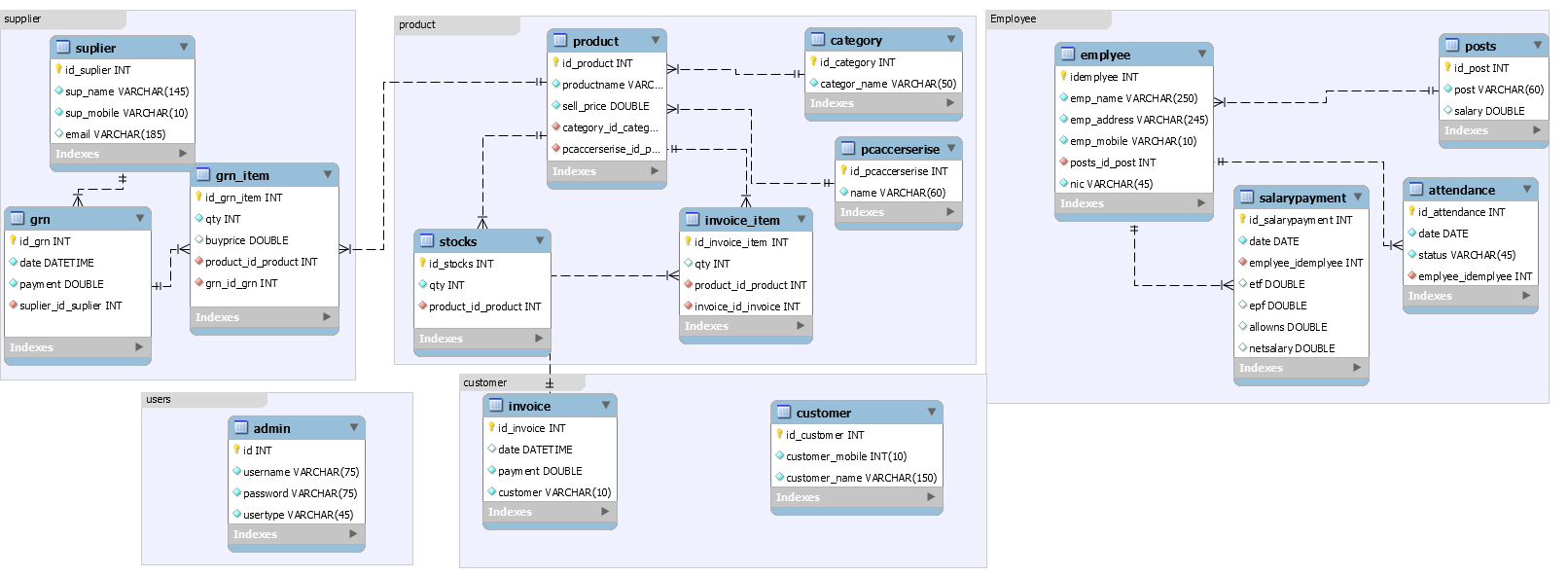


**ER Diagram**

****

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Welcome Page** | | **GUI Number** : 01 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   The opening page of the software is the welcome page. It takes some time for the other pages of the software and everything related to the software to load and that time can be seen by the users in the progress bar. It is 0% to 100% full and the openness of the software makes it viewable to all. After it is loaded, it is added to the user login page. The user cannot close or minimize the welcome page until the welcome page is loaded by opening the software welcome page shows to user computer shop name and shop logo.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | To showing loading text name and parentage value | When user open software and show value of loading parentage and loading text name | | Progress Bar | View loading software | Open the software and progress bar will fill o% to 100% | | | |

**Source Code**

welcomelodingpage sp = new welcomelodingpage();

sp.setVisible(true);

try{

for (int i = 0; i < 100; i++) {

Thread.sleep(60);

sp.loadingprasentageL.setText(i + "%");

if (i == 10) {

sp.jLabel2.setText("Loading....");

}

if (i == 20) {

sp.jLabel2.setText("Loading........");

}

if (i == 30) {

sp.jLabel2.setText("Loading.............");

}

if (i == 40) {

sp.jLabel2.setText("Loading....");

}

if (i == 50) {

sp.jLabel2.setText("Loading............");

}

if (i == 60) {

sp.jLabel2.setText("Loading.................");

}

if (i == 70) {

sp.jLabel2.setText("Connecting Successful");

}

if (i == 87) {

sp.jLabel2.setText("Launching Application");

}

sp.jProgressBar1.setValue(i);

}

new logingpage().setVisible(true);

sp.setVisible(false);

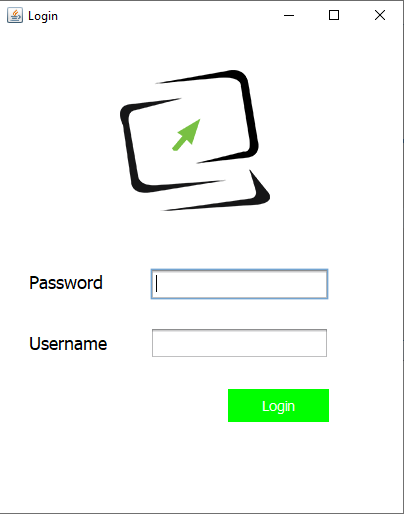
} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Logging Page** | | **GUI Number** : 02 |
| **Use case Name: login process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Once the welcome page is loaded, the user is taken to the login page. There the user will be taken to the home page of the software by entering the username and password. This software will be limited to 4 users. That is, the owner, manager, cashier and product controller assign a separate user type for those users, and at the time of logging into the use’s software, enter the username and password and click the login button to check user username and password Will be checked. Varies the accessibility of the software to each user. Also user can minimize and close this logging page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | To describe textFiled | Label shows to user what is a username filed and password filed | | jButton | Log into system homepage | When press button its get username and password then check username and password and if correct input data and move to home page | | | |

**Source Code**

****

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

String username = usernamefiled.getText();

char[] charpassword = passwordfiled.getPassword();

String password = new String(charpassword);

if(username.isEmpty()){

JOptionPane.showConfirmDialog(rootPane, "Plase Enter User Name", "Error", -1, 0);

}else if(password.isEmpty()){

JOptionPane.showConfirmDialog(rootPane, "Plase Enter Password", "Error", -1, 0);

}else{

String result =logtM.login(username, password);

logingpage p = new logingpage();

home h = new home();

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

this.dispose();

h.setVisible(true);

JOptionPane.showMessageDialog(this,"WelCome Home!" , "Message", JOptionPane.INFORMATION\_MESSAGE);

} else if (result.equals("manager")) {

this.dispose();

h.adminbt.setEnabled(false);

h.setVisible(true);

JOptionPane.showMessageDialog(this,"WelCome Home!" , "Message", JOptionPane.INFORMATION\_MESSAGE);

} else if (result.equals("cachier")) {

this.dispose();

h.adminbt.setEnabled(false);

h.supplierbt.setEnabled(false);

h.databasebt.setEnabled(false);

h.jButton1.setEnabled(false);

h.jButton4.setEnabled(false);

h.jButton11.setEnabled(false);

h.jButton40.setEnabled(false);

h.setVisible(true);

JOptionPane.showMessageDialog(this,"WelCome Home!" , "Message", JOptionPane.INFORMATION\_MESSAGE);

} else if (result.equals("prodcut")) {

this.dispose();

h.adminbt.setEnabled(false);

h.databasebt.setEnabled(false);

h.jButton1.setEnabled(false);

//h.jButton62.setEnabled(false);

h.jButton90.setEnabled(false);

h.jButton2.setEnabled(false);

h.customerbt.setEnabled(false);

h.setVisible(true);

JOptionPane.showMessageDialog(this,"WelCome Home!" , "Message", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this,"Please Check Username or Password " , "Message", JOptionPane.INFORMATION\_MESSAGE);

usernamefiled.setText("");

passwordfiled.setText("");

}

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

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

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(logingpage.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 logingpage().setVisible(true);

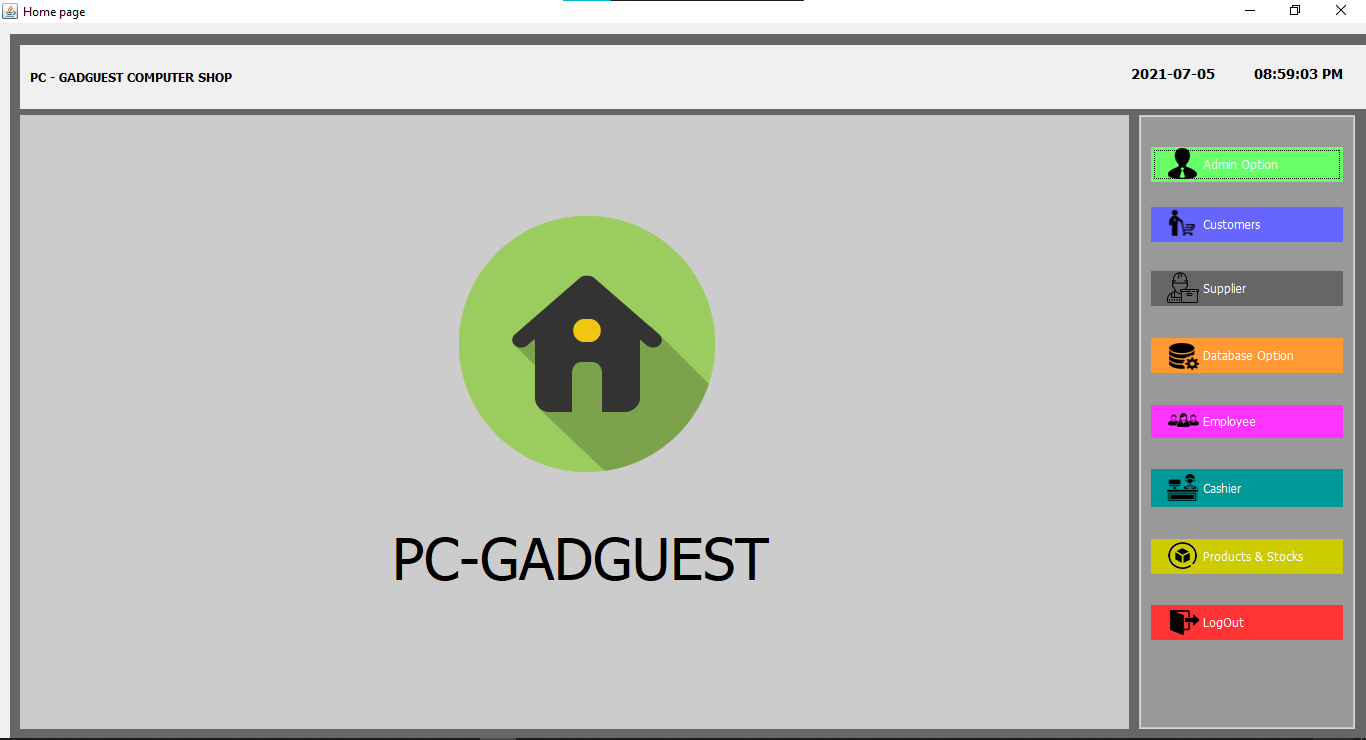
}

});

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Home Page** | | **GUI Number** : 03 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   After checking the username, password and user type on the login page, the user is redirected to the home page. After the user comes to the home page, the user will see the shop name and the home logo. And the user has the ability to know the date and time through the software. The user can know the date and time on any page. On the right side of the home page, you can see the relevant section of the software, namely admin option, customer option, supplier, database, employee, cashier, product and stock and logout button. The user can see those sections from any page   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | To using to view date and time and company name | When move to home page and show current date and time and view shop name also using jlable to set image(home icon) | | Desktop Pane | To add internal frame | Desktop pane added to jframe because use multiple internal frames | | Internal frames | To view shop management sections | When click specified button (eg, admin) and call specified internal frame and view admin graphical user interface | | jButton | Move to specific shop management sections | When user Click home page button and move to the specified section  Eg:  Customer section, admin section, database option, etc.…. | | | |

**Source Code**

****

void showdate() {

Date d = new Date();

SimpleDateFormat s = new SimpleDateFormat("yyyy-MM-dd");

date.setText(s.format(d));

}

void showtime() {

new Timer(0, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Date d = new Date();

SimpleDateFormat s = new SimpleDateFormat("hh:mm:ss a");

time.setText(s.format(d));

}

}).start();

}



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

f1.setVisible(false);

f2.setVisible(true);

}



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

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(true); }



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

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(false);

f4.setVisible(false);

f5.setVisible(false);

f6.setVisible(false);

f7.setVisible(false);

f8.setVisible(true);

}

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

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(false);

f4.setVisible(true); }



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

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(false);

f4.setVisible(false);

f5.setVisible(false);

f6.setVisible(true);

f7.setVisible(false);

f8.setVisible(false);

}



private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(false);

f4.setVisible(false);

f5.setVisible(true);

}



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

f1.setVisible(false);

f2.setVisible(false);

f3.setVisible(false);

f4.setVisible(false);

f5.setVisible(false);

f6.setVisible(false);

f7.setVisible(true);

f8.setVisible(false);

}



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

new logingpage().setVisible(true);

}



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

delete(attentionComboBox.getSelectedIndex());

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Admin page** | | **GUI Number** : 04 |
| **Use case Name: Admin Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Only the shop owner has access to the admin page .This creates the user who uses the software. They happen to the shop owner. Select the username, password and user type from the combo bon and when the internet button is clicked, if the data is inserted correctly, the information message will be displayed and the data will be added to the table on the right side. When updating or deleting, you must first select the relevant row from the table and then update or delete and the user will be able to search for the entered user details.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fileds to users | When move to home page and show current date and time and view shop name also using jlable to set image(home icon) | | Combo Box | List down user types | When add new user and user can select useer type easily and manage data | | Joption pane | View messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert Users Details | When add new user and show user details in jtable and also apply update and delete also search some user and view result in jtable | | | |

**Source Code**



private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

adminIDfiled.setText("");

adminusernameField.setText("");

passwordField.setText("");

adminM.loadtableAdmin("", admintdm);

}



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

String aId = adminIDfiled.getText();

String username = adminusernameField.getText();

char[] passwordArr = passwordField.getPassword();

String password = new String(passwordArr, 0, passwordArr.length);

Object a = admintypeCombo.getSelectedItem();

String usertype = String.valueOf(a);

if (username.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Username", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (password.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Password", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insertAdmin = adminM.insertAdmin(username, password,usertype);

JOptionPane.showMessageDialog(this, "Insert Details", "massege", JOptionPane.INFORMATION\_MESSAGE);

adminusernameField.setText("");

passwordField.setText("");

adminM.loadtableAdmin("", admintdm);

}

}



private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String aId = adminIDfiled.getText();

String username = adminusernameField.getText();

char[] passwordArr = passwordField.getPassword();

String password = new String(passwordArr, 0, passwordArr.length);

Object a = admintypeCombo.getSelectedItem();

String usertype = String.valueOf(a);

if (username.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds are emply?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (password.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds are emply?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String updateAdmin = adminM.updateadmin(aId, username, password,usertype);

JOptionPane.showMessageDialog(this, "Updated Details", "massege", JOptionPane.OK\_CANCEL\_OPTION);

adminIDfiled.setText("");

adminusernameField.setText("");

passwordField.setText("");

adminM.loadtableAdmin("", admintdm);

}

}



private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

String aId = adminIDfiled.getText();

String username = adminusernameField.getText();

if (username.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select data From Table!", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String deleteAdmin = adminM.deleteAdmin(aId);

JOptionPane.showMessageDialog(this, "Deleted Details", "massege", JOptionPane.QUESTION\_MESSAGE);

adminIDfiled.setText("");

adminusernameField.setText("");

passwordField.setText("");

adminM.loadtableAdmin("", admintdm);

}

}

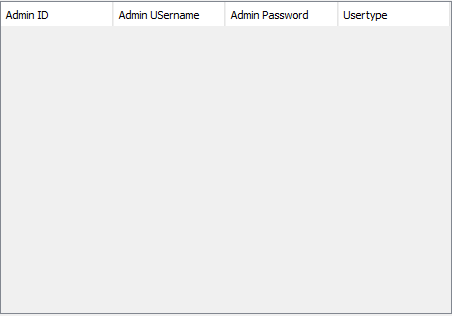


private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = adminSearchFiled.getText();

adminM.loadtableAdmin(keyword, admintdm);

}



private void admintableMouseClicked(java.awt.event.MouseEvent evt) {

int row = admintable.getSelectedRow();

int adminId = (int) admintdm.getValueAt(row, 0);

Object[] admindata = adminM.getSelectId(adminId);

// adminIDfiled.setText(new String(admindata[0]+""));

adminIDfiled.setText(String.valueOf(admindata[0]));

adminusernameField.setText((String) admindata[1]);

passwordField.setText((String) admindata[2]);

admintypeCombo.setSelectedItem((String) admindata[3]);

}

**Model Package code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class adminModel {

static Logger log = Logger.getLogger(adminModel.class.getName());

Connection con;

public adminModel() {

con= db.dbConnection.getconnection();

}

public String insertAdmin(String username, String password, String usertype) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.admin(username, password,usertype)VALUES(?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,username);

ps.setString(2,password);

ps.setString(3,usertype);

ps.execute();

msg="Success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="error"+e.getMessage();

}

return null;

}

public void loadtableAdmin(String keyword, DefaultTableModel admintdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.admin WHERE username like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

admintdm.setRowCount(0);

while (rs.next()) {

int id = rs.getInt("id");

String username = rs.getString("username");

String password = rs.getString("password");

String usertype = rs.getString("usertype");

rowData = new Object[]{id, username, password,usertype};

admintdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getSelectId(int adminId) {

String getData ="SELECT \* FROM pcgadguestshop.admin WHERE id =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setInt(1, adminId);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

int adminid = rs.getInt("id");

String username = rs.getString("username");

String password = rs.getString("password");

String usertype = rs.getString("usertype");

rowData = new Object[]{adminid,username,password,usertype};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

}

return null;

}

public String updateadmin(String aId, String username, String password, String usertype) {

String msg;

String updateQuary="UPDATE pcgadguestshop.admin SET username=?, password=?,usertype=? WHERE id=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1, username);

ps.setString(2, password);

ps.setString(3, usertype);

ps.setString(4, aId);

ps.execute();

msg="Success";

} catch (Exception e) {

e.printStackTrace();

msg = "Error"+e.getMessage();

log.debug(e.getMessage());

}

return null;

}

public String deleteAdmin(String aId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.admin WHERE id=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, aId);

ps.execute();

msg="success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Customer page** | | **GUI Number** : 05 |
| **Use case Name: Customer Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Customers who come to the shop can register through the customer page. Enter the customer's name and phone number. The phone number will be used as the primary key. After entering the customer into system 1, the table on the right will be displayed. In order to update or delete the client to be added, the relevant clients must be selected from the table. As a customer can search by his phone number. This page is accessible only to the owner, manager and cashier. After every process will done successfully (insert, update and delete) view information dialog box to user.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize | When adding data to fields label helps to recognize each fileds | | Joption pane | View messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert Customer Details | When add new customer and show customer details in jtable and also apply update and delete also search some customer and view result in jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Can get report all registered customer details |

**Source Code**

****

private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

customerM.loadtableAdmin("", customertdm);

customerM.loadtableAdmin("", customer2tdm);

}



private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

String cusId = customerIDfiled.getText();

String customername = customernameFiled.getText();

String customerModile = customerMdeilfiled.getText();

if (customerModile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Customer Name", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (customername.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Customer Name", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertcustomer = customerM.insertCustomer(customername, customerModile);

JOptionPane.showConfirmDialog(rootPane, insertcustomer, "Massege", -1, 1);

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

customerM.loadtableAdmin("", customertdm);

customerM.loadtableAdmin("", customer2tdm);

}

}



private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

String cusId = customerIDfiled.getText();

String customername = customernameFiled.getText();

String customerModile = customerMdeilfiled.getText();

if (customername.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds ", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (customerModile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds ", "massege", JOptionPane.INFORMATION\_MESSAGE);

}else {

String updateCustomer = customerM.updateCustomer(cusId, customername, customerModile);

JOptionPane.showConfirmDialog(rootPane, updateCustomer, "Massege", -1, 1);

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

customerM.loadtableAdmin("", customertdm);

customerM.loadtableAdmin("", customer2tdm);

}

}



private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {

String cusId = customerIDfiled.getText();

String customername = customernameFiled.getText();

if (customername.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select data from table! ", "massege", JOptionPane.QUESTION\_MESSAGE);

System.out.println("cuid" + cusId);

} else {

String deleteCustomer = customerM.deleteCustomer(cusId);

JOptionPane.showMessageDialog(this, deleteCustomer, "massege", JOptionPane.QUESTION\_MESSAGE);

System.out.println("cuid" + cusId);

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

customerM.loadtableAdmin("", customertdm);

customerM.loadtableAdmin("", customer2tdm);

}

}

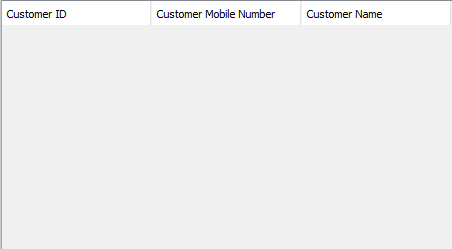


private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = customerSearchfiled.getText();

customerM.loadtableAdmin(keyword, customertdm);

}



private void customertableMouseClicked(java.awt.event.MouseEvent evt) {

int row = customertable.getSelectedRow();

String cusId = (String) customertdm.getValueAt(row, 0);

Object[] customerdata = customerM.getSelectId(cusId);

customerIDfiled.setText(String.valueOf(customerdata[0]));

customerMdeilfiled.setText((String) customerdata[1]);

customernameFiled.setText((String) customerdata[2]);

}



private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class customerModel {

static Logger log = Logger.getLogger(customerModel.class.getName());

Connection con;

public customerModel() {

con= db.dbConnection.getconnection();

}

public String insertCustomer(String customername, String customerModile) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.customer(customer\_mobile, customer\_name)VALUES(?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,customerModile);

ps.setString(2,customername);

ps.execute();

msg="Successfully Insert Customer";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Mobile Number Already taken!";

}

return msg;

}

public void loadtableAdmin(String keyword, DefaultTableModel customertdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.customer WHERE customer\_mobile like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

customertdm.setRowCount(0);

while (rs.next()) {

String idCustomer = rs.getString("id\_customer");

String customerMobile = rs.getString("customer\_mobile");

String customername = rs.getString("customer\_name");

rowData = new Object[]{idCustomer,customerMobile, customername};

customertdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getSelectId(String cusId) {

String getData ="SELECT \* FROM pcgadguestshop.customer WHERE id\_customer =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, cusId);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String idCustomer = rs.getString("id\_customer");

String customerMobile = rs.getString("customer\_mobile");

String customerName = rs.getString("customer\_name");

rowData = new Object[]{idCustomer,customerMobile,customerName};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String updateCustomer(String cusId, String customername, String customerModile) {

String msg;

String updateQuary="UPDATE pcgadguestshop.customer SET customer\_mobile=?, customer\_name=? WHERE id\_customer=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1, customerModile);

ps.setString(2, customername);

ps.setString(3, cusId);

ps.execute();

msg="Successfully Updated Customer";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Cannot Update Mobile Number Already taken!";

}

return msg;

}

public String deleteCustomer(String cusId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.customer WHERE id\_customer=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, cusId);

ps.execute();

msg="Successfully Delete Customer";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Database page(backup page)** | | **GUI Number** : 06 |
| **Use case Name: Database Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Database page have two sub pagers. Backup database page can backup system database to computer or laptop .there are main two button in this page .one button is select path for save backup database file in computer and selected path will show in text field .clicked backup button bye user and create a backup of database will save in selected location and save entire database with their saved data and show information message if successfully backup database. Only Owner have to backup database access in this system because if everyone have database access there can be happen some Fraud.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | To View image | Using jlable to view image | | Joption pane | View messages | When user do some activity (backup DB)after show message above activity | | jButton | It use for set location and back database | When click backup path and can set location after click backup save db selected location | | | |

**Source Code**

****

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

JFileChooser backupPathfilebt = new JFileChooser();

int returnval = backupPathfilebt.showOpenDialog(this);

if (returnval == JFileChooser.APPROVE\_OPTION) {

String selectedFilename = backupPathfilebt.getSelectedFile().getName();

String selectedFilepath = backupPathfilebt.getCurrentDirectory().getPath();

String backuppath = selectedFilepath + "//" + selectedFilename;

backupField.setText(backuppath);

}}



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

String backuppath = backupField.getText();

System.out.println(backuppath);

String result = new backResoreCode().backupDb(backuppath);

JOptionPane.showConfirmDialog(backbt, result, "info", -1, 1);

backupField.setText("");

}

**Model Package Code**

package model;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class backResoreCode {

static Logger log = Logger.getLogger(backResoreCode.class.getName());

public String backupDb(String backuppath){

String msg = null;

try {

Runtime runtime = Runtime.getRuntime();

Process p =runtime.exec("C:\\Program Files\\MySQL\\MySQL Server 8.0\\bin\\mysqldump.exe -uroot -pchamodjava.01 --add-drop-database --port=3307 -B school\_managment -r "+backuppath);

int processComplete = p.waitFor();

System.out.println(processComplete);

msg ="Success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return msg;

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Database page(Restore page)** | | **GUI Number** : 07 |
| **Use case Name: Database Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Other page is database restore. there are have three components. There are have main three components (two buttons and one text filed). When user wants to restore database first he must select database from computer for this activity can do in restore path button after user selected path will show in text filed and finally click restore button if correct insert to the system and show information massage to user. In this page also have same access authority to only shop owner for security purposes.     1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | To View image | Using jlable to view image | | Joption pane | View messages | When user do some activity (restore DB)after show message above activity | | jButton | It use for set location and back database | When click Restore path and can set location after click restore button and apply db to system | | | |

**Source Code**

****

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

JFileChooser restorePathfilebt = new JFileChooser();

int returnvall = restorePathfilebt.showOpenDialog(this);

if (returnvall == JFileChooser.APPROVE\_OPTION) {

String selectedFilename = restorePathfilebt.getSelectedFile().getName();

String selectedFilepath = restorePathfilebt.getCurrentDirectory().getPath();

String resotre = selectedFilepath + "//" + selectedFilename;

restoreField.setText(resotre);

}

}



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

String restorepath = restoreField.getText();

System.out.println(restorepath);

String result = new backResoreCode().restore(restorepath);

JOptionPane.showConfirmDialog(restorebt, result, "info", -1, 1);

restoreField.setText("");

}

**Model Package Code**

package model;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public String restore(String restorepath){

String msg = null;

try {

Runtime runtime = Runtime.getRuntime();

String [] restoreCmd =new String[]{"C:\\Program Files\\MySQL\\MySQL Server 8.0\\bin\\mysql.exe",

"--user=root" ,

"--password=chamodjava.01" ,

"--port=3307" ,

"-e" ,

" source " + restorepath};

Process p =runtime.exec(restoreCmd);

int processComplete = p.waitFor();

System.out.println(processComplete);

msg ="Success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (Supplier Registration )** | | **GUI Number** : 08 |
| **Use case Name: supplier Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   There is a possibility of registering people who supply goods for the business. Those people must enter their name, phone number and email when registering. However, when registering, the email is not considered as mandatory information. If you want to update or delete the supplier's information, you can update or delete the selected row from the table containing the supplier details on the register side. Here is the ability to search. This page is accessible to owner, manager and product controller. After every process will done successfully (insert, update and delete) view information dialog box to user.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize | When adding data to fields label helps to recognize each fileds | | Joption pane | View messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View registered supplier details | When add new supplier and show supplier details in jtable and also apply update and delete also search using keyword and view result in jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Click report button wen open report Dialog box and can get report all registered Suppliers details |

**Source Code**



private void jButton68ActionPerformed(java.awt.event.ActionEvent evt) {

supplierIDFiled.setText("");

suppnameFiled.setText("");

supMobileFiled.setText("");

supEmailField.setText("");

suplierM.loadtable("", supliertdm);

component();

}



private void jButton65ActionPerformed(java.awt.event.ActionEvent evt) {

String sid = supplierIDFiled.getText();

String sname = suppnameFiled.getText();

String sMbile = supMobileFiled.getText();

String sEmail = supEmailField.getText();

if (sname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert supplier name", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (sMbile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert Supplier Mobile", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String inserts = suplierM.inserSuplier(sname, sMbile, sEmail);

JOptionPane.showMessageDialog(this, inserts, "massege", JOptionPane.INFORMATION\_MESSAGE);

supplierIDFiled.setText("");

suppnameFiled.setText("");

supMobileFiled.setText("");

supEmailField.setText("");

suplierM.loadtable("", supliertdm);

component();

}

}



private void jButton66ActionPerformed(java.awt.event.ActionEvent evt) {

String sid = supplierIDFiled.getText();

String sname = suppnameFiled.getText();

String sMbile = supMobileFiled.getText();

String sEmail = supEmailField.getText();

if (sMbile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From Table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String update = suplierM.updateSuplier(sid, sname, sMbile, sEmail);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.OK\_CANCEL\_OPTION);

supplierIDFiled.setText("");

suppnameFiled.setText("");

supMobileFiled.setText("");

supEmailField.setText("");

suplierM.loadtable("", supliertdm);

component();

}



private void jButton67ActionPerformed(java.awt.event.ActionEvent evt) {

String sid = supplierIDFiled.getText();

String delete = suplierM.deleteSuplier(sid);

String sMbile = supMobileFiled.getText();

if (sMbile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From Table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, delete, "massege", JOptionPane.QUESTION\_MESSAGE);

supplierIDFiled.setText("");

suppnameFiled.setText("");

supMobileFiled.setText("");

supEmailField.setText("");

suplierM.loadtable("", supliertdm);

component();

}

}

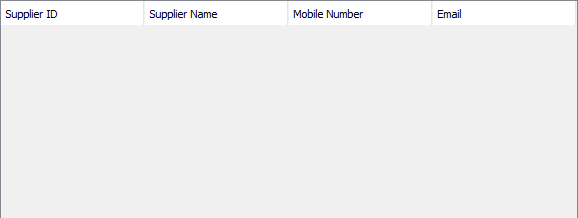


private void jButton69ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = supSearchFiled.getText();

suplierM.loadtable(keyword, supliertdm);

}



private void suplierTableMouseClicked(java.awt.event.MouseEvent evt) {

int row = suplierTable.getSelectedRow();

String sid = (String) supliertdm.getValueAt(row, 0);

Object[] supllierdata = suplierM.detselectid(sid);

supplierIDFiled.setText(String.valueOf(supllierdata[0]));

suppnameFiled.setText((String) supllierdata[1]);

supMobileFiled.setText((String) supllierdata[2]);

supEmailField.setText((String) supllierdata[3]);

}



private void jButton72ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class supplierModel {

static Logger log = Logger.getLogger(supplierModel.class.getName());

Connection con;

public supplierModel() {

con = db.dbConnection.getconnection();

}

public String inserSuplier(String sname, String sMbile, String sEmail) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.suplier(sup\_name, sup\_mobile, email)VALUES(?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,sname);

ps.setString(2,sMbile);

ps.setString(3,sEmail);

ps.execute();

msg="Successfully Insert Supplier";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Mobile Number Aready taken!";

}

return msg;

}

public String updateSuplier(String sid, String sname, String sMbile, String sEmail) {

String msg;

String updateQuary="UPDATE pcgadguestshop.suplier SET sup\_name=?, sup\_mobile=?, email=? WHERE id\_suplier=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1,sname);

ps.setString(2,sMbile);

ps.setString(3,sEmail);

ps.setString(4,sid);

ps.execute();

msg="Successfully Updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Mobile Number Aready taken!";

}

return msg;

}

public String deleteSuplier(String sid) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.suplier WHERE id\_suplier=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, sid);

ps.execute();

msg="successfully Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Cannot Delete,Supplier Number Already Using";

}

return null;

}

public void loadtable(String keyword, DefaultTableModel supliertdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.suplier WHERE sup\_name like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

supliertdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_suplier");

String name = rs.getString("sup\_name");

String mobile = rs.getString("sup\_mobile");

String email = rs.getString("email");

rowData = new Object[]{id,name,mobile,email};

supliertdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] detselectid(String sid) {

String getData ="SELECT \* FROM pcgadguestshop.suplier WHERE id\_suplier =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, sid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String id = rs.getString("id\_suplier");

String name = rs.getString("sup\_name");

String mobile = rs.getString("sup\_mobile");

String email = rs.getString("email");

rowData = new Object[]{id,name,mobile,email};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Page )** | | **GUI Number** : 09 |
| **Use case Name: GRN and GRN Item Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   The GOND receive note is used to indicate information about the goods delivered to the suppliers shop .It is the GOND received note and received. Items are included in the GOND recive item page .To fill in the GRN, you need to enter 3 items namely supplier name, date and payment .If you want to update or delete, you can select the data row from the table and update or delete. And the GRN on this page. This page is also accessible only to those who have access to the supplier page. After every process will done successfully (insert, update and delete) view information dialog box to user.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | ComboBox | List down supplier name | When insert or update grn and can get registered supplier name using combo box | | jtable | View insert Good Receive notes details | When add new Good Receive note and insert grn details in jtable and also apply update and delete also search using keyword and show result inside jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | When click supplier button and show registered supplier details and can search supplies using their name |
|  |  |  | Click report button wen open report Dialog box and Can get report all insert good receive notes details and can get report with receive items |

**Source Code**



private void jButton78ActionPerformed(java.awt.event.ActionEvent evt) {

grnIdFiled.setText("");

grndateChooser.setDate(null);

grnAmountField.setText("");

grnM.loadtable("", grntdm);

}



private void jButton75ActionPerformed(java.awt.event.ActionEvent evt) {

String grnid = grnIdFiled.getText();

Date grndate = grndateChooser.getDate();

String amunt = grnAmountField.getText();

Object o = grnsupplerCombo.getSelectedItem();

String suplier = String.valueOf(o);

if (amunt.isEmpty()) {

JOptionPane.showMessageDialog(this, "Pleace Check Data input fields are empty?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insert = grnM.insertgrn(grndate, amunt, suplier);

JOptionPane.showMessageDialog(this, insert, "massege", JOptionPane.INFORMATION\_MESSAGE);

grnIdFiled.setText("");

grndateChooser.setDate(null);

grnAmountField.setText("");

grnM.loadtable("", grntdm);

}

}



private void jButton76ActionPerformed(java.awt.event.ActionEvent evt) {

String grnid = grnIdFiled.getText();

Date grndate = grndateChooser.getDate();

String amunt = grnAmountField.getText();

Object o = grnsupplerCombo.getSelectedItem();

String suplier = String.valueOf(o);

if (amunt.isEmpty()) {

JOptionPane.showMessageDialog(this, "Pleace Select data from table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String update = grnM.updategrn(grnid, grndate, amunt, suplier);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.OK\_CANCEL\_OPTION);

grnIdFiled.setText("");

grndateChooser.setDate(null);

grnAmountField.setText("");

grnM.loadtable("", grntdm);

}

}



private void jButton77ActionPerformed(java.awt.event.ActionEvent evt) {

String grnid = grnIdFiled.getText();

String amunt = grnAmountField.getText();

if (amunt.isEmpty()) {

JOptionPane.showMessageDialog(this, "Pleace Select data from table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String delete = grnM.deletegrn(grnid);

JOptionPane.showMessageDialog(this, delete, "massege", JOptionPane.QUESTION\_MESSAGE);

grnIdFiled.setText("");

grndateChooser.setDate(null);

grnAmountField.setText("");

grnM.loadtable("", grntdm);

}

}



private void jButton85ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

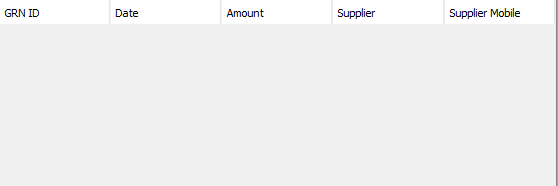


private void jButton79ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = grnSearchFiled.getText();

grnM.loadtable(keyword, grntdm);

}



private void grntableMouseClicked(java.awt.event.MouseEvent evt) {

int row = grntable.getSelectedRow();

String grid = (String) grntdm.getValueAt(row, 0);

Object[] grndata = grnM.getselectid(grid);

grnIdFiled.setText(String.valueOf(grndata[0]));

grndateChooser.setDate((Date) grndata[1]);

grnAmountField.setText((String) grndata[2]);

grnsupplerCombo.setSelectedItem((String) grndata[3]);

}



private void jButton73ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class grnModel {

static Logger log = Logger.getLogger(grnModel.class.getName());

Connection con;

public grnModel() {

con=db.dbConnection.getconnection();

}

public String insertgrn(Date grndate, String amunt, String suplier) {

String msg="";

String supid ="SELECT id\_suplier FROM pcgadguestshop.suplier WHERE sup\_name=?";

String insertQuery ="INSERT INTO pcgadguestshop.grn(date, payment, suplier\_id\_suplier)VALUES(?,?,?)";

try {

PreparedStatement ps2 = con.prepareStatement(supid);

ps2.setString(1, suplier);

ResultSet rs = ps2.executeQuery();

while(rs.next()){

int re =rs.getInt("id\_suplier");

String supplierid = String.valueOf(re);

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setDate(1,new java.sql.Date(grndate.getTime()));

ps.setString(2,amunt);

ps.setString(3,supplierid);

ps.execute();

msg="Successfully Insert";

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="error";

}

return msg;

}

public String updategrn(String grnid, Date grndate, String amunt, String suplier) {

String msg="";

String supid ="SELECT id\_suplier FROM pcgadguestshop.suplier WHERE sup\_name=?";

String updateQuary="UPDATE pcgadguestshop.grn SET date=?, payment=?, suplier\_id\_suplier=? WHERE id\_grn=?";

try {

PreparedStatement ps2 = con.prepareStatement(supid);

ps2.setString(1, suplier);

ResultSet rs = ps2.executeQuery();

while(rs.next()){

int re =rs.getInt("id\_suplier");

String supplierid = String.valueOf(re);

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setDate(1,new java.sql.Date(grndate.getTime()));

ps.setString(2,amunt);

ps.setString(3,supplierid);

ps.setString(4,grnid);

ps.execute();

msg="Successfully Updated";

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Error"+e.getMessage();

}

return msg;

}

public String deletegrn(String grnid) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.grn WHERE id\_grn=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, grnid);

ps.execute();

msg="success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return null;

}

public void loadtable(String keyword, DefaultTableModel grntdm) {

String loadtb = "SELECT suplier.sup\_name,grn.suplier\_id\_suplier,suplier.id\_suplier,suplier.sup\_mobile,grn.payment,grn.date,id\_grn "

+ "FROM pcgadguestshop.suplier INNER JOIN pcgadguestshop.grn ON pcgadguestshop.grn.suplier\_id\_suplier = pcgadguestshop.suplier.id\_suplier WHERE id\_grn like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

grntdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_grn");

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

String payment = rs.getString("payment");

String name = rs.getString("sup\_name");

String mobile = rs.getString("sup\_mobile");

rowData = new Object[]{id,date,payment,name,mobile};

grntdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselectid(String grid) {

String getData ="SELECT \* FROM pcgadguestshop.grn WHERE id\_grn = ?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, grid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

int id = rs.getInt("id\_grn");

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

String payment = rs.getString("payment");

String sid = rs.getString("suplier\_id\_suplier");

String getpostname ="SELECT sup\_name FROM pcgadguestshop.suplier WHERE id\_suplier=? ";

PreparedStatement ps2 = con.prepareStatement(getpostname);

ps2.setString(1, sid);

ResultSet rs2 =ps2.executeQuery();

while(rs2.next()){

String supname = rs2.getString("sup\_name");

sid=supname;

rowData = new Object[]{id,date,payment,sid};

return rowData;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Page. supplier view dialog page )** | | **GUI Number** : 10 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This frame includes in good receive note page in front of supplier id index. When use input some good receive note when use wants to get supplier registration id number from system It may help to user get supplier id number from supplier dialog box. Just click supplier button then will open this box if user wants search some supplier’s details from system. User can enter supplier name in supplier dialog box and click search button in this box and result will show in this supplier table. And user can clear search input filed if there is not any keyword and click search button show all registered supplier details in supplier table. Owner, manager and product controller can access this subpage.     1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | It use for search by entered keyword | Get keyword in input field and search using entered keyword | | jtable | View registered supplier details | List down registered supplier details. if search using supplier name and result will show inside jtable | | | |

**Source Code**

****

private void jButton69ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = supSearchFiled3.getText();

suplierM.loadtable(keyword, supliertdm);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Item Page )** | | **GUI Number** : 11 |
| **Use case Name: GRN and GRN Item Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   After accepting the supplier's goods, enter the payment details and then the GRN page using that ID and enter the OD, QTY and by price of the product obtained. The relevant product can be entered into the system. The ID of the relevant product is the product button. You can get the ID of the registered products in the shop by clicking on one. When QTY is added to that product, the QTY associated with the ID in the stock table changes. Similarly, stocks are updated for update and delete. Supplier, owner, manager and product controller have access. After every process will done successfully (insert, update and delete) view information dialog box to user.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert Good Receive notes items details | When add new Good Receive note items and insert grn details shows in jtable and also apply update and delete and search using keyword and show result inside jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | When click product button and show registered product details and can search product using product name |
|  |  |  | When click grn button and show insert grn details and can search grn using grn id |
|  |  |  | Click report button wen open report Dialog box and Can get report all insert good receive items details |

**Source Code**

****

private void jButton84ActionPerformed(java.awt.event.ActionEvent evt) {

grnItemIdFiled.setText("");

grnitemPIdFiled1.setText("");

grnitemQtyFiled.setText("");

grnitembuypFiled.setText("");

grnitemGrnIdFeild.setText("");

// grnMfdDateChooser.setDate(null);

grnitemM.loadtable("", grnitemtdm);

}



private void jButton81ActionPerformed(java.awt.event.ActionEvent evt) {

String grnitemid = grnItemIdFiled.getText();

String pid = grnitemPIdFiled1.getText();

String qty = grnitemQtyFiled.getText();

String buyprice = grnitembuypFiled.getText();

String grnid = grnitemGrnIdFeild.getText();

// Date grnMfd = grnMfdDateChooser.getDate();

if (pid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert Product ID", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (qty.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert qty", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (buyprice.isEmpty()) {

JOptionPane.showMessageDialog(this, "Plaase Insert buprice", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (grnid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert GRN ID", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insert = grnitemM.insergrnitem(pid, qty, buyprice, grnid);

JOptionPane.showMessageDialog(this, "Insert Details", "massege", JOptionPane.INFORMATION\_MESSAGE);

grnItemIdFiled.setText("");

grnitemPIdFiled1.setText("");

grnitemQtyFiled.setText("");

grnitembuypFiled.setText("");

grnitemGrnIdFeild.setText("");

grnitemM.loadtable("", grnitemtdm);

stockM.loadStocktb("", stocktdm);

}

}



private void jButton82ActionPerformed(java.awt.event.ActionEvent evt) {

String grnitemid = grnItemIdFiled.getText();

String pid = grnitemPIdFiled1.getText();

String qty = grnitemQtyFiled.getText();

String buyprice = grnitembuypFiled.getText();

String grnid = grnitemGrnIdFeild.getText();

// Date grnMfd = grnMfdDateChooser.getDate();

if (pid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Somthing wrong Check Data Input Filed", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (grnid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Somthing wrong Check Data Input Filed", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String update = grnitemM.updategrnitem(grnitemid, pid, qty, buyprice, grnid);

JOptionPane.showMessageDialog(this, "Updated Details", "massege", JOptionPane.OK\_CANCEL\_OPTION);

grnItemIdFiled.setText("");

grnitemPIdFiled1.setText("");

grnitemQtyFiled.setText("");

grnitembuypFiled.setText("");

grnitemGrnIdFeild.setText("");

// grnMfdDateChooser.setDate(null);

grnitemM.loadtable("", grnitemtdm);

stockM.loadStocktb("", stocktdm);

}

}



private void jButton83ActionPerformed(java.awt.event.ActionEvent evt) {

String grnitemid = grnItemIdFiled.getText();

String pid = grnitemPIdFiled1.getText();

String qty = grnitemQtyFiled.getText();

if (pid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String delete = grnitemM.deletegrnitem(grnitemid, qty, pid);

JOptionPane.showMessageDialog(this, "Deleted Details", "massege", JOptionPane.QUESTION\_MESSAGE);

grnItemIdFiled.setText("");

grnitemPIdFiled1.setText("");

grnitemQtyFiled.setText("");

grnitembuypFiled.setText("");

grnitemGrnIdFeild.setText("");

grnitemM.loadtable("", grnitemtdm);

stockM.loadStocktb("", stocktdm);

}

}



private void jButton86ActionPerformed(java.awt.event.ActionEvent evt) {

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

}



private void jButton87ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

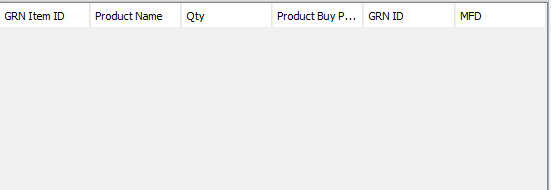


private void jButton80ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = grnitemSearchFiled.getText();

grnitemM.loadtable(keyword, grnitemtdm);

}



private void grnitemtableMouseClicked(java.awt.event.MouseEvent evt) {

int row = grnitemtable.getSelectedRow();

String grnitemid = (String) grnitemtdm.getValueAt(row, 0);

Object[] grnitemdata = grnitemM.getselectid(grnitemid);

grnItemIdFiled.setText(((String) grnitemdata[0]));

grnitemPIdFiled1.setText(((String) grnitemdata[1]));

grnitemQtyFiled.setText(((String) grnitemdata[2]));

grnitembuypFiled.setText(((String) grnitemdata[3]));

grnitemGrnIdFeild.setText(((String) grnitemdata[4]));

// grnMfdDateChooser.setDate((Date) grnitemdata[5]);

}



private void jButton74ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Source Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class grnItemModel {

static Logger log = Logger.getLogger(testlogin.class.getName());

Connection con;

public grnItemModel() {

con = db.dbConnection.getconnection();

}

public String insergrnitem(String pid,String qty, String buyprice, String grnid) {

String msg;

String getstockqty = "SELECT qty FROM pcgadguestshop.stocks WHERE product\_id\_product=?";

String updateqty = "UPDATE pcgadguestshop.stocks SET qty=? WHERE product\_id\_product=?";

String insertQuery ="INSERT INTO pcgadguestshop.grn\_item(qty, buyprice, product\_id\_product, grn\_id\_grn)VALUES(?,?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,qty);

ps.setString(2,buyprice);

ps.setString(3,pid);

ps.setString(4,grnid);

// ps.setDate(5,new java.sql.Date(grnMfd.getTime()));

ps.execute();

msg="Success";

PreparedStatement ps2 = con.prepareStatement(getstockqty);

ps2.setString(1, pid);

ResultSet rs2 = ps2.executeQuery();

while(rs2.next()){

int availableQty = rs2.getInt("qty");

int intqty = Integer.parseInt(qty);

int newqty = availableQty+intqty;

PreparedStatement ps3 = con.prepareStatement(updateqty);

ps3.setInt(1, newqty);

ps3.setString(2, pid);

ps3.execute();

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="error"+e.getMessage();

}

return msg;

}

public String updategrnitem(String grnitemid, String pid, String qty, String buyprice, String grnid) {

String msg;

int updatedqty;

int newupdatedqty;

String getstockqty = "SELECT qty FROM pcgadguestshop.stocks WHERE product\_id\_product=?";

String updateqty = "UPDATE pcgadguestshop.stocks SET qty=? WHERE product\_id\_product=?";

String grnitemqty = "SELECT qty FROM pcgadguestshop.grn\_item WHERE id\_grn\_item=?";

String updateQuary="UPDATE pcgadguestshop.grn\_item SET qty=?, buyprice=?, product\_id\_product=?, grn\_id\_grn=? WHERE id\_grn\_item=?";

try {

int newqty = Integer.parseInt(qty);

PreparedStatement ps1 = con.prepareStatement(grnitemqty);

ps1.setString(1, grnitemid);

ResultSet rs = ps1.executeQuery();

while(rs.next()){

int grnqty = rs.getInt("qty");

PreparedStatement ps2 = con.prepareStatement(getstockqty);

ps2.setString(1, pid);

ResultSet rs2 = ps2.executeQuery();

while(rs2.next()){

int stockqty =rs2.getInt("qty");

if(newqty>grnqty){

updatedqty = newqty - grnqty;

newupdatedqty = stockqty + updatedqty;

}else if(grnqty>newqty){

updatedqty = grnqty -newqty ;

newupdatedqty = stockqty - updatedqty;

}else{

newupdatedqty = 0 + stockqty;

}

PreparedStatement ps3 = con.prepareStatement(updateqty);

ps3.setInt(1, newupdatedqty);

ps3.setString(2, pid);

ps3.execute();

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1,qty);

ps.setString(2,buyprice);

ps.setString(3,pid);

ps.setString(4,grnid);

// ps.setDate(5,new java.sql.Date(grnMfd.getTime()));

ps.setString(5,grnitemid);

ps.execute();

msg="Success";

} }

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Error"+e.getMessage();

}

return null;

}

public String deletegrnitem(String grnitemid, String qty, String pid) {

String msg;

String getstockqty = "SELECT qty FROM pcgadguestshop.stocks WHERE product\_id\_product=?";

String updateqty = "UPDATE pcgadguestshop.stocks SET qty=? WHERE product\_id\_product=?";

String DeleteQueary ="DELETE FROM pcgadguestshop.grn\_item WHERE id\_grn\_item=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, grnitemid);

ps.execute();

msg="success";

PreparedStatement ps2 = con.prepareStatement(getstockqty);

ps2.setString(1, pid);

ResultSet rs2 = ps2.executeQuery();

while(rs2.next()){

int availableQty = rs2.getInt("qty");

int intqty = Integer.parseInt(qty);

int newqty = availableQty-intqty;

PreparedStatement ps3 = con.prepareStatement(updateqty);

ps3.setInt(1, newqty);

ps3.setString(2, pid);

ps3.execute();

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return null;

}

public void loadtable(String keyword, DefaultTableModel grnitemtdm) {

String loadtb = "SELECT grn\_item.id\_grn\_item,grn\_item.qty,grn\_item.buyprice,grn\_item.product\_id\_product,"

+ "grn\_item.grn\_id\_grn,product.id\_product,pcgadguestshop.product.productname FROM pcgadguestshop.product INNER JOIN pcgadguestshop.grn\_item ON "

+ "pcgadguestshop.grn\_item.product\_id\_product = pcgadguestshop.product.id\_product WHERE grn\_id\_grn like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

grnitemtdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_grn\_item");

String name = rs.getString("productname");

String qty = rs.getString("qty");

String buyprice = rs.getString("buyprice");

String grnid = rs.getString("grn\_id\_grn");

//Date mfdDate = rs.getDate("mfd");

rowData = new Object[]{id,name,qty,buyprice,grnid};

grnitemtdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselectid(String grnitemid) {

String getData ="SELECT \* FROM pcgadguestshop.grn\_item WHERE id\_grn\_item =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, grnitemid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String id = rs.getString("id\_grn\_item");

String pname = rs.getString("product\_id\_product");

String qty = rs.getString("qty");

String buyprice = rs.getString("buyprice");

String grnid = rs.getString("grn\_id\_grn");

//Date mfdDate = rs.getDate("mfd");

rowData = new Object[]{id,pname,qty,buyprice,grnid};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Item Page. Product view dialog page )** | | **GUI Number** : 12 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This subpage includes a good receive item page and stocks page in this system. Its help to get details about shop registered products and user do not go to the product page. User can see product id number, product name, sell price, brand name, accessory name in this page and user can search products by using product name. The result will show in product table and after search some products by user and if user want to see all registered products details again then user must clear the search field and click search button and user can see all registered products details in the table. Owner, manager and product controller can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | It use for search by entered keyword | Get keyword in input field and search using entered keyword | | jtable | View registered Product details | List down registered product details. search using product name and result will show inside jtable | | | |

**Source Code**

****

private void jButton64ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = proSearchFiled2.getText();

productM.loadtable(keyword, producttdm);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Item Page. GRN Details view dialog page )** | | **GUI Number** : 13 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This subpage includes in good receive items page. It helps to user when entered some good receive items when user must give good receive note id number to enter a good receive item to system. This subpage includes in front of good receive note id field and click GRN button when open a subpage and it shows all system entered good receive notes of all time and user can get details in this sup page and also user can search entered good receive note by using good receive id number after searched some id number after user can clear the search filed and click search button and again will show system have all good receive id numbers. Owner, manager and product controller can access this subpage in good receive item page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | It use for search by entered keyword | Get keyword in input field and search using entered keyword | | jtable | View registered GRN details | List down Insert GRN details. search using GRN ID and result will show inside jtable | | | |

**Source Code**

****

private void jButton79ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = grnSearchFiled2.getText();

grnM.loadtable(keyword,grntdm);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Employee post Page)** | | **GUI Number** : 14 |
| **Use case Name: Employee Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This is shop employees post registration page. It provides to add post of employee in the system and also user can add salary relevant their job position. if user successfully insert data and click insert button the result will show in employees post table and system will show information message to user after data successfully inserted. User can search employee post by using post name keyword and if he or she wants to update or delete some employee post name then first user must select data in employee post details table then selected data will show to user in input fields in page if user want to change some data user can change input filed data and click update button show information message to user also user can delete data row by click delete button. Only owner and manager can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert post details | When add new post note and details in jtable and also apply update and delete also search using keyword and show result inside jtable | | | |

**Source Code**

****

private void jButton35ActionPerformed(java.awt.event.ActionEvent evt) {

postidField.setText("");

postnameFiled.setText("");

postsalaryFiled.setText("");

postM.loadtablepost("", posttdm);

component();

}



private void jButton32ActionPerformed(java.awt.event.ActionEvent evt) {

String postId = postidField.getText();

String postname = postnameFiled.getText();

String salary = postsalaryFiled.getText();

if (postname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Post Name", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (salary.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Amount Of Salary", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertpost = postM.insertpost(postname, salary);

JOptionPane.showMessageDialog(this, insertpost, "massege", JOptionPane.INFORMATION\_MESSAGE);

postidField.setText("");

postnameFiled.setText("");

postsalaryFiled.setText("");

postM.loadtablepost("", posttdm);

component();

}

}



private void jButton33ActionPerformed(java.awt.event.ActionEvent evt) {

String postId = postidField.getText();

String postname = postnameFiled.getText();

String salary = postsalaryFiled.getText();

if (postname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From Table!", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String updatepost = postM.updatepost(postId, postname, salary);

JOptionPane.showMessageDialog(this, updatepost, "massege", JOptionPane.OK\_CANCEL\_OPTION);

postidField.setText("");

postnameFiled.setText("");

postsalaryFiled.setText("");

postM.loadtablepost("", posttdm);

component();

}

}



private void jButton34ActionPerformed(java.awt.event.ActionEvent evt) {

String postId = postidField.getText();

String postname = postnameFiled.getText();

if (postname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From Table!", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String deletepost = postM.deletepost(postId);

JOptionPane.showMessageDialog(this, deletepost, "massege", JOptionPane.QUESTION\_MESSAGE);

postidField.setText("");

postnameFiled.setText("");

postsalaryFiled.setText("");

postM.loadtablepost("", posttdm);

component();

}

}

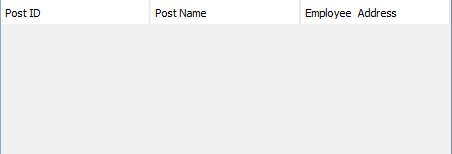


private void jButton43ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = postSearchfiled3.getText();

postM.loadtablepost(keyword, posttdm);

}



private void posttableMouseClicked(java.awt.event.MouseEvent evt) {

int row = posttable.getSelectedRow();

int postID = (int) posttdm.getValueAt(row, 0);

Object[] postdata = postM.getSelectId(postID);

//customernameFiled.setText(new String(customerdata[0]+""));

postidField.setText(String.valueOf(postdata[0]));

postnameFiled.setText((String) postdata[1]);

postsalaryFiled.setText((String) postdata[2]);

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class postModel {

static Logger log = Logger.getLogger(testlogin.class.getName());

Connection con;

public postModel() {

con = db.dbConnection.getconnection();

}

public String insertpost(String postname, String salary) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.posts(post, salary)VALUES(?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,postname);

ps.setString(2,salary);

ps.execute();

msg="Successfully Insert!";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Post Already Using!";

}

return msg;

}

public void loadtablepost(String keyword, DefaultTableModel posttdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.posts WHERE post like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

posttdm.setRowCount(0);

while (rs.next()) {

int id = rs.getInt("id\_post");

String postname = rs.getString("post");

String postsalary = rs.getString("salary");

rowData = new Object[]{id, postname, postsalary};

posttdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

}

}

public String deletepost(String postId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.posts WHERE id\_post=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, postId);

ps.execute();

msg="successfully Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing wrong check Input Fileds";

}

return msg;

}

public String updatepost(String postId, String postname, String salary) {

String msg;

String updateQuary="UPDATE pcgadguestshop.posts SET post=?, salary=? WHERE id\_post=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1, postname);

ps.setString(2, salary);

ps.setString(3, postId);

ps.execute();

msg="Successfully Updated!";

} catch (Exception e) {

e.printStackTrace();

msg = "This post Name Already Using!";

}

return msg;

}

public Object[] getSelectId(int postID) {

String getData ="SELECT \* FROM pcgadguestshop.posts WHERE id\_post =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setInt(1, postID);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String postid = rs.getString("id\_post");

String postname = rs.getString("post");

String postsalary = rs.getString("salary");

rowData = new Object[]{postID,postname,postsalary};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Employee registration Page )** | | **GUI Number** : 15 |
| **Use case Name: Employee Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   In this page shop employees can register under five requirements they are employee name, employee address, employee mobile number, employee National identity card number and relevant post name. In this system mainly user four main characters. They are owner, manager, cashier and product controller. They are main post in this shop and user can select relevant post name for employee by combo box. After correctly user entered data to input filed click insert button the data will show in employee table and user can search employees by their name. If user want update or deleted employee details first user must click data row what he wants to delete or update data in employee table after user can click update or deleted button if process have done and display information massage to user   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | ComboBox | List down supplier name | When insert or update grn and can get registered supplier name using combo box | | jtable | View insert Good Receive notes details | When add new Good Receive note and insert grn details in jtable and also apply update and delete also search using keyword and show result inside jtable | | jTabpane | View multiple interfaces in employee section | Show employee, attendance, payment and post views in employee section | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | When click post button and show registered post details and can search product using post name |
|  |  |  | Click report button wen open report Dialog box and can get report all insert post details |

**Source Code**

****

private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {

employeeIdFiled.setText("");

empnnameFiled.setText("");

empaddressFiled.setText("");

empmobileFiled.setText("");

employeeNicFiled.setText("");

empM.loadtableEmp("", emptdm); }



private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {

String empId = employeeIdFiled.getText();

String empname = empnnameFiled.getText();

String empnaddress = empaddressFiled.getText();

String empmodile = empmobileFiled.getText();

String empnic = employeeNicFiled.getText();

Object o = postCombo.getSelectedItem();

String empoost = String.valueOf(o);

if (empname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Name", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empnaddress.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Address", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empmodile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Mobile", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empnic.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee NIC", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertEmp = empM.insertemp(empname, empnaddress, empmodile, empoost,empnic);

JOptionPane.showMessageDialog(this, insertEmp, "massege", JOptionPane.INFORMATION\_MESSAGE);

employeeIdFiled.setText("");

empnnameFiled.setText("");

empaddressFiled.setText("");

empmobileFiled.setText("");

employeeNicFiled.setText("");

empM.loadtableEmp("", emptdm); }}



private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {

String empId = employeeIdFiled.getText();

String empname = empnnameFiled.getText();

String empnaddress = empaddressFiled.getText();

String empmodile = empmobileFiled.getText();

String empnic = employeeNicFiled.getText();

Object o = postCombo.getSelectedItem();

String empoost = String.valueOf(o);

if (empId.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select data from table", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Name", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empnaddress.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Address", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empmodile.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Mobile", "Massege", JOptionPane.WARNING\_MESSAGE);

} else if (empnic.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee NIC Number", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertEmp = empM.updateemp(empId, empname, empnaddress, empmodile, empoost,empnic);

JOptionPane.showMessageDialog(this, insertEmp, "massege", JOptionPane.OK\_CANCEL\_OPTION);

employeeIdFiled.setText("");

empnnameFiled.setText("");

empaddressFiled.setText("");

empmobileFiled.setText("");

employeeNicFiled.setText("");

empM.loadtableEmp("", emptdm); }

}



private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {

String empId = employeeIdFiled.getText();

String empname = empnnameFiled.getText();

if (empname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Plese Select data from table", "massege", JOptionPane.QUESTION\_MESSAGE);

} else {

String deleteEmp = empM.deleteemp(empId);

JOptionPane.showMessageDialog(this, deleteEmp, "massege", JOptionPane.QUESTION\_MESSAGE);

employeeIdFiled.setText("");

empnnameFiled.setText("");

empaddressFiled.setText("");

empmobileFiled.setText("");

empM.loadtableEmp("", emptdm);

}

}



private void jButton24ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = emprSearchfiled.getText();

empM.loadtableEmp(keyword, emptdm);

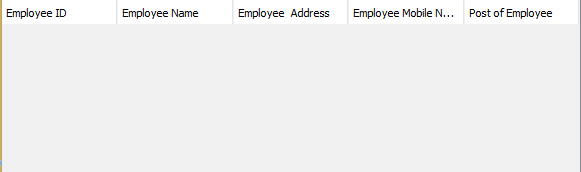
}



private void jButton71ActionPerformed(java.awt.event.ActionEvent evt) {

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

}



private void employeeTableMouseClicked(java.awt.event.MouseEvent evt) {

int row = employeeTable.getSelectedRow();

int empID = (int) emptdm.getValueAt(row, 0);

Object[] empdata = empM.getSelectId(empID);

//customernameFiled.setText(new String(customerdata[0]+""));

employeeIdFiled.setText(String.valueOf(empdata[0]));

empnnameFiled.setText((String) empdata[1]);

empaddressFiled.setText((String) empdata[2]);

empmobileFiled.setText((String) empdata[3]);

postCombo.setSelectedItem((String) empdata[4]);

}



private void jButton50ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class employeeModel {

static Logger log = Logger.getLogger(testlogin.class.getName());

Connection con;

public employeeModel() {

con=db.dbConnection.getconnection();

}

public String insertemp(String empname, String empnaddress, String empmodile, String empoost) {

String msg="";

String postid ="SELECT id\_post FROM pcgadguestshop.posts WHERE post=?";

String insertQuery ="INSERT INTO pcgadguestshop.emplyee(emp\_name, emp\_address, emp\_mobile, posts\_id\_post)VALUES(?,?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(postid);

ps.setString(1,empoost);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

int ID = rs.getInt("id\_post");

System.out.println(ID);

String postID = String.valueOf(ID);

PreparedStatement ps2 = con.prepareStatement(insertQuery);

ps2.setString(1, empname);

ps2.setString(2, empnaddress);

ps2.setString(3, empmodile);

ps2.setString(4, postID);

ps2.execute();

msg="Successfully Insert";

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Employee number Already taken";

}

return msg;

}

public String deleteemp(String empId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.emplyee WHERE idemplyee=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, empId);

ps.execute();

msg="Employee Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Cannot Delete Employee,Employee Id Already using";

}

return msg;

}

public String updateemp(String empId, String empname, String empnaddress, String empmodile, String empoost) {

String msg="";

String postid ="SELECT id\_post FROM pcgadguestshop.posts WHERE post=?";

String updateQuary="UPDATE pcgadguestshop.emplyee SET emp\_name=?, emp\_address=?, emp\_mobile=?, posts\_id\_post=? WHERE idemplyee=?";

try {

PreparedStatement ps = con.prepareStatement(postid);

ps.setString(1,empoost);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

int ID = rs.getInt("id\_post");

System.out.println(ID);

String postID = String.valueOf(ID);

PreparedStatement ps2 = con.prepareStatement(updateQuary);

ps2.setString(1, empname);

ps2.setString(2, empnaddress);

ps2.setString(3, empmodile);

ps2.setString(4, postID);

ps2.setString(5, empId);

ps2.execute();

msg="Successfully updated";

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Cannot Updated Mobile Numer already taken";

}

return msg;

}

public void loadtableEmp(String keyword, DefaultTableModel emptdm) {

String loadtable = "SELECT posts.post,emplyee.idemplyee,emplyee.emp\_name,emplyee.emp\_address,"

+ "emplyee.emp\_mobile,emplyee.posts\_id\_post,posts.id\_post FROM emplyee INNER JOIN posts ON emplyee.posts\_id\_post = posts.id\_post WHERE emp\_name like ? ";

try {

PreparedStatement ps = con.prepareStatement(loadtable);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

emptdm.setRowCount(0);

while (rs.next()) {

int empid = rs.getInt("idemplyee");

String empname = rs.getString("emp\_name");

String empaddre = rs.getString("emp\_address");

String empmobile = rs.getString("emp\_mobile");

String emppost = rs.getString("post");

rowData = new Object[]{empid,empname,empaddre,empmobile,emppost};

emptdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getSelectId(int empID) {

String getData ="SELECT \* FROM pcgadguestshop.emplyee WHERE idemplyee = ?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setInt(1, empID);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

int empiid = rs.getInt("idemplyee");

String empname = rs.getString("emp\_name");

String empaddress = rs.getString("emp\_address");

String empmobile = rs.getString("emp\_mobile");

String emppostid = rs.getString("posts\_id\_post");

String getpostname ="SELECT post FROM pcgadguestshop.posts WHERE id\_post=? ";

PreparedStatement ps2 = con.prepareStatement(getpostname);

ps2.setString(1, emppostid);

ResultSet rs2 =ps2.executeQuery();

while(rs2.next()){

String postname = rs2.getString("post");

System.out.println(postname);

emppostid=postname;

rowData = new Object[]{empiid,empname,empaddress,empmobile,emppostid};

return rowData;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Employee Page. post view dialog page )** | | **GUI Number** : 16 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Employee post subpage include in employee registration page and this page help to user when register a new employee in system and change some employee post and user can see salary of post name and user can open this subpage by clicking post button in employees registration page and user can search shop entered post name by using post name and result will load post details table. If user want to get back all registered post names details after search one post name first user must clear the search filed then click search button again and load all shop entered post name related their details. Owner and manager can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | It use for search by entered keyword | Get keyword in input field and search using entered keyword | | jtable | View registered Product details | List down registered product details. search using product name and result will show inside jtable | | | |

**Source Code**



private void jButton43ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = postSearchfiled4.getText();

postM.loadtablepost(keyword, posttdm);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Attendance Page )** | | **GUI Number** : 17 |
| **Use case Name: Employee Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Attendance page can record about employees attendance details and system will require three main data . They are employee registered id number, absent or present and current date. These main data must include to record data in system. After successfully insert data and system will displayed information message to user. And user can search attendance data using employee id number. If user wants to update or deleted entered attendance data user must click data row in attendance details table and relevant data will set data in to input fields after user can update or deleted details and process have done successfully and show information message to user   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | ComboBox | Can select employee absent or present | Combo box helps to select absent or present employee. | | jtable | View inserts attendance details | When add attendance details and insert attendance details in jtable and also apply update and delete also search using keyword and show result inside jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Click report button wen open report Dialog box and can get report about all attendance s |

**Source Code**

****

private void jButton28ActionPerformed(java.awt.event.ActionEvent evt) {

attendanceiDFiled.setText("");

attendanceEmpIdFiled.setText("");

searchdateChooser.setDate(null);

attendanceM.loadtablebyid("", attendancetdm);

} 

private void jButton25ActionPerformed(java.awt.event.ActionEvent evt) {

String attId = attendanceiDFiled.getText();

String attEmpId = attendanceEmpIdFiled.getText();

Date ada = searchdateChooser.getDate();

Object o = attendanceCombo.getSelectedItem();

String status = String.valueOf(o);

if (attEmpId.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee Id", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertatten = attendanceM.insertatten(attEmpId, ada, status);

JOptionPane.showMessageDialog(this, insertatten, "massege", JOptionPane.INFORMATION\_MESSAGE);

attendanceiDFiled.setText("");

attendanceEmpIdFiled.setText("");

searchdateChooser.setDate(null);

attendanceM.loadtablebyid("", attendancetdm);

}

}



private void jButton26ActionPerformed(java.awt.event.ActionEvent evt) {

String attId = attendanceiDFiled.getText();

String attEmpId = attendanceEmpIdFiled.getText();

Date ada = searchdateChooser.getDate();

System.out.println(ada);

Object o = attendanceCombo.getSelectedItem();

String status = String.valueOf(o);

if (attEmpId.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select data from table! ", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String updateatt = attendanceM.attenUpdate(attId, attEmpId, ada, status);

JOptionPane.showMessageDialog(this, updateatt, "massege", JOptionPane.OK\_CANCEL\_OPTION);

attendanceiDFiled.setText("");

attendanceEmpIdFiled.setText("");

searchdateChooser.setDate(null);

attendanceM.loadtablebyid("", attendancetdm);

}

}



private void jButton27ActionPerformed(java.awt.event.ActionEvent evt) {

String attId = attendanceiDFiled.getText();

String attEmpId = attendanceEmpIdFiled.getText();

if (attEmpId.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select data from table! ", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String deleteatt = attendanceM.attenDelete(attId);

JOptionPane.showMessageDialog(this, "Deleted Details", "massege", JOptionPane.QUESTION\_MESSAGE);

attendanceiDFiled.setText("");

attendanceEmpIdFiled.setText("");

searchdateChooser.setDate(null);

attendanceM.loadtablebyid("", attendancetdm);

}

}

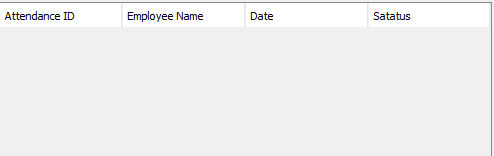


private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchAttendaceFiled.getText();

attendanceM.loadtablebyid(keyword, attendancetdm);

}



private void attendanceTableMouseClicked(java.awt.event.MouseEvent evt) {

int row = attendanceTable.getSelectedRow();

int attenID = (int) attendancetdm.getValueAt(row, 0);

Object[] attData = attendanceM.getSelectId(attenID);

attendanceiDFiled.setText(String.valueOf(attData[0]));

attendanceEmpIdFiled.setText((String) attData[1]);

searchdateChooser.setDate((Date) attData[2]);

attendanceCombo.setSelectedItem((String) attData[3]);

}



private void jButton54ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class attendanceModel {

static Logger log = Logger.getLogger(testlogin.class.getName());

Connection con;

public attendanceModel() {

con = db.dbConnection.getconnection();

}

public String insertatten(String attId, Date ada, String status) {

String msg;

String insert = "INSERT INTO pcgadguestshop.attendance (date, status, emplyee\_idemplyee)VALUES(?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(insert);

ps.setDate(1, new java.sql.Date(ada.getTime()));

ps.setString(2, status);

ps.setString(3, attId);

ps.execute();

msg = "Successfully insert!";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Somthing wrong check Input Fileds!";

}

return msg;

}

public void loadtablebyid(String keyword, DefaultTableModel attendancetdm) {

String search = "SELECT pcgadguestshop.emplyee.idemplyee,emplyee.emp\_name,attendance.id\_attendance,attendance.date,"

+ "attendance.status,attendance.emplyee\_idemplyee FROM pcgadguestshop.attendance "

+ "INNER JOIN pcgadguestshop.emplyee ON pcgadguestshop.emplyee.idemplyee = pcgadguestshop.attendance.emplyee\_idemplyee WHERE emplyee\_idemplyee like ?";

//String search="SELECT \* FROM pcgadguestshop.attendance WHERE id\_attendance like ?";

try {

PreparedStatement ps = con.prepareStatement(search);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

attendancetdm.setRowCount(0);

while (rs.next()) {

int attid = rs.getInt("id\_attendance");

String date = rs.getString("date");

String status = rs.getString("status");

String empName = rs.getString("emp\_name");

rowData = new Object[]{attid, empName, date, status};

attendancetdm.addRow(rowData);

}

} catch (Exception e) {

log.debug(e.getMessage());

e.printStackTrace();

}

}

public void loadtablebydate(Date date, DefaultTableModel attendancetdm) {

String search = "SELECT pcgadguestshop.emplyee.idemplyee,\n"

+ " pcgadguestshop.emplyee.emp\_name,\n"

+ " pcgadguestshop.attendance.id\_attendance,\n"

+ " pcgadguestshop.attendance.date,\n"

+ " pcgadguestshop.attendance.status,\n"

+ " pcgadguestshop.attendance.emplyee\_idemplyee\n"

+ "FROM pcgadguestshop.attendance\n"

+ " INNER JOIN pcgadguestshop.emplyee ON \n"

+ " pcgadguestshop.emplyee.idemplyee = pcgadguestshop.attendance.emplyee\_idemplyee WHERE date like ?";

try {

PreparedStatement ps = con.prepareStatement(search);

ps.setString(1, "%" + date + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

attendancetdm.setRowCount(0);

while (rs.next()) {

int attid = rs.getInt("id\_attendance");

String datee = rs.getString("date");

String status = rs.getString("status");

String empName = rs.getString("emp\_name");

System.out.println("date");

rowData = new Object[]{attid, empName, datee, status};

System.out.println("done 53");

attendancetdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getSelectId(int attenID) {

String getData ="SELECT \* FROM pcgadguestshop.attendance WHERE id\_attendance =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setInt(1, attenID);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

int attid = rs.getInt("id\_attendance");

// String datee = rs.getString("date");

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

String status = rs.getString("status");

String empid = rs.getString("emplyee\_idemplyee");

rowData = new Object[]{attid, empid, datee, status};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String attenUpdate(String attId, String attEmpId, Date ada, String status) {

String msg;

String updateQuary="UPDATE pcgadguestshop.attendance SET date=?, status=?, emplyee\_idemplyee=? WHERE id\_attendance=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setDate(1, new java.sql.Date(ada.getTime()));

ps.setString(2, status);

ps.setString(3, attEmpId);

ps.setString(4, attId);

ps.execute();

msg="Successfully Updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Somthing wrong check again!";

}

return msg;

}

public String attenDelete(String attId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.attendance WHERE id\_attendance=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, attId);

ps.execute();

msg="success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing wrong check again!";

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Payment Page )** | | **GUI Number** : 18 |
| **Use case Name: Employee Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Employee monthly salary will create in this page and system will required first employee registered id number after set employee id in employee id filed press enter button and load data about employee registered name and relevant salary about their post. If user want to change salary software give access to change amount and in page have two check box their ETF and EPF and user when select check box and it will calculate by employee salary amount. ETF takes 10% and EPF takes 3% present of employee salary. And shop will pay allowances to employee its will apply to employee net salary. Then click cal button and net salary will create after user can click insert button and details will show in payment details and user can search data by using employee id and user can update and delete payment details from system. And owner and manager can access this page   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View inserts payment details | When add payments details and insert attendance details in to the jtable and also apply update and delete and search using keyword and show result inside jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Click report button wen open report Dialog box and can get report about all payment details and single report about employee payment. |

**Source Code**

****

private void jButton47ActionPerformed(java.awt.event.ActionEvent evt) {

salempNameFild.setText("");

salgrosssalarayFiled.setText("");

salSararyFiled.setText("");

salEmpIDFiled.setText("");

paymentidFiled.setText("");

salETFFiled.setText("");

salEPFFiled.setText("");

salAllowanceFiled.setText("");

salNetSalaryFiled.setText("");

payM.loadtable("", paymenttdm); }



private void jButton44ActionPerformed(java.awt.event.ActionEvent evt) {

String payId = paymentidFiled.getText();

Date payDate = paymentDateChooser1.getDate();

String payempID = salEmpIDFiled.getText();

String etf = salETFFiled.getText();

String epf = salEPFFiled.getText();

String allowns = salAllowanceFiled.getText();

String netsalary = salNetSalaryFiled.getText();

if (payempID.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee ID! ", "Massege", JOptionPane.WARNING\_MESSAGE);

}else if(netsalary.isEmpty()){

JOptionPane.showMessageDialog(this, "Please Check Again ", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String insertpay = payM.insertpay(payDate, payempID,etf,epf,allowns,netsalary);

JOptionPane.showMessageDialog(this, insertpay, "massege", JOptionPane.INFORMATION\_MESSAGE);

salEmpIDFiled.setText("");

paymentidFiled.setText("");

salETFFiled.setText("");

salEPFFiled.setText("");

salAllowanceFiled.setText("");

salNetSalaryFiled.setText("");

payM.loadtable("", paymenttdm);

}

}



private void jButton45ActionPerformed(java.awt.event.ActionEvent evt) {

String payId = paymentidFiled.getText();

Date payDate = paymentDateChooser1.getDate();

String payempID = salEmpIDFiled.getText();

String etf = salETFFiled.getText();

String epf = salEPFFiled.getText();

String allowns = salAllowanceFiled.getText();

String netsalary = salNetSalaryFiled.getText();

if (payempID.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Employee ID! ", "Massege", JOptionPane.WARNING\_MESSAGE);

}else if(netsalary.isEmpty()){

JOptionPane.showMessageDialog(this, "Please Check Again ", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String update = payM.updatepay(payId,payDate, payempID,etf,epf,allowns,netsalary);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.OK\_CANCEL\_OPTION);

salEmpIDFiled.setText("");

paymentidFiled.setText("");

salETFFiled.setText("");

salEPFFiled.setText("");

salgrosssalarayFiled.setText("");

salAllowanceFiled.setText("");

salNetSalaryFiled.setText("");

payM.loadtable("", paymenttdm);

}

}



private void jButton46ActionPerformed(java.awt.event.ActionEvent evt) {

String payId = emppaymentIdFiled.getText();

String payempID = emppaymentEmpIdFiled.getText();

if (payempID.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From table! ", "Massege", JOptionPane.WARNING\_MESSAGE);

} else {

String delete = payM.deletepay(payId);

JOptionPane.showMessageDialog(this, delete, "massege", JOptionPane.QUESTION\_MESSAGE);

emppaymentIdFiled.setText("");

paymentDateChooser.setDate(null);

emppaymentEmpIdFiled.setText("");

payM.loadtable("", paymenttdm);

}

}



private void jButton47ActionPerformed(java.awt.event.ActionEvent evt) {

emppaymentIdFiled.setText("");

paymentDateChooser.setDate(null);

emppaymentEmpIdFiled.setText("");

payM.loadtable("", paymenttdm);

}



private void employeePaymentTableMouseClicked(java.awt.event.MouseEvent evt) {

int row = employeePaymentTable.getSelectedRow();

String paymeid = (String) paymenttdm.getValueAt(row, 0);

Object[] paydata = payM.getselectid(paymeid);

paymentidFiled.setText(String.valueOf(paydata[0]));

paymentDateChooser1.setDate((Date) paydata[1]);

salEmpIDFiled.setText((String) paydata[2]);

salETFFiled.setText((String) paydata[3]);

salEPFFiled.setText((String) paydata[4]);

salAllowanceFiled.setText((String) paydata[5]);

salNetSalaryFiled.setText((String) paydata[6]); }



private void jButton49ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class paymentEmployeeModel {

static Logger log = Logger.getLogger(paymentEmployeeModel.class.getName());

Connection con;

public paymentEmployeeModel() {

con=db.dbConnection.getconnection();

}

public String deletepay(String payId) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.salarypayment WHERE id\_salarypayment=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, payId);

ps.execute();

msg="Successfully Deleted!";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing wrong check Input Fileds!";

}

return msg;

}

public void loadtable(String keyword, DefaultTableModel paymenttdm) {

String loadtb = "SELECT pcgadguestshop.emplyee.idemplyee,\n" +

" pcgadguestshop.emplyee.emp\_name,\n" +

" pcgadguestshop.salarypayment.id\_salarypayment,\n" +

" pcgadguestshop.salarypayment.date,\n" +

" pcgadguestshop.salarypayment.emplyee\_idemplyee,\n" +

" pcgadguestshop.salarypayment.etf,\n" +

" pcgadguestshop.salarypayment.epf,\n" +

" pcgadguestshop.salarypayment.allowns,\n" +

" pcgadguestshop.salarypayment.netsalary\n" +

"FROM pcgadguestshop.salarypayment\n" +

" INNER JOIN pcgadguestshop.emplyee ON \n" +

" pcgadguestshop.salarypayment.emplyee\_idemplyee = pcgadguestshop.emplyee.idemplyee WHERE emplyee\_idemplyee like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

paymenttdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_salarypayment");

String pdate = rs.getString("date");

String payemid = rs.getString("emplyee\_idemplyee");

String etf = rs.getString("etf");

String epf = rs.getString("epf");

String allowns = rs.getString("allowns");

String netsalary = rs.getString("netsalary");

rowData = new Object[]{id,pdate,payemid,etf,epf,allowns,netsalary};

paymenttdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselectid(String paymeid) {

String getData ="SELECT \* FROM pcgadguestshop.salarypayment WHERE id\_salarypayment =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, paymeid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String id = rs.getString("id\_salarypayment");

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

String payemid = rs.getString("emplyee\_idemplyee");

String etf = rs.getString("etf");

String epf = rs.getString("epf");

String allowns = rs.getString("allowns");

String netsalary = rs.getString("netsalary");

rowData = new Object[]{id,pdate,payemid,etf,epf,allowns,netsalary};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public Object[] getempdata(String emp) {

String empdata ="SELECT \* FROM pcgadguestshop.emplyee WHERE idemplyee=?";

String emppostdata ="SELECT \* FROM pcgadguestshop.posts WHERE id\_post=?";

try {

PreparedStatement ps = con.prepareStatement(empdata);

ps.setString(1, emp);

ResultSet rs = ps.executeQuery();

Object[] rowData;

while(rs.next()){

String name = rs.getString("emp\_name");

String postid = rs.getString("posts\_id\_post");

PreparedStatement ps2 = con.prepareStatement(emppostdata);

ps2.setString(1, postid);

ResultSet rs2 = ps2.executeQuery();

while(rs2.next()){

String salaray = rs2.getString("salary");

rowData = new Object[]{name,salaray};

return rowData;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String insertpay(Date payDate, String payempID, String etf, String epf, String allowns, String netsalary) {

String msg;

String insert = "INSERT INTO pcgadguestshop.salarypayment(date, emplyee\_idemplyee, etf, epf,allowns,netsalary)VALUES(?,?,?,?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(insert);

ps.setDate(1, new java.sql.Date(payDate.getTime()));

ps.setString(2, payempID);

ps.setString(3, etf);

ps.setString(4, epf);

ps.setString(5, allowns);

ps.setString(6, netsalary);

ps.execute();

msg = "Successfully insert!";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Somthing wrong check Input Fileds!";

}

return msg;

}

public String updatepay(String payId, Date payDate, String payempID, String etf, String epf, String allowns, String netsalary) {

String msg="";

String updateQuary="UPDATE pcgadguestshop.salarypayment SET date=?, emplyee\_idemplyee=?, etf=?, epf=?, allowns=?, netsalary=? WHERE id\_salarypayment=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setDate(1, new java.sql.Date(payDate.getTime()));

ps.setString(2, payempID);

ps.setString(3, etf);

ps.setString(4, epf);

ps.setString(5, allowns);

ps.setString(6, netsalary);

ps.setString(7, payId);

ps.execute();

msg="Successfully updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Somthing wrong check Input Fileds";

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee Section (Registered employee dialog page )** | | **GUI Number** : 19 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   When user entered attendance details about employees. This employee subpage helps to user. This registered employee subpage includes an employee attendance page. This employee page will open when click employee button user can see details about all registered to the shop system and user can search registered employees by using their name and search results will show to user in employee details table. When user search some employee detail using their name if user wants to all registered employees’ detail first clears the search input filed and click search button and again will load all details about shop registered. Owner and manager can access this employee subpage.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components esily | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | It use for search by entered keyword | Get keyword in input field and search using entered keyword | | jtable | View registered empoyee details .and its easily can get employee id | List down registered employee details. search using product name and result will show inside the jtable | | | |

Source Code



private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchfiled2.getText();

empM.loadtableEmp(keyword, emptdm);

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Product Section (Brand Registered Page)** | | **GUI Number** : 20 |
| **Use case Name: Product Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Brand register page include in product section and main purpose of separate this brand register page because every product has band name so when user add product to system definitely user must add brand so it will take more storage. If user want to add brand in system and he have to add only brand name in brand text field after the input result shows in right side table and user can search data using brand keyword and if he wants to update or delete first user have select data row in brand details table after user can update or delete added brand name form system.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields to user | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert registered brand details | When add new brand details and insert brand details in to the jtable and also apply update and delete also search using keyword and show result inside jtable | | | |

**Source Code**

****

private void jButton58ActionPerformed(java.awt.event.ActionEvent evt) {

categoryIdFiled.setText("");

categoryNameFiled.setText("");

categoryM.loadtable("", categorytdm);

component();

}



private void jButton55ActionPerformed(java.awt.event.ActionEvent evt) {

String id = categoryIdFiled.getText();

String name = categoryNameFiled.getText();

if (name.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Product Name", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insertcategory = categoryM.insertcate(name);

JOptionPane.showMessageDialog(this, insertcategory, "massege", JOptionPane.INFORMATION\_MESSAGE);

categoryIdFiled.setText("");

categoryNameFiled.setText("");

categoryM.loadtable("", categorytdm);

component();

}

}



private void jButton56ActionPerformed(java.awt.event.ActionEvent evt) {

String id = categoryIdFiled.getText();

String name = categoryNameFiled.getText();

if (name.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Feilds", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String updateCategoty = categoryM.updateCate(id, name);

JOptionPane.showMessageDialog(this, updateCategoty, "massege", JOptionPane.OK\_CANCEL\_OPTION);

categoryIdFiled.setText("");

categoryNameFiled.setText("");

categoryM.loadtable("", categorytdm);

component();

}

}



private void jButton57ActionPerformed(java.awt.event.ActionEvent evt) {

String id = categoryIdFiled.getText();

String name = categoryNameFiled.getText();

if (name.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data From Table", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String deleteCategoty = categoryM.deletecate(id);

JOptionPane.showMessageDialog(this, deleteCategoty, "massege", JOptionPane.QUESTION\_MESSAGE);

categoryIdFiled.setText("");

categoryNameFiled.setText("");

categoryM.loadtable("", categorytdm);

component();

}

}

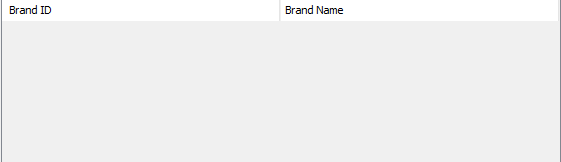


private void jButton59ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = categorySearchFiled.getText();

categoryM.loadtable(keyword, categorytdm);

}



private void categorytableMouseClicked(java.awt.event.MouseEvent evt) {

int row = categorytable.getSelectedRow();

String cid = (String) categorytdm.getValueAt(row, 0);

Object[] categorydata = categoryM.getselectid(cid);

categoryIdFiled.setText(String.valueOf(categorydata[0]));

categoryNameFiled.setText((String) categorydata[1]);

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class pCategoryModel {

static Logger log = Logger.getLogger(pCategoryModel.class.getName());

Connection con;

public pCategoryModel() {

con = db.dbConnection.getconnection();

}

public String updateCate(String id, String name) {

String msg;

String updateQuary="UPDATE pcgadguestshop.category SET categor\_name=? WHERE id\_category=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1, name);

ps.setString(2, id);

ps.execute();

msg="Successfully Updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Cannot Update Category Name Alredy exsist";

}

return msg;

}

public String deletecate(String id) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.category WHERE id\_category=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, id);

ps.execute();

msg="successfully Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing Wrong Plase Chack Again";

}

return msg;

}

public void loadtable(String keyword, DefaultTableModel categorytdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.category WHERE categor\_name like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

categorytdm.setRowCount(0);

while (rs.next()) {

String idc = rs.getString("id\_category");

String catName = rs.getString("categor\_name");

rowData = new Object[]{idc,catName};

categorytdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public String insertcate(String name) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.category(categor\_name)VALUES(?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,name);

ps.execute();

msg="Successfully Updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Cannot Insert Category Name Alredy exsist";

}

return msg;

}

public Object[] getselectid(String cid) {

String getData ="SELECT \* FROM pcgadguestshop.category WHERE id\_category =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, cid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String idcat = rs.getString("id\_category");

String catename = rs.getString("categor\_name");

rowData = new Object[]{idcat,catename};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Product Section (Accessory Registered Page)** | | **GUI Number** : 21 |
| **Use case Name: Product Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   In computer shop have different type of computer accessories and its do not change by brand name (ex. Ram, monitor, vga, hard disk, power supply and etc.) accessories can register by adding accessory name in accessory name field. User can search accessory using accessory name. And if added new name the input data will shows in accessories table and user can update or delete added details user must click data row what user want delete or update data after selected data will shows in Relevant data input fields and after user can want user wants.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields to user | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View insert registered brand details | When add new brand details and insert brand details in to the jtable and also apply update and delete also search using keyword and show result inside jtable | | | |

**Source Code**

****

private void jButton97ActionPerformed(java.awt.event.ActionEvent evt) {

accessoryIdFiled.setText("");

accessoryNameFiled.setText("");

accessotyM.loadtable("", accessorytdm);

component();

}



private void jButton94ActionPerformed(java.awt.event.ActionEvent evt) {

String getAccid = accessoryIdFiled.getText();

String getAccname = accessoryNameFiled.getText();

if (getAccname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Insert Accrssory Name!", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insert = accessotyM.insertAcc(getAccname);

JOptionPane.showMessageDialog(this, insert, "massege", JOptionPane.INFORMATION\_MESSAGE);

accessoryIdFiled.setText("");

accessoryNameFiled.setText("");

accessotyM.loadtable("", accessorytdm);

component();

}

}



private void jButton95ActionPerformed(java.awt.event.ActionEvent evt) {

String vaccidd = accessoryIdFiled.getText();

String vaccnamee = accessoryNameFiled.getText();

System.out.println(vaccidd);

System.out.println(vaccnamee);

if (vaccnamee.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check input fileds are empty!", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String update = accessotyM.updateacc(vaccidd, vaccnamee);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.INFORMATION\_MESSAGE);

accessoryIdFiled.setText("");

accessoryNameFiled.setText("");

accessotyM.loadtable("", accessorytdm);

component();

}

}



private void jButton96ActionPerformed(java.awt.event.ActionEvent evt) {

String getAccid = accessoryIdFiled.getText();

String getAccname = accessoryNameFiled.getText();

if (getAccname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Select Data from table!", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String delete = accessotyM.deleteAcc(getAccid);

JOptionPane.showMessageDialog(this, delete, "massege", JOptionPane.INFORMATION\_MESSAGE);

accessoryIdFiled.setText("");

accessoryNameFiled.setText("");

accessotyM.loadtable("", accessorytdm);

component();

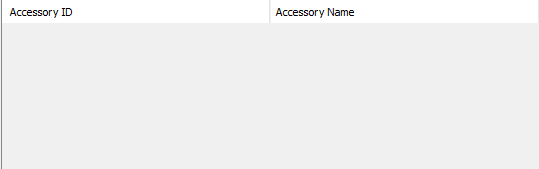
}

}



String keyword = accessorySearchFiled.getText();

accessotyM.loadtable(keyword, accessorytdm);



private void accessoryTableMouseClicked(java.awt.event.MouseEvent evt) {

int row = accessoryTable.getSelectedRow();

int id = (int) accessorytdm.getValueAt(row, 0);

Object[] accdata = accessotyM.getselectid(id);

accessoryIdFiled.setText((String) accdata[0]);

accessoryNameFiled.setText((String) accdata[1]);

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class accessoryModel {

static Logger log = Logger.getLogger(accessoryModel.class.getName());

Connection con;

public accessoryModel() {

con=db.dbConnection.getconnection();

}

public String insertAcc(String accname) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.pcaccerserise(name)VALUES(?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,accname);

ps.execute();

msg="Successfully Insert!";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Accessory part Already Using!";

}

return msg;

}

public String deleteAcc(String accid) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.pcaccerserise WHERE id\_pcaccerserise=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, accid);

ps.execute();

msg="successfully Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing wrong check Input Fileds";

}

return msg;

}

public void loadtable(String keyword, DefaultTableModel accessorytdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.pcaccerserise WHERE name like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

accessorytdm.setRowCount(0);

while (rs.next()) {

int id = rs.getInt("id\_pcaccerserise");

String name = rs.getString("name");

rowData = new Object[]{id,name};

accessorytdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselectid(int id) {

String getData ="SELECT \* FROM pcgadguestshop.pcaccerserise WHERE id\_pcaccerserise =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setInt(1, id);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String idacc = rs.getString("id\_pcaccerserise");

String name = rs.getString("name");

rowData = new Object[]{idacc,name};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String updateacc(String vaccidd, String vaccnamee) {

String msg="";

String update = "UPDATE pcgadguestshop.pcaccerserise SET name=? WHERE id\_pcaccerserise=?";

try {

PreparedStatement ps = con.prepareStatement(update);

ps.setString(1, vaccnamee);

ps.setString(2, vaccidd);

ps.execute();

msg="Scussessfully Updated!";

} catch (Exception e) {

log.debug(e.getMessage());

e.printStackTrace();

msg="Cannot updated Accessoty name already taken!";

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Product Section (Stock Page )** | | **GUI Number** : 22 |
| **Use case Name: Stocks Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**     Stock’s page will helps to see available quantity and required product items in shop. First user must register their selling product in this page. To register a stock product user have enter product id (user can get registered product id by click product button and user can search product using required product name) and quantity if first time user registers a stocks product user can set quantity as a 0 value. Also, user search stocks products. User can update and delete stocks 3products after select data row in stocks details table. And stocks product quantity will update when get product by suppliers and provide invoice to customer. Owner, manager and product controller can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | jtable | View inserts Stocks details | When add stocks details in to the jtable show in table and also apply update and delete and search using keyword and show result inside jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Click report button wen open report Dialog box and can get report about all Stocks details. |
|  |  |  | User can get product id in this dialog box and user can search products by using product name |

**Source Code**

****

private void jButton91ActionPerformed(java.awt.event.ActionEvent evt) {

stocksIdFiled.setText("");

StocksPIdFiled.setText("");

StocksQtyFiled1.setText("");

// stocksMfdChooser.setDate(null);

stockM.loadStocktb("", stocktdm);

}



private void jButton88ActionPerformed(java.awt.event.ActionEvent evt) {

String stid = stocksIdFiled.getText();

String stpid = StocksPIdFiled.getText();

String stqty = StocksQtyFiled1.getText();

if (stpid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Product ID", "massege", JOptionPane.QUESTION\_MESSAGE);

} else if (stqty.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Qty", "massege", JOptionPane.QUESTION\_MESSAGE);

} else {

String inter = stockM.insertStock(stpid, stqty);

JOptionPane.showMessageDialog(this, inter, "massege", JOptionPane.QUESTION\_MESSAGE);

stocksIdFiled.setText("");

StocksPIdFiled.setText("");

StocksQtyFiled1.setText("");

// stocksMfdChooser.setDate(null);

stockM.loadStocktb("", stocktdm);

}

}



private void jButton89ActionPerformed(java.awt.event.ActionEvent evt) {

String stid = stocksIdFiled.getText();

String stpid = StocksPIdFiled.getText();

String stqty = StocksQtyFiled1.getText();

if (stid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Cannot Update Please Check input Fileds!", "massege", JOptionPane.QUESTION\_MESSAGE);

} else if (stqty.isEmpty()) {

JOptionPane.showMessageDialog(this, "Cannot Update Please Check input Fileds!", "massege", JOptionPane.QUESTION\_MESSAGE);

} else {

String update = stockM.updatetStock(stid, stpid, stqty);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.QUESTION\_MESSAGE);

stocksIdFiled.setText("");

StocksPIdFiled.setText("");

StocksQtyFiled1.setText("");

stockM.loadStocktb("", stocktdm);

}

}



private void jButton90ActionPerformed(java.awt.event.ActionEvent evt) {

String stid = stocksIdFiled.getText();

String stpid = StocksPIdFiled.getText();

String stqty = StocksQtyFiled1.getText();

if (stid.isEmpty()) {

JOptionPane.showMessageDialog(this, "Select Data from table", "massege", JOptionPane.QUESTION\_MESSAGE);

} else if (stqty.isEmpty()) {

JOptionPane.showMessageDialog(this, "Select Data from table", "massege", JOptionPane.QUESTION\_MESSAGE);

} else {

String delete = stockM.deleteStock(stid);

JOptionPane.showMessageDialog(this, delete, "massege", JOptionPane.QUESTION\_MESSAGE);

stocksIdFiled.setText("");

StocksPIdFiled.setText("");

StocksQtyFiled1.setText("");

stockM.loadStocktb("", stocktdm);

}

}

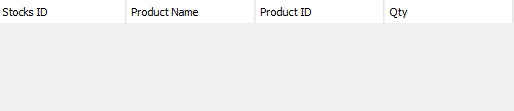


private void jButton92ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = stocksSearchFiled.getText();

stockM.loadStocktb(keyword, stocktdm);

}



private void stokstableMouseClicked(java.awt.event.MouseEvent evt) {

int row = stokstable.getSelectedRow();

String id = (String) stocktdm.getValueAt(row, 0);

Object[] stockdata = stockM.getselecedid(id);

stocksIdFiled.setText(((String) stockdata[0]));

StocksPIdFiled.setText(((String) stockdata[1]));

StocksQtyFiled1.setText(((String) stockdata[2]));

// stocksMfdChooser.setDate((Date) stockdata[3]);

}



private void jButton53ActionPerformed(java.awt.event.ActionEvent evt) {

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

}



private void jButton93ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class stocksModel {

static Logger log = Logger.getLogger(stocksModel.class.getName());

Connection con;

public stocksModel() {

con = db.dbConnection.getconnection();

}

public String updatetStock(String stid, String stpid, String stqty) {

String msg;

String updateQuary="UPDATE pcgadguestshop.stocks SET qty=?, product\_id\_product=? WHERE id\_stocks=?";

try {

PreparedStatement ps = con.prepareStatement(updateQuary);

ps.setString(1,stqty);

ps.setString(2,stpid);

// ps.setDate(3, new java.sql.Date(stdate.getTime()));

ps.setString(3,stid);

ps.execute();

msg="Successfully Updated";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg = "Product ID Aready taken!";

}

return msg;

}

public String deleteStock(String stid) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.stocks WHERE id\_Stocks=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, stid);

ps.execute();

msg="success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Error"+e.getMessage();

}

return null;

}

public void loadStocktb(String keyword, DefaultTableModel stocktdm) {

String loadtb = "SELECT stocks.`id\_stocks`,"

+ "stocks.qty,stocks.product\_id\_product, product.id\_product,product.productname FROM pcgadguestshop.stocks INNER JOIN pcgadguestshop.product ON pcgadguestshop.stocks.product\_id\_product = pcgadguestshop.product.id\_product WHERE product\_id\_product like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

stocktdm.setRowCount(0);

while (rs.next()) {

String sid = rs.getString("id\_stocks");

String qty = rs.getString("qty");

String pid = rs.getString("product\_id\_product");

// Date date = rs.getDate("mfd");

String pname = rs.getString("productname");

rowData = new Object[]{sid,pname,pid,qty};

stocktdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselecedid(String id) {

String getData ="SELECT \* FROM pcgadguestshop.stocks WHERE id\_stocks =?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, id);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String sid = rs.getString("id\_stocks");

String qty = rs.getString("qty");

String pid = rs.getString("product\_id\_product");

//Date date = rs.getDate("mfd");

rowData = new Object[]{sid,pid,qty};

}

return rowData;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String insertStock(String stpid, String stqty) {

String msg="";

String insert = "INSERT INTO pcgadguestshop.stocks (qty, product\_id\_product)VALUES(?,?)";

try {

PreparedStatement ps = con.prepareStatement(insert);

ps.setString(1, stqty);

ps.setString(2, stpid);

ps.execute();

msg="Successfully Insert";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Somthing wrong plase check input Data";

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (Product Registration Page )** | | **GUI Number** : 23 |
| **Use case Name: Product Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**     In this page give access to user register products in shop and require four main details. There is product name, product sell price, product brand name and product accessory name. Product brand name and accessory name can select in combo box. Its only view shop registered brand names and accessories name. These above requirements fill user can register a product in system and input data will show product details table .user can search product details user product name and user can update and delete input data after select row in table. After system will show massage to user the process has done. Owner, manager and product controller can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete and Search user | When add data to input filed after click specified button and call their command | | ComboBox | List down product category and accessory name | When insert or update grn and can get insert category name and accessory name by combo box | | jtable | View insert product details | When add product and show data in jtable and also apply update and delete also search using keyword and show result inside the jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Click report button wen open report Dialog box and can get report about all product details |

**Source Code**

****

private void jButton63ActionPerformed(java.awt.event.ActionEvent evt) {

home h = new home();

h.component();

prodIdFiled.setText("");

proNameFiled.setText("");

// proQtyFied.setText("");

prosellFiled.setText("");

productM.loadtable("", producttdm);

String category = "SELECT \* FROM pcgadguestshop.category";

try {

PreparedStatement ps = con.prepareStatement(category);

ResultSet rs = ps.executeQuery();

Vector v = new Vector();

while (rs.next()) {

v.add(rs.getString("categor\_name"));

}

DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);

procategoryComboo.setModel(dcm);

} catch (Exception e) {

e.printStackTrace();

}

}



private void jButton60ActionPerformed(java.awt.event.ActionEvent evt) {

String pid = prodIdFiled.getText();

String pname = proNameFiled.getText();

//String pqty = proQtyFied.getText();

String psellprice = prosellFiled.getText();

Object o = procategoryComboo.getSelectedItem();

String pcategory = String.valueOf(o);

Object o2 = partSelectCombo.getSelectedItem();

String pacc = String.valueOf(o2);

System.out.println(pacc);

if (pname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Product Name", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (psellprice.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter Product Sell Price", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String insertp = productM.insertpro(pname, psellprice, pcategory, pacc);

JOptionPane.showMessageDialog(this, insertp, "massege", JOptionPane.INFORMATION\_MESSAGE);

prodIdFiled.setText("");

proNameFiled.setText("");

// proQtyFied.setText("");

prosellFiled.setText("");

productM.loadtable("", producttdm);

}

}



private void jButton61ActionPerformed(java.awt.event.ActionEvent evt) {

String pid = prodIdFiled.getText();

String pname = proNameFiled.getText();

// String pqty = proQtyFied.getText();

String psellprice = prosellFiled.getText();

Object o = procategoryComboo.getSelectedItem();

String pcategory = String.valueOf(o);

Object o2 = partSelectCombo.getSelectedItem();

String pacc = String.valueOf(o2);

if (pname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (psellprice.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String update = productM.updatepro(pid, pname, psellprice, pcategory, pacc);

JOptionPane.showMessageDialog(this, update, "massege", JOptionPane.OK\_CANCEL\_OPTION);

prodIdFiled.setText("");

proNameFiled.setText("");

// proQtyFied.setText("");

prosellFiled.setText("");

productM.loadtable("", producttdm);

}

}



private void jButton62ActionPerformed(java.awt.event.ActionEvent evt) {

String pid = prodIdFiled.getText();

String pname = proNameFiled.getText();

// String pqty = proQtyFied.getText();

String psellprice = prosellFiled.getText();

if (pname.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else if (psellprice.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Check Input Fileds?", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String delete = productM.deletepro(pid);

JOptionPane.showMessageDialog(this, "Deleted Details", "massege", JOptionPane.QUESTION\_MESSAGE);

prodIdFiled.setText("");

proNameFiled.setText("");

// proQtyFied.setText("");

prosellFiled.setText("");

productM.loadtable("", producttdm);

}

}

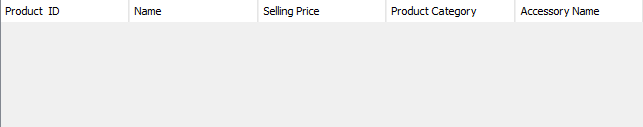


private void jButton64ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = proSearchFiled.getText();

productM.loadtable(keyword, producttdm);

}



private void producttableMouseClicked(java.awt.event.MouseEvent evt) {

int row = producttable.getSelectedRow();

String proid = (String) producttdm.getValueAt(row, 0);

Object[] productdata = productM.getselectid(proid);

prodIdFiled.setText(String.valueOf(productdata[0]));

proNameFiled.setText((String) productdata[1]);

prosellFiled.setText((String) productdata[2]);

procategoryComboo.setSelectedItem((String) productdata[3]);

partSelectCombo.setSelectedItem((String) productdata[4]);

}



private void jButton52ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

/\*\*

\*

\* @author Acer

\*/

public class productModel {

static Logger log = Logger.getLogger(productModel.class.getName());

Connection con;

public productModel() {

con=db.dbConnection.getconnection();

}

public String insertpro(String pname,String psellprice,String pcategory,String pacc) {

String msg="";

String getcatid ="SELECT \* FROM pcgadguestshop.category WHERE categor\_name =?";

String getaccessoryid ="SELECT \* FROM pcgadguestshop.pcaccerserise WHERE name =?";

String insertQuery ="INSERT INTO pcgadguestshop.product(productname, sell\_price, category\_id\_category, pcaccerserise\_id\_pcaccerserise)VALUES(?,?,?,?)";

try {

PreparedStatement ps1=con.prepareStatement(getcatid);

ps1.setString(1, pcategory);

ResultSet rs = ps1.executeQuery();

while(rs.next()){

int categoryid = rs.getInt("id\_category");

System.out.println(categoryid);

PreparedStatement ps3 = con.prepareStatement(getaccessoryid);

ps3.setString(1, pacc);

System.out.println("accse"+pacc);

ResultSet r2 =ps3.executeQuery();

while(r2.next()){

int accid = r2.getInt("id\_pcaccerserise");

System.out.println(accid);

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,pname);

ps.setString(2,psellprice);

ps.setInt(3, categoryid);

ps.setInt(4, accid);

ps.execute();

msg="Successfully Insert";

}

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Something Wrong Please Chack Input Data";

}

return msg;

}

public String updatepro(String pid, String pname,String psellprice,String pcategory, String pacc) {

String msg="";

String getcatid ="SELECT \* FROM pcgadguestshop.category WHERE categor\_name =?";

String getaccessoryid ="SELECT \* FROM pcgadguestshop.pcaccerserise WHERE name =?";

String updateQuery ="UPDATE pcgadguestshop.product SET productname=?, sell\_price=?, category\_id\_category=?, pcaccerserise\_id\_pcaccerserise=? WHERE id\_product=?";

try {

PreparedStatement ps1=con.prepareStatement(getcatid);

ps1.setString(1, pcategory);

ResultSet rs = ps1.executeQuery();

while(rs.next()){

int categoryid = rs.getInt("id\_category");

PreparedStatement ps3 = con.prepareStatement(getaccessoryid);

ps3.setString(1, pacc);

ResultSet r2 =ps3.executeQuery();

while(r2.next()){

int accid = r2.getInt("id\_pcaccerserise");

System.out.println(accid);

PreparedStatement ps = con.prepareStatement(updateQuery);

ps.setString(1,pname);

ps.setString(2,psellprice);

ps.setInt(3, categoryid);

ps.setInt(4, accid);

ps.setString(5, pid);

ps.execute();

msg="Successfully Updated";

}

}

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Something Wrong Please Chack Input Data";

}

return msg;

}

public String deletepro(String pid) {

String msg;

String DeleteQueary ="DELETE FROM pcgadguestshop.product WHERE id\_product=?";

try {

PreparedStatement ps = con.prepareStatement(DeleteQueary);

ps.setString(1, pid);

ps.execute();

msg="successfully Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Something Wrong Please Chack Input Data";

}

return msg;

}

public void loadtable(String keyword, DefaultTableModel producttdm) {

String loadtb = "SELECT pcgadguestshop.product.id\_product,product.productname,product.sell\_price,product.category\_id\_category,"

+ "product.pcaccerserise\_id\_pcaccerserise,category.id\_category,category.categor\_name,"

+ "pcaccerserise.id\_pcaccerserise,pcaccerserise.name FROM pcgadguestshop.product INNER JOIN pcgadguestshop.pcaccerserise ON pcgadguestshop.product.pcaccerserise\_id\_pcaccerserise = pcgadguestshop.pcaccerserise.id\_pcaccerserise "

+ "INNER JOIN pcgadguestshop.category ON "

+ " pcgadguestshop.category.id\_category = pcgadguestshop.product.category\_id\_category WHERE productname like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

producttdm.setRowCount(0);

while (rs.next()) {

String productid = rs.getString("id\_product");

String name = rs.getString("productname");

String price = rs.getString("sell\_price");

String catid = rs.getString("categor\_name");

String accid = rs.getString("name");

rowData = new Object[]{productid,name,price,catid,accid};

producttdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public Object[] getselectid(String proid) {

String getData ="SELECT \* FROM pcgadguestshop.product WHERE id\_product = ?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, proid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

int prodid = rs.getInt("id\_product");

String proname = rs.getString("productname");

String productsell = rs.getString("sell\_price");

String cid = rs.getString("category\_id\_category");

String accid = rs.getString("pcaccerserise\_id\_pcaccerserise");

String getcatname ="SELECT categor\_name FROM pcgadguestshop.category WHERE id\_category=? ";

PreparedStatement ps2 = con.prepareStatement(getcatname);

ps2.setString(1, cid);

ResultSet rs2 =ps2.executeQuery();

while(rs2.next()){

String cname = rs2.getString("categor\_name");

cid=cname;

String getaccname ="SELECT name FROM pcgadguestshop.pcaccerserise WHERE id\_pcaccerserise=? ";

PreparedStatement ps3 = con.prepareStatement(getaccname);

ps3.setString(1, accid);

ResultSet rs3 =ps3.executeQuery();

while(rs3.next()){

String accname = rs3.getString("name");

accid=accname;

rowData = new Object[]{prodid,proname,productsell,cid,accid};

return rowData;

}

}

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Cashier Section (payment Page )** | | **GUI Number** : 24 |
| **Use case Name: customer payment process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This page can access only owner, Manager and cashier. First user must enter product id name on product id data filed and press enter. the relevant data of entered product id will set to fields they are product name and sell price. And user can set burying product quantity value on qty field and quantity and sell price will calculate and total amount of product will show in total field. And click add button the entered product will add to product table and user can remove or updates added product data. And show to user total amount of buying product by customer and user can entered payment value and also user can register a new customer in this page and user can add buying customer name to invoice. If did not customer want to register in this shop and print customer name field as a guest and click bill button process have dan show massage to user and invoice will display to user.     1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert ,update, delete)after show message above activity | | jButton | It use for Inert, update, delete, get invoice and Search | When add data to input filed after click specified button and call their command | | jtable | View insert product details  , invoice items and registered customers | Shows added product details and view registered customers in table, when update or remove products also update jtable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Cashier insert customer with going to customer page |

**Source Code**



private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {

totalFiled.setText("");

paymentFiled.setText("");

balanceFiled.setText("");

invoicedate.setDate(null);

cuMobileFiled.setText("Guest");

porodictid.setText("");

pnamefiled.setText("");

priceFiled.setText("");

amountFiled.setText("");

customerM.loadtableAdmin("", customer2tdm);

invoiceItemM.loadtableitem("", invoiceItemtdm);

invoiceM.loadtableInvoice("", invoicedatatdm);

}

**(Customer)**

private void jButton36ActionPerformed(java.awt.event.ActionEvent evt) {

customerM.loadtableAdmin("", customer2tdm);

}



private void jButton31ActionPerformed(java.awt.event.ActionEvent evt) {

invoiceProducttdm.addRow(new Object[]{

porodictid.getText(), pnamefiled.getText(), qty2Filed.getText(), priceFiled.getText(), amountFiled.getText()

});

double sum = 0;

for (double i = 0; i < invoicebuytable.getRowCount(); i++) {

sum = sum + Double.parseDouble(invoicebuytable.getValueAt((int) i, 4).toString());

}

totalFiled.setText(Double.toString(sum));

porodictid.setText("");

pnamefiled.setText("");

priceFiled.setText("");

amountFiled.setText("");

qty2Filed.setText("0");

porodictid.requestFocus();

}



private void jButton39ActionPerformed(java.awt.event.ActionEvent evt) {

int i = invoicebuytable.getSelectedRow();

if(i>=0){

invoicebuyitemtdm.setValueAt(porodictid.getText(), i, 0);

invoicebuyitemtdm.setValueAt(qty2Filed.getText(), i, 2);

invoicebuyitemtdm.setValueAt(priceFiled.getText(), i, 3);

invoicebuyitemtdm.setValueAt(amountFiled.getText(), i, 4);

porodictid.setText("");

pnamefiled.setText("");

priceFiled.setText("");

amountFiled.setText("");

qty2Filed.setText("0");

}

double sum = 0;

for (double m = 0; m < invoicebuytable.getRowCount(); m++) {

sum = sum + Double.parseDouble(invoicebuytable.getValueAt((int) m, 4).toString());

}

totalFiled.setText(Double.toString(sum));

}



private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {

int row = invoicebuytable.getSelectedRow();

invoicebuyitemtdm.removeRow(row);

double sum = 0;

for (double i = 0; i < invoicebuytable.getRowCount(); i++) {

sum = sum + Double.parseDouble(invoicebuytable.getValueAt((int) i, 4).toString());

}

totalFiled.setText(Double.toString(sum));

porodictid.setText("");

pnamefiled.setText("");

priceFiled.setText("");

amountFiled.setText("");

// qtySpinner.setValue(0);

porodictid.requestFocus();

}



private void qty2FiledKeyReleased(java.awt.event.KeyEvent evt) {

String qt = qty2Filed.getText();

if(qt!=null){

int qty = Integer.parseInt(qt);

double price = Double.parseDouble(priceFiled.getText());

double total = qty \* price;

amountFiled.setText(String.valueOf(total));

}else{

amountFiled.setText("");

}

}



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

if (evt.getKeyCode() == KeyEvent.VK\_ENTER) {

try {

String pid = porodictid.getText();

Object[] getpdetails = invoiceM.searchProduct(pid);

// JOptionPane.showMessageDialog((Component) getpdetails[2], "Please Select Data from table!", "massege", JOptionPane.INFORMATION\_MESSAGE);

pnamefiled.setText((String) getpdetails[0]);

priceFiled.setText((String) getpdetails[1]);

} catch (Exception e) {

e.printStackTrace();

}

}

}



private void jButton30ActionPerformed(java.awt.event.ActionEvent evt) {

balance();

String invid = invoiceIDfiled.getText();

String cusmob = cuMobileFiled.getText();

String invTotal = totalFiled.getText();

String payment = paymentFiled.getText();

String balance = balanceFiled.getText();

Date pdate = invoicedate.getDate();

if (pdate.toString().isEmpty()) {

JOptionPane.showMessageDialog(this, "Enter payment date", "massege", JOptionPane.INFORMATION\_MESSAGE);

} else {

String inserInv = invoiceM.insertinv(invid, cusmob, invTotal, pdate);

invoiceItemM.loadtableitem("", invoiceItemtdm);

invoiceItemM.loadtableitem("", invoiceItemtdm);

System.out.println(invid);

JOptionPane.showMessageDialog(this, inserInv, "massege", JOptionPane.INFORMATION\_MESSAGE);

autoId();

for (int i = 0; i < invoicebuytable.getRowCount(); i++) {

String iPId = (String) invoicebuyitemtdm.getValueAt(i, 0);

String iqty = (String) invoicebuyitemtdm.getValueAt(i, 2);

System.out.println(iPId + "" + iqty);

String invoiceItem = invoiceM.insertItem(iPId, iqty, invid);

String stockupdate = invoiceM.updateStock(iPId, iqty);

}

invoicebuyitemtdm.setRowCount(0);

invoiceItemM.loadtableitem("", invoiceItemtdm);

try {

String getid = "SELECT id\_invoice FROM pcgadguestshop.invoice WHERE id\_invoice=?";

PreparedStatement ps = con.prepareStatement(getid);

ps.setString(1, invid);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

String id = rs.getString("id\_invoice");

System.out.println(id + "id");

String idd = "11";

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\bill1.jrxml";

Map m = new HashMap();

m.put("p1", id);

m.put("cpayment", payment);

m.put("balance", balance);

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

billframe bf = new billframe();

bf.billtabepane.add(jvv);

bf.setVisible(true);

// billtabe.add(jvv);

stockM.loadStocktb("", stocktdm);

} catch (Exception e) {

e.printStackTrace();

}

}

} catch (Exception e) {

e.printStackTrace();

}

totalFiled.setText("");

paymentFiled.setText("");

balanceFiled.setText("");

cuMobileFiled.setText("Guest");

}

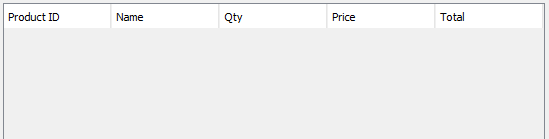
}



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

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

}



private void invoicebuytableMouseClicked(java.awt.event.MouseEvent evt) {

int row = invoicebuytable.getSelectedRow();

porodictid.setText(String.valueOf(invoiceProducttdm.getValueAt(row, 0)));

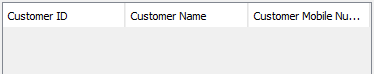
pnamefiled.setText(String.valueOf(invoiceProducttdm.getValueAt(row, 1)));

qty2Filed.setText(String.valueOf(invoiceProducttdm.getValueAt(row, 2)));

priceFiled.setText(String.valueOf(invoiceProducttdm.getValueAt(row, 3)));

amountFiled.setText(String.valueOf(invoiceProducttdm.getValueAt(row, 4)));

}



private void cutomer2tableMouseClicked(java.awt.event.MouseEvent evt) {

int row = cutomer2table.getSelectedRow();

String cusId = (String) customer2tdm.getValueAt(row, 0);

Object[] customerdata = customerM.getSelectId(cusId);

cuMobileFiled.setText(String.valueOf(customerdata[1]));

}

**Model Package Code**

package model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Date;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import static model.testlogin.log;

import org.apache.log4j.Logger;

public class invoiceModel {

static Logger log = Logger.getLogger(invoiceModel.class.getName());

Connection con;

public invoiceModel() {

con=db.dbConnection.getconnection();

}

public Object[] searchProduct(String pid) {

String msg="";

String getData ="SELECT \* FROM pcgadguestshop.product WHERE id\_product=?";

try {

PreparedStatement ps = con.prepareStatement(getData);

ps.setString(1, pid);

ResultSet rs = ps.executeQuery();

Object[] rowData=null;

while(rs.next()){

String pname = rs.getString("productname");

String sellprice = rs.getString("sell\_price");

rowData = new Object[]{pname,sellprice,msg};

}

return rowData;

// }

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

}

return null;

}

public String insertinv(String invid, String cusmob, String invTotal,Date pdate) {

String msg="";

String inser = "INSERT INTO pcgadguestshop.invoice (id\_invoice, date, payment, customer)VALUES(?,?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(inser);

ps.setString(1, invid);

ps.setDate(2, new java.sql.Date(pdate.getTime()));

ps.setString(3, invTotal);

ps.setString(4, cusmob);

ps.execute();

msg="Success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Please Select Date Of Payment";

}

return msg;

}

public String insertItem(String iPId, String iqty, String invid) {

String msg="";

String inserti ="INSERT INTO pcgadguestshop.invoice\_item (qty, product\_id\_product, invoice\_id\_invoice)VALUES(?,?,?)";

try {

PreparedStatement ps = con.prepareStatement(inserti);

ps.setString(1, iqty);

ps.setString(2, iPId);

ps.setString(3, invid);

ps.execute();

msg="Success";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="error";

}

return msg;

}

public String updateStock(String iPId, String iqty) {

String msg="";

String getqty ="SELECT qty FROM pcgadguestshop.stocks WHERE product\_id\_product=?";

String stock = "UPDATE pcgadguestshop.stocks SET qty=? WHERE product\_id\_product=?";

try {

PreparedStatement ps = con.prepareStatement(getqty);

ps.setString(1, iPId);

ResultSet rs = ps.executeQuery();

while(rs.next()){

int pQty = rs.getInt("qty");

int buyQty = Integer.parseInt(iqty);

int newQty = pQty - buyQty;

PreparedStatement ps2 = con.prepareStatement(stock);

ps2.setInt(1, newQty);

ps2.setString(2, iPId);

ps2.execute();

}

} catch (Exception e) {

log.debug(e.getMessage());

}

return msg;

}

public void loadtableInvoice(String keyword, DefaultTableModel invoicedatatdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.invoice WHERE id\_invoice like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

invoicedatatdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_invoice");

String date = rs.getString("date");

String payment = rs.getString("payment");

String customer = rs.getString("customer");

rowData = new Object[]{id,date,payment,customer};

invoicedatatdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public String delete() {

String msg="";

String delete = "DELETE pcgadguestshop.invoice";

try {

PreparedStatement ps = con.prepareStatement(delete);

ps.execute();

msg="Successfully All Deleted";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

}

return msg;

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Cashier Section (customer registration Dialog Page )** | | **GUI Number** : 25 |
| **Use case Name: Customer payment Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This page includes in cashier page. When if customer wants to register in this shop but cashier do not move to the customer registration page cashier can register new customer in cashier page. When cashier click new customer button and will open customer registration subpage in cashier page. Casher can register new customer using customer name and mobile number. When successfully insert customer when display information message to cashier. If customer want to change registered details, then cashier have to go customer registration page and casher can select relevant details about customer and changes details about customer. Owner, manager and cashier can access this customer registration subpage.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert)after show message above activity | | jButton | Its use for Inert new customer | When add data to input filed after click specified button and call their command | | | |

**Source Code**



private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

customerM.loadtableAdmin("", customertdm);

customerM.loadtableAdmin("", customer2tdm);

}



private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

String cusId = customerIDfiled.getText();

String customername = customernameFiled.getText();

String customerModile = customerMdeilfiled.getText();

String insertcustomer = customerM.insertCustomer(customername,customerModile);

JOptionPane.showMessageDialog(this, "Insert Details", "massege",JOptionPane.INFORMATION\_MESSAGE );

customerIDfiled.setText("");

customernameFiled.setText("");

customerMdeilfiled.setText("");

}

**Model Package Code**

public String insertCustomer(String customername, String customerModile) {

String msg;

String insertQuery ="INSERT INTO pcgadguestshop.customer(customer\_mobile, customer\_name)VALUES(?,?)";

try {

PreparedStatement ps = con.prepareStatement(insertQuery);

ps.setString(1,customerModile);

ps.setString(2,customername);

ps.execute();

msg="Successfully Insert Customer";

} catch (Exception e) {

e.printStackTrace();

log.debug(e.getMessage());

msg="Mobile Number Already taken!";

}

return msg;

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Cashier Section (Invoice Page)** | | **GUI Number** : 26 |
| **Use case Name: Customer Payment Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   All invoice details will record in invoice page and main user can see invoice id number, date of payment and total value of invoice will show to user if user wants to search invoice details by using invoice id. And result will show in payment salary page in this page .and user can get report about all invoice details by clicking report button and this page also can access by owner. Manager and cashier   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert)after show message above activity | | jButton | Its use for get report and clear data | When add data to input filed after click specified button and call their command | | jtable | View invoice details | Show history of paid invoice details inside a table | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **GUI** |  | **Descriptions** |
| Dialog  Creation |  |  | Cashier can get all invoice details |

**Source Code**



private void jButton23ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchfiled3.getText();

invoiceM.loadtableInvoice(keyword, invoicedatatdm);

invoiceItemM.loadtableitem("", invoiceItemtdm);

}



private void jButton37ActionPerformed(java.awt.event.ActionEvent evt) {

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

}

**Model package Source Code**

public void loadtableInvoice(String keyword, DefaultTableModel invoicedatatdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.invoice WHERE id\_invoice like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

invoicedatatdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_invoice");

String date = rs.getString("date");

String payment = rs.getString("payment");

String customer = rs.getString("customer");

rowData = new Object[]{id,date,payment,customer};

invoicedatatdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Cashier Section (Invoice item Page)** | | **GUI Number** : 27 |
| **Use case Name: Customer Payment Process** | | |
|  | | |
| 1. **Brief Description of the GUI:**   Take invoice id number by added last good receive note. And it automatically adds data from cashier page. And data will save invoice item about buying details. They are shows product id, buying quantity. the product will show about product id, selling price. if user wants to check id name and user can search product details from them. In product table will shows product name. Id and etc... user can check invoices item tables data and product names from product table. Owner, manger and cashier can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | Joption pane | View information messages | When user do some activity (insert)after show message above activity | | jButton | Its use for search data in table | When add data to input filed after click specified button and call their command | | jtable | View invoice item details and registered product details | Show history of paid invoice item details and product details inside a table | | | |

**Source Code**

**(Invoice item)**

private void jButton22ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = itemsearchfiled.getText();

invoiceItemM.loadtableitem(keyword, invoiceItemtdm);

}

**(Products)**

private void jButton99ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = proSearchFiled3.getText();

productM.loadtable(keyword, product3tdm);

}

**Model Package Code**

/\*

\* 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;

/\*\*

\*

\* @author Acer

\*/

public class invoiceItemModel {

Connection con;

public invoiceItemModel() {

con=db.dbConnection.getconnection();

}

public void loadtableitem(String keyword, DefaultTableModel invoiceItemtdm) {

String loadtb = "SELECT \* FROM pcgadguestshop.invoice\_item WHERE invoice\_id\_invoice like ?";

try {

PreparedStatement ps = con.prepareStatement(loadtb);

ps.setString(1, "%" + keyword + "%");

ResultSet rs = ps.executeQuery();

Object[] rowData;

invoiceItemtdm.setRowCount(0);

while (rs.next()) {

String id = rs.getString("id\_invoice\_item");

String pid = rs.getString("product\_id\_product");

String qty = rs.getString("qty");

String inId = rs.getString("invoice\_id\_invoice");

rowData = new Object[]{id,pid,qty,inId};

invoiceItemtdm.addRow(rowData);

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Customer page (customer report Page)** | | **GUI Number** : 28 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   When user wants to get registered all customers names and their phone number and user can click report button in customer registration page and open a page and it include search input filed, search button, customers detail table and report button and user can search registered customers details and see details about all registered customers details in the table. And user when click report button and report will generate and after generate report will shows to user right side in the customer report page and user can get one more than report in this page and each one of reports will open by single tab. User can save or print this report.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search customer data | When add data to input filed after click specified button and call their command | | jtable | Show registered customers | When user get report about all registered customers and table helps to check printed details | | | |

**Source Code**

package gui;

import java.sql.Connection;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.customerModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class customerReportjDialog extends javax.swing.JDialog {

/\*\*

\* Creates new form customerReportjDialog

\*/

private customerModel customerM;

Connection con;

private DefaultTableModel customertdm;

public customerReportjDialog(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

con=db.dbConnection.getconnection();

customerM = new customerModel();

customertdm= (DefaultTableModel)custable.getModel();

customerM.loadtableAdmin("", customertdm);

}

/\*\*

\* 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();

custombertabpane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jScrollPane2 = new javax.swing.JScrollPane();

custable = new javax.swing.JTable();

jLabel11 = new javax.swing.JLabel();

searchfiled = new javax.swing.JTextField();

jButton14 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Customer", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

custable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Customer ID", "Customer Name", "Customer Mobile Number"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, true, true

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

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

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

custableMouseClicked(evt);

}

});

jScrollPane2.setViewportView(custable);

jLabel11.setText("By ");

jButton14.setBackground(new java.awt.Color(255, 255, 102));

jButton14.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton14.setText("Search");

jButton14.setContentAreaFilled(false);

jButton14.setOpaque(true);

jButton14.addActionListener(new java.awt.event.ActionListener() {

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

jButton14ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Get All Details about Registered Customers");

javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);

jPanel8.setLayout(jPanel8Layout);

jPanel8Layout.setHorizontalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(searchfiled, javax.swing.GroupLayout.PREFERRED\_SIZE, 263, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton14, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap(14, Short.MAX\_VALUE)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(searchfiled, javax.swing.GroupLayout.DEFAULT\_SIZE, 25, Short.MAX\_VALUE)

.addComponent(jButton14, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jLabel11, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 261, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 117, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(custombertabpane, javax.swing.GroupLayout.DEFAULT\_SIZE, 537, 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)

.addComponent(custombertabpane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchfiled.getText();

customerM.loadtableAdmin(keyword, customertdm);

}

private void custableMouseClicked(java.awt.event.MouseEvent evt) {

int row = custable.getSelectedRow();

String cusId = (String) customertdm.getValueAt(row, 0);

Object[] customerdata = customerM.getSelectId(cusId);

//customernameFiled.setText(new String(customerdata[0]+""));

// rid.setText(String.valueOf(customerdata[0]));

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// String invoiceId = invoiceIDField1.getText();

// System.out.println(invoiceId);

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\customerReport3.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

// m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

custombertabpane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

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

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(customerReportjDialog.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() {

customerReportjDialog dialog = new customerReportjDialog(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 custable;

private javax.swing.JTabbedPane custombertabpane;

private javax.swing.JButton jButton14;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextField searchfiled;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier page (supplier report Page)** | | **GUI Number** : 29 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   In this page user can get registered all suppliers as a report. User can see already registered suppliers in suppliers’ details table and user can search supplier’s details in this page. In supplier table shows to user their registered name, mobile number and email. When user clicked get report button and report will generate and generated report will display in right side in this page. User can get one more than report and each generated reports will open single tabs. When report display and user can print this report or save to the computer. Owner, manager and product controller can use this report   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search Supplier data | When add data to input filed after click specified button and call their command | | jtable | Show registered supplier | When user get report about all registered supplier and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.supplierModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class supplerReportDialogbox extends javax.swing.JDialog {

/\*\*

\* Creates new form supplerReportDialogbox

\*/

private supplierModel suplierM;

private DefaultTableModel supliertdm;

Connection con;

public supplerReportDialogbox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

suplierM = new supplierModel();

supliertdm = (DefaultTableModel) suplierTable2.getModel();

suplierM.loadtable("", supliertdm);

con=db.dbConnection.getconnection();

}

/\*\*

\* 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();

suppliertabpane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

supSearchFiled2 = new javax.swing.JTextField();

jButton69 = new javax.swing.JButton();

jScrollPane16 = new javax.swing.JScrollPane();

suplierTable2 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Supplier", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel11.setText("By ");

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Get All Details about Registered Supplier");

jButton69.setBackground(new java.awt.Color(255, 255, 102));

jButton69.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton69.setText("Search");

jButton69.setContentAreaFilled(false);

jButton69.setOpaque(true);

jButton69.addActionListener(new java.awt.event.ActionListener() {

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

jButton69ActionPerformed(evt);

}

});

suplierTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Supplier ID", "Supplier Name", "Mobile Number", "Email"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, true

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

suplierTable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

suplierTable2MouseClicked(evt);

}

});

jScrollPane16.setViewportView(suplierTable2);

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(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(supSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 246, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton69, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(22, 22, 22)

.addComponent(jScrollPane16, javax.swing.GroupLayout.PREFERRED\_SIZE, 468, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(45, Short.MAX\_VALUE))

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(supSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton69, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane16, javax.swing.GroupLayout.PREFERRED\_SIZE, 211, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 91, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(suppliertabpane, javax.swing.GroupLayout.DEFAULT\_SIZE, 665, 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)

.addComponent(suppliertabpane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButton69ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = supSearchFiled2.getText();

suplierM.loadtable(keyword, supliertdm);

}

private void suplierTable2MouseClicked(java.awt.event.MouseEvent evt) {

int row = suplierTable2.getSelectedRow();

String sid = (String) supliertdm.getValueAt(row, 0);

Object[] supllierdata = suplierM.detselectid(sid);

// supid.setText(String.valueOf(supllierdata[0]));

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\supplierReport.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

// m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

suppliertabpane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(supplerReportDialogbox.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() {

supplerReportDialogbox dialog = new supplerReportDialogbox(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.JButton jButton2;

private javax.swing.JButton jButton69;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane16;

private javax.swing.JTextField supSearchFiled2;

private javax.swing.JTable suplierTable2;

private javax.swing.JTabbedPane suppliertabpane;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier page (GRN report Page)** | | **GUI Number** : 30 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This page includes a good receive note page when user click report button in good receive note page and this report page will open and this search bar and search button, details of entered good receive notes and one textbox to get good receive note id and two buttons. User can search good receive notes and the results will show in good receive note table and user want get one report about one good receive note then user have selected GRN id form GRN table then click relevant id from table and selected GRN id will set GRN id text box and click get one report button the report will generate and show to user. If user want get all entered GRN data then user can click get report button and user can get all entered GRN data in single report. Owner, manager and product controller can use this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search GRN data | When add data to input filed after click specified button and call their command | | jtable | Show Insert all GRN details and GRN and GRN item details | When user get report about all insert grn and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import model.grnModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class grnInvoiceDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form grnInvoiceDialogBox

\*/

private grnModel grnM;

private DefaultTableModel grntdm;

Connection con;

public grnInvoiceDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

grnM = new grnModel();

grntdm = (DefaultTableModel) grntable2.getModel();

grnM.loadtable("",grntdm);

con=db.dbConnection.getconnection();

}

/\*\*

\* 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();

grntabPane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

grnidfiled = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

jScrollPane18 = new javax.swing.JScrollPane();

grntable2 = new javax.swing.JTable();

grnSearchFiled2 = new javax.swing.JTextField();

jButton79 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Supplier GRN", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel11.setText("By ");

jButton1.setBackground(new java.awt.Color(0, 51, 51));

jButton1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton1.setForeground(new java.awt.Color(255, 255, 255));

jButton1.setText("Get One Report about one Supplier GRN");

jButton1.setContentAreaFilled(false);

jButton1.setOpaque(true);

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

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

jButton1ActionPerformed(evt);

}

});

jLabel12.setText("GRN ID");

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Get All Details about Supplier GRN");

grntable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"GRN ID", "Date", "Amount", "Supplier", "Supplier Mobile"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

grntable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

grntable2MouseClicked(evt);

}

});

jScrollPane18.setViewportView(grntable2);

jButton79.setBackground(new java.awt.Color(255, 255, 102));

jButton79.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton79.setText("Search");

jButton79.setContentAreaFilled(false);

jButton79.setOpaque(true);

jButton79.addActionListener(new java.awt.event.ActionListener() {

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

jButton79ActionPerformed(evt);

}

});

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(56, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(419, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(grnidfiled, javax.swing.GroupLayout.PREFERRED\_SIZE, 56, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1)))

.addGap(28, 28, 28))))

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(1, 1, 1)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addGap(144, 144, 144)

.addComponent(grnSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 274, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton79, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane18, javax.swing.GroupLayout.DEFAULT\_SIZE, 558, Short.MAX\_VALUE))

.addGap(2, 2, 2)))

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 342, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(grnidfiled, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton79, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(grnSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane18, javax.swing.GroupLayout.PREFERRED\_SIZE, 242, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(116, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(grntabPane, javax.swing.GroupLayout.DEFAULT\_SIZE, 674, 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)

.addComponent(grntabPane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

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

String grinid = grnidfiled.getText();

int gidd = Integer.parseInt(grinid);

if(grinid.isEmpty()){

JOptionPane.showMessageDialog(this, "Cannot Print Plase Select Grn Id !", "massege", JOptionPane.QUESTION\_MESSAGE);

}else{

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\grnWithItem2.jrxml";

Map m = new HashMap();

m.put("gid", gidd);

// m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

grntabPane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Cannot Print Plase Select Grn Id !", "massege", JOptionPane.QUESTION\_MESSAGE);

}

}

}

private void grntable2MouseClicked(java.awt.event.MouseEvent evt) {

int row = grntable2.getSelectedRow();

String grid = (String)grntdm.getValueAt(row, 0);

Object[] grndata = grnM.getselectid(grid);

grnidfiled.setText(String.valueOf(grndata[0]));

}

private void jButton79ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = grnSearchFiled2.getText();

grnM.loadtable(keyword,grntdm);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\grnReport1.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

grntabPane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(grnInvoiceDialogBox.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() {

grnInvoiceDialogBox dialog = new grnInvoiceDialogBox(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.JTextField grnSearchFiled2;

private javax.swing.JTextField grnidfiled;

private javax.swing.JTabbedPane grntabPane;

private javax.swing.JTable grntable2;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton79;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane18;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Supplier Section (GRN Item report Page)** | | **GUI Number** : 31 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This page can get all good receive items as a report. This page includes in good receive item page and click report button in good receive item page and will open good receive item report page and user can see search bar, good receive items details include table and report button. User can search good receive item by using good receive note id number and result will show in table and click get report button and report will generate and generated report will show to user in right side of GUI. Entered all good receive items will shows in this report. And user can save and print report. Owner, manager and product controller can use this page.     1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search Grn item by grn id data | When add data to input filed after click specified button and call their command | | jtable | Show Insert all GRN item details and GRN item | When user get report about all insert grn item and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import javax.swing.JMenuItem;

import javax.swing.JPopupMenu;

import javax.swing.SwingUtilities;

import javax.swing.table.DefaultTableModel;

import model.grnItemModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class grnitemReportDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form grnitemReportDialogBox

\*/

private grnItemModel grnitemM;

private DefaultTableModel grnitemtdm;

Connection con;

public grnitemReportDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

grnitemM = new grnItemModel();

grnitemtdm = (DefaultTableModel) grnitemtable2.getModel();

grnitemM.loadtable("",grnitemtdm);

con = db.dbConnection.getconnection();

}

/\*\*

\* 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();

grnitemtabepane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

jScrollPane19 = new javax.swing.JScrollPane();

grnitemtable2 = new javax.swing.JTable();

grnitemSearchFiled2 = new javax.swing.JTextField();

jButton80 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Customer", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel11.setText("By ");

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Get All Details about Registered Customers");

grnitemtable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"GRN Item ID", "Product Name", "Qty", "Product Buy Price", "GRN ID"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, true

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

grnitemtable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

grnitemtable2MouseClicked(evt);

}

});

jScrollPane19.setViewportView(grnitemtable2);

jButton80.setBackground(new java.awt.Color(255, 255, 102));

jButton80.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton80.setText("Search");

jButton80.setContentAreaFilled(false);

jButton80.setOpaque(true);

jButton80.addActionListener(new java.awt.event.ActionListener() {

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

jButton80ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);

jPanel8.setLayout(jPanel8Layout);

jPanel8Layout.setHorizontalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap(37, Short.MAX\_VALUE)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 31, Short.MAX\_VALUE)

.addComponent(grnitemSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 214, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(79, 79, 79)

.addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton80, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(39, 39, 39))

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jScrollPane19, javax.swing.GroupLayout.PREFERRED\_SIZE, 500, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 13, Short.MAX\_VALUE)))

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton80, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(grnitemSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 300, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(40, 40, 40))

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(55, 55, 55)

.addComponent(jScrollPane19, javax.swing.GroupLayout.PREFERRED\_SIZE, 244, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(119, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(grnitemtabepane, javax.swing.GroupLayout.DEFAULT\_SIZE, 646, 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)

.addComponent(grnitemtabepane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void grnitemtable2MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jButton80ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = grnitemSearchFiled2.getText();

grnitemM.loadtable(keyword,grnitemtdm);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\grnitemReport.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

grnitemtabepane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(grnitemReportDialogBox.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() {

grnitemReportDialogBox dialog = new grnitemReportDialogBox(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.JTextField grnitemSearchFiled2;

private javax.swing.JTabbedPane grnitemtabepane;

private javax.swing.JTable grnitemtable2;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton80;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane19;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee page (All Employee report Page)** | | **GUI Number** : 31 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   User can get report about shop registered employee’s details. This page includes in employee registration page and click get report button in employee registration page and employee report page will open and user can see search bar, details of employee table and report button. User can search employee’s data by using employee name and result will show in employee table. Employee table shows employee name, address, mobile number, post of employee and national identity card number. When click get report button and report will generate and generated report will show right side in GUI page. Owner and manager can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search Employee data | When add data to input filed after click specified button and call their command | | jtable | Show Insert all Employee details. | When user get report about all employees details and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.employeeModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class empoyeeReportDialogbox extends javax.swing.JDialog {

/\*\*

\* Creates new form empoyeeDialogbox

\*/

private employeeModel empM;

private DefaultTableModel emptdm;

Connection con;

public empoyeeReportDialogbox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

empM = new employeeModel();

emptdm = (DefaultTableModel) employeeTable2.getModel();

empM.loadtableEmp("", emptdm);

con=db.dbConnection.getconnection();

}

/\*\*

\* 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();

employeerepostPane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

searchfiled2 = new javax.swing.JTextField();

jButton14 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

jScrollPane6 = new javax.swing.JScrollPane();

employeeTable2 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Employee", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel11.setText("By ");

jButton14.setBackground(new java.awt.Color(255, 255, 102));

jButton14.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton14.setText("Search");

jButton14.setContentAreaFilled(false);

jButton14.setOpaque(true);

jButton14.addActionListener(new java.awt.event.ActionListener() {

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

jButton14ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Print All Employees Details");

employeeTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Employee ID", " Name", " Address", "Mobile Number", "Post of Employee"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.Object.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

employeeTable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

employeeTable2MouseClicked(evt);

}

});

jScrollPane6.setViewportView(employeeTable2);

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()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 245, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(searchfiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 263, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton14, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 628, Short.MAX\_VALUE))

.addContainerGap())

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(searchfiled2, javax.swing.GroupLayout.DEFAULT\_SIZE, 25, Short.MAX\_VALUE)

.addComponent(jButton14, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jLabel11, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

.addComponent(jScrollPane6, javax.swing.GroupLayout.PREFERRED\_SIZE, 258, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(21, 21, 21)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(147, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(18, 18, 18)

.addComponent(employeerepostPane, javax.swing.GroupLayout.PREFERRED\_SIZE, 554, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(employeerepostPane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchfiled2.getText();

empM.loadtableEmp(keyword, emptdm);

}

private void employeeTable2MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\employeeReport1.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

// m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

employeerepostPane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(empoyeeReportDialogbox.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() {

empoyeeReportDialogbox dialog = new empoyeeReportDialogbox(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 employeeTable2;

private javax.swing.JTabbedPane employeerepostPane;

private javax.swing.JButton jButton14;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JTextField searchfiled2;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee page (All Employee Attendance report Page)** | | **GUI Number** : 32 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   User can get report about employee attendance in this page and it include in employee attendance page. When click get report button in employee attendance page and open employee attendance report page and user can see search bar, details, attendance of employee table. When clicked get report button and report will generate and generated report will show in right side of page as a tab. And user can get one more than report about attendance. Also use can print generated report and save to computer. Owner and manager can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search attendance data | When add data to input filed after click specified button and call their command | | jtable | Show Insert all Employee attendance details. | When user get report about all employees attendance details and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.attendanceModel;

import model.employeeModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class attendanceReportDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form attendanceReportDialogBox

\*/

private attendanceModel attendanceM;

private DefaultTableModel attendancetdm;

Connection con;

public attendanceReportDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

attendanceM = new attendanceModel();

attendancetdm = (DefaultTableModel) attendanceTable2.getModel();

attendanceM.loadtablebyid("", attendancetdm);

con = db.dbConnection.getconnection();

}

/\*\*

\* 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() {

javax.swing.JPanel jPanel1 = new javax.swing.JPanel();

attendancetabePane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

jScrollPane7 = new javax.swing.JScrollPane();

attendanceTable2 = new javax.swing.JTable();

searchAttendaceFiled2 = new javax.swing.JTextField();

jButton29 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Employee Attendance", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel11.setText("By ");

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Print All Employees Attendance Details");

attendanceTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Attendance ID", "Employee Name", "Date", "Satatus"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

attendanceTable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

attendanceTable2MouseClicked(evt);

}

});

jScrollPane7.setViewportView(attendanceTable2);

jButton29.setBackground(new java.awt.Color(255, 255, 102));

jButton29.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton29.setText("Search");

jButton29.setContentAreaFilled(false);

jButton29.setOpaque(true);

jButton29.addActionListener(new java.awt.event.ActionListener() {

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

jButton29ActionPerformed(evt);

}

});

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()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 245, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 121, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(47, 47, 47))))

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(15, 15, 15)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(135, 135, 135)

.addComponent(searchAttendaceFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 244, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33)

.addComponent(jButton29, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane7, javax.swing.GroupLayout.PREFERRED\_SIZE, 493, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(15, Short.MAX\_VALUE)))

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 328, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton29, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(searchAttendaceFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane7, javax.swing.GroupLayout.PREFERRED\_SIZE, 255, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(98, 98, 98)))

);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

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

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

.addContainerGap(51, Short.MAX\_VALUE)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(attendancetabePane, javax.swing.GroupLayout.PREFERRED\_SIZE, 566, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(attendancetabePane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void attendanceTable2MouseClicked(java.awt.event.MouseEvent evt) {

// int row = attendanceTable2.getSelectedRow();

// int attenID = (int) attendancetdm.getValueAt(row, 0);

// Object[] attData = attendanceM.getSelectId(attenID);

}

private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = searchAttendaceFiled2.getText();

attendanceM.loadtablebyid(keyword, attendancetdm);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\attendanceReport.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

attendancetabePane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(attendanceReportDialogBox.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() {

attendanceReportDialogBox dialog = new attendanceReportDialogBox(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 attendanceTable2;

private javax.swing.JTabbedPane attendancetabePane;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton29;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel13;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane7;

private javax.swing.JTextField searchAttendaceFiled2;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Employee page (All Employee payments report Page)** | | **GUI Number** : 33 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This page include in employee payment page and user can get two types of reports. When click get report button in employee payment page and employee payment report page will open. User can see search bar, details of employee’s payment table, text box and two buttons. User can search details by using employee id and if user want to get single report about single employee payment, then user must click relevant data row in employee payment table then selected employee payment id will add to employee id text box and click get one report button then generate report and user can get all paid salaries as a report by clicking get report button in employee payment report GUI. Also, user can print and save invoice to computer. Owner and manager can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search employee payment data | When add data to input filed after click specified button and call their command | | jtable | Show Insert all Employee payment details. | When user get report about all employees details and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.paymentEmployeeModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class empPaymentInvoiceDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form empPaymentInvoiceDialogBox

\*/

private paymentEmployeeModel payM;

private DefaultTableModel paymenttdm;

Connection con;

public empPaymentInvoiceDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

payM = new paymentEmployeeModel();

paymenttdm = (DefaultTableModel) employeePaymentTable2.getModel();

payM.loadtable("", paymenttdm);

con=db.dbConnection.getconnection();

}

/\*\*

\* 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();

emppaymenttabpane = new javax.swing.JTabbedPane();

jPanel8 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

empid = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel13 = new javax.swing.JLabel();

jScrollPane11 = new javax.swing.JScrollPane();

employeePaymentTable2 = new javax.swing.JTable();

jLabel48 = new javax.swing.JLabel();

emprPaymentSearchfield2 = new javax.swing.JTextField();

jButton48 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel8.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Employee Payments", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jButton1.setBackground(new java.awt.Color(0, 51, 51));

jButton1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton1.setForeground(new java.awt.Color(255, 255, 255));

jButton1.setText("Get One Emloyee Payment slip");

jButton1.setContentAreaFilled(false);

jButton1.setOpaque(true);

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

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

jButton1ActionPerformed(evt);

}

});

jLabel12.setText("Employee ID");

jButton2.setBackground(new java.awt.Color(0, 51, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Get Report");

jButton2.setContentAreaFilled(false);

jButton2.setOpaque(true);

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

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

jButton2ActionPerformed(evt);

}

});

jLabel13.setText("Get All Details about Employee Payment");

employeePaymentTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Payment ID", "Date of Payment", "Employee ID", "amount"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.Object.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

employeePaymentTable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

employeePaymentTable2MouseClicked(evt);

}

});

jScrollPane11.setViewportView(employeePaymentTable2);

jLabel48.setText("By ");

jButton48.setBackground(new java.awt.Color(255, 255, 102));

jButton48.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton48.setText("Search");

jButton48.setContentAreaFilled(false);

jButton48.setOpaque(true);

jButton48.addActionListener(new java.awt.event.ActionListener() {

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

jButton48ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);

jPanel8.setLayout(jPanel8Layout);

jPanel8Layout.setHorizontalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap(74, Short.MAX\_VALUE)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel8Layout.createSequentialGroup()

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(empid, javax.swing.GroupLayout.PREFERRED\_SIZE, 56, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 123, Short.MAX\_VALUE)

.addComponent(jButton1))))

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane11, javax.swing.GroupLayout.DEFAULT\_SIZE, 530, Short.MAX\_VALUE)))

.addContainerGap())

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel48, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(emprPaymentSearchfield2, javax.swing.GroupLayout.PREFERRED\_SIZE, 274, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton48, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap()))

);

jPanel8Layout.setVerticalGroup(

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel8Layout.createSequentialGroup()

.addContainerGap(84, Short.MAX\_VALUE)

.addComponent(jScrollPane11, javax.swing.GroupLayout.PREFERRED\_SIZE, 212, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(empid, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel8Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton48, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel48, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(emprPaymentSearchfield2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(350, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jPanel8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(emppaymenttabpane, javax.swing.GroupLayout.DEFAULT\_SIZE, 575, 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)

.addComponent(emppaymenttabpane)

.addComponent(jPanel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, 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()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void employeePaymentTable2MouseClicked(java.awt.event.MouseEvent evt) {

int row = employeePaymentTable2.getSelectedRow();

String paymeid = (String) paymenttdm.getValueAt(row, 0);

Object[] paydata = payM.getselectid(paymeid);

empid.setText(String.valueOf(paydata[0]));

}

private void jButton48ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = emprPaymentSearchfield2.getText();

payM.loadtable(keyword, paymenttdm);

}

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

String emppid = empid.getText();

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\SingleEmployeePaymentreport1.jrxml";

Map m = new HashMap();

m.put("emppayid", emppid);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

emppaymenttabpane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\allpaymentReportEmployee.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

emppaymenttabpane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(empPaymentInvoiceDialogBox.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() {

empPaymentInvoiceDialogBox dialog = new empPaymentInvoiceDialogBox(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.JTextField empid;

private javax.swing.JTable employeePaymentTable2;

private javax.swing.JTabbedPane emppaymenttabpane;

private javax.swing.JTextField emprPaymentSearchfield2;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton48;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel48;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel8;

private javax.swing.JScrollPane jScrollPane11;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: product page (All Products report Page)** | | **GUI Number** : 34 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   In this page user have access to get report about shop registered products. This page includes at products registration page. When click get report button in product page and view product report page and user can see search bar and use can search any product in shop registered in this page and result will show in product table. When clicked get report button and report will generate and generated report will show to user in the tab. User can get one more than reports and every report will generate in separate tab. After get report user can compare product table details and report printed details. Manager, owner and product controller can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search products details | When add data to input filed after click specified button and call their command | | jtable | Show Insert all products details. | When user get report about all products details and table helps to check printed details | | | |

**Source Code**

package gui;

import java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.productModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class productreportDialogbox extends javax.swing.JDialog {

/\*\*

\* Creates new form productreportDialogbox

\*/

private productModel productM;

private DefaultTableModel producttdm;

Connection con;

public productreportDialogbox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

con=db.dbConnection.getconnection();

productM = new productModel();

producttdm = (DefaultTableModel) producttable2.getModel();

productM.loadtable("", producttdm);

}

/\*\*

\* 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() {

javax.swing.JPanel jPanel1 = new javax.swing.JPanel();

producttabepane = new javax.swing.JTabbedPane();

jPanel53 = new javax.swing.JPanel();

jLabel59 = new javax.swing.JLabel();

proSearchFiled2 = new javax.swing.JTextField();

jButton64 = new javax.swing.JButton();

jScrollPane15 = new javax.swing.JScrollPane();

producttable2 = new javax.swing.JTable();

jPanel76 = new javax.swing.JPanel();

jLabel45 = new javax.swing.JLabel();

jButton52 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel53.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Product ", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel59.setText("By ");

jButton64.setBackground(new java.awt.Color(255, 255, 102));

jButton64.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton64.setText("Search");

jButton64.setContentAreaFilled(false);

jButton64.setOpaque(true);

jButton64.addActionListener(new java.awt.event.ActionListener() {

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

jButton64ActionPerformed(evt);

}

});

producttable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Product ID", "Name", "Selling Price", "Product Category", "Accessory Name"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

producttable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

producttable2MouseClicked(evt);

}

});

jScrollPane15.setViewportView(producttable2);

jLabel45.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel45.setText("Generate Report");

jButton52.setBackground(new java.awt.Color(0, 102, 102));

jButton52.setForeground(new java.awt.Color(255, 255, 255));

jButton52.setText("Report");

jButton52.setContentAreaFilled(false);

jButton52.setOpaque(true);

jButton52.addActionListener(new java.awt.event.ActionListener() {

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

jButton52ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel76Layout = new javax.swing.GroupLayout(jPanel76);

jPanel76.setLayout(jPanel76Layout);

jPanel76Layout.setHorizontalGroup(

jPanel76Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel76Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel45, javax.swing.GroupLayout.PREFERRED\_SIZE, 248, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton52, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(27, Short.MAX\_VALUE))

);

jPanel76Layout.setVerticalGroup(

jPanel76Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel76Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel76Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel45, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton52, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)))

);

javax.swing.GroupLayout jPanel53Layout = new javax.swing.GroupLayout(jPanel53);

jPanel53.setLayout(jPanel53Layout);

jPanel53Layout.setHorizontalGroup(

jPanel53Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel53Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel59, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(proSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 274, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton64, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane15, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel53Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel76, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(110, 110, 110))

);

jPanel53Layout.setVerticalGroup(

jPanel53Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel53Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel53Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton64, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel59, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(proSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane15, javax.swing.GroupLayout.PREFERRED\_SIZE, 343, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(jPanel76, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(25, 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(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel53, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(producttabepane, javax.swing.GroupLayout.DEFAULT\_SIZE, 687, Short.MAX\_VALUE)

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

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

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

.addContainerGap()

.addComponent(producttabepane, javax.swing.GroupLayout.DEFAULT\_SIZE, 529, Short.MAX\_VALUE)

.addContainerGap())

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

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel53, 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)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButton64ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = proSearchFiled2.getText();

productM.loadtable(keyword, producttdm);

}

private void producttable2MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jButton52ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\productreport1.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

producttabepane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(productreportDialogbox.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() {

productreportDialogbox dialog = new productreportDialogbox(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.JButton jButton52;

private javax.swing.JButton jButton64;

private javax.swing.JLabel jLabel45;

private javax.swing.JLabel jLabel59;

private javax.swing.JPanel jPanel53;

private javax.swing.JPanel jPanel76;

private javax.swing.JScrollPane jScrollPane15;

private javax.swing.JTextField proSearchFiled2;

private javax.swing.JTabbedPane producttabepane;

private javax.swing.JTable producttable2;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Stock page (All Stocks report Page)** | | **GUI Number** : 35 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   User can get report about shop available products details. This page includes at stocks page. When clicked get report button and this interface will display. User can search products with their quantity. The search result will show in stock table when click report button the report will generate and generated report will show to user right side in GUI. User can print generated report or save to computer. User can compare printed details and stocks table details. Owner. Manager and product controller can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search products stocks details | When add data to input filed after click specified button and call their command | | jtable | Show Insert all products Stocks details. | When user get report about all products stocks details and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import javax.swing.table.DefaultTableModel;

import model.stocksModel;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class stockreportDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form stockreportDialogBox

\*/private stocksModel stockM;

private DefaultTableModel stocktdm;

Connection con;

public stockreportDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

con=db.dbConnection.getconnection();

stockM = new stocksModel();

stocktdm = (DefaultTableModel) stokstable2.getModel();

stockM.loadStocktb("", stocktdm);

}

/\*\*

\* 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() {

javax.swing.JPanel jPanel1 = new javax.swing.JPanel();

stocktabepane = new javax.swing.JTabbedPane();

jPanel72 = new javax.swing.JPanel();

jLabel73 = new javax.swing.JLabel();

stocksSearchFiled2 = new javax.swing.JTextField();

jButton92 = new javax.swing.JButton();

jScrollPane17 = new javax.swing.JScrollPane();

stokstable2 = new javax.swing.JTable();

jPanel77 = new javax.swing.JPanel();

jLabel47 = new javax.swing.JLabel();

jButton53 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel72.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Details of Stocks", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 204, 0))); // NOI18N

jLabel73.setText("By ");

jButton92.setBackground(new java.awt.Color(255, 255, 102));

jButton92.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jButton92.setText("Search");

jButton92.setContentAreaFilled(false);

jButton92.setOpaque(true);

jButton92.addActionListener(new java.awt.event.ActionListener() {

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

jButton92ActionPerformed(evt);

}

});

stokstable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Stocks ID", "Product Name", "Product ID", "Qty", "MFD"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

stokstable2.addMouseListener(new java.awt.event.MouseAdapter() {

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

stokstable2MouseClicked(evt);

}

});

jScrollPane17.setViewportView(stokstable2);

jLabel47.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel47.setText("Generate Stock Details as Report");

jButton53.setBackground(new java.awt.Color(0, 102, 102));

jButton53.setForeground(new java.awt.Color(255, 255, 255));

jButton53.setText("Report");

jButton53.setContentAreaFilled(false);

jButton53.setOpaque(true);

jButton53.addActionListener(new java.awt.event.ActionListener() {

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

jButton53ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel77Layout = new javax.swing.GroupLayout(jPanel77);

jPanel77.setLayout(jPanel77Layout);

jPanel77Layout.setHorizontalGroup(

jPanel77Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel77Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel47, javax.swing.GroupLayout.PREFERRED\_SIZE, 248, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton53, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(27, Short.MAX\_VALUE))

);

jPanel77Layout.setVerticalGroup(

jPanel77Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel77Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel77Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel47, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton53, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)))

);

javax.swing.GroupLayout jPanel72Layout = new javax.swing.GroupLayout(jPanel72);

jPanel72.setLayout(jPanel72Layout);

jPanel72Layout.setHorizontalGroup(

jPanel72Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel72Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel72Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel72Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabel73, javax.swing.GroupLayout.PREFERRED\_SIZE, 86, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(stocksSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 274, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton92, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane17, javax.swing.GroupLayout.DEFAULT\_SIZE, 518, Short.MAX\_VALUE)))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel72Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel77, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(46, 46, 46))

);

jPanel72Layout.setVerticalGroup(

jPanel72Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel72Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel72Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton92, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel73, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(stocksSearchFiled2, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane17, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addGap(18, 18, 18)

.addComponent(jPanel77, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(22, 22, 22))

);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

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

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

.addGap(12, 12, 12)

.addComponent(jPanel72, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(18, 18, 18)

.addComponent(stocktabepane, javax.swing.GroupLayout.DEFAULT\_SIZE, 483, Short.MAX\_VALUE)

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

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

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

.addContainerGap()

.addComponent(stocktabepane, javax.swing.GroupLayout.DEFAULT\_SIZE, 507, Short.MAX\_VALUE)

.addContainerGap())

.addComponent(jPanel72, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 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)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void jButton92ActionPerformed(java.awt.event.ActionEvent evt) {

String keyword = stocksSearchFiled2.getText();

stockM.loadStocktb(keyword, stocktdm);

}

private void stokstable2MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jButton53ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\stockreportt.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

//m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

stocktabepane.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(stockreportDialogBox.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() {

stockreportDialogBox dialog = new stockreportDialogBox(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.JButton jButton53;

private javax.swing.JButton jButton92;

private javax.swing.JLabel jLabel47;

private javax.swing.JLabel jLabel73;

private javax.swing.JPanel jPanel72;

private javax.swing.JPanel jPanel77;

private javax.swing.JScrollPane jScrollPane17;

private javax.swing.JTextField stocksSearchFiled2;

private javax.swing.JTabbedPane stocktabepane;

private javax.swing.JTable stokstable2;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Cashier page (invoice report Page)** | | **GUI Number** : 36 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   This interface includes in cashier page. When cashier entered buy product of customer and click bill button and it will show to user this invoice report GUI. This interface does not have any functions and only view relevant buying products of customer s. In this page cashier or user can print invoice or save invoice in computer and owner, manager and cashier can access this invoice report interface in this system   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jtabepane | To view invoice | When click bill button in cashier page all payment details and product details will print in this tabpane | | | |

**Source Code**

/\*

\* 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;

/\*\*

\*

\* @author Acer

\*/

public class billframe extends javax.swing.JFrame {

/\*\*

\* Creates new form billframe

\*/

public billframe() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

billtabepane = new javax.swing.JTabbedPane();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

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

public void windowClosed(java.awt.event.WindowEvent evt) {

formWindowClosed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addComponent(billtabepane, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 736, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addComponent(billtabepane, javax.swing.GroupLayout.PREFERRED\_SIZE, 457, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void formWindowClosed(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(billframe.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 billframe().setVisible(true);

}

});

}

// Variables declaration - do not modify

public javax.swing.JTabbedPane billtabepane;

// End of variables declaration

}

|  |  |  |
| --- | --- | --- |
| **GUI REPORT**  H7DW 04/AS/01    FORM NO: H7DW 04/AS/01/FRM-02 | | |
| **Group Name :** | **Candidate ID : 200022402104** | |
| **Name of the Candidate contributed this UI:** | | |
| **Title of the GUI: Invoice page (All Invoice item report Page)** | | **GUI Number** : 37 |
| **Use case Name:** | | |
|  | | |
| 1. **Brief Description of the GUI:**   User can get all brought products details as a report. This interface includes in invoice item page. When click get report button in invoice item page and displayed this interface. There is one button then click report button and report will generate and generated report will show to user in the tab and also user can get one more than report and each report will generate in separate tabs and user can print invoice or save to computer. Owner. Manager and cashier can access this page.   1. **Justification of GUI component usage:**      |  |  |  | | --- | --- | --- | | Components | Usage | Justification | | jPannel | Using for group jframe components | When adding components to jpanel it easy to manage jframe components and its help to group components. | | jLable | Specifying input fields to recognize each one of input fields | When adding data to fields label helps to recognize each fields | | jButton | Its use for get report and search invoice items details | When add data to input filed after click specified button and call their command | | jtable | Show Insert all invoice item details. | When user get report about all invoice item stocks details and table helps to check printed details | | | |

**Source Code**

/\*

\* 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 java.sql.Connection;

import java.util.HashMap;

import java.util.Map;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

/\*\*

\*

\* @author Acer

\*/

public class invoiceReportDialogBox extends javax.swing.JDialog {

/\*\*

\* Creates new form invoiceReportDialogBox

\*/

Connection con;

public invoiceReportDialogBox(java.awt.Frame parent, boolean modal) {

super(parent, modal);

initComponents();

con=db.dbConnection.getconnection();

}

/\*\*

\* 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();

jButton37 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

jButton37.setBackground(new java.awt.Color(0, 102, 102));

jButton37.setForeground(new java.awt.Color(255, 255, 255));

jButton37.setText("Get Report");

jButton37.setContentAreaFilled(false);

jButton37.setOpaque(true);

jButton37.addActionListener(new java.awt.event.ActionListener() {

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

jButton37ActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

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

.addGap(48, 48, 48)

.addComponent(jButton37, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jTabbedPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 737, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton37, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 404, Short.MAX\_VALUE))

.addComponent(jTabbedPane1))

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButton37ActionPerformed(java.awt.event.ActionEvent evt) {

String reportPath = "C:\\Users\\Acer\\JaspersoftWorkspace\\MyReports\\invoicesall1.jrxml";

Map m = new HashMap();

//m.put("id", invoiceId);

// m.put("Date",new Date(System.currentTimeMillis()));

try {

JasperReport jr = JasperCompileManager.compileReport(reportPath);

JasperPrint jp = JasperFillManager.fillReport(jr, m, con);

JRViewer jvv = new JRViewer(jp);

jTabbedPane1.add(jvv);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

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

if ("Nimbus".equals(info.getName())) {

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

break;

}

}

} catch (ClassNotFoundException ex) {

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(invoiceReportDialogBox.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() {

invoiceReportDialogBox dialog = new invoiceReportDialogBox(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.JButton jButton37;

private javax.swing.JTabbedPane jTabbedPane1;

// End of variables declaration

}