm.setString(2, deliveryNo);
        psm.execute();
        logger.debug("Query execution done");
        logger.debug("return result");
        return "Success";
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public void loadDeliveryTable(String keyword, DefaultTableModel deliveryTableModel) {
    String query = "SELECT * FROM delivery INNER JOIN transporter INNER JOIN customer_invoice "
    + "ON delivery.transporter_no = transporter.transporter_no AND delivery.customer_invoice_no = "
    + "customer_invoice.customer_invoice_no "
    + "WHERE (delivery_no LIKE ? OR delivery_date LIKE ? OR transporter_name LIKE ? OR "
    + "delivery.customer_invoice_no LIKE ?) AND delivery_status = ? ORDER BY delivery_no";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);

```

```

psm.setString(1, "%" + keyword + "%");
psm.setString(2, "%" + keyword + "%");
psm.setString(3, "%" + keyword + "%");
psm.setString(4, "%" + keyword + "%");
psm.setInt(5, 1);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

deliveryTableModel.setRowCount(0);

while (rs.next()) {

    int deliveryNO = rs.getInt("delivery_no");

    Date date = rs.getDate("delivery_date");

    String transporterName = rs.getString("transporter_name");

    int invoiceNo = rs.getInt("customer_invoice_no");

    String startLocation = rs.getString("delivery_start_location");

    String endtLocation = rs.getString("delivery_end_location");

    double deliveryCharge = rs.getDouble("delivery_charge");

    rowData = new
Object[]{deliveryNO,date,transporterName,invoiceNo,startLocation,endtLocation,deliveryCharge};

    deliveryTableModel.addRow(rowData);

    logger.debug("loadDeliveryTable [addRow]");

}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

public Object[] getSelectedDeliveryData(int deliveryNo) {

String query = "SELECT * FROM delivery INNER JOIN transporter INNER JOIN customer_invoice "
+ "ON delivery.transporter_no = transporter.transporter_no AND delivery.customer_invoice_no =
customer_invoice.customer_invoice_no"

```

```

+ "WHERE delivery_no = ?";

logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setInt(1, deliveryNo);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    while (rs.next()) {

        int deliveryNoo = rs.getInt("delivery_no");

        Date date = rs.getDate("delivery_date");

        int transporterNo = rs.getInt("transporter_no");

        String transporterName = rs.getString("transporter_name");

        int invoiceNo = rs.getInt("customer_invoice_no");

        String startLocation = rs.getString("delivery_start_location");

        String endLocation = rs.getString("delivery_end_location");

        double deliveryCharge = rs.getDouble("delivery_charge");

        rowData = new

Object[]{deliveryNoo,date,transporterNo,transporterName,invoiceNo,startLocation,endLocation,deliveryCharge};

        logger.debug("return rowData");

        return rowData;

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}

```

```

public void loadDeliveryTableFull(String keywordFull, DefaultTableModel deliveryTableModelFull) {

    String query = "SELECT * FROM delivery INNER JOIN transporter INNER JOIN customer_invoice "
        + "ON delivery.transporter_no = transporter.transporter_no AND delivery.customer_invoice_no = "
        + "customer_invoice.customer_invoice_no "
        + "WHERE (delivery_no LIKE ? OR delivery_date LIKE ? OR transporter_name LIKE ? OR "
        + "delivery.customer_invoice_no LIKE ?) AND delivery_status = ? ORDER BY delivery_no";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setString(1, "%" + keywordFull + "%");
        psm.setString(2, "%" + keywordFull + "%");
        psm.setString(3, "%" + keywordFull + "%");
        psm.setString(4, "%" + keywordFull + "%");
        psm.setInt(5, 1);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");

        Object[] rowData = null;
        deliveryTableModelFull.setRowCount(0);
        while (rs.next()) {
            int deliveryNO = rs.getInt("delivery_no");
            Date date = rs.getDate("delivery_date");
            String transporterName = rs.getString("transporter_name");
            int invoiceNo = rs.getInt("customer_invoice_no");
            String startLocation = rs.getString("delivery_start_location");
            String endtLocation = rs.getString("delivery_end_location");
            double deliveryCharge = rs.getDouble("delivery_charge");
            rowData = new Object[]{deliveryNO,date,transporterName,invoiceNo,startLocation,endtLocation,deliveryCharge};
            deliveryTableModelFull.addRow(rowData);
            logger.debug("loadDeliveryTableFull [addRow]");
        }
    }
}

```

```
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
  
    }  
  
}
```

Batch Name : Griffins

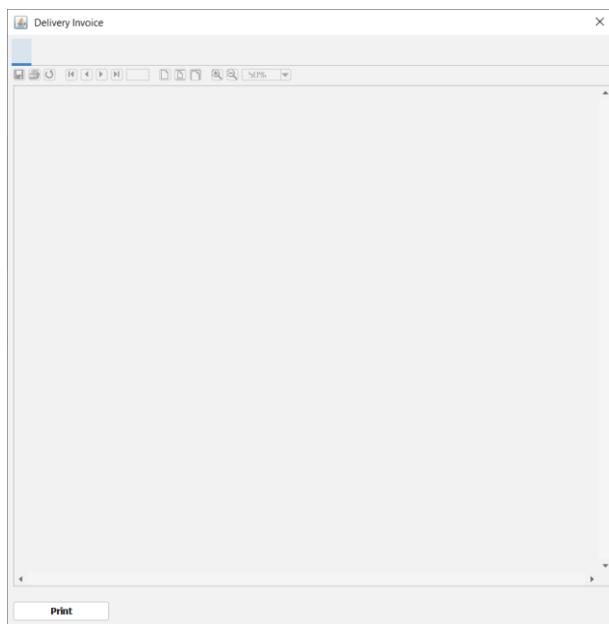
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate Delivery Invoice Report

GUI Number : 18

Use case Name: Delivery Management



05. Brief Description of the GUI:

The main objective of this interface is to generate, display and print delivery invoice reports.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display delivery invoice report | The delivery invoice report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated delivery invoice report | A button is required to print the generated delivery invoice report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateDeliveryInvoiceDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GenerateDeliveryInvoiceDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form GenerateDeliveryInvoiceDialog  
     */  
  
    static Logger logger = Logger.getLogger(GenerateDeliveryInvoiceDialog.class.getName());  
  
    String reportPath = "C:\\\\Users\\\\EZIO  
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila_hardware_delivery_invoice.jasper";  
  
    Map m = new HashMap();  
  
    int deliveryNumber;  
  
  
    public GenerateDeliveryInvoiceDialog(java.awt.Frame parent, boolean modal, int deliveryNo) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to GenerateDeliveryInvoiceDialog.java");  
  
  
        deliveryNumber = deliveryNo;  
        m.put("delivery_no", deliveryNumber);  
        try {  
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());  
            JRViewer jv = new JRViewer(jp);  
            jv.setZoomRatio((float) 0.5);  
            jTabbedPane1.add(jv);  
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
    }  
}
```

```
/**  
  
 * 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();  
  
    jTabbedPane1 = new javax.swing.JTabbedPane();  
  
    printButton = new javax.swing.JButton();  
  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
  
    setTitle("Delivery Invoice");  
  
    setResizable(false);  
  
  
    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
  
    printButton.setText("Print");  
  
    printButton.addActionListener(new java.awt.event.ActionListener() {  
  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
            printButtonActionPerformed(evt);  
  
        }  
  
    });  
  
  
    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(0, 0, 0)  
            .addGroup(jPanel1Layout.createSequentialGroup()  
                .addGap(0, 0, 0)  
                .addContainerGap()  
            )  
        )  
    );  
}  
  
}
```

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

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel1Layout.createSequentialGroup()

        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap()

);

jPanel1Layout.setVerticalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

    .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

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)

);

layout.setVerticalGroup

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

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

);

};
```

```

pack();

 setLocationRelativeTo(null);

}// </editor-fold>

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

// TODO add your handling code here:

m.put("delivery_no", deliveryNumber);

try {

JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());

JasperPrintManager.printReport(jp, true);

logger.debug("Print Report");

} catch (Exception e) {

e.printStackTrace();

logger.error("Exception: " + e.getMessage());

}

}

}

/**

* @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 {

UIManager.setLookAndFeel(new FlatIntelliJLaf());

UIManager.put("Table.showHorizontalLines", true);

UIManager.put("Table.showVerticalLines", true);

}

```

```

        UIManager.put("ScrollBar.showButtons", true);

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//
//         if ("Windows".equals(info.getName())) {
//
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//             break;
//         }
//     }
//
// } catch (ClassNotFoundException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateDeliveryInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
//
// } catch (InstantiationException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateDeliveryInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
//
// } catch (IllegalAccessException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateDeliveryInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
//
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//     java.util.logging.Logger.getLogger(GenerateDeliveryInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
//
// }
//
// //</editor-fold>


```

```

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

```

```
int deliveryNo = 0;

GenerateDeliveryInvoiceDialog dialog = new GenerateDeliveryInvoiceDialog(new javax.swing.JFrame(), true,
deliveryNo);

dialog.addWindowListener(new java.awt.event.WindowAdapter() {

    @Override

    public void windowClosing(java.awt.event.WindowEvent e) {

        System.exit(0);

    }

});

dialog.setVisible(true);

}

});

}

}

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JButton printButton;

// End of variables declaration

}
```

Batch Name : Griffins

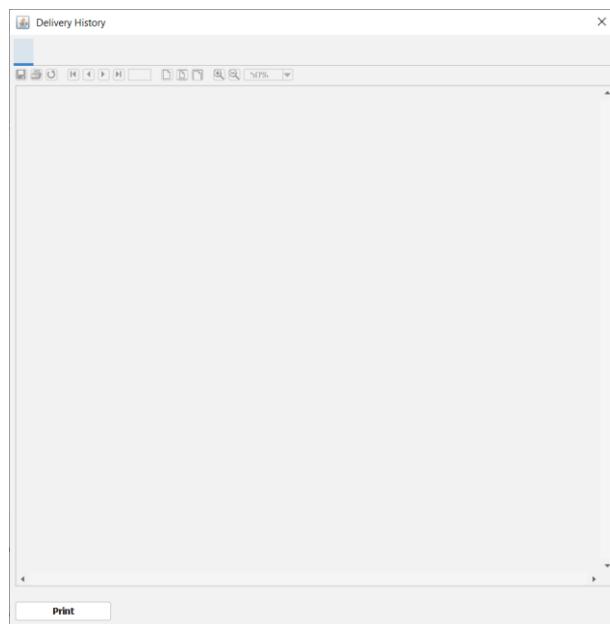
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Generate Delivery History Report

GUI Number : 19

Use case Name: Delivery Management



05. Brief Description of the GUI:

The main objective of this interface is to generate, display and print the delivery invoice history report.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|--|
| JTabbedPane | Load JRViewer | Space is required to load the JRViewer. JTabbedPane is used to provide that space. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JRViewer | Display delivery invoice history report | The delivery invoice history report needs to be generated and displayed. The JRViewer is used for this. |
| JButtons | Print generated delivery invoice history report | A button is required to print the generated delivery invoice history report. This JButton meets that requirement. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



GenerateDeliveryHistoryDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.util.HashMap;
import java.util.Map;
import javax.swing.UIManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.swing.JRViewer;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class GenerateDeliveryHistoryDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form GenerateDeliveryHistoryDialog  
     */  
  
    static Logger logger = Logger.getLogger(GenerateDeliveryHistoryDialog.class.getName());  
  
    String reportPath = "C:\\\\Users\\\\EZIO  
AUDITORY\\\\JaspersoftWorkspace\\\\MyReports\\\\pamunuwila_hardware_delivery_history.jasper";  
    Map m = new HashMap();  
  
    public GenerateDeliveryHistoryDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to GenerateDeliveryHistoryDialog.java");  
  
        try {  
            JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());  
            JRViewer jv = new JRViewer(jp);  
            jv.setZoomRatio((float) 0.5);  
            jTabbedPane1.addTab(jv);  
        } catch (Exception e) {  
            e.printStackTrace();  
            logger.error("Exception: " + e.getMessage());  
        }  
    }  
  
    /**  
     * 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();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    printButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Delivery History");
    setResizable(false);

    printButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    printButton.setText("Print");
    printButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            printButtonActionPerformed(evt);
        }
    });
});

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(0, 0, 0)
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 913, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
);
```

```
.addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 831,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(printButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
);

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)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
```

```
//</editor-fold>

private void printButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        JasperPrint jp = JasperFillManager.fillReport(reportPath, m, database.dbConnection.getConnection());
        JasperPrintManager.printReport(jp, true);
        logger.debug("Print Report");
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
    }
}

/**
 * @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 {
        UIManager.setLookAndFeel(new FlatIntelliJLaf());
        UIManager.put("Table.showHorizontalLines", true);
        UIManager.put("Table.showVerticalLines", true);
        UIManager.put("ScrollBar.showButtons", true);
        UIManager.put("ScrollBar.width", 16);
    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }
}
```

```

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(GenerateDeliveryHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(GenerateDeliveryHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(GenerateDeliveryHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(GenerateDeliveryHistoryDialog.class.getName()).log(java.util.logging.Level.SEVERE
// , null, ex);
// }

//</editor-fold>
```

```

/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        GenerateDeliveryHistoryDialog dialog = new GenerateDeliveryHistoryDialog(new javax.swing.JFrame(),
true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {
```

```
        System.exit(0);

    }

});  
dialog.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JButton printButton;

// End of variables declaration

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Product Manager Product Details

GUI Number : 20

Use case Name: Product Management

The screenshot shows a Java Swing application window titled "Product Manager". It has a tab bar at the top with "Product Details" selected, followed by "Extra Info" and "Products".

- Details Panel:** Contains fields for Product No (text box), Name (text box), Quantity (text box with value 0.0), Unit (combobox with "Number" selected), Unit Sell Price (text box with value 0.0), Discount (text box with value 0.0), and Tax (text box with value 0.0). Below these are buttons for "Register", "Update", "Remove", and a "New" button.
- Products Panel:** Contains a table header row with columns: Product No, Name, Quantity, Unit, and Unit Sell Price. A search bar is located above the table.

01. Brief Description of the GUI:

The main objective of this interface is to register products, update products and remove products.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Product Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JTable | Display products | During a search operation, the user can find nearly similar product lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert product details | To enter or display the details needed to register a product. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register products, Update Products, Remove products & Clear text fields | Buttons are needed to register products, update products, remove products, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

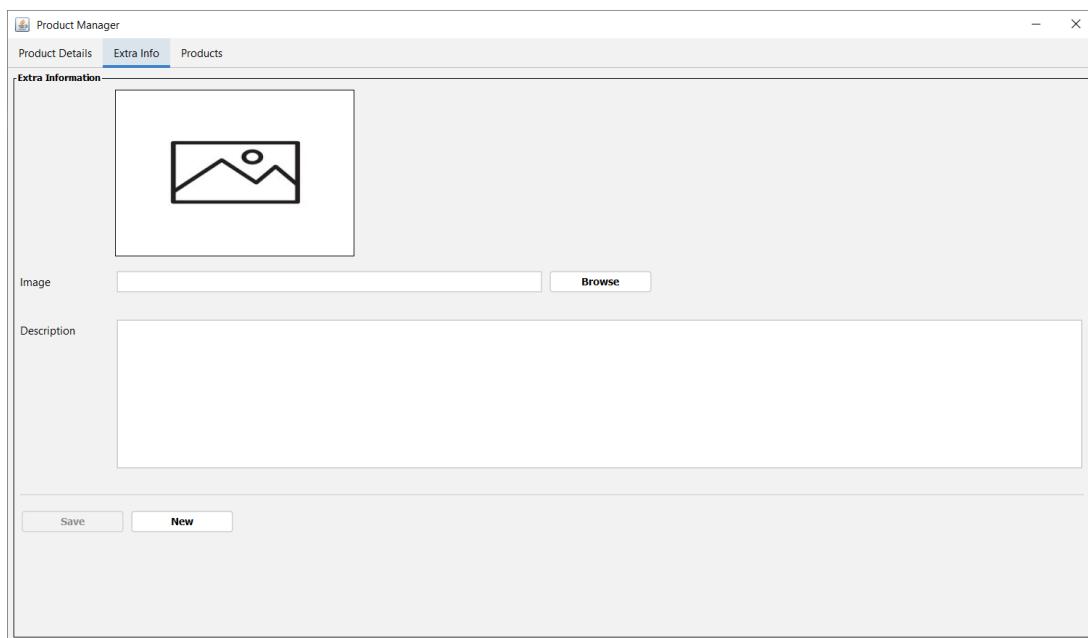
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Product Manager Extra Info

GUI Number : 21

Use case Name: Product Management



01. Brief Description of the GUI:

The main objective of this interface is to save products extra information.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Product Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JLabel | Display text & product image | To identify text fields, display text, and display product images. |
| JTextField | Insert product details | To enter or display the extra information needed to save a product. |
| JButtons | Save products extra info & Clear text fields | Buttons are needed to save products extra information and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

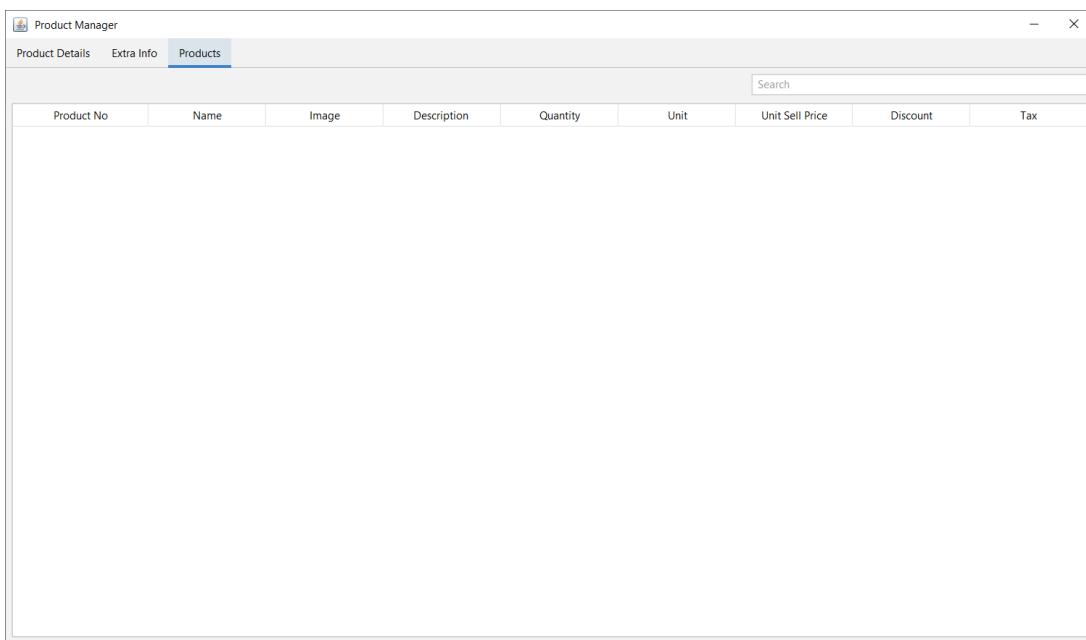
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Product Manager Products

GUI Number : 22

Use case Name: Product Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search the products registered in the system.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Product Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanels makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display products | During a search operation, the user can find nearly similar product lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



ProductTableFullRenderer.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.Component;
import java.awt.Image;
import java.awt.event.KeyEvent;
import java.io.File;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
```

```
import javax.swing.JTable;
import javax.swing.UIManager;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableCellRenderer;
import model.ComboBoxLoadModel;
import model.ProductModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
class ProductTableFullRenderer implements TableCellRenderer {

    @Override
    public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {
        return (Component) value;
    }
}

public class ProductManager extends javax.swing.JFrame {

    /**
     * Creates new form ProductManager
     */
    private ProductModel productModel;
    private ComboBoxLoadModel comboBoxLoadModel;
    static Logger logger = Logger.getLogger(ProductManager.class.getName());
}
```

```
DefaultTableModel productTableModel;
DefaultTableModel productTableModelFull;

DefaultComboBoxModel unitComboBoxModel;

public ProductManager() {
    initComponents();
    logger.debug("Access to ProductManager.java");

    productTableFullSettings(productTableFull);

    productTableModel = (DefaultTableModel) productTable.getModel();
    productTableModelFull = (DefaultTableModel) productTableFull.getModel();

    productModel = new ProductModel();
    comboBoxLoadModel = new ComboBoxLoadModel();

    productModel.loadProductTable("", productTableModel);
    productModel.loadProductTableFull("", productTableModelFull);

    Vector vDistrict = comboBoxLoadModel.unitComboBoxLoad();
    unitComboBoxModel = new DefaultComboBoxModel(vDistrict);
    unitComboBox.setModel(unitComboBoxModel);
    unitComboBox.setSelectedIndex(0);

    imageIconLoad("");
}

/**
 * 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() {

jTabbedPane1 = new javax.swing.JTabbedPane();
jPanel1 = new javax.swing.JPanel();
jPanel4 = new javax.swing.JPanel();
productNoField = new javax.swing.JTextField();
nameField = new javax.swing.JTextField();
quantityField = new javax.swing.JTextField();
unitSellPriceField = new javax.swing.JTextField();
discountField = new javax.swing.JTextField();
taxField = new javax.swing.JTextField();
unitComboBox = new javax.swing.JComboBox<>();
jLabel1 = new javax.swing.JLabel();
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();
jLabel8 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
registerButton = new javax.swing.JButton();
updateButton = new javax.swing.JButton();
removeButton = new javax.swing.JButton();
newButton = new javax.swing.JButton();
jSeparator3 = new javax.swing.JSeparator();
jSeparator4 = new javax.swing.JSeparator();
```

```
jSeparator2 = new javax.swing.JSeparator();
jPanel5 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
productTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();
jPanel2 = new javax.swing.JPanel();
jPanel6 = new javax.swing.JPanel();
imageField = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jSeparator5 = new javax.swing.JSeparator();
saveButton1 = new javax.swing.JButton();
newButton1 = new javax.swing.JButton();
jScrollPane2 = new javax.swing.JScrollPane();
descriptionArea = new javax.swing.JTextArea();
productImageLabel = new javax.swing.JLabel();
browseButton = new javax.swing.JButton();
jPanel3 = new javax.swing.JPanel();
keywordFieldFull = new javax.swing.JTextField();
jScrollPane3 = new javax.swing.JScrollPane();
productTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Product Manager");
setResizable(false);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

productNoField.setEditable(false);
```

```
quantityField.setText("0.0");

quantityField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        quantityFieldKeyPressed(evt);

    }

});
```

```
unitSellPriceField.setText("0.0");

unitSellPriceField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        unitSellPriceFieldKeyPressed(evt);

    }

});
```

```
discountField.setText("0.0");

discountField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        discountFieldKeyPressed(evt);

    }

});
```

```
taxField.setText("0.0");

taxField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        taxFieldKeyPressed(evt);

    }

});
```

```
jLabel1.setText("Product No");
```

```
jLabel2.setText("Name");

jLabel3.setText("Quantity");

jLabel4.setText("Unit Sell Price");

jLabel5.setText("Discount");

jLabel6.setText("Tax");

jLabel8.setText("Unit");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
registerButton.setText("Register");
registerButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        registerButtonActionPerformed(evt);
    }
});

updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
updateButton.setText("Update");
updateButton.setEnabled(false);
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});

removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
removeButton.setText("Remove");
```

```
removeButton.setEnabled(false);

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

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

        removeButtonActionPerformed(evt);

    }

});

newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

newButton.setText("New");

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

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

        newButtonActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel4Layout.createSequentialGroup()

        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

            .addComponent(jSeparator3)

            .addComponent(jSeparator4)

            .addComponent(jSeparator1)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
                    javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(18, 18, 18)  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(discountField)  
            .addComponent(taxField)))  
  
.addGroup(jPanel4Layout.createSequentialGroup()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(18, 18, 18)  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(productNoField)  
            .addComponent(nameField)))  
  
.addGroup(jPanel4Layout.createSequentialGroup()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(18, 18, 18)  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)  
            .addComponent(unitComboBox, javax.swing.GroupLayout.Alignment.LEADING, 0, 318,  
Short.MAX_VALUE)  
            .addComponent(unitSellPriceField)  
            .addComponent(quantityField, javax.swing.GroupLayout.Alignment.LEADING)))  
  
.addGroup(jPanel4Layout.createSequentialGroup()  
        .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addContainerGap()

);

jPanel4Layout.setVerticalGroup

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(38, 38, 38)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(productNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(nameField)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(quantityField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(unitComboBox)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(unitSellPriceField)
    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addComponent(jSeparator4, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(discountField)
        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(taxField)
        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(233, Short.MAX_VALUE))
);
jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);
```

```
jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Products", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

productTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Product No", "Name", "Quantity", "Unit ", "Unit Sell Price"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}

});

productTable.setRowHeight(25);

productTable.addMouseListener(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
productTableMouseClicked(evt);
}
});

jScrollPane1.setViewportView(productTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));
keywordField.setText("Search");
```



```
.addGroup(jPanel5Layout.createSequentialGroup()
    .addGap(32, 32, 32)
    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jScrollPane1)
    .addContainerGap())
);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)
        .addContainerGap())
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addComponent(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jSeparator2)
                        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    )
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    )
                )
            )
        )
    )
);
```

```
.addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap()

);

jTabbedPane1.addTab("Product Details", jPanel1);

jPanel6.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Extra Information", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

jLabel9.setText("Image");

jLabel10.setText("Description");

saveButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
saveButton1.setText("Save");
saveButton1.setEnabled(false);
saveButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
saveButton1ActionPerformed(evt);
}
});

newButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newButton1.setText("New");
newButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
newButton1ActionPerformed(evt);
}
});

});
```

```
descriptionArea.setColumns(20);
descriptionArea.setRows(5);
jScrollPane2.setViewportView(descriptionArea);

productImageLabel.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/basic-design-picture-image-icon-260nw-1475351210.jpg"))); // NOI18N
productImageLabel.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

browseButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
browseButton.setText("Browse");
browseButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        browseButtonActionPerformed(evt);
    }
});
```



```
javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
jPanel6.setLayout(jPanel6Layout);
jPanel6Layout.setHorizontalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel6Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jScrollPane2))
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(jPanel6Layout.createSequentialGroup()
                        .addGap(18, 18, 18)
                        .addComponent(jScrollPane2))
                    .addGroup(jPanel6Layout.createSequentialGroup()
                        .addGap(18, 18, 18)
                        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel6Layout.createSequentialGroup()
                                .addGap(18, 18, 18)
                                .addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(jPanel6Layout.createSequentialGroup()
                                .addGap(18, 18, 18)
                                .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swing.GroupLayout.PREFERRED_SIZE)))))))))))
```

```
        .addComponent(saveButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(newButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel6Layout.createSequentialGroup()

            .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(productImageLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 330,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGroup(jPanel6Layout.createSequentialGroup()

                    .addComponent(imageField, javax.swing.GroupLayout.PREFERRED_SIZE, 591,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

                    .addComponent(browseButton, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)))))

            .addGap(0, 564, Short.MAX_VALUE)))

        .addContainerGap()

);

jPanel6Layout.setVerticalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel6Layout.createSequentialGroup()
        .addContainerGap()

        .addComponent(productImageLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 230,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(imageField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(browseButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(33, 33, 33)  
        .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 210,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(32, 32, 32)  
        .addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addGap(18, 18, 18)  
    .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
        .addComponent(saveButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
            javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addComponent(newButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
            javax.swing.GroupLayout.PREFERRED_SIZE))  
    .addContainerGap(131, Short.MAX_VALUE))  
);  
  
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(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
            Short.MAX_VALUE)  
        .addContainerGap())  
);  
jPanel2Layout.setVerticalGroup(  
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()  
        .addContainerGap())
```

```
.addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addContainerGap()

);

jTabbedPane1.addTab("Extra Info", jPanel2);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFullKeyReleased(evt);

    }

});

productTableFull.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

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

    new String [] {
        "Product No", "Name", "Image", "Description ", "Quantity", "Unit", "Unit Sell Price", "Discount", "Tax"
    }
},
```

```
        }

    ) {

        boolean[] canEdit = new boolean [] {
            false, false, false, false, false, false, false, false
        };
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

jScrollPane3.setViewportView(productTableFull);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE))
);
```

```

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

    .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 738, Short.MAX_VALUE)

    .addContainerGap()

);

jTabbedPane1.addTab("Products", jPanel3);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1501,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 835,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE)
        );
    );
}

pack();

setLocationRelativeTo(null);

}// </editor-fold>

```

```

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

    // TODO add your handling code here:
    String name = nameField.getText();
    String qty = quantityField.getText();
}

```

```
String unitSellPrice = unitSellPriceField.getText();

String discount = discountField.getText();

String tax = taxField.getText();

int unitNo = unitComboBox.getSelectedIndex() + 1;

String image = imageField.getText();

String description = descriptionArea.getText();

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (unitSellPrice.equals("0.0")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Unit Sell Price !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    unitSellPriceField.requestFocus(true);

} else {

    String result = productModel.registerProduct(name, qty, unitSellPrice, discount, tax, unitNo, image,
description);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        productModel.loadProductTable("", productTableModel);

        productModel.loadProductTableFull("", productTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

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

```
// TODO add your handling code here:

String productNo = productNoField.getText();

String name = nameField.getText();

String qty = quantityField.getText();

String unitSellPrice = unitSellPriceField.getText();

String discount = discountField.getText();

String tax = taxField.getText();

int unitNo = unitComboBox.getSelectedIndex() + 1;

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (unitSellPrice.equals("0.0")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Unit Sell Price !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    unitSellPriceField.requestFocus(true);

} else {

    String result = productModel.updateProduct(productNo, name, qty, unitSellPrice, discount, tax, unitNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        productModel.loadProductTable("", productTableModel);

        productModel.loadProductTableFull("", productTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}
```

```
private void removeButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    String productNo = productNoField.getText();  
  
  
    int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",  
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);  
  
  
    if (response == JOptionPane.YES_OPTION) {  
  
        String result = productModel.removeProduct(productNo);  
  
        if (result.equals("Success")) {  
  
            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);  
  
            productModel.loadProductTable("", productTableModel);  
  
            productModel.loadProductTableFull("", productTableModelFull);  
  
            logger.debug("Message: " + result);  
  
        } else {  
  
            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);  
  
            logger.error("Error: " + result);  
  
        }  
  
  
        clearFields();  
  
    }  
  
  
}  
  
  
private void newButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    clearFields();  
  
}  
  
  
private void productTableMouseClicked(java.awt.event.MouseEvent evt) {
```

```
// TODO add your handling code here:  
  
clearFields();  
  
  
int row = productTable.getSelectedRow();  
  
int productNo = (int) productTableModel.getValueAt(row, 0);  
  
Object[] productData = productModel.getSelectedProductData(productNo);  
  
  
productNoField.setText(new String(productData[0] + ""));  
  
nameField.setText((String) productData[1]);  
  
quantityField.setText(new String(productData[2] + ""));  
  
unitComboBox.setSelectedIndex((int) productData[3] - 1);  
  
unitSellPriceField.setText(new String(productData[4] + ""));  
  
discountField.setText(new String(productData[5] + ""));  
  
taxField.setText(new String(productData[6] + ""));  
  
imageField.setText((String) productData[7]);  
  
descriptionArea.setText((String) productData[8]);  
  
  
String path = (String) productData[7];  
  
Image getAbsolutePath = null;  
  
imageIconLoad(path);  
  
  
if (productNoField.getText().length() > 0) {  
  
    registerButton.setEnabled(false);  
  
    updateButton.setEnabled(true);  
  
    removeButton.setEnabled(true);  
  
    saveButton1.setEnabled(true);  
  
} else {  
  
    //    registerButton.setEnabled(true);  
    //    updateButton.setEnabled(false);  
    //    removeButton.setEnabled(false);  
  
}  

```

```
}
```

```
private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (keywordField.getText().equals("Search")) {  
  
        keywordField.setText("");  
  
        keywordField.setForeground(new Color(0, 0, 0));  
  
    }  
  
}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (keywordField.getText().equals("")) {  
  
        keywordField.setText("Search");  
  
        keywordField.setForeground(new Color(153, 153, 153));  
  
    }  
  
}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keyword = keywordField.getText();  
  
    productModel.loadProductTable(keyword, productTableModel);  
  
}
```

```
private void saveButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    String productNo = productNoField.getText();  
  
    String image = imageField.getText();  
  
    String description = descriptionArea.getText();  
  
  
    String result = productModel.saveProduct(productNo, image, description);
```

```
if (result.equals("Success")) {  
    JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);  
  
    productModel.loadProductTable("", productTableModel);  
  
    productModel.loadProductTableFull("", productTableModelFull);  
  
    logger.debug("Message: " + result);  
  
} else {  
  
    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);  
  
    logger.error("Error: " + result);  
  
}  
  
}
```

```
private void newButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    imageField.setText("");  
  
    descriptionArea.setText("");  
  
    imageIconLoad("");  
  
  
    logger.debug("clearFields tab 2");  
  
  
    //saveButton1.setEnabled(false);  
}  
  
}
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {  
  
    if (keywordFieldFull.getText().equals("Search")) {  
  
        keywordFieldFull.setText("");  
  
        keywordFieldFull.setForeground(new Color(0, 0, 0));  
  
    }  
  
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
}
```

```
if (keywordFieldFull.getText().equals("")) {  
    keywordFieldFull.setText("Search");  
    keywordFieldFull.setForeground(new Color(153, 153, 153));  
}  
}  
  
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keywordFull = keywordFieldFull.getText();  
    productModel.loadProductTableFull(keywordFull, productTableModelFull);  
  
}  
  
private void quantityFieldKeyPressed(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String qty = quantityField.getText();  
    char c = evt.getKeyChar();  
  
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
        quantityField.setEditable(true);  
    } else {  
  
        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==  
KeyEvent.VK_DELETE) {  
  
            quantityField.setEditable(true);  
        } else {  
  
            quantityField.setEditable(false);  
        }  
    }  
}  
  
private void unitSellPriceFieldKeyPressed(java.awt.event.KeyEvent evt) {  
  
    String qty = unitSellPriceField.getText();
```

```
char c = evt.getKeyChar();

if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {

    unitSellPriceField.setEditable(true);

} else {

    if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() == KeyEvent.VK_DELETE) {

        unitSellPriceField.setEditable(true);

    } else {

        unitSellPriceField.setEditable(false);

    }

}

}
```

```
private void discountFieldKeyPressed(java.awt.event.KeyEvent evt) {

String qty = discountField.getText();

char c = evt.getKeyChar();

if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {

    discountField.setEditable(true);

} else {

    if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() == KeyEvent.VK_DELETE) {

        discountField.setEditable(true);

    } else {

        discountField.setEditable(false);

    }

}

}
```

```
private void taxFieldKeyPressed(java.awt.event.KeyEvent evt) {

String qty = taxField.getText();

char c = evt.getKeyChar();

if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {
```

```
taxField.setEditable(true);

} else {

    if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==
KeyEvent.VK_DELETE) {

        taxField.setEditable(true);

    } else {

        taxField.setEditable(false);

    }

}

}
```

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

```
JFileChooser chooser = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter("All pic", "png", "jpg", "jpeg", "gif");
chooser.addChoosableFileFilter(filter);
```

```
chooser.showOpenDialog(null);
```

```
File file = chooser.getSelectedFile();
String path = file.getAbsolutePath();
imageField.setText(path);
Image getAbsolutePath = null;
imageIconLoad(path);
```

```
}
```

```
private void clearFields() {
    productNoField.setText("");
    nameField.setText("");
```

```
quantityField.setText("0.0");

unitSellPriceField.setText("0.0");

discountField.setText("0.0");

taxField.setText("0.0");

unitComboBox.setSelectedIndex(0);

imageField.setText("");

descriptionArea.setText("");

imageIconLoad("");

registerButton.setEnabled(true);

updateButton.setEnabled(false);

removeButton.setEnabled(false);

logger.debug("clearFields");

}
```

```
private void imageIconLoad(String path) {

if (path.length() > 0) {

ImageIcon icon = new ImageIcon(path);

Image image = icon.getImage().getScaledInstance(productImageLabel.getWidth(),
productImageLabel.getHeight(), Image.SCALE_SMOOTH);

ImageIcon icon1 = new ImageIcon(image);

productImageLabel.setIcon(icon1);

} else {

ImageIcon icon = new ImageIcon("C:\\\\Users\\\\EZIO
AUDITORY\\\\Documents\\\\NetBeansProjects\\\\Pamunuwila_Hardware\\\\src\\\\images\\\\basic-design-picture-image-icon-
260nw-1475351210.jpg");

Image image = icon.getImage().getScaledInstance(productImageLabel.getWidth(),
productImageLabel.getHeight(), Image.SCALE_SMOOTH);

ImageIcon icon1 = new ImageIcon(image);

productImageLabel.setIcon(icon1);

}

}
```

```

}

// public ImageIcon setIcon(String m,byte[] image){
//     if(m!=null){
//         productImage = new ImageIcon(m);
//     }else{
//         productImage = new ImageIcon(image);
//     }
// }

// Image img1=productImage.getImage();
// Image img2=img1.getScaledInstance(productImageLabel.getWidth(), productImageLabel.getHeight(),
// Image.SCALE_SMOOTH);
// ImageIcon i = new ImageIcon(img2)
// return i;
// }

```

```

public void productTableFullSettings(JTable jTable) {
    jTable.setRowHeight(100);
    jTable.getColumn("Image").setCellRenderer(new ProductTableFullRenderer());
    jTable.getColumnModel().getColumn(0).setPreferredWidth(100);
}

```

```

/**
 * @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 {

```

```
UIManager.setLookAndFeel(new FlatIntelliJLaf());
UIManager.put("Table.showHorizontalLines", true);
UIManager.put("Table.showVerticalLines", true);
UIManager.put("ScrollBar.showButtons", true);
UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// }

// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(ProductManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(ProductManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(ProductManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(ProductManager.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// }

// }

//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
```

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

// Variables declaration - do not modify
private javax.swing.JButton browseButton;
private javax.swing.JTextArea descriptionArea;
private javax.swing.JTextField discountField;
private javax.swing.JTextField imageField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
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 jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
```

```
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JSeparator jSeparator5;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTextField keywordField;
private javax.swing.JTextField keywordFieldFull;
private javax.swing.JTextField nameField;
private javax.swing.JButton newButton;
private javax.swing.JButton newButton1;
private javax.swing.JLabel productImageLabel;
private javax.swing.JTextField productNoField;
private javax.swing.JTable productTable;
private javax.swing.JTable productTableFull;
private javax.swing.JTextField quantityField;
private javax.swing.JButton registerButton;
private javax.swing.JButton removeButton;
private javax.swing.JButton saveButton1;
private javax.swing.JTextField taxField;
private javax.swing.JComboBox<String> unitComboBox;
private javax.swing.JTextField unitSellPriceField;
private javax.swing.JButton updateButton;
// End of variables declaration
}
```

ProductModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.

```

```
*/  
  
package model;  
  
  
import java.awt.Image;  
  
import java.sql.Connection;  
  
import java.sql.PreparedStatement;  
  
import java.sql.ResultSet;  
  
import javax.swing.ImageIcon;  
  
import javax.swing.JLabel;  
  
import javax.swing.table.DefaultTableModel;  
  
import org.apache.log4j.Logger;  
  
  
/**  
 *  
 * @author EZIO AUDITORY  
 */  
  
public class ProductModel {  
  
  
    private Connection dbConnection;  
  
    static Logger logger = Logger.getLogger(ProductModel.class.getName());  
  
  
    public ProductModel() {  
        dbConnection = database.dbConnection.getConnection();  
    }  
  
  
    public String registerProduct(String name, String qty, String unitSellPrice, String discount, String tax, int unitNo,  
        String image, String description) {  
        String query = "INSERT INTO product(product_name, product_image_url, product_description,  
            product_quantity, unit_no, product_unit_sell_price, product_discount, product_tax, product_status)"  
            + "VALUE(?,?,?,?,?,?,?,?,?)";  
        logger.debug("SQL query String: " + query);  
    }  
}
```

```

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, name);
    psm.setString(2, image);
    psm.setString(3, description);
    psm.setString(4, qty);
    psm.setInt(5, unitNo);
    psm.setString(6, unitSellPrice);
    psm.setString(7, discount);
    psm.setString(8, tax);
    psm.setInt(9, 1);
    psm.execute();

    logger.debug("Query execution done");
    logger.debug("return result");

    return "Success";
} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
    logger.error("return result");

    return "Error !";
}

}

public String updateProduct(String productNo, String name, String qty, String unitSellPrice, String discount, String tax, int unitNo) {

    String query = "UPDATE product SET product_name = ?, product_quantity = ?, unit_no = ?, "
    product_unit_sell_price = ?, product_discount = ?, product_tax = ? "
    + "WHERE product_no = ?";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

```

```

psm.setString(1, name);
psm.setString(2, qty);
psm.setInt(3, unitNo);
psm.setString(4, unitSellPrice);
psm.setString(5, discount);
psm.setString(6, tax);
psm.setString(7, productNo);
psm.execute();
logger.debug("Query execution done");
logger.debug("return result");
return "Success";
} catch (Exception e) {
e.printStackTrace();
logger.error("Exception: " + e.getMessage());
logger.error("return result");
return "Error !";
}
}

public String saveProduct(String productNo, String image, String description) {
String query = "UPDATE product SET product_image_url = ?, product_description = ? WHERE product_no = ?";
logger.debug("SQL query String: " + query);

try {
PreparedStatement psm = dbConnection.prepareStatement(query);
psm.setString(1, image);
psm.setString(2, description);
psm.setString(3, productNo);
psm.execute();
logger.debug("Query execution done");
logger.debug("return result");
}

```

```

        return "Success";

    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public String removeProduct(String productNo) {
    String query = "UPDATE product SET product_status = ? WHERE product_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, 0);
        psm.setString(2, productNo);
        psm.execute();
        logger.debug("Query execution done");
        logger.debug("return result");
        return "Success";
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public void loadProductTable(String keyword, DefaultTableModel productTableModel) {
    String query = "SELECT * FROM product INNER JOIN unit "

```

```

+ "ON product.unit_no = unit.unit_no"
+ "WHERE (product_no LIKE ? OR product_name LIKE ?) AND product_status = ? ORDER BY product_no";
logger.debug("SQL query String: " + query);

try {
    PreparedStatement psm = dbConnection.prepareStatement(query);
    psm.setString(1, "%" + keyword + "%");
    psm.setString(2, "%" + keyword + "%");
    psm.setInt(3, 1);
    ResultSet rs = psm.executeQuery();
    logger.debug("Query execution done");
    Object[] rowData = null;
    productTableModel.setRowCount(0);
    while (rs.next()) {
        int productNo = rs.getInt("product_no");
        String name = rs.getString("product_name");
        double qty = rs.getDouble("product_quantity");
        String unit = rs.getString("unit_name");
        double unitSellPrice = rs.getDouble("product_unit_sell_price");
        rowData = new Object[]{productNo, name, qty, unit, unitSellPrice};
        productTableModel.addRow(rowData);
        logger.debug("loadProductTable [addRow]");
    }
} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

public Object[] getSelectedProductData(int productNo) {
    String query = "SELECT * FROM product INNER JOIN unit ON product.unit_no = unit.unit_no WHERE product_no"

```

```
= "?";  
  
logger.debug("SQL query String: " + query);  
  
  
try {  
  
    PreparedStatement psm = dbConnection.prepareStatement(query);  
  
    psm.setInt(1, productNo);  
  
    ResultSet rs = psm.executeQuery();  
  
    logger.debug("Query execution done");  
  
    Object[] rowData = null;  
  
    while (rs.next()) {  
  
        int productNoo = rs.getInt("product_no");  
  
        String name = rs.getString("product_name");  
  
        double qty = rs.getDouble("product_quantity");  
  
        int unitNo = rs.getInt("unit_no");  
  
        double unitSellPrice = rs.getDouble("product_unit_sell_price");  
  
        double discount = rs.getDouble("product_discount");  
  
        double tax = rs.getDouble("product_tax");  
  
        String image = rs.getString("product_image_url");  
  
        String description = rs.getString("product_description");  
  
        String unitName = rs.getString("unit_name");  
  
        rowData = new Object[]{productNoo, name, qty, unitNo, unitSellPrice, discount, tax, image, description, unitName};  
  
        logger.debug("return rowData");  
  
        return rowData;  
  
    }  
  
} catch (Exception e) {  
  
    e.printStackTrace();  
  
    logger.error("Exception: " + e.getMessage());  
  
}  
  
logger.debug("return null");  
  
return null;
```

```
}

public void loadProductTableFull(String keywordFull, DefaultTableModel productTableModelFull) {

    String query = "SELECT * FROM product INNER JOIN unit "
        + "ON product.unit_no = unit.unit_no "
        + "WHERE (product_no LIKE ? OR product_name LIKE ?) AND product_status = ? ORDER BY product_no";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setString(1, "%" + keywordFull + "%");
        psm.setString(2, "%" + keywordFull + "%");
        psm.setInt(3, 1);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");

        Object[] rowData = null;
        productTableModelFull.setRowCount(0);
        while (rs.next()) {
            int productNo = rs.getInt("product_no");
            String name = rs.getString("product_name");
            String image = rs.getString("product_image_url");

            ImageIcon ii = new ImageIcon(new ImageIcon(image).getImage().getScaledInstance(160, 100,
Image.SCALE_SMOOTH));
            JLabel imageLabel = new JLabel();
            imageLabel.setIcon(ii);

            String description = rs.getString("product_description");
            double qty = rs.getDouble("product_quantity");
            String unit = rs.getString("unit_name");
            double unitSellPrice = rs.getDouble("product_unit_sell_price");
        }
    }
}
```

```
        double discount = rs.getDouble("product_discount");

        double tax = rs.getDouble("product_tax");

        rowData = new Object[]{productNo, name, imageLable, description, qty, unit, unitSellPrice, discount, tax};

        productTableModelFull.addRow(rowData);

        logger.debug("loadProductTableFull [addRow]");

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Supplier Manager Supplier Details

GUI Number : 23

Use case Name: Supplier Management

01. Brief Description of the GUI:

The main objective of this interface is to register suppliers, update suppliers and remove suppliers.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Supplier Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display suppliers | During a search operation, the user can find nearly similar supplier lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert supplier details | To enter or display the details needed to register a supplier. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register suppliers, Update suppliers, Remove suppliers & Clear text fields | Buttons are needed to register suppliers, update suppliers, remove suppliers, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

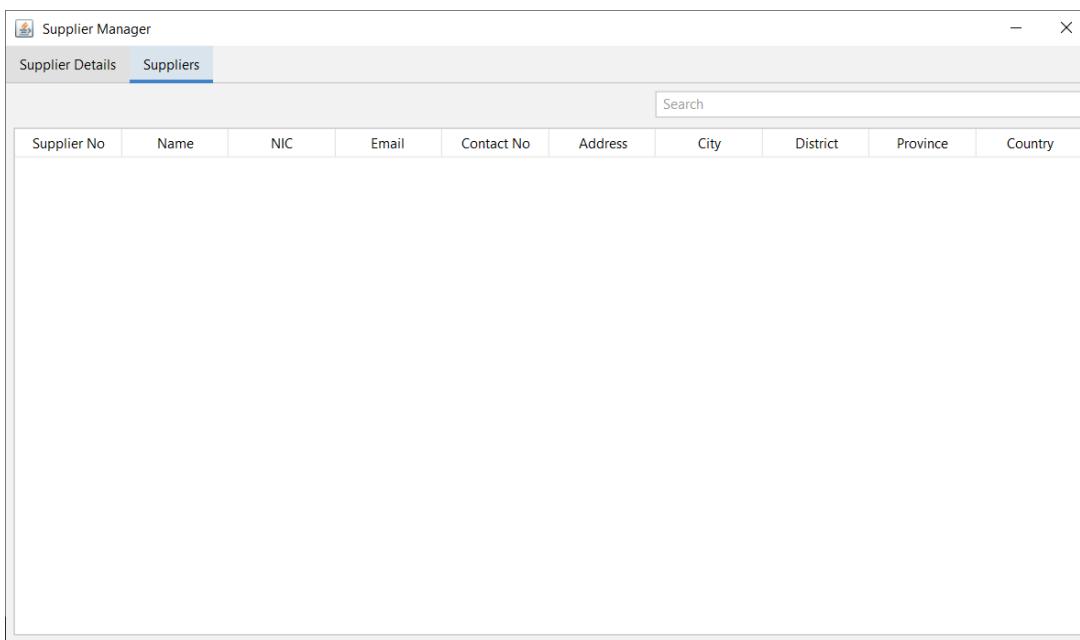
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Supplier Manager Suppliers

GUI Number : 24

Use case Name: Supplier Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search the suppliers registered in the system.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Supplier Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display suppliers | During a search operation, the user can find nearly similar supplier lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



SupplierManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import static gui.CompanyManager.logger;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.ComboBoxLoadModel;
import model.SupplierModel;
```

```
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */

public class SupplierManager extends javax.swing.JFrame {

    /**
     * Creates new form SupplierManager
     */
    private SupplierModel supplierModel;
    private ComboBoxLoadModel comboBoxLoadModel;
    static Logger logger = Logger.getLogger(SupplierManager.class.getName());

    DefaultTableModel supplierTableModel;
    DefaultTableModel supplierTableModelFull;

    DefaultComboBoxModel districtComboBoxModel;
    DefaultComboBoxModel provinceComboBoxModel;
    DefaultComboBoxModel companyComboBoxModel;

    public SupplierManager() {
        initComponents();
        logger.debug("Access to SupplierManager.java");

        supplierTableModel = (DefaultTableModel) supplierTable.getModel();
        supplierTableModelFull = (DefaultTableModel) supplierTableFull.getModel();

        supplierModel = new SupplierModel();
        comboBoxLoadModel = new ComboBoxLoadModel();
    }
}
```

```

supplierModel.loadSupplierTable("", supplierTableModel);

supplierModel.loadSupplierTableFull("", supplierTableModelFull);

Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();

districtComboBoxModel = new DefaultComboBoxModel(vDistrict);

districtComboBox.setModel(districtComboBoxModel);

districtComboBox.setSelectedIndex(0);

Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();

provinceComboBoxModel = new DefaultComboBoxModel(vProvince);

provinceComboBox.setModel(provinceComboBoxModel);

provinceComboBox.setSelectedIndex(0);

Vector vCompany = comboBoxLoadModel.companyComboBoxLoad();

companyComboBoxModel = new DefaultComboBoxModel(vCompany);

companyComboBox.setModel(companyComboBoxModel);

companyComboBox.setSelectedIndex(0);

}

/***
 * 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() {

jTabbedPane1 = new javax.swing.JTabbedPane();

jPanel1 = new javax.swing.JPanel();

```

```
jPanel2 = new javax.swing.JPanel();
supplierNoField = new javax.swing.JTextField();
nameField = new javax.swing.JTextField();
nicField = new javax.swing.JTextField();
emailField = new javax.swing.JTextField();
contactNoField = new javax.swing.JTextField();
addressField = new javax.swing.JTextField();
cityField = new javax.swing.JTextField();
districtComboBox = new javax.swing.JComboBox<>();
provinceComboBox = new javax.swing.JComboBox<>();
companyComboBox = new javax.swing.JComboBox<>();
jLabel1 = new javax.swing.JLabel();
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();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
registerButton = new javax.swing.JButton();
updateButton = new javax.swing.JButton();
removeButton = new javax.swing.JButton();
newButton = new javax.swing.JButton();
jSeparator2 = new javax.swing.JSeparator();
jPanel4 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
supplierTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();
```

```
jPanel3 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

supplierTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Supplier Manager");

setResizable(false);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

supplierNoField.setEditable(false);

contactNoField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        contactNoFieldKeyPressed(evt);

    }

});

jLabel1.setText("Supplier No");

jLabel2.setText("Name");

jLabel3.setText("NIC");

jLabel4.setText("Email");

jLabel5.setText("Contact No");
```

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

jLabel7.setText("City");

jLabel8.setText("District");

jLabel9.setText("Province");

jLabel10.setText("Company");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
registerButton.setText("Register");
registerButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        registerButtonActionPerformed(evt);
    }
});
```



```
updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
updateButton.setText("Update");
updateButton.setEnabled(false);
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});
```



```
removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
removeButton.setText("Remove");
removeButton.setEnabled(false);
removeButton.addActionListener(new java.awt.event.ActionListener() {
```

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    removeButtonActionPerformed(evt);  
}  
});  
  
newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
newButton.setText("New");  
newButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        newButtonActionPerformed(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()  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jSeparator1)  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE))  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 399,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel2Layout.createSequentialGroup()

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

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

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

.addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)

.addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(nicField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

.addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
```

```
        .addComponent(companyComboBox, javax.swing.GroupLayout.Alignment.TRAILING, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(supplierNoField)))

        .addGap(0, 0, Short.MAX_VALUE))

    .addContainerGap()

);

jPanel2Layout.setVerticalGroup

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

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(30, 30, 30)

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

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(supplierNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nameField)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nicField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(emailField)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(contactNoField)
```

```
.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(addressField)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(cityField)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(districtComboBox)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(provinceComboBox)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(companyComboBox)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

.addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(47, Short.MAX_VALUE))

);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Suppliers", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

supplierTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Supplier No", "Name", "Email", "Contact No", "City", "Company"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}
});

});
```

```
supplierTable.setRowHeight(25);

supplierTable.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        supplierTableMouseClicked(evt);

    }

});

jScrollPane1.setViewportView(supplierTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));

keywordField.setText("Search");

keywordField.setToolTipText("Search");

keywordField.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFocusLost(evt);

    }

});

keywordField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldKeyReleased(evt);

    }

});

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(0, 0, 0)
        .addContainerGap())
)
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    )
    .addContainerGap())
);

javafx.scene.layout.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(0, 0, 0)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(0, 0, 0)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(0, 0, 0)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, 0)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
            .addGap(0, 0, 0)
            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        )
        .addGap(0, 0, 0)
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    )
    .addGap(0, 0, 0)
)
```

```

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jSeparator2)
.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addContainerGap())
);

jTabbedPane1.addTab("Supplier Details", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));
keywordFieldFull.setText("Search");
keywordFieldFull.setToolTipText("Search");
keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {
public void focusGained(java.awt.event.FocusEvent evt) {
keywordFieldFullFocusGained(evt);
}
public void focusLost(java.awt.event.FocusEvent evt) {
keywordFieldFullFocusLost(evt);
}
});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {
public void keyReleased(java.awt.event.KeyEvent evt) {

```

```
keywordFieldFullKeyReleased(evt);

}

});

supplierTableFull.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},  

new String [] {
"Supplier No", "Name", "NIC", "Email ", "Contact No", "Address", "City", "District", "Province", "Country"
}
) {  

boolean[] canEdit = new boolean [] {
false, false, false, false, false, false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}

});  

supplierTableFull.setRowHeight(25);  

jScrollPane3.setViewportView(supplierTableFull);  

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
.addContainerGap()
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jScrollPane3)

```

```
.addGroup(jPanel3Layout.createSequentialGroup()
    .addGap(0, 700, Short.MAX_VALUE)
    .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 473,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())
);

jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 559, Short.MAX_VALUE)
        .addContainerGap())
);

jTabbedPane1.addTab("Suppliers", jPanel3);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTabbedPane1)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTabbedPane1)
);
pack();
setLocationRelativeTo(null);
```

```
}// </editor-fold>

private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {
    if (keywordFieldFull.getText().equals("Search")) {
        keywordFieldFull.setText("");
        keywordFieldFull.setForeground(new Color(0, 0, 0));
    }
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordFieldFull.getText().equals("")) {
        keywordFieldFull.setText("Search");
        keywordFieldFull.setForeground(new Color(153, 153, 153));
    }
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String keywordFull = keywordFieldFull.getText();
    supplierModel.loadSupplierTableFull(keywordFull, supplierTableModelFull);
}
```

```
private void contactNoFieldKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String contactNo = contactNoField.getText();
    int length = contactNo.length();
    char c = evt.getKeyChar();
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {
        if (length < 10) {
            contactNoField.setEditable(true);

```

```

} else {
    contactNoField.setEditable(false);
}

} else {
    if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==
KeyEvent.VK_DELETE) {

        contactNoField.setEditable(true);

    } else {

        contactNoField.setEditable(false);

    }
}
}
}

```

```

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

    // TODO add your handling code here:

    String name = nameField.getText();

    String nic = nicField.getText();

    String email = emailField.getText();

    String contactNo = contactNoField.getText();

    String address = addressField.getText();

    String city = cityField.getText();

    int distictNo = districtComboBox.getSelectedIndex() + 1;

    int provinceNo = provinceComboBox.getSelectedIndex() + 1;

    int companyNo = companyComboBox.getSelectedIndex() + 1;

    logger.debug("Check validations");

    if (name.equals("")) {

        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

        nameField.requestFocus(true);

    } else if (contactNo.equals("")) {

        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,

```

```

JOptionPane.ERROR_MESSAGE);

contactNoField.requestFocus(true);

} else if (city.equals("")) {

JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

cityField.requestFocus(true);

} else if (email.length() > 0) {

String PATTERN = "^[a-zA-Z0-9./]{0,50}[@][a-zA-Z]{0,10}{.}[a-zA-Z]{0,5}$";

Pattern p = Pattern.compile(PATTERN);

Matcher m = p.matcher(email);

if (!m.matches()) {

JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

emailField.requestFocus(true);

} else {

String result = supplierModel.registerSupplier(name, nic, email, contactNo, address, city, distictNo,
provinceNo, companyNo);

if (result.equals("Success")) {

JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

supplierModel.loadSupplierTable("", supplierTableModel);

supplierModel.loadSupplierTableFull("", supplierTableModelFull);

logger.debug("Message: " + result);

} else {

JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

} else {

String result = supplierModel.registerSupplier(name, nic, email, contactNo, address, city, distictNo,
provinceNo, companyNo);

if (result.equals("Success")) {

JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

supplierModel.loadSupplierTable("", supplierTableModel);
}
}
}

```

```

supplierModel.loadSupplierTableFull("", supplierTableModelFull);

logger.debug("Message: " + result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: " + result);

}

}

```

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

```

// TODO add your handling code here:

String supplierNo = supplierNoField.getText();

String name = nameField.getText();

String nic = nicField.getText();

String email = emailField.getText();

String contactNo = contactNoField.getText();

String address = addressField.getText();

String city = cityField.getText();

int distictNo = districtComboBox.getSelectedIndex() + 1;

int provinceNo = provinceComboBox.getSelectedIndex() + 1;

int companyNo = companyComboBox.getSelectedIndex() + 1;

```

```
logger.debug("Check validations");
```

```

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {

```

```

JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

cityField.requestFocus(true);

} else if (email.length() > 0) {

String PATTERN = "^[a-zA-Z0-9./]{0,50}@{0,10}.[a-zA-Z]{0,5}$";
Pattern p = Pattern.compile(PATTERN);
Matcher m = p.matcher(email);

if (!m.matches()) {

JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

emailField.requestFocus(true);

} else {

String result = supplierModel.updateSupplier(supplierNo, name, nic, email, contactNo, address, city,
districtNo, provinceNo, companyNo);

if (result.equals("Success")) {

JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

supplierModel.loadSupplierTable("", supplierTableModel);

supplierModel.loadSupplierTableFull("", supplierTableModelFull);

logger.debug("Message: " + result);

} else {

JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

} else {

String result = supplierModel.updateSupplier(supplierNo, name, nic, email, contactNo, address, city, districtNo,
provinceNo, companyNo);

if (result.equals("Success")) {

JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

supplierModel.loadSupplierTable("", supplierTableModel);

supplierModel.loadSupplierTableFull("", supplierTableModelFull);

logger.debug("Message: " + result);

} else {

```

```
JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

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

    // TODO add your handling code here:

    String supplierNo = supplierNoField.getText();

    int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);

    if (response == JOptionPane.YES_OPTION) {

        String result = supplierModel.removeSupplier(supplierNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            supplierModel.loadSupplierTable("", supplierTableModel);

            supplierModel.loadSupplierTableFull("", supplierTableModelFull);

            logger.debug("Message: " + result);

        } else {

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

        clearFields();
    }
}
```

```
private void newButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    clearFields();  
}  
  
private void supplierTableMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    clearFields();  
  
    int row = supplierTable.getSelectedRow();  
  
    int supplierNo = (int) supplierTableModel.getValueAt(row, 0);  
  
    Object[] supplierData = supplierModel.getSelectedSupplierData(supplierNo);  
  
    supplierNoField.setText(new String(supplierData[0] + ""));  
  
    nameField.setText((String) supplierData[1]);  
  
    nicField.setText((String) supplierData[2]);  
  
    emailField.setText((String) supplierData[3]);  
  
    contactNoField.setText((String) supplierData[4]);  
  
    addressField.setText((String) supplierData[5]);  
  
    cityField.setText((String) supplierData[6]);  
  
    districtComboBox.setSelectedIndex((int) supplierData[7] - 1);  
  
    provinceComboBox.setSelectedIndex((int) supplierData[8] - 1);  
  
    companyComboBox.setSelectedIndex((int) supplierData[9] - 1);  
  
    if (supplierNoField.getText().length() > 0) {  
        registerButton.setEnabled(false);  
        updateButton.setEnabled(true);  
        removeButton.setEnabled(true);  
    } else {  
        // registerButton.setEnabled(true);  
        // updateButton.setEnabled(false);  
    }  
}
```

```
//      removeButton.setEnabled(false);

}

}

private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("Search")) {

    keywordField.setText("");

    keywordField.setForeground(new Color(0, 0, 0));

}

}

private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("")) {

    keywordField.setText("Search");

    keywordField.setForeground(new Color(153, 153, 153));

}

}

private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String keyword = keywordField.getText();

supplierModel.loadSupplierTable(keyword, supplierTableModel);

}

private void clearFields() {

supplierNoField.setText("");

nameField.setText("");

nicField.setText("");

emailField.setText("");
```

```
contactNoField.setText("");
addressField.setText("");
cityField.setText("");
districtComboBox.setSelectedIndex(0);
provinceComboBox.setSelectedIndex(0);
companyComboBox.setSelectedIndex(0);
```

```
registerButton.setEnabled(true);
updateButton.setEnabled(false);
removeButton.setEnabled(false);
```

```
logger.debug("clearFields");
```

```
}
```

```
/**
```

```
* @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 {
```

```
UIManager.setLookAndFeel(new FlatIntelliJLaf());
```

```
UIManager.put("Table.showHorizontalLines", true);
```

```
UIManager.put("Table.showVerticalLines", true);
```

```
UIManager.put("ScrollBar.showButtons", true);
```

```
UIManager.put("ScrollBar.width", 16);
```

```
} catch (Exception ex) {
```

```
System.err.println("Failed to initialize LaF");
```

```
}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//         if ("Windows".equals(info.getName())) {

//             javax.swing.UIManager.setLookAndFeel(info.getClassName());

//             break;

//         }

//     }

// } catch (ClassNotFoundException ex) {

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

// } catch (InstantiationException ex) {

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

// } catch (IllegalAccessException ex) {

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

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

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

// }

// //</editor-fold>

// //</editor-fold>

// //</editor-fold>

// //</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable {

    public void run() {

        new SupplierManager().setVisible(true);

    }

});
```

```
// Variables declaration - do not modify

private javax.swing.JTextField addressField;

private javax.swing.JTextField cityField;

private javax.swing.JComboBox<String> companyComboBox;

private javax.swing.JTextField contactNoField;

private javax.swing.JComboBox<String> districtComboBox;

private javax.swing.JTextField emailField;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

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.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JTextField keywordField;

private javax.swing.JTextField keywordFieldFull;

private javax.swing.JTextField nameField;

private javax.swing.JButton newButton;
```

```
private javax.swing.JTextField nicField;

private javax.swing.JComboBox<String> provinceComboBox;

private javax.swing.JButton registerButton;

private javax.swing.JButton removeButton;

private javax.swing.JTextField supplierNoField;

private javax.swing.JTable supplierTable;

private javax.swing.JTable supplierTableFull;

private javax.swing.JButton updateButton;

// End of variables declaration

}
```

SupplierModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
 */
public class SupplierModel {
```

```
private Connection dbConnection;

static Logger logger = Logger.getLogger(SupplierModel.class.getName());

public SupplierModel() {

    dbConnection = database.dbConnection.getConnection();

}

public String registerSupplier(String name, String nic, String email, String contactNo, String address, String city, int distictNo, int provinceNo, int companyNo) {

    String query = "INSERT INTO supplier(supplier_name, supplier_nic, supplier_email, supplier_contact_no,
supplier_address, supplier_city, district_no, province_no, company_no, supplier_status)"

        + "VALUE(?,?,?,?,?,?,?,?,?,?)";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setString(1, name);

        psm.setString(2, nic);

        psm.setString(3, email);

        psm.setString(4, contactNo);

        psm.setString(5, address);

        psm.setString(6, city);

        psm.setInt(7, distictNo);

        psm.setInt(8, provinceNo);

        psm.setInt(9, companyNo);

        psm.setInt(10, 1);

        psm.execute();

        logger.debug("Query execution done");

        logger.debug("return result");

        return "Success";

    } catch (Exception e) {
```

```
e.printStackTrace();

logger.error("Exception: " + e.getMessage());
logger.error("return result");

return "Error !";

}

}

public String updateSupplier(String supplierNo, String name, String nic, String email, String contactNo, String address, String city, int distictNo, int provinceNo, int companyNo) {

    String query = "UPDATE supplier SET supplier_name = ?, supplier_nic = ?, supplier_email = ?, supplier_contact_no = ?, supplier_address = ?, supplier_city = ?, "
        + "district_no = ?, province_no = ?, company_no = ? WHERE supplier_no = ?";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, name);
    psm.setString(2, nic);
    psm.setString(3, email);
    psm.setString(4, contactNo);
    psm.setString(5, address);
    psm.setString(6, city);
    psm.setInt(7, distictNo);
    psm.setInt(8, provinceNo);
    psm.setInt(9, companyNo);
    psm.setString(10, supplierNo);

    psm.execute();

    logger.debug("Query execution done");
    logger.debug("return result");

    return "Success";

} catch (Exception e) {

    e.printStackTrace();
}
```

```

        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public String removeSupplier(String supplierNo) {
    String query = "UPDATE supplier SET supplier_status = ? WHERE supplier_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, 0);
        psm.setString(2, supplierNo);
        psm.execute();
        logger.debug("Query execution done");
        logger.debug("return result");
        return "Success";
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public void loadSupplierTable(String keyword, DefaultTableModel supplierTableModel) {
    String query = "SELECT * FROM supplier INNER JOIN district INNER JOIN province INNER JOIN company "
        + "ON supplier.district_no = district.district_no AND supplier.province_no = province.province_no AND "
        + "supplier.company_no = company.company_no "
        + "WHERE (supplier_no LIKE ? OR supplier_name LIKE ? OR supplier_contact_no LIKE ? OR supplier_city LIKE "
        + "? OR company_name LIKE ?) AND supplier_status = ? ORDER BY supplier_no";
}

```

```
logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keyword + "%");

    psm.setString(2, "%" + keyword + "%");

    psm.setString(3, "%" + keyword + "%");

    psm.setString(4, "%" + keyword + "%");

    psm.setString(5, "%" + keyword + "%");

    psm.setInt(6, 1);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    supplierTableModel.setRowCount(0);

    while (rs.next()) {

        int supplierNo = rs.getInt("supplier_no");

        String name = rs.getString("supplier_name");

        String email = rs.getString("supplier_email");

        String contactNo = rs.getString("supplier_contact_no");

        String city = rs.getString("supplier_city");

        String company = rs.getString("company_name");

        rowData = new Object[]{supplierNo, name, email, contactNo, city, company};

        supplierTableModel.addRow(rowData);

        logger.debug("loadSupplierTable [addRow]");

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}
```

```
public Object[] getSelectedSupplierData(int supplierNo) {  
    String query = "SELECT * FROM supplier WHERE supplier_no = ?";  
    logger.debug("SQL query String: " + query);  
  
    try {  
        PreparedStatement psm = dbConnection.prepareStatement(query);  
        psm.setInt(1, supplierNo);  
        ResultSet rs = psm.executeQuery();  
        logger.debug("Query execution done");  
        Object[] rowData = null;  
        while (rs.next()) {  
            int supplierNoo = rs.getInt("supplier_no");  
            String name = rs.getString("supplier_name");  
            String nic = rs.getString("supplier_nic");  
            String email = rs.getString("supplier_email");  
            String contactNo = rs.getString("supplier_contact_no");  
            String address = rs.getString("supplier_address");  
            String city = rs.getString("supplier_city");  
            int districtNo = rs.getInt("district_no");  
            int provinceNo = rs.getInt("province_no");  
            int companyNo = rs.getInt("company_no");  
            rowData = new Object[]{supplierNoo, name, nic, email, contactNo, address, city, districtNo, provinceNo, companyNo};  
            logger.debug("return rowData");  
            return rowData;  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
        logger.error("Exception: " + e.getMessage());  
    }  
    logger.debug("return null");  
}
```

```

return null;
}

public void loadSupplierTableFull(String keywordFull, DefaultTableModel supplierTableModelFull) {
    String query = "SELECT * FROM supplier INNER JOIN district INNER JOIN province INNER JOIN company "
        + "ON supplier.district_no = district.district_no AND supplier.province_no = province.province_no AND "
        + "supplier.company_no = company.company_no "
        + "WHERE (supplier_no LIKE ? OR supplier_name LIKE ? OR supplier_contact_no LIKE ? OR supplier_city LIKE "
        + "? OR company_name LIKE ?) AND supplier_status = ? ORDER BY supplier_no";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setString(1, "%" + keywordFull + "%");
        psm.setString(2, "%" + keywordFull + "%");
        psm.setString(3, "%" + keywordFull + "%");
        psm.setString(4, "%" + keywordFull + "%");
        psm.setString(5, "%" + keywordFull + "%");
        psm.setInt(6, 1);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");
        Object[] rowData = null;
        supplierTableModelFull.setRowCount(0);
        while (rs.next()) {
            int supplierNo = rs.getInt("supplier_no");
            String name = rs.getString("supplier_name");
            String nic = rs.getString("supplier_nic");
            String email = rs.getString("supplier_email");
            String contactNo = rs.getString("supplier_contact_no");
            String address = rs.getString("supplier_address");
            String city = rs.getString("supplier_city");
            String district = rs.getString("district_name");

```

```
String province = rs.getString("province_name");

String company = rs.getString("company_name");

rowData = new Object[]{supplierNo, name, nic, email, contactNo, address, city, district, province, company};

supplierTableModelFull.addRow(rowData);

logger.debug("loadSupplierTableFull [addRow]");

}

} catch (Exception e) {

e.printStackTrace();

logger.error("Exception: " + e.getMessage());

}

}

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Customer Manager Customer Details

GUI Number : 25

Use case Name: Customer Management

03. Brief Description of the GUI:

The main objective of this interface is to register customers, update customers and remove customers.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Customer Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JTable | Display customers | During a search operation, the user can find nearly similar customer lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert customer details | To enter or display the details needed to register a customer. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register customers, Update customers, Remove customers & Clear text fields | Buttons are needed to register customers, update customers, remove customers, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)



GUI REPORT

(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-02

Batch Name : Griffins

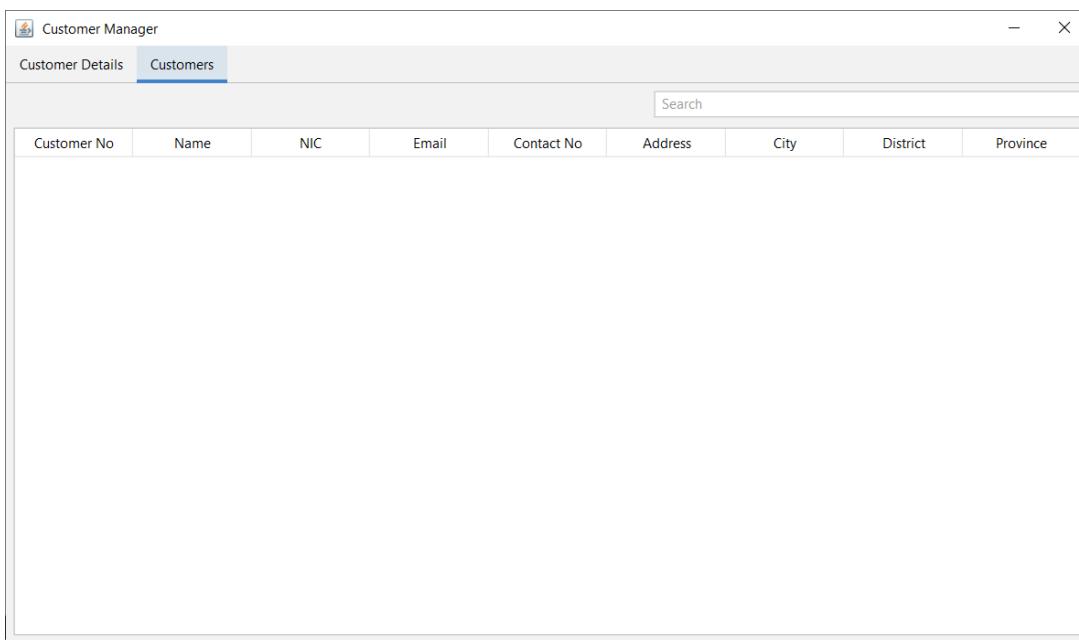
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Customer Manager Customers

GUI Number : 26

Use case Name: Customers Management



03. Brief Description of the GUI:

The main objective of this interface is to display and search the customers registered in the system.

04. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Customer Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display customers | During a search operation, the user can find nearly similar customer lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

CustomerManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.ComboBoxLoadModel;
import model.CustomerModel;
import org.apache.log4j.Logger;
```

```
/**  
*  
* @author EZIO AUDITORY  
*/  
  
public class CustomerManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form CustomerManager  
     */  
  
    private CustomerModel customerModel;  
    private ComboBoxLoadModel comboBoxLoadModel;  
    static Logger logger = Logger.getLogger(CustomerManager.class.getName());  
  
    DefaultTableModel customerTableModel;  
    DefaultTableModel customerTableModelFull;  
  
    DefaultComboBoxModel districtComboBoxModel;  
    DefaultComboBoxModel provinceComboBoxModel;  
  
    public CustomerManager() {  
        initComponents();  
        logger.debug("Access to CustomerManager.java");  
  
        customerTableModel = (DefaultTableModel) customerTable.getModel();  
        customerTableModelFull = (DefaultTableModel) customerTableFull.getModel();  
  
        customerModel = new CustomerModel();  
        comboBoxLoadModel = new ComboBoxLoadModel();  
  
        customerModel.loadCustomerTable("", customerTableModel);
```

```
customerModel.loadCustomerTableFull("", customerTableModelFull);

Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();
districtComboBoxModel = new DefaultComboBoxModel(vDistrict);
districtComboBox.setModel(districtComboBoxModel);
districtComboBox.setSelectedIndex(0);

Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();
provinceComboBoxModel = new DefaultComboBoxModel(vProvince);
provinceComboBox.setModel(provinceComboBoxModel);
provinceComboBox.setSelectedIndex(0);

}

/***
 * 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() {

jTabbedPane1 = new javax.swing.JTabbedPane();
jPanel1 = new javax.swing.JPanel();
jPanel2 = new javax.swing.JPanel();
customerNoField = new javax.swing.JTextField();
nameField = new javax.swing.JTextField();
nicField = new javax.swing.JTextField();
emailField = new javax.swing.JTextField();
contactNoField = new javax.swing.JTextField();
addressField = new javax.swing.JTextField();
}
```

```
cityField = new javax.swing.JTextField();

districtComboBox = new javax.swing.JComboBox<>();

provinceComboBox = new javax.swing.JComboBox<>();

jLabel1 = new javax.swing.JLabel();

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();

jLabel9 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

registerButton = new javax.swing.JButton();

updateButton = new javax.swing.JButton();

removeButton = new javax.swing.JButton();

newButton = new javax.swing.JButton();

jSeparator2 = new javax.swing.JSeparator();

jPanel4 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

customerTable = new javax.swing.JTable();

keywordField = new javax.swing.JTextField();

jPanel3 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

customerTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Customer Manager");

setResizable(false);
```

```
jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

customerNoField.setEditable(false);

contactNoField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        contactNoFieldKeyPressed(evt);

    }

});;

jLabel1.setText("Customer No");

jLabel2.setText("Name");

jLabel3.setText("NIC");

jLabel4.setText("Email");

jLabel5.setText("Contact No");

jLabel6.setText("Address");

jLabel7.setText("City");

jLabel8.setText("District");

jLabel9.setText("Province");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
registerButton.setText("Register");

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

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

        registerButtonActionPerformed(evt);

    }

});

updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

updateButton.setText("Update");

updateButton.setEnabled(false);

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

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

        updateButtonActionPerformed(evt);

    }

});

removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

removeButton.setText("Remove");

removeButton.setEnabled(false);

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

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

        removeButtonActionPerformed(evt);

    }

});

newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

newButton.setText("New");

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

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

        newButtonActionPerformed(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()  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jSeparator1)  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                        javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                        javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 399,  
                            javax.swing.GroupLayout.PREFERRED_SIZE)  
                    .addGroup(jPanel2Layout.createSequentialGroup()  
                        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)
            .addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(nicField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(customerNoField)))
        .addGap(0, 0, Short.MAX_VALUE)))

        .addContainerGap()

    );

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(30, 30, 30)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(customerNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    )
);
```

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

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

.addComponent(nameField)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nicField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(emailField)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(contactNoField)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(addressField)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(cityField)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(districtComboBox)
```

```

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Customers", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

customerTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},

},

```

```
new String [] {
    "Customer No", "Name", "Email", "Contact No", "City"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

customerTable.setRowHeight(25);
customerTable.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        customerTableMouseClicked(evt);
    }
});

jScrollPane1.setViewportView(customerTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));
keywordField.setText("Search");
keywordField.setToolTipText("Search");
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        keywordFieldFocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        keywordFieldFocusLost(evt);
    }
});
});
```

```
keywordField.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        keywordFieldKeyReleased(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addGroup(jPanel4Layout.createSequentialGroup()  
                .addGroup(jPanel4Layout.createSequentialGroup()  
                    .addGap(0, 0, Short.MAX_VALUE)  
                    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
                .addContainerGap())  
);  
jPanel4Layout.setVerticalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap(29, Short.MAX_VALUE)  
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
            javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,  
            javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addContainerGap())  
);
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(5, 5, 5)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(5, 5, 5)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(5, 5, 5)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(5, 5, 5)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(5, 5, 5)
                .addComponent(jSeparator2)
                .addGap(5, 5, 5)
                .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(5, 5, 5)
                .addContainerGap())
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(5, 5, 5)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(5, 5, 5)
                .addComponent(jSeparator2)
                .addGap(5, 5, 5)
                .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(5, 5, 5)
                .addContainerGap())))
        .addContainerGap())
    );
jTabbedPane1.addTab("Customer Details", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));
keywordFieldFull.setText("Search");
```

```

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFullKeyReleased(evt);

    }

});

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

    new String [] {
        "Customer No", "Name", "NIC", "Email", "Contact No", "Address", "City", "District", "Province"
    }
} {

    boolean[] canEdit = new boolean [] {

        false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {

        return canEdit [columnIndex];

    }

});

```

```
customerTableFull.setRowHeight(25);

jScrollPane3.setViewportView(customerTableFull);

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(10, 10, 10)
            .addComponent(jScrollPane3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 473, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
            )
            .addGap(10, 10, 10)
        )
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 560, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
    )
);

jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 473, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
            )
            .addGap(10, 10, 10)
        )
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 560, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
    )
);

jTabbedPane1.addTab("Customers", jPanel3);

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

```
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);  
  
layout.setVerticalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);  
  
pack();  
setLocationRelativeTo(null);  
}// </editor-fold>
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {  
  
    if (keywordFieldFull.getText().equals("Search")) {  
  
        keywordFieldFull.setText("");  
        keywordFieldFull.setForeground(new Color(0, 0, 0));  
    }  
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (keywordFieldFull.getText().equals("")) {  
  
        keywordFieldFull.setText("Search");  
        keywordFieldFull.setForeground(new Color(153, 153, 153));  
    }  
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keywordFull = keywordFieldFull.getText();
```

```
customerModel.loadCustomerTableFull(keywordFull, customerTableModelFull);

}

private void contactNoFieldKeyPressed(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    String contactNo = contactNoField.getText();

    int length = contactNo.length();

    char c = evt.getKeyChar();

    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {

        if (length < 10) {

            contactNoField.setEditable(true);

        } else {

            contactNoField.setEditable(false);

        }

    } else {

        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() == KeyEvent.VK_DELETE) {

            contactNoField.setEditable(true);

        } else {

            contactNoField.setEditable(false);

        }

    }

}

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

    // TODO add your handling code here:

    String name = nameField.getText();

    String nic = nicField.getText();

    String email = emailField.getText();

    String contactNo = contactNoField.getText();

    String address = addressField.getText();
```

```

String city = cityField.getText();

int distictNo = districtComboBox.getSelectedIndex() + 1;

int provinceNo = provinceComboBox.getSelectedIndex() + 1;

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}[@][a-zA-Z]{0,10}[.][a-zA-Z]{0,5}$";

    Pattern p = Pattern.compile(PATTERN);

    Matcher m = p.matcher(email);

    if (!m.matches()) {

        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = customerModel.registerCustomer(name, nic, email, contactNo, address, city, distictNo,
provinceNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            customerModel.loadCustomerTable("", customerTableModel);

            customerModel.loadCustomerTableFull("", customerTableModelFull);

            logger.debug("Message: " + result);

        } else {


```

```

JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

} else {

    String result = customerModel.registerCustomer(name, nic, email, contactNo, address, city, distictNo,
provinceNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        customerModel.loadCustomerTable("", customerTableModel);

        customerModel.loadCustomerTableFull("", customerTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

```

```

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

    // TODO add your handling code here:

    String customerNo = customerNoField.getText();

    String name = nameField.getText();

    String nic = nicField.getText();

    String email = emailField.getText();

    String contactNo = contactNoField.getText();

    String address = addressField.getText();

    String city = cityField.getText();

    int distictNo = districtComboBox.getSelectedIndex() + 1;

    int provinceNo = provinceComboBox.getSelectedIndex() + 1;

```

```

logger.debug("Check validations");

if (name.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.?[a-zA-Z]{0,5}$";
    Pattern p = Pattern.compile(PATTERN);
    Matcher m = p.matcher(email);

    if (!m.matches()) {
        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = customerModel.updateCustomer(customerNo, name, nic, email, contactNo, address, city,
distictNo, provinceNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            customerModel.loadCustomerTable("", customerTableModel);

            customerModel.loadCustomerTableFull("", customerTableModelFull);

            logger.debug("Message: " + result);

        } else {

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

    }

}

```

```
    } else {

        String result = customerModel.updateCustomer(customerNo, name, nic, email, contactNo, address, city,
distictNo, provinceNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            customerModel.loadCustomerTable("", customerTableModel);

            customerModel.loadCustomerTableFull("", customerTableModelFull);

            logger.debug("Message: " + result);

        } else {

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

    }

}
```

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

```
    // TODO add your handling code here:

    String customerNo = customerNoField.getText();
```

```
    int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
```

```
    if (response == JOptionPane.YES_OPTION) {

        String result = customerModel.removeCustomer(customerNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            customerModel.loadCustomerTable("", customerTableModel);

            customerModel.loadCustomerTableFull("", customerTableModelFull);

            logger.debug("Message: " + result);

        } else {

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

    }
```

```
    }

    clearFields();

}

}

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

    // TODO add your handling code here:

    clearFields();

}

private void customerTableMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

    clearFields();

    int row = customerTable.getSelectedRow();

    int customerNo = (int) customerTableModel.getValueAt(row, 0);

    Object[] customerData = customerModel.getSelectedCustomerData(customerNo);

    customerNoField.setText(new String(customerData[0] + ""));

    nameField.setText((String) customerData[1]);

    nicField.setText((String) customerData[2]);

    emailField.setText((String) customerData[3]);

    contactNoField.setText((String) customerData[4]);

    addressField.setText((String) customerData[5]);

    cityField.setText((String) customerData[6]);

    districtComboBox.setSelectedIndex((int) customerData[7] - 1);

    provinceComboBox.setSelectedIndex((int) customerData[8] - 1);

    if (customerNoField.getText().length() > 0) {

        registerButton.setEnabled(false);

    }

}
```

```
updateButton.setEnabled(true);

removeButton.setEnabled(true);

} else {

//    registerButton.setEnabled(true);

//    updateButton.setEnabled(false);

//    removeButton.setEnabled(false);

}

}

private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("Search")) {

keywordField.setText("");

keywordField.setForeground(new Color(0, 0, 0));

}

}

private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("")) {

keywordField.setText("Search");

keywordField.setForeground(new Color(153, 153, 153));

}

}

private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String keyword = keywordField.getText();

customerModel.loadCustomerTable(keyword, customerTableModel);

}
```

```
private void clearFields() {  
    customerNoField.setText("");  
    nameField.setText("");  
    nicField.setText("");  
    emailField.setText("");  
    contactNoField.setText("");  
    addressField.setText("");  
    cityField.setText("");  
    districtComboBox.setSelectedIndex(0);  
    provinceComboBox.setSelectedIndex(0);  
  
    registerButton.setEnabled(true);  
    updateButton.setEnabled(false);  
    removeButton.setEnabled(false);  
  
    logger.debug("clearFields");  
}  
 
```

```
/**  
 * @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 {  
        UIManager.setLookAndFeel(new FlatIntelliJLaf());  
        UIManager.put("Table.showHorizontalLines", true);  
        UIManager.put("Table.showVerticalLines", true);  
    }  
}
```

```

UIManager.put("ScrollBar.showButtons", true);
UIManager.put("ScrollBar.width", 16);
} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}
// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(CustomerManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(CustomerManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(CustomerManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(CustomerManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// }
// //
```

```

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

```

```
});  
}  
  
// Variables declaration - do not modify  
  
private javax.swing.JTextField addressField;  
private javax.swing.JTextField cityField;  
private javax.swing.JTextField contactNoField;  
private javax.swing.JTextField customerNoField;  
private javax.swing.JTable customerTable;  
private javax.swing.JTable customerTableFull;  
private javax.swing.JComboBox<String> districtComboBox;  
private javax.swing.JTextField emailField;  
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.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JSeparator jSeparator1;  
private javax.swing.JSeparator jSeparator2;  
private javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JTextField keywordField;
```

```
private javax.swing.JTextField keywordFieldFull;  
private javax.swing.JTextField nameField;  
private javax.swing.JButton newButton;  
private javax.swing.JTextField nicField;  
private javax.swing.JComboBox<String> provinceComboBox;  
private javax.swing.JButton registerButton;  
private javax.swing.JButton removeButton;  
private javax.swing.JButton updateButton;  
// End of variables declaration  
  
}
```

CustomerModel.java

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package model;  
  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.Vector;  
import javax.swing.table.DefaultTableModel;  
import org.apache.log4j.Logger;  
  
  
/**  
 *  
 * @author EZIO AUDITORY  
 */
```

```
public class CustomerModel {  
  
    private Connection dbConnection;  
  
    static Logger logger = Logger.getLogger(CustomerModel.class.getName());  
  
    public CustomerModel(){  
        dbConnection = database.dbConnection.getConnection();  
    }  
  
    public String registerCustomer(String name, String nic, String email, String contactNo, String address, String city, int distictNo, int provinceNo) {  
        String query = "INSERT INTO customer(customer_name, customer_nic, customer_email, customer_contact_no, customer_address, customer_city, district_no, province_no, customer_status)"  
        + "VALUE(?,?,?,?,?,?)";  
        logger.debug("SQL query String: " + query);  
  
        try {  
            PreparedStatement psm = dbConnection.prepareStatement(query);  
            psm.setString(1, name);  
            psm.setString(2, nic);  
            psm.setString(3, email);  
            psm.setString(4, contactNo);  
            psm.setString(5, address);  
            psm.setString(6, city);  
            psm.setInt(7, distictNo);  
            psm.setInt(8, provinceNo);  
            psm.setInt(9, 1);  
            psm.execute();  
            logger.debug("Query execution done");  
            logger.debug("return result");  
            return "Success";  
        } catch (Exception e) {  
        }  
    }  
}
```

```

e.printStackTrace();

logger.error("Exception: " + e.getMessage());

logger.error("return result");

return "Error !";

}

}

public String updateCustomer(String customerNo, String name, String nic, String email, String contactNo, String address, String city, int distictNo, int provinceNo) {

    String query = "UPDATE customer SET customer_name = ?, customer_nic = ?, customer_email = ?, customer_contact_no = ?, customer_address = ?, customer_city = ?, "
        + "district_no = ?, province_no = ? WHERE customer_no = ?";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, name);

    psm.setString(2, nic);

    psm.setString(3, email);

    psm.setString(4, contactNo);

    psm.setString(5, address);

    psm.setString(6, city);

    psm.setInt(7, distictNo);

    psm.setInt(8, provinceNo);

    psm.setString(9, customerNo);

    psm.execute();

    logger.debug("Query execution done");

    logger.debug("return result");

    return "Success";

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());
}

```

```

        logger.error("return result");

        return "Error !";

    }

}

public String removeCustomer(String customerNo) {

    String query = "UPDATE customer SET customer_status = ? WHERE customer_no = ?";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setInt(1, 0);

        psm.setString(2, customerNo);

        psm.execute();

        logger.debug("Query execution done");

        logger.debug("return result");

        return "Success";

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

        logger.error("return result");

        return "Error !";

    }

}

public void loadCustomerTable(String keyword, DefaultTableModel customerTableModel) {

    String query = "SELECT * FROM customer INNER JOIN district INNER JOIN province "

        + "ON customer.district_no = district.district_no AND customer.province_no = province.province_no "

        + "WHERE (customer_no LIKE ? OR customer_name LIKE ? OR customer_contact_no LIKE ? OR
customer_city LIKE ?) AND customer_status = ? ORDER BY customer_no";

    logger.debug("SQL query String: " + query);
}

```

```

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keyword + "%");
    psm.setString(2, "%" + keyword + "%");
    psm.setString(3, "%" + keyword + "%");
    psm.setString(4, "%" + keyword + "%");
    psm.setInt(5, 1);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    customerTableModel.setRowCount(0);

    while(rs.next()){

        int customerNo = rs.getInt("customer_no");

        String name = rs.getString("customer_name");

        String email = rs.getString("customer_email");

        String contactNo = rs.getString("customer_contact_no");

        String city = rs.getString("customer_city");

        rowData = new Object[]{customerNo,name,email,contactNo,city};

        customerTableModel.addRow(rowData);

        logger.debug("loadCustomerTable [addRow]");

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

```

```

public Object[] getSelectedCustomerData(int customerNo) {

    String query = "SELECT * FROM customer WHERE customer_no = ?";

    logger.debug("SQL query String: " + query);

```

```

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setInt(1, customerNo);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    while(rs.next()){

        int customerNoo = rs.getInt("customer_no");

        String name = rs.getString("customer_name");

        String nic = rs.getString("customer_nic");

        String email = rs.getString("customer_email");

        String contactNo = rs.getString("customer_contact_no");

        String address = rs.getString("customer_address");

        String city = rs.getString("customer_city");

        int districtNo = rs.getInt("district_no");

        int provinceNo = rs.getInt("province_no");

        rowData = new Object[]{customerNoo,name,nic,email,contactNo,address,city,districtNo,provinceNo};

        logger.debug("return rowData");

        return rowData;

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}

public void loadCustomerTableFull(String keywordFull, DefaultTableModel customerTableModelFull) {

String query = "SELECT * FROM customer INNER JOIN district INNER JOIN province "

```

```

+ "ON customer.district_no = district.district_no AND customer.province_no = province.province_no "
+ "WHERE (customer_no LIKE ? OR customer_name LIKE ? OR customer_contact_no LIKE ? OR
customer_city LIKE ?) AND customer_status = ? ORDER BY customer_no";

logger.debug("SQL query String: " + query);

try {

PreparedStatement psm = dbConnection.prepareStatement(query);

psm.setString(1, "%" + keywordFull + "%");
psm.setString(2, "%" + keywordFull + "%");
psm.setString(3, "%" + keywordFull + "%");
psm.setString(4, "%" + keywordFull + "%");
psm.setInt(5, 1);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

customerTableModelFull.setRowCount(0);

while(rs.next()){

    int customerNo = rs.getInt("customer_no");

    String name = rs.getString("customer_name");

    String nic = rs.getString("customer_nic");

    String email = rs.getString("customer_email");

    String contactNo = rs.getString("customer_contact_no");

    String address = rs.getString("customer_address");

    String city = rs.getString("customer_city");

    String district = rs.getString("district_name");

    String province = rs.getString("province_name");

    rowData = new Object[]{customerNo, name, nic, email, contactNo, address, city, district, province};

    customerTableModelFull.addRow(rowData);

    logger.debug("loadCustomerTableFull [addRow]");

}

} catch (Exception e) {

```

```
e.printStackTrace();  
logger.error("Exception: " + e.getMessage());  
}  
}  
  
}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Company Manager Company Details

GUI Number : 27

Use case Name: Company Management

05. Brief Description of the GUI:

The main objective of this interface is to register companies, update companies and remove companies.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Company Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display companies | During a search operation, the user can find nearly similar company lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert company details | To enter or display the details needed to register a company. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register companies, Update companies, Remove companies & Clear text fields | Buttons are needed to register companies, update companies, remove companies, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)



GUI REPORT

(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-02

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Company Manager Companies

GUI Number : 28

Use case Name: Company Management

The screenshot shows a Java Swing application window titled "Company Manager". It has two tabs at the top: "Company Details" and "Companies", with "Companies" being the active tab. Below the tabs is a search bar labeled "Search". The main area contains a table with the following columns: Company No, Name, Email, Contact No, Address, City, District, Province, and Country. There is one row of data with the following values:

| Company No | Name | Email | Contact No | Address | City | District | Province | Country |
|------------|--------------|-------|------------|---------|------|--------------|--------------|--------------|
| 1 | Not Selected | null | 000000000 | null | null | Not Selected | Not Selected | Not Selected |

05. Brief Description of the GUI:

The main objective of this interface is to display and search the companies registered in the system.

06. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Company Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display companies | During a search operation, the user can find nearly similar company lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



CompanyManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.ComboBoxLoadModel;
import model.CompanyModel;
import org.apache.log4j.Logger;
```

```
/**  
 *  
 * @author EZIO AUDITORY  
 */  
  
public class CompanyManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form CompanyManager  
     */  
  
    private CompanyModel companyModel;  
    private ComboBoxLoadModel comboBoxLoadModel;  
    static Logger logger = Logger.getLogger(CompanyManager.class.getName());  
  
    DefaultTableModel companyTableModelFull;  
    DefaultTableModel companyTableModel;  
  
    DefaultComboBoxModel districtComboBoxModel;  
    DefaultComboBoxModel provinceComboBoxModel;  
    DefaultComboBoxModel countryComboBoxModel;  
  
    public CompanyManager() {  
        initComponents();  
        logger.debug("Access to CompanyManager.java");  
  
        companyTableModel = (DefaultTableModel) companyTable.getModel();  
        companyTableModelFull = (DefaultTableModel) companyTableFull.getModel();  
  
        companyModel = new CompanyModel();  
        comboBoxLoadModel = new ComboBoxLoadModel();
```

```
companyModel.loadCompanyTable("", companyTableModel);  
companyModel.loadCompanyTableFull("", companyTableModelFull);
```

```
Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();  
districtComboBoxModel = new DefaultComboBoxModel(vDistrict);  
districtComboBox.setModel(districtComboBoxModel);  
districtComboBox.setSelectedIndex(0);
```

```
Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();  
provinceComboBoxModel = new DefaultComboBoxModel(vProvince);  
provinceComboBox.setModel(provinceComboBoxModel);  
provinceComboBox.setSelectedIndex(0);
```

```
Vector vCountry = comboBoxLoadModel.countryComboBoxLoad();  
countryComboBoxModel = new DefaultComboBoxModel(vCountry);  
countryComboBox.setModel(countryComboBoxModel);  
countryComboBox.setSelectedIndex(0);
```

```
}
```

```
/**
```

```
* 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() {
```

```
jTabbedPane1 = new javax.swing.JTabbedPane();  
jPanel1 = new javax.swing.JPanel();  
jPanel2 = new javax.swing.JPanel();
```

```
companyNoField = new javax.swing.JTextField();

nameField = new javax.swing.JTextField();

emailField = new javax.swing.JTextField();

contactNoField = new javax.swing.JTextField();

addressField = new javax.swing.JTextField();

cityField = new javax.swing.JTextField();

districtComboBox = new javax.swing.JComboBox<>();

provinceComboBox = new javax.swing.JComboBox<>();

countryComboBox = new javax.swing.JComboBox<>();

jLabel1 = new javax.swing.JLabel();

jLabel2 = 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();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

registerButton = new javax.swing.JButton();

updateButton = new javax.swing.JButton();

removeButton = new javax.swing.JButton();

newButton = new javax.swing.JButton();

jSeparator2 = new javax.swing.JSeparator();

jPanel4 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

companyTable = new javax.swing.JTable();

keywordField = new javax.swing.JTextField();

jPanel3 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();
```

```
companyTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Company Manager");
setResizable(false);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

companyNoField.setEditable(false);

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

jLabel1.setText("Company No");

jLabel2.setText("Name");

jLabel4.setText("Email");

jLabel5.setText("Contact No");

jLabel6.setText("Address");

jLabel7.setText("City");

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

```
jLabel9.setText("Province");

jLabel10.setText("Country");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
registerButton.setText("Register");
registerButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        registerButtonActionPerformed(evt);
    }
});;

updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
updateButton.setText("Update");
updateButton.setEnabled(false);
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});;

removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
removeButton.setText("Remove");
removeButton.setEnabled(false);
removeButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        removeButtonActionPerformed(evt);
    }
});;
```

```
newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newButton.setText("New");
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newButtonActionPerformed(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()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jSeparator1)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
                javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
                javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
                .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
                javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 399,
                javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        )
    )
);
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)
            .addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(countryComboBox, javax.swing.GroupLayout.Alignment.TRAILING, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(companyNoField)))))

        .addGap(0, 0, Short.MAX_VALUE)))

        .addContainerGap()

    );

jPanel2Layout.setVerticalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel2Layout.createSequentialGroup()  
        .addGap(30, 30, 30)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,  
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
            .addComponent(companyNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(nameField)  
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(emailField)  
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(contactNoField)  
            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(addressField)  
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
            .addComponent(cityField)  
            .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
                javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(districtComboBox)
    .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(provinceComboBox)
        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(countryComboBox)
        .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
```

```
java.awt.Color(0, 0, 0)), "Companies", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

companyTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Company No", "Name", "Email", "Contact No", "City", "Country"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}

});

companyTable.setRowHeight(25);

companyTable.addMouseListener(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
companyTableMouseClicked(evt);
}
});

jScrollPane1.setViewportView(companyTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));
keywordField.setText("Search");
keywordField.setToolTipText("Search");
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
public void focusGained(java.awt.event.FocusEvent evt) {
    keywordFieldFocusGained(evt);
}

public void focusLost(java.awt.event.FocusEvent evt) {
    keywordFieldFocusLost(evt);
}

});

keywordField.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        keywordFieldKeyReleased(evt);
    }
});

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGroup(jPanel4Layout.createSequentialGroup()
                    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel4Layout.createSequentialGroup()
                            .addGap(0, 0, Short.MAX_VALUE)
                            .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
                                javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addContainerGap())
                .addContainerGap())
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addContainerGap())))
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addContainerGap())
    ).addGap(0, 0, Short.MAX_VALUE)
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addContainerGap())))
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addContainerGap())
    ).addGap(0, 0, Short.MAX_VALUE)
);
```

```
.addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap()

);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jSeparator2)

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

```

```
.addContainerGap()

);

jTabbedPane1.addTab("Company Details", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFullKeyReleased(evt);

    }

});

companyTableFull.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "Company No", "Name", "Email", "Contact No ", "Address", "City", "District", "Province", "Country"
        }
    },
    new String [] {

    }
) {

    boolean[] canEdit = new boolean [] {
```



```
.addContainerGap())
);

jTabbedPane1.addTab("Companies", jPanel3);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTabbedPane1)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTabbedPane1)
);
pack();
setLocationRelativeTo(null);
}// </editor-fold>

private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {
    if (keywordFieldFull.getText().equals("Search")) {
        keywordFieldFull.setText("");
        keywordFieldFull.setForeground(new Color(0, 0, 0));
    }
}

private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordFieldFull.getText().equals("")) {
        keywordFieldFull.setText("Search");
    }
}
```

```
keywordFieldFull.setForeground(new Color(153, 153, 153));  
}  
}  
  
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keywordFull = keywordFieldFull.getText();  
  
    companyModel.loadCompanyTableFull(keywordFull, companyTableModelFull);  
}  
  
private void contactNoFieldKeyPressed(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String contactNo = contactNoField.getText();  
  
    int length = contactNo.length();  
  
    char c = evt.getKeyChar();  
  
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
  
        if (length < 10) {  
  
            contactNoField.setEditable(true);  
  
        } else {  
  
            contactNoField.setEditable(false);  
  
        }  
    } else {  
  
        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==  
KeyEvent.VK_DELETE) {  
  
            contactNoField.setEditable(true);  
  
        } else {  
  
            contactNoField.setEditable(false);  
  
        }  
    }  
}
```

```
private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String name = nameField.getText();  
  
    String email = emailField.getText();  
  
    String contactNo = contactNoField.getText();  
  
    String address = addressField.getText();  
  
    String city = cityField.getText();  
  
    int districtNo = districtComboBox.getSelectedIndex() + 1;  
  
    int provinceNo = provinceComboBox.getSelectedIndex() + 1;  
  
    int countryNo = countryComboBox.getSelectedIndex() + 1;  
  
  
    logger.debug("Check validations");  
  
    if (name.equals("")) {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        nameField.requestFocus(true);  
  
    } else if (contactNo.equals("")) {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        contactNoField.requestFocus(true);  
  
    } else if (city.equals("")) {  
  
        JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,  
        JOptionPane.ERROR_MESSAGE);  
  
        cityField.requestFocus(true);  
  
    } else if (email.length() > 0) {  
  
        String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.?[a-zA-Z]{0,5}$";  
  
        Pattern p = Pattern.compile(PATTERN);  
  
        Matcher m = p.matcher(email);  
  
        if (!m.matches()) {  
  
            JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);  
  
            emailField.requestFocus(true);  
  
        } else {  
  
        }  
    }  
}
```

```

String result = companyModel.registerCompany(name, email, contactNo, address, city, distictNo,
provinceNo, countryNo);

if (result.equals("Success")) {

    JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

    companyModel.loadCompanyTable("", companyTableModel);

    companyModel.loadCompanyTableFull("", companyTableModelFull);

    logger.debug("Message: " + result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: " + result);

}

}

} else {

    String result = companyModel.registerCompany(name, email, contactNo, address, city, distictNo, provinceNo,
countryNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        companyModel.loadCompanyTable("", companyTableModel);

        companyModel.loadCompanyTableFull("", companyTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

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

// TODO add your handling code here:

String companyNo = companyNoField.getText();

String name = nameField.getText();

String email = emailField.getText();

```

```

String contactNo = contactNoField.getText();

String address = addressField.getText();

String city = cityField.getText();

int distictNo = districtComboBox.getSelectedIndex() + 1;

int provinceNo = provinceComboBox.getSelectedIndex() + 1;

int countryNo = countryComboBox.getSelectedIndex() + 1;

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.\\[a-zA-Z]{0,5}$";

    Pattern p = Pattern.compile(PATTERN);

    Matcher m = p.matcher(email);

    if (!m.matches()) {

        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = companyModel.updateCompany(companyNo, name, email, contactNo, address, city,
distictNo, provinceNo, countryNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            companyModel.loadCompanyTable("", companyTableModel);

```

```

companyModel.loadCompanyTableFull("", companyTableModelFull);

logger.debug("Message: " + result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: " + result);

}

}

} else {

    String result = companyModel.updateCompany(companyNo, name, email, contactNo, address, city, distictNo,
provinceNo, countryNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        companyModel.loadCompanyTable("", companyTableModel);

        companyModel.loadCompanyTableFull("", companyTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

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

    // TODO add your handling code here:

    String companyNo = companyNoField.getText();

    int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);

    if (response == JOptionPane.YES_OPTION) {

        String result = companyModel.removeCompany(companyNo);

        if (result.equals("Success")) {

```

```
JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

companyModel.loadCompanyTable("", companyTableModel);

companyModel.loadCompanyTableFull("", companyTableModelFull);

logger.debug("Message: " + result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: " + result);

}

clearFields();

}

}

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

    // TODO add your handling code here:

    clearFields();

}

private void companyTableMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

    clearFields();

    int row = companyTable.getSelectedRow();

    int companyNo = (int) companyTableModel.getValueAt(row, 0);

    Object[] companyData = companyModel.getSelectedCompanyData(companyNo);

    companyNoField.setText(new String(companyData[0] + ""));

    nameField.setText((String) companyData[1]);

    emailField.setText((String) companyData[2]);

    contactNoField.setText((String) companyData[3]);

    addressField.setText((String) companyData[4]);

}
```

```
cityField.setText((String) companyData[5]);

districtComboBox.setSelectedIndex((int) companyData[6] - 1);

provinceComboBox.setSelectedIndex((int) companyData[7] - 1);

countryComboBox.setSelectedIndex((int) companyData[8] - 1);

if (companyNoField.getText().length() > 0) {

    registerButton.setEnabled(false);

    updateButton.setEnabled(true);

    removeButton.setEnabled(true);

} else {

    //    registerButton.setEnabled(true);

    //    updateButton.setEnabled(false);

    //    removeButton.setEnabled(false);

}

}
```

```
private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("Search")) {

    keywordField.setText("");

    keywordField.setForeground(new Color(0, 0, 0));

}

}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("")) {

    keywordField.setText("Search");

    keywordField.setForeground(new Color(153, 153, 153));

}

}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String keyword = keywordField.getText();  
  
    companyModel.loadCompanyTable(keyword, companyTableModel);  
}  
  
  
private void clearFields() {  
  
    companyNoField.setText("");  
  
    nameField.setText("");  
  
    emailField.setText("");  
  
    contactNoField.setText("");  
  
    addressField.setText("");  
  
    cityField.setText("");  
  
    districtComboBox.setSelectedIndex(0);  
  
    provinceComboBox.setSelectedIndex(0);  
  
    countryComboBox.setSelectedIndex(0);  
  
  
    registerButton.setEnabled(true);  
  
    updateButton.setEnabled(false);  
  
    removeButton.setEnabled(false);  
  
  
    logger.debug("clearFields");  
}  
  
  
/**  
 * @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 {
    UIManager.setLookAndFeel(new FlatIntelliJLaf());
    UIManager.put("Table.showHorizontalLines", true);
    UIManager.put("Table.showVerticalLines", true);
    UIManager.put("ScrollBar.showButtons", true);
    UIManager.put("ScrollBar.width", 16);
} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(CompanyManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(CompanyManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(CompanyManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(CompanyManager.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// }
// </editor-fold>
```

```
//</editor-fold>
//</editor-fold>
//</editor-fold>

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

}

// Variables declaration - do not modify
private javax.swing.JTextField addressField;
private javax.swing.JTextField cityField;
private javax.swing.JTextField companyNoField;
private javax.swing.JTable companyTable;
private javax.swing.JTable companyTableFull;
private javax.swing.JTextField contactNoField;
private javax.swing.JComboBox<String> countryComboBox;
private javax.swing.JComboBox<String> districtComboBox;
private javax.swing.JTextField emailField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
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.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTextField keywordField;
private javax.swing.JTextField keywordFieldFull;
private javax.swing.JTextField nameField;
private javax.swing.JButton newButton;
private javax.swing.JComboBox<String> provinceComboBox;
private javax.swing.JButton registerButton;
private javax.swing.JButton removeButton;
private javax.swing.JButton updateButton;
// End of variables declaration
}


```

CompanyModel.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class CompanyModel {

    private Connection dbConnection;
    static Logger logger = Logger.getLogger(CompanyModel.class.getName());

    public CompanyModel() {
        dbConnection = database.dbConnection.getConnection();
    }

    public String registerCompany(String name, String email, String contactNo, String address, String city, int distictNo, int provinceNo, int countryNo) {
        String query = "INSERT INTO company(company_name, company_email, company_contact_no, company_address, company_city, district_no, province_no, country_no, company_status)"
                + "VALUE(?,?,?,?,?,?,?,?,?)";
        logger.debug("SQL query String: " + query);

        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            psm.setString(1, name);
            psm.setString(2, email);
            psm.setString(3, contactNo);
            psm.setString(4, address);
            psm.setString(5, city);
            psm.setInt(6, distictNo);
            psm.setInt(7, provinceNo);
            psm.setInt(8, countryNo);
            psm.setInt(9, distictNo);
        } catch (SQLException e) {
            logger.error("Error occurred while inserting company details: " + e.getMessage());
        }
    }
}
```

```
psm.setInt(7, provinceNo);
psm.setInt(8, countryNo);
psm.setInt(9, 1);
psm.execute();
logger.debug("Query execution done");
logger.debug("return result");
return "Success";
} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
    logger.error("return result");
    return "Error !";
}
}
```

```
public String updateCompany(String companyNo, String name, String email, String contactNo, String address,
String city, int distictNo, int provinceNo, int countryNo) {
    String query = "UPDATE company SET company_name = ?, company_email = ?, company_contact_no = ?,
company_address = ?, company_city = ?, district_no = ?, "
    + "province_no = ?, country_no = ? WHERE company_no = ?";
    logger.debug("SQL query String: " + query);

try {
    PreparedStatement psm = dbConnection.prepareStatement(query);
    psm.setString(1, name);
    psm.setString(2, email);
    psm.setString(3, contactNo);
    psm.setString(4, address);
    psm.setString(5, city);
    psm.setInt(6, distictNo);
    psm.setInt(7, provinceNo);
    psm.setInt(8, countryNo);
}
```

```
psm.setString(9, companyNo);

psm.execute();

logger.debug("Query execution done");

logger.debug("return result");

return "Success";

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

    logger.error("return result");

    return "Error !";

}

}

public String removeCompany(String companyNo) {

    String query = "UPDATE company SET company_status = ? WHERE company_no = ?";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setInt(1, 0);

        psm.setString(2, companyNo);

        psm.execute();

        logger.debug("Query execution done");

        logger.debug("return result");

        return "Success";

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

        logger.error("return result");

        return "Error !";

    }

}
```

```
}

public void loadCompanyTable(String keyword, DefaultTableModel companyTableModel) {

    String query = "SELECT * FROM company INNER JOIN district INNER JOIN province INNER JOIN country "

        + "ON company.district_no = district.district_no AND company.province_no = province.province_no AND
company.country_no = country.country_no "

        + "WHERE (company_no LIKE ? OR company_name LIKE ? OR company_contact_no LIKE ? OR company_city
LIKE ? OR country_name LIKE ?) AND company_status = ? ORDER BY company_no";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setString(1, "%" + keyword + "%");
        psm.setString(2, "%" + keyword + "%");
        psm.setString(3, "%" + keyword + "%");
        psm.setString(4, "%" + keyword + "%");
        psm.setString(5, "%" + keyword + "%");
        psm.setInt(6, 1);

        ResultSet rs = psm.executeQuery();

        logger.debug("Query execution done");

        Object[] rowData = null;

        companyTableModel.setRowCount(0);

        while (rs.next()) {

            int companyNo = rs.getInt("company_no");

            String name = rs.getString("company_name");

            String email = rs.getString("company_email");

            String contactNo = rs.getString("company_contact_no");

            String city = rs.getString("company_city");

            String country = rs.getString("country_name");

            rowData = new Object[]{companyNo, name, email, contactNo, city, country};

            companyTableModel.addRow(rowData);

            logger.debug("loadCompanyTable [addRow]");
        }
    }
}
```

```

    }

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

}

public Object[] getSelectedCompanyData(int companyNo) {
    String query = "SELECT * FROM company WHERE company_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, companyNo);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");

        Object[] rowData = null;
        while (rs.next()) {
            int companyNoo = rs.getInt("company_no");
            String name = rs.getString("company_name");
            String email = rs.getString("company_email");
            String contactNo = rs.getString("company_contact_no");
            String address = rs.getString("company_address");
            String city = rs.getString("company_city");
            int districtNo = rs.getInt("district_no");
            int provinceNo = rs.getInt("province_no");
            int countryNo = rs.getInt("country_no");

            rowData = new Object[]{companyNoo, name, email, contactNo, address, city, districtNo, provinceNo, countryNo};

            logger.debug("return rowData");
        }
        return rowData;
    }
}

```

```

    }

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
}

public void loadCompanyTableFull(String keywordFull, DefaultTableModel companyTableModelFull) {
    String query = "SELECT * FROM company INNER JOIN district INNER JOIN province INNER JOIN country "
        + "ON company.district_no = district.district_no AND company.province_no = province.province_no AND
company.country_no = country.country_no "
        + "WHERE (company_no LIKE ? OR company_name LIKE ? OR company_contact_no LIKE ? OR company_city
LIKE ? OR country_name LIKE ?) AND company_status = ? ORDER BY company_no";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setString(1, "%" + keywordFull + "%");
        psm.setString(2, "%" + keywordFull + "%");
        psm.setString(3, "%" + keywordFull + "%");
        psm.setString(4, "%" + keywordFull + "%");
        psm.setString(5, "%" + keywordFull + "%");
        psm.setInt(6, 1);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");

        Object[] rowData = null;
        companyTableModelFull.setRowCount(0);
        while (rs.next()) {
            int companyNo = rs.getInt("company_no");
            String name = rs.getString("company_name");

```

```
String email = rs.getString("company_email");

String contactNo = rs.getString("company_contact_no");

String address = rs.getString("company_address");

String city = rs.getString("company_city");

String district = rs.getString("district_name");

String province = rs.getString("province_name");

String country = rs.getString("country_name");

rowData = new Object[]{companyNo, name, email, contactNo, address, city, district, province, country};

companyTableModelFull.addRow(rowData);

logger.debug("loadCompanyTableFull [addRow]");

}

} catch (Exception e) {

e.printStackTrace();

logger.error("Exception: " + e.getMessage());

}

}

}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Transporter Manager Transporter Details

GUI Number : 29

Use case Name: Transporter Management

07. Brief Description of the GUI:

The main objective of this interface is to register transporters, update transporters and remove transporters.

08. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|---|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Transporter Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JTable | Display transporters | During a search operation, the user can find nearly similar transporter lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert transporter details | To enter or display the details needed to register a transporter. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register transporters, Update transporters, Remove transporters & Clear text fields | Buttons are needed to register transporters, update transporters, remove transporters, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)



GUI REPORT

(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-02

Batch Name : Griffins

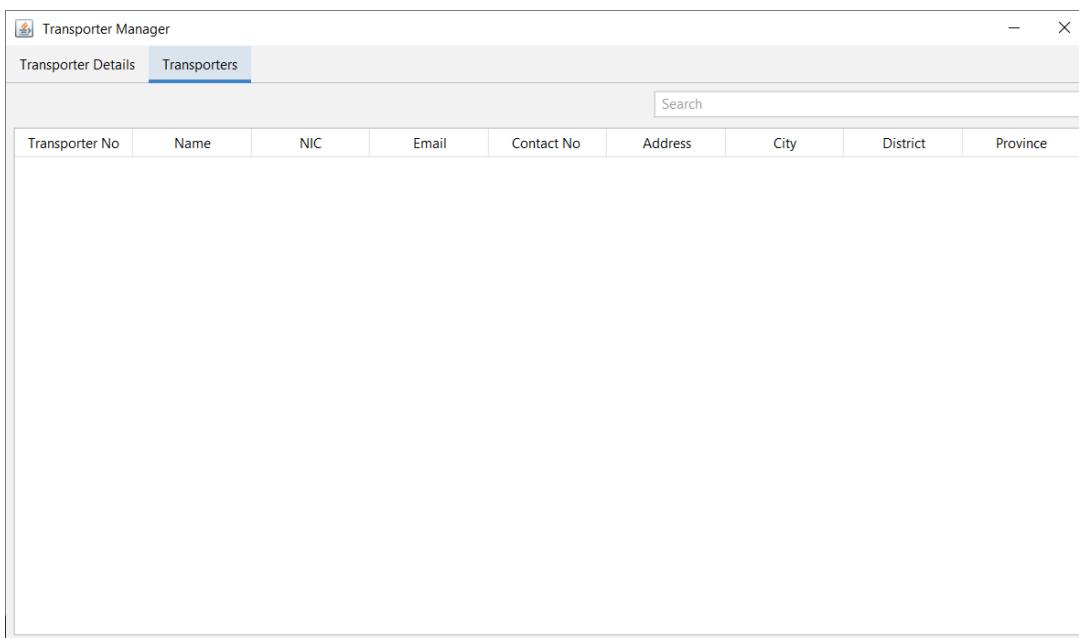
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Transporter Manager Transporters

GUI Number : 30

Use case Name: Transporter Management



07. Brief Description of the GUI:

The main objective of this interface is to display and search the transporters registered in the system.

08. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|---|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the Transporter Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanels makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display transporters | During a search operation, the user can find nearly similar transporter lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

| | | |
|--------------------|--|--|
| <p>Apply Theme</p> | | <p>com.formdev.flatlaf.FlatIntelliJLaf (FlatIntelliJLaf)</p> |
|--------------------|--|--|

TransporterManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.ComboBoxLoadModel;
import model.TransporterModel;
import org.apache.log4j.Logger;
```

```
/**  
*  
* @author EZIO AUDITORY  
*/  
  
public class TransporterManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form TransporterManager  
     */  
  
    private TransporterModel transporterModel;  
    private ComboBoxLoadModel comboBoxLoadModel;  
    static Logger logger = Logger.getLogger(TransporterManager.class.getName());  
  
    DefaultTableModel transporterTableModel;  
    DefaultTableModel transporterTableModelFull;  
  
    DefaultComboBoxModel districtComboBoxModel;  
    DefaultComboBoxModel provinceComboBoxModel;  
  
    public TransporterManager() {  
        initComponents();  
        logger.debug("Access to TransporterManager.java");  
  
        transporterTableModel = (DefaultTableModel) transporterTable.getModel();  
        transporterTableModelFull = (DefaultTableModel) transporterTableFull.getModel();  
  
        transporterModel = new TransporterModel();  
        comboBoxLoadModel = new ComboBoxLoadModel();  
  
        transporterModel.loadTransporterTable("", transporterTableModel);
```

```
transporterModel.loadTransporterTableFull("", transporterTableModelFull);

Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();
districtComboBoxModel = new DefaultComboBoxModel(vDistrict);
districtComboBox.setModel(districtComboBoxModel);
districtComboBox.setSelectedIndex(0);

Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();
provinceComboBoxModel = new DefaultComboBoxModel(vProvince);
provinceComboBox.setModel(provinceComboBoxModel);
provinceComboBox.setSelectedIndex(0);

}

/***
 * 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() {

jTabbedPane1 = new javax.swing.JTabbedPane();
jPanel1 = new javax.swing.JPanel();
jPanel2 = new javax.swing.JPanel();
transporterNoField = new javax.swing.JTextField();
nameField = new javax.swing.JTextField();
nicField = new javax.swing.JTextField();
emailField = new javax.swing.JTextField();
contactNoField = new javax.swing.JTextField();
addressField = new javax.swing.JTextField();
}
```

```
cityField = new javax.swing.JTextField();

districtComboBox = new javax.swing.JComboBox<>();

provinceComboBox = new javax.swing.JComboBox<>();

jLabel1 = new javax.swing.JLabel();

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();

jLabel9 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

registerButton = new javax.swing.JButton();

updateButton = new javax.swing.JButton();

removeButton = new javax.swing.JButton();

newButton = new javax.swing.JButton();

jSeparator2 = new javax.swing.JSeparator();

jPanel4 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

transporterTable = new javax.swing.JTable();

keywordField = new javax.swing.JTextField();

jPanel3 = new javax.swing.JPanel();

keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

transporterTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Transporter Manager");

setResizable(false);
```

```
jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

transporterNoField.setEditable(false);

contactNoField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyPressed(java.awt.event.KeyEvent evt) {

        contactNoFieldKeyPressed(evt);

    }

});;

jLabel1.setText("Transporter No");

jLabel2.setText("Name");

jLabel3.setText("NIC");

jLabel4.setText("Email");

jLabel5.setText("Contact No");

jLabel6.setText("Address");

jLabel7.setText("City");

jLabel8.setText("District");

jLabel9.setText("Province");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
registerButton.setText("Register");

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

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

        registerButtonActionPerformed(evt);

    }

});

updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

updateButton.setText("Update");

updateButton.setEnabled(false);

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

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

        updateButtonActionPerformed(evt);

    }

});

removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

removeButton.setText("Remove");

removeButton.setEnabled(false);

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

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

        removeButtonActionPerformed(evt);

    }

});

newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

newButton.setText("New");

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

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

        newButtonActionPerformed(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()  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jSeparator1)  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 125,  
                    javax.swing.GroupLayout.PREFERRED_SIZE))  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 399,  
                        javax.swing.GroupLayout.PREFERRED_SIZE)  
                    .addGroup(jPanel2Layout.createSequentialGroup()  
                        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                javax.swing.GroupLayout.PREFERRED_SIZE)  
                            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,  
                                javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)
            .addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(nicField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)
            .addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(transporterNoField)))
        .addGap(0, 0, Short.MAX_VALUE)))

        .addContainerGap()

    );

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(30, 30, 30)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(transporterNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
    )
);
```

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

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

.addComponent(nameField)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(nicField)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(emailField)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(contactNoField)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(addressField)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(cityField)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(districtComboBox)
```

```

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Transporters", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

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

}, 
```

```
new String [] {
    "Customer No", "Name", "Email", "Contact No", "City"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

transporterTable.setRowHeight(25);
transporterTable.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        transporterTableMouseClicked(evt);
    }
});

jScrollPane1.setViewportView(transporterTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));
keywordField.setText("Search");
keywordField.setToolTipText("Search");
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        keywordFieldFocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        keywordFieldFocusLost(evt);
    }
});
});
```

```
keywordField.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        keywordFieldKeyReleased(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addGroup(jPanel4Layout.createSequentialGroup()  
                .addGroup(jPanel4Layout.createSequentialGroup()  
                    .addGap(0, 0, Short.MAX_VALUE)  
                    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
                .addContainerGap())  
);  
jPanel4Layout.setVerticalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap(29, Short.MAX_VALUE)  
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,  
            javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,  
            javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addContainerGap())  
);
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(10, 10, 10)
        .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(10, 10, 10)
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(10, 10, 10)
        .addComponent(jPanel1Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addComponent(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                );
                jPanel1Layout.setVerticalGroup(
                    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jPanel1Layout.createSequentialGroup()
                            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jSeparator2)
                                .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            )
                            .addContainerGap()
                        )
                    )
                )
            )
        )
    )
);
jTabbedPane1.addTab("Transporter Details", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));
keywordFieldFull.setText("Search");
```

```

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFullKeyReleased(evt);

    }

});

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

        new String [] {
            "Transporter No", "Name", "NIC", "Email", "Contact No ", "Address", "City", "District", "Province"
        }
    },
    new String [] {

    }
) {

    boolean[] canEdit = new boolean [] {

        false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {

        return canEdit [columnIndex];

    }

});

```

```
transporterTableFull.setRowHeight(25);

jScrollPane3.setViewportView(transporterTableFull);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane3)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(0, 700, Short.MAX_VALUE)
                .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 473,
                    javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap())
);
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 560, Short.MAX_VALUE)
        .addContainerGap())
);
jTabbedPane1.addTab("Transporters", jPanel3);

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

```
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);  
  
layout.setVerticalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addComponent(jTabbedPane1)  
);  
  
pack();  
setLocationRelativeTo(null);  
}// </editor-fold>
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {  
  
    if (keywordFieldFull.getText().equals("Search")) {  
  
        keywordFieldFull.setText("");  
        keywordFieldFull.setForeground(new Color(0, 0, 0));  
    }  
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {  
  
    // TODO add your handling code here:  
  
    if (keywordFieldFull.getText().equals("")) {  
  
        keywordFieldFull.setText("Search");  
        keywordFieldFull.setForeground(new Color(153, 153, 153));  
    }  
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {  
  
    // TODO add your handling code here:  
  
    String keywordFull = keywordFieldFull.getText();
```

```
transporterModel.loadTransporterTableFull(keywordFull, transporterTableModelFull);  
}
```

```
private void contactNoKeyPressed(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    String contactNo = contactNoField.getText();  
  
    int length = contactNo.length();  
  
    char c = evt.getKeyChar();  
  
    if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {  
  
        if (length < 10) {  
  
            contactNoField.setEditable(true);  
  
        } else {  
  
            contactNoField.setEditable(false);  
  
        }  
  
    } else {  
  
        if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==  
KeyEvent.VK_DELETE) {  
  
            contactNoField.setEditable(true);  
  
        } else {  
  
            contactNoField.setEditable(false);  
  
        }  
  
    }  
}
```

```
private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String name = nameField.getText();  
  
    String nic = nicField.getText();  
  
    String email = emailField.getText();  
  
    String contactNo = contactNoField.getText();  
  
    String address = addressField.getText();
```

```

String city = cityField.getText();

int distictNo = districtComboBox.getSelectedIndex() + 1;

int provinceNo = provinceComboBox.getSelectedIndex() + 1;

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.\\[a-zA-Z]{0,5}$";

    Pattern p = Pattern.compile(PATTERN);

    Matcher m = p.matcher(email);

    if (!m.matches()) {

        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = transporterModel.registerTransporter(name, nic, email, contactNo, address, city, distictNo,
provinceNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            transporterModel.loadTransporterTable("", transporterTableModel);

            transporterModel.loadTransporterTableFull("", transporterTableModelFull);

            logger.debug("Message: " + result);

        } else {

```

```

JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

logger.error("Error: " + result);

}

}

} else {

    String result = transporterModel.registerTransporter(name, nic, email, contactNo, address, city, distictNo,
provinceNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        transporterModel.loadTransporterTable("", transporterTableModel);

        transporterModel.loadTransporterTableFull("", transporterTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

```

```

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

    // TODO add your handling code here:

    String transporterNo = transporterNoField.getText();

    String name = nameField.getText();

    String nic = nicField.getText();

    String email = emailField.getText();

    String contactNo = contactNoField.getText();

    String address = addressField.getText();

    String city = cityField.getText();

    int distictNo = districtComboBox.getSelectedIndex() + 1;

    int provinceNo = provinceComboBox.getSelectedIndex() + 1;

```

```

logger.debug("Check validations");

if (name.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.?[a-zA-Z]{0,5}$";
    Pattern p = Pattern.compile(PATTERN);
    Matcher m = p.matcher(email);

    if (!m.matches()) {
        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = transporterModel.updateTransporter(transporterNo, name, nic, email, contactNo, address,
city, distictNo, provinceNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            transporterModel.loadTransporterTable("", transporterTableModel);

            transporterModel.loadTransporterTableFull("", transporterTableModelFull);

            logger.debug("Message: " + result);

        } else {

            JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

            logger.error("Error: " + result);

        }

    }

}

```

```

} else {

    String result = transporterModel.updateTransporter(transporterNo, name, nic, email, contactNo, address,
city, distictNo, provinceNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        transporterModel.loadTransporterTable("", transporterTableModel);

        transporterModel.loadTransporterTableFull("", transporterTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

```

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

```

// TODO add your handling code here:

String transporterNo = transporterNoField.getText();

```

```

int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);

```

```

if (response == JOptionPane.YES_OPTION) {

    String result = transporterModel.removeTransporter(transporterNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        transporterModel.loadTransporterTable("", transporterTableModel);

        transporterModel.loadTransporterTableFull("", transporterTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

```

```
}

    clearFields();

}

}

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

    // TODO add your handling code here:

    clearFields();

}

private void transporterTableMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

    clearFields();

    int row = transporterTable.getSelectedRow();

    int transporterNo = (int) transporterTableModel.getValueAt(row, 0);

    Object[] transporterData = transporterModel.getSelectedTransporterData(transporterNo);

    transporterNoField.setText(new String(transporterData[0] + ""));

    nameField.setText((String) transporterData[1]);

    nicField.setText((String) transporterData[2]);

    emailField.setText((String) transporterData[3]);

    contactNoField.setText((String) transporterData[4]);

    addressField.setText((String) transporterData[5]);

    cityField.setText((String) transporterData[6]);

    districtComboBox.setSelectedIndex((int) transporterData[7] - 1);

    provinceComboBox.setSelectedIndex((int) transporterData[8] - 1);

    if (transporterNoField.getText().length() > 0) {

        registerButton.setEnabled(false);

    }

}
```

```
updateButton.setEnabled(true);

removeButton.setEnabled(true);

} else {

// registerButton.setEnabled(true);

// updateButton.setEnabled(false);

// removeButton.setEnabled(false);

}

}
```

```
private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("Search")) {

keywordField.setText("");

keywordField.setForeground(new Color(0, 0, 0));

}

}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordField.getText().equals("")) {

keywordField.setText("Search");

keywordField.setForeground(new Color(153, 153, 153));

}

}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String keyword = keywordField.getText();

transporterModel.loadTransporterTable(keyword, transporterTableModel);

}
```

```
private void clearFields() {  
    transporterNoField.setText("");  
    nameField.setText("");  
    nicField.setText("");  
    emailField.setText("");  
    contactNoField.setText("");  
    addressField.setText("");  
    cityField.setText("");  
    districtComboBox.setSelectedIndex(0);  
    provinceComboBox.setSelectedIndex(0);  
  
    registerButton.setEnabled(true);  
    updateButton.setEnabled(false);  
    removeButton.setEnabled(false);  
  
    logger.debug("clearFields");  
}  
 
```

```
/**  
 * @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 {  
        UIManager.setLookAndFeel(new FlatIntelliJLaf());  
        UIManager.put("Table.showHorizontalLines", true);  
        UIManager.put("Table.showVerticalLines", true);  
    }  
}
```

```

        UIManager.put("ScrollBar.showButtons", true);

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {
        System.err.println("Failed to initialize LaF");
    }

//    try {

//        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//
//            if ("Windows".equals(info.getName())) {
//
//                javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//                break;
//            }
//        }
//
//    } catch (ClassNotFoundException ex) {
//
//        java.util.logging.Logger.getLogger(TransporterManager.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    } catch (InstantiationException ex) {
//
//        java.util.logging.Logger.getLogger(TransporterManager.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    } catch (IllegalAccessException ex) {
//
//        java.util.logging.Logger.getLogger(TransporterManager.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//        java.util.logging.Logger.getLogger(TransporterManager.class.getName()).log(java.util.logging.Level.SEVERE,
//null, ex);
//
//    }
//
//    //
```

```

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new TransporterManager().setVisible(true);
    }
})
```

```
});  
}  
  
// Variables declaration - do not modify  
  
private javax.swing.JTextField addressField;  
private javax.swing.JTextField cityField;  
private javax.swing.JTextField contactNoField;  
private javax.swing.JComboBox<String> districtComboBox;  
private javax.swing.JTextField emailField;  
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.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JSeparator jSeparator1;  
private javax.swing.JSeparator jSeparator2;  
private javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JTextField keywordField;  
private javax.swing.JTextField keywordFieldFull;  
private javax.swing.JTextField nameField;  
private javax.swing.JButton newButton;
```

```
private javax.swing.JTextField nicField;

private javax.swing.JComboBox<String> provinceComboBox;

private javax.swing.JButton registerButton;

private javax.swing.JButton removeButton;

private javax.swing.JTextField transporterNoField;

private javax.swing.JTable transporterTable;

private javax.swing.JTable transporterTableFull;

private javax.swing.JButton updateButton;

// End of variables declaration

}
```

TransporterModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import org.apache.log4j.Logger;

/**
 *
 *
 * @author EZIO AUDITORY
 */
public class TransporterModel {
```

```
private Connection dbConnection;

static Logger logger = Logger.getLogger(TransporterModel.class.getName());

public TransporterModel() {

    dbConnection = database.dbConnection.getConnection();

}

public String registerTransporter(String name, String nic, String email, String contactNo, String address, String city,
int distictNo, int provinceNo) {

    String query = "INSERT INTO transporter(transporter_name, transporter_nic, transporter_email,
transporter_contact_no, transporter_address, transporter_city, district_no, province_no, transporter_status)"

        + "VALUE(?,?,?,?,?, ?,?,?)";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setString(1, name);

        psm.setString(2, nic);

        psm.setString(3, email);

        psm.setString(4, contactNo);

        psm.setString(5, address);

        psm.setString(6, city);

        psm.setInt(7, distictNo);

        psm.setInt(8, provinceNo);

        psm.setInt(9, 1);

        psm.execute();

        logger.debug("Query execution done");

        logger.debug("return result");

        return "Success";

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());
    }
}
```

```

        logger.error("return result");

        return "Error !";

    }

}

public String updateTransporter(String transporterNo, String name, String nic, String email, String contactNo,
String address, String city, int distictNo, int provinceNo) {

    String query = "UPDATE transporter SET transporter_name = ?, transporter_nic = ?, transporter_email = ?,
transporter_contact_no = ?, transporter_address = ?, transporter_city = ?, "
    + "district_no = ?, province_no = ? WHERE transporter_no = ?";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, name);

    psm.setString(2, nic);

    psm.setString(3, email);

    psm.setString(4, contactNo);

    psm.setString(5, address);

    psm.setString(6, city);

    psm.setInt(7, distictNo);

    psm.setInt(8, provinceNo);

    psm.setString(9, transporterNo);

    psm.execute();

    logger.debug("Query execution done");

    logger.debug("return result");

    return "Success";

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

    logger.error("return result");

    return "Error !";
}

```

```

    }

}

public String removeTransporter(String transporterNo) {
    String query = "UPDATE transporter SET transporter_status = ? WHERE transporter_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, 0);
        psm.setString(2, transporterNo);
        psm.execute();
        logger.debug("Query execution done");
        logger.debug("return result");
        return "Success";
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("Exception: " + e.getMessage());
        logger.error("return result");
        return "Error !";
    }
}

public void loadTransporterTable(String keyword, DefaultTableModel transporterTableModel) {
    String query = "SELECT * FROM transporter INNER JOIN district INNER JOIN province "
        + "ON transporter.district_no = district.district_no AND transporter.province_no = province.province_no "
        + "WHERE (transporter_no LIKE ? OR transporter_name LIKE ? OR transporter_contact_no LIKE ? OR
    transporter_city LIKE ?) AND transporter_status = ? ORDER BY transporter_no";
    logger.debug("SQL query String: " + query);

    try {

```

```

PreparedStatement psm = dbConnection.prepareStatement(query);

psm.setString(1, "%" + keyword + "%");
psm.setString(2, "%" + keyword + "%");
psm.setString(3, "%" + keyword + "%");
psm.setString(4, "%" + keyword + "%");

psm.setInt(5, 1);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

transporterTableModel.setRowCount(0);

while (rs.next()) {

    int transporterNo = rs.getInt("transporter_no");

    String name = rs.getString("transporter_name");

    String email = rs.getString("transporter_email");

    String contactNo = rs.getString("transporter_contact_no");

    String city = rs.getString("transporter_city");

    rowData = new Object[]{transporterNo, name, email, contactNo, city};

    transporterTableModel.addRow(rowData);

    logger.debug("loadTransporterTable [addRow]");

}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

public Object[] getSelectedTransporterData(int transporterNo) {

String query = "SELECT * FROM transporter WHERE transporter_no = ?";

logger.debug("SQL query String: " + query);

try {

```

```

PreparedStatement psm = dbConnection.prepareStatement(query);

psm.setInt(1, transporterNo);

ResultSet rs = psm.executeQuery();

logger.debug("Query execution done");

Object[] rowData = null;

while (rs.next()) {

    int transporterNoo = rs.getInt("transporter_no");

    String name = rs.getString("transporter_name");

    String nic = rs.getString("transporter_nic");

    String email = rs.getString("transporter_email");

    String contactNo = rs.getString("transporter_contact_no");

    String address = rs.getString("transporter_address");

    String city = rs.getString("transporter_city");

    int districtNo = rs.getInt("district_no");

    int provinceNo = rs.getInt("province_no");

    rowData = new Object[]{transporterNoo, name, nic, email, contactNo, address, city, districtNo,
    provinceNo};

    logger.debug("return rowData");

    return rowData;
}

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
}

public void loadTransporterTableFull(String keywordFull, DefaultTableModel transporterTableModelFull) {

String query = "SELECT * FROM transporter INNER JOIN district INNER JOIN province "
+ "ON transporter.district_no = district.district_no AND transporter.province_no = province.province_no "

```

```

+ "WHERE (transporter_no LIKE ? OR transporter_name LIKE ? OR transporter_contact_no LIKE ? OR
transporter_city LIKE ?) AND transporter_status = ? ORDER BY transporter_no";

logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keywordFull + "%");
    psm.setString(2, "%" + keywordFull + "%");
    psm.setString(3, "%" + keywordFull + "%");
    psm.setString(4, "%" + keywordFull + "%");
    psm.setInt(5, 1);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    transporterTableModelFull.setRowCount(0);

    while (rs.next()) {

        int transporterNo = rs.getInt("transporter_no");

        String name = rs.getString("transporter_name");

        String nic = rs.getString("transporter_nic");

        String email = rs.getString("transporter_email");

        String contactNo = rs.getString("transporter_contact_no");

        String address = rs.getString("transporter_address");

        String city = rs.getString("transporter_city");

        String district = rs.getString("district_name");

        String province = rs.getString("province_name");

        rowData = new Object[]{transporterNo, name, nic, email, contactNo, address, city, district, province};

        transporterTableModelFull.addRow(rowData);

        logger.debug("loadTransporterTableFull [addRow]");

    }

} catch (Exception e) {

    e.printStackTrace();
}

```

```
    logger.error("Exception: " + e.getMessage());  
}  
}  
  
}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: User Manager User Details

GUI Number : 31

Use case Name: User Management

09. Brief Description of the GUI:

The main objective of this interface is to register users, update users and remove users.

10. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|--|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the User Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments & give titles to the component groups | Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels. |
| JTable | Display users | During a search operation, the user can find nearly similar user lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JLabel | Display text | To identify text fields and display text. |
| JTextField | Insert users details | To enter or display the details needed to register a user. |
| JComboBox | Display selection option list | When you have to choose one of several options, you need combo boxes to display those options and select one of them. These JComboBoxes meet that requirement. |
| JButtons | Register users, Update users, Remove users & Clear text fields | Buttons are needed to register users, update users, remove users, and clear text fields. These JButtons meet that requirement. |
| JSeparator | Make a division | To separate and display fields |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

Batch Name : Griffins

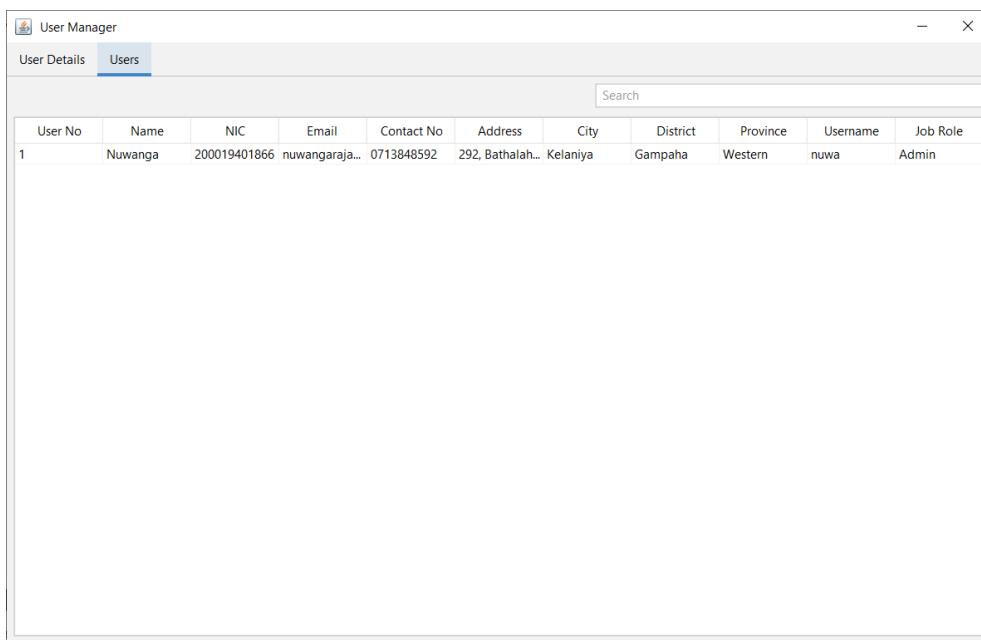
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: User Manager Users

GUI Number : 32

Use case Name: User Management



09. Brief Description of the GUI:

The main objective of this interface is to display and search the users registered in the system.

10. Justification of GUI component usage:

| Component | Usage | Justification |
|-------------|----------------------|--|
| JTabbedPane | Display menu bar | Requires access to other interfaces related to the User Manager interface. A menu bar is suitable for that. This JTabbedPane meets that requirement. |
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. Therefore all components are installed on JPanel. |
| JTable | Display users | During a search operation, the user can find nearly similar user lists according to the searching key. If the database consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

UserManager.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.event.KeyEvent;
import java.util.Vector;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.ComboBoxLoadModel;
import model.UserModel;
import org.apache.log4j.Logger;
```

```
/**  
 *  
 * @author EZIO AUDITORY  
 */  
  
public class UserManager extends javax.swing.JFrame {  
  
    /**  
     * Creates new form UserManager  
     */  
  
    private UserModel userModel;  
    private ComboBoxLoadModel comboBoxLoadModel;  
    static Logger logger = Logger.getLogger(UserManager.class.getName());  
  
    DefaultTableModel userTableModel;  
    DefaultTableModel userTableModelFull;  
  
    DefaultComboBoxModel districtComboBoxModel;  
    DefaultComboBoxModel provinceComboBoxModel;  
    DefaultComboBoxModel jobroleComboBoxModel;  
  
    public UserManager() {  
        initComponents();  
        logger.debug("Access to UserManager.java");  
  
        userTableModel = (DefaultTableModel) userTable.getModel();  
        userTableModelFull = (DefaultTableModel) userTableFull.getModel();  
  
        userModel = new UserModel();  
        comboBoxLoadModel = new ComboBoxLoadModel();
```

```
userModel.loadUserTable("", userTableModel);

userModel.loadUserTableFull("", userTableModelFull);
```

```
Vector vDistrict = comboBoxLoadModel.districtComboBoxLoad();

districtComboBoxModel = new DefaultComboBoxModel(vDistrict);

districtComboBox.setModel(districtComboBoxModel);

districtComboBox.setSelectedIndex(0);
```

```
Vector vProvince = comboBoxLoadModel.provinceComboBoxLoad();

provinceComboBoxModel = new DefaultComboBoxModel(vProvince);

provinceComboBox.setModel(provinceComboBoxModel);

provinceComboBox.setSelectedIndex(0);
```

```
Vector vCompany = comboBoxLoadModel.jobroleComboBoxLoad();

jobroleComboBoxModel = new DefaultComboBoxModel(vCompany);

jobroleComboBox.setModel(jobroleComboBoxModel);

jobroleComboBox.setSelectedIndex(0);

}
```

```
/**

* 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() {
```

```
jTabbedPane1 = new javax.swing.JTabbedPane();

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();
```

```
userNoField = new javax.swing.JTextField();

nameField = new javax.swing.JTextField();

nicField = new javax.swing.JTextField();

emailField = new javax.swing.JTextField();

contactNoField = new javax.swing.JTextField();

addressField = new javax.swing.JTextField();

cityField = new javax.swing.JTextField();

districtComboBox = new javax.swing.JComboBox<>();

provinceComboBox = new javax.swing.JComboBox<>();

jobroleComboBox = new javax.swing.JComboBox<>();

jLabel1 = new javax.swing.JLabel();

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();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

registerButton = new javax.swing.JButton();

updateButton = new javax.swing.JButton();

removeButton = new javax.swing.JButton();

newButton = new javax.swing.JButton();

jSeparator3 = new javax.swing.JSeparator();

jLabel11 = new javax.swing.JLabel();

usernameField = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

PasswordField = new javax.swing.JPasswordField();

jSeparator2 = new javax.swing.JSeparator();
```

```
jPanel4 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
userTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();
jPanel3 = new javax.swing.JPanel();
keywordFieldFull = new javax.swing.JTextField();
jScrollPane3 = new javax.swing.JScrollPane();
userTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("User Manager");
setResizable(false);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Details", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

userNoField.setEditable(false);

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

jLabel1.setText("User No");

jLabel2.setText("Name");

jLabel3.setText("NIC");
```

```
jLabel4.setText("Email");

jLabel5.setText("Contact No");

jLabel6.setText("Address");

jLabel7.setText("City");

jLabel8.setText("District");

jLabel9.setText("Province");

jLabel10.setText("Job Role");

registerButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
registerButton.setText("Register");
registerButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        registerButtonActionPerformed(evt);
    }
});

updateButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
updateButton.setText("Update");
updateButton.setEnabled(false);
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});
```

```
removeButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
removeButton.setText("Remove");
removeButton.setEnabled(false);
removeButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        removeButtonActionPerformed(evt);
    }
});
```



```
newButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
newButton.setText("New");
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newButtonActionPerformed(evt);
    }
});
```



```
jLabel11.setText("Username");
```



```
jLabel12.setText("Password");
```



```
PasswordField.setText("1234");
```



```
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(10, 10, 10)
            .addComponent(jSeparator3))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jSeparator4))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jSeparator5))))));
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jSeparator3))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jSeparator4))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jSeparator5))))));
```



```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

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

            .addComponent(cityField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

            .addComponent(addressField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

            .addComponent(contactNoField, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)

            .addComponent(emailField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

            .addComponent(nicField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

            .addComponent(nameField, javax.swing.GroupLayout.DEFAULT_SIZE, 268, Short.MAX_VALUE)

            .addComponent(districtComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(provinceComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(userNoField)))

        .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 399,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap()

);

jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
```

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

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

    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addComponent(userNoField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(nameField)

    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(nicField)

    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(emailField)

    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(contactNoField)

    .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(addressField)

    .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

    .addComponent(cityField)
```

```
.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(districtComboBox)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(provinceComboBox)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

.addComponent(usernameField)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addComponent(jLabel12, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

.addComponent(PasswordField))

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

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

.addComponent(jobroleComboBox)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(registerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(removeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(21, 21, 21))
);

jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Users", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

userTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "User No", "Name", "Email", "Contact No", "City", "Username", "Job Role"
        }
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

```

```
}

});

userTable.setRowHeight(25);

userTable.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        userTableMouseClicked(evt);

    }

});

jScrollPane1.setViewportView(userTable);

keywordField.setForeground(new java.awt.Color(153, 153, 153));

keywordField.setText("Search");

keywordField.setToolTipText("Search");

keywordField.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFocusLost(evt);

    }

});

keywordField.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldKeyReleased(evt);

    }

});

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel4Layout.createSequentialGroup()

    .addContainerGap()

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, 690,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel4Layout.createSequentialGroup()

            .addGap(0, 0, Short.MAX_VALUE)

            .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addContainerGap()

);

jPanel4Layout.setVerticalGroup

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

    .addGap(29, 29, 29)

    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, 546,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

javaswing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

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

    .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jSeparator2)

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

            .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addContainerGap())
);

jTabbedPane1.addTab("User Details", jPanel1);

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});
```

```
keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        keywordFieldFullKeyReleased(evt);
    }
});

userTableFull.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "User No", "Name", "NIC", "Email ", "Contact No", "Address", "City", "District", "Province", "Username",
            "Job Role"
        }
    },
    new String [] {
        ""
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

});

userTableFull.setRowHeight(25);

jScrollPane3.setViewportView(userTableFull);

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(0, 0, 0)
        .addContainerGap())
)
```

```
.addContainerGap()

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

.addComponent(jScrollPane3)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGap(0, 700, Short.MAX_VALUE)

.addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 473,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap())

);

jPanel3Layout.setVerticalGroup(

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

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 630, Short.MAX_VALUE)

.addContainerGap())

);

jTabbedPane1.addTab("Users", jPanel3);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup()

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

.addComponent(jTabbedPane1)

);

layout.setVerticalGroup()

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

.addComponent(jTabbedPane1)

);
```

```
pack();

 setLocationRelativeTo(null);

}// </editor-fold>

private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {

if (keywordFieldFull.getText().equals("Search")) {

    keywordFieldFull.setText("");
    keywordFieldFull.setForeground(new Color(0, 0, 0));
}

}

private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if (keywordFieldFull.getText().equals("")) {

    keywordFieldFull.setText("Search");
    keywordFieldFull.setForeground(new Color(153, 153, 153));
}

}

private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String keywordFull = keywordFieldFull.getText();
userModel.loadUserTableFull(keywordFull, userTableModelFull);
}

private void contactNoFieldKeyPressed(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String contactNo = contactNoField.getText();
int length = contactNo.length();
char c = evt.getKeyChar();
```

```

if (evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9') {
    if (length < 10) {
        contactNoField.setEditable(true);
    } else {
        contactNoField.setEditable(false);
    }
} else {
    if (evt.getExtendedKeyCode() == KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode() ==
KeyEvent.VK_DELETE) {
        contactNoField.setEditable(true);
    } else {
        contactNoField.setEditable(false);
    }
}
}

```

```

private void registerButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name = nameField.getText();
    String nic = nicField.getText();
    String email = emailField.getText();
    String contactNo = contactNoField.getText();
    String address = addressField.getText();
    String city = cityField.getText();
    int distictNo = districtComboBox.getSelectedIndex() + 1;
    int provinceNo = provinceComboBox.getSelectedIndex() + 1;
    String username = usernameField.getText();
    String password = PasswordField.getText();
    int jobroleNo = jobroleComboBox.getSelectedIndex() + 1;

    logger.debug("Check validations");
}

```

```

if (name.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (username.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Username !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    usernameField.requestFocus(true);

} else if (password.equals("")) {
    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Password !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    usernameField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.\\[a-zA-Z]{0,5}$";
    Pattern p = Pattern.compile(PATTERN);
    Matcher m = p.matcher(email);

    if (!m.matches()) {
        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);

        emailField.requestFocus(true);

    } else {

        String result = userModel.registerUser(name, nic, email, contactNo, address, city, distictNo, provinceNo,
username, password, jobroleNo);

        if (result.equals("Success")) {

            JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

            userModel.loadUserTable("", userTableModel);
        }
    }
}

```

```

        userModel.loadUserTableFull("", userTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

} else {

    String result = userModel.registerUser(name, nic, email, contactNo, address, city, distictNo, provinceNo,
    username, password, jobroleNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        userModel.loadUserTable("", userTableModel);

        userModel.loadUserTableFull("", userTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

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

    // TODO add your handling code here:

    String userNo = userNoField.getText();

    String name = nameField.getText();

    String nic = nicField.getText();

    String email = emailField.getText();

    String contactNo = contactNoField.getText();

    String address = addressField.getText();

    String city = cityField.getText();
}

```

```
int distictNo = districtComboBox.getSelectedIndex() + 1;

int provinceNo = provinceComboBox.getSelectedIndex() + 1;

String username = usernameField.getText();

String password = PasswordField.getText();

int jobroleNo = jobroleComboBox.getSelectedIndex() + 1;

logger.debug("Check validations");

if (name.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Name !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    nameField.requestFocus(true);

} else if (contactNo.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Contact No !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    contactNoField.requestFocus(true);

} else if (city.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The City !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    cityField.requestFocus(true);

} else if (username.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Username !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    usernameField.requestFocus(true);

} else if (password.equals("")) {

    JOptionPane.showConfirmDialog(rootPane, "Please Enter The Password !", "Error", -1,
JOptionPane.ERROR_MESSAGE);

    passwordField.requestFocus(true);

} else if (email.length() > 0) {

    String PATTERN = "^[a-zA-Z0-9./]{0,50}@[a-zA-Z]{0,10}\\.?[a-zA-Z]{0,5}$";

    Pattern p = Pattern.compile(PATTERN);

    Matcher m = p.matcher(email);

    if (!m.matches()) {

        JOptionPane.showConfirmDialog(rootPane, "Email is Wrong !", "Error", -1, JOptionPane.ERROR_MESSAGE);
    }
}
```

```

emailField.requestFocus(true);

} else {

    String result = userModel.updateUser(userNo, name, nic, email, contactNo, address, city, distictNo,
provinceNo, username, password, jobroleNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        userModel.loadUserTable("", userTableModel);

        userModel.loadUserTableFull("", userTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

} else {

    String result = userModel.updateUser(userNo, name, nic, email, contactNo, address, city, distictNo,
provinceNo, username, password, jobroleNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        userModel.loadUserTable("", userTableModel);

        userModel.loadUserTableFull("", userTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

}

}

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

// TODO add your handling code here:

String userNo = userNoField.getText();

```

```
int response = JOptionPane.showConfirmDialog(this, "Do You Want To Remove ?", "Confirm",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);

if (response == JOptionPane.YES_OPTION) {

    String result = userModel.removeUser(userNo);

    if (result.equals("Success")) {

        JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

        userModel.loadUserTable("", userTableModel);

        userModel.loadUserTableFull("", userTableModelFull);

        logger.debug("Message: " + result);

    } else {

        JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

        logger.error("Error: " + result);

    }

    clearFields();

}

}

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

    // TODO add your handling code here:

    clearFields();

}

private void userTableMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

    clearFields();

    int row = userTable.getSelectedRow();

    int userNo = (int) userTableModel.getValueAt(row, 0);
```

```
Object[] userData = userModel.getSelectedUserData(userNo);

userNoField.setText(new String(userData[0] + ""));
nameField.setText((String) userData[1]);
nicField.setText((String) userData[2]);
emailField.setText((String) userData[3]);
contactNoField.setText((String) userData[4]);
addressField.setText((String) userData[5]);
cityField.setText((String) userData[6]);
districtComboBox.setSelectedIndex((int) userData[7] - 1);
provinceComboBox.setSelectedIndex((int) userData[8] - 1);
usernameField.setText((String) userData[9]);
PasswordField.setText((String) userData[10]);
jobroleComboBox.setSelectedIndex((int) userData[11] - 1);

if (userNoField.getText().length() > 0) {
    registerButton.setEnabled(false);
    updateButton.setEnabled(true);
    removeButton.setEnabled(true);
} else {
    //    registerButton.setEnabled(true);
    //    updateButton.setEnabled(false);
    //    removeButton.setEnabled(false);
}
}

private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordField.getText().equals("Search")) {
        keywordField.setText("");
        keywordField.setForeground(new Color(0, 0, 0));
    }
}
```

```
    }

}

private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

    if (keywordField.getText().equals("")) {

        keywordField.setText("Search");

        keywordField.setForeground(new Color(153, 153, 153));

    }

}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    String keyword = keywordField.getText();

    userModel.loadUserTable(keyword, userTableModel);

}
```

```
private void clearFields() {

    userNoField.setText("");

    nameField.setText("");

    nicField.setText("");

    emailField.setText("");

    contactNoField.setText("");

    addressField.setText("");

    cityField.setText("");

    districtComboBox.setSelectedIndex(0);

    provinceComboBox.setSelectedIndex(0);

    usernameField.setText("");

    PasswordField.setText("");

    jobroleComboBox.setSelectedIndex(0);

}
```

```
registerButton.setEnabled(true);

updateButton.setEnabled(false);

removeButton.setEnabled(false);

logger.debug("clearFields");

}

/***
 * @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 {

UIManager.setLookAndFeel(new FlatIntelliJLaf());

UIManager.put("Table.showHorizontalLines", true);

UIManager.put("Table.showVerticalLines", true);

UIManager.put("ScrollBar.showButtons", true);

UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {

System.err.println("Failed to initialize LaF");

}

// try {

//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//         if ("Windows".equals(info.getName())) {

//             javax.swing.UIManager.setLookAndFeel(info.getClassName());

//             break;

//         }

//     }

// }
```

```
//      }

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

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

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

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

//  }

//</editor-fold>

//</editor-fold>
```

```
/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new UserManager().setVisible(true);
    }
});
```

```
// Variables declaration - do not modify

private javax.swing.JPasswordField PasswordField;
private javax.swing.JTextField addressField;
private javax.swing.JTextField cityField;
private javax.swing.JTextField contactNoField;
private javax.swing.JComboBox<String> districtComboBox;
private javax.swing.JTextField emailField;
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
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.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JComboBox<String> jobroleComboBox;
private javax.swing.JTextField keywordField;
private javax.swing.JTextField keywordFieldFull;
private javax.swing.JTextField nameField;
private javax.swing.JButton newButton;
private javax.swing.JTextField nicField;
private javax.swing.JComboBox<String> provinceComboBox;
private javax.swing.JButton registerButton;
private javax.swing.JButton removeButton;
private javax.swing.JButton updateButton;
```

```
private javax.swing.JTextField userNoField;  
private javax.swing.JTable userTable;  
private javax.swing.JTable userTableFull;  
private javax.swing.JTextField usernameField;  
// End of variables declaration  
  
}
```

UserModel.java

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package model;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLIntegrityConstraintViolationException;  
import javax.swing.table.DefaultTableModel;  
import org.apache.log4j.Logger;  
  
/**  
 *  
 * @author EZIO AUDITORY  
 */  
  
public class UserModel {  
  
    private Connection dbConnection;  
    static Logger logger = Logger.getLogger(UserModel.class.getName());
```

```
public UserModel(){

    dbConnection = database.dbConnection.getConnection();

}

public String registerUser(String name, String nic, String email, String contactNo, String address, String city, int
districtNo, int provinceNo, String username, String password, int jobroleNo) {

    String query = "INSERT INTO user(user_name, user_nic, user_email, user_contact_no, user_address, user_city,
district_no, province_no, user_username, user_password, jobrole_no, user_status)"

    + "VALUE(?,?,?,?,?,?,?,?,?,?)";

    logger.debug("SQL query String: " + query);

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, name);

    psm.setString(2, nic);

    psm.setString(3, email);

    psm.setString(4, contactNo);

    psm.setString(5, address);

    psm.setString(6, city);

    psm.setInt(7, districtNo);

    psm.setInt(8, provinceNo);

    psm.setString(9, username);

    psm.setString(10, password);

    psm.setInt(11, jobroleNo);

    psm.setInt(12, 1);

    psm.execute();

    logger.debug("Query execution done");

    logger.debug("return result");

    return "Success";

}catch(SQLIntegrityConstraintViolationException e){

    return "Username Already Exist !";
}
```

```

}catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
    logger.error("return result");
    return "Error !";
}

}

public String updateUser(String userNo, String name, String nic, String email, String contactNo, String address,
String city, int distictNo, int provinceNo, String username, String password, int jobroleNo) {

    String query = "UPDATE user SET user_name = ?, user_nic = ?, user_email = ?, user_contact_no = ?,
user_address = ?, user_city = ?, "
        + "district_no = ?, province_no = ?,user_username = ?, user_password = ?, jobrole_no = ? WHERE user_no =
?";

    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setString(1, name);
        psm.setString(2, nic);
        psm.setString(3, email);
        psm.setString(4, contactNo);
        psm.setString(5, address);
        psm.setString(6, city);
        psm.setInt(7, distictNo);
        psm.setInt(8, provinceNo);
        psm.setString(9, username);
        psm.setString(10, password);
        psm.setInt(11, jobroleNo);
        psm.setString(12, userNo);
        psm.execute();
        logger.debug("Query execution done");
    }
}

```

```
logger.debug("return result");

return "Success";

}catch(SQLIntegrityConstraintViolationException e){

    return "Username Already Exist !;

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

    logger.error("return result");

    return "Error !;

}

}

public String removeUser(String userNo) {

    String query = "UPDATE user SET user_status = ? WHERE user_no = ?";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        psm.setInt(1, 0);

        psm.setString(2, userNo);

        psm.execute();

        logger.debug("Query execution done");

        logger.debug("return result");

        return "Success";

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

        logger.error("return result");

        return "Error !;

    }

}
```

```
public void loadUserTable(String keyword, DefaultTableModel userTableModel) {  
  
    String query = "SELECT * FROM user INNER JOIN district INNER JOIN province INNER JOIN jobrole "  
  
        + "ON user.district_no = district.district_no AND user.province_no = province.province_no AND  
    user.jobrole_no = jobrole.jobrole_no "  
  
        + "WHERE (user_no LIKE ? OR user_name LIKE ? OR user_contact_no LIKE ? OR user_city LIKE ? OR  
    jobrole_name LIKE ?) AND user_status = ? ORDER BY user_no";  
  
    logger.debug("SQL query String: " + query);  
  
    try {  
  
        PreparedStatement psm = dbConnection.prepareStatement(query);  
  
        psm.setString(1, "%" + keyword + "%");  
  
        psm.setString(2, "%" + keyword + "%");  
  
        psm.setString(3, "%" + keyword + "%");  
  
        psm.setString(4, "%" + keyword + "%");  
  
        psm.setString(5, "%" + keyword + "%");  
  
        psm.setInt(6, 1);  
  
        ResultSet rs = psm.executeQuery();  
  
        logger.debug("Query execution done");  
  
        Object[] rowData = null;  
  
        userTableModel.setRowCount(0);  
  
        while(rs.next()){  
  
            int userNo = rs.getInt("user_no");  
  
            String name = rs.getString("user_name");  
  
            String email = rs.getString("user_email");  
  
            String contactNo = rs.getString("user_contact_no");  
  
            String city = rs.getString("user_city");  
  
            String username = rs.getString("user_username");  
  
            String jobroleNo = rs.getString("jobrole_name");  
  
            rowData = new Object[]{userNo, name, email, contactNo, city, username, jobroleNo};  
  
            userTableModel.addRow(rowData);  
  
            logger.debug("loadUserTable [addRow]");  
    }  
}
```

```
    }

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

}

public Object[] getSelectedUserData(int userNo) {
    String query = "SELECT * FROM user WHERE user_no = ?";
    logger.debug("SQL query String: " + query);

    try {
        PreparedStatement psm = dbConnection.prepareStatement(query);
        psm.setInt(1, userNo);
        ResultSet rs = psm.executeQuery();
        logger.debug("Query execution done");
        Object[] rowData = null;
        while(rs.next()){
            int userNoo = rs.getInt("user_no");
            String name = rs.getString("user_name");
            String nic = rs.getString("user_nic");
            String email = rs.getString("user_email");
            String contactNo = rs.getString("user_contact_no");
            String address = rs.getString("user_address");
            String city = rs.getString("user_city");
            int districtNo = rs.getInt("district_no");
            int provinceNo = rs.getInt("province_no");
            String username = rs.getString("user_username");
            String password = rs.getString("user_password");
            int jobroleNo = rs.getInt("jobrole_no");
            rowData = new
        }
    }
}
```

```

Object[] {userNoo, name, nic, email, contactNo, address, city, districtNo, provinceNo, username, password, jobroleNo};

        logger.debug("return rowData");

        return rowData;

    }

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}

```

```

public void loadUserTableFull(String keywordFull, DefaultTableModel userTableModelFull) {

    String query = "SELECT * FROM user INNER JOIN district INNER JOIN province INNER JOIN jobrole "

        + "ON user.district_no = district.district_no AND user.province_no = province.province_no AND
user.jobrole_no = jobrole.jobrole_no "

        + "WHERE (user_no LIKE ? OR user_name LIKE ? OR user_contact_no LIKE ? OR user_city LIKE ? OR
jobrole_name LIKE ?) AND user_status = ? ORDER BY user_no";

    logger.debug("SQL query String: " + query);

```

```

try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    psm.setString(1, "%" + keywordFull + "%");

    psm.setString(2, "%" + keywordFull + "%");

    psm.setString(3, "%" + keywordFull + "%");

    psm.setString(4, "%" + keywordFull + "%");

    psm.setString(5, "%" + keywordFull + "%");

    psm.setInt(6, 1);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Object[] rowData = null;

    userTableModelFull.setRowCount(0);

```

```
while(rs.next()){

    int userNo = rs.getInt("user_no");

    String name = rs.getString("user_name");

    String nic = rs.getString("user_nic");

    String email = rs.getString("user_email");

    String contactNo = rs.getString("user_contact_no");

    String address = rs.getString("user_address");

    String city = rs.getString("user_city");

    String district = rs.getString("district_name");

    String province = rs.getString("province_name");

    String username = rs.getString("user_username");

    String jobroleNo = rs.getString("jobrole_name");

    rowData = new
Object[]{userNo,name,nic,email,contactNo,address,city,district,province,username,jobroleNo};

    userTableModelFull.addRow(rowData);

    logger.debug("loadUserTableFull [addRow]");

}

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

}

}
```

Batch Name : Griffins

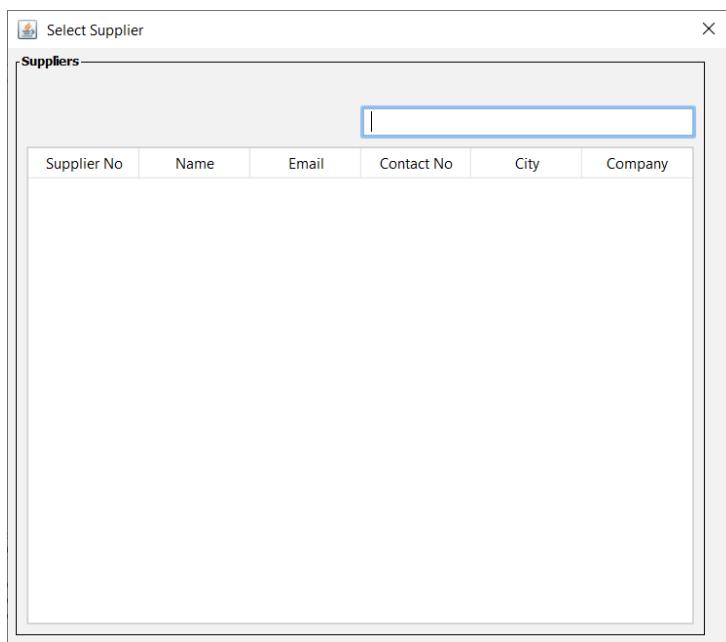
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Load Supplier

GUI Number : 33

Use case Name: Supplier Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search for suppliers registered in the system and to pass the details of a selected supplier into the GRN Manager interface.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|--|
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display suppliers | During a search operation, the user can find nearly similar supplier lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

LoadSupplierDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package gui;

import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.SupplierModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class LoadSupplierDialog extends javax.swing.JDialog {
```

```
/**  
  
 * Creates new form LoadSupplierDialog  
 */  
  
private SupplierModel supplierModel;  
  
static Logger logger = Logger.getLogger(LoadSupplierDialog.class.getName());  
  
DefaultTableModel supplierTableModel;  
  
  
int supplierNoo;  
  
String supplierName;  
  
  
public LoadSupplierDialog(java.awt.Frame parent, boolean modal) {  
  
    super(parent, modal);  
  
    initComponents();  
  
    logger.debug("Access to LoadSupplierDialog.java");  
  
  
    supplierTableModel = (DefaultTableModel) supplierTable.getModel();  
  
    supplierModel = new SupplierModel();  
  
  
    supplierModel.loadSupplierTable("", supplierTableModel);  
  
}  
  
  
/**  
  
 * 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() {  
  
    jPanel4 = new javax.swing.JPanel();
```

```

jScrollPane1 = new javax.swing.JScrollPane();
supplierTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Select Supplier");
setResizable(false);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Suppliers", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

supplierTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
    "Supplier No", "Name", "Email", "Contact No", "City", "Company"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

supplierTable.setRowHeight(25);
supplierTable.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {

```

```
        supplierTableMouseClicked(evt);

    }

});  
jScrollPane1.setViewportView(supplierTable);  
  
keywordField.setForeground(new java.awt.Color(153, 153, 153));  
keywordField.setText("Search");  
keywordField.setToolTipText("Search");  
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusGained(evt);  
    }  
    public void focusLost(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusLost(evt);  
    }  
});  
keywordField.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        keywordFieldKeyReleased(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,  
                javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(jPanel4Layout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)
    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,
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)
    .addGap(0, 748, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 611, Short.MAX_VALUE)
);
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap())))
);

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

private void supplierTableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int row = supplierTable.getSelectedRow();
    int supplierNo = (int) supplierTableModel.getValueAt(row, 0);
    Object[] supplierData = supplierModel.getSelectedSupplierData(supplierNo);

    supplierNoo = (int) supplierData[0];
    supplierName = (String) supplierData[1];

    this.dispose();
    logger.debug("LoadSupplierDialog.java Dispose");
}

private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordField.getText().equals("Search")) {
        keywordField.setText("");
        keywordField.setForeground(new Color(0, 0, 0));
    }
}
```

```
}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    if (keywordField.getText().equals("")) {
```

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

```
        keywordField.setForeground(new Color(153, 153, 153));
```

```
}
```

```
}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    String keyword = keywordField.getText();
```

```
    supplierModel.loadSupplierTable(keyword, supplierTableModel);
```

```
}
```

```
Object[] getSupplierData() {
```

```
    Object[] rowdata = new Object[]{supplierNoo,supplierName};
```

```
    logger.debug("return rowData");
```

```
    return rowdata;
```

```
}
```

```
/**
```

```
 * @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 {
```

```
UIManager.setLookAndFeel(new FlatIntelliJLaf());
UIManager.put("Table.showHorizontalLines", true);
UIManager.put("Table.showVerticalLines", true);
UIManager.put("ScrollBar.showButtons", true);
UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(LoadSupplierDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(LoadSupplierDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(LoadSupplierDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(LoadSupplierDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// }

//</editor-fold>
//</editor-fold>

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
public void run() {  
    LoadSupplierDialog dialog = new LoadSupplierDialog(new javax.swing.JFrame(), true);  
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {  
        @Override  
        public void windowClosing(java.awt.event.WindowEvent e) {  
            System.exit(0);  
        }  
    });  
    dialog.setVisible(true);  
}  
  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel jPanel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTextField keywordField;  
private javax.swing.JTable supplierTable;  
// End of variables declaration  
  
}
```

Batch Name : Griffins

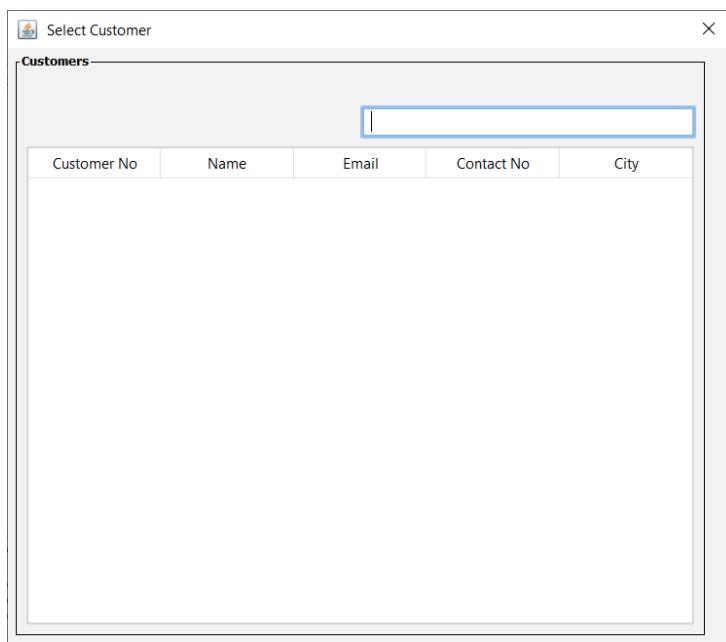
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Load Customer

GUI Number : 34

Use case Name: Customer Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search for customers registered in the system and to pass the details of a selected customer into the Sales Manager interface.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|--|
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display customers | During a search operation, the user can find nearly similar customer lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



LoadCustomerDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package gui;

import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.CustomerModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class LoadCustomerDialog extends javax.swing.JDialog {
```

```
/**  
  
 * Creates new form LoadCustomerDialog  
  
 */  
  
private CustomerModel customerModel;  
  
static Logger logger = Logger.getLogger(LoadCustomerDialog.class.getName());  
  
DefaultTableModel customerTableModel;  
  
  
int customerNoo;  
  
String customerName;  
  
  
public LoadCustomerDialog(java.awt.Frame parent, boolean modal) {  
  
    super(parent, modal);  
  
    initComponents();  
  
    logger.debug("Access to LoadCustomerDialog.java");  
  
  
    customerTableModel = (DefaultTableModel) customerTable.getModel();  
  
    customerModel = new CustomerModel();  
  
  
    customerModel.loadCustomerTable("", customerTableModel);  
  
}  
  
  
/**  
  
 * 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() {  
  
    jPanel4 = new javax.swing.JPanel();
```

```
jScrollPane1 = new javax.swing.JScrollPane();
customerTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Select Customer");
setResizable(false);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Customers", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

customerTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Customer No", "Name", "Email", "Contact No", "City"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}

customerTable.setRowHeight(25);
customerTable.addMouseListener(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        customerTableMouseClicked(evt);

    }

});  
jScrollPane1.setViewportView(customerTable);  
  
keywordField.setForeground(new java.awt.Color(153, 153, 153));  
keywordField.setText("Search");  
keywordField.setToolTipText("Search");  
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusGained(evt);  
    }  
    public void focusLost(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusLost(evt);  
    }  
});  
keywordField.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        keywordFieldKeyReleased(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,  
                javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(jPanel4Layout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)
    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,
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)
    .addGap(0, 748, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 611, Short.MAX_VALUE)
);
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap())))
);
```

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

```
private void customerTableMouseClicked(java.awt.event.MouseEvent evt) {
```

```
// TODO add your handling code here:
```

```
int row = customerTable.getSelectedRow();
int customerNo = (int) customerTableModel.getValueAt(row, 0);
Object[] customerData = customerModel.getSelectedCustomerData(customerNo);
```

```
customerNoo = (int) customerData[0];
```

```
customerName = (String) customerData[1];
```

```
this.dispose();
```

```
logger.debug("LoadCustomerDialog.java Dispose");
```

```
}
```

```
private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {
```

```
// TODO add your handling code here:
```

```
if (keywordField.getText().equals("Search")) {
```

```
    keywordField.setText("");
    keywordField.setForeground(new Color(0, 0, 0));
```

```
    }

}

private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {

    // TODO add your handling code here:

    if (keywordField.getText().equals("")) {

        keywordField.setText("Search");

        keywordField.setForeground(new Color(153, 153, 153));

    }

}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    String keyword = keywordField.getText();

    customerModel.loadCustomerTable(keyword, customerTableModel);

}
```

```
Object[] getCustomerData() {

    Object[] rowdata = new Object[]{customerNoo, customerName};

    logger.debug("return rowData");

    return rowdata;

}
```

```
/**

 * @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 {  
  
    UIManager.setLookAndFeel(new FlatIntelliJLaf());  
  
    UIManager.put("Table.showHorizontalLines", true);  
  
    UIManager.put("Table.showVerticalLines", true);  
  
    UIManager.put("ScrollBar.showButtons", true);  
  
    UIManager.put("ScrollBar.width", 16);  
  
} catch (Exception ex) {  
  
    System.err.println("Failed to initialize LaF");  
  
}  
  
// try {  
  
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  
  
//         if ("Windows".equals(info.getName())) {  
  
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());  
  
//             break;  
  
//         }  
  
//     }  
  
// } catch (ClassNotFoundException ex) {  
  
//     java.util.logging.Logger.getLogger(LoadCustomerDialog.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (InstantiationException ex) {  
  
//     java.util.logging.Logger.getLogger(LoadCustomerDialog.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (IllegalAccessException ex) {  
  
//     java.util.logging.Logger.getLogger(LoadCustomerDialog.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
  
//     java.util.logging.Logger.getLogger(LoadCustomerDialog.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// }  
  
//</editor-fold>  
  
//</editor-fold>
```

```
/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        LoadCustomerDialog dialog = new LoadCustomerDialog(new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});

// Variables declaration - do not modify
private javax.swing.JTable customerTable;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField keywordField;
// End of variables declaration

}
```

Batch Name : Griffins

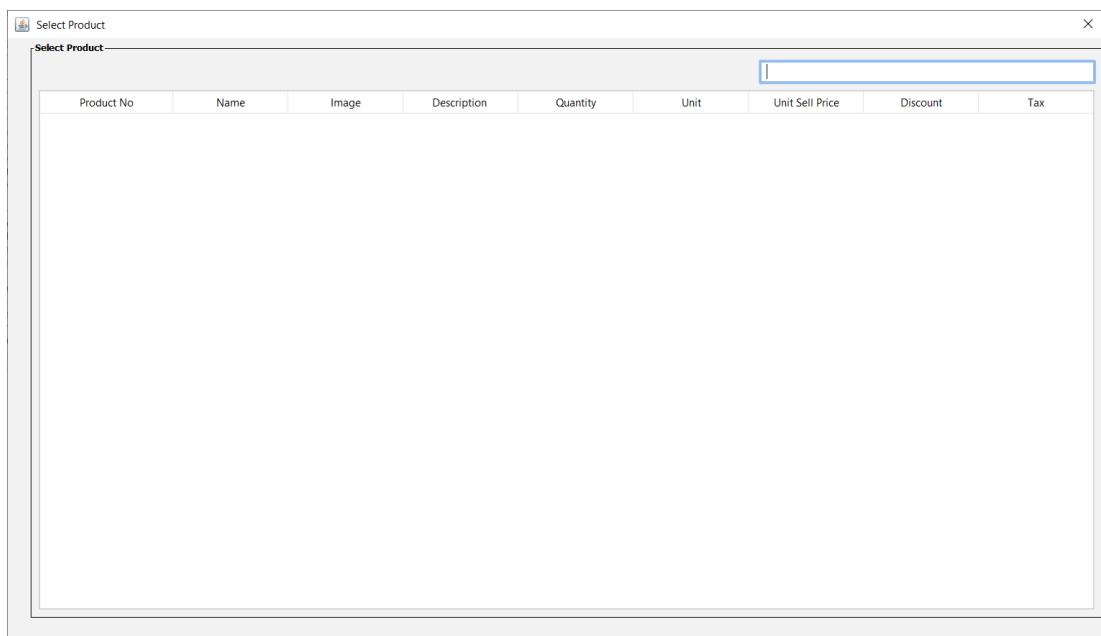
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Load Product

GUI Number : 35

Use case Name: Product Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search for products registered in the system and to pass the details of a selected product into the GRN Manager interface or the Sales Manager interface.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|--|
| JPanel | To adjust alignments | <i>Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel.</i> |
| JTable | Display products | <i>During a search operation, the user can find nearly similar product lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data.</i> |
| JTextField | Insert keyword | <i>You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword.</i> |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



LoadProductDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import java.awt.Component;
import javax.swing.JTable;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableCellRenderer;
import model.ProductModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/  
  
public class LoadProductDialog extends javax.swing.JDialog {  
  
    /**  
     * Creates new form LoadProductDialog  
     */  
  
    private ProductModel productModel;  
  
    static Logger logger = Logger.getLogger(LoadProductDialog.class.getName());  
  
    DefaultTableModel productTableModelFull;  
  
  
    int productNoo;  
  
    String productName;  
  
    double productQty;  
  
    int productUnitNo;  
  
    double productUnitSellPrice;  
  
    double productDiscount;  
  
    double productTax;  
  
    String productUnitName = "Number";  
  
  
    public LoadProductDialog(java.awt.Frame parent, boolean modal) {  
        super(parent, modal);  
        initComponents();  
        logger.debug("Access to LoadProductDialog.java");  
  
  
        productTableFullSettings(productTableFull);  
  
  
        productTableModelFull = (DefaultTableModel) productTableFull.getModel();  
        productModel = new ProductModel();  
  
  
        productModel.loadProductTableFull("", productTableModelFull);  
    }  
}
```

```
/**  
  
 * 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() {  
  
    jPanel3 = new javax.swing.JPanel();  
  
    keywordFieldFull = new javax.swing.JTextField();  
  
    jScrollPane3 = new javax.swing.JScrollPane();  
  
    productTableFull = new javax.swing.JTable();  
  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);  
  
    setTitle("Select Product");  
  
    setResizable(false);  
  
  
    jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new  
        java.awt.Color(0, 0, 0)), "Select Product", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,  
        javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N  
  
  
    keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));  
  
    keywordFieldFull.setText("Search");  
  
    keywordFieldFull.setToolTipText("Search");  
  
    keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {  
  
        public void focusGained(java.awt.event.FocusEvent evt) {  
  
            keywordFieldFullFocusGained(evt);  
  
        }  
  
        public void focusLost(java.awt.event.FocusEvent evt) {  
  
    
```

```
keywordFieldFullFocusLost(evt);

}

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyReleased(java.awt.event.KeyEvent evt) {

keywordFieldFullKeyReleased(evt);

}

});

productTableFull.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

productTableFull.setModel(new javax.swing.table.DefaultTableModel(


new Object [][] {

},


new String [] {

"Product No", "Name", "Image", "Description ", "Quantity", "Unit", "Unit Sell Price", "Discount", "Tax"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

productTableFull.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

productTableFullMouseClicked(evt);

}

});

});
```

```
jScrollPane3.setViewportView(productTableFull);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
        .addGap()
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane3)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(0, 1003, Short.MAX_VALUE)
                .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469,
                    javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap()
    )
);

jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addGap()
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 726, Short.MAX_VALUE)
        .addGap()
    )
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 1530, Short.MAX_VALUE)
);
```

```

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

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

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap()))
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 842, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel3, 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();
setLocationRelativeTo(null);
}// </editor-fold>

private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {
    if (keywordFieldFull.getText().equals("Search")) {
        keywordFieldFull.setText("");
        keywordFieldFull.setForeground(new Color(0, 0, 0));
    }
}

private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordFieldFull.getText().equals("")) {

```

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

keywordFieldFull.setForeground(new Color(153, 153, 153));

}

}

private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

String keywordFull = keywordFieldFull.getText();

productModel.loadProductTableFull(keywordFull, productTableModelFull);

}

private void productTableFullMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = productTableFull.getSelectedRow();

int productNo = (int) productTableModelFull.getValueAt(row, 0);

Object[] productData = productModel.getSelectedProductData(productNo);

productNoo = (int) productData[0];

productName = (String) productData[1];

productQty = (double) productData[2];

productUnitNo = (int) productData[3];

productUnitSellPrice = (double) productData[4];

productDiscount = (double) productData[5];

productTax = (double) productData[6];

productUnitName = (String) productData[9];

this.dispose();

logger.debug("LoadProductDialog.java Dispose");

}

public void productTableFullSettings(JTable jTable) {
```

```

jtable.setRowHeight(100);

jtable.getColumn("Image").setCellRenderer(new ProductTableFullRenderer());

jtable.getColumnModel().getColumn(0).setPreferredWidth(100);

}

Object[] getProductData() {

    Object[] rowdata = new Object[]{productNoo, productName, productQty, productUnitNo, productUnitName,
productUnitSellPrice, productDiscount,productTax};

    logger.debug("return rowData");

    return rowdata;

}

/***
 * @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 {

        UIManager.setLookAndFeel(new FlatIntelliJLaf());

        UIManager.put("Table.showHorizontalLines", true);

        UIManager.put("Table.showVerticalLines", true);

        UIManager.put("ScrollBar.showButtons", true);

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {

        System.err.println("Failed to initialize LaF");

    }

//    try {

```

```
//      for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//
//          if ("Windows".equals(info.getName())) {
//
//              javax.swing.UIManager.setLookAndFeel(info.getClassName());
//
//              break;
//
//          }
//
//      }
//
//  } catch (ClassNotFoundException ex) {
//
//      java.util.logging.Logger.getLogger(LoadProductDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
//  } catch (InstantiationException ex) {
//
//      java.util.logging.Logger.getLogger(LoadProductDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
//  } catch (IllegalAccessException ex) {
//
//      java.util.logging.Logger.getLogger(LoadProductDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
//  } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//
//      java.util.logging.Logger.getLogger(LoadProductDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
//
//  }
//
//  }
//
//  //</editor-fold>
//
//  //</editor-fold>
```

```
/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        LoadProductDialog dialog = new LoadProductDialog(new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});
```

```
    }

});  
}  
  
// Variables declaration - do not modify  
  
private javax.swing.JPanel jPanel3;  
  
private javax.swing.JScrollPane jScrollPane3;  
  
private javax.swing.JTextField keywordFieldFull;  
  
private javax.swing.JTable productTableFull;  
  
// End of variables declaration  
  
}
```

Batch Name : Griffins

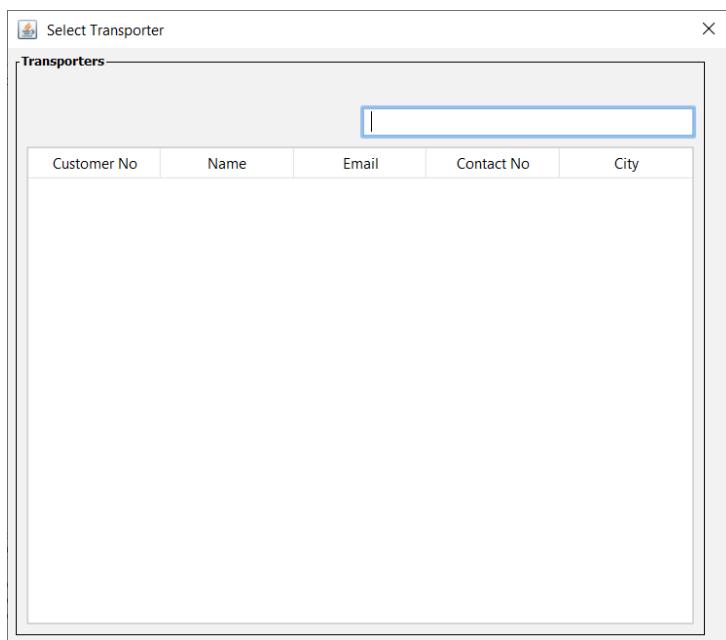
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Load Transporter

GUI Number : 36

Use case Name: Transporter Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search for transporters registered in the system and to pass the details of a selected transporter into the Delivery Manager interface.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|--|
| JPanel | To adjust alignments | <i>Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels.</i> |
| JTable | Display transporters | <i>During a search operation, the user can find nearly similar transporter lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data.</i> |
| JTextField | Insert keyword | <i>You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword.</i> |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

LoadTransporterDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package gui;

import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.TransporterModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class LoadTransporterDialog extends javax.swing.JDialog {
```

```
/**  
  
 * Creates new form LoadTransporterDialog  
  
 */  
  
private TransporterModel transporterModel;  
  
static Logger logger = Logger.getLogger(LoadTransporterDialog.class.getName());  
  
DefaultTableModel transporterTableModel;  
  
  
int transporterNoo;  
  
String transpoterName;  
  
  
public LoadTransporterDialog(java.awt.Frame parent, boolean modal) {  
  
    super(parent, modal);  
  
    initComponents();  
  
    logger.debug("Access to LoadTransporterDialog.java");  
  
  
    transporterTableModel = (DefaultTableModel) transporterTable.getModel();  
  
    transporterModel = new TransporterModel();  
  
  
    transporterModel.loadTransporterTable("", transporterTableModel);  
  
}  
  
  
/**  
  
 * 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() {  
  
  
    jPanel4 = new javax.swing.JPanel();
```

```
jScrollPane1 = new javax.swing.JScrollPane();
transporterTable = new javax.swing.JTable();
keywordField = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Select Transporter");
setResizable(false);

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), "Transporters", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13)))); // NOI18N

transporterTable.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
"Customer No", "Name", "Email", "Contact No", "City"
}
) {
boolean[] canEdit = new boolean [] {
false, false, false, false, false
};

public boolean isCellEditable(int rowIndex, int columnIndex) {
return canEdit [columnIndex];
}

transporterTable.setRowHeight(25);
transporterTable.addMouseListener(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
transporterTableMouseClicked(evt);  
}  
});  
jScrollPane1.setViewportView(transporterTable);  
  
keywordField.setForeground(new java.awt.Color(153, 153, 153));  
keywordField.setText("Search");  
keywordField.setToolTipText("Search");  
keywordField.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusGained(evt);  
    }  
    public void focusLost(java.awt.event.FocusEvent evt) {  
        keywordFieldFocusLost(evt);  
    }  
});  
keywordField.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        keywordFieldKeyReleased(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  
jPanel4.setLayout(jPanel4Layout);  
jPanel4Layout.setHorizontalGroup(  
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel4Layout.createSequentialGroup()  
        .addContainerGap()  
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 690,  
                javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(jPanel4Layout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)
    .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())
);

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(keywordField, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 495,
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)
    .addGap(0, 748, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 611, Short.MAX_VALUE)
);
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap())))
);

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

private void transporterTableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int row = transporterTable.getSelectedRow();
    int transporterNo = (int) transporterTableModel.getValueAt(row, 0);
    Object[] transporterData = transporterModel.getSelectedTransporterData(transporterNo);

    transporterNoo = (int) transporterData[0];
    transpoterName = (String) transporterData[1];

    this.dispose();
    logger.debug("LoadTransporterDialog.java Dispose");
}

private void keywordFieldFocusGained(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (keywordField.getText().equals("Search")) {
        keywordField.setText("");
        keywordField.setForeground(new Color(0, 0, 0));
    }
}
```

```
}
```

```
private void keywordFieldFocusLost(java.awt.event.FocusEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    if (keywordField.getText().equals("")) {
```

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

```
        keywordField.setForeground(new Color(153, 153, 153));
```

```
}
```

```
}
```

```
private void keywordFieldKeyReleased(java.awt.event.KeyEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    String keyword = keywordField.getText();
```

```
    transporterModel.loadTransporterTable(keyword, transporterTableModel);
```

```
}
```

```
/**
```

```
* @param args the command line arguments
```

```
*/
```

```
Object[] getTransporterData() {
```

```
    Object[] rowdata = new Object[]{transporterNoo,transpoterName};
```

```
    logger.debug("return rowData");
```

```
    return rowdata;
```

```
}
```

```
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 {
```

```
UIManager.setLookAndFeel(new FlatIntelliJLaf());
UIManager.put("Table.showHorizontalLines", true);
UIManager.put("Table.showVerticalLines", true);
UIManager.put("ScrollBar.showButtons", true);
UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(LoadTransporterDialog.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(LoadTransporterDialog.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(LoadTransporterDialog.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(LoadTransporterDialog.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
// }

//</editor-fold>
//</editor-fold>
```

```
/* Create and display the dialog */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        LoadTransporterDialog dialog = new LoadTransporterDialog(new javax.swing.JFrame(), true);

        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override

            public void windowClosing(java.awt.event.WindowEvent e) {

                System.exit(0);

            }

        });

        dialog.setVisible(true);

    }

});

// Variables declaration - do not modify

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField keywordField;

private javax.swing.JTable transporterTable;

// End of variables declaration

}

}
```

Batch Name : Griffins

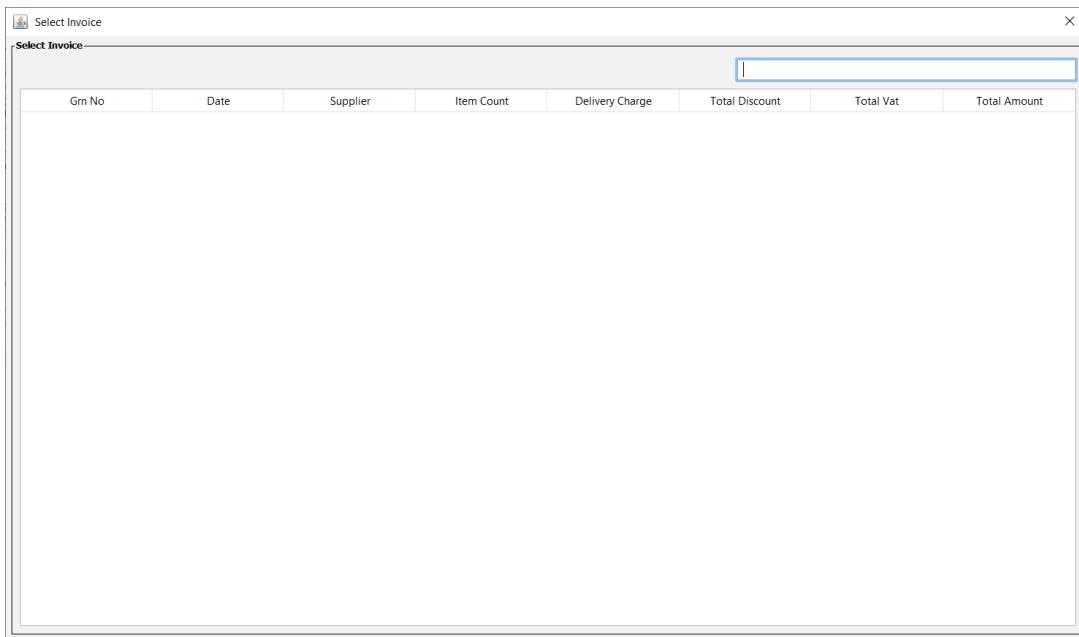
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Load Invoice

GUI Number : 37

Use case Name: Sales Management



01. Brief Description of the GUI:

The main objective of this interface is to display and search for invoices saved in the system and to pass the details of a selected invoice into the Delivery Manager interface.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|---|
| JPanel | To adjust alignments | Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel. |
| JTable | Display invoices | During a search operation, the user can find nearly similar invoice lists according to the searching key. If the database was consists of nearly similar data User can select the most appropriate or the exact data from the data. |
| JTextField | Insert keyword | You need a keyword to search for what you want from the data in the table. This JTextField is used to enter that keyword. |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|---------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

LoadInvoiceDialog.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package gui;

import com.formdev.flatlaf.FlatIntelliJLaf;
import java.awt.Color;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import model.SalesModel;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class LoadInvoiceDialog extends javax.swing.JDialog {
```

```
/**  
  
 * Creates new form LoadInvoiceDialog  
  
 */  
  
private SalesModel salesModel;  
  
static Logger logger = Logger.getLogger(LoadInvoiceDialog.class.getName());  
  
DefaultTableModel invoiceTableModelFull;  
  
  
int invoiceNoo;  
  
double deliveryCharge;  
  
  
public LoadInvoiceDialog(java.awt.Frame parent, boolean modal) {  
  
    super(parent, modal);  
  
    initComponents();  
  
    logger.debug("Access to LoadCustomerDialog.java");  
  
  
    invoiceTableModelFull = (DefaultTableModel) invoiceTableFull.getModel();  
  
    salesModel = new SalesModel();  
  
  
    salesModel.loadInvoiceTableFullForLoadInvoiceDialog("", invoiceTableModelFull);  
  
}  
  
  
/**  
  
 * 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();
```

```
keywordFieldFull = new javax.swing.JTextField();

jScrollPane3 = new javax.swing.JScrollPane();

invoiceTableFull = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Select Invoice");

setResizable(false);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Select Invoice", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

keywordFieldFull.setForeground(new java.awt.Color(153, 153, 153));

keywordFieldFull.setText("Search");

keywordFieldFull.setToolTipText("Search");

keywordFieldFull.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusGained(evt);

    }

    public void focusLost(java.awt.event.FocusEvent evt) {

        keywordFieldFullFocusLost(evt);

    }

});

keywordFieldFull.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        keywordFieldFullKeyReleased(evt);

    }

});

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

```



```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 1438, Short.MAX_VALUE)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(0, 969, Short.MAX_VALUE)
        .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 469,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap())))
);

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 793, Short.MAX_VALUE)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(keywordFieldFull, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 726, Short.MAX_VALUE)
            .addContainerGap())))
    .addContainerGap()))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addContainerGap())
    .addContainerGap()))
);

layout.setVerticalGroup(

```

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

```
.addContainerGap()
```

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

```
.addContainerGap())
```

```
);
```

```
pack();
```

```
setLocationRelativeTo(null);
```

```
}// </editor-fold>
```

```
private void keywordFieldFullFocusGained(java.awt.event.FocusEvent evt) {
```

```
if (keywordFieldFull.getText().equals("Search")) {
```

```
keywordFieldFull.setText("");
```

```
keywordFieldFull.setForeground(new Color(0, 0, 0));
```

```
}
```

```
}
```

```
private void keywordFieldFullFocusLost(java.awt.event.FocusEvent evt) {
```

```
// TODO add your handling code here:
```

```
if (keywordFieldFull.getText().equals("")) {
```

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

```
keywordFieldFull.setForeground(new Color(153, 153, 153));
```

```
}
```

```
}
```

```
private void keywordFieldFullKeyReleased(java.awt.event.KeyEvent evt) {
```

```
// TODO add your handling code here:
```

```
String keywordFull = keywordFieldFull.getText();
```

```
salesModel.loadInvoiceTableFullForLoadInvoiceDialog(keywordFull, invoiceTableModelFull);
```

```
}
```

```
private void invoiceTableFullMouseClicked(java.awt.event.MouseEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    int row = invoiceTableFull.getSelectedRow();
```

```
    int invoiceNo = (int) invoiceTableModelFull.getValueAt(row, 0);
```

```
    Object[] invoiceData = salesModel.getSelectedInvoiceData(invoiceNo);
```

```
    invoiceNoo = (int) invoiceData[0];
```

```
    deliveryCharge = (double) invoiceData[1];
```

```
    this.dispose();
```

```
    logger.debug("LoadInvoiceDialog.java Dispose");
```

```
}
```

```
Object[] getInvoiceData() {
```

```
    Object[] rowdata = new Object[]{invoiceNoo, deliveryCharge};
```

```
    logger.debug("return rowData");
```

```
    return rowdata;
```

```
}
```

```
/**
```

```
* @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 {
```

```
UIManager.setLookAndFeel(new FlatIntelliJLaf());
UIManager.put("Table.showHorizontalLines", true);
UIManager.put("Table.showVerticalLines", true);
UIManager.put("ScrollBar.showButtons", true);
UIManager.put("ScrollBar.width", 16);

} catch (Exception ex) {
    System.err.println("Failed to initialize LaF");
}

}

// try {
//     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
//         if ("Windows".equals(info.getName())) {
//             javax.swing.UIManager.setLookAndFeel(info.getClassName());
//             break;
//         }
//     }
// }

// } catch (ClassNotFoundException ex) {
//     java.util.logging.Logger.getLogger(LoadInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (InstantiationException ex) {
//     java.util.logging.Logger.getLogger(LoadInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (IllegalAccessException ex) {
//     java.util.logging.Logger.getLogger(LoadInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {
//     java.util.logging.Logger.getLogger(LoadInvoiceDialog.class.getName()).log(java.util.logging.Level.SEVERE,
// null, ex);
// }

// }

//</editor-fold>
//</editor-fold>

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
public void run() {  
  
    LoadInvoiceDialog dialog = new LoadInvoiceDialog(new javax.swing.JFrame(), true);  
  
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {  
  
        @Override  
  
        public void windowClosing(java.awt.event.WindowEvent e) {  
  
            System.exit(0);  
  
        }  
  
    });  
  
    dialog.setVisible(true);  
  
}  
  
});  
  
}  
  
  
// Variables declaration - do not modify  
  
private javax.swing.JTable invoiceTableFull;  
  
private javax.swing.JPanel jPanel1;  
  
private javax.swing.JScrollPane jScrollPane3;  
  
private javax.swing.JTextField keywordFieldFull;  
  
// End of variables declaration  
  
}  
}
```

Batch Name : Griffins

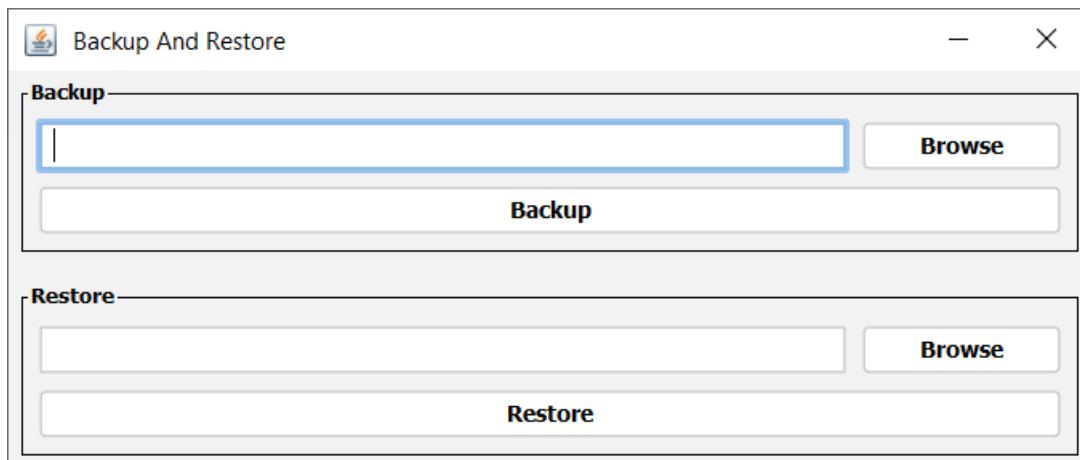
Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: Backup and Restore

GUI Number : 38

Use case Name:



01. Brief Description of the GUI:

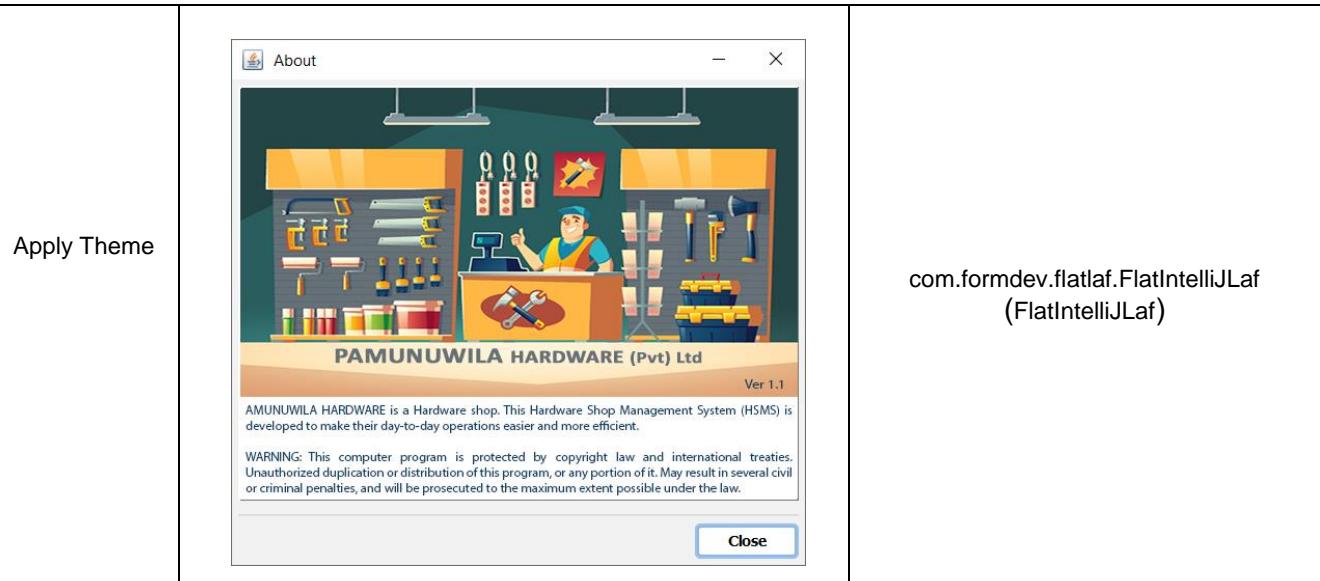
The main objective of this interface is to backup and restore the database of this system.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|--|--|
| JPanel | To adjust alignments | <i>Using JPanels makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanels.</i> |
| JTextField | Insert path | <i>To enter and display the path to the location where the database should be backed up and the path to the location where the restore file is located.</i> |
| JButtons | Browse paths, Backup database & Restore database | <i>Buttons are required to browse the path to the location where the database should be backed up and the path to the location where the file is restored, to back up the database and restore the database. These JButtons meet that requirement.</i> |
| JSeparator | Make a division | <i>To separate and display fields</i> |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |



BackupAndRestore.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import model.BackupAndRestoreModel;
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
```

```
*/\n\npublic class BackupAndRestore extends javax.swing.JFrame {\n\n    /**\n     * Creates new form BackupAndRestore\n     */\n\n    private BackupAndRestoreModel backupAndRestoreModel;\n\n    static Logger logger = Logger.getLogger(BackupAndRestore.class.getName());\n\n\n    public BackupAndRestore() {\n\n        initComponents();\n\n        logger.debug("Access to BackupAndRestore.java");\n\n        backupAndRestoreModel = new BackupAndRestoreModel();\n\n    }\n\n    /**\n     * This method is called from within the constructor to initialize the form.\n     *\n     * WARNING: Do NOT modify this code. The content of this method is always\n     *\n     * regenerated by the Form Editor.\n     */\n\n    @SuppressWarnings("unchecked")\n    // <editor-fold defaultstate="collapsed" desc="Generated Code">\n\n    private void initComponents() {\n\n        jPanel1 = new javax.swing.JPanel();\n\n        backupPathField = new javax.swing.JTextField();\n\n        backupPathBrowseButton = new javax.swing.JButton();\n\n        backupButton = new javax.swing.JButton();\n\n        jPanel2 = new javax.swing.JPanel();\n\n        restorePathField = new javax.swing.JTextField();\n\n        restorePathBrowseButton = new javax.swing.JButton();\n\n    }\n\n}\n\n
```

```
restoreButton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Backup And Restore ");
setAlwaysOnTop(true);
setResizable(false);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Backup", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

backupPathBrowseButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
backupPathBrowseButton.setText("Browse");
backupPathBrowseButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backupPathBrowseButtonActionPerformed(evt);
    }
});;

backupButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
backupButton.setText("Backup");
backupButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backupButtonActionPerformed(evt);
    }
});;

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

```

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

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

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addComponent(backupPathField, javax.swing.GroupLayout.PREFERRED_SIZE, 508,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

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

            .addComponent(backupButton, javax.swing.GroupLayout.DEFAULT_SIZE, 642, Short.MAX_VALUE))

        .addContainerGap()

    );

jPanel1Layout.setVerticalGroup

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

.addGroup(jPanel1Layout.createSequentialGroup()

    .addContainerGap()

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

        .addComponent(backupPathField)

        .addComponent(backupPathBrowseButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(backupButton, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

        .addContainerGap()

    );

);

```

```

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)), "Restore", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 13))); // NOI18N

```

```

restorePathBrowseButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
restorePathBrowseButton.setText("Browse");
restorePathBrowseButton.addActionListener(new java.awt.event.ActionListener() {

```

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    restorePathBrowseButtonActionPerformed(evt);  
}  
};  
  
restoreButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
restoreButton.setText("Restore");  
restoreButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        restoreButtonActionPerformed(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()  
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addGroup(jPanel2Layout.createSequentialGroup()  
                .addComponent	restorePathField, javax.swing.GroupLayout.PREFERRED_SIZE, 508,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
                .addComponent	restorePathBrowseButton, javax.swing.GroupLayout.DEFAULT_SIZE,  
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))  
                .addComponent	restoreButton, javax.swing.GroupLayout.DEFAULT_SIZE, 642, Short.MAX_VALUE))  
                .addContainerGap())  
);  
jPanel2Layout.setVerticalGroup(  
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addContainerGap()

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

.addComponent	restorePathField

.addComponent	restorePathBrowseButton, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addComponent	restoreButton, javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)

.addContainerGap()

);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

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

.addComponent(jPanel2, 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(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.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

```
};

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

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

// TODO add your handling code here:
JFileChooser backupPath = new JFileChooser();
int returnValue = backupPath.showOpenDialog(this);

String date = new SimpleDateFormat("YYYY-MM-dd_hh-mm-ss").format(new Date());

if (returnValue == JFileChooser.APPROVE_OPTION) {
    File backupFile = backupPath.getSelectedFile();
    String backupLocation = backupFile.getAbsolutePath();
    String backupFilepath = backupLocation + " " + date + ".sql";
    backupPathField.setText(backupFilepath);
    logger.debug("set backup file path: "+backupFilepath);
}

}

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

// TODO add your handling code here:
String backupPath = backupPathField.getText();

if (backupPath.equals("")) {
    backupPathField.requestFocus();
} else {
    String result = backupAndRestoreModel.backup(backupPath);
    if (result.equals("Backup Success")) {
```

```
JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

logger.debug("Message: "+result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: "+result);

}

}

}
```

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

    // TODO add your handling code here:

    JFileChooser restorePath = new JFileChooser();

    int returnValue = restorePath.showOpenDialog(this);

    if (returnValue == JFileChooser.APPROVE_OPTION) {

        File restoreFile = restorePath.getSelectedFile();

        String restoreFileLocation = restoreFile.getAbsolutePath();

        restorePathField.setText(restoreFileLocation);

        logger.debug("set restore file path: "+restoreFileLocation);

    }

}
```

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

    // TODO add your handling code here:

    String restorePath = restorePathField.getText();

    if (restorePath.equals("")) {

        restorePathField.requestFocus();

    } else {

        String result = backupAndRestoreModel.restore(restorePath);

        if (result.equals("Restore Success")) {
```

```
JOptionPane.showConfirmDialog(rootPane, result, "Message", -1, 1);

logger.debug("Message: "+result);

} else {

    JOptionPane.showConfirmDialog(rootPane, result, "Error", -1, JOptionPane.ERROR_MESSAGE);

    logger.error("Error: "+result);

}

}

}

/**

 * @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 {

        UIManager.setLookAndFeel(new FlatIntelliJLaf());

        UIManager.put("Table.showHorizontalLines", true);

        UIManager.put("Table.showVerticalLines", true);

        UIManager.put("ScrollBar.showButtons", true);

        UIManager.put("ScrollBar.width", 16);

    } catch (Exception ex) {

        System.err.println("Failed to initialize LaF");

    }

//    try {

//        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

//            if ("Windows".equals(info.getName())) {

//                javax.swing.UIManager.setLookAndFeel(info.getClassName());


```

```
//         break;  
//     }  
// }  
  
// } catch (ClassNotFoundException ex) {  
//     java.util.logging.Logger.getLogger(BackupAndRestore.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (InstantiationException ex) {  
//     java.util.logging.Logger.getLogger(BackupAndRestore.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (IllegalAccessException ex) {  
//     java.util.logging.Logger.getLogger(BackupAndRestore.class.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
  
// } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
//     java.util.logging.Logger.getLogger(BackupAndRestore.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 BackupAndRestore().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JButton backupButton;  
  
private javax.swing.JButton backupPathBrowseButton;  
  
private javax.swing.JTextField backupPathField;  
  
private javax.swing.JPanel jPanel1;  
  
private javax.swing.JPanel jPanel2;  
  
private javax.swing.JButton restoreButton;
```

```
private javax.swing.JButton restorePathBrowseButton;
private javax.swing.JTextField restorePathField;
// End of variables declaration
}
```

BackupAndRestoreModel.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package model;
```

```
import org.apache.log4j.Logger;
```

```
/**
 *
 * @author EZIO AUDITORY
 */
```

```
public class BackupAndRestoreModel {
```

```
    static Logger logger = Logger.getLogger(BackupAndRestoreModel.class.getName());
```

```
    public String backup(String backupPath) {
```

```
        String msg = null;
```

```
        Process p = null;
```

```
        try {
```

```
            Runtime runtime = Runtime.getRuntime();
```

```
            p = runtime.exec("C:\\Program Files\\MySQL\\MySQL Server 8.0\\bin\\mysqldump.exe --host=localhost --port=3307 -uroot -pass34 --add-drop-database -B pamunuwila_hw
```

```
_hardware_db -r" + backupPath);
```

```
        logger.debug(p);
```

```
int processComplete = p.waitFor();

if (processComplete == 0) {

    msg = "Backup Success";

} else {

    msg = "Backup Error !";

}

logger.debug("Set msg");

} catch (Exception e) {

    e.printStackTrace();

    logger.debug("Exception: " + e.getMessage());

}

logger.debug("return msg");

return msg;

}

public String restore(String restorePath) {

String msg = null;

String dbUserName = "root";

String dbPassword = "ass34";



String[] restoreCmd = new String[]{"C:\\Program Files\\MySQL\\MySQL Server 8.0\\bin\\mysql.exe", "--host=localhost", "--port=3307", "--user=" + dbUserName, "--password=" + dbPassword, "-e", "source " + restorePath};

logger.debug(restoreCmd);

Process runtimeProcess;

try {

    runtimeProcess = Runtime.getRuntime().exec(restoreCmd);

    int processComplete = runtimeProcess.waitFor();

    System.out.println(processComplete);

    if (processComplete == 0) {
```

```
msg = "Restore Success";  
 } else {  
     msg = "Restore Error !";  
 }  
 logger.debug("Set msg");  
 } catch (Exception ex) {  
     ex.printStackTrace();  
     logger.debug("Exception: " + ex.getMessage());  
 }  
 logger.debug("return msg");  
 return msg;  
}  
  
}
```

Batch Name : Griffins

Candidate ID : 200019401866

Name of the Candidate contributed this UI: M.R.P.N.Tharuksha Rajapaksha

Title of the GUI: About

GUI Number : 39

Use case Name:



01. Brief Description of the GUI:

The main objective of this interface is to display about this system.

02. Justification of GUI component usage:

| Component | Usage | Justification |
|------------|----------------------|--|
| JPanel | To adjust alignments | <i>Using JPanel makes it easy to adjust the alignments of components. You can also create a component group and give them a title. Therefore all components are installed on JPanel.</i> |
| JLabel | Display image | <i>To add an image to beautify the interface</i> |
| JButtons | Close interface | <i>A button is required to close this interface. This JButton meets that requirement.</i> |
| JSeparator | Make a division | <i>To separate and display fields</i> |

PART 2:

| Topic | Sample Screen shots | Descriptions |
|------------------------|---------------------|--------------|
| Menu Creation | 01. | |
| | 02. | |
| | 03. | |
| Dialog Creation | 01. | |
| | 02. | |
| | 03. | |
| Popup menu Creation | 01. | |
| | 02. | |
| | 03. | |

Apply Theme



com.formdev.flatlaf.FlatIntelliJLaf
(FlatIntelliJLaf)

About.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package gui;
```

```
import com.formdev.flatlaf.FlatIntelliJLaf;
```

```
import javax.swing.UIManager;
```

```
import org.apache.log4j.Logger;
```

```
/**
```

```
*
```

```
* @author EZIO AUDITORY
```

```
*/
```

```
public class About extends javax.swing.JFrame {
```

```
/**
```

```
* Creates new form About
```

```
*/
```

```
static Logger logger = Logger.getLogger(About.class.getName());

public About() {

    initComponents();

    logger.debug("Access to About.java");

}

/**

 * 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();

    jSeparator1 = new javax.swing.JSeparator();

    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

    setTitle("About ");

    setResizable(false);

    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/AboutPic.jpg"))); // NOI18N

    jLabel1.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));

    jButton2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

    jButton2.setText("Close");

}
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {  
  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
        jButton2ActionPerformed(evt);  
  
    }  
  
});  
  
  
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
  
jPanel1.setLayout(jPanel1Layout);  
  
jPanel1Layout.setHorizontalGroup(  
  
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addGroup(jPanel1Layout.createSequentialGroup()  
  
.addComponentGap()  
  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  
  
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 540,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addComponent(jSeparator1)  
  
.addComponentGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))  
  
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()  
  
.addComponentGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
  
.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 100,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addComponentGap())  
  
);  
  
jPanel1Layout.setVerticalGroup(  
  
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
.addGroup(jPanel1Layout.createSequentialGroup()  
  
.addComponentGap()  
  
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 400,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
  
.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

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

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

);

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)
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

this.dispose();

logger.debug("About.java Dispose");

}

/***
 * @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 {  
        UIManager.setLookAndFeel(new FlatIntelliJLaf());  
        UIManager.put("Table.showHorizontalLines", true);  
        UIManager.put("Table.showVerticalLines", true);  
        UIManager.put("ScrollBar.showButtons", true);  
        UIManager.put("ScrollBar.width", 16);  
    } catch (Exception ex) {  
        System.err.println("Failed to initialize LaF");  
    }  
  
    // try {  
    //     for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  
    //         if ("Windows".equals(info.getName())) {  
    //             javax.swing.UIManager.setLookAndFeel(info.getClassName());  
    //             break;  
    //         }  
    //     }  
    // } catch (ClassNotFoundException ex) {  
    //     java.util.logging.Logger.getLogger(About.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    // } catch (InstantiationException ex) {  
    //     java.util.logging.Logger.getLogger(About.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    // } catch (IllegalAccessException ex) {  
    //     java.util.logging.Logger.getLogger(About.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    // } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
    //     java.util.logging.Logger.getLogger(About.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 About().setVisible(true);
    }
});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JSeparator jSeparator1;
// End of variables declaration

}

```

ComboBoxLoadModel.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```
import java.util.Vector;
import org.apache.log4j.Logger;

/**
 *
 * @author EZIO AUDITORY
 */
public class ComboBoxLoadModel {

    private Connection dbConnection;
    static Logger logger = Logger.getLogger(ComboBoxLoadModel.class.getName());

    public ComboBoxLoadModel(){
        dbConnection = database.dbConnection.getConnection();
    }

    public Vector districtComboBoxLoad() {
        String query = "SELECT * FROM district ORDER BY district_no";
        logger.debug("SQL query String: " + query);
        try {
            PreparedStatement psm = dbConnection.prepareStatement(query);
            ResultSet rs = psm.executeQuery();
            logger.debug("Query execution done");
            Vector v = new Vector();
            while(rs.next()){
                v.add(rs.getString("district_name"));
            }
            logger.debug("return Vector");
            return v;
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
        logger.error("Exception: " + e.getMessage());

    }

    logger.debug("return null");

    return null;

}

public Vector provinceComboBoxLoad() {

    String query = "SELECT * FROM province ORDER BY province_no";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        ResultSet rs = psm.executeQuery();

        logger.debug("Query execution done");

        Vector v = new Vector();

        while(rs.next()){

            v.add(rs.getString("province_name"));

        }

        logger.debug("return Vector");

        return v;

    } catch (Exception e) {

        e.printStackTrace();

        logger.error("Exception: " + e.getMessage());

    }

    logger.debug("return null");

    return null;

}

public Vector countryComboBoxLoad() {

    String query = "SELECT * FROM country ORDER BY country_no";

    logger.debug("SQL query String: " + query);
```

```
try {

    PreparedStatement psm = dbConnection.prepareStatement(query);

    ResultSet rs = psm.executeQuery();

    logger.debug("Query execution done");

    Vector v = new Vector();

    while(rs.next()){

        v.add(rs.getString("country_name"));

    }

    logger.debug("return Vector");

    return v;

} catch (Exception e) {

    e.printStackTrace();

    logger.error("Exception: " + e.getMessage());

}

logger.debug("return null");

return null;

}
```

```
public Vector companyComboBoxLoad() {

    String query = "SELECT * FROM company ORDER BY company_no";

    logger.debug("SQL query String: " + query);

    try {

        PreparedStatement psm = dbConnection.prepareStatement(query);

        ResultSet rs = psm.executeQuery();

        logger.debug("Query execution done");

        Vector v = new Vector();

        while(rs.next()){

            v.add(rs.getString("company_name"));

        }

        logger.debug("return Vector");

    }
```

```
    return v;

} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
}

public Vector jobroleComboBoxLoad() {

String query = "SELECT * FROM jobrole ORDER BY jobrole_no";
logger.debug("SQL query String: " + query);

try {
    PreparedStatement psm = dbConnection.prepareStatement(query);
    ResultSet rs = psm.executeQuery();
    logger.debug("Query execution done");
    Vector v = new Vector();
    while(rs.next()){
        v.add(rs.getString("jobrole_name"));
    }
    logger.debug("return Vector");
    return v;
} catch (Exception e) {
    e.printStackTrace();
    logger.error("Exception: " + e.getMessage());
}

logger.debug("return null");

return null;
}
```

```
public Vector unitComboBoxLoad() {  
    String query = "SELECT * FROM unit ORDER BY unit_no";  
    logger.debug("SQL query String: " + query);  
    try {  
        PreparedStatement psm = dbConnection.prepareStatement(query);  
        ResultSet rs = psm.executeQuery();  
        logger.debug("Query execution done");  
        Vector v = new Vector();  
        while(rs.next()){  
            v.add(rs.getString("unit_name"));  
        }  
        logger.debug("return Vector");  
        return v;  
    } catch (Exception e) {  
        e.printStackTrace();  
        logger.error("Exception: " + e.getMessage());  
    }  
    logger.debug("return null");  
    return null;  
}
```



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01**

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : GRN Report

Report Number : 01

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : GRN Invoice (Purchase Invoice)

PAMUNUWILA HARDWARE

PURCHASE INVOICE

GRN No : 1

Date : July 20, 2021

Supplier No : 1

Supplier Name : Nimal

| Pno | Product | Qty | Unit | Unit Buy Price | Discount | VAT | Amount |
|-----|---------|------|--------|----------------|----------|-----|--------|
| 1 | PVC | 50.0 | Number | 8.0 | 0.0 | 5.0 | 650.0 |
| 2 | LED | 50.0 | Number | 150.0 | 2.0 | 5.0 | 7650.0 |
| 3 | Tape | 50.0 | Number | 40.0 | 2.0 | 5.0 | 2150.0 |
| 4 | Enamal | 10.0 | Number | 170.0 | 0.0 | 5.0 | 1750.0 |

Thank You !

Come Again !

Item Count : 4

Discount : 200.0

Total Amount : 12200.0

Vat : 800.0

Net Amount : 12200.0

Email : pamunuwilahardware@gmail.com

Contact No : 0775561834

Description of the KPI :

This business has two parts, buying products and selling products. Suppliers supply products when purchasing products. At that point, the user accepts the products and inserts them into the system as a GRN. Here an invoice with data on the products purchased by the business is printed and issued to the supplier. In this case, the invoice is printed as the GRN that is included in the system to accept the relevant products.

Description of the Report :

The main objective of this invoice is to display the GRN number, the invoice issued date, details of the supplier who supplied the products to the business (supplier number, supplier name), product details supplied (quantity purchased, unit buy price, discount on each product, VAT on each product, product amount), item count, total discount, total VAT, total amount and net amount.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01**

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : GRN History Report

Report Number : 02

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : GRN History

PAMUNUWILA HARDWARE

GRN History

Date : July 20, 2021

Time : 2:59 AM

| Gno | Issue Date | Supplier | Item Count | Discount | VAT | Amount |
|-----|------------|----------|------------|----------|-------|---------|
| 1 | 2021/7/20 | Nimal | 4 | 200.0 | 800.0 | 12200.0 |
| 2 | 2021/7/20 | Kamal | 4 | 0.0 | 350.0 | 5910.0 |

Email : pamunuwilahardware@gmail.com

Contact No : 0775561834

Description of the KPI :

History records should be kept for good management in the business. The business records the purchase of products on the GRN. Therefore, a GRN history report is generated and printed to keep records of it.

Description of the Report :

The main objective of this report is to display the GRN number, the report generates date, the report generates time, the invoice issued date, details of the supplier who supplied the products to the business (supplier name), item count, total discount, total VAT, and total amount.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS**
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : GRN Item History Report

Report Number : 03

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : GRN Item History

PAMUNUWILA HARDWARE

GRN Item History

Date : July 20, 2021

Time : 3:00 AM

| Gno | Buy Date | Pno | Product | Qty | Unit | Unit Buy Price | Discount | VAT | Amount | Gno |
|-----|-----------|-----|---------|------|--------|----------------|----------|-----|--------|-----|
| 1 | 2021/7/20 | 1 | PVC | 50.0 | Number | 8.0 | 0.0 | 5.0 | 650.0 | 1 |
| 2 | 2021/7/20 | 2 | LED | 50.0 | Number | 150.0 | 2.0 | 5.0 | 7650.0 | 1 |
| 3 | 2021/7/20 | 3 | Tape | 50.0 | Number | 40.0 | 2.0 | 5.0 | 2150.0 | 1 |
| 4 | 2021/7/20 | 4 | Enamal | 10.0 | Number | 170.0 | 0.0 | 5.0 | 1750.0 | 1 |
| 5 | 2021/7/20 | 1 | PVC | 20.0 | Number | 8.0 | 0.0 | 5.0 | 260.0 | 2 |
| 6 | 2021/7/20 | 2 | LED | 20.0 | Number | 150.0 | 0.0 | 5.0 | 3100.0 | 2 |
| 7 | 2021/7/20 | 3 | Tape | 20.0 | Number | 40.0 | 0.0 | 5.0 | 900.0 | 2 |
| 8 | 2021/7/20 | 4 | Enamal | 10.0 | Number | 160.0 | 0.0 | 5.0 | 1650.0 | 2 |

Email : pamunuwilahardware@gmail.com

Contact No : 0775561834

Description of the KPI :

History records should be kept for good management in the business. The business records the purchase of products on the GRN. The business often buys the same product kind over and over again. Therefore, a GRN item history report is generated and printed to keep records of it.

Description of the Report :

The main objective of this report is to display the GRN item number, the report generates date, the report generates time, product buy date, product number, product name, quantity, unit, unit buy price, unit discount, unit VAT, unit amount and the GRN number.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01**

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : Invoice Report

Report Number : 04

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : Invoice

| PAMUNUWILA HARDWARE | | | | | | | |
|--|---------|------|--------|-------------------------|----------|-----|--------|
| INVOICE | | | | | | | |
| Invoice No : 1 | | | | Date : July 20, 2021 | | | |
| Customer No : 1 Customer Name : Sunil | | | | | | | |
| Pno | Product | Qty | Unit | Unit Sell Price | Discount | Tax | Amount |
| 1 | PVC | 10.0 | Number | 10.0 | 0.0 | 0.0 | 100.0 |
| 2 | LED | 5.0 | Number | 200.0 | 0.0 | 0.0 | 1000.0 |
| 3 | Tape | 10.0 | Number | 50.0 | 2.0 | 0.0 | 480.0 |
| 4 | Enamal | 2.0 | Number | 200.0 | 0.0 | 0.0 | 400.0 |
| Thank You ! | | | | | | | |
| Come Again ! | | | | | | | |
| Item Count : 4 | | | | Delivery Charge : 500.0 | | | |
| Discount : 20.0 | | | | Total Amount : 1980.0 | | | |
| Tax : 0.0 | | | | Net Amount : 1980.0 | | | |
| Email : pamunuwilahardware@gmail.com | | | | Contact No : 0775561834 | | | |

Description of the KPI :

This business has two parts, buying products and selling products. Customers buy products when they sell products. At that point, the user issues the product and inserts it into the system as an invoice. Here an invoice with the data of the products sold by the business is printed and issued to the customer. In this case, the invoice is printed as the invoice that is included in the system to sell the relevant products.

Description of the Report :

The main objective of this invoice is to display the invoice number, the invoice issued date, details of the customer who bought the products in the business (customer number, customer name), product details bought (the quantity bought, unit buy price, discount on each product, tax on each product, product amount), item count, total discount, total tax, total amount and net amount.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01**

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : Invoice History Report

Report Number : 05

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : Invoice History

| PAMUNUWILA HARDWARE | | | | | | | |
|----------------------------|------------|----------|------------|-----------------|----------|-----|--------|
| Invoice History | | | | | | | |
| <hr/> | | | | | | | |
| Date : July 20, 2021 | | | | | | | |
| Time : 3:03 AM | | | | | | | |
| Ino | Issue Date | Customer | Item Count | Delivery Charge | Discount | Tax | Amount |
| 1 | 2021/7/20 | Sunil | 4 | 500.0 | 20.0 | 0.0 | 1980.0 |
| 2 | 2021/7/20 | saman | 4 | 1000.0 | 500.0 | 0.0 | 6500.0 |

Email : pamunuwilahardware@gmail.com Contact No : 0775561834

Description of the KPI :

History records should be kept for good management in the business. The business records the sale of products on an invoice. Therefore, an invoice history report is generated and printed to keep records of it.

Description of the Report :

The main objective of this report is to display the invoice number, the report generates date, the report generates time, the invoice issued date, details of the customer who bought the products in the business (customer name), item count, delivery charge, total discount, total tax and total amount.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS**
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : Sales History Report

Report Number : 06

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : Sales History

PAMUNUWILA HARDWARE

Sales History

Date : July 20, 2021

Time : 3:03 AM

| Sno | Sell Date | Pno | Product | Qty | Unit | Unit Sell Price | Discount | Tax | Amount | Ino |
|-----|-----------|-----|---------|------|--------|-----------------|----------|-----|--------|-----|
| 1 | 2021/7/20 | 1 | PVC | 10.0 | Number | 10.0 | 0.0 | 0.0 | 100.0 | 1 |
| 2 | 2021/7/20 | 2 | LED | 5.0 | Number | 200.0 | 0.0 | 0.0 | 1000.0 | 1 |
| 3 | 2021/7/20 | 3 | Tape | 10.0 | Number | 50.0 | 2.0 | 0.0 | 480.0 | 1 |
| 4 | 2021/7/20 | 4 | Enamal | 2.0 | Number | 200.0 | 0.0 | 0.0 | 400.0 | 1 |
| 5 | 2021/7/20 | 1 | PVC | 50.0 | Number | 10.0 | 5.0 | 0.0 | 250.0 | 2 |
| 6 | 2021/7/20 | 2 | LED | 10.0 | Number | 200.0 | 0.0 | 0.0 | 2000.0 | 2 |
| 7 | 2021/7/20 | 3 | Tape | 50.0 | Number | 50.0 | 5.0 | 0.0 | 2250.0 | 2 |
| 8 | 2021/7/20 | 4 | Enamal | 5.0 | Number | 200.0 | 0.0 | 0.0 | 1000.0 | 2 |

Email : pamunuwilahardware@gmail.com

Contact No : 0775561834

Description of the KPI :

History records should be kept for good management in the business. The business records the sale of products on an invoice. The business often sells the same product kind over and over again. Therefore, a sales history report is generated and printed to keep records of it.

Description of the Report :

The main objective of this report is to display the sales number, the report generates date, the report generates time, product sell date, product Number, product name, quantity, unit, unit sell price, unit discount, unit tax, unit amount, and the invoice number.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS**
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : Delivery Invoice Report

Report Number : 07

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : Delivery Invoice

| PAMUNUWILA HARDWARE DELIVERY INVOICE | | | | |
|---|---------|------|----------------------|--------|
| Delivery No : 2 Invoice No : 1 | | | Date : July 20, 2021 | |
| Transporter No : 1 Transporter Name : Danith | | | | |
| Pno | Product | Qty | Unit | Amount |
| 1 | PVC | 10.0 | Number | 100.0 |
| 2 | LED | 5.0 | Number | 1000.0 |
| 3 | Tape | 10.0 | Number | 480.0 |
| 4 | Enamel | 2.0 | Number | 400.0 |

Thank You !
Come Again !

Customer No : 1 **Start Location : shop**
Customer Name : Sunil **End Location : Nigambo**
Contact No : 0776584289 **Delivery Charge : 500.0**

Email : pamunuwilahardware@gmail.com Contact No : 0775561834

Description of the KPI :

In this business, we have to deliver products to the customers regularly. Therefore, when a customer wants to deliver the products he has purchased, the user enters the relevant details into the system as a delivery. An invoice containing the details of the delivery is then printed and delivered to the transporter making the delivery.

Description of the Report :

The main objective of this invoice is to display the invoice number, the invoice issued date, details of the customer who bought the products in the business (customer number, customer name), product details bought, the quantity bought, unit buy price, discount on each product, tax on each product, product amount, item count, total discount, total tax, total amount and net amount.



**FORM FOR ANALYTICAL BUSINESS
INTELLIGENCE REPORTS
(INDIVIDUAL ASSIGNMENT) – H7DW 04/AS/01**

FORM NO: H7DW 04/AS/01/FRM-05

Report Name : Delivery History Report

Report Number : 08

Batch Name : Griffins

Candidate ID : 200019401866

Name of the group member contributed this Report : M.R.P.N.Tharuksha Rajapaksha

KPI Name : Delivery History

PAMUNUWILA HARDWARE

Delivery History

Date : July 20, 2021
Time : 3:06 AM

| Dno | Delivery Date | Transporter | Ino | Start Location | End Location | Delivery Charge |
|-----|---------------|-------------|-----|----------------|--------------|-----------------|
| 1 | 2021/7/20 | Danith | 2 | shop | Nigambo | 1000.0 |
| 2 | 2021/7/20 | Danith | 1 | shop | Nigambo | 500.0 |
| 3 | 2021/7/20 | Danith | 1 | shop | Nigambo | 500.0 |

Email : pamunuwilahardware@gmail.com

Contact No : 0775561834

Description of the KPI(50 words)(2 marks) :

History records should be kept for good management in the business. The business records deliveries on delivery. Therefore, a delivery invoice history report is generated and printed to keep records.

Description of the Report(100 words)(4 marks):

The main objective of this report is to display the delivery number, the report generates date, the report generates time, the invoice issued date, details of the transporter who delivered the products (transporter name), the invoice number, start location, end location, and delivery charge.

