package *aldiwan*;

*public* *class* ALDIWAN {

*public* *static* void *main*(String[] args) {

        java.*awt*.*EventQueue*.*invokeLater*(() -> {

*new* *LoginFrame*().*setVisible*(true);

        });

    }*//main()*

}*// class ALDIWAN {}*

*//////////////////////////////////////////*

*///*

package *aldiwan*;

import *java*.*sql*.*\**;

import *javax*.*swing*.*\**;

*public* *class* CreditForm *extends* javax.swing.JFrame {

*// declare a double to save it to database*

*// declare as public here, for other objects to use it.*

*public* *static* double dblBillAmount;

*// ### Constructor Polymorphism ###*

*public* *CreditForm*() {

*initComponents*();

    }

*/\*\**

*\* Creates new form CreditForm*

*\* Pass double to the constructor*

*\** @param *billAmount*

*\*/*

*public* *CreditForm*(double billAmount) {

        dblBillAmount = billAmount;

*setDefaultCloseOperation*(DISPOSE\_ON\_CLOSE);

*setTitle*("Pay By Credit Card");

*initComponents*();

*setLocationRelativeTo*(null); *// Center the window*

*// get the logged in person name from the SessionPerson Class*

        lblCurrentPerson.*setText*("User: " + SessionPerson.*getCurrentPersonName*());

        lblAmountWithdraw.*setText*("Amount to withdraw: " + billAmount);

    }*// CreditForm() Constructor*

*/\*\**

*\* 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*();

        btnCreditCancel = *new* javax.*swing*.*JButton*();

        btnCreditDone = *new* javax.*swing*.*JButton*();

        lblAmountWithdraw = *new* javax.*swing*.*JLabel*();

        txtCardNumber = *new* javax.*swing*.*JTextField*();

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

        txtNameOnCard = *new* javax.*swing*.*JTextField*();

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

        jLabel4 = *new* javax.*swing*.*JLabel*();

        jSpinnerExpDate = *new* javax.*swing*.*JSpinner*();

        lblCurrentPerson = *new* javax.*swing*.*JLabel*();

*setDefaultCloseOperation*(javax.*swing*.*WindowConstants*.*EXIT\_ON\_CLOSE*);

*setMinimumSize*(*new* java.*awt*.*Dimension*(745, 584));

*setPreferredSize*(*new* java.*awt*.*Dimension*(745, 584));

*getContentPane*().*setLayout*(*new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteLayout*());

        jPanel1.*setBackground*(*new* java.*awt*.*Color*(217, 206, 197));

        jPanel1.*setLayout*(*new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteLayout*());

        btnCreditCancel.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCreditCancel.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        btnCreditCancel.*setText*("Cancel");

        btnCreditCancel.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCreditCancelActionPerformed*(evt);

            }

        });

        jPanel1.*add*(btnCreditCancel, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(460, 450, 140, -1));

        btnCreditDone.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCreditDone.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        btnCreditDone.*setText*("Done");

        btnCreditDone.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCreditDoneActionPerformed*(evt);

            }

        });

        jPanel1.*add*(btnCreditDone, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(140, 450, 140, -1));

        lblAmountWithdraw.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 24)); *// NOI18N*

        lblAmountWithdraw.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        lblAmountWithdraw.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        lblAmountWithdraw.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*LOWERED*));

        jPanel1.*add*(lblAmountWithdraw, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(180, 120, 410, 70));

        txtCardNumber.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtCardNumber.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jPanel1.*add*(txtCardNumber, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(300, 220, 280, 40));

        jLabel2.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel2.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel2.*setText*("Name on Card");

        jPanel1.*add*(jLabel2, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(170, 270, -1, 40));

        txtNameOnCard.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtNameOnCard.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jPanel1.*add*(txtNameOnCard, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(300, 270, 280, 40));

        jLabel3.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel3.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel3.*setText*("Expiry Date");

        jPanel1.*add*(jLabel3, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(170, 320, -1, 40));

        jLabel4.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel4.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel4.*setText*("Credit Card #");

        jPanel1.*add*(jLabel4, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(170, 220, -1, 40));

        jSpinnerExpDate.*setFont*(*new* java.*awt*.*Font*("Dialog", 1, 18)); *// NOI18N*

        jSpinnerExpDate.*setModel*(*new* javax.*swing*.*SpinnerDateModel*());

        jSpinnerExpDate.*setEditor*(*new* javax.*swing*.*JSpinner*.*DateEditor*(jSpinnerExpDate, "dd-MM-yyyy"));

        jPanel1.*add*(jSpinnerExpDate, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(300, 320, 280, 40));

        lblCurrentPerson.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 24)); *// NOI18N*

        lblCurrentPerson.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        lblCurrentPerson.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        lblCurrentPerson.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*LOWERED*));

        jPanel1.*add*(lblCurrentPerson, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(180, 40, 410, 70));

*getContentPane*().*add*(jPanel1, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(0, 0, 940, 650));

*pack*();

    }*// </editor-fold>*

*private* void *btnCreditCancelActionPerformed*(java.awt.event.ActionEvent evt) {

*// close the Credit Card frame,*

*// then the menu will appear*

        this.*dispose*();

    }

*private* void *btnCreditDoneActionPerformed*(java.awt.event.ActionEvent evt) {

*try*{

*// prepare the values*

*// get the current Person ID from SessionPerson Class*

            int current\_person\_id = SessionPerson.*getCurrentPersonID*();

            double cash\_Withdraw = dblBillAmount;

            String card\_no = txtCardNumber.*getText*();

            String name\_On\_Card = txtNameOnCard.*getText*();

*// Retrieve the selected date as a java.util.Date*

            java.util.Date selectedDate = (java.*util*.*Date*) jSpinnerExpDate.*getValue*();

*// Convert the java.util.Date to java.sql.Date*

            java.sql.Date exp\_date = *new* java.*sql*.*Date*(selectedDate.*getTime*());

*// DBConnection.connection ==> declared in the DBConnection CLASS*

            Connection conn = (Connection) DBConnection.*connection*;

            int quantity = SessionPerson.*getTableRowsCount*();

            String sqlInsertOrders = "INSERT INTO orders (user\_id, item\_id, type, quantity, total\_cost)"

            + " VALUES (?,?,?,?,?)";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sqlInsertOrders);

            preparedStatement.*setInt*(1, current\_person\_id); *// for customer column too*

            preparedStatement.*setInt*(2, current\_person\_id);

            preparedStatement.*setString*(3, "Credit");

            preparedStatement.*setInt*(4, quantity);

            preparedStatement.*setDouble*(5, dblBillAmount);

            int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                JOptionPane.*showMessageDialog*(this,

                    "Order Saved in ORDERS Successfully",

                    "Save Done",

                    JOptionPane.*INFORMATION\_MESSAGE*);

            } *else* {

                JOptionPane.*showMessageDialog*(this,

                    "Failed to save in ORDERS table",

                    "Save Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            }*// if (rowsAffected)*

            String sql = "INSERT INTO paycredit (customer\_id, person\_id, cash\_withdraw, cardNo, name\_on\_card, expDate)"

                    + "VALUES (?,?,?,?,?,?)";

            PreparedStatement ps = conn.*prepareStatement* (sql);

            ps.*setInt*(1, current\_person\_id); *// for customer column too*

            ps.*setInt*(2, current\_person\_id);

            ps.*setDouble*(3, cash\_Withdraw);

            ps.*setString*(4, card\_no);

            ps.*setString*(5, name\_On\_Card);

            ps.*setDate*(6, exp\_date);

*// Execute the query*

            rowsAffected = ps.*executeUpdate*();

*if* (rowsAffected > 0) {

                JOptionPane.*showMessageDialog*(this,

                    "Order Saved in PayCredit Successfully",

                    "Save Done",

                    JOptionPane.*INFORMATION\_MESSAGE*);

            } *else* {

                JOptionPane.*showMessageDialog*(this,

                    "Failed to save Order Information",

                    "Save Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            }*// if (rowsAffected)*

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\nCredit Form\n""" + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// save to database then close the Credit Card frame,*

*// then the menu will appear to input a new order*

        this.*dispose*();

    }

*/\*\**

*\** @param *args the command line arguments*

*\*/*

*public* *static* void *main*(String args[]) {

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

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

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

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

*\*/*

*try* {

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

        } *catch* (InstantiationException ex) {

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

        } *catch* (IllegalAccessException ex) {

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

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

            java.*util*.*logging*.*Logger*.*getLogger*(CreditForm.*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*() {

*// call the polymorphismed class with no arguments*

*new* *CreditForm*().*setVisible*(true);

            }

        });

    }*// main()*

*// Variables declaration - do not modify*

*private* javax.swing.JButton btnCreditCancel;

*private* javax.swing.JButton btnCreditDone;

*private* javax.swing.JLabel jLabel2;

*private* javax.swing.JLabel jLabel3;

*private* javax.swing.JLabel jLabel4;

*private* javax.swing.JPanel jPanel1;

*private* javax.swing.JSpinner jSpinnerExpDate;

*private* javax.swing.JLabel lblAmountWithdraw;

*private* javax.swing.JLabel lblCurrentPerson;

*private* javax.swing.JTextField txtCardNumber;

*private* javax.swing.JTextField txtNameOnCard;

*// End of variables declaration*

} *// CreditForm{}*

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

*// this is the database connection Class.*

import *java*.*sql*.*\**;

*// it contains the connection variables.*

*// to be used in all other classes.*

*public* *class* DBConnection {

*public* *static* Connection connection;

*public* *static* *final* String URL = "jdbc:mysql://localhost:3306/aldiwandb";

*public* *static* *final* String DBUser = "root";

*public* *static* *final* String DBPassword = "root";

*static*{

*try*{

*// Replace with your database URL, username, and password*

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

        String username = "root";

        String password = "root";

*// Initialize the connection*

        connection = DriverManager.*getConnection*(url, username, password);

    }*catch*(SQLException e) {

*throw* *new* *RuntimeException*("""

                                   Failed to establish database connection.

                                   Reason: """ +  e.*getMessage*());

    } *// try catch block// try catch{}*

}

*// a method to retrieve the connection*

*public* *static* Connection *getConnection*() {

*return* connection;

    }*// getConnection()*

} *// DBConnection {}*

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

import *javax*.*swing*.*\**;

import *java*.*awt*.*\**;

import *java*.*awt*.*event*.*ActionEvent*;

import *java*.*awt*.*event*.*ActionListener*;

import *java*.*sql*.*Connection*;

import *java*.*sql*.*PreparedStatement*;

import *java*.*sql*.*ResultSet*;

import *java*.*sql*.*SQLException*;

import *javax*.*swing*.*border*.*Border*;

import *javax*.*swing*.*plaf*.*basic*.*BasicTabbedPaneUI*;

import *org*.*jfree*.*chart*.*ChartFactory*;

import *org*.*jfree*.*chart*.*ChartPanel*;

import *org*.*jfree*.*chart*.*JFreeChart*;

import *org*.*jfree*.*data*.*category*.*DefaultCategoryDataset*;

import *org*.*jfree*.*data*.*general*.*DefaultPieDataset*;

*public* *class* Dash2 *extends* javax.swing.JFrame {

*/\*\**

*\* Creates new form Dash2*

*\* Pass the Full name and Role from the database to the Class*

*\** @param *fullname*

*\** @param *role*

*\*/*

*public* *Dash2*(String fullname, String role) {

        Font font16Bold = *new* *Font*("Dialog", Font.*BOLD*, 16);

        Color fontColor = *new* *Color*(89, 44, 73);

*// Create the JFrame*

        JFrame frame = *new* *JFrame*("ALDIWAN Restaurant Dashboard");

*setDefaultCloseOperation*(JFrame.*DISPOSE\_ON\_CLOSE*);

*setLocationRelativeTo*(null); *// Center to frame*

        frame.*setExtendedState*(Dash2.*MAXIMIZED\_BOTH*); *// Maximize the window*

*// Set the default size of the window*

        frame.*setSize*(900, 700); *// the window restore state Width & Height*

        frame.*setLayout*(*new* *BorderLayout*());

*// Top Panel (Title)*

        JPanel topPanel = *new* *JPanel*();

        topPanel.*setBackground*(Color.*decode*("#400601"));

        topPanel.*setPreferredSize*(*new* *Dimension*(frame.*getWidth*(), 50));

        JLabel titleLabel = *new* *JLabel*("", SwingConstants.*CENTER*);

        titleLabel.*setText*("Welcome: " + fullname);

        titleLabel.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 20));

        titleLabel.*setForeground*(Color.*WHITE*);

        topPanel.*add*(titleLabel);

        frame.*add*(topPanel, BorderLayout.*NORTH*);

*// Left Navigation Panel*

        JPanel leftPanel = *new* *JPanel*();

*// GridLayout(Rows, Columns, H Space, V space)*

*//  leftPanel.setLayout(new GridLayout(4, 1, 0, 10));*

        leftPanel.*setLayout*(*new* *GridLayout*(12, 1, 0, 20));

        leftPanel.*setBackground*(Color.*decode*("#F2DCEB"));

        leftPanel.*setPreferredSize*(*new* *Dimension*(200, frame.*getHeight*()));

        JLabel gapLabel = *new* *JLabel*("", SwingConstants.*CENTER*);

        gapLabel.*setOpaque*(false);

*// Define the top and bottom borders*

        Border topBottomBorder = BorderFactory.*createMatteBorder*(

                1, *// Top border thickness (pixels)*

                0, *// Left border thickness (pixels)*

                1, *// Bottom border thickness (pixels)*

                0, *// Right border thickness (pixels)*

                Color.*decode*("#400601")); *// Border color*

        JLabel lblDashboardTitle = *new* *JLabel*("Dashboard", SwingConstants.*CENTER*);

        lblDashboardTitle.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 18));

        lblDashboardTitle.*setForeground*(Color.*decode*("#400601"));

        lblDashboardTitle.*setOpaque*(true);

        lblDashboardTitle.*setBackground*(Color.*decode*("#F2D7B6"));

        lblDashboardTitle.*setBorder*(topBottomBorder);

        JLabel lbladdUser = *new* *JLabel*("Add User", SwingConstants.*CENTER*);

        lbladdUser.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 18));

        lbladdUser.*setForeground*(Color.*decode*("#400601"));

        lbladdUser.*setOpaque*(true);

        lbladdUser.*setBackground*(Color.*decode*("#F2CAB3"));

        lbladdUser.*setBorder*(topBottomBorder);

        JLabel lblUupdateUser = *new* *JLabel*("Update/ Delete User", SwingConstants.*CENTER*);

        lblUupdateUser.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 18));

        lblUupdateUser.*setForeground*(Color.*decode*("#400601"));

        lblUupdateUser.*setOpaque*(true);

        lblUupdateUser.*setBackground*(Color.*decode*("#D9C5D2"));

        lblUupdateUser.*setBorder*(topBottomBorder);

        JLabel exitLabel = *new* *JLabel*("Exit", SwingConstants.*CENTER*);

        exitLabel.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 18));

        exitLabel.*setForeground*(Color.*decode*("#400601"));

        exitLabel.*setOpaque*(true);

        exitLabel.*setBackground*(Color.*decode*("#F294AD"));

        exitLabel.*setBorder*(topBottomBorder);

        leftPanel.*add*(gapLabel);

        leftPanel.*add*(lblDashboardTitle);

        leftPanel.*add*(lbladdUser);

        leftPanel.*add*(lblUupdateUser);

        leftPanel.*add*(exitLabel);

        frame.*add*(leftPanel, BorderLayout.*WEST*);

*// Content Area (JTabbedPane) define properties*

        JTabbedPane tabbedPane = *new* *JTabbedPane*();

        tabbedPane.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 16));

*// Custom UI to set the selected or NOT selected tabs background color*

            tabbedPane.*setUI*(*new* *BasicTabbedPaneUI*() {

                @Override

*protected* void *paintTabBackground*(

                        Graphics g,

                        int tabPlacement,

                        int tabIndex,

                        int x, int y,

                        int w, int h,

                        boolean isSelected) {

*if* (isSelected) {

                        g.*setColor*(Color.*decode*("#400601")); *// Set background color for selected tab*

                        tabbedPane.*setForeground*(Color.*decode*("#F2DCEB"));

                    } *else* {

                        g.*setColor*(Color.*decode*("#F2CAB3")); *// Set background color for unselected tabs*

                        tabbedPane.*setForeground*(Color.*decode*("#400601"));

                    }

                    g.*fillRect*(x, y, w, h);

                }

            });

*// First Tab contents: Numeric Values and Charts*

        JPanel firstTab = *new* *JPanel*(*new* *BorderLayout*());

        firstTab.*setBackground*(Color.*decode*("#F25430"));

*// calculate and retreave the statistics from the database tabels*

*// to fill the 4 boxes on top*

        int userCount = 0;

        int mgrCount = 0;

        int orderCount = 0;

        int cashCount = 0;

        int creditCount = 0;

        int takeawayCount = 0;

        int menuItemsCount = 0;

*try*{

            Connection conn = DBConnection.*connection*;

*// this sql statement counts the number of rows in the table*

*// the alias in the resultSet will be used to get the values*

            String sql\_userCount = "SELECT COUNT(\*) AS userCount FROM person WHERE role = 'user'";

            String sql\_mgrCount = "SELECT COUNT(\*) AS mgrCount FROM person WHERE role = 'manager'";

            String sql\_order\_Count = "SELECT COUNT(\*) AS orderCount FROM orders";

            String sql\_cash\_Count = "SELECT COUNT(\*) AS cashCount FROM orders WHERE type = 'Cash'";

            String sql\_credit\_Count = "SELECT COUNT(\*) AS creditCount FROM orders WHERE type = 'Credit'";

            String sql\_takeaway\_Count = "SELECT COUNT(\*) AS takeawayCount FROM orders WHERE type = 'Takeaway'";

            String sql\_menuItemsCount = "SELECT COUNT(\*) AS menuItemsCount FROM menu\_items";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql\_userCount);

            ResultSet userCountResultSet = preparedStatement.*executeQuery*();

*if* (userCountResultSet.*next*()){

                userCount = userCountResultSet.*getInt*("userCount");

*// this also works, since the result is noe column,*

*// and no need to use the Alias "AS userCount"*

*// userCount = userCountResultSet.getInt(1);*

            }

            PreparedStatement mgrCountPS = conn.*prepareStatement* (sql\_mgrCount);

            ResultSet mgrCountResultSet = mgrCountPS.*executeQuery*();

*if* (mgrCountResultSet.*next*()){

                mgrCount = mgrCountResultSet.*getInt*("mgrCount");

            }

            PreparedStatement orderCountPS = conn.*prepareStatement* (sql\_order\_Count);

            ResultSet orderCountResultSet = orderCountPS.*executeQuery*();

*if* (orderCountResultSet.*next*()){

                orderCount = orderCountResultSet.*getInt*("orderCount");

            }

            PreparedStatement cashCountPS = conn.*prepareStatement* (sql\_cash\_Count);

            ResultSet cashCountResultSet = cashCountPS.*executeQuery*();

*if* (cashCountResultSet.*next*()){

                cashCount = cashCountResultSet.*getInt*("cashCount");

            }

            PreparedStatement creditCountPS = conn.*prepareStatement* (sql\_credit\_Count);

            ResultSet creditCountResultSet = creditCountPS.*executeQuery*();

*if* (creditCountResultSet.*next*()){

                creditCount = creditCountResultSet.*getInt*("creditCount");

            }

            PreparedStatement takeaway\_orderCountPS = conn.*prepareStatement* (sql\_takeaway\_Count);

            ResultSet takeawayCountResultSet = takeaway\_orderCountPS.*executeQuery*();

*if* (takeawayCountResultSet.*next*()){

                takeawayCount = takeawayCountResultSet.*getInt*("takeawayCount");

            }

            PreparedStatement menuItemsCountPS = conn.*prepareStatement* (sql\_menuItemsCount);

            ResultSet menuItemsCountResultSet = menuItemsCountPS.*executeQuery*();

*if* (menuItemsCountResultSet.*next*()){

                menuItemsCount = menuItemsCountResultSet.*getInt*("menuItemsCount");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(null,

                    """

                    Failed Connect

                    \nReason:\n Dash2 Class \n""" + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// use these values to Set padding using an EmptyBorder*

        int top = 10;    *// Top padding*

        int left = 0;   *// Left padding*

        int bottom = 10; *// Bottom padding*

        int right = 0;  *// Right padding*

*// Numeric Value Boxes on top*

*//                          GridLayout(Rows, Columns, H Space, V space)*

        JPanel numericBoxesPanel = *new* *JPanel*(*new* *GridLayout*(1, 4, 10, 10));

        numericBoxesPanel.*setBackground*(Color.*decode*("#F2DCEB"));

        JLabel box1Label = *new* *JLabel*("", SwingConstants.*CENTER*);

        box1Label.*setOpaque*(true);

        box1Label.*setForeground*(Color.*decode*("#400601"));

        box1Label.*setBackground*(Color.*decode*("#F294AD"));

        box1Label.*setText*("Employees: " + userCount);

        box1Label.*setBorder*(BorderFactory.*createEmptyBorder*(top, left, bottom, right));

        JLabel box2Label = *new* *JLabel*("Managers: ", SwingConstants.*CENTER*);

        box2Label.*setOpaque*(true);

        box2Label.*setForeground*(Color.*decode*("#400601"));

        box2Label.*setBackground*(Color.*decode*("#F294AD"));

        box2Label.*setText*("Managers: " + mgrCount);

        box2Label.*setBorder*(BorderFactory.*createEmptyBorder*(top, left, bottom, right));

        JLabel box3Label = *new* *JLabel*("Orders: ", SwingConstants.*CENTER*);

        box3Label.*setOpaque*(true);

        box3Label.*setForeground*(Color.*decode*("#400601"));

        box3Label.*setBackground*(Color.*decode*("#F294AD"));

        box3Label.*setText*("Orders: " + orderCount);

        box3Label.*setBorder*(BorderFactory.*createEmptyBorder*(top, left, bottom, right));

        JLabel box4Label = *new* *JLabel*("Menu Items:", SwingConstants.*CENTER*);

        box4Label.*setOpaque*(true);

        box4Label.*setForeground*(Color.*decode*("#400601"));

        box4Label.*setBackground*(Color.*decode*("#F294AD"));

        box4Label.*setText*("Menu Items: " + menuItemsCount);

        box4Label.*setBorder*(BorderFactory.*createEmptyBorder*(top, left, bottom, right));

        box1Label.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 20));

        box2Label.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 20));

        box3Label.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 20));

        box4Label.*setFont*(*new* *Font*("Arial", Font.*BOLD*, 20));

        box1Label.*setHorizontalAlignment*(JTextField.*CENTER*);

        box2Label.*setHorizontalAlignment*(JTextField.*CENTER*);

        box3Label.*setHorizontalAlignment*(JTextField.*CENTER*);

        box4Label.*setHorizontalAlignment*(JTextField.*CENTER*);

        numericBoxesPanel.*add*(box1Label);

        numericBoxesPanel.*add*(box2Label);

        numericBoxesPanel.*add*(box3Label);

        numericBoxesPanel.*add*(box4Label);

        firstTab.*add*(numericBoxesPanel, BorderLayout.*NORTH*);

*// Charts Panel*

        JPanel chartsPanel = *new* *JPanel*(*new* *GridLayout*(2, 2, 10, 10));

        chartsPanel.*setBackground*(Color.*decode*("#F2D7B6"));

*// Column Chart*

        DefaultCategoryDataset columnDataset = *new* *DefaultCategoryDataset*();

        columnDataset.*addValue*(mgrCount, "Managers", "Managers");

        columnDataset.*addValue*(userCount, "Employees", "Employees");

        JFreeChart columnChart = ChartFactory.*createBarChart*("Number of Managers to Employees", "Category", "Number", columnDataset);

        ChartPanel columnChartPanel = *new* *ChartPanel*(columnChart);

        chartsPanel.*add*(columnChartPanel);

*// Bar Chart*

        DefaultCategoryDataset barDataset = *new* *DefaultCategoryDataset*();

        barDataset.*addValue*(cashCount, "Cash", "Cash Payed");

        barDataset.*addValue*(creditCount, "Credit", "Credit Payed");

        barDataset.*addValue*(takeawayCount, "Takeaway", "Pay On Delivery");

        JFreeChart barChart = ChartFactory.*createBarChart*("Number of Pay Types", "", "", barDataset);

        ChartPanel barChartPanel = *new* *ChartPanel*(barChart);

        chartsPanel.*add*(barChartPanel);

*// Pie Chart*

        DefaultPieDataset pieDataset = *new* *DefaultPieDataset*();

        pieDataset.*setValue*("cash Orders: " + cashCount, cashCount);

        pieDataset.*setValue*("Credit Orders: " + creditCount, creditCount);

        pieDataset.*setValue*("Takeaway Orders: " + takeawayCount, takeawayCount);

        JFreeChart pieChart = ChartFactory.*createPieChart*("Orders Categorized as Cash, Credit card and Takeaway", pieDataset);

        ChartPanel pieChartPanel = *new* *ChartPanel*(pieChart);

        chartsPanel.*add*(pieChartPanel);

        firstTab.*add*(chartsPanel, BorderLayout.*CENTER*);

        tabbedPane.*addTab*("Dashboard", firstTab);

*///////////////////////////////////////////////////////////////////////////////////*

*// Second Tab -- Add New User to the database table person*

*// addUser Tab: Textboxes and Save Button*

        JPanel addUserTab = *new* *JPanel*(*new* *BorderLayout*());

        addUserTab.*setBackground*(Color.*decode*("#F2DCEB"));

        JPanel inputFormPane = *new* *JPanel*(*new* *BorderLayout*());

        inputFormPane.*setBackground*(Color.*decode*("#F2CAB3"));

        inputFormPane.*setPreferredSize*(*new* *Dimension*(800,600));*// 60% of screen size*

        JLabel lblTitle = *new* *JLabel*("Add New User");

*// JLabel.setBounds( X, Y + i \* gapBetweenRows, labelWidth, componentHeight);*

        lblTitle.*setBounds*(250, 40, 180, 40);

        lblTitle.*setBorder*(topBottomBorder);

        lblTitle.*setFont*(*new* *Font*("Dialog", Font.*BOLD*, 25));

        lblTitle.*setForeground*(fontColor);

        inputFormPane.*add*(lblTitle, BorderLayout.*CENTER*);

        JLabel lblUsername = *new* *JLabel*("Username: ");

*// JLabel.setBounds( X, Y + i \* gapBetweenRows, labelWidth, componentHeight);*

        lblUsername.*setBounds*(100, 100, 100, 40);

        lblUsername.*setFont*(font16Bold);

        lblUsername.*setForeground*(fontColor);

        inputFormPane.*add*(lblUsername);

        JTextField txtUsername = *new* *JTextField*();

*// JTextField.setBounds( X, Y + i \* gapBetweenRows, textWidth, componentHeight);*

        txtUsername.*setBounds*(220, 100, 250, 40);

        txtUsername.*setFont*(font16Bold);

        txtUsername.*setForeground*(fontColor);

        inputFormPane.*add*(txtUsername);

        JLabel lblPassword = *new* *JLabel*("Password: ");

        lblPassword.*setBounds*(100, 150, 100, 40);

        lblPassword.*setFont*(font16Bold);

        lblPassword.*setForeground*(fontColor);

        inputFormPane.*add*(lblPassword);

        JTextField txtPassword = *new* *JTextField*();

        txtPassword.*setBounds*(220, 150, 250, 40);

        txtPassword.*setFont*(font16Bold);

        txtPassword.*setForeground*(fontColor);

        inputFormPane.*add*(txtPassword);

        JLabel lblRole = *new* *JLabel*("Role: ");

        lblRole.*setBounds*(100, 200, 100, 40);

        lblRole.*setFont*(font16Bold);

        lblRole.*setForeground*(fontColor);

        inputFormPane.*add*(lblRole);

        JComboBox<String> comboRole = *new* JComboBox<>();

        comboRole.*addItem*("Manager");

        comboRole.*addItem*("User");

        comboRole.*setSelectedIndex*(1); *// Selects "User" as default Value*

        comboRole.*setBounds*(220, 200, 250, 40);

        comboRole.*setFont*(font16Bold);

        comboRole.*setForeground*(fontColor);

        inputFormPane.*add*(comboRole);

        JLabel lblFullname = *new* *JLabel*("Fullname: ");

        lblFullname.*setBounds*(100, 250, 100, 40);

        lblFullname.*setFont*(font16Bold);

        lblFullname.*setForeground*(fontColor);

        inputFormPane.*add*(lblFullname);

        JTextField txtFullname = *new* *JTextField*();

        txtFullname.*setBounds*(220, 250, 250, 40);

        txtFullname.*setFont*(font16Bold);

        txtFullname.*setForeground*(fontColor);

        inputFormPane.*add*(txtFullname);

        JButton btnSave = *new* *JButton*("Save");

        btnSave.*setBounds*(250, 300, 100, 30);

        btnSave.*setFont*(font16Bold);

        btnSave.*setForeground*(fontColor);

        inputFormPane.*add*(btnSave);

*// the last object takes up the rest of the space*

*// I created this Bummy lbl to keep the interface intact.*

        JLabel lblDummy = *new* *JLabel*("");

        inputFormPane.*add*(lblDummy);

        addUserTab.*add*(inputFormPane, BorderLayout.*CENTER*);

        tabbedPane.*addTab*("Add User", addUserTab);

*// ============ Save Button ActionListener ===========*

        btnSave.*addActionListener*(*new* *ActionListener*(){

            @Override

*public* void *actionPerformed*(ActionEvent e){

*// Read all the values*

                String username = txtUsername.*getText*();

                String selectedRole = (String) comboRole.*getSelectedItem*();

                String password = txtPassword.*getText*();

                String fullname = txtFullname.*getText*();

*if*(username.*equals*("") ||

                    password.*equals*("") ||

                    fullname.*equals*("")){

                    JOptionPane.*showMessageDialog*(addUserTab,

                    """

                    Empty Field.\n

                    Reason:\n At least one of the feilds is EMPTY\n """,

                    "Please, fill All the fields.",

                    JOptionPane.*ERROR\_MESSAGE*);

                }*else*{

*try*{

*// DBConnection.connection ==> declared in the DBConnection CLASS*

                        Connection conn = (Connection) DBConnection.*connection*;

                        String sqlInsert = "INSERT INTO person (name, role, password, fullname)"

                                + " VALUES (?,?,?,?)";

                        PreparedStatement preparedStatement = conn.*prepareStatement* (sqlInsert);

                        preparedStatement.*setString*(1, username); *// for customer column too*

                        preparedStatement.*setString*(2, selectedRole);

                        preparedStatement.*setString*(3, password);

                        preparedStatement.*setString*(4, fullname);

*// Execute the query*

                        int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                User Updated\n

                                Fullname: """+fullname,

                                "Saving Done",

                                JOptionPane.*INFORMATION\_MESSAGE*);

*// Clear the ttext fields to be ready for the next input*

                            txtUsername.*setText*("");

                            comboRole.*setSelectedIndex*(1);

                            txtPassword.*setText*("");

                            txtFullname.*setText*("");

                        } *else* {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                "Failed to save User Information",

                                "Save Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                        }*// if (rowsAffected)*

                        }*catch*(SQLException ex) {

                        JOptionPane.*showMessageDialog*(null,

                                """

                                Failed to Connect\n

                                Reason:\nUpdate/ Delete Form\n""" + ex.*getMessage*(),

                                "Connection Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                    } *// try catch block*

                }*// if(username.equals("")|| ......*

            } *// btnSave action Performed*

        });*// btnSave add action listener*

*// ----------- Save Button ENDs Here -----------*

*///////////////////////////////////////////////////////////////////////////*

*// Third Tab: Textboxes and Save Button*

*// Update Delete Tab: Labels, Textboxes and Save Button*

        JPanel updateDeleteTab = *new* *JPanel*(*new* *BorderLayout*());

        updateDeleteTab.*setBackground*(Color.*decode*("#F2DCEB"));

        JPanel updateDeletePanel = *new* *JPanel*(*new* *BorderLayout*());

        updateDeletePanel.*setBackground*(Color.*decode*("#D9C5D2"));

        JLabel lblTitleUD = *new* *JLabel*("Update OR Delete User From the DATABASE");

*// JLabel.setBounds( X, Y + i \* gapBetweenRows, labelWidth, componentHeight);*

        lblTitleUD.*setBounds*(100, 40, 550, 40);

        lblTitleUD.*setBorder*(topBottomBorder);

        lblTitleUD.*setFont*(*new* *Font*("Dialog", Font.*BOLD*, 25));

        lblTitleUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(lblTitleUD, BorderLayout.*CENTER*);

*// Use the "DefaultComboBoxModel" to be able to refresh the ComboBox*

        DefaultComboBoxModel<String> comboBoxModel = *new* DefaultComboBoxModel<>();

        comboBoxModel.*removeAllElements*(); *// Clear at every Entry*

        JComboBox<String> comboPersonName = *new* JComboBox<>(comboBoxModel);

*try*{

*// set "Select Person Name" as the first item*

            comboPersonName.*insertItemAt*("Select Person Name", 0);

            comboPersonName.*setSelectedIndex*(0); *// Set as default*

*// DBConnection.connection ==> declared in the DBConnection CLASS*

            Connection conn = (Connection) DBConnection.*connection*;

            String sqlSelect = "SELECT fullname FROM person ORDER BY fullname";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sqlSelect);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*// fill the comboBox with Fullnames from the database*

*while*(resultSet.*next*()){

*// "fullname" is the query result from the database table column*

                String personName = resultSet.*getString*("fullname");

                comboPersonName.*addItem*(personName);

            }*// while*

            }*catch*(SQLException ex) {

                JOptionPane.*showMessageDialog*(addUserTab,

                    """

                    Update/ Delete Form...Fill Combobox\n

                    Failed to Connect\n

                    Reason:\n""" + ex.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            } *// try catch block*

*//========================================================================*

        comboPersonName.*setBounds*(220, 100, 250, 40);

        comboPersonName.*setFont*(font16Bold);

        comboPersonName.*setForeground*(fontColor);

        updateDeletePanel.*add*(comboPersonName);

        JLabel lblUsernameUD = *new* *JLabel*("Username: ");

*// JLabel.setBounds( X, Y + i \* gapBetweenRows, labelWidth, componentHeight);*

        lblUsernameUD.*setBounds*(100, 150, 100, 40);

        lblUsernameUD.*setFont*(font16Bold);

        lblUsernameUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(lblUsernameUD);

        JTextField txtUsernameUD = *new* *JTextField*();

*// JTextField.setBounds( X, Y + i \* gapBetweenRows, textWidth, componentHeight);*

        txtUsernameUD.*setBounds*(220, 150, 250, 40);

        txtUsernameUD.*setFont*(font16Bold);

        txtUsernameUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(txtUsernameUD);

        JLabel lblPasswordUD = *new* *JLabel*("Password: ");

        lblPasswordUD.*setBounds*(100, 200, 100, 40);

        lblPasswordUD.*setFont*(font16Bold);

        lblPasswordUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(lblPasswordUD);

        JTextField txtPasswordUD = *new* *JTextField*();

        txtPasswordUD.*setBounds*(220, 200, 250, 40);

        txtPasswordUD.*setFont*(font16Bold);

        txtPasswordUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(txtPasswordUD);

        JLabel lblRoleUD = *new* *JLabel*("Role: ");

        lblRoleUD.*setBounds*(100, 250, 100, 40);

        lblRoleUD.*setFont*(font16Bold);

        lblRoleUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(lblRoleUD);

        JTextField txtRoleUD = *new* *JTextField*();

        txtRoleUD.*setBounds*(220, 250, 250, 40);

        txtRoleUD.*setFont*(font16Bold);

        txtRoleUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(txtRoleUD);

        JLabel lblFullnameUD = *new* *JLabel*("Fullname: ");

        lblFullnameUD.*setBounds*(100, 300, 100, 40);

        lblFullnameUD.*setFont*(font16Bold);

        lblFullnameUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(lblFullnameUD);

        JTextField txtFullnameUD = *new* *JTextField*();

        txtFullnameUD.*setBounds*(220, 300, 250, 40);

        txtFullnameUD.*setFont*(font16Bold);

        txtFullnameUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(txtFullnameUD);

        JButton btnUpdateUD = *new* *JButton*("Update");

        btnUpdateUD.*setBounds*(220, 350, 100, 30);

        btnUpdateUD.*setFont*(font16Bold);

        btnUpdateUD.*setForeground*(fontColor);

        updateDeletePanel.*add*(btnUpdateUD);

        JButton btnDeleteUD = *new* *JButton*("Delete");

        btnDeleteUD.*setBounds*(370, 350, 100, 30);

        btnDeleteUD.*setFont*(font16Bold);

        btnDeleteUD.*setForeground*(fontColor);

        btnDeleteUD.*setBackground*(Color.*decode*("#F294AD"));

        updateDeletePanel.*add*(btnDeleteUD);

*// the last object takes up the rest of the space*

*// I created this Bummy lbl to keep the interface intact.*

        JLabel lblDummyUD = *new* *JLabel*("");

        updateDeletePanel.*add*(lblDummyUD);

*// ----------- comboPersonName add Action Listener -----------*

        comboPersonName.*addActionListener*(*new* *ActionListener*() {

            @Override

*public* void *actionPerformed*(ActionEvent e) {

*// Get the selected item from the combo box*

                String selectedPerson = (String) comboPersonName.*getSelectedItem*();

*// Check if the selected item is not the placeholder*

*if* ( ! "Select Person Name".*equals*(selectedPerson)) {

*// assign the text fields with the selected Person name data*

*try*{

*// DBConnection.connection ==> declared in the DBConnection CLASS*

                        Connection conn = (Connection) DBConnection.*connection*;

                        String sqlSelect = "SELECT \* FROM person WHERE fullname = ?";

                        PreparedStatement preparedStatement = conn.*prepareStatement* (sqlSelect);

                        preparedStatement.*setString*(1, selectedPerson);

                        ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

                            txtUsernameUD.*setText*(resultSet.*getString*("name"));

                            txtPasswordUD.*setText*(resultSet.*getString*("password"));

                            txtRoleUD.*setText*(resultSet.*getString*("role"));

                            txtFullnameUD.*setText*(resultSet.*getString*("fullname"));

                        } *else* {

*// Clear the text field if the placeholder is selected*

                            txtUsernameUD.*setText*("");

                            txtPasswordUD.*setText*("");

                            txtRoleUD.*setText*("");

                            txtFullnameUD.*setText*("");

                        }*// if(resultSet.next())*

                    }*catch*(SQLException ex) {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                Update/ Delete Form... Populate text fields\n

                                Failed to Connect\n

                                Reason:\n""" + ex.*getMessage*(),

                                "Connection Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                        } *// try catch block*

                }*// if ( ! "Select Person Name".equals(selectedItem))*

            }*// combobPersonName actionPerformed(ActionEvent e)*

        }); *// comboPersonName addActionListener(new ActionListener()*

*// ----------- comboPersonName ActionListener ENDs Here -----------*

*// ============ Update Button add Action Listener ===========*

        btnUpdateUD.*addActionListener*(*new* *ActionListener*(){

            @Override

*public* void *actionPerformed*(ActionEvent e){

*// Read all the values*

                String username = txtUsernameUD.*getText*().*trim*();

                String role = txtRoleUD.*getText*().*trim*();

                String password = txtPasswordUD.*getText*().*trim*();

                String fullname = txtFullnameUD.*getText*().*trim*();

*if*(username.*equals*("") ||

                    role.*equals*("") ||

                    password.*equals*("") ||

                    fullname.*equals*("")){

                    JOptionPane.*showMessageDialog*(addUserTab,

                    """

                    Empty Field On UPDATE.\n

                    Reason:\n At least one of the feilds is EMPTY\n """,

                    "Please, fill All the fields.",

                    JOptionPane.*ERROR\_MESSAGE*);

                }*else*{

*try*{

*// the "selectedPerson" in the combobox is the string that has not changed yet.*

                        String selectedPerson = (String) comboPersonName.*getSelectedItem*();

                        int personID = *getPersonIdByName*(selectedPerson);

*// DBConnection.connection ==> declared in the DBConnection CLASS*

                        Connection conn = (Connection) DBConnection.*connection*;

                        String sqlUpdate = "UPDATE person SET name = ?, role = ?, password = ?, fullname = ?"

                                + " WHERE person\_id = ?";

                        PreparedStatement preparedStatement = conn.*prepareStatement* (sqlUpdate);

                        preparedStatement.*setString*(1, username);

                        preparedStatement.*setString*(2, role);

                        preparedStatement.*setString*(3, password);

                        preparedStatement.*setString*(4, fullname);

                        preparedStatement.*setInt*(5, personID);

*// Execute the query*

                        int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                Person Updated\n

                                Fullname: """ + fullname,

                                "Update Done",

                                JOptionPane.*INFORMATION\_MESSAGE*);

                        } *else* {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                "Failed to update User data",

                                "Update Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                        }*// if (rowsAffected)*

                        }*catch*(SQLException ex) {

                        JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                Update Form\n

                                Failed to Connect\n

                                Reason:\n""" + ex.*getMessage*(),

                                "Connection Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                    } *// try catch block*

                }*// if(username.equals("")|| ......*

            } *// btnUpdateUD action Performed*

        });*// btnUpdateUD add action listener*

*// ============ Update Button add Action Listener ENDs here ===========*

*// ============ DELETE Button add Action Listener ===========*

        btnDeleteUD.*addActionListener*(*new* *ActionListener*(){

            @Override

*public* void *actionPerformed*(ActionEvent e){

*if*(comboPersonName.*getSelectedIndex*() == 0 ) {

                    JOptionPane.*showMessageDialog*(addUserTab,

                    """

                    No Person Selected.\n

                    Reason:\n Please Secet a person name\n """,

                    "",

                    JOptionPane.*ERROR\_MESSAGE*);

                }*else*{

*try*{

*// the "selectedPerson" in the combobox is the string that has not changed yet.*

                        String selectedPersonToDelete = (String) comboPersonName.*getSelectedItem*();

                        int personID = *getPersonIdByName*(selectedPersonToDelete);

*// DBConnection.connection ==> declared in the DBConnection CLASS*

                        Connection conn = (Connection) DBConnection.*connection*;

                        String sqlDelete = "DELETE FROM person WHERE person\_id = ?";

                        PreparedStatement preparedStatement = conn.*prepareStatement* (sqlDelete);

                        preparedStatement.*setInt*(1, personID);

*// Execute the query*

                        int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                Person Deleted\n""",

                                "DELETE Done",

                                JOptionPane.*INFORMATION\_MESSAGE*);

                            comboPersonName.*setSelectedIndex*(0); *// Set as default*

                            txtUsernameUD.*setText*("");

                            txtPasswordUD.*setText*("");

                            txtRoleUD.*setText*("");

                            txtFullnameUD.*setText*("");

                        } *else* {

                            JOptionPane.*showMessageDialog*(addUserTab,

                                "Failed to Delete User",

                                "Delete Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                        }*// if (rowsAffected)*

                        }*catch*(SQLException ex) {

                        JOptionPane.*showMessageDialog*(addUserTab,

                                """

                                Delete Form\n

                                Failed to Connect\n

                                Reason:\n""" + ex.*getMessage*(),

                                "Connection Failed",

                                JOptionPane.*ERROR\_MESSAGE*);

                    } *// try catch block*

                }*// if(comboPersonName.getSelectedIndex() ......*

            } *// btnDeleteUD action Performed*

        });*// btnDeleteUD add action listener*

*// ============ DELETE Button add Action Listener ENDs here ===========*

        updateDeleteTab.*add*(updateDeletePanel, BorderLayout.*CENTER*);

        tabbedPane.*addTab*("Update/ Delete User", updateDeleteTab);

*// Add JTabbedPane to Frame*

        frame.*add*(tabbedPane, BorderLayout.*CENTER*);

*// Add Action Listeners to Navigation Labels*

        lblDashboardTitle.*addMouseListener*(*new* java.*awt*.*event*.*MouseAdapter*() {

*public* void *mouseClicked*(java.*awt*.*event*.*MouseEvent* evt) {

                tabbedPane.*setSelectedIndex*(0);

            }

        });

        lbladdUser.*addMouseListener*(*new* java.*awt*.*event*.*MouseAdapter*() {

*public* void *mouseClicked*(java.*awt*.*event*.*MouseEvent* evt) {

                tabbedPane.*setSelectedIndex*(1);

            }

        });

        lblUupdateUser.*addMouseListener*(*new* java.*awt*.*event*.*MouseAdapter*() {

*public* void *mouseClicked*(java.*awt*.*event*.*MouseEvent* evt) {

                tabbedPane.*setSelectedIndex*(2);

            }

        });

        exitLabel.*addMouseListener*(*new* java.*awt*.*event*.*MouseAdapter*() {

*public* void *mouseClicked*(java.*awt*.*event*.*MouseEvent* evt) {

                frame.*setVisible*(false); *// hide the current frame*

*dispose*(); *// close the current frame*

*new* *LoginFrame*().*setVisible*(true); *// show the new frame*

            }

        });

*// Make the Frame Visible*

        frame.*setVisible*(true);

    }*// Dash2(String fullname, String role) Constructor*

*/\*\**

*\* 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*() {

*setDefaultCloseOperation*(javax.*swing*.*WindowConstants*.*DISPOSE\_ON\_CLOSE*);

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

*getContentPane*().*setLayout*(layout);

        layout.*setHorizontalGroup*(

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

            .*addGap*(0, 223, Short.*MAX\_VALUE*)

        );

        layout.*setVerticalGroup*(

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

            .*addGap*(0, 123, Short.*MAX\_VALUE*)

        );

*pack*();

    }*// </editor-fold>*

*/\*\**

*\** @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*(Dash2.*class*.*getName*()).*log*(java.*util*.*logging*.*Level*.*SEVERE*, null, ex);

        } *catch* (InstantiationException ex) {

            java.*util*.*logging*.*Logger*.*getLogger*(Dash2.*class*.*getName*()).*log*(java.*util*.*logging*.*Level*.*SEVERE*, null, ex);

        } *catch* (IllegalAccessException ex) {

            java.*util*.*logging*.*Logger*.*getLogger*(Dash2.*class*.*getName*()).*log*(java.*util*.*logging*.*Level*.*SEVERE*, null, ex);

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

            java.*util*.*logging*.*Logger*.*getLogger*(Dash2.*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 Dash2().setVisible(true);*

            }

        });

    }*// main()*

*// Variables declaration - do not modify*

*// End of variables declaration*

*// Method to retrieve the person\_id of a person by name*

*private* *static* int *getPersonIdByName*(String name) {

*try* {

*// DBConnection.connection ==> declared in the DBConnection CLASS*

            Connection conn = (Connection) DBConnection.*connection*;

            String sqlSelect = "SELECT person\_id FROM person WHERE fullname = ?";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sqlSelect);

            preparedStatement.*setString*(1, name);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if* (resultSet.*next*()) {

*return* resultSet.*getInt*("person\_id");

            }

        } *catch* (SQLException ex) {

            JOptionPane.*showMessageDialog*(null,

                """

                getPersonIdByName Method\n

                Coulldn't retrieve person\_id\n

                Reason:\n""" + ex.*getMessage*(),

                "Retrieve Failed",

                JOptionPane.*ERROR\_MESSAGE*);

        }

*return* -1; *// Return -1 if no ID is found*

    }

} *// class Dash2{}*

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

import *java*.*math*.*BigDecimal*;

import *java*.*math*.*RoundingMode*;

import *java*.*sql*.*\**;

import *javax*.*swing*.*\**;

*public* *class* DeliveryForm *extends* javax.swing.JFrame {

*// declare a double to save it to database*

*// declare as public here, for other objects to use it.*

*public* *static* double dblAmountDue;

*public* *static* double dlv\_fee;

*// ### Constructor polymorphism ###*

*public* *DeliveryForm*() {

*initComponents*();

    }

*public* *DeliveryForm*(double bill) {

*initComponents*();

*setDefaultCloseOperation*(DISPOSE\_ON\_CLOSE);

*setTitle*("Delivery Order");

*setLocationRelativeTo*(null); *// Center the window*

*// get the logged in person name from the SessionPerson Class*

        dblAmountDue = bill;

        lblUser.*setText*("User: " + SessionPerson.*getCurrentPersonName*());

        lblDueTotal.*setText*("Due Amount Total: " + bill);

    }

*/\*\**

*\* 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*();

        lblUser = *new* javax.*swing*.*JLabel*();

        lblDueTotal = *new* javax.*swing*.*JLabel*();

        jLabel1 = *new* javax.*swing*.*JLabel*();

        txtDeliveryFee = *new* javax.*swing*.*JTextField*();

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

        txtCustomerName = *new* javax.*swing*.*JTextField*();

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

        txtCustomerAddress = *new* javax.*swing*.*JTextField*();

        txtCustomerTel = *new* javax.*swing*.*JTextField*();

        jLabel4 = *new* javax.*swing*.*JLabel*();

        btnDone = *new* javax.*swing*.*JButton*();

        btnCancel = *new* javax.*swing*.*JButton*();

*setDefaultCloseOperation*(javax.*swing*.*WindowConstants*.*EXIT\_ON\_CLOSE*);

*setMinimumSize*(*new* java.*awt*.*Dimension*(745, 584));

*setPreferredSize*(*new* java.*awt*.*Dimension*(745, 584));

*getContentPane*().*setLayout*(*new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteLayout*());

        jPanel1.*setBackground*(*new* java.*awt*.*Color*(234, 233, 231));

        jPanel1.*setLayout*(*new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteLayout*());

        lblUser.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        lblUser.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        lblUser.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        lblUser.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*LOWERED*));

        jPanel1.*add*(lblUser, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(110, 30, 490, 50));

        lblDueTotal.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        lblDueTotal.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        lblDueTotal.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        lblDueTotal.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*LOWERED*));

        jPanel1.*add*(lblDueTotal, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(110, 330, 490, 50));

        jLabel1.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel1.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel1.*setText*("Delivery Fee");

        jPanel1.*add*(jLabel1, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(120, 110, 160, 30));

        txtDeliveryFee.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtDeliveryFee.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        txtDeliveryFee.*addFocusListener*(*new* java.*awt*.*event*.*FocusAdapter*() {

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

*txtDeliveryFeeFocusLost*(evt);

            }

        });

        txtDeliveryFee.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*txtDeliveryFeeActionPerformed*(evt);

            }

        });

        jPanel1.*add*(txtDeliveryFee, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(290, 110, 290, 30));

        jLabel2.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel2.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel2.*setText*("Customer Name");

        jPanel1.*add*(jLabel2, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(120, 160, 150, 30));

        txtCustomerName.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtCustomerName.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jPanel1.*add*(txtCustomerName, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(290, 160, 290, 30));

        jLabel3.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 17)); *// NOI18N*

        jLabel3.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel3.*setText*("Customer Address");

        jPanel1.*add*(jLabel3, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(120, 210, 150, 20));

        txtCustomerAddress.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtCustomerAddress.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jPanel1.*add*(txtCustomerAddress, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(290, 210, 290, 30));

        txtCustomerTel.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        txtCustomerTel.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jPanel1.*add*(txtCustomerTel, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(290, 260, 290, 30));

        jLabel4.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel4.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        jLabel4.*setText*("Customer Tel");

        jPanel1.*add*(jLabel4, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(120, 260, 150, 30));

        btnDone.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnDone.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        btnDone.*setText*("Done");

        btnDone.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnDoneActionPerformed*(evt);

            }

        });

        jPanel1.*add*(btnDone, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(180, 410, 140, 40));

        btnCancel.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCancel.*setForeground*(*new* java.*awt*.*Color*(64, 31, 52));

        btnCancel.*setText*("Cancel");

        btnCancel.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCancelActionPerformed*(evt);

            }

        });

        jPanel1.*add*(btnCancel, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(410, 410, 140, 40));

*getContentPane*().*add*(jPanel1, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(0, 0, 730, 530));

*pack*();

    }*// </editor-fold>*

*private* void *txtDeliveryFeeActionPerformed*(java.awt.event.ActionEvent evt) {

*// in case user pressed Enter Key*

*if* ( ! txtDeliveryFee.*getText*().*equals*("")){

            dlv\_fee = Double.*parseDouble*(txtDeliveryFee.*getText*());

            double totalFee = dblAmountDue + dlv\_fee;

            lblDueTotal.*setText*(Double.*toString*(totalFee));

        }

    }

*private* void *btnCancelActionPerformed*(java.awt.event.ActionEvent evt) {

*// close the Credit Card frame,*

*// then the menu will appear*

        this.*dispose*();

    }

*private* void *txtDeliveryFeeFocusLost*(java.awt.event.FocusEvent evt) {

*// in case the user pressed TAB, or Clicked the mouse.*

*if* ( ! txtDeliveryFee.*getText*().*equals*("")){

            dlv\_fee = Double.*parseDouble*(txtDeliveryFee.*getText*());

            double totalFee = dblAmountDue + dlv\_fee;

            lblDueTotal.*setText*(Double.*toString*(totalFee));

        }

    }

*private* void *btnDoneActionPerformed*(java.awt.event.ActionEvent evt) {

*try*{

*// prepare the values*

*// get the current Person ID from SessionPerson Class*

            int current\_person\_id = SessionPerson.*getCurrentPersonID*();

*// dlv\_fee is calculated up above*

            String customerName = txtCustomerName.*getText*();

            String customerAddress = txtCustomerAddress.*getText*();

            String customerTel = txtCustomerTel.*getText*();

*// dblAmountDue AND dlv\_fee are public variables*

            double totalFee = dblAmountDue + dlv\_fee;

*// go through 3 steps to covert a double to BigDecimal then back to double*

*// Convert the double to BigDecimal*

            BigDecimal twoDecimal\_iTotal = *new* *BigDecimal*(totalFee);

*// Set the scale to 2 decimal places with rounding*

            BigDecimal formatted\_iTotal = twoDecimal\_iTotal.*setScale*(2, RoundingMode.*HALF\_UP*);

*// Convert back to double to display it.*

            totalFee = formatted\_iTotal.*doubleValue*();

*// DBConnection.connection ==> declared in the DBConnection CLASS*

            Connection conn = (Connection) DBConnection.*connection*;

            int quantity = SessionPerson.*getTableRowsCount*();

            String sqlInsertOrders = "INSERT INTO orders (user\_id, item\_id, type, quantity, total\_cost)"

            + " VALUES (?,?,?,?,?)";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sqlInsertOrders);

            preparedStatement.*setInt*(1, current\_person\_id); *// for customer column too*

            preparedStatement.*setInt*(2, current\_person\_id);

            preparedStatement.*setString*(3, "Takeaway");

            preparedStatement.*setInt*(4, quantity);

            preparedStatement.*setDouble*(5, totalFee);

            int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                JOptionPane.*showMessageDialog*(this,

                    "Order Saved in ORDERS Successfully",

                    "Save Done",

                    JOptionPane.*INFORMATION\_MESSAGE*);

            } *else* {

                JOptionPane.*showMessageDialog*(this,

                    "Failed to save in ORDERS table",

                    "Save Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            }*// if (rowsAffected)*

            String sqlInsertTakeaway = "INSERT INTO takeaway (customer\_id, order\_id, delivery\_fees, customer\_name, "

                    + "customer\_addr, customer\_tel, amount\_due) "

                    + "VALUES (?,?,?,?,?,?,?)";

            PreparedStatement ps = conn.*prepareStatement* (sqlInsertTakeaway);

            ps.*setInt*(1, current\_person\_id); *// for customer column too*

            ps.*setInt*(2, current\_person\_id);

            ps.*setDouble*(3, dlv\_fee);

            ps.*setString*(4, customerName);

            ps.*setString*(5, customerAddress);

            ps.*setString*(6, customerTel);

            ps.*setDouble*(7, totalFee);

*// Execute the query*

            rowsAffected = ps.*executeUpdate*();

*if* (rowsAffected > 0) {

                JOptionPane.*showMessageDialog*(this,

                    "Takeaway Order Saved Successfully",

                    "Save Done",

                    JOptionPane.*INFORMATION\_MESSAGE*);

            } *else* {

                JOptionPane.*showMessageDialog*(this,

                    "Failed to save Takeaway Order Information",

                    "Save Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            }*// if (rowsAffected)*

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed to Connect

                    \nReason:\nDelivery Form\n""" + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            e.*printStackTrace*();

        } *// try catch block*

*// save to database then close the Credit Card frame,*

*// then the menu will appear to input a new order*

        this.*dispose*();

    }

*/\*\**

*\** @param *args the command line arguments*

*\*/*

*public* *static* void *main*(String args[]) {

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

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

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

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

*\*/*

*try* {

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

        } *catch* (InstantiationException ex) {

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

        } *catch* (IllegalAccessException ex) {

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

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

            java.*util*.*logging*.*Logger*.*getLogger*(DeliveryForm.*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* *DeliveryForm*().*setVisible*(true);

            }

        });

    }

*// Variables declaration - do not modify*

*private* javax.swing.JButton btnCancel;

*private* javax.swing.JButton btnDone;

*private* javax.swing.JLabel jLabel1;

*private* javax.swing.JLabel jLabel2;

*private* javax.swing.JLabel jLabel3;

*private* javax.swing.JLabel jLabel4;

*private* javax.swing.JPanel jPanel1;

*private* javax.swing.JLabel lblDueTotal;

*private* javax.swing.JLabel lblUser;

*private* javax.swing.JTextField txtCustomerAddress;

*private* javax.swing.JTextField txtCustomerName;

*private* javax.swing.JTextField txtCustomerTel;

*private* javax.swing.JTextField txtDeliveryFee;

*// End of variables declaration*

}

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

import *javax*.*swing*.*\**;

import *java*.*sql*.*\**;

import *java*.*util*.*logging*.*Level*;

import *java*.*util*.*logging*.*Logger*;

*public* *class* LoginFrame *extends* javax.swing.JFrame {

*public* *LoginFrame*() {

*setDefaultCloseOperation*(DISPOSE\_ON\_CLOSE);

*setTitle*("Login Form");

*initComponents*();

*setLocationRelativeTo*(null); *// Center the window*

    }

*/\*\**

*\* This method is called from within the constructor to initialize the form.*

*\* WARNING: Do NOT modify this code. The content of this method is always*

*\* regenerated by the Form Editor.*

*\*/*

    @SuppressWarnings("unchecked")

*// <editor-fold defaultstate="collapsed" desc="Generated Code">*

*private* void *initComponents*() {

        jPanel1 = *new* javax.*swing*.*JPanel*();

        jLabel1 = *new* javax.*swing*.*JLabel*();

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

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

        jPanel2 = *new* javax.*swing*.*JPanel*();

        jLabel4 = *new* javax.*swing*.*JLabel*();

        jLabel5 = *new* javax.*swing*.*JLabel*();

        jLabel6 = *new* javax.*swing*.*JLabel*();

        txtUsername = *new* javax.*swing*.*JTextField*();

        txtPassword = *new* javax.*swing*.*JPasswordField*();

        btnCancel = *new* javax.*swing*.*JButton*();

        btnLogin = *new* javax.*swing*.*JButton*();

*setDefaultCloseOperation*(javax.*swing*.*WindowConstants*.*EXIT\_ON\_CLOSE*);

*setBackground*(*new* java.*awt*.*Color*(45, 59, 64));

*setMinimumSize*(*new* java.*awt*.*Dimension*(745, 584));

        jPanel1.*setBackground*(*new* java.*awt*.*Color*(219, 191, 170));

        jLabel1.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/LogoGreen.png"))); *// NOI18N*

        jLabel1.*setToolTipText*("");

        jLabel2.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 36)); *// NOI18N*

        jLabel2.*setForeground*(*new* java.*awt*.*Color*(45, 59, 64));

        jLabel2.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        jLabel2.*setText*("ALDIWAN");

        jLabel2.*setHorizontalTextPosition*(javax.*swing*.*SwingConstants*.*LEADING*);

        jLabel3.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 36)); *// NOI18N*

        jLabel3.*setForeground*(*new* java.*awt*.*Color*(45, 59, 64));

        jLabel3.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        jLabel3.*setText*("Restaurant");

        javax.swing.GroupLayout jPanel1Layout = *new* javax.*swing*.*GroupLayout*(jPanel1);

        jPanel1.*setLayout*(jPanel1Layout);

        jPanel1Layout.*setHorizontalGroup*(

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

            .*addGroup*(jPanel1Layout.*createSequentialGroup*()

                .*addGroup*(jPanel1Layout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*CENTER*)

                    .*addComponent*(jLabel2, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 209, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(jLabel3, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 220, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addGap*(0, 6, Short.*MAX\_VALUE*))

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

                .*addContainerGap*(javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                .*addComponent*(jLabel1, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 169, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                .*addContainerGap*(javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

        );

        jPanel1Layout.*setVerticalGroup*(

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

            .*addGroup*(jPanel1Layout.*createSequentialGroup*()

                .*addGap*(59, 59, 59)

                .*addComponent*(jLabel1, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 201, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                .*addGap*(57, 57, 57)

                .*addComponent*(jLabel2, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                .*addGap*(18, 18, 18)

                .*addComponent*(jLabel3, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                .*addGap*(83, 83, 83))

        );

        jPanel2.*setBackground*(*new* java.*awt*.*Color*(217, 206, 197));

        jLabel4.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 24)); *// NOI18N*

        jLabel4.*setForeground*(*new* java.*awt*.*Color*(89, 74, 60));

        jLabel4.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        jLabel4.*setText*("Username");

        jLabel4.*setHorizontalTextPosition*(javax.*swing*.*SwingConstants*.*LEADING*);

        jLabel5.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 24)); *// NOI18N*

        jLabel5.*setForeground*(*new* java.*awt*.*Color*(89, 74, 60));

        jLabel5.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        jLabel5.*setText*("Password");

        jLabel6.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 36)); *// NOI18N*

        jLabel6.*setForeground*(*new* java.*awt*.*Color*(89, 74, 60));

        jLabel6.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        jLabel6.*setText*("Login");

        jLabel6.*setHorizontalTextPosition*(javax.*swing*.*SwingConstants*.*LEADING*);

        txtUsername.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 0, 24)); *// NOI18N*

        txtUsername.*setForeground*(*new* java.*awt*.*Color*(45, 59, 64));

        txtPassword.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 0, 18)); *// NOI18N*

        btnCancel.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCancel.*setForeground*(*new* java.*awt*.*Color*(45, 59, 64));

        btnCancel.*setText*("Exit");

        btnCancel.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCancelActionPerformed*(evt);

            }

        });

        btnLogin.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnLogin.*setForeground*(*new* java.*awt*.*Color*(45, 59, 64));

        btnLogin.*setText*("Login");

        btnLogin.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnLoginActionPerformed*(evt);

            }

        });

        javax.swing.GroupLayout jPanel2Layout = *new* javax.*swing*.*GroupLayout*(jPanel2);

        jPanel2.*setLayout*(jPanel2Layout);

        jPanel2Layout.*setHorizontalGroup*(

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

            .*addGroup*(jPanel2Layout.*createSequentialGroup*()

                .*addContainerGap*()

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

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

                        .*addGap*(0, 189, Short.*MAX\_VALUE*)

                        .*addComponent*(jLabel6, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 132, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                        .*addGap*(150, 150, 150))

                    .*addGroup*(jPanel2Layout.*createSequentialGroup*()

                        .*addGroup*(jPanel2Layout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*TRAILING*)

                            .*addGroup*(jPanel2Layout.*createSequentialGroup*()

                                .*addComponent*(btnLogin, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 90, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                                .*addGap*(39, 39, 39)

                                .*addComponent*(btnCancel, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 102, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                            .*addGroup*(jPanel2Layout.*createSequentialGroup*()

                                .*addGroup*(jPanel2Layout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*CENTER*)

                                    .*addComponent*(jLabel4, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 138, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                                    .*addComponent*(jLabel5, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 138, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

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

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

                                    .*addComponent*(txtPassword)

                                    .*addComponent*(txtUsername, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 190, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))))

                        .*addContainerGap*(javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))))

        );

        jPanel2Layout.*setVerticalGroup*(

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

            .*addGroup*(jPanel2Layout.*createSequentialGroup*()

                .*addGap*(199, 199, 199)

                .*addComponent*(jLabel6, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, 48, Short.*MAX\_VALUE*)

                .*addGap*(18, 18, 18)

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

                    .*addComponent*(jLabel4, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                    .*addComponent*(txtUsername, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 32, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addGap*(18, 18, 18)

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

                    .*addComponent*(jLabel5, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                    .*addComponent*(txtPassword, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 31, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addGap*(59, 59, 59)

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

                    .*addComponent*(btnCancel)

                    .*addComponent*(btnLogin))

                .*addGap*(118, 118, 118))

        );

        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*.*PREFERRED\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

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

                .*addComponent*(jPanel2, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                .*addContainerGap*(javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

        );

        layout.*setVerticalGroup*(

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

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

                .*addContainerGap*()

                .*addGroup*(layout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*TRAILING*)

                    .*addComponent*(jPanel2, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                    .*addComponent*(jPanel1, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

                .*addContainerGap*())

        );

*pack*();

    }*// </editor-fold>*

*private* void *btnLoginActionPerformed*(java.awt.event.ActionEvent evt) {

        String strUsername = txtUsername.*getText*();

        String strPassword = txtPassword.*getText*();

*try*{

            Connection conn = DBConnection.*connection*;

*// this sql statement requires 2 parameters in place of "?"*

            String sql = "SELECT \* FROM person WHERE name=? AND password=?";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            preparedStatement.*setString*(1, strUsername);

            preparedStatement.*setString*(2, strPassword);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*// declare a logedPerson object from the Person class*

            PersonClass loggedPerson = *new* *PersonClass*();

*if*(resultSet.*next*()){

*// loggedPerson.property = resultSet.getString("table field title")*

                loggedPerson.*name* = resultSet.*getString*("name");

                loggedPerson.*role* = resultSet.*getString*("role").*trim*().*toLowerCase*();

                loggedPerson.*password* = resultSet.*getString*("password");

                loggedPerson.*fullName* = resultSet.*getString*("fullname");

                SessionPerson.*setCurrentPersonID*(resultSet.*getInt*("person\_id"));

                SessionPerson.*setCurrentPersonName*(resultSet.*getString*("fullname"));

                this.*dispose*(); *// Close the login window*

*if* (loggedPerson.*role*.*equals*("manager")){

*// pass the name and role to Dash2 window through the constructor*

                    Dash2 dash2 = *new* *Dash2*(loggedPerson.*fullName*, loggedPerson.*role*);

*// Open the dashboard window*

                    dash2.*setVisible*(true);

                }*else* *if*(loggedPerson.*role*.*equals*("user")){

                    Menu menu = *new* *Menu*(loggedPerson.*fullName*, loggedPerson.*role*);

                    menu.*setVisible*(true);

                }*// end of if (loggedPerson.role.equals("manager"))*

            }*else*{

                JOptionPane.*showMessageDialog*(this,

                    "Can't find user",

                    "NOT Authenticated",

                    JOptionPane.*ERROR\_MESSAGE*);

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed to Connect.

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

    }

*private* void *btnCancelActionPerformed*(java.awt.event.ActionEvent evt) {

*// close the database connection then Exit, terminate the Application.*

*try* {

            DBConnection.*connection*.*close*();

        } *catch* (SQLException ex) {

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

        }

        System.*exit*(0); *// terminate the Application*

    }

*/\*\**

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

        } *catch* (InstantiationException ex) {

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

        } *catch* (IllegalAccessException ex) {

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

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

            java.*util*.*logging*.*Logger*.*getLogger*(LoginFrame.*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* *LoginFrame*().*setVisible*(true);

            }

        });

    }

*// Variables declaration - do not modify*

*private* javax.swing.JButton btnCancel;

*private* javax.swing.JButton btnLogin;

*private* javax.swing.JLabel jLabel1;

*private* javax.swing.JLabel jLabel2;

*private* javax.swing.JLabel jLabel3;

*private* javax.swing.JLabel jLabel4;

*private* javax.swing.JLabel jLabel5;

*private* javax.swing.JLabel jLabel6;

*private* javax.swing.JPanel jPanel1;

*private* javax.swing.JPanel jPanel2;

*private* javax.swing.JPasswordField txtPassword;

*private* javax.swing.JTextField txtUsername;

*// End of variables declaration*

}

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

import *java*.*awt*.*BorderLayout*;

import *java*.*awt*.*Dimension*;

import *java*.*sql*.*Connection*;

import *java*.*sql*.*PreparedStatement*;

import *java*.*sql*.*ResultSet*;

import *java*.*sql*.*SQLException*;

import *java*.*math*.*\**;

import *javax*.*swing*.*JOptionPane*;

import *javax*.*swing*.*table*.*DefaultTableModel*;

*public* *class* Menu *extends* javax.swing.JFrame {

*// declare public String fullNamePub, to be able to use it inside other objects*

*public* *static* String fullNamePub="";

*// create a table model as instance variable, so it can be reached from all objects*

*private* *final* DefaultTableModel model;

*public* *Menu*(String fullname, String role) {

*initComponents*();

*setDefaultCloseOperation*(DISPOSE\_ON\_CLOSE);

*setExtendedState*(Menu.*MAXIMIZED\_BOTH*); *// Maximize the window*

*setLocationRelativeTo*(null);

*setTitle*("ALDIWAN Restaurant Menu");

        lblMenuTitle.*setText*("Hello: " + fullname);

        fullNamePub = fullname;

        mainContentPanel.*setLayout*(*new* *BorderLayout*()); *// Use BorderLayout for resizing*

*setContentPane*(mainContentPanel);

*// Allow the inner jPanels to resize inside the outer jPanel*

        mainContentPanel.*add*(usernamePanel, BorderLayout.*NORTH*);

        mainContentPanel.*add*(contentsPanel, BorderLayout.*CENTER*);

*// set the panels widths to 60% and 40%, and respond to window sizing*

        menuItemsPanel.*setPreferredSize*(*new* *Dimension*((int)

                (contentsPanel.*getWidth*() \* 0.6), contentsPanel.*getHeight*()));

        pricesPanel.*setPreferredSize*(*new* *Dimension*((int)

                (contentsPanel.*getWidth*() \* 0.4), contentsPanel.*getHeight*()));

*// declare the table model inside the constructor.*

*// so it can be accessed from all buttons*

        model = (DefaultTableModel) reciptModel.*getModel*();

    } *// Menu() Constructor*

*/\*\**

*\* 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*() {

        mainContentPanel = *new* javax.*swing*.*JPanel*();

        usernamePanel = *new* javax.*swing*.*JPanel*();

        lblMenuTitle = *new* javax.*swing*.*JLabel*();

        contentsPanel = *new* javax.*swing*.*JPanel*();

        menuItemsPanel = *new* javax.*swing*.*JPanel*();

        PizzaHeart = *new* javax.*swing*.*JButton*();

        doubleBurger = *new* javax.*swing*.*JButton*();

        burger = *new* javax.*swing*.*JButton*();

        phileCheeseSteak = *new* javax.*swing*.*JButton*();

        phileDouble = *new* javax.*swing*.*JButton*();

        phileLong = *new* javax.*swing*.*JButton*();

        cola2Leter = *new* javax.*swing*.*JButton*();

        cola1Leter = *new* javax.*swing*.*JButton*();

        colaCan = *new* javax.*swing*.*JButton*();

        frise = *new* javax.*swing*.*JButton*();

        pastaPan = *new* javax.*swing*.*JButton*();

        spagetti = *new* javax.*swing*.*JButton*();

        btnExit = *new* javax.*swing*.*JButton*();

        btnNewReceipt = *new* javax.*swing*.*JButton*();

        pricesPanel = *new* javax.*swing*.*JPanel*();

        jScrollPane1 = *new* javax.*swing*.*JScrollPane*();

        reciptModel = *new* javax.*swing*.*JTable*();

        jLabel1 = *new* javax.*swing*.*JLabel*();

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

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

        txtSubTotal = *new* javax.*swing*.*JTextField*();

        txtTax = *new* javax.*swing*.*JTextField*();

        txtTotal = *new* javax.*swing*.*JTextField*();

        btnCash = *new* javax.*swing*.*JButton*();

        btnDelivery = *new* javax.*swing*.*JButton*();

        btnCredit = *new* javax.*swing*.*JButton*();

*setDefaultCloseOperation*(javax.*swing*.*WindowConstants*.*EXIT\_ON\_CLOSE*);

        usernamePanel.*setBackground*(*new* java.*awt*.*Color*(89, 44, 73));

        lblMenuTitle.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 24)); *// NOI18N*

        lblMenuTitle.*setForeground*(*new* java.*awt*.*Color*(234, 233, 231));

        lblMenuTitle.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*CENTER*);

        lblMenuTitle.*setText*("username");

        javax.swing.GroupLayout usernamePanelLayout = *new* javax.*swing*.*GroupLayout*(usernamePanel);

        usernamePanel.*setLayout*(usernamePanelLayout);

        usernamePanelLayout.*setHorizontalGroup*(

            usernamePanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addComponent*(lblMenuTitle, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

        );

        usernamePanelLayout.*setVerticalGroup*(

            usernamePanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addGroup*(usernamePanelLayout.*createSequentialGroup*()

                .*addContainerGap*()

                .*addComponent*(lblMenuTitle)

                .*addContainerGap*(javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

        );

        contentsPanel.*setBackground*(*new* java.*awt*.*Color*(201, 141, 117));

        menuItemsPanel.*setBackground*(*new* java.*awt*.*Color*(217, 193, 180));

        PizzaHeart.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/pizza5Heart.jpg"))); *// NOI18N*

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

        PizzaHeart.*setContentAreaFilled*(false);

        PizzaHeart.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*PizzaHeartActionPerformed*(evt);

            }

        });

        doubleBurger.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/b2double.jpg"))); *// NOI18N*

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

        doubleBurger.*setContentAreaFilled*(false);

        doubleBurger.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*doubleBurgerActionPerformed*(evt);

            }

        });

        burger.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/burger2.png"))); *// NOI18N*

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

        burger.*setContentAreaFilled*(false);

        burger.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*burgerActionPerformed*(evt);

            }

        });

        phileCheeseSteak.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/Cheesesteak1.jpg"))); *// NOI18N*

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

        phileCheeseSteak.*setContentAreaFilled*(false);

        phileCheeseSteak.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*phileCheeseSteakActionPerformed*(evt);

            }

        });

        phileDouble.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/Cheesesteak2.jpeg"))); *// NOI18N*

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

        phileDouble.*setContentAreaFilled*(false);

        phileDouble.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*phileDoubleActionPerformed*(evt);

            }

        });

        phileLong.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/Cheesesteak3.jpg"))); *// NOI18N*

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

        phileLong.*setContentAreaFilled*(false);

        phileLong.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*phileLongActionPerformed*(evt);

            }

        });

        cola2Leter.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/colaBottle.jpg"))); *// NOI18N*

        cola2Leter.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*RAISED*));

        cola2Leter.*setContentAreaFilled*(false);

        cola2Leter.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*cola2LeterActionPerformed*(evt);

            }

        });

        cola1Leter.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/colaBottle200.jpg"))); *// NOI18N*

        cola1Leter.*setBorder*(javax.*swing*.*BorderFactory*.*createBevelBorder*(javax.*swing*.*border*.*BevelBorder*.*RAISED*));

        cola1Leter.*setContentAreaFilled*(false);

        cola1Leter.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*cola1LeterActionPerformed*(evt);

            }

        });

        colaCan.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/colaCan.jpg"))); *// NOI18N*

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

        colaCan.*setContentAreaFilled*(false);

        colaCan.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*colaCanActionPerformed*(evt);

            }

        });

        frise.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/ff.png"))); *// NOI18N*

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

        frise.*setContentAreaFilled*(false);

        frise.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*friseActionPerformed*(evt);

            }

        });

        pastaPan.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/pasta1 pan.jpg"))); *// NOI18N*

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

        pastaPan.*setContentAreaFilled*(false);

        pastaPan.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*pastaPanActionPerformed*(evt);

            }

        });

        spagetti.*setIcon*(*new* javax.*swing*.*ImageIcon*(*getClass*().*getResource*("/aldiwan/images/pasta2.jpg"))); *// NOI18N*

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

        spagetti.*setContentAreaFilled*(false);

        spagetti.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*spagettiActionPerformed*(evt);

            }

        });

        btnExit.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnExit.*setText*("Exit");

        btnExit.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnExitActionPerformed*(evt);

            }

        });

        btnNewReceipt.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnNewReceipt.*setText*("New Receipt");

        btnNewReceipt.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnNewReceiptActionPerformed*(evt);

            }

        });

        javax.swing.GroupLayout menuItemsPanelLayout = *new* javax.*swing*.*GroupLayout*(menuItemsPanel);

        menuItemsPanel.*setLayout*(menuItemsPanelLayout);

        menuItemsPanelLayout.*setHorizontalGroup*(

            menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addGroup*(menuItemsPanelLayout.*createSequentialGroup*()

                .*addContainerGap*()

                .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                    .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                        .*addGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*TRAILING*, menuItemsPanelLayout.*createSequentialGroup*()

                            .*addComponent*(phileDouble, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(phileLong, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(cola2Leter, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(cola1Leter, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                        .*addGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*TRAILING*, menuItemsPanelLayout.*createSequentialGroup*()

                            .*addComponent*(colaCan, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(frise, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(pastaPan, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(18, 18, 18)

                            .*addComponent*(spagetti, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                        .*addGroup*(menuItemsPanelLayout.*createSequentialGroup*()

                            .*addGap*(55, 55, 55)

                            .*addComponent*(btnExit, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 98, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                            .*addGap*(66, 66, 66)

                            .*addComponent*(btnNewReceipt, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 154, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)))

                    .*addGroup*(menuItemsPanelLayout.*createSequentialGroup*()

                        .*addComponent*(PizzaHeart, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                        .*addGap*(18, 18, 18)

                        .*addComponent*(doubleBurger, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                        .*addGap*(18, 18, 18)

                        .*addComponent*(burger, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                        .*addGap*(18, 18, 18)

                        .*addComponent*(phileCheeseSteak, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 227, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)))

                .*addContainerGap*(171, Short.*MAX\_VALUE*))

        );

        menuItemsPanelLayout.*setVerticalGroup*(

            menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addGroup*(menuItemsPanelLayout.*createSequentialGroup*()

                .*addContainerGap*()

                .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                    .*addComponent*(phileCheeseSteak, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(burger, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(doubleBurger, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(PizzaHeart, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addGap*(18, 18, 18)

                .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                    .*addComponent*(cola1Leter, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(cola2Leter, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(phileLong, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(phileDouble, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addGap*(21, 21, 21)

                .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                    .*addComponent*(spagetti, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(pastaPan, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(frise, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(colaCan, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 221, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addPreferredGap*(javax.*swing*.*LayoutStyle*.*ComponentPlacement*.*RELATED*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

                .*addGroup*(menuItemsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

                    .*addComponent*(btnExit, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 41, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                    .*addComponent*(btnNewReceipt, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, 41, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*))

                .*addContainerGap*())

        );

        pricesPanel.*setBackground*(*new* java.*awt*.*Color*(234, 233, 231));

        pricesPanel.*setLayout*(*new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteLayout*());

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

        reciptModel.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 14)); *// NOI18N*

        reciptModel.*setModel*(*new* javax.*swing*.*table*.*DefaultTableModel*(

*new* Object [][] {

            },

*new* String [] {

                "Item Name", "Price"

            }

        ));

        reciptModel.*setAlignmentX*(1.0F);

        reciptModel.*setAlignmentY*(1.0F);

        jScrollPane1.*setViewportView*(reciptModel);

        pricesPanel.*add*(jScrollPane1, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(12, 12, 440, 488));

        jLabel1.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel1.*setForeground*(*new* java.*awt*.*Color*(89, 44, 73));

        jLabel1.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*RIGHT*);

        jLabel1.*setText*("Sub Total");

        pricesPanel.*add*(jLabel1, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(70, 530, 115, 44));

        jLabel2.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel2.*setForeground*(*new* java.*awt*.*Color*(89, 44, 73));

        jLabel2.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*RIGHT*);

        jLabel2.*setText*("Tax");

        pricesPanel.*add*(jLabel2, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(70, 590, 115, 44));

        jLabel3.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        jLabel3.*setForeground*(*new* java.*awt*.*Color*(89, 44, 73));

        jLabel3.*setHorizontalAlignment*(javax.*swing*.*SwingConstants*.*RIGHT*);

        jLabel3.*setText*("Total");

        pricesPanel.*add*(jLabel3, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(70, 650, 115, 44));

        txtSubTotal.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        pricesPanel.*add*(txtSubTotal, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(200, 530, 192, 44));

        txtTax.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        pricesPanel.*add*(txtTax, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(200, 590, 192, 44));

        txtTotal.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        pricesPanel.*add*(txtTotal, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(200, 650, 192, 50));

        btnCash.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCash.*setText*("Cash");

        btnCash.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCashActionPerformed*(evt);

            }

        });

        pricesPanel.*add*(btnCash, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(20, 720, 140, -1));

        btnDelivery.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnDelivery.*setText*("Delivery");

        btnDelivery.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnDeliveryActionPerformed*(evt);

            }

        });

        pricesPanel.*add*(btnDelivery, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(410, 720, 120, -1));

        btnCredit.*setFont*(*new* java.*awt*.*Font*("Dialog.plain", 1, 18)); *// NOI18N*

        btnCredit.*setText*("Credit");

        btnCredit.*addActionListener*(*new* java.*awt*.*event*.*ActionListener*() {

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

*btnCreditActionPerformed*(evt);

            }

        });

        pricesPanel.*add*(btnCredit, *new* org.*netbeans*.*lib*.*awtextra*.*AbsoluteConstraints*(210, 720, 150, -1));

        javax.swing.GroupLayout contentsPanelLayout = *new* javax.*swing*.*GroupLayout*(contentsPanel);

        contentsPanel.*setLayout*(contentsPanelLayout);

        contentsPanelLayout.*setHorizontalGroup*(

            contentsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addGroup*(contentsPanelLayout.*createSequentialGroup*()

                .*addComponent*(menuItemsPanel, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

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

                .*addComponent*(pricesPanel, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))

        );

        contentsPanelLayout.*setVerticalGroup*(

            contentsPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addComponent*(menuItemsPanel, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

            .*addComponent*(pricesPanel, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

        );

        javax.swing.GroupLayout mainContentPanelLayout = *new* javax.*swing*.*GroupLayout*(mainContentPanel);

        mainContentPanel.*setLayout*(mainContentPanelLayout);

        mainContentPanelLayout.*setHorizontalGroup*(

            mainContentPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addComponent*(usernamePanel, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

            .*addComponent*(contentsPanel, javax.*swing*.*GroupLayout*.*Alignment*.*TRAILING*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

        );

        mainContentPanelLayout.*setVerticalGroup*(

            mainContentPanelLayout.*createParallelGroup*(javax.*swing*.*GroupLayout*.*Alignment*.*LEADING*)

            .*addGroup*(mainContentPanelLayout.*createSequentialGroup*()

                .*addComponent*(usernamePanel, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

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

                .*addComponent*(contentsPanel, 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*()

                .*addComponent*(mainContentPanel, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*PREFERRED\_SIZE*)

                .*addGap*(0, 0, Short.*MAX\_VALUE*))

        );

        layout.*setVerticalGroup*(

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

            .*addComponent*(mainContentPanel, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, javax.*swing*.*GroupLayout*.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

        );

*pack*();

*setLocationRelativeTo*(null);

    }*// </editor-fold>*

*private* void *btnExitActionPerformed*(java.awt.event.ActionEvent evt) {

        this.*dispose*(); *// close the current frame.*

*new* *LoginFrame*().*setVisible*(true); *// show the new frame*

    }

*private* void *PizzaHeartActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

*try*{

            Connection conn = DBConnection.*connection*;

*// item\_id=1 ==> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 1";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// Call function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *btnNewReceiptActionPerformed*(java.awt.event.ActionEvent evt) {

        model.*setRowCount*(0);

        txtSubTotal.*setText*("");

        txtTax.*setText*("");

        txtTotal.*setText*("");

    }

*private* void *doubleBurgerActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 2 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 2";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *burgerActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 3 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 3";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *phileCheeseSteakActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 4 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 4";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *phileDoubleActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 5 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 5";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *phileLongActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 6 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 6";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *cola2LeterActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 7 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 7";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *cola1LeterActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 8 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 8";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *colaCanActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 9 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 9";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *friseActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 10 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 10";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *pastaPanActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 11 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 11";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *spagettiActionPerformed*(java.awt.event.ActionEvent evt) {

        String itemName = "";

        String itemPrice = "";

*// DBConnection.connection ==> declared in the DBConnection CLASS*

        Connection conn = DBConnection.*connection*;

*try*{

*// item\_id = 12 ===> first record in the menu\_items table.*

            String sql = "SELECT \* FROM menu\_items WHERE item\_id = 12";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sql);

            ResultSet resultSet = preparedStatement.*executeQuery*();

*if*(resultSet.*next*()){

*// variable\_name = resultSet.getString("table field title")*

                itemName = resultSet.*getString*("name");

                itemPrice = resultSet.*getString*("price");

            }

        }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed Connect

                    \nReason:\n Login Frame Class\n """ + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

        } *// try catch block*

*// insert add a new row in the table 2 columns.*

        model.*addRow*(*new* Object[]{itemName, itemPrice});

*// function to calculate receipt subtotal, tax and total.*

*receipt*();

    }

*private* void *btnCashActionPerformed*(java.awt.event.ActionEvent evt) {

        int current\_person\_id = SessionPerson.*getCurrentPersonID*();

        double totalCost=*receipt*();

        SessionPerson.*setTableRowsCount*(model.*getRowCount*());  *// Directly from the JTable*

*try*{

*// DBConnection.connection ==> declared in the DBConnection CLASS*

            Connection conn = (Connection) DBConnection.*connection*;

            String sqlInsert = "INSERT INTO orders (user\_id, item\_id, type, quantity, total\_cost)"

                    + " VALUES (?,?,?,?,?)";

            PreparedStatement preparedStatement = conn.*prepareStatement* (sqlInsert);

            preparedStatement.*setInt*(1, current\_person\_id); *// for customer column too*

            preparedStatement.*setInt*(2, current\_person\_id);

            preparedStatement.*setString*(3, "Cash");

            preparedStatement.*setInt*(4, SessionPerson.*getTableRowsCount*());

            preparedStatement.*setDouble*(5, totalCost);

*// Execute the query*

            int rowsAffected = preparedStatement.*executeUpdate*();

*if* (rowsAffected > 0) {

                JOptionPane.*showMessageDialog*(this,

                    """

                    Order Saved

                    \nTotal: """+totalCost,

                    "Payment Done",

                    JOptionPane.*INFORMATION\_MESSAGE*);

            } *else* {

                JOptionPane.*showMessageDialog*(this,

                    "Failed to save Order Information",

                    "Save Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            }*// if (rowsAffected)*

            }*catch*(SQLException e) {

            JOptionPane.*showMessageDialog*(this,

                    """

                    Failed to Connect

                    \nReason:\nMenu Form\n""" + e.*getMessage*(),

                    "Connection Failed",

                    JOptionPane.*ERROR\_MESSAGE*);

            e.*printStackTrace*();

        } *// try catch block*

*// Clear the text fields*

        model.*setRowCount*(0);

        txtSubTotal.*setText*("");

        txtTax.*setText*("");

        txtTotal.*setText*("");

    }

*private* void *btnDeliveryActionPerformed*(java.awt.event.ActionEvent evt) {

        double billAmount = *receipt*();

        SessionPerson.*setTableRowsCount*(model.*getRowCount*());  *// Directly from the JTable*

*// Pass 1 double to the DeliveryForm(double)*

        DeliveryForm deliveryForm = *new* *DeliveryForm*(billAmount);

*// Open the deliveryForm window*

        deliveryForm.*setVisible*(true);

    }

*private* void *btnCreditActionPerformed*(java.awt.event.ActionEvent evt) {

        double billAmount = *receipt*();

        SessionPerson.*setTableRowsCount*(model.*getRowCount*());  *// Directly from the JTable*

*// Pass 1 double to the CreditForm(double)*

        CreditForm creditForm = *new* *CreditForm*(billAmount);

*// Open the creditForm window*

        creditForm.*setVisible*(true);

    }

*// ----------------- Application Functions -----------------*

*// ================== Function receipt() ==================*

*public* double *receipt*(){

*// sum all the values in the 3rd column "Amount"*

        double sum = 0;

        double tax = 0.03; *// to indicate 3%*

*for* (int i = 0; i< reciptModel.*getRowCount*(); i++){

            sum = sum + Double.*parseDouble*(reciptModel.*getValueAt*(i, 1).*toString*());

        }

        txtSubTotal.*setText*(Double.*toString*(sum));

        double cTotal1 = Double.*parseDouble*(txtSubTotal.*getText*());

        double cTax = (cTotal1 \* tax);

        String iTaxTotal = String.*format*("$ %.2f", cTax);

        txtTax.*setText*(iTaxTotal);

        double iTotal = cTotal1 + cTax;

        txtTotal.*setText*(String.*format*("$ %.2f",iTotal));

*// go through 3 steps to covert a double to BigDecimal then back to double*

*// Convert the double to BigDecimal*

        BigDecimal twoDecimal\_iTotal = *new* *BigDecimal*(iTotal);

*// Set the scale to 2 decimal places with rounding*

        BigDecimal formatted\_iTotal = twoDecimal\_iTotal.*setScale*(2, RoundingMode.*HALF\_UP*);

*// Convert back to double to use it as a return value*

        iTotal = formatted\_iTotal.*doubleValue*();

*return* iTotal;

    }*//recipt()*

*/\*\**

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

        } *catch* (InstantiationException ex) {

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

        } *catch* (IllegalAccessException ex) {

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

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

            java.*util*.*logging*.*Logger*.*getLogger*(Menu.*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*() {

            }

        });

    }

*// Variables declaration - do not modify*

*private* javax.swing.JButton PizzaHeart;

*private* javax.swing.JButton btnCash;

*private* javax.swing.JButton btnCredit;

*private* javax.swing.JButton btnDelivery;

*private* javax.swing.JButton btnExit;

*private* javax.swing.JButton btnNewReceipt;

*private* javax.swing.JButton burger;

*private* javax.swing.JButton cola1Leter;

*private* javax.swing.JButton cola2Leter;

*private* javax.swing.JButton colaCan;

*private* javax.swing.JPanel contentsPanel;

*private* javax.swing.JButton doubleBurger;

*private* javax.swing.JButton frise;

*private* javax.swing.JLabel jLabel1;

*private* javax.swing.JLabel jLabel2;

*private* javax.swing.JLabel jLabel3;

*private* javax.swing.JScrollPane jScrollPane1;

*private* javax.swing.JLabel lblMenuTitle;

*private* javax.swing.JPanel mainContentPanel;

*private* javax.swing.JPanel menuItemsPanel;

*private* javax.swing.JButton pastaPan;

*private* javax.swing.JButton phileCheeseSteak;

*private* javax.swing.JButton phileDouble;

*private* javax.swing.JButton phileLong;

*private* javax.swing.JPanel pricesPanel;

*private* javax.swing.JTable reciptModel;

*private* javax.swing.JButton spagetti;

*private* javax.swing.JTextField txtSubTotal;

*private* javax.swing.JTextField txtTax;

*private* javax.swing.JTextField txtTotal;

*private* javax.swing.JPanel usernamePanel;

*// End of variables declaration*

}

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

*public* *class* PersonClass {

*// make variables public to access them from outside of this class*

*public* String name;

*public* String role;

*public* String password;

*public* String fullName;

}

*//////////////////////////////////////////////////////////////////*

package *aldiwan*;

*public* *class* SessionPerson {

*// Utility class to store global user data*

*private* *static* String currentPersonName;

*private* *static* int currentPersonID;

*private* *static* int tableRowsCount;

*// setter method*

*public* *static* void *setCurrentPersonName*(String username) {

        currentPersonName = username;

    }

*// getter method*

*public* *static* String *getCurrentPersonName*() {

*return* currentPersonName;

    }

*// setter method*

*public* *static* void *setCurrentPersonID*(int userID) {

        currentPersonID = userID;

    }

*// getter method*

*public* *static* int *getCurrentPersonID*() {

*return* currentPersonID;

    }

*// setter method*

*public* *static* void *setTableRowsCount*(int count) {

        tableRowsCount = count;

    }

*// getter method*

*public* *static* int *getTableRowsCount*() {

*return* tableRowsCount;

    }

}*// SessionPerson {}*

*//////////////////////////////////////////////////////////////////*