

ORDERING APP REPORT

PROJECT 2

NAME : NURSYAFIQAH IZANI BINTI MOHD ZAIN

MATRICS NUMBER : BI19110120

IC NUMBER : 001230120206

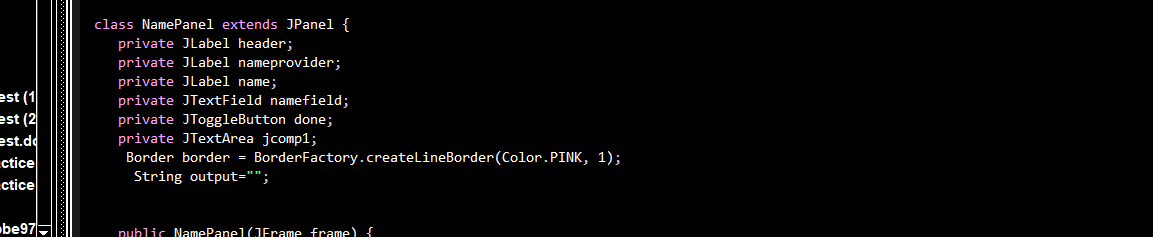
SUBJECT : KK14203 (OBJECT ORIENTED PROGRAMMING)

1) Java Code

import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
import javax.swing.event.\*;  
import javax.swing.JOptionPane;  
import java.util.Random;  
//required for border  
import javax.swing.BorderFactory;  
import javax.swing.border.Border;  
//required for file IO  
import java.io.File;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.PrintWriter;  
import java.io.BufferedWriter;  
import java.io.BufferedReader;  
//required for exception  
import java.io.IOException;  
  
 class NamePanel extends JPanel {  
 private JLabel header;  
 private JLabel nameprovider;  
 private JLabel name;  
 private JTextField namefield;  
 private JToggleButton done;  
 private JTextArea jcomp1;  
 Border border = BorderFactory.createLineBorder(Color.PINK, 1);  
 String output="";  
  
  
 public NamePanel(JFrame frame) {  
 setLayout(new FlowLayout(FlowLayout.LEFT));   
   
 //construct components  
 header = new JLabel("Welcome to Purr Cafe");  
 nameprovider = new JLabel ("Please enter your name");  
 name = new JLabel ("Name");  
 namefield = new JTextField (20);  
 done = new JToggleButton ("done", false);  
 jcomp1 = new JTextArea (5, 5);  
   
  
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add(header);  
 add (nameprovider);  
 add (name);  
 add (namefield);  
 add (done);  
 add (jcomp1);  
  
 //set component bounds (only needed by Absolute Positioning)  
 header.setBounds(390, 155, 160, 25);  
 nameprovider.setBounds (385, 140, 147, 25);  
 name.setBounds (255, 175, 100, 25);  
 namefield.setBounds (325, 175, 290, 25);  
 done.setBounds (400, 225, 100, 25);  
 jcomp1.setBounds (40, 500, 230, 30);  
   
 //adding action listener  
  
 done.addActionListener(new ActionListener(){   
 public void actionPerformed(ActionEvent e){   
   
   
 if(namefield.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please Enter Your Name :");  
   
 }  
 else {  
 JOptionPane.showMessageDialog (null, "Hello " + namefield.getText() + " Thank you for choosing to eat at our cafe");  
 if(printOutputName()){  
 writeInputName();   
 JOptionPane.showMessageDialog(null,"Successfully Saved");  
 }  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new StatusPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }   
   
 }  
 });  
 }  
   
 public boolean printOutputName(){  
   
 output += namefield.getText();  
  
   
 jcomp1.setText(output);  
 return true;  
 }  
   
 public void writeInputName(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 jcomp1.setText(io.toString());  
 }  
  
 String input = "Name :" +namefield.getText() ;  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 jcomp1.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 jcomp1.setText(e.toString());  
 }  
 }  
 }  
}  
   
   
   
   
 class StatusPanel extends JPanel {  
 private JLabel statuslbl;  
 private JLabel statuscheck;  
 private JComboBox <String> selection;  
 private JTextArea statuslb;  
 String output ="";  
 public StatusPanel(JFrame frame) {  
 //construct preComponents  
 String[] statusselectionItems = {"[Select]", "Dine In", "Take Away", "Delivery"};  
  
 //construct components  
 statuslbl = new JLabel ("Status");  
 statuscheck = new JLabel ("Please choose a status :");  
 JComboBox <String> selection = new JComboBox <String> (statusselectionItems);  
 statuslb = new JTextArea (5, 5);  
   
 selection.addItemListener (new ItemListener () {;  
 public void itemStateChanged (ItemEvent event){   
 if(event.getStateChange () == ItemEvent.SELECTED) {  
   
 if (selection.getSelectedItem().equals("Dine In")){  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new DineInPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 else if(selection.getSelectedItem().equals("Take Away")){  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new TakeAwayPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 else if (selection.getSelectedItem().equals("Delivery")){  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new DeliveryPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
   
 }  
 }  
 });  
   
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (statuslbl);  
 add (statuscheck);  
 add (selection);  
 add (statuslb);  
 //set component bounds (only needed by Absolute Positioning)  
 statuslbl.setBounds (420, 45, 100, 25);  
 statuscheck.setBounds (380, 100, 155, 25);  
 selection.setBounds (400, 155, 100, 25);  
 statuslb.setBounds (240, 380, 375, 155);  
   
 }  
   
   
 class DineInPanel extends JPanel {  
 private JLabel dinein\_chck;  
 private JLabel temp\_lbl;  
 private JTextField tempfield;  
 private JButton submit\_btn;  
 private JButton finish\_btn;  
 private JLabel notabel;  
 private JTextField tablelbl;  
 private JButton nooftable;  
 private JTextArea dinein;  
  
 public DineInPanel(JFrame frame) {  
 //construct components  
 dinein\_chck = new JLabel ("You chose to Dine In");  
 temp\_lbl = new JLabel ("Temperature (Degree Celcius) ");  
 tempfield = new JTextField (100);  
 notabel = new JLabel ("Number Of Table : ");  
 tablelbl = new JTextField (5);  
 dinein = new JTextArea (5, 5);  
   
 nooftable = new JButton ("Click To See Your Table Number");  
 submit\_btn = new JButton ("submit");  
 finish\_btn = new JButton ("finish");  
   
   
   
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (dinein\_chck);  
 add (temp\_lbl);  
 add (tempfield);  
 add (submit\_btn);  
 add (finish\_btn);  
 add (notabel);  
 add (tablelbl);  
 add (nooftable);  
 add (dinein);  
  
 //set component bounds (only needed by Absolute Positioning)  
 dinein\_chck.setBounds (380, 75, 128, 25);  
 temp\_lbl.setBounds (110, 115, 350, 25);  
 tempfield.setBounds (335, 120, 100, 25);  
 submit\_btn.setBounds (455, 120, 100, 25);  
 finish\_btn.setBounds (400, 495, 100, 25);  
 notabel.setBounds (150, 330, 129, 25);  
 tablelbl.setBounds (300, 335, 100, 25);  
 nooftable.setBounds (425, 335, 250, 25);  
 dinein.setBounds (765, 170, 170, 255);  
   
 submit\_btn.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent e) {  
   
 try {  
 double i = Double.parseDouble (tempfield.getText());  
   
 if(i >36.5 && i <= 37.50){  
 JOptionPane.showMessageDialog(null,"You're normal, you can dine in");   
 }  
 else if (i > 37.5){  
 JOptionPane.showMessageDialog(null,"You have a fever, you can not dine in");  
 System.exit(0);  
 }  
 else if(i<36.5) {  
 JOptionPane.showMessageDialog(null,"You are hypothermia, you can dine in");  
 }  
   
   
 } catch (NumberFormatException ae) {  
 JOptionPane.showMessageDialog(null,"Please enter numbers only");   
   
 }  
 }  
 });  
   
 nooftable.addActionListener( new ActionListener () {  
 public void actionPerformed (ActionEvent e) {  
 Random table = new Random ();  
 int t = table.nextInt (20)+1;  
 String value = String.valueOf (t);  
 tablelbl.setText(value);  
 }  
 });  
   
   
 finish\_btn.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent e) {  
 if(tempfield.getText().equals("") || tablelbl.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please make sure everyhing is filled before you proceed");  
 }  
 else{  
 if(printOutputDineIn()){  
 writeInputDineIn();  
 JOptionPane.showMessageDialog(null,"Successfully Saced");  
 }  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new MenuPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 }  
 });  
   
 }  
 public boolean printOutputDineIn(){  
   
 output += "Table number:" +tablelbl.getText()+"\n";  
 output += "Temperature: " +tempfield.getText()+"\n";  
  
 JOptionPane.showMessageDialog(null, "The data is saved successfully.", "INFORMATION",JOptionPane.INFORMATION\_MESSAGE);   
 dinein.setText(output);  
 return true;  
 }  
   
 public void writeInputDineIn(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 dinein.setText(io.toString());  
 }  
  
 String input = "No.of table " + tablelbl.getText() + " Temperature " + tempfield.getText();  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 dinein.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 dinein.setText(e.toString());  
 }  
 }  
 }  
}  
   
   
  
 }  
   
 class TakeAwayPanel extends JPanel {  
 private JLabel takeawaylbl;  
 private JLabel patientlylbl;  
 private JLabel turnlbl;  
 private JTextField turn;  
 private JButton generatebtn;  
 private JButton finish\_btn2;  
 private JTextArea takeaway;  
 String output ="";  
 public TakeAwayPanel(JFrame frame) {  
 //construct components  
 takeawaylbl = new JLabel ("Take Away");  
 patientlylbl = new JLabel ("Please patiently wait for your turn ");  
 turnlbl = new JLabel ("This is your turn: ");  
 turn = new JTextField (100) ;  
   
 generatebtn = new JButton ("Click to see your turn");  
 finish\_btn2 = new JButton ("finish");  
 takeaway = new JTextArea (5, 5);  
   
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (takeawaylbl);  
 add (patientlylbl);  
 add (turnlbl);  
 add (turn);  
 add(finish\_btn2);  
 add(generatebtn);  
 add (takeaway);  
  
 //set component bounds (only needed by Absolute Positioning)  
 takeawaylbl.setBounds (420, 55, 100, 25);  
 patientlylbl.setBounds (345, 100, 206, 25);  
 turnlbl.setBounds (255, 160, 100, 25);  
 turn.setBounds (375, 155, 105, 30);  
 finish\_btn2.setBounds (370, 275, 100, 25);  
 generatebtn.setBounds(255, 200, 300, 25);  
 takeaway.setBounds (725, 125, 170, 255);  
   
 finish\_btn2.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 if (turn.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please take a number first");  
 }  
 else{  
 if(printOutputTakeAway()){  
 writeInputTakeAway();  
 JOptionPane.showMessageDialog(null,"Successfully save");  
 }  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new MenuPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 }  
 });   
   
   
 generatebtn.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e) {  
   
 Random trn = new Random();  
 int f = trn.nextInt(100)+1;  
 String val = String.valueOf(f);  
 turn.setText(val);  
 }   
 });   
   
 }   
 public boolean printOutputTakeAway(){  
   
 output += "Turn : " + turn.getText();  
  
   
 takeaway.setText(output);  
 return true;  
 }  
   
 public void writeInputTakeAway(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 takeaway.setText(io.toString());  
 }  
  
 String input = "Turn : " +turn.getText();  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 takeaway.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 takeaway.setText(e.toString());  
 }  
 }  
 }  
}   
   
 class DeliveryPanel extends JPanel {  
 private JLabel dliverylbl;  
 private JLabel addrlbl;  
 private JTextField addressfield;  
 private JLabel phonenolbl;  
 private JTextField phonenumber;  
 private JButton finishbutton;  
 private JTextArea delivery;  
 String output ="";  
  
 public DeliveryPanel(JFrame frame) {  
 //construct components  
 dliverylbl = new JLabel ("Delivery");  
 addrlbl = new JLabel ("Address");  
 addressfield = new JTextField (100);  
 phonenolbl = new JLabel ("Phone Number");  
 phonenumber = new JTextField (100);  
 finishbutton = new JButton ("finish");  
 delivery = new JTextArea (5, 5);  
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (dliverylbl);  
 add (addrlbl);  
 add (addressfield);  
 add (phonenolbl);  
 add (phonenumber);  
 add (finishbutton);  
 add (delivery);  
 //set component bounds (only needed by Absolute Positioning)  
 dliverylbl.setBounds (350, 80, 100, 25);  
 addrlbl.setBounds (145, 140, 100, 25);  
 addressfield.setBounds (225, 140, 400, 25);  
 phonenolbl.setBounds (120, 185, 100, 25);  
 phonenumber.setBounds (225, 190, 250, 25);  
 finishbutton.setBounds (365, 250, 100, 25);  
 delivery.setBounds (765, 120, 170, 255);  
   
 finishbutton.addActionListener (new ActionListener (){  
 public void actionPerformed(ActionEvent e){  
 try {  
 int pn = Integer.parseInt (phonenumber.getText());  
 String phoneNumber = phonenumber.getText();   
   
 if(phoneNumber.length()>10 || phoneNumber.length()<10){  
 JOptionPane.showMessageDialog(null, "Phone number is invalid");  
 }  
 else if (addressfield.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please make sure you have key in everything before you proceed");  
 }  
 else{   
 if(printOutput()){  
 writeInput();  
 JOptionPane.showMessageDialog (null,"Successfully save");  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new MenuPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 }  
   
 } catch (NumberFormatException ae) {  
 JOptionPane.showMessageDialog (null, "Please make sure you key in the correct input");  
 }  
   
 }  
 });   
   
   
 }  
 public boolean printOutput(){  
   
 output += "Phone Number " + phonenumber.getText()+"\n";  
 output += "Address " + addressfield.getText() + "\n" ;  
   
 delivery.setText(output);  
 return true;  
 }  
   
 public void writeInput(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 delivery.setText(io.toString());  
 }  
  
 String input = "Phone Number : " +phonenumber.getText()+ "Address " +addressfield.getText();  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 delivery.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 delivery.setText(e.toString());  
 }  
 }  
 }  
}   
   
   
   
 class MenuPanel extends JPanel {  
   
 private JLabel menulbl;  
 private JLabel foodlbl;  
 private JLabel drinkslbl;  
 private JLabel f1lbl;  
 private JLabel f2lbl;  
 private JLabel f3lbl;  
 private JLabel foodprlbl;  
 private JLabel drinkprice;  
 private JLabel pf1lbl;  
 private JLabel pf2lbl;  
 private JLabel pf3lbl;  
 private JLabel d1lbl;  
 private JLabel d2lbl;  
 private JLabel jcomp14;  
 private JLabel pd1lbl;  
 private JLabel pd2lbl;  
 private JLabel pd3lbl;  
 private JLabel d4lbl;  
 private JLabel d5lbl;  
 private JLabel pd4lbl;  
 private JLabel pd5lbl;  
 private JLabel totallbl;  
 private JTextField jcomp23;  
 private JButton okbtn;  
 private JLabel quantityfood;  
 private JLabel quantitydrink;  
 private JTextField f1;  
 private JTextField f2;  
 private JTextField f3;  
 private JTextField d1;  
 private JTextField d2;  
 private JTextField d3;  
 private JTextField d4;  
 private JTextField d5;  
 private JLabel totaltaxlbl;  
 private JTextField totaltax;  
 private JLabel balancelbl;  
 private JTextField balancetext;  
 private JTextField moneytext;  
 private JLabel moneylbl;  
 private JButton pay;  
 private JButton cardoption;  
 private JButton gofeedback;  
 private JTextArea output\_field;  
 private JButton clear;  
   
 String output="";  
 String filePath="data.txt";  
  
 public MenuPanel(JFrame frame) {  
 //construct components  
 menulbl = new JLabel ("Purr Cafe Menu");  
 foodlbl = new JLabel ("Foods");  
 drinkslbl = new JLabel ("Drinks");  
 f1lbl = new JLabel ("Burger");  
 f2lbl = new JLabel ("Spaghetti");  
 f3lbl = new JLabel ("Fried Chicken");  
 foodprlbl = new JLabel ("Price");  
 drinkprice = new JLabel ("Price");  
 pf1lbl = new JLabel ("RM 2.50");  
 pf2lbl = new JLabel ("RM 5.00");  
 pf3lbl = new JLabel ("RM 3.00");  
 d1lbl = new JLabel ("Cola");  
 d2lbl = new JLabel ("Pepsi");  
 jcomp14 = new JLabel ("Sprite");  
 pd1lbl = new JLabel ("RM 2.00");  
 pd2lbl = new JLabel ("RM 2.00");  
 pd3lbl = new JLabel ("RM 2.00");  
 d4lbl = new JLabel ("Tea");  
 d5lbl = new JLabel ("Coffee");  
 pd4lbl = new JLabel ("RM 1.80");  
 pd5lbl = new JLabel ("RM 1.80");  
 totallbl = new JLabel ("Order Total : ");  
 jcomp23 = new JTextField (200);  
 okbtn = new JButton ("Ok");  
 quantityfood = new JLabel ("Quantity");  
 quantitydrink = new JLabel ("Quantity");  
 f1 = new JTextField (5);  
 f2 = new JTextField (5);  
 f3 = new JTextField (5);  
 d1 = new JTextField (5);  
 d2 = new JTextField (5);  
 d3 = new JTextField (5);  
 d4 = new JTextField (5);  
 d5 = new JTextField (5);  
 totaltax = new JTextField (100);  
 totaltaxlbl = new JLabel ("Total Inc Tax 5%");  
 balancelbl = new JLabel ("balance");  
 balancetext = new JTextField (5);  
 moneytext = new JTextField (5);  
 moneylbl = new JLabel ("key in your money here");  
 pay = new JButton ("Pay here to pay cash");  
 cardoption = new JButton ("Click here if you want to pay with Credit/Debit Card");  
 gofeedback = new JButton ("Make a feedback");  
 output\_field = new JTextArea(5,5);  
 clear = new JButton ("clear");  
 output\_field.setPreferredSize(new Dimension(15, 20));  
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (menulbl);  
 add (foodlbl);  
 add (drinkslbl);  
 add (f1lbl);  
 add (f2lbl);  
 add (f3lbl);  
 add (foodprlbl);  
 add (drinkprice);  
 add (pf1lbl);  
 add (pf2lbl);  
 add (pf3lbl);  
 add (d1lbl);  
 add (d2lbl);  
 add (jcomp14);  
 add (pd1lbl);  
 add (pd2lbl);  
 add (pd3lbl);  
 add (d4lbl);  
 add (d5lbl);  
 add (pd4lbl);  
 add (pd5lbl);  
 add (totallbl);  
 add (jcomp23);  
 add (okbtn);  
 add (quantityfood);  
 add (quantitydrink);  
 add (f1);  
 add (f2);  
 add (f3);  
 add (d1);  
 add (d2);  
 add (d3);  
 add (d4);  
 add (d5);  
 add (totaltax);  
 add (totaltaxlbl);  
 add (balancelbl);  
 add (balancetext);  
 add (moneytext);  
 add (moneylbl);  
 add (pay);  
 add (cardoption);  
 add (gofeedback);  
 add (output\_field);  
 add(clear);  
 //set component bounds (only needed by Absolute Positioning)  
 menulbl.setBounds (435, 20, 100, 25);  
 foodlbl.setBounds (160, 95, 100, 25);  
 drinkslbl.setBounds (600, 95, 100, 25);  
 f1lbl.setBounds (140, 120, 100, 25);  
 f2lbl.setBounds (140, 150, 100, 25);  
 f3lbl.setBounds (140, 180, 104, 25);  
 foodprlbl.setBounds (310, 95, 100, 25);  
 drinkprice.setBounds (700, 95, 100, 25);  
 pf1lbl.setBounds (305, 120, 100, 25);  
 pf2lbl.setBounds (305, 155, 100, 25);  
 pf3lbl.setBounds (305, 185, 100, 25);  
 d1lbl.setBounds (580, 115, 100, 25);  
 d2lbl.setBounds (580, 145, 100, 25);  
 jcomp14.setBounds (575, 175, 100, 25);  
 pd1lbl.setBounds (690, 115, 100, 25);  
 pd2lbl.setBounds (690, 150, 100, 25);  
 pd3lbl.setBounds (695, 180, 100, 25);  
 d4lbl.setBounds (580, 210, 100, 25);  
 d5lbl.setBounds (585, 245, 100, 25);  
 pd4lbl.setBounds (695, 210, 100, 25);  
 pd5lbl.setBounds (695, 245, 100, 25);  
 totallbl.setBounds (255, 395, 100, 25);  
 jcomp23.setBounds (400, 390, 200, 25);  
 okbtn.setBounds (385, 345, 100, 25);  
 quantityfood.setBounds (390, 90, 100, 25);  
 quantitydrink.setBounds (790, 95, 100, 25);  
 f1.setBounds (380, 120, 100, 25);  
 f2.setBounds (380, 155, 100, 25);  
 f3.setBounds (380, 190, 100, 25);  
 d1.setBounds (765, 120, 100, 25);  
 d2.setBounds (765, 150, 100, 25);  
 d3.setBounds (765, 180, 100, 25);  
 d4.setBounds (765, 210, 100, 25);  
 d5.setBounds (765, 245, 100, 25);  
 totaltaxlbl.setBounds (110, 430, 100, 25);  
 totaltax.setBounds (105, 465, 100, 25);  
 balancelbl.setBounds (740, 435, 100, 25);  
 balancetext.setBounds (715, 470, 100, 25);  
 moneytext.setBounds (400, 470, 100, 25);  
 moneylbl.setBounds (385, 445, 135, 25);  
 pay.setBounds (380, 415, 250, 20);  
 cardoption.setBounds (205, 505, 460, 45);  
 gofeedback.setBounds (740, 515, 195, 35);  
 output\_field.setBounds (20, 220, 170, 175);  
 clear.setBounds (20, 525, 105, 25);  
   
 clear.addActionListener(new ActionListener(){   
 public void actionPerformed(ActionEvent e){   
 moneytext.setText("");   
 balancetext.setText("");  
 totaltax.setText("");   
 f1.setText("");  
 f2.setText("");  
 f3.setText("");  
 d1.setText("");  
 d2.setText("");  
 d3.setText("");  
 d4.setText("");  
 d5.setText("");  
 output\_field.setText("");  
 jcomp23.setText("");  
 }   
 });  
   
 okbtn.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent e) {  
   
 try {  
 int a = Integer.parseInt (f1.getText());  
 int b = Integer.parseInt (f2.getText());  
 int c= Integer.parseInt (f3.getText());  
 int d = Integer.parseInt (d1.getText());  
 int x = Integer.parseInt (d2.getText());  
 int f= Integer.parseInt (d3.getText());  
 int g = Integer.parseInt (d4.getText());  
 int h= Integer.parseInt (d5.getText());  
   
 double total, totalwithtax, purchase1, purchase2, purchase3, purchase4, purchase5, purchase6, purchase7, purchase8;  
 purchase1 = a\*2.50;  
 purchase2 = b\*5.00;  
 purchase3 = c\*3.00;  
 purchase4 = d\*2.00;  
 purchase5 = x\*2.00;  
 purchase6 = f\*2.00;  
 purchase7 = g\*1.80;  
 purchase8 = h\*1.80;  
   
   
 total = purchase1 + purchase2 + purchase3 + purchase4 + purchase5 + purchase6 + purchase7 + purchase8;  
 totalwithtax = (total\*0.05)+total;  
   
 String thetotal = String.format ("%.2f", total);  
 jcomp23.setText(thetotal);  
 String totalwtax = String.format ("%.2f",totalwithtax);  
 totaltax.setText(totalwtax);  
   
   
  
 } catch (NumberFormatException ae) {  
 JOptionPane.showMessageDialog(null, "Please enter numbers only and don't leave any empty spaces");  
 }  
   
 }  
   
 });  
   
   
 pay.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent e){  
 try {  
 double c = Double.parseDouble (moneytext.getText());  
 double t = Double.parseDouble (totaltax.getText());  
 if(c<t) {  
 JOptionPane.showMessageDialog (null, "Sorry you don't have enough money");  
 }  
 else {  
 double balance = c-t;  
 String thebalance = String.format ("%.2f",balance);  
 balancetext.setText(thebalance);  
 }  
   
 }catch (NumberFormatException ae) {  
 JOptionPane.showMessageDialog (null, "Please make sure you have already submitted the menu then click ok & make sure your input is valid");  
 }  
 }  
 });  
   
 cardoption.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent b){  
 if ( f1.getText().equals("") || f2.getText().equals("") || f3.getText().equals("") || d1.getText().equals("") || d2.getText().equals("")||d3.getText().equals("")||d4.getText().equals("")||d5.getText().equals("")||totaltax.getText().equals("")||jcomp23.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please make sure to fill in the menu before you proceed with the payment and make sure to press ok to see the total you have to pay");  
 }  
 else if (moneytext.getText()!=""){  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new CreditDebitPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
   
 else {   
 JOptionPane.showMessageDialog (null, "You've already paid");  
   
 }  
 }  
 });  
   
   
   
   
   
 gofeedback.addActionListener (new ActionListener (){  
 public void actionPerformed (ActionEvent b){  
 if ( moneytext.getText().equals("") || f1.getText().equals("") || f2.getText().equals("") || f3.getText().equals("") || d1.getText().equals("") || d2.getText().equals("")||d3.getText().equals("")||d4.getText().equals("")||d5.getText().equals("")){  
 JOptionPane.showMessageDialog (null, "Please make sure to fill in the menu before you do the feedback");  
 }  
 else {  
 if(printOutput()){  
 writeInput();   
 JOptionPane.showMessageDialog(null,"Successfully Saved");  
 }  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new FeedbackPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 }  
 });  
   
 }  
   
   
   
   
 public boolean printOutput(){  
 output = "<html>";  
 output += "Thank you for ordering\n";  
 output += "Total inc tax:"+totaltax.getText()+"\n";  
 output += "Total You Paid"+moneytext.getText()+"\n";  
 output += "Your Balance:"+balancetext.getText()+"\n";  
   
 JOptionPane.showMessageDialog(null, "The data is saved successfully.", "INFORMATION",JOptionPane.INFORMATION\_MESSAGE);   
 output\_field.setText(output);  
 return true;  
 }  
   
 public void writeInput(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 output\_field.setText(io.toString());  
 }  
  
 String input = "total spent:" +totaltax.getText() + "total paid: " + moneytext.getText() + "balance : " + balancetext.getText() ;  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 output\_field.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 output\_field.setText(e.toString());  
 }  
 }  
 }  
}  
   
   
   
   
   
 class CreditDebitPanel extends JPanel {  
 private JLabel creditdebitlbl;  
 private JLabel fname;  
 private JLabel lastnsme;  
 private JLabel cardnum;  
 private JLabel expired;  
 private JTextField fnamee;  
 private JTextField lnamee;  
 private JTextField cardd;  
 private JComboBox <String> year ;  
 private JComboBox <String> month ;  
 private JLabel cvv;  
 private JTextField cvvtext;  
 private JButton finish3;  
 private JTextArea credit;  
 private JButton clear;  
 String output = "";  
 public CreditDebitPanel(JFrame frame) {  
 //construct preComponents  
 String[] yearItems = {"[YEAR]", "2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029", "2030", "2031", "2032", "2035", "2036", "2037", "2038", "2039", "2040"};  
 String[] monthItems = {"[MONTH]", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"};  
  
 //construct components  
 creditdebitlbl = new JLabel ("Credit / Debit Card");  
 fname = new JLabel ("First Name");  
 lastnsme = new JLabel ("Last Name");  
 cardnum = new JLabel ("Card Number");  
 expired = new JLabel ("Expiry Date");  
 fnamee = new JTextField (5);  
 lnamee = new JTextField (5);  
 cardd = new JTextField (5);  
 year = new JComboBox <String> (yearItems);  
 month = new JComboBox <String> (monthItems);  
 cvv = new JLabel ("CVV");  
 cvvtext = new JTextField (5);  
 credit = new JTextArea (5, 5);  
 clear = new JButton ("clear");  
   
 finish3 = new JButton ("Make a feedback");  
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (creditdebitlbl);  
 add (fname);  
 add (lastnsme);  
 add (cardnum);  
 add (expired);  
 add (fnamee);  
 add (lnamee);  
 add (cardd);  
 add (year);  
 add (month);  
 add (cvv);  
 add (cvvtext);  
 add (finish3);  
 add (credit);  
 add(clear);  
   
   
   
  
 //set component bounds (only needed by Absolute Positioning)  
 creditdebitlbl.setBounds (380, 60, 121, 25);  
 fname.setBounds (105, 110, 100, 25);  
 lastnsme.setBounds (105, 140, 100, 25);  
 cardnum.setBounds (105, 170, 100, 25);  
 expired.setBounds (95, 210, 100, 25);  
 fnamee.setBounds (195, 110, 150, 25);  
 lnamee.setBounds (195, 140, 150, 25);  
 cardd.setBounds (195, 175, 150, 25);  
 year.setBounds (345, 210, 100, 25);  
 month.setBounds (195, 210, 100, 25);  
 cvv.setBounds (100, 245, 100, 25);  
 cvvtext.setBounds (190, 250, 100, 25);  
 finish3.setBounds (400, 495, 200, 25);  
 credit.setBounds (765, 120, 170, 255);  
 clear.setBounds (20, 525, 105, 25);  
   
 clear.addActionListener(new ActionListener(){   
 public void actionPerformed(ActionEvent e){   
 fnamee.setText("");   
 lnamee.setText("");  
 cardd.setText("");   
 cvvtext.setText("");  
 year.setSelectedIndex(0);  
 month.setSelectedIndex(0);  
  
 }   
 });  
  
   
   
   
   
   
   
 finish3.addActionListener (new ActionListener(){  
 public void actionPerformed (ActionEvent e){  
 if (fnamee.getText().equals("") || lnamee.getText().equals("") || cardd.getText().equals("")|| year.getSelectedItem().equals("[YEAR]") || month.getSelectedItem().equals ("[MONTH]") || cvvtext.getText().equals("")){  
 JOptionPane.showMessageDialog (null,"Make sure you have already key in all inputs before you proceed");  
 }  
 else{  
 if(printOutput()){  
 writeInput();  
 JOptionPane.showMessageDialog(null,"Successfully Save");  
 }  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new FeedbackPanel(frame));   
 frame.pack();  
 frame.setVisible (true);  
 }  
 }  
 });  
   
}  
  
   
   
 public boolean printOutput(){  
   
 output += "First name : " +fnamee.getText()+ "\n";  
 output += "Last name : " +lnamee.getText()+ "\n";  
 output += "Card Number : " +cardd.getText()+"\n";  
 output += "CVV : " +cvvtext.getText()+"\n";  
   
 credit.setText(output);  
 return true;  
 }  
   
 public void writeInput(){  
 BufferedReader reader;  
 BufferedWriter br = null;  
 String data="";  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 data += line+"\n";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 credit.setText(io.toString());  
 }  
  
 String input = "First Name : " +fnamee.getText() + "Last Name " +lnamee.getText() + "Card Number " +cardd.getText() + "Expiry Date : Year " +year.getSelectedItem()+ " CVV " +cvvtext.getText();  
 try {  
 br = new BufferedWriter(new FileWriter("purrcafe.txt"));  
 br.write(data+input);  
 } catch (IOException e) {   
 credit.setText(e.toString());  
 } finally {  
 try {  
 br.close();  
 } catch (IOException e) {  
 credit.setText(e.toString());  
 }  
 }  
 }  
 }  
   
 class FeedbackPanel extends JPanel {  
 private JLabel fblbl;  
 private JLabel ratelbl;  
 private JRadioButton goodlbl;  
 private JRadioButton finelbl;  
 private JRadioButton badlbl;  
 private JTextArea feedback;  
 private JButton load;  
   
 String output="";  
 String filePath="data.txt";  
 String rb\_selection="";  
   
 public FeedbackPanel(JFrame frame) {  
 //construct components  
 fblbl = new JLabel ("Feedback");  
 ratelbl = new JLabel ("How woulf you rate us?");  
 load = new JButton ("Load Data");  
   
 goodlbl = new JRadioButton ("Very Good :D");  
 //implement action listener for goodlbl  
 goodlbl.addItemListener (new ItemListener () {  
 public void itemStateChanged (ItemEvent e) {  
 JOptionPane.showMessageDialog (null, "Thank you very much ! Have a nice day");  
 int result = JOptionPane.showConfirmDialog(null , "Do you want to exit?", "Close",  
 JOptionPane.YES\_NO\_OPTION,  
 JOptionPane.QUESTION\_MESSAGE);  
 if(result == JOptionPane.YES\_OPTION){  
 System.exit(0);  
 }  
 }  
 });  
 finelbl = new JRadioButton ("Just Fine :)");  
 //implement action listener for finelbl  
 finelbl.addItemListener (new ItemListener () {  
 public void itemStateChanged (ItemEvent e) {  
  
 JOptionPane.showMessageDialog (null, "We will improve, for better performance next time :)");  
 int result = JOptionPane.showConfirmDialog(null , "Do you want to exit?", "Close",  
 JOptionPane.YES\_NO\_OPTION,  
 JOptionPane.QUESTION\_MESSAGE);  
 if(result == JOptionPane.YES\_OPTION){  
 System.exit(0);  
 }  
  
 }  
 });  
  
 badlbl = new JRadioButton ("Bad :(");  
 //implement action listener for badlbl  
 badlbl.addItemListener (new ItemListener () {  
 public void itemStateChanged (ItemEvent e) {  
 JOptionPane.showMessageDialog (null, "We truly apologize, we promise that you won't experience this bad performance again ! :)");  
 int result = JOptionPane.showConfirmDialog(null , "Do you want to exit?", "Close",  
 JOptionPane.YES\_NO\_OPTION,  
 JOptionPane.QUESTION\_MESSAGE);  
 if(result == JOptionPane.YES\_OPTION){  
 System.exit(0);  
 }  
  
 }  
 });  
   
  
   
  
 //adjust size and set layout  
 setPreferredSize (new Dimension (944, 563));  
 setLayout (null);  
  
 //add components  
 add (fblbl);  
 add (ratelbl);  
 add (goodlbl);  
 add (finelbl);  
 add (badlbl);  
 add(load);  
  
 //set component bounds (only needed by Absolute Positioning)  
 fblbl.setBounds (400, 80, 100, 25);  
 ratelbl.setBounds (355, 110, 195, 41);  
 goodlbl.setBounds (180, 170, 100, 25);  
 finelbl.setBounds (375, 170, 100, 25);  
 badlbl.setBounds (585, 170, 100, 25);  
 load.setBounds (345, 430, 205, 45);  
   
 load.addActionListener(new ActionListener(){  
 public void actionPerformed(ActionEvent e){  
 if(rb\_selection.equals("")){  
 JOptionPane.showMessageDialog (null, "Please select your feedback before you want to load your data");  
 }  
 else{  
 frame.getContentPane().removeAll();  
 frame.getContentPane().add (new load());   
 frame.pack();  
 frame.setVisible (true);  
 }  
  
 }   
 });  
   
   
   
   
   
   
 }  
 }  
   
 class load extends JPanel{  
 private JLabel title ;  
 private JLabel p;  
 private JScrollPane jsp;  
 private Border border = BorderFactory.createLineBorder(Color.BLACK, 1);  
  
 public load(){  
 title = new JLabel("Purr Cafe Record");  
 p = new JLabel("");  
 p.setPreferredSize(new Dimension(667, 366));  
 p.setBorder(border);  
 p.setVerticalAlignment(JLabel.TOP);  
 jsp = new JScrollPane(p);  
 String output="<html>";  
 add (title);  
 add (p);  
 add (jsp);  
   
 title.setBounds(210, 10, 200, 25);  
 p.setBounds(45, 50, 500, 205);  
 BufferedReader reader;  
 try {  
 reader = new BufferedReader(new FileReader("purrcafe.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 output += line+"<br>";  
 line = reader.readLine();  
 }  
 reader.close();  
 } catch (IOException io) {  
 JOptionPane.showMessageDialog(null,io.toString());  
 }  
 p.setText(output);  
 }  
}  
 public class OrderingRecordGUI {  
 public static void main (String[] args) {  
   
 JFrame frame = new JFrame ("Purr Cafe");  
 frame.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);  
   
   
   
 frame.getContentPane().add (new NamePanel(frame));  
 frame.getContentPane().repaint();  
 frame.pack();  
 frame.setVisible (true);  
   
   
   
 }   
   
 }  
  
2) Object Oriented Concept Implimentation

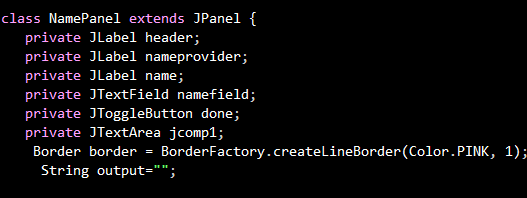
In this project, I have implemented a few object oriented implementations.

a) Inheritence



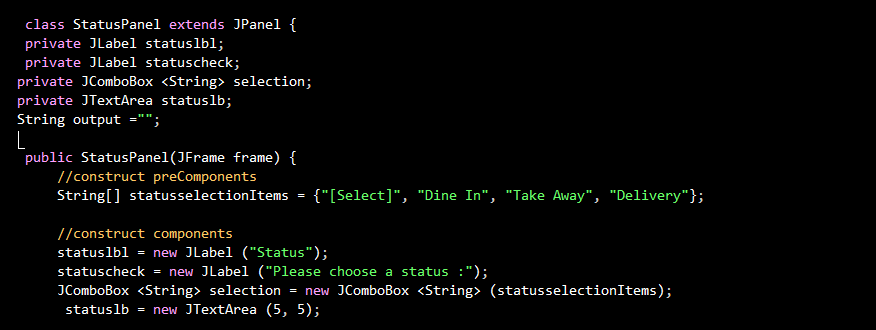
NamePanel is inheriting JPanel. The NamePanel extends from JPanel.  An inherited class is called a subclass of its parent class or super class.

b) Encapsulation



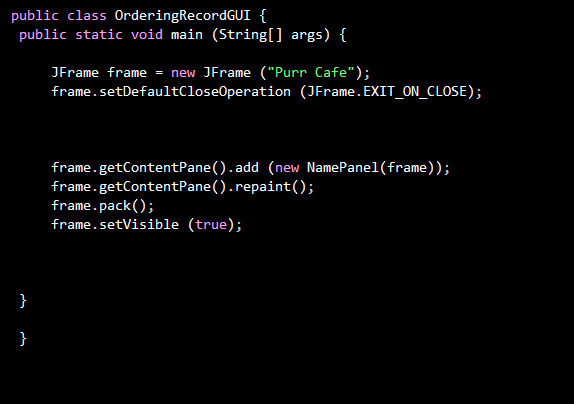
 the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. The variables of the class NamePanel are all private, which proves that this is an implementation of an OOP concept which is encapsulation.

c) Interface and Inner class

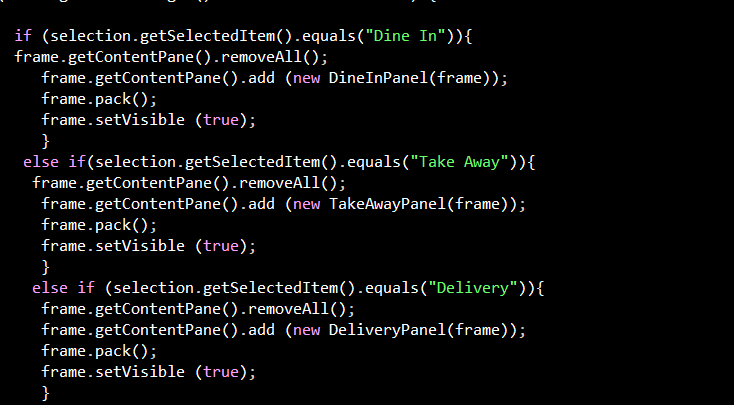


Class StatusPanel is the interface and public StatusPanel (Jframe frame) is the inner class . An inner class is defined within another class or within a method of some class interfaces are used to define protocols for user defined functionality. Interfaces define first class types and as such can be used as the types of variables and parameters.

d) Classes and *Objects*

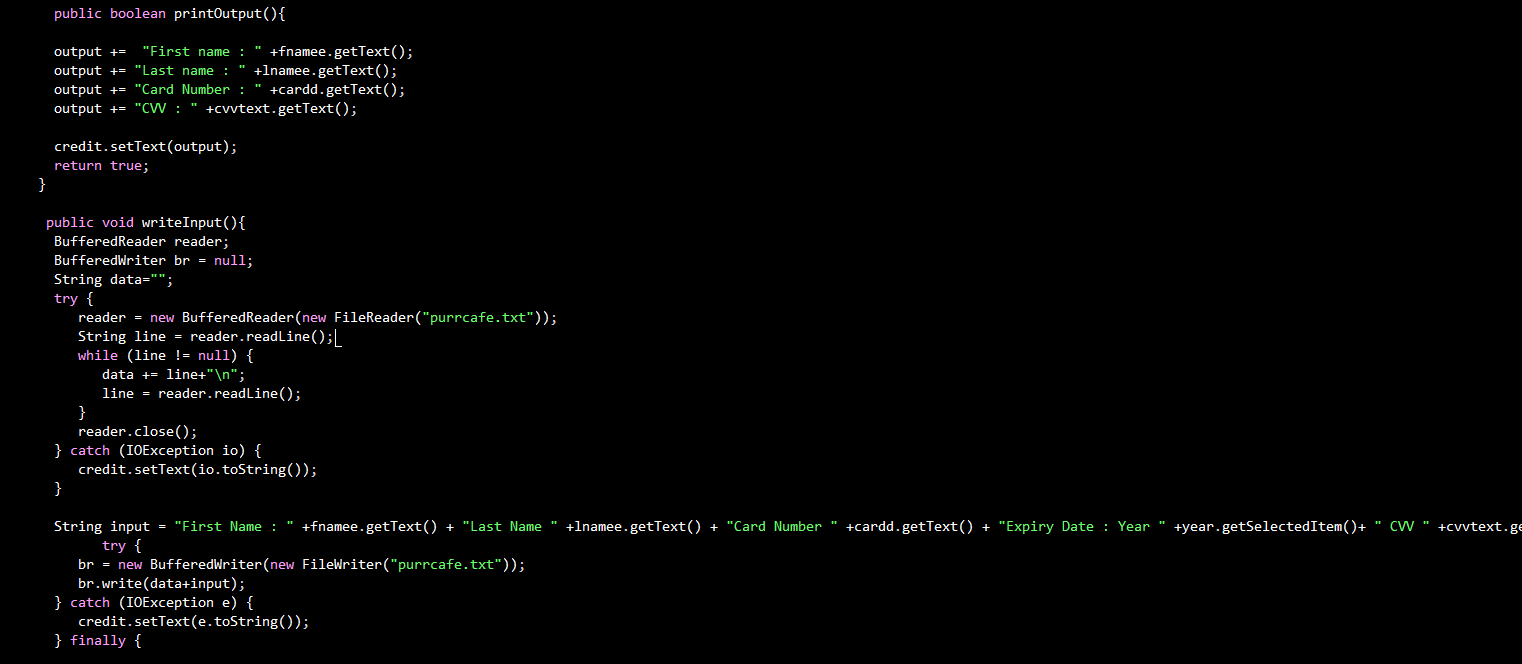


Classes are used to create and manage new objects and support inheritance. This program uses classes and objects. This class have main (String args[]) method which is runnable (executable). The object in this class is the JFrame which is written as JFrame frame = new JFrame(“Purr Cafe”) which creates a frame labeled Purr Café. This is the main method of this program. The JFrame is used as a frame for every panel in this program. For example, NamePanel , DineInPanel, MenuPanel.

e) Polymorphism

Method overloading is an **example** of static **polymorphism**, as we can see here the method getContentPane overloads in the class StatusPanel. Polymorphism is the ability to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes. In this program, the getContent pane is overriding. Method overriding occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.

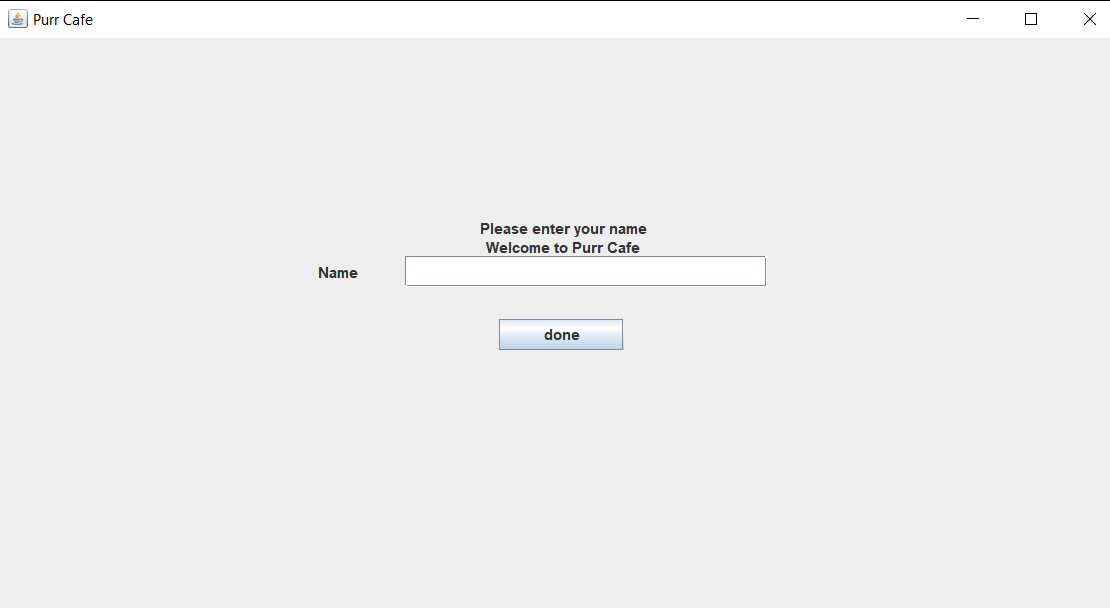
3)Read and Write



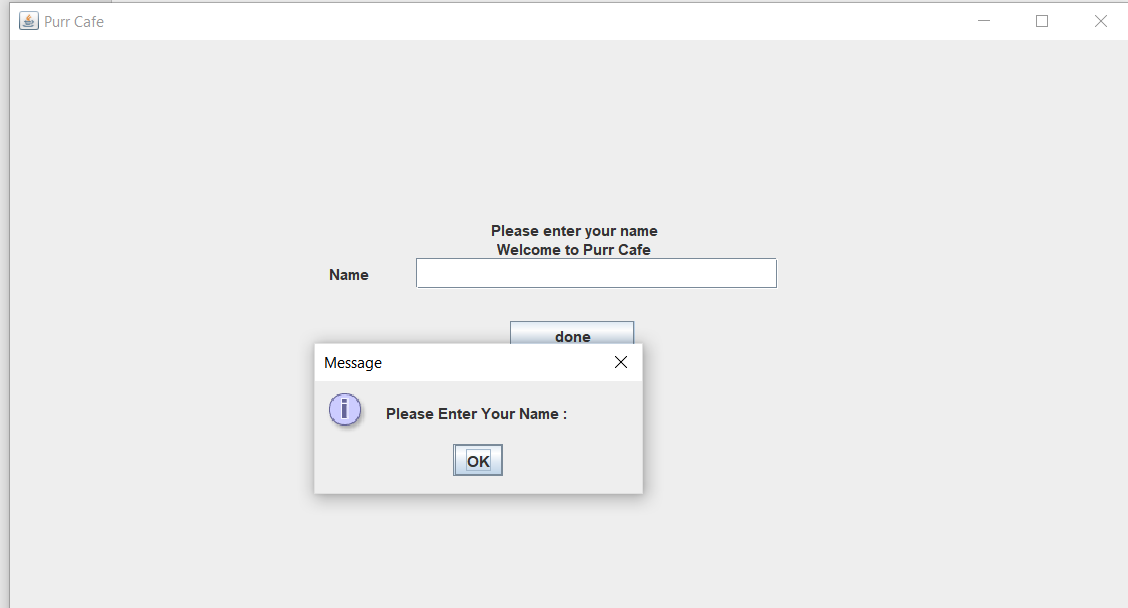
This is an example of the read and write OOP implementation from the Order Record, this implementation is capable of being displayed (read) and modified (written to). Most objects (disks, files, directories) are read/write, but operating systems also allow you to protect objects with a read-only attribute that prevents other users from modifying the object. This program has a textfile called “purrcafe.txt” . From the method writeinput(), The FileReader will read then it will be written in by the FileWriter .

4) User Manual

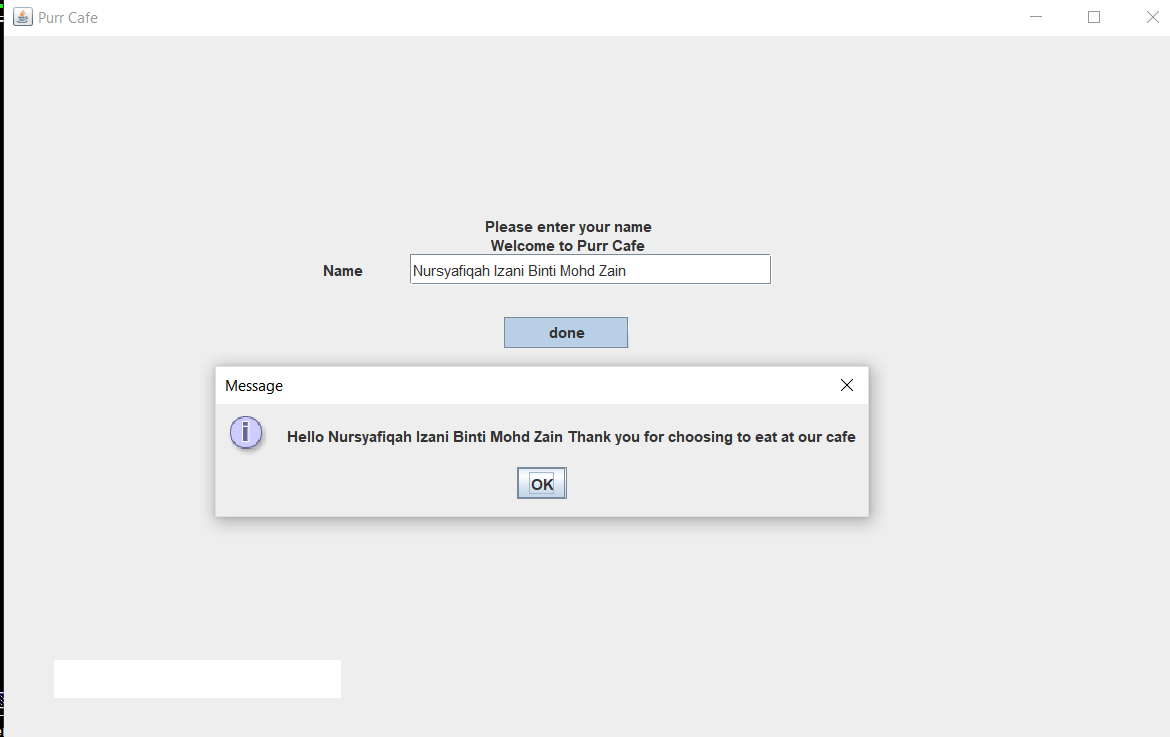
1.Firstly, when you run the program you will see the NamePanel in the Purr Cafe frame



2. If the user does not input any name, there will be message dialog to make sure you insert your name

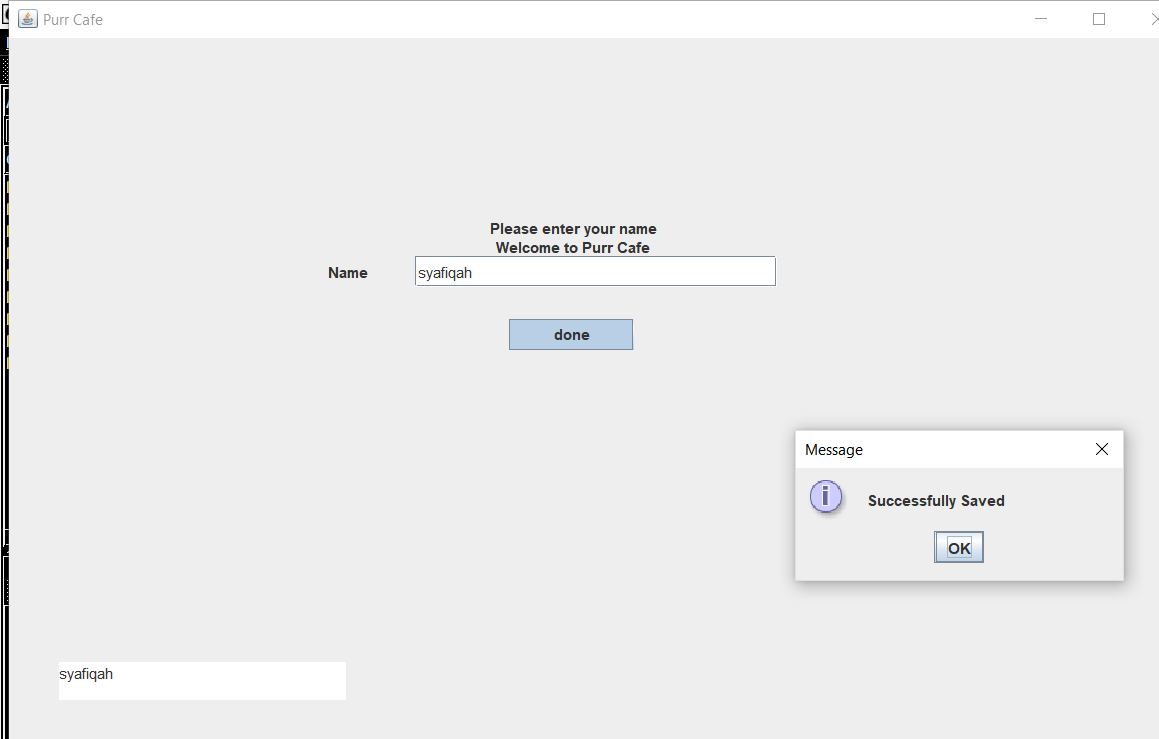


3. If you have successfully entered your name there will be a message dialog thanking you for choosing Purr Café



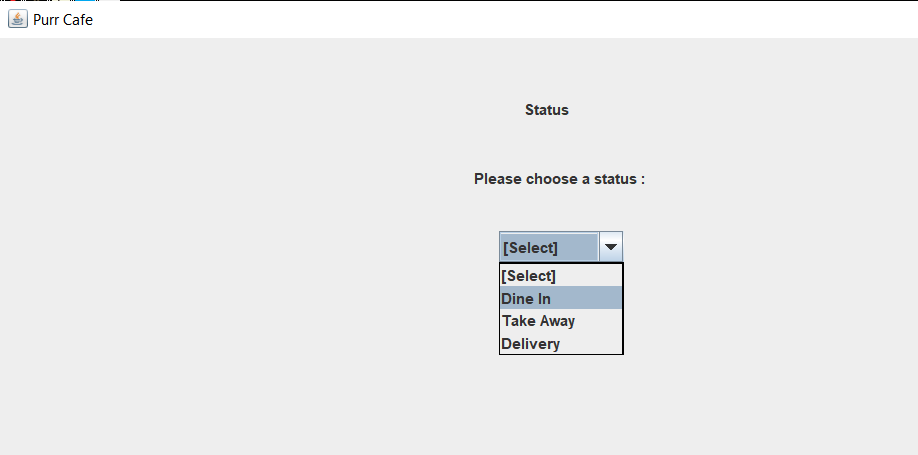
4. If you click the button ok from the previous message dialog, the program then will save the data of your name in the Purrcafe.txt file. It also shows the output on the text field at lower left corner of the frame.

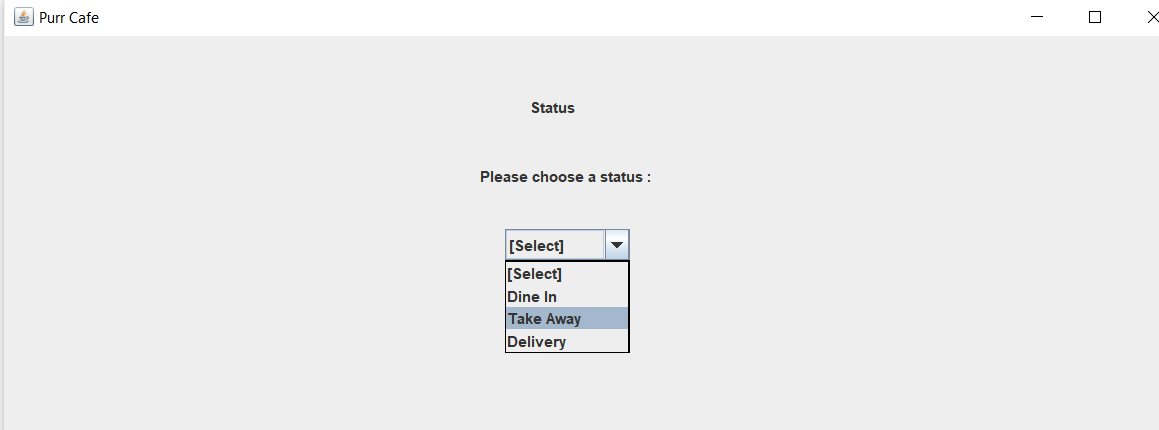
You’ll next go to the StatusPanel

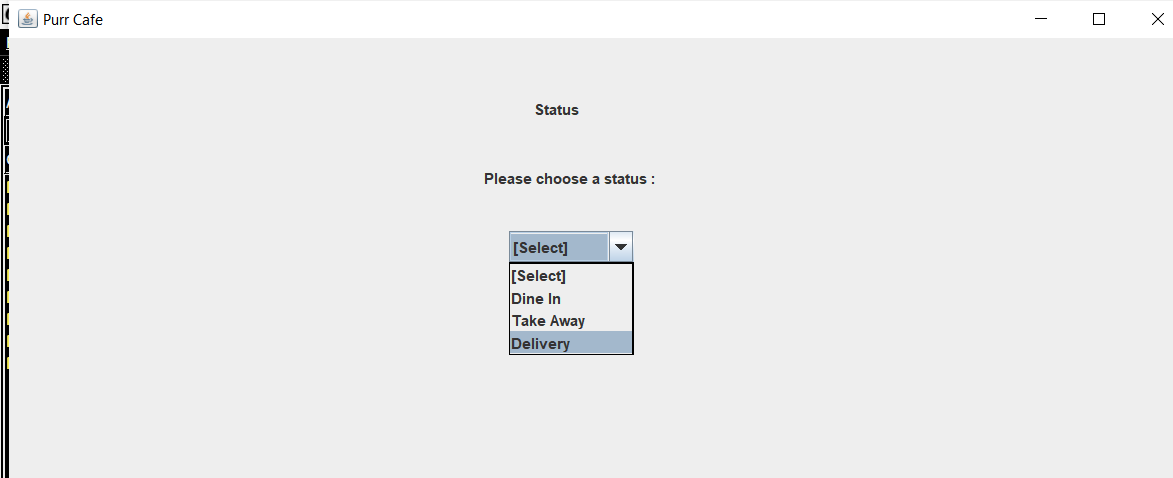


5. After you already submitted your name in the NamePanel you will then see the StatusPanel at the Purr Café frame. You, will select your choice of status to dine in, take away or delivery. You’ll be taken to the DineInPanel,

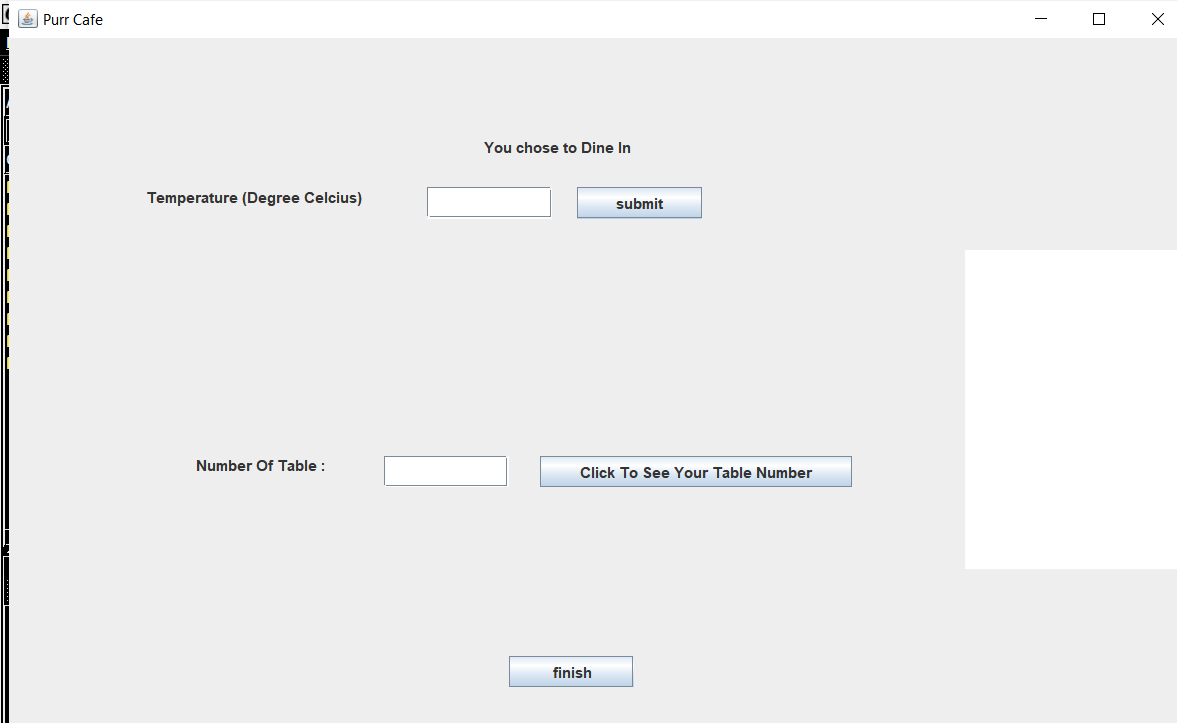
If you choose Dine In, you’ll go to the TakeAwayPanel if you choose Take Away you’ll go to the DeliveryPanel if you chose Delivery.



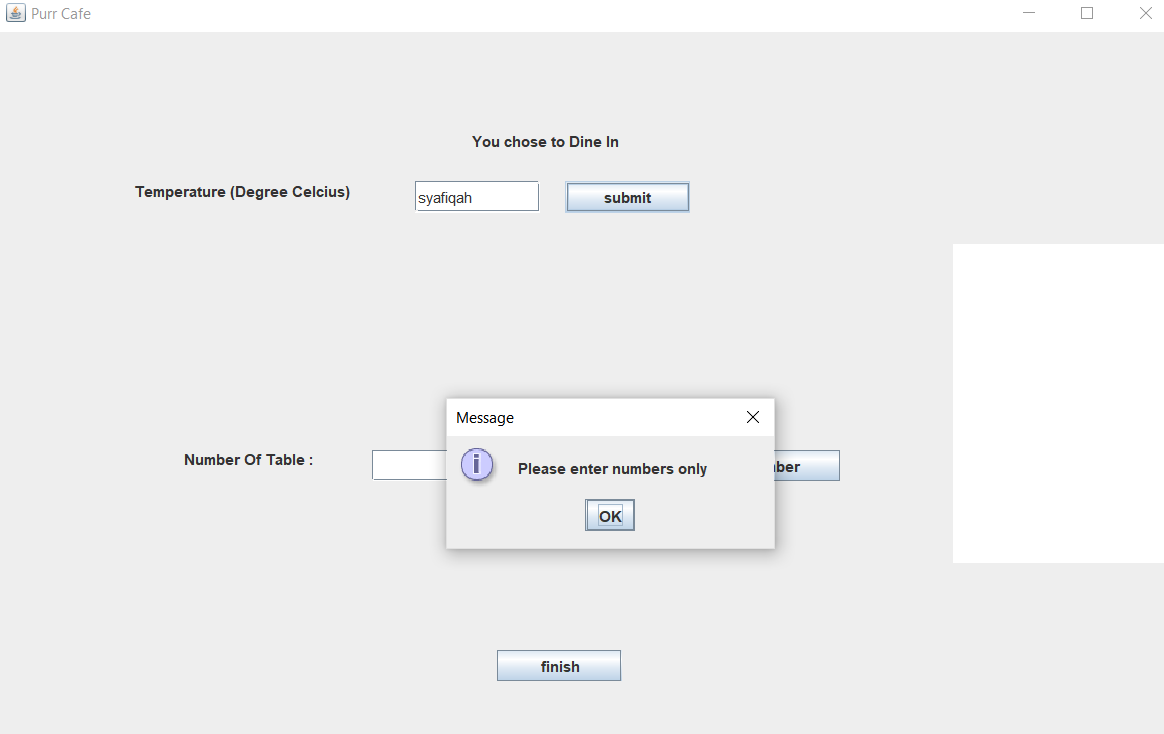


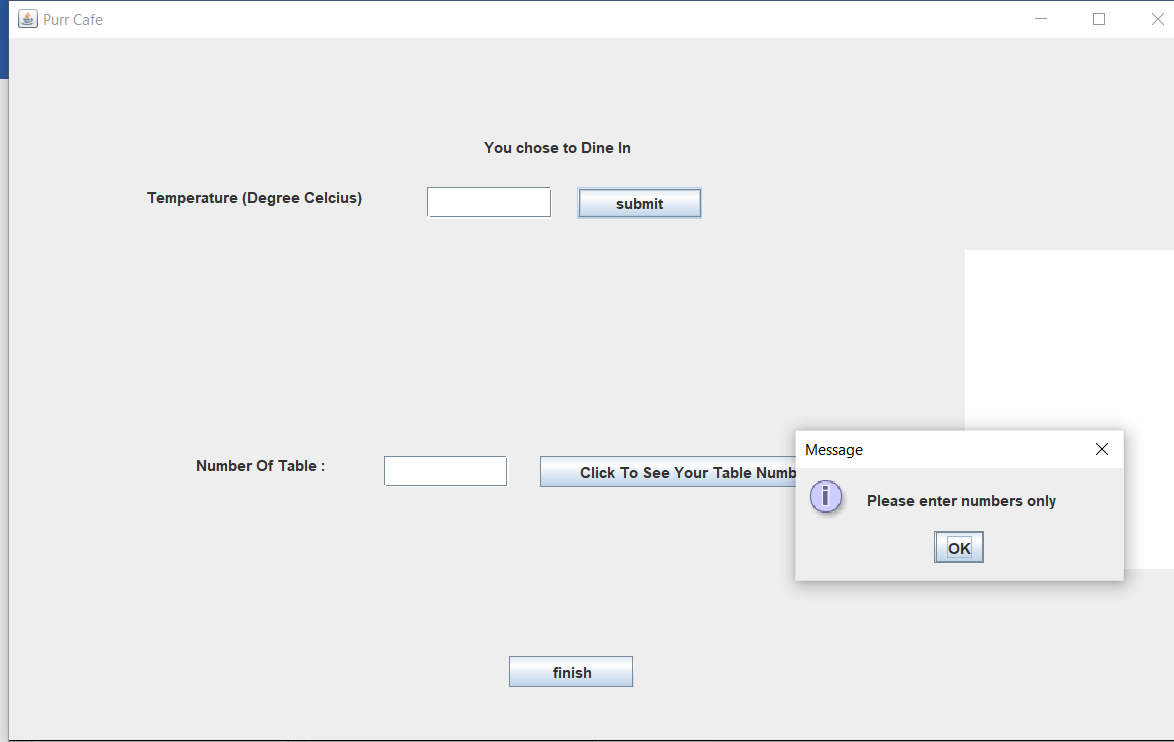


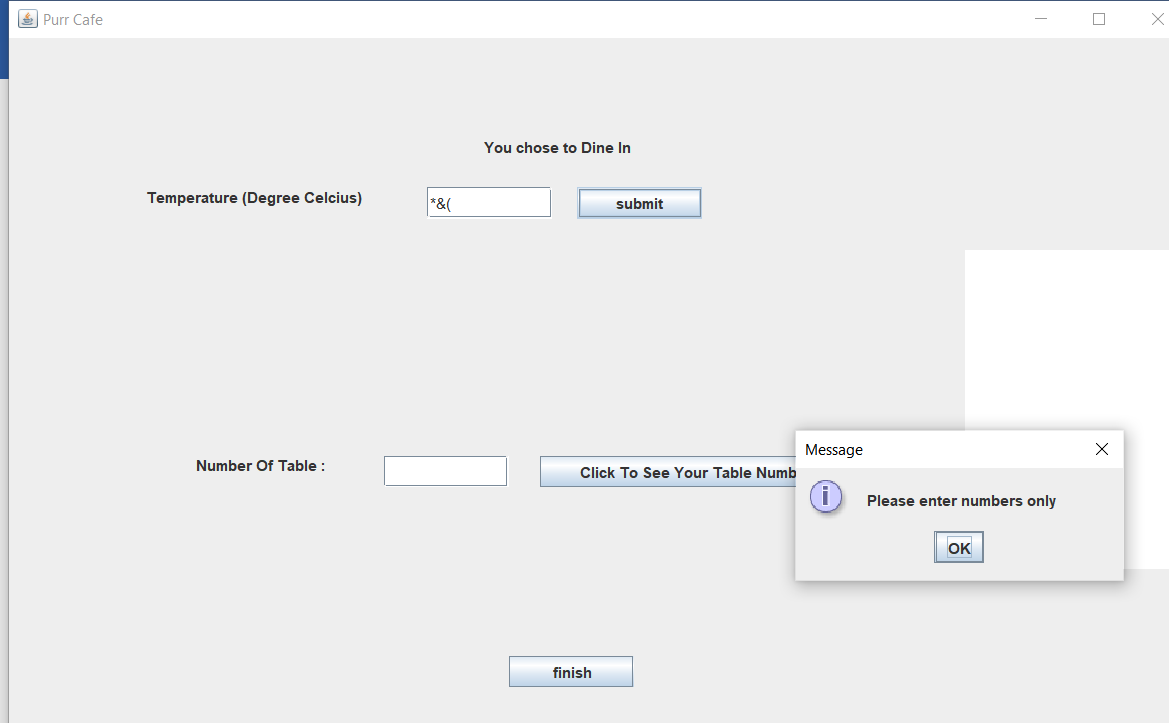
6. If you chose Dine in, the DineInPanel will show up at the Purr Café frame



7.If you click the submit with values that are not numbers for instance like symbols, strings or you don’t enter anything there will be a message dialog telling you key in the valid input.

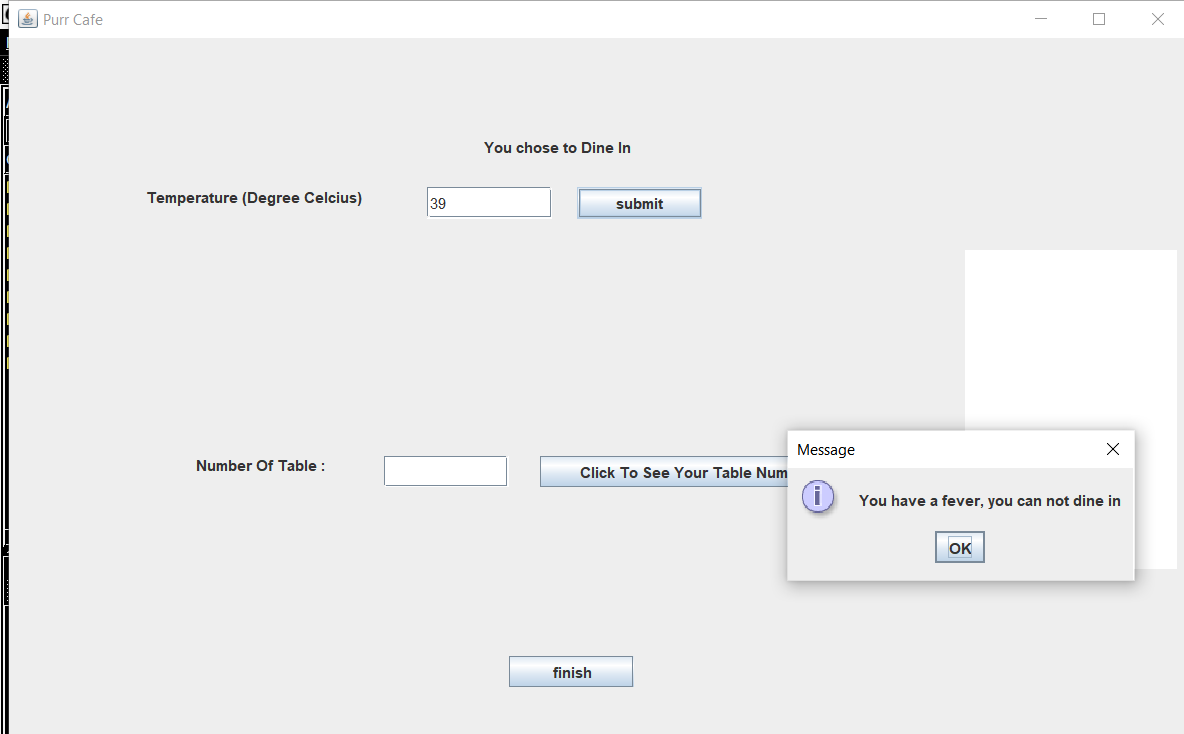






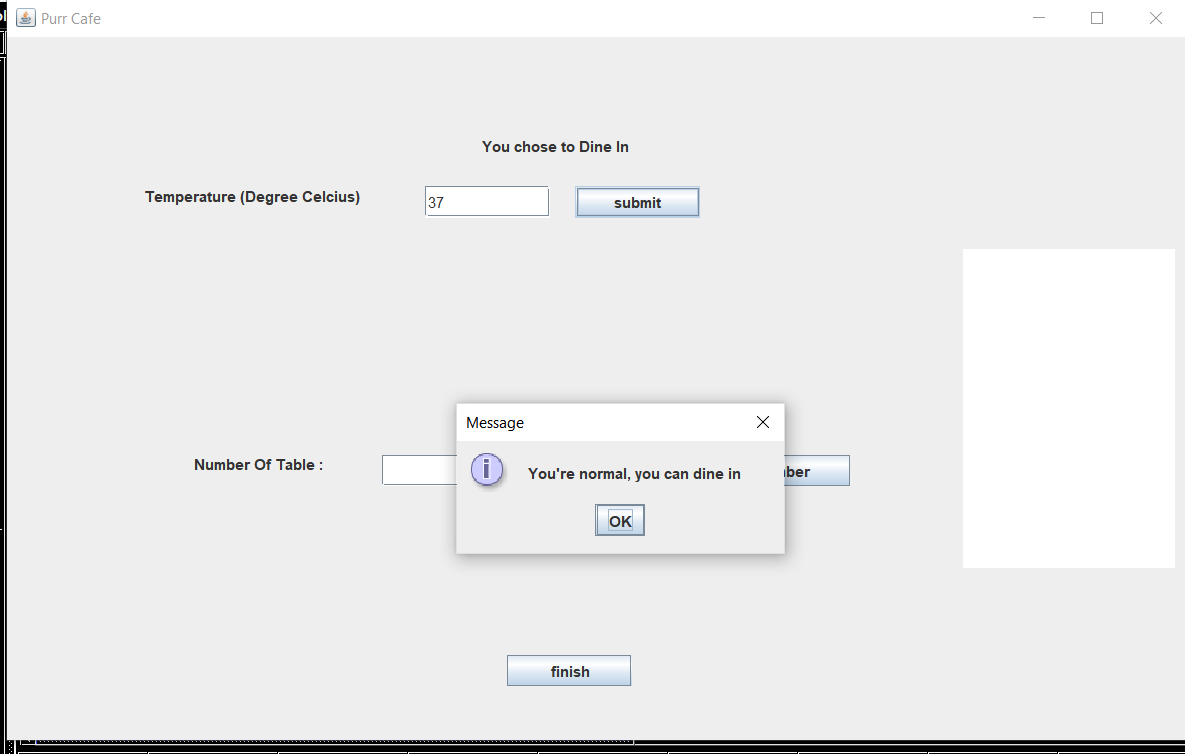
8. Since the coronavirus issue is concerning, there will be message shown if the temperature you key in is above 37.5 degree celcius saying that you have a fever and can not dine in to prevent the high risk of coronavirus.

Once you clicked ok at the message bar you will be exited from the program.

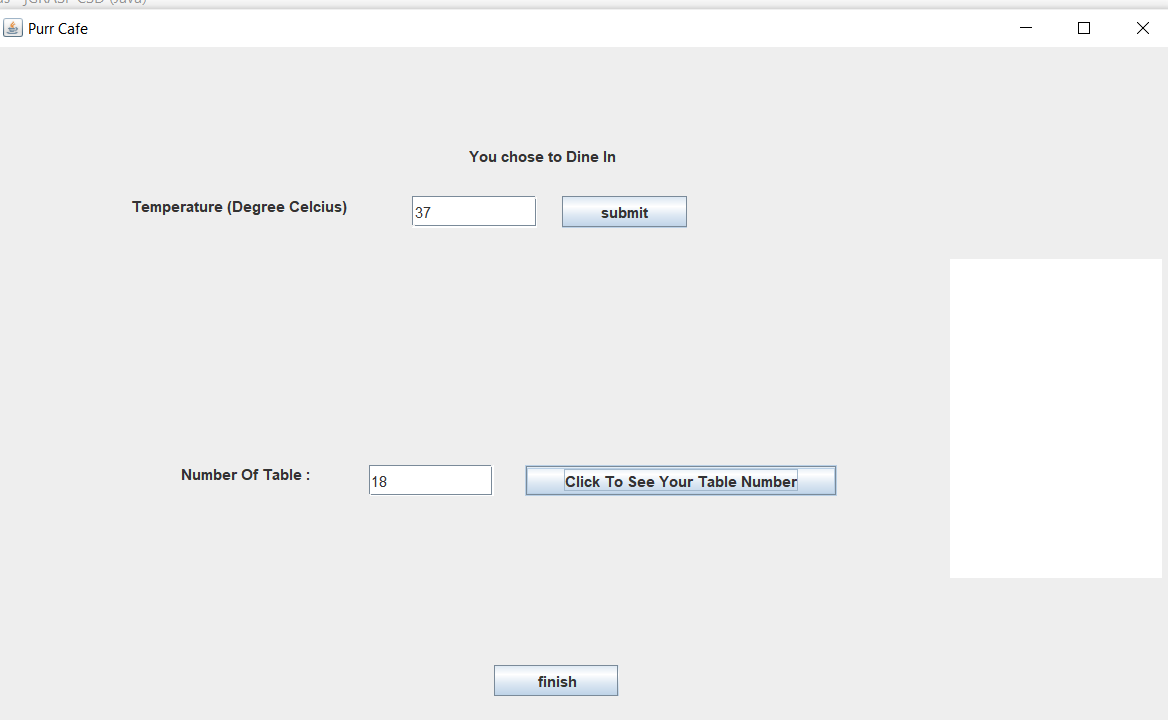


9. When you’re temperature is below 37.5 you can successfully dine in in the café.

9. When your temperature is 37.5 there will be a message saying that you can dine in in the restaurant because the risk of you having coronavirus is low.

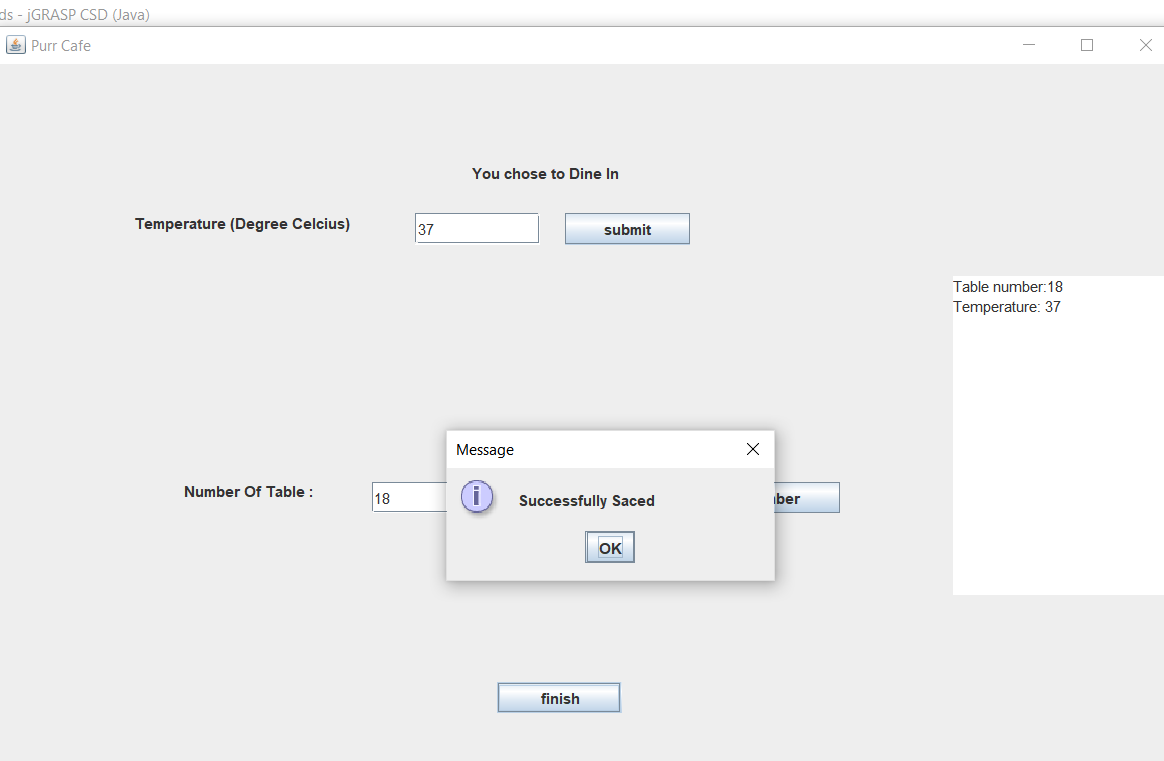


10. Next, you must click the table number to see your table number.

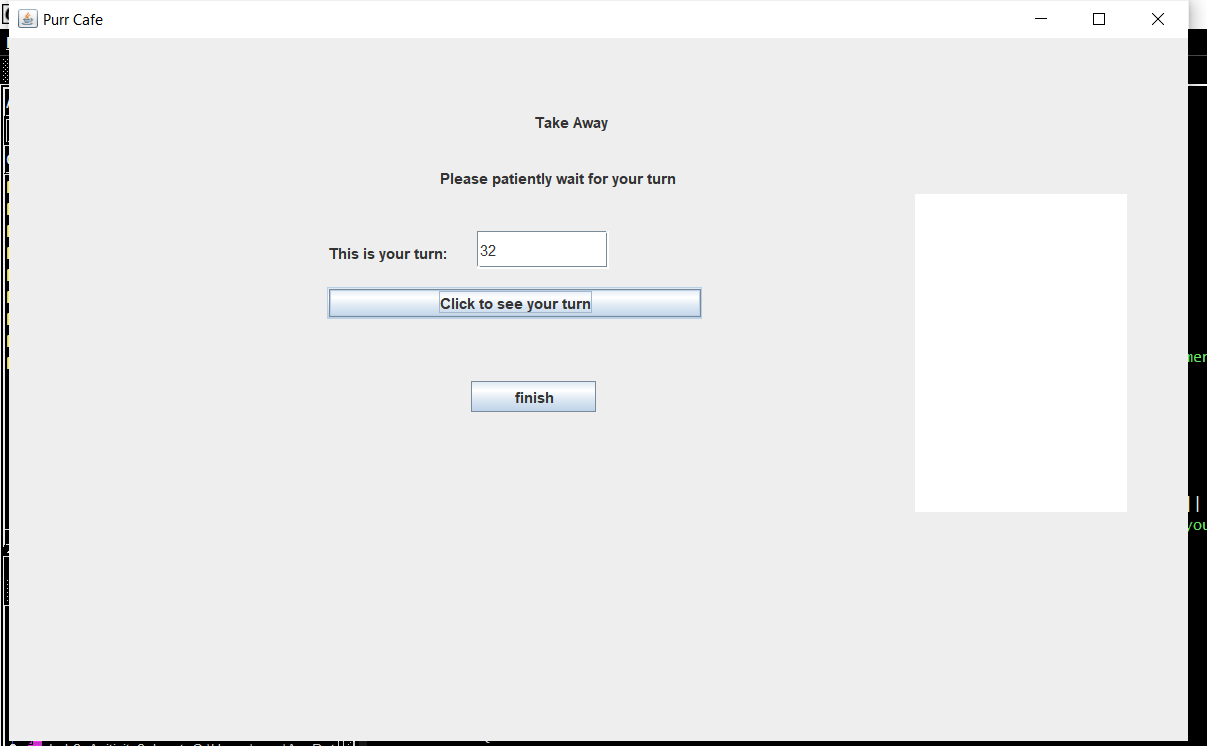


11. If you already submitted all the inputs, the inputs will be saved into a file and will be displayed at the text area on the right corner of the frame.

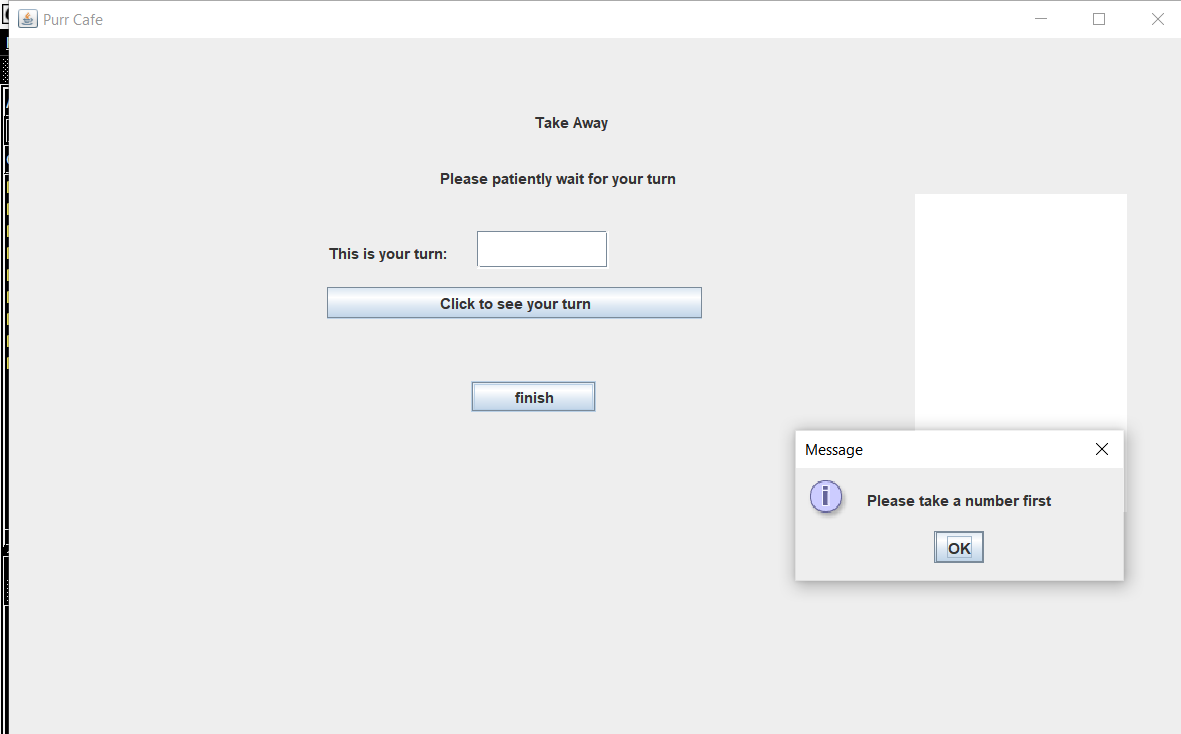
You’ll be taken to the MenuPanel



12. If you choose to take away from the status panel, you will taken to TakeAwayPanel and you will be asked to click the “Click to see your turn” to see your turn.

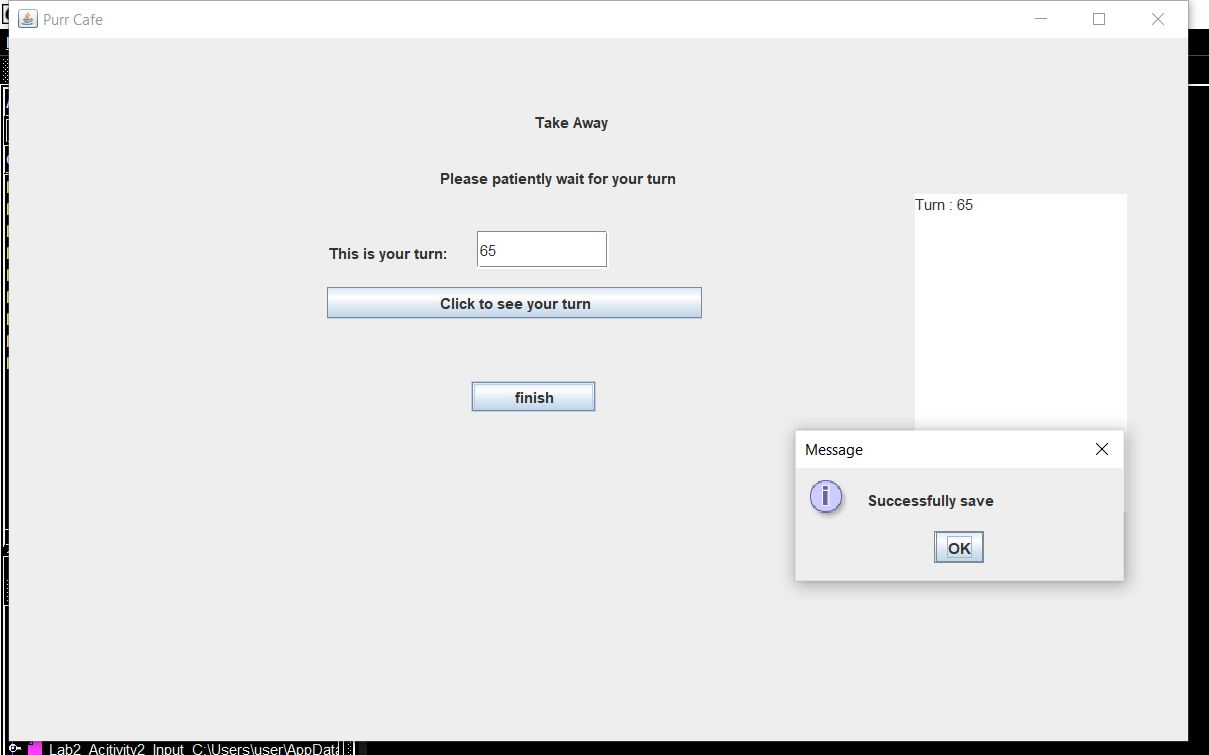


13. If you don’t click, the “Click to see your turn” button there will be message asking you to click the button



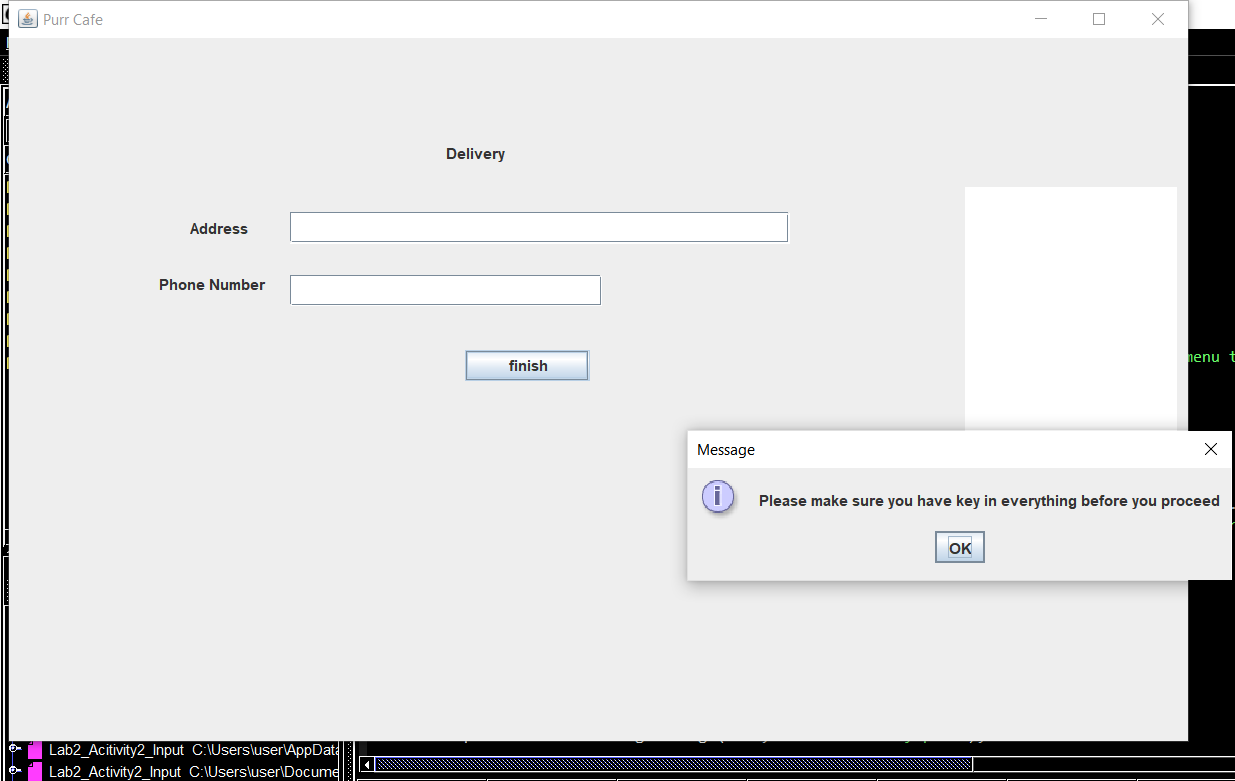
If you already submitted everything, the program will save the data and save it to your Purrcafetxt file and display it the text area on the right side of the panel

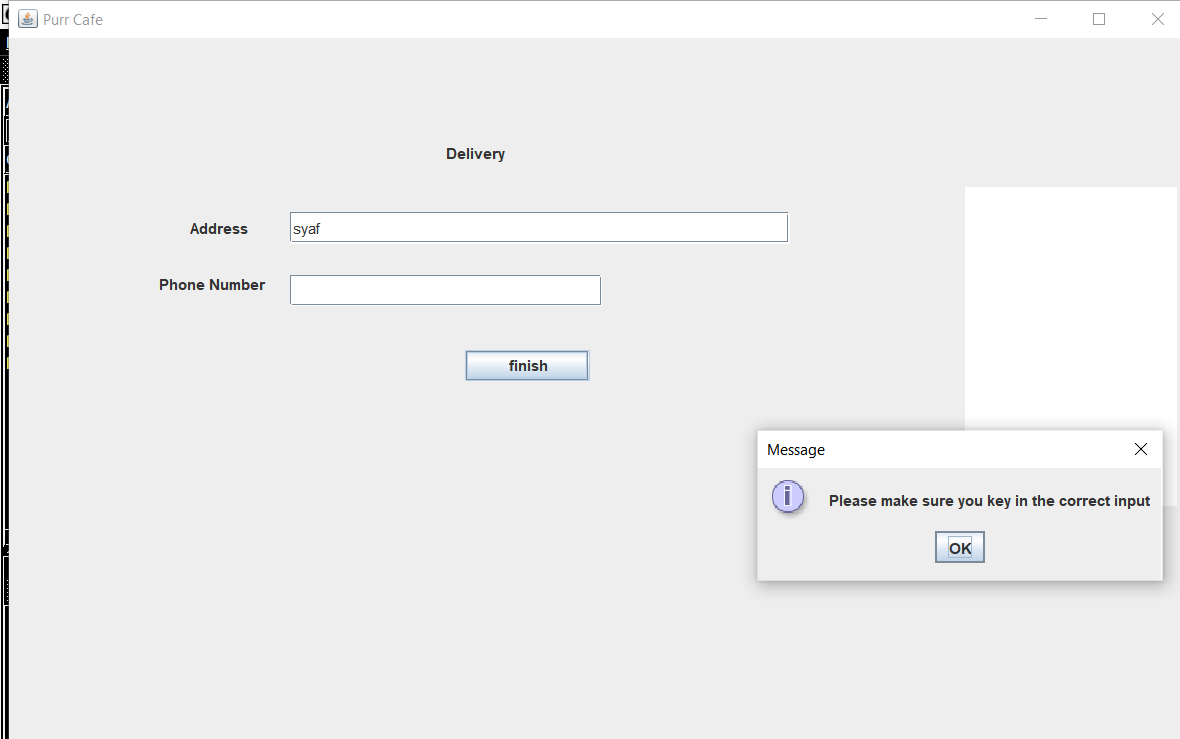
After that, you’ll be taken to MenuPanel



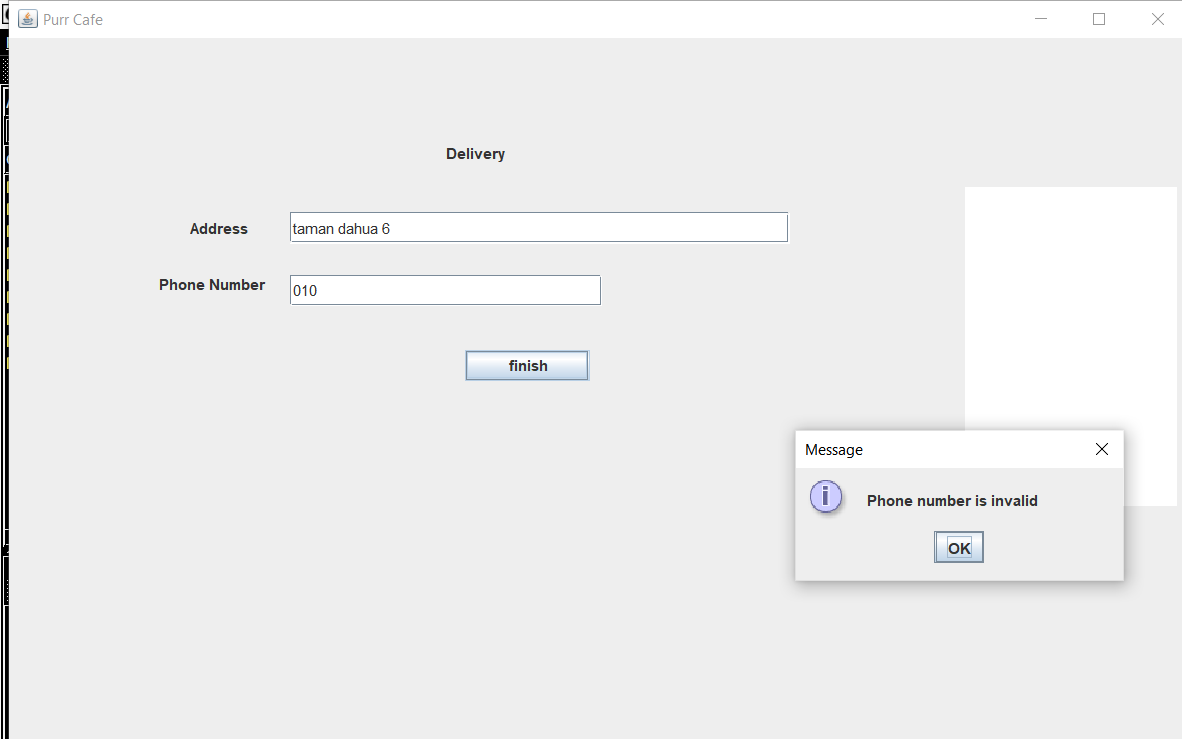
14. If you chose delivery you will enter the Delivery Panel, where you must submit your address and phone number.

If you don’t key in any input or one of the inputs there will be message showing that you must key in the required inputs.





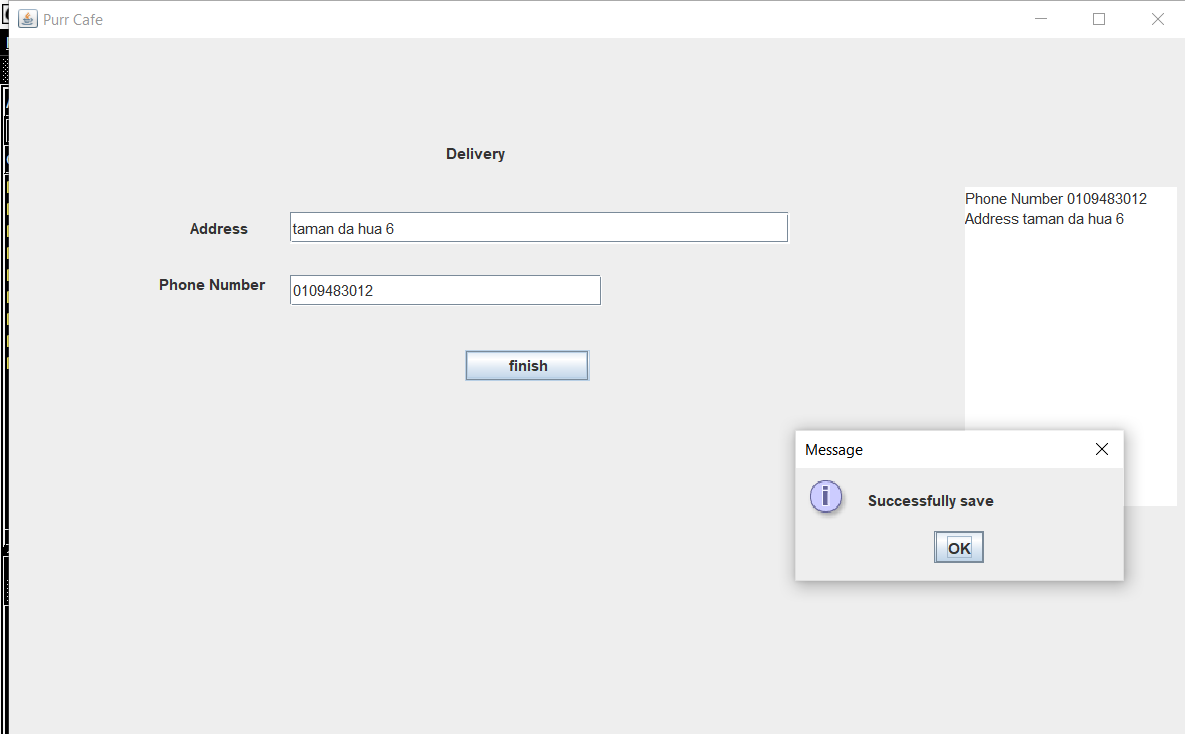
14. If you key in an invalid phone number there will be a message showing that you have key in an invalid phone number.



15.If you have successfully key in all data correctly, click the finish button to save your file and proceed

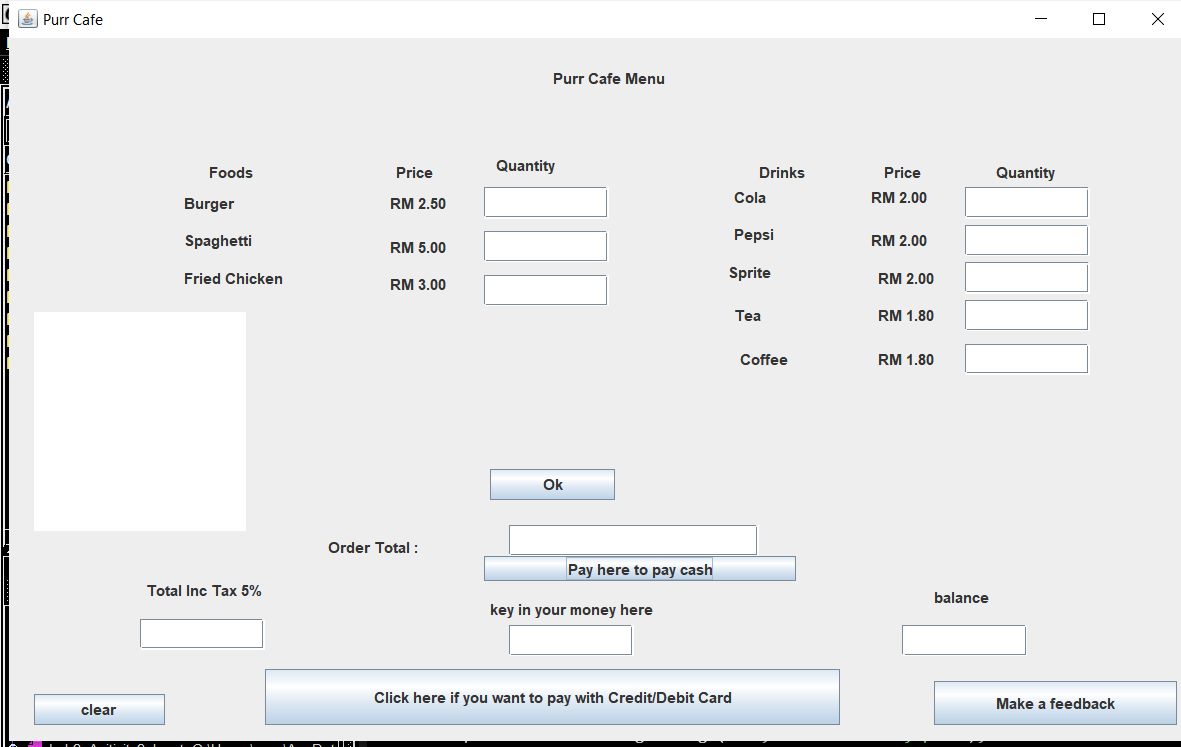
There will be message saying that you have successfully saved your data to a file and will be displayed at the right corner of your frame.

After that, you”ll be taken to the MenuPanel

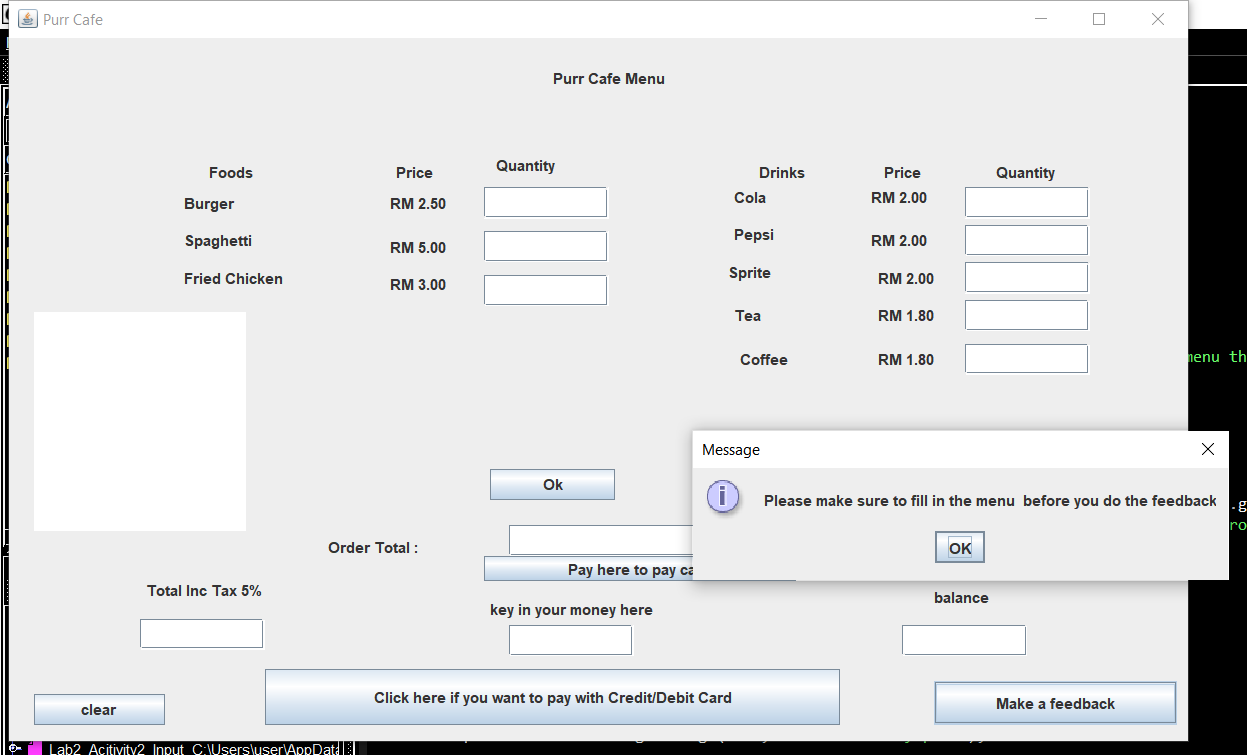


17. After you have done choosing your status whether your status is dine in, delivery or take away you will be taken to the next panel which MenuPanel.

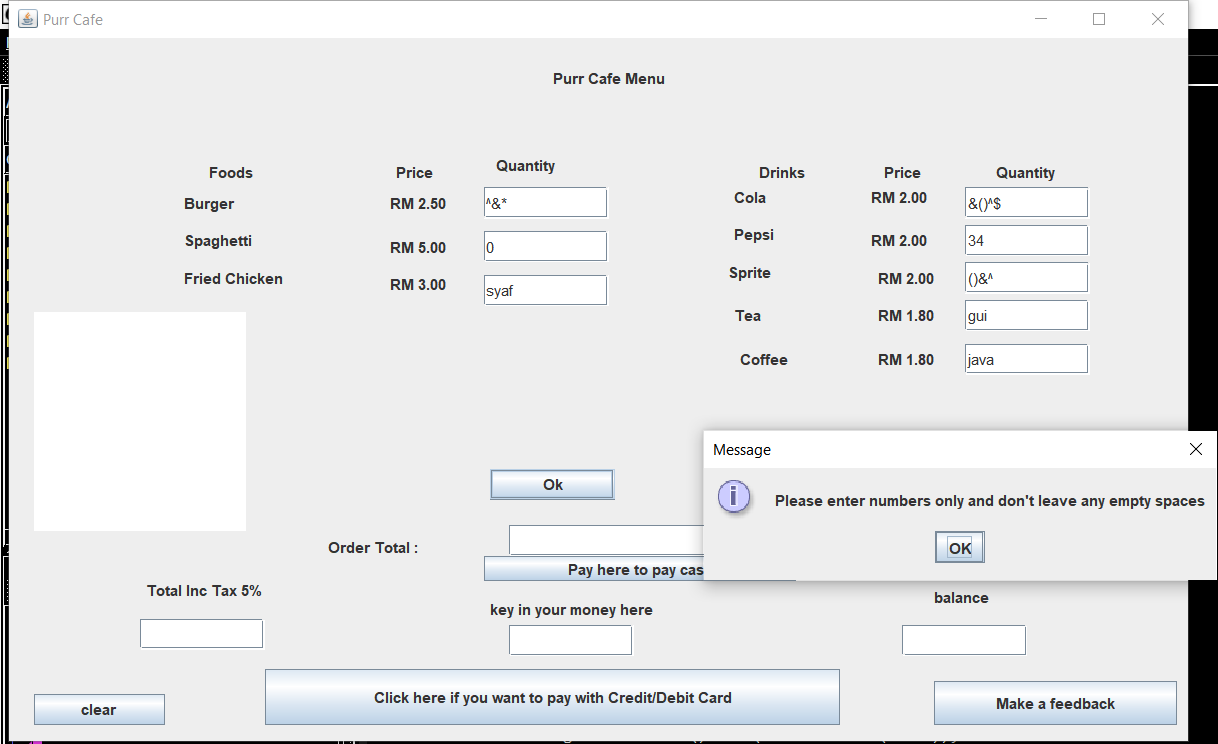
In this menu panel you have two options for payment, if you want to pay cash you can pay in the menu panel but if you chose to pay using card then you will be taken to the credit debit panel.



18. If you have not inputted any data after clicking the ok button there will be a message saying you must input all data before proceeding.

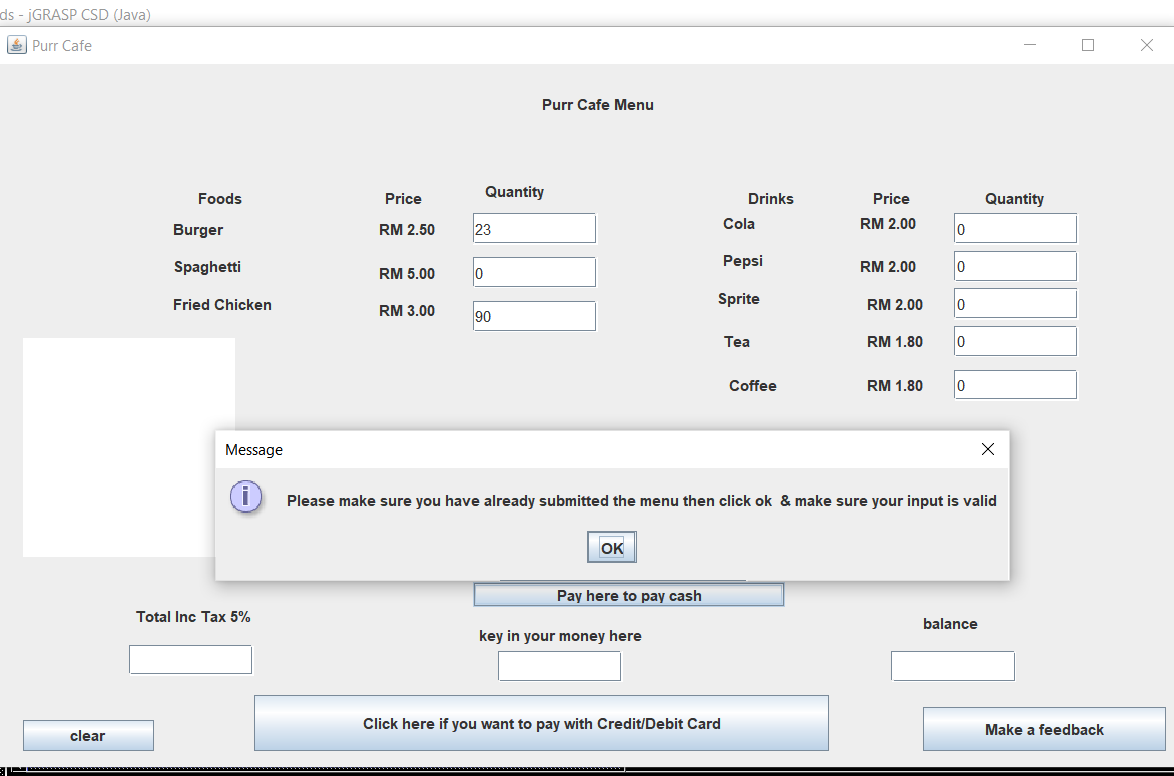


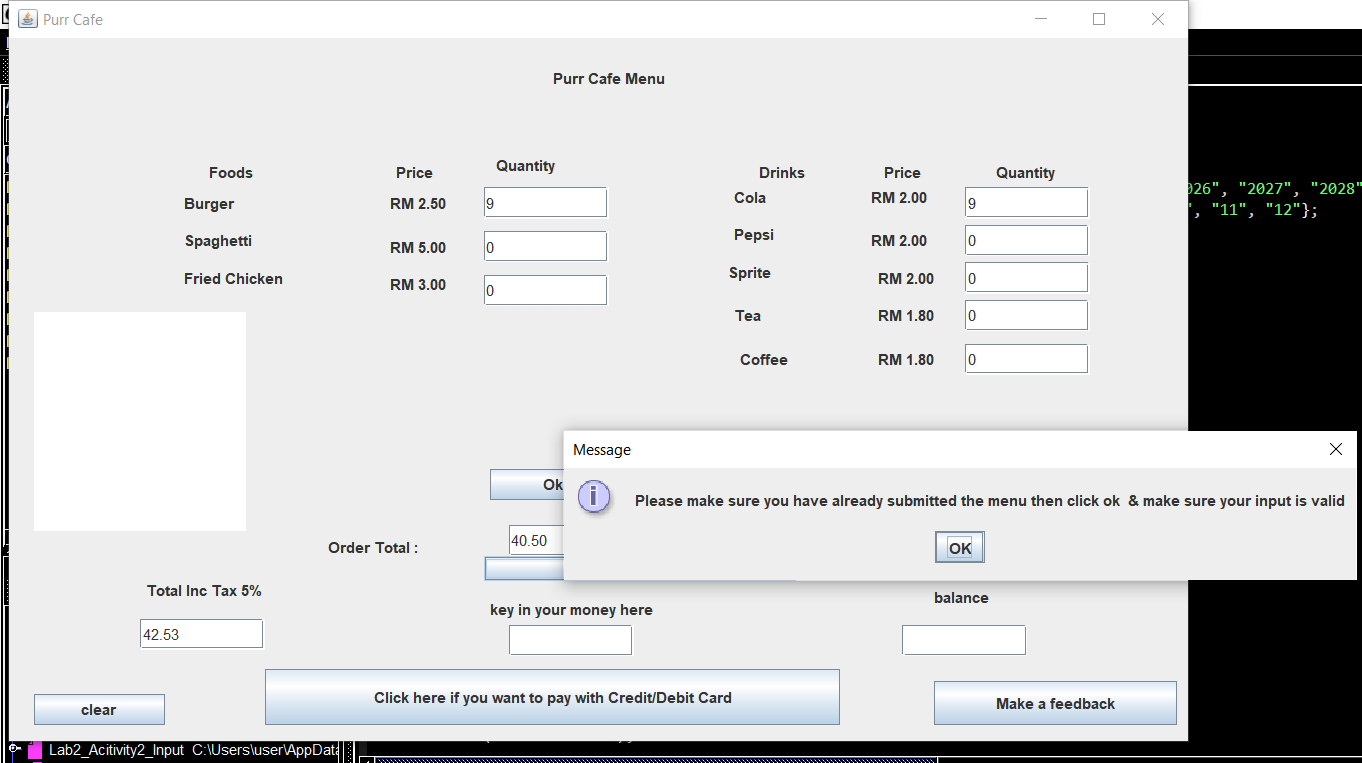
19. If you input an invalid input after you click the ok button you will be shown a message that says you have inputted an invalid input.



20.When you click the “Pay here to pay cash” button or “Click here if you want to pay with Credit/Debit Card” but haven’t key in the menu inputs or click the ok button you will be shown a message to make sure the menu is inputted , the input must be valid and you must click the ok button.

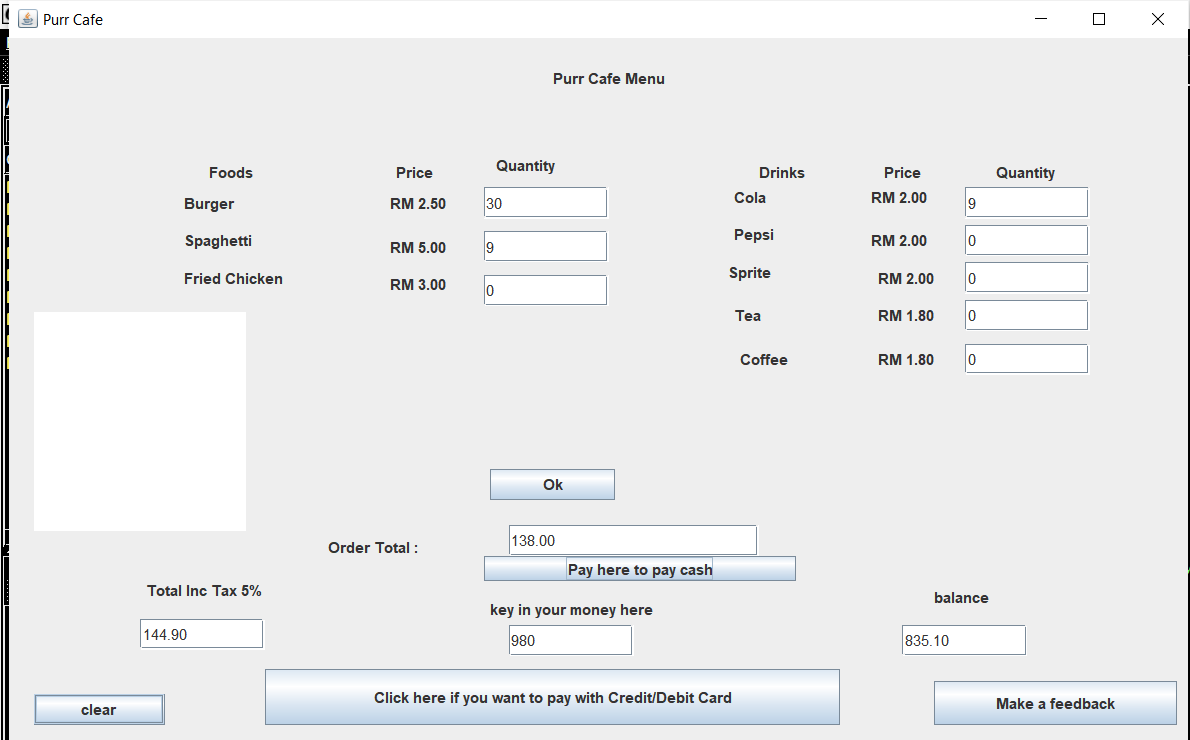
For “Pay here to pay cash” button if you have not input your money it will also show this message



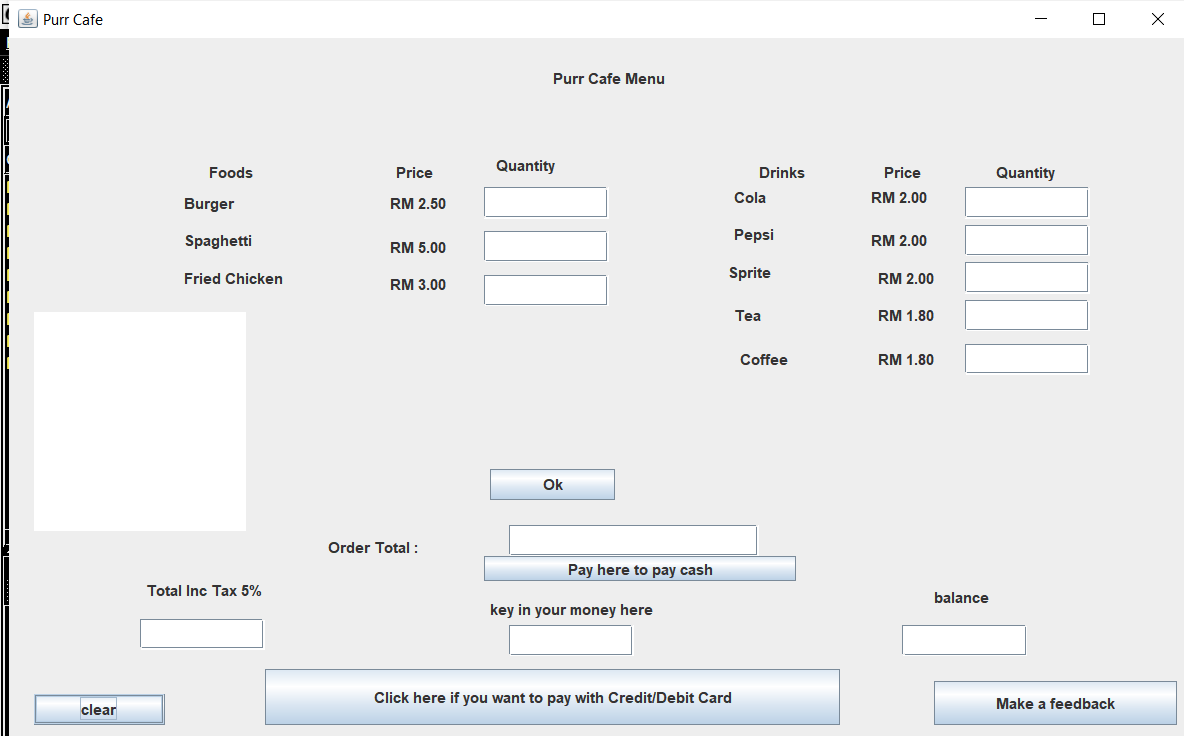


21. The clear button helps to clear all the inputs in the panel.

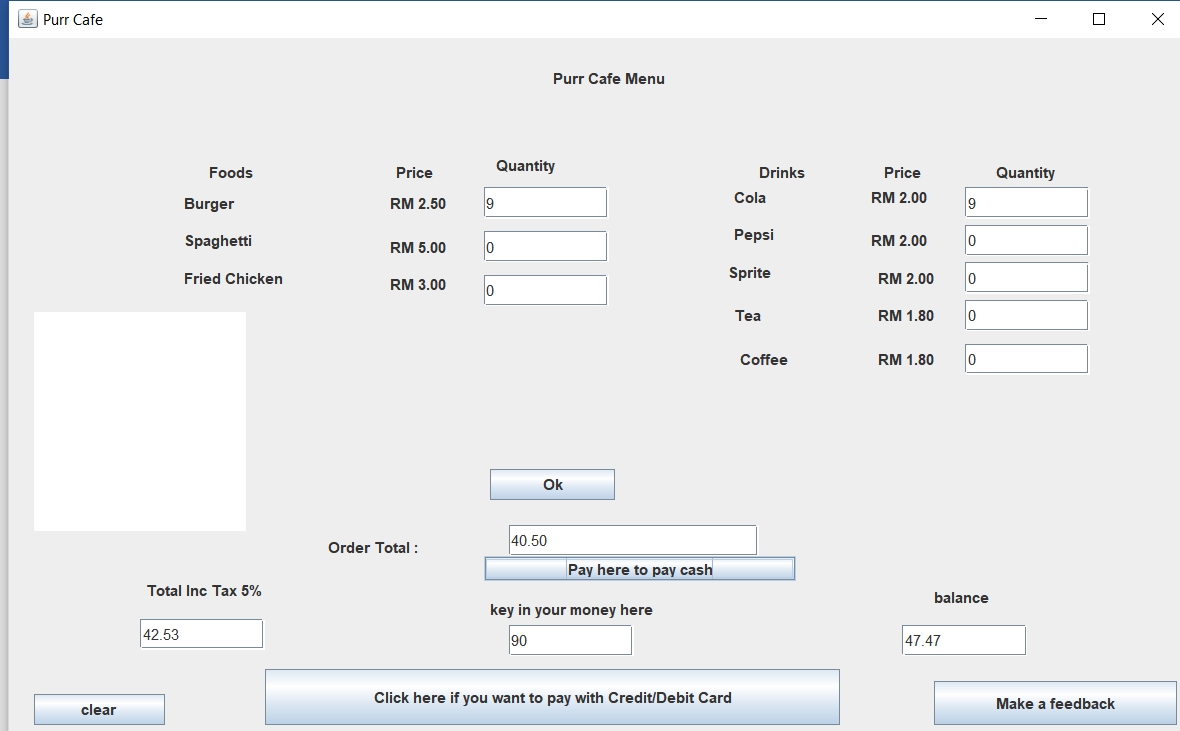
Before clicking clear button :



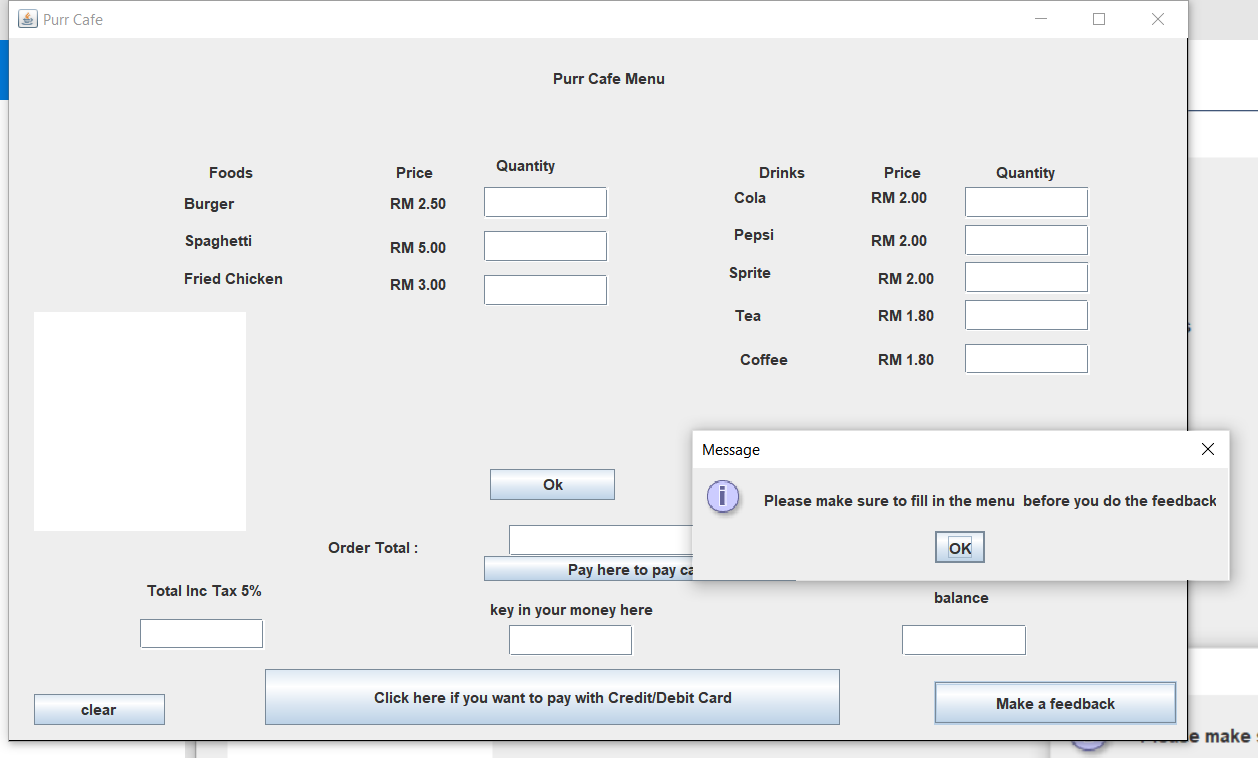
After clicking clear button :

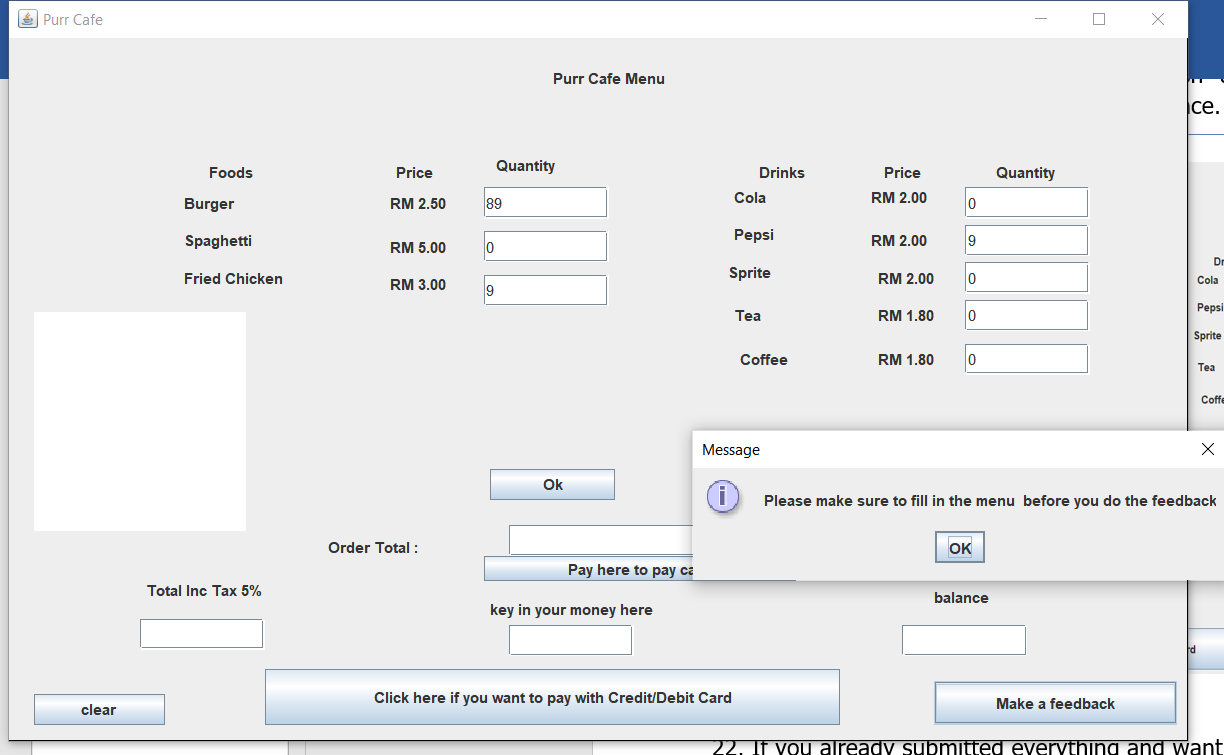


21.If you click the button “Pay here to pay cash” after inputing all of the data that is required you’ll be shown the balance.

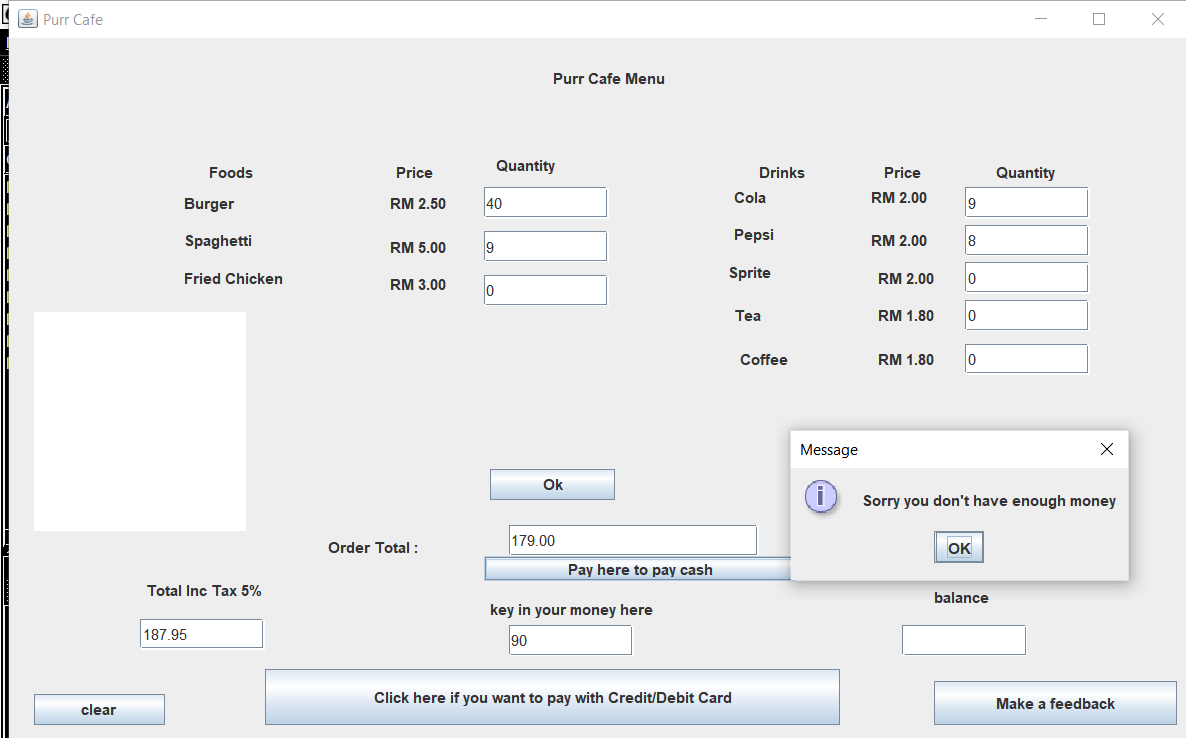


22.. If you click the “Make a feedback” button and you still haven’t finish inputing your data you’ll be shown a message saying you haven’t finish inputing your data



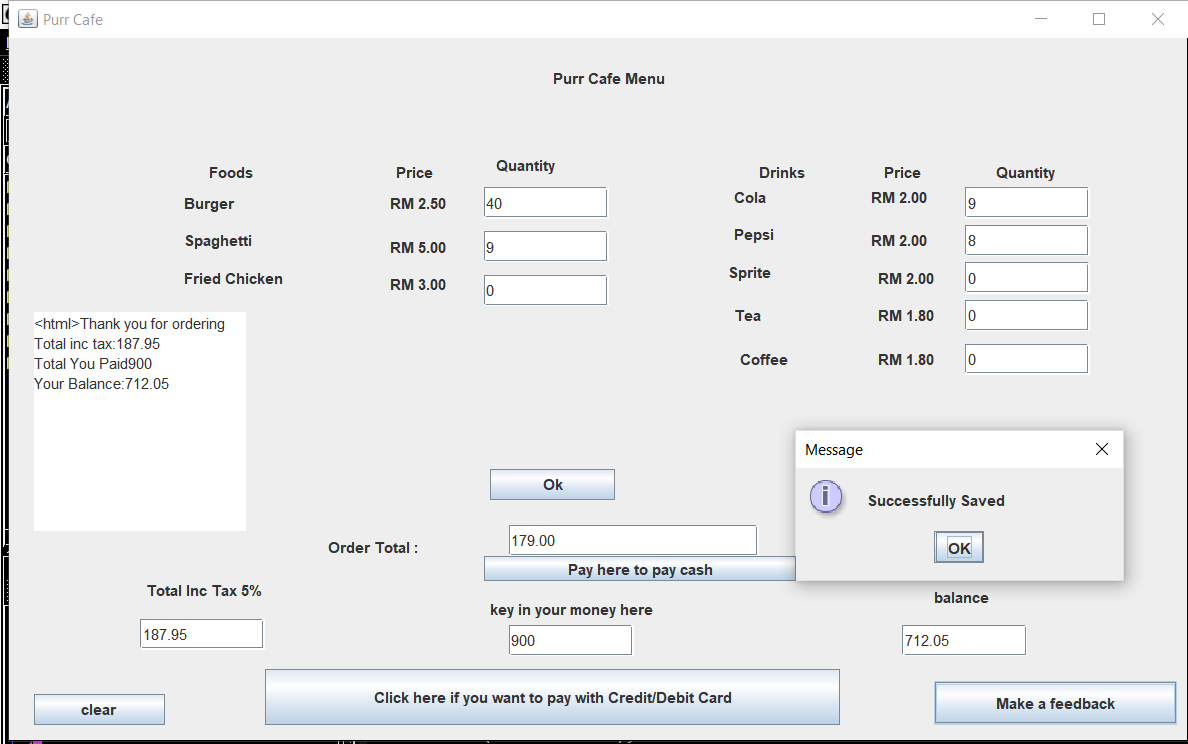


23. If you key in the money less than the total including tax there will message showing you don’t have enough money.

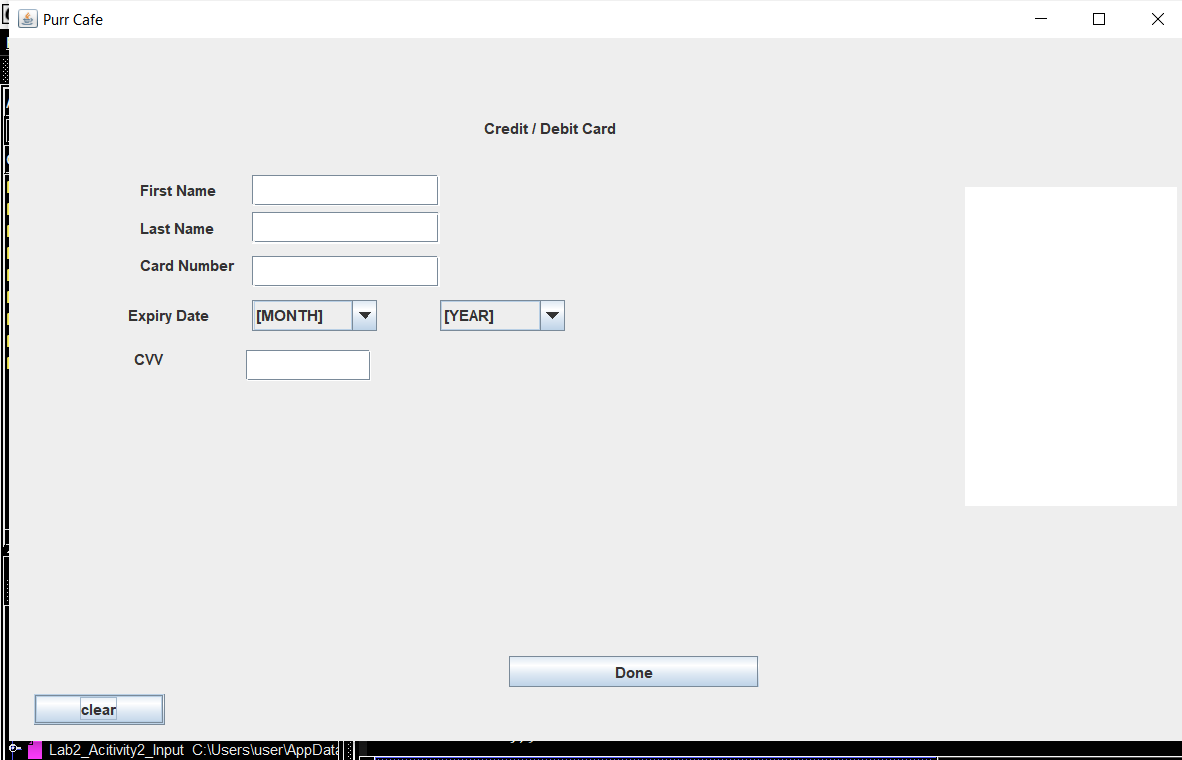


24. If you click the Make a feedback button and you’ve already submitted everything, the program will save the amount of money you paid on the purrcafe.txt file and will display it at the text area on left side of the frame.

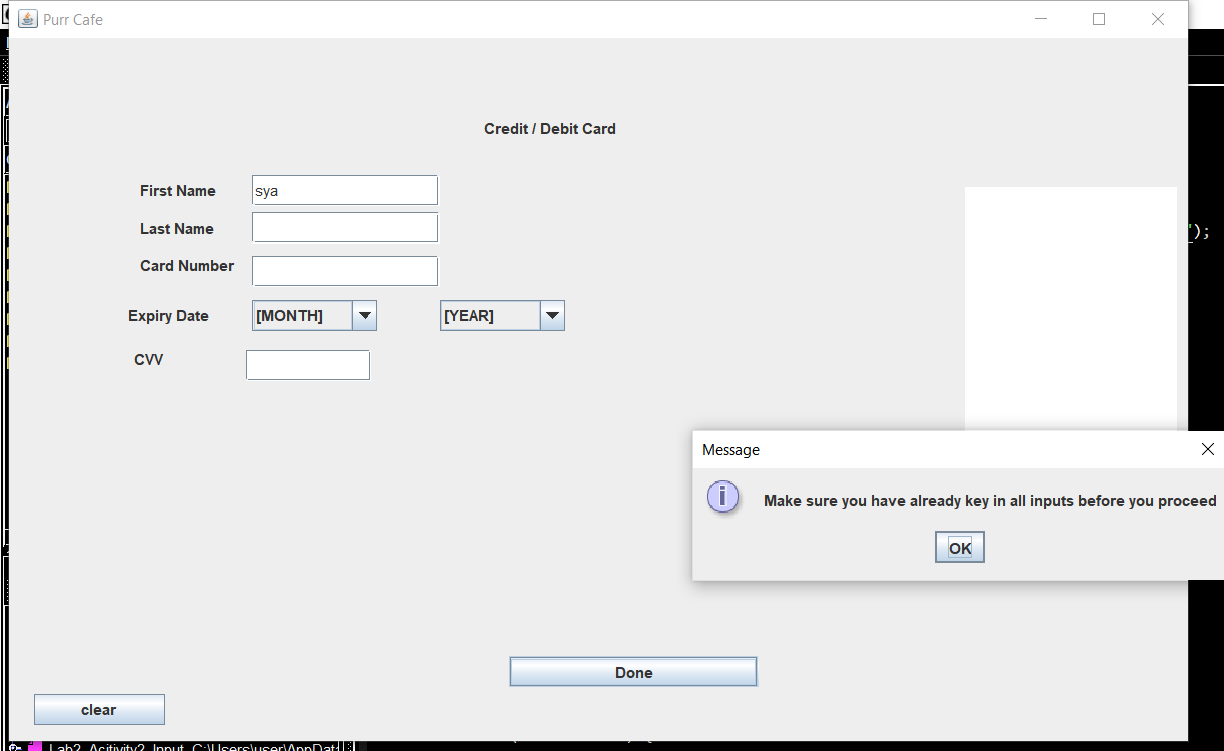
After you click ok at the message you will be taken to the feedback panel.



25. If you chose to pay with card, you’ll be taken to CreditDebitPanel

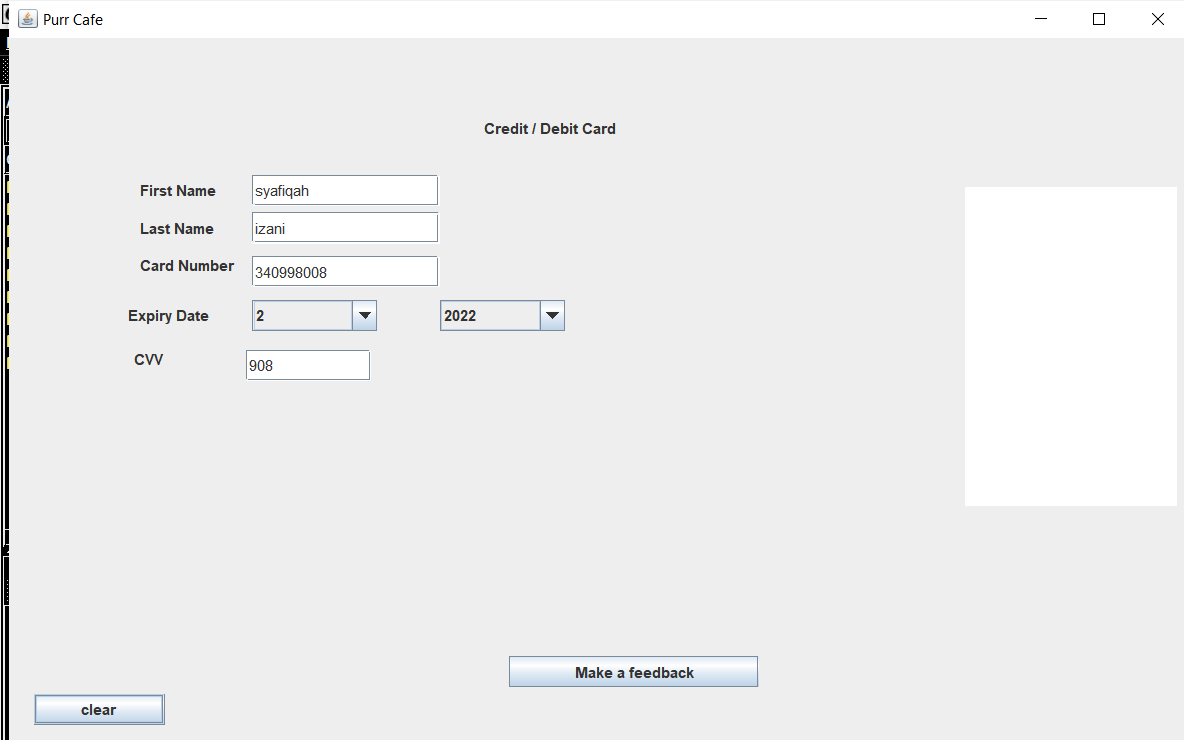


25. If you left an empty space there will be a message saying that you’re not done inputting.

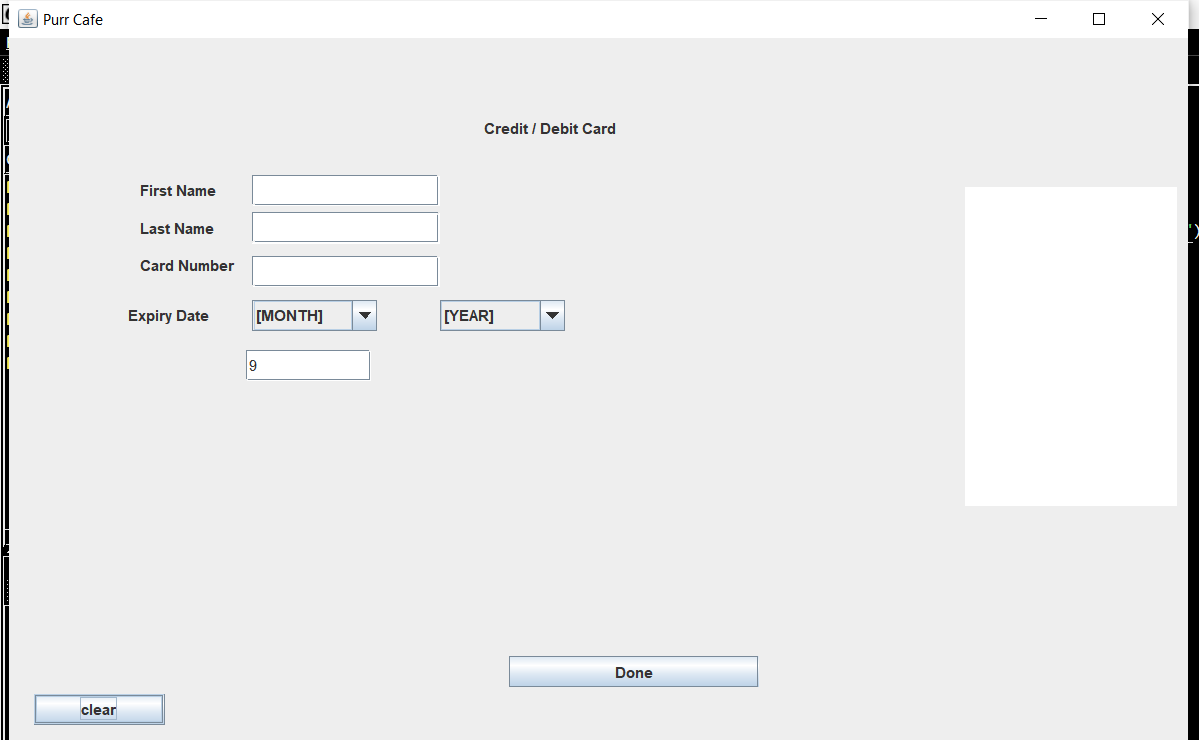


26. If you click the clear button, every input will be deleted.

Before selecting the clear button :

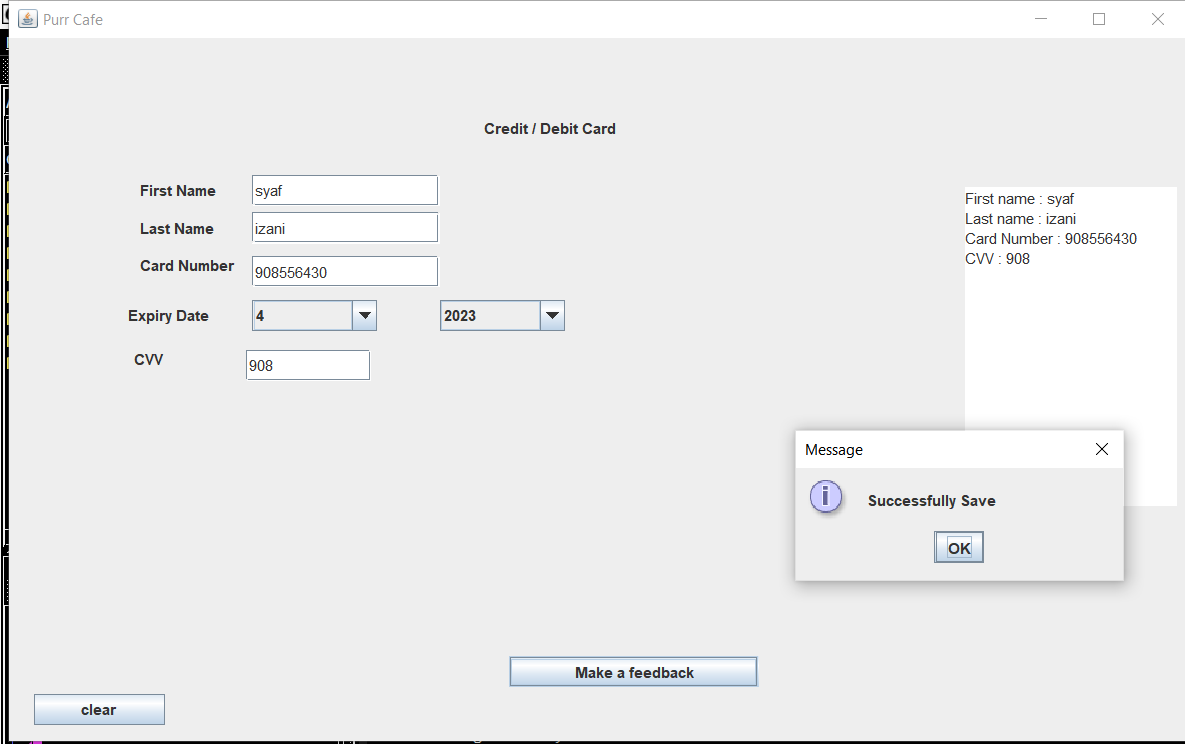


After selecting clear button :

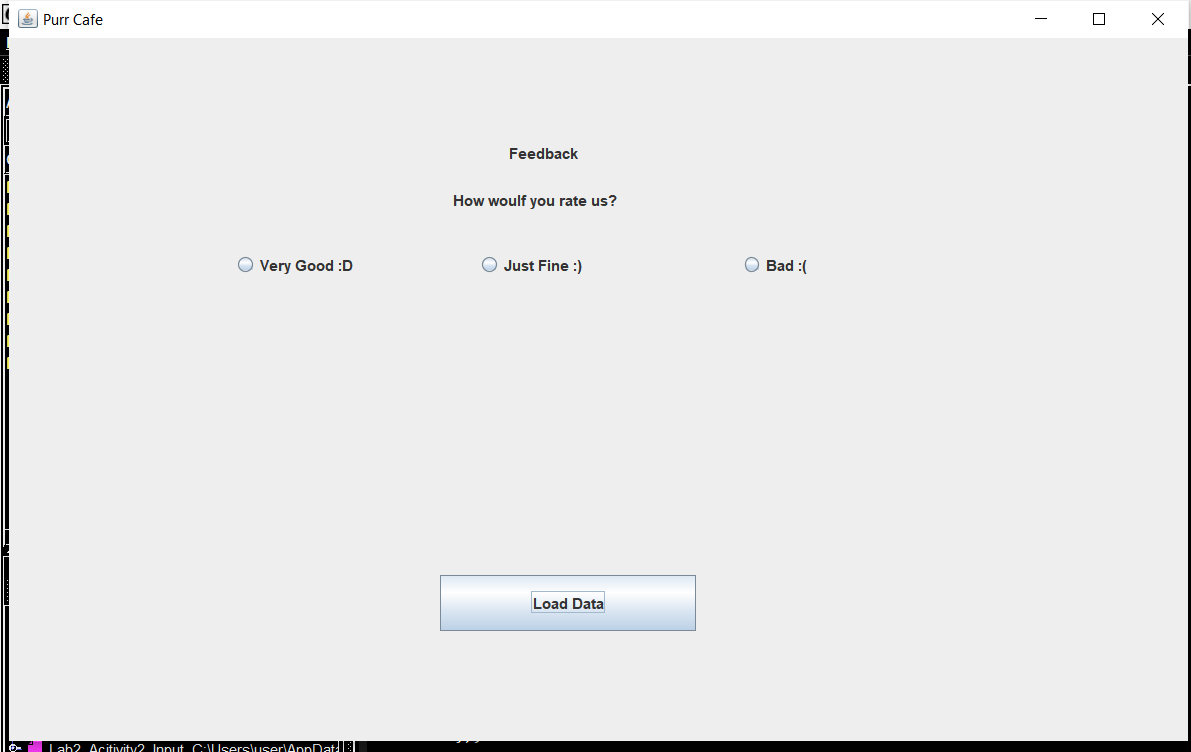


27. If you click Make a feedback and already submitted everything correctly, the program will save the data into the purrcafe.txt file and display it at the text area at right side of the frame.

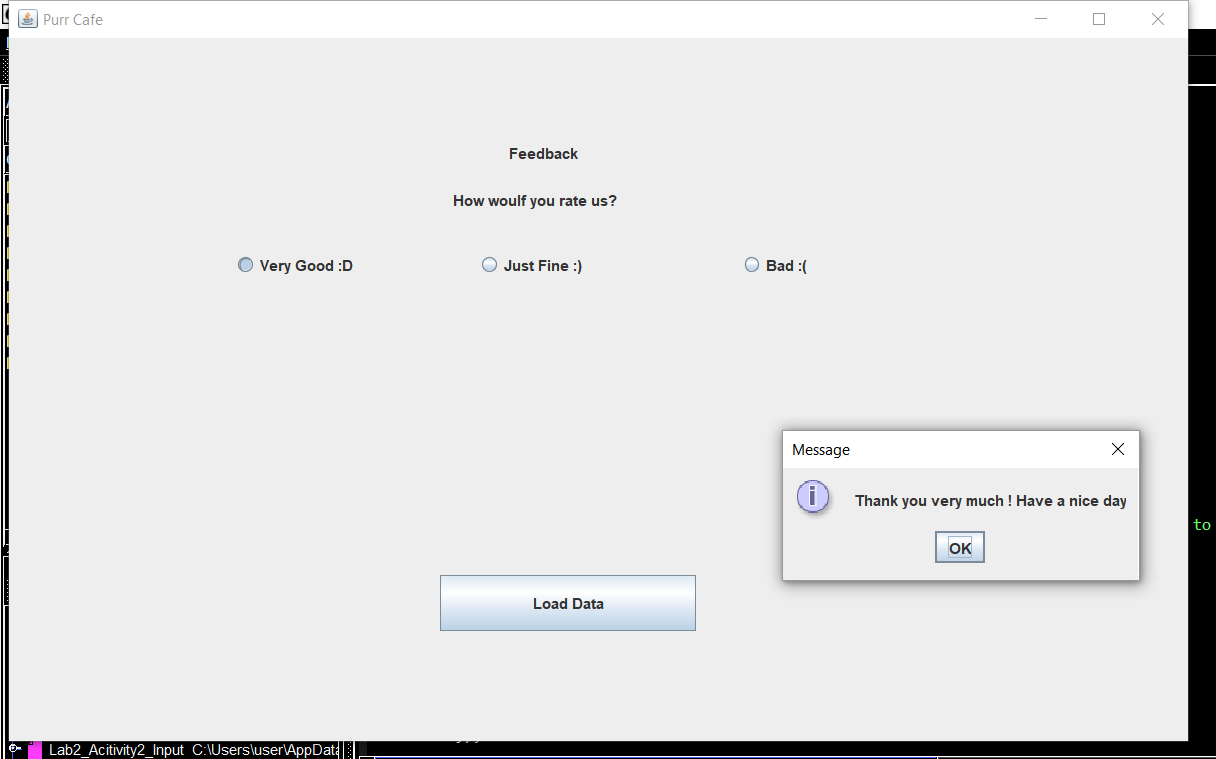
After that the program will take you to the FeedbackPanel

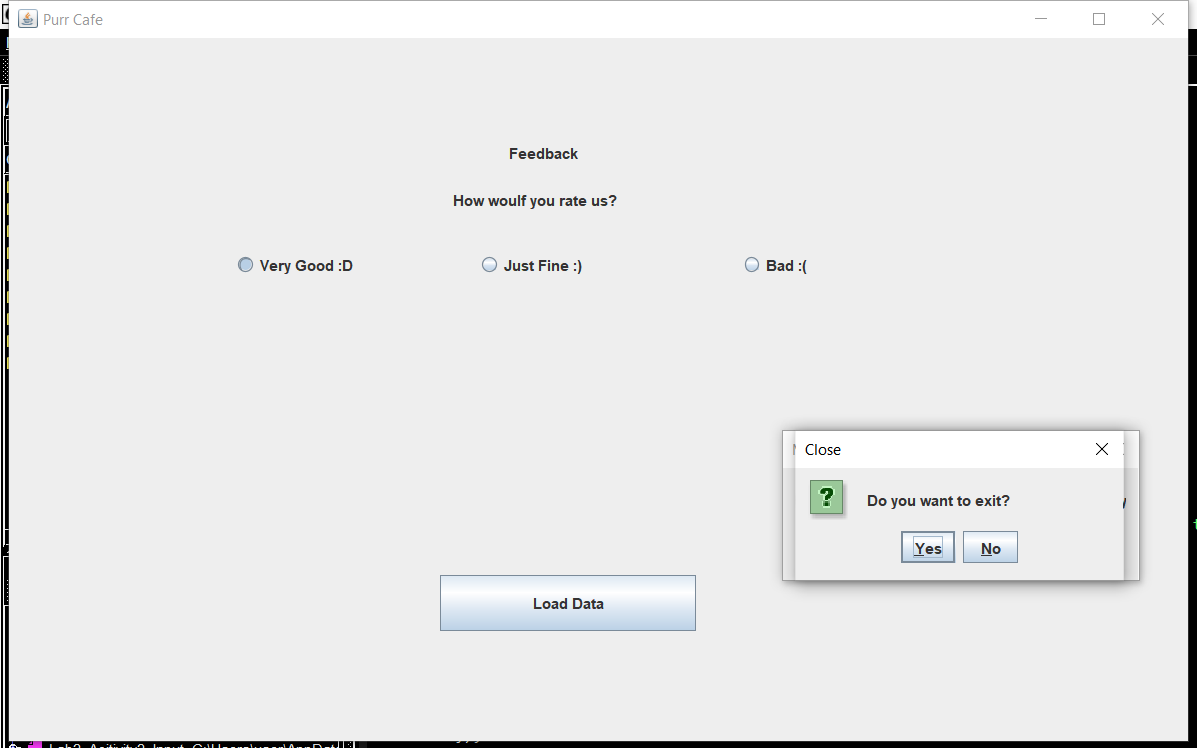


27. If you have already finish your payment, you’ll make a feedback

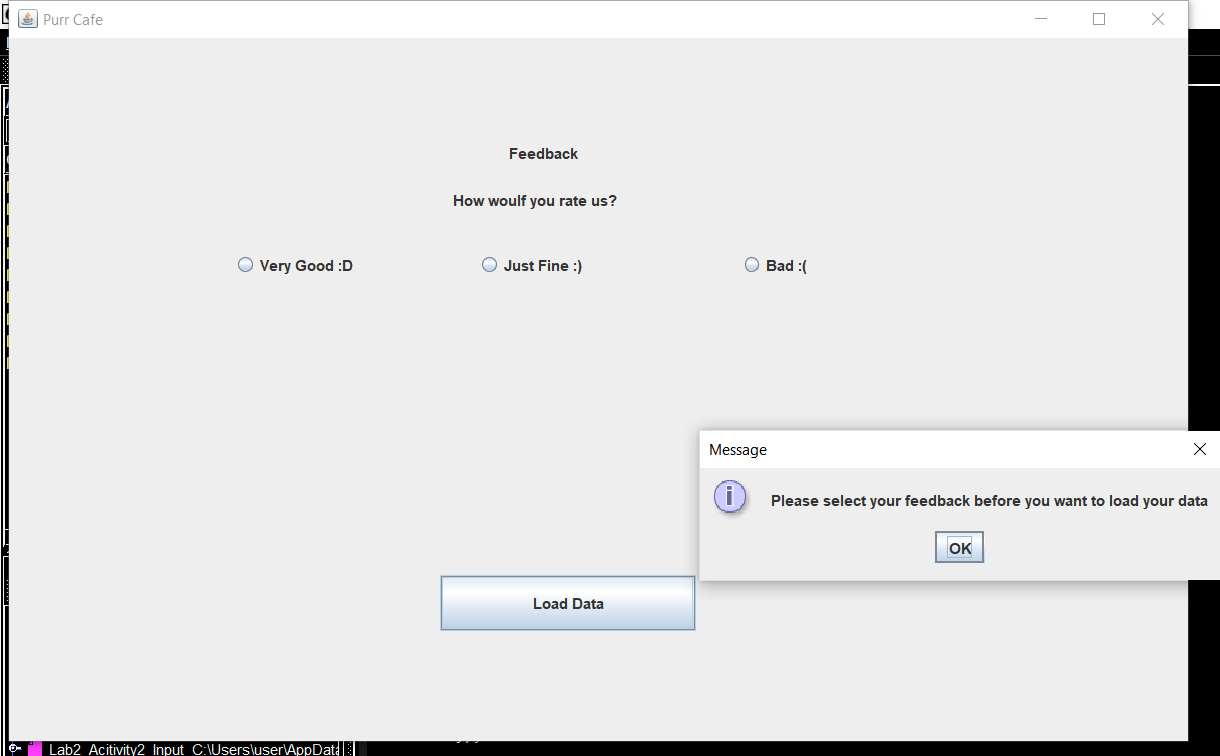


28. After you rate, you’ll be shown messages for your feedback and will be ask for your confirmation if you want to exit or not. If you chose, yes than you’ll be exited from the panel. If you say no, then you will stay at the FeedbackPanel





29. If you click the button Load Data before rating you’ll be shown a message saying that you must rate first before loading data



30. Finally, after you click the Load Data button after you have finished submitted everything, you’ll see the ordering record in Purr Café.

