Food Delivery Application

MINI PROJECT REPORT

Submitted by:

THARANI M.N

HARINISRIVALLI V

KEERTHANA V

BOWYAA L

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAMCO INSTITUTE OF TECHNOLOGY.

RAJAPALAYAM

NOVEMBER 2021

ABSTRACT

OUR PROJECT IS TO:

1· Display the details of the food products

2· Display the bill amount of the customer after each order

3·Display the products ordered by the customer with the following options – based on price

Scope of the Project:

- A big boost to customer experience
- Best GUI Design
- Competitive Prices
- People of Remote places can order easily
- Build a visible menu
- The data of customers such as name number address amount ordered are stored safely in file for later reference
- Exceptions are handled efficiently

METHODS IN THE PROGRAM:

actionPerformed():

This method is used to get action of button and perform the action assigned to corresponding buttons.

getClass():

This method returns the runtime class of an object.

dispose():

This method causes the JFrame window to be destroyed and cleaned up by the operating system.

getContentPane():

This method retrieves the content pane layer so that you can add an object to it.

setVisible():

This method is used to make frame to visible by passing arguments as true.

AIM:

To design an object oriented programming model for food delivery application with the

following modules.

PROGRAM:

```
package mainmenu;
import java.util.*;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Font;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import javax.imageio.ImageIO;
import javax.swing.Box;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import java.awt.Panel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;

@SuppressWarnings("serial")
public class MainMenu extends JPanel {
        JLabel picLabel, title;
        JButton button;
        private Panel panel_1;
        private Panel panel_2;
        static JFrame frame;

        public void createAndShowGUI() throws IOException {
                JPanel panel = new JPanel(new BorderLayout());
                Image image =
ImageIO.read(this.getClass().getResource("food_logo.png"));
                Image imageScaled = image.getScaledInstance(350,
300, Image.SCALE_SMOOTH);
                 ImageIcon imageIcon = new
ImageIcon(imageScaled);
```

```
                                picLabel = new JLabel(imageIcon);
                                Box right = Box.createVerticalBox();
                                panel_1 = new Panel();
                      title = new JLabel("898 Food Restaurant");
                      title.setAlignmentX(Component.CENTER_ALIGNMENT);
                      title.setAlignmentY(0.0f);
                      title.setHorizontalAlignment(SwingConstants.CENTER);
                      title.setFont(new Font("Serif", Font.ITALIC + Font.BOLD, 18));
                      title.setForeground(Color.BLUE);

                      // Button, with filler
                      button = new JButton("Order Food Now >>");
                      panel_1.add(button);
                      button.setAlignmentX(Component.LEFT_ALIGNMENT);

                      panel.add(picLabel, BorderLayout.CENTER);
                      panel.add(right, BorderLayout.SOUTH);
                      right.add(title);
                      right.add(panel_1);
                      add(panel);

                      button.addActionListener(new ActionListener() {
                            public void actionPerformed(ActionEvent e) {
                                  FoodMenu food;
                                  try {
                                        food = new FoodMenu();
                                        food.createAndShowGUI();
                                        food.setVisible(true);
                                        setVisible(false);
                                        frame.dispose();
                                  } catch (IOException e1) {
                                        // TODO Auto-generated catch block
                                        e1.printStackTrace();
                                  }
                            }
                      });
            }

            public static void main(String args[]) throws IOException {
          System.out.print("Enter Customer or Owner:");
          Scanner in=new Scanner(System.in);
            String ch=in.next();
            if(ch.equals("owner")){
              System.out.println("Ruppees\tName\tNumber\t\tAddress\t\tDate");
          int i=1;
        File file = new File("D:\\food.txt");
        BufferedReader br= new BufferedReader(new FileReader(file));
        String st;
        while ((st = br.readLine()) != null){
          System.out.println(i+")"+st);
```

```
           i+=1;
       }
       }
               else if(ch.equals("Customer")){
                         MainMenu main = new MainMenu();
                         main.createAndShowGUI();
                         frame = new JFrame();
                         frame.setTitle("898 Food Ordering System");
                         frame.getContentPane().add(main);
                         frame.pack();
                         frame.setLocationRelativeTo(null);
                         frame.setVisible(true);
               }
               else{
                   System.out.println("Wrong Choice");
               }
             }
}
package mainmenu;

import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Image;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.BorderLayout;
import java.awt.Color;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.SwingConstants;
import javax.swing.border.Border;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.imageio.ImageIO;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;

public class OrderMenu {

	private JFrame frame;
	private JTextField textField;
	private JTextField textField_1;
	private JTextArea textArea;

	/**
	* Launch the application.
	*/
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					OrderMenu window = new OrderMenu();
					window.frame.setVisible(true);
				} catch (Exception e) {
					 e.printStackTrace();
				}
			}
		});
	}

	/**
	* Create the application.
	*
	 * @throws IOException
	*/
	public OrderMenu() throws IOException {
		createAndShowGUI();
	}

	/**
	* Initialize the contents of the frame.
	*
	 * @throws IOException
	*/
	void createAndShowGUI() throws IOException {
	  BufferedReader w1=new BufferedReader(new FileReader("D:\\food.txt"));
	  BufferedWriter w=new BufferedWriter(new FileWriter("D:\\food.txt",true));
	  String m;
	  while((m=w1.readLine())!=null){

	  }
		frame = new JFrame("Order Form");
		frame.setBounds(100, 100, 420, 300); // x,y,width,height
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.getContentPane().setLayout(null);
```

```
        frame.setLocationRelativeTo(null);

        JLabel lblFillInDetails = new JLabel("Fill in Details");
        lblFillInDetails.setBounds(120, 11, 131, 26);
        lblFillInDetails.setFont(new Font("Serif", Font.PLAIN, 16));
        lblFillInDetails.setForeground(Color.BLUE);
        frame.getContentPane().add(lblFillInDetails);

        JLabel lblName = new JLabel("Name");
        lblName.setBounds(10, 46, 46, 14);
        frame.getContentPane().add(lblName);

         textField = new JTextField();
        textField.setBounds(82, 43, 176, 20);
        frame.getContentPane().add(textField);
        textField.setColumns(10);


        JLabel lblAddress = new JLabel("Tel No");
        lblAddress.setBounds(10, 92, 46, 14);
        frame.getContentPane().add(lblAddress);

        textField_1 = new JTextField();
        textField_1.setBounds(82, 89, 176, 20);
        frame.getContentPane().add(textField_1);
        textField_1.setColumns(10);

        JLabel lblAddress_1 = new JLabel("Address");
        lblAddress_1.setBounds(10, 137, 62, 14);
        frame.getContentPane().add(lblAddress_1);

        Border border = BorderFactory.createLineBorder(Color.BLACK);
        textArea = new JTextArea();
        textArea.setBounds(82, 132, 236, 85);
        textArea.setMargin(new Insets(10,10,10,10)); // top,left,bottom,right
        textArea.setBorder(BorderFactory.createCompoundBorder(border,
BorderFactory.createEmptyBorder(10, 10, 10, 10)));
        frame.getContentPane().add(textArea);

        JButton btnCancel = new JButton("Cancel ");
        btnCancel.setBounds(130, 228, 89, 23);
        frame.getContentPane().add(btnCancel);

        btnCancel.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                FoodMenu food = new FoodMenu();
                try {
                    food.createAndShowGUI();
                    food.setVisible(true);
                    setVisible(false);
```

```
               } catch (IOException e1) {
                      // TODO Auto-generated catch block
                      e1.printStackTrace();
               }

               frame.dispose();
       }
});

JButton btnConfirm = new JButton("Confirm");
btnConfirm.setBounds(229, 228, 89, 23);
frame.getContentPane().add(btnConfirm);

btnConfirm.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                              if (textField.getText().equals("") ||
                               textField_1.getText().equals("")
                              || textArea.getText().equals("")) {
                        JOptionPane.showMessageDialog(null,
                                    "Field cannot be empty");
                                                    } else {
                                                       try {
                               w.write(textField.getText()+"\t");
                             w.write(textField_1.getText()+"\t");
                               w.write(textArea.getText()+"\t");
                                     DateTimeFormatter dtf =
                   DateTimeFormatter.ofPattern("yyyy/MM/dd");
                   LocalDateTime now = LocalDateTime.now();
                                 w.write(dtf.format(now)+"\n");

                                                  w.close();
                                                         }

                                     catch (IOException ex) {
       Logger.getLogger(OrderMenu.class.getName()).log(Level.S
                                           EVERE, null, ex);
                                                         }
                        JOptionPane.showMessageDialog(null,
         "Your foods will be delivered soon, thanks for ordering. ");
                           MainMenu main = new MainMenu();
                                                       try {
                                            main.main(null);
                                    } catch (IOException e1) {
                                // TODO Auto-generated catch
                                                     block
                                        e1.printStackTrace();
                                                         }
                           FoodMenu food = new FoodMenu();

                                        main.setVisible(true);
```

```
                                                            // setVisible(false);
                                                              frame.dispose();
                                                                             }
                        }
                });
                Image image =
ImageIO.read(this.getClass().getResource("order.png"));
                Image imageScaled = image.getScaledInstance(80, 95,
Image.SCALE_SMOOTH);
                ImageIcon imageIcon = new ImageIcon(imageScaled);
                JLabel lblNewLabel = new JLabel(imageIcon);
                lblNewLabel.setBounds(268, 19, 126, 90);
                frame.getContentPane().add(lblNewLabel);

        }

        public void setVisible(boolean b) {
                // TODO Auto-generated method stub

        }
}
package mainmenu;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Image;
import java.awt.Insets;
import java.awt.TextArea;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JSpinner;
import javax.swing.JTabbedPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SpinnerNumberModel;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import javax.swing.table.DefaultTableModel;

public class FoodMenu  {

        static private JFrame frame;
        static private JButton backButton, orderButton;
        static private JTextField textField;
        static private GridBagConstraints gbc;
        private JTable table;
        DefaultTableModel dtm;
        Double[] price;
        Double[] priceDrinks;
        Double[] priceDesserts;
        double totalPrice = 0;
        double p1, p2, p3, p4, p5, p6, p7, p8, p9;
        double d1, d2, d3, d4, d5;
        double de1, de2;

        private JSpinner[] numSpinner;
        static private JLabel[] foodLabel;
        static private JLabel[] foodImage;
        private String[] file;

        private JSpinner[] numSpinnerDrinks;
        static private JLabel[] drinkLabel;
        static private JLabel[] drinkImage;
        private String[] fileDrinks;

        private JSpinner[] numSpinnerDesserts;
        static private JLabel[] dessertLabel;
        static private JLabel[] dessertImage;
        private String[] fileDesserts;

        private static final int ELEMENTS = 9;
        private static final int DRINK_ELEMENTS = 5;
        private static final int DESSERT_ELEMENTS = 2;

        double total = 0;
        double food1, food2, food3, food4, food5, food6, food7, food8, food9;
        double drink1, drink2, drink3, drink4, drink5;
        double pr, pr1;

        double totalForFoods;
        double totalForDrinks;
        double totalForDesserts;

        void createAndShowGUI() throws IOException {
```

```
		frame = new JFrame("Main Menu ");
		frame.setBounds(100, 100, 750, 550);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.getContentPane().setLayout(null);
		frame.setLocationRelativeTo(null);
		JLabel lblFoodOrdered = new JLabel("Food Ordered");
		lblFoodOrdered.setBounds(529, 11, 81, 14);

		frame.getContentPane().add(lblFoodOrdered);

		table = new JTable();
		backButton = new JButton();
		orderButton = new JButton();
		dtm = new DefaultTableModel(0, 0);
		final String header[] = new String[] { "Item", "Qty", "Price", "Spinner"
};
		dtm.setColumnIdentifiers(header);
		dtm.addRow(header);
		table = new JTable();
		table.setModel(dtm);
		table.setBounds(475, 31, 1, 1); // int x, int y, int width, int height
		table.setSize(245, 300); // width,height
		table.getColumnModel().getColumn(0).setPreferredWidth(80);
		table.getColumnModel().getColumn(1).setPreferredWidth(30);
		table.getColumnModel().getColumn(2).setPreferredWidth(30);
		table.getColumnModel().getColumn(3).setMinWidth(0); // hide
spinner

							// column
		table.getColumnModel().getColumn(3).setMaxWidth(0); // hide
spinner

							// column
		table.setShowGrid(false); // remove cell boarder
		frame.getContentPane().add(table);
		JLabel lblTotal = new JLabel("Total  : ");
		lblTotal.setBounds(519, 340, 46, 14);
		frame.getContentPane().add(lblTotal);
		textField = new JTextField();
		textField.setBounds(585, 340, 86, 20);
		frame.getContentPane().add(textField);
		textField.setColumns(10);
		orderButton = new JButton("Order");
		orderButton.setBounds(500, 385, 89, 23);
		frame.getContentPane().add(orderButton);
		backButton = new JButton("Back");
		backButton.setBounds(610, 385, 89, 23);
		frame.getContentPane().add(backButton);
```

```
            JTabbedPane tabbedPane = new
JTabbedPane(JTabbedPane.TOP);
            addIt(tabbedPane, "Foods");
            addIt1(tabbedPane, "Drinks");
            addIt2(tabbedPane, "Desserts");
            tabbedPane.setBounds(18, 11, 450, 450);
            frame.getContentPane().add(tabbedPane);
            frame.setVisible(true);

            backButton.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        try {
                              MainMenu menu = new MainMenu();
                              menu.main(header);
                              // menu.createAndShowGUI();
                              menu.setVisible(true);
                              // setVisible(false);
                              frame.dispose();
                        } catch (IOException e1) {
                              // TODO Auto-generated catch block
                              e1.printStackTrace();
                        }
                  }
            });

            orderButton.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        if (textField.getText().equals("")) {
                              JOptionPane.showMessageDialog(null,
"You not ordered anything yet");
                        } else {
                              try {
                                    OrderMenu order = new
OrderMenu();
                                    order.main(header);
                     BufferedReader w1=new BufferedReader(new
FileReader("D:\\food.txt"));
                 BufferedWriter w=new BufferedWriter(new
FileWriter("D:\\food.txt",true));
                 String m;
                 while((m=w1.readLine())!=null){

                 }
                 w.write(textField.getText()+"\t");
                 w.close();
                                    order.setVisible(true);
                                    setVisible(false);
                                    frame.dispose();
                              } catch (IOException e1) {
```

```
                                                    // TODO Auto-generated catch
block
                                                    e1.printStackTrace();
                                            }
                                    }
                            }

                    });
            }

            void addIt(JTabbedPane tabbedPane, String text) throws IOException {
                    JPanel panel = new JPanel(new GridBagLayout());
                    gbc = new GridBagConstraints(); // getting constraints for the
specified

 // component
                    gbc.insets = new Insets(10, 0, 0, 0);
                    foodImage = new JLabel[ELEMENTS];
                    foodLabel = new JLabel[ELEMENTS];
                    numSpinner = new JSpinner[ELEMENTS];
                    file = new String[ELEMENTS];
                    price = new Double[ELEMENTS];

                    file[0] = new String("MedSalad.png");
                    file[1] = new String("JapanesePanNoodles.png");
                    file[2] = new String("spaghetti.jpg");
                    file[3] = new String("PadThai.png");
                    file[4] = new String("RamenNoodles.png");
                    file[5] = new String("kids_spaghetti.png");
                    file[6] = new String("chickenRice.jpg");
                    file[7] = new String("thaiFood.jpg");
                    file[8] = new String("vietnamFood.jpg");
                    foodLabel[0] = new JLabel("Salad");
                    foodLabel[1] = new JLabel("Japanese Noodles");
                    foodLabel[2] = new JLabel("Spaghetti");
                    foodLabel[3] = new JLabel("Spaghetti Meat Balls");
                    foodLabel[4] = new JLabel("Noodles");
                    foodLabel[5] = new JLabel("Kids Spaghetti");
                    foodLabel[6] = new JLabel("Chicken Rice");
                    foodLabel[7] = new JLabel("Thai Food");
                    foodLabel[8] = new JLabel("Vietnam Food");
                    price[0] = 350.00;
                    price[1] = 450.00;
                    price[2] = 370.0;
                    price[3] = 450.0;
                    price[4] = 550.0;
                    price[5] = 400.0;
                    price[6] = 350.0;
                    price[7] = 650.0;
                    price[8] = 650.00;
```

```
        for (int i = 0; i < ELEMENTS; i++) {

            System.out.print(file[i]);
            try {

            Image image =
ImageIO.read(this.getClass().getResource(file[i]));
            Image imageScaled = image.getScaledInstance(80, 95,
Image.SCALE_SMOOTH);
//          Image image = ImageIO.read(file[i]);
//          Image imageScaled = image.getScaledInstance(80, 95,
Image.SCALE_SMOOTH);
            ImageIcon imageIcon = new ImageIcon(imageScaled);
            SpinnerNumberModel spnummodel = new
SpinnerNumberModel(0, 0, 10, 1); // value,minimum,maximum,stepSize
            numSpinner[i] = new JSpinner(spnummodel);
            numSpinner[i].addChangeListener(listener);
            foodImage[i] = new JLabel(imageIcon);
            }catch(Exception e) {
                System.out.print(e);
            }
        }
        gbc.gridx = 0; // gridx 0 represent the most left
        for (int i = 0; i < ELEMENTS; i++) {
            if (i % 3 == 0) {
                gbc.gridy += 2;
                gbc.gridx = 0;
            }
            panel.add(foodImage[i], gbc);
            gbc.gridy++; // gridy---> add one row,for foodLabel
            panel.add(foodLabel[i], gbc);
            gbc.gridy--; // remove the row
            gbc.gridx++; // move to next column
            panel.add(numSpinner[i], gbc);
            gbc.gridx++; // move to next column
            tabbedPane.addTab(text, panel);
        }
    }

    void addIt2(JTabbedPane tabbedPane, String text) throws IOException {
        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        dessertImage = new JLabel[DESSERT_ELEMENTS];
        dessertLabel = new JLabel[DESSERT_ELEMENTS];
        numSpinnerDesserts = new JSpinner[DESSERT_ELEMENTS];
        priceDesserts = new Double[DESSERT_ELEMENTS];

        fileDesserts = new String[DESSERT_ELEMENTS];
        fileDesserts[0] = new String("strawberry cake.jpg");
        fileDesserts[1] = new String("chocolate cake.jpg");
        dessertLabel[0] = new JLabel("Strawberry Cake");
        dessertLabel[1] = new JLabel("Chocolate Cake");

        priceDesserts[0] = 250.00;
        priceDesserts[1] = 300.00;

        for (int i = 0; i < DESSERT_ELEMENTS; i++) {
            Image image =
ImageIO.read(this.getClass().getResource(fileDesserts[i]));
            Image imageScaled = image.getScaledInstance(80, 95,
Image.SCALE_SMOOTH);
            ImageIcon imageIcon = new ImageIcon(imageScaled);
            SpinnerNumberModel spnummodel = new
SpinnerNumberModel(0, 0, 10, 1); // value,minimum,maximum,stepSize
            numSpinnerDesserts[i] = new JSpinner(spnummodel);
            numSpinnerDesserts[i].addChangeListener(listenerForDes
serts);
            dessertImage[i] = new JLabel(imageIcon);
        }
        gbc.gridx = 0; // gridx 0 represent the most left
        gbc.insets = new Insets(10, 5, 0, 0); // top,left,bottom,right
        for (int i = 0; i < DESSERT_ELEMENTS; i++) {
            if (i % 3 == 0) {
                gbc.gridx = 0;
                gbc.gridy += 2;
            }
            panel.add(dessertImage[i], gbc);
            gbc.gridy++; // gridy---> add one row,for foodLabel
            panel.add(dessertLabel[i], gbc);
            gbc.gridy--; // remove the row
            gbc.gridx++; // move to next column
            panel.add(numSpinnerDesserts[i], gbc);
            gbc.gridx++; // move to next column
            tabbedPane.addTab(text, panel);
        }

    }

    void addIt1(JTabbedPane tabbedPane, String text) throws IOException {

        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        drinkImage = new JLabel[DRINK_ELEMENTS];
        drinkLabel = new JLabel[DRINK_ELEMENTS];
        numSpinnerDrinks = new JSpinner[DRINK_ELEMENTS];
        priceDrinks = new Double[DRINK_ELEMENTS];

        fileDrinks = new String[DRINK_ELEMENTS];
        fileDrinks[0] = new String("raspberry.jpg");
        fileDrinks[1] = new String("chocalate_pudding.jpg");
        fileDrinks[2] = new String("blue hawailan.jpg");
        fileDrinks[3] = new String("Pina.jpg");
        fileDrinks[4] = new String("lemon ice.jpg");

        drinkLabel[0] = new JLabel("Raspberry");
        drinkLabel[1] = new JLabel("Chocolate Pudding");
        drinkLabel[2] = new JLabel("Blue Hawailan");
        drinkLabel[3] = new JLabel("Pina");
        drinkLabel[4] = new JLabel("Lemon Ice");

        priceDrinks[0] = 35.0;
        priceDrinks[1] = 45.0;
        priceDrinks[2] = 30.0;
        priceDrinks[3] = 50.0;
        priceDrinks[4] = 30.0;

        for (int i = 0; i < DRINK_ELEMENTS; i++) {
            Image image =
ImageIO.read(this.getClass().getResource(fileDrinks[i]));
            Image imageScaled = image.getScaledInstance(80, 95,
Image.SCALE_SMOOTH);
            ImageIcon imageIcon = new ImageIcon(imageScaled);
            SpinnerNumberModel spnummodel = new
SpinnerNumberModel(0, 0, 10, 1); // value,minimum,maximum,stepSize
            numSpinnerDrinks[i] = new JSpinner(spnummodel);
            numSpinnerDrinks[i].addChangeListener(listenerForDrinks
);
            drinkImage[i] = new JLabel(imageIcon);
        }
        gbc.gridx = 0; // gridx 0 represent the most left
        gbc.insets = new Insets(10, 5, 0, 0); // top,left,bottom,right
        for (int i = 0; i < DRINK_ELEMENTS; i++) {
            if (i % 3 == 0) {
                gbc.gridx = 0;
                gbc.gridy += 2;
            }
            panel.add(drinkImage[i], gbc);
            gbc.gridy++; // gridy---> add one row,for foodLabel
            panel.add(drinkLabel[i], gbc);
            gbc.gridy--; // remove the row
            gbc.gridx++; // move to next column
            panel.add(numSpinnerDrinks[i], gbc);
            gbc.gridx++; // move to next column
            tabbedPane.addTab(text, panel);

        }
    }

                ChangeListener listener = new ChangeListener() {
```

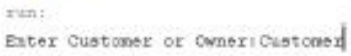
```
public void stateChanged(ChangeEvent e) {

final int quantity = (int) ((JSpinner)
e.getSource()).getValue();
final int rows = table.getRowCount();
for (int row = 0; row < rows; row++) {
if (dtm.getValueAt(row, 3) == e.getSource()) {
if (dtm.getValueAt(row,
0).equals("Salad")) {
dtm.setValueAt(quantity, row, 1);
// obj, row, column
dtm.setValueAt(p1 * quantity,
row, 2);
food1 = p1 * quantity;

} else if (dtm.getValueAt(row,
0).equals("Japanese Noodles")) {

dtm.setValueAt(quantity, row, 1);
dtm.setValueAt(p2 * quantity,
row, 2);
food2 = p2 * quantity;
} else if (dtm.getValueAt(row,
0).equals("Spaghetti")) {

dtm.setValueAt(quantity, row, 1);
dtm.setValueAt(p3 * quantity,
row, 2);
food3 = p3 * quantity;
} else if (dtm.getValueAt(row,
0).equals("Spaghetti Meat Balls")) {

dtm.setValueAt(quantity, row, 1);
dtm.setValueAt(p4 * quantity,
row, 2);
food4 = p4 * quantity;
} else if (dtm.getValueAt(row,
0).equals("Noodles")) {

dtm.setValueAt(quantity, row, 1);
dtm.setValueAt(p5 * quantity,
row, 2);
food5 = p5 * quantity;
} else if (dtm.getValueAt(row,
0).equals("Kids Spaghetti")) {

dtm.setValueAt(quantity, row, 1);
dtm.setValueAt(p6 * quantity,
row, 2);
food6 = p6 * quantity;
```

```
                               } else if (dtm.getValueAt(row,
                                0).equals("Chicken Rice")) {

                           dtm.setValueAt(quantity, row, 1);
                                                  // obj, row,

                                                    // column
                               dtm.setValueAt(p7 * quantity,
                                                     row, 2);
                                       food7 = p7 * quantity;
                               } else if (dtm.getValueAt(row,
                                    0).equals("Thai Food")) {

                           dtm.setValueAt(quantity, row, 1);
                                                  // obj, row,

                                                    // column
                               dtm.setValueAt(p8 * quantity,
                                                     row, 2);
                                       food8 = p8 * quantity;
                               } else if (dtm.getValueAt(row,
                                0).equals("Vietnam Food")) {

                           dtm.setValueAt(quantity, row, 1);
                                                  // obj, row,

                                                    // column
                               dtm.setValueAt(p9 * quantity,
                                                     row, 2);
                                       food9 = p9 * quantity;
                                                            }

                                           if (quantity == 0) {
                                      dtm.removeRow(row);
                                                            }
                   totalForFoods = food1 + food2 + food3 +
            food4 + food5 + food6 + food7 + food8 + food9;
                    total = totalForFoods + totalForDrinks +
                                           totalForDesserts;
                                 textField.setText(total + "");
                                                      return;
                                                            }
                                                            }

// there was no row with this JSpinner, so we have to add it
for (int i = 0; i < ELEMENTS; i++) {
         // looking for the "clicked" JSpinner
         if (numSpinner[i] == e.getSource()) {
                 totalPrice = price[i];
                 switch (foodLabel[i].getText()) {
```

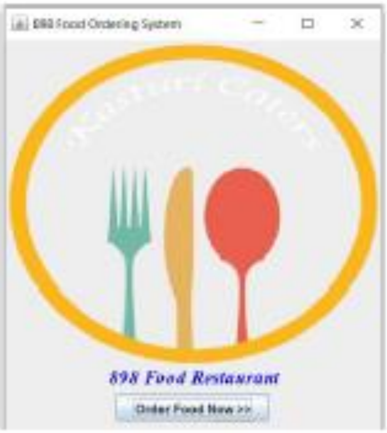
```
          case "Salad":
              p1 = 350.0;
              food1 = p1;
              break;
          case "Japanese Noodles":
              p2 = 450.0;
              food2 = p2;
              break;
          case "Spaghetti":
              p3 = 370.0;
              food3 = p3;
              break;
          case "Spaghetti Meat Balls":
              p4 = 450.0;
              food4 = p4;
              break;
          case "Noodles":
              p5 = 55.0;
              food5 = p5;
              break;
          case "Kids Spaghetti":
              p6 = 400.0;
              food6 = p6;
              break;
          case "Chicken Rice":
              p7 = 350.0;
              food7 = p7;
&nb
```

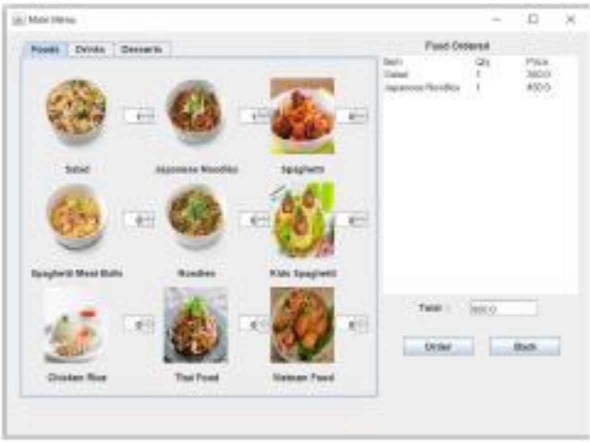
OUTPUT:

OUTPUT:



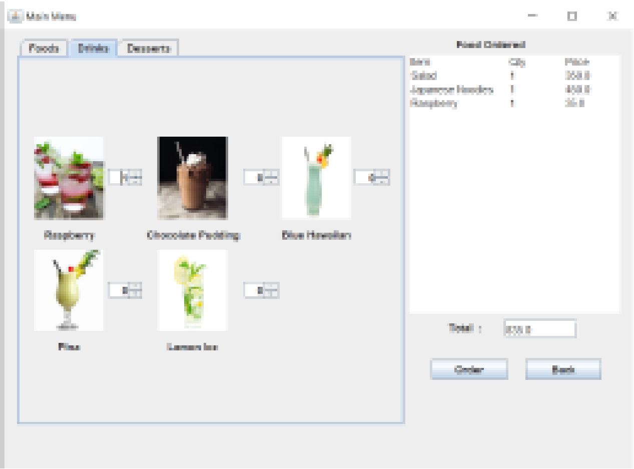
```
run:
Enter Customer or Owner:Customer
```





Item	Qty	Price
Salad	1	[illegible]
Japanese Noodles	1	[illegible]

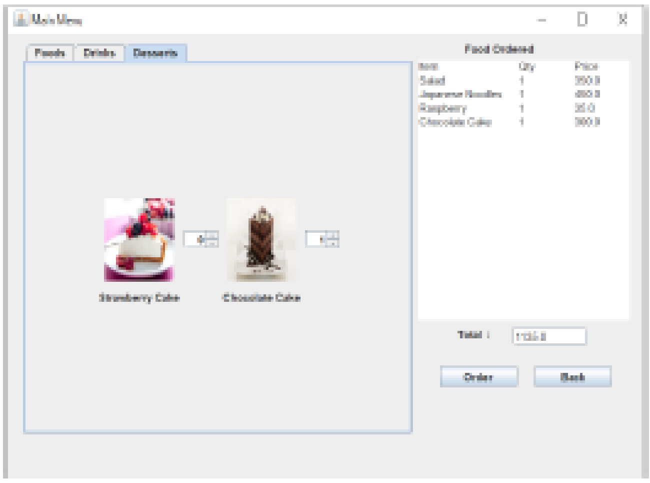
Total : [illegible]



Food Ordered

Item	Qty	Price
Salad	1	350.0
Japanese Noodles	1	450.0
Raspberry	1	35.0

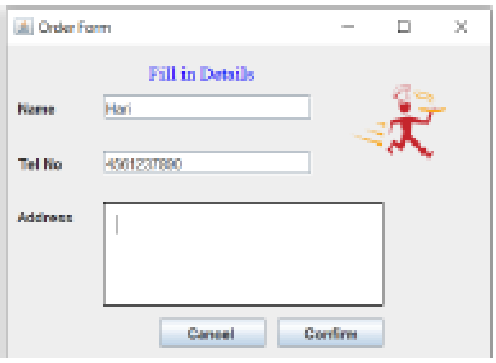
Total : 835.0



Food Ordered

Item	Qty	Price
Salad	1	350.0
Japanese Noodles	1	450.0
Raspberry	1	35.0
Chocolate Cake	1	300.0

Total : 1135.0

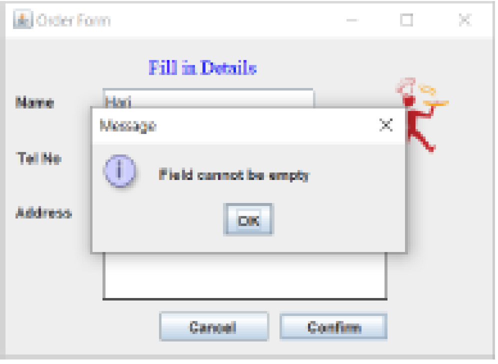


Fill in Details

Name Hari

Tel No 4561237890

Address



Fill in Details

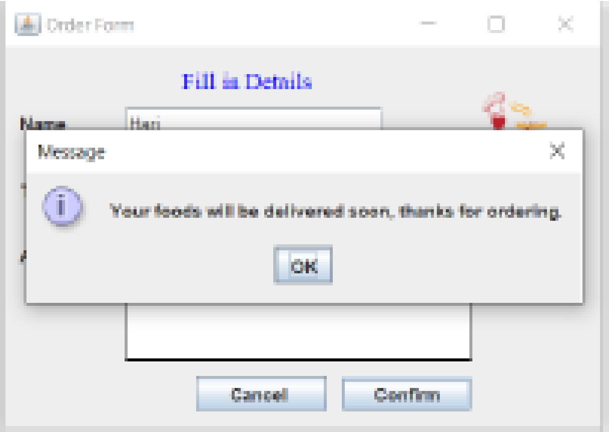
Name Hari

Tel No

Address

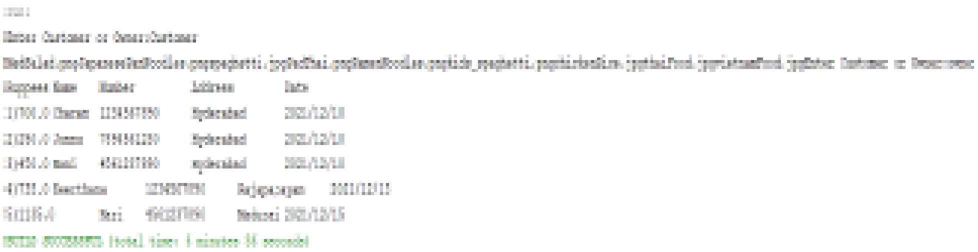
Message

Field cannot be empty



Name Hari

Your foods will be delivered soon, thanks for ordering.



```
run:
Enter Customer or Owner:Customer
MedSalad.pngJapaneseUdonNoodles.pngspaghetti.jpgRedThai.pngRamenNoodles.pngkids_spaghetti.pngchickenRice.jpgthaiFood.jpgvietnamFood.jpgEnter Customer or Owner:owner
Expenses Name   Number          Address         Date
1|700.0 Charan  1234567890      Hyderabad       2021/12/13
2|1250.0 Jeevan 7894561230      Hyderabad       2021/12/13
3|450.0 mani    4561237890      Hyderabad       2021/12/13
4|731.0 SwastHana       1234567890      Rajapalayam     2021/12/13
5|1185.0        Hari    4561237890      Medwai 2021/12/15
BUILD SUCCESSFUL (total time: 1 minutes 36 seconds)
```