“OBJECT ORIENTED PROGRAMMING”

PROJECT REPORT



Department Of SOFTWARE Engineering

**OOP (LAB)**

**RESTAURANT MANAGEMENT SYSTEM**

**Group Members**

|  |  |
| --- | --- |
| **MUHAMMAD USMAN KHAN** | **65171** |
| **MUHAMMAD HASSAM** | **65135** |
| **MUSTAFA HUSSAIN** | **65120** |
|  |  |

**BSE-2B**

**Spring 2020**

**Submitted to**

**ENGNR.SANIYA SHEIKH**

Table of Contents

1. Abstract 6

2. Introduction 7

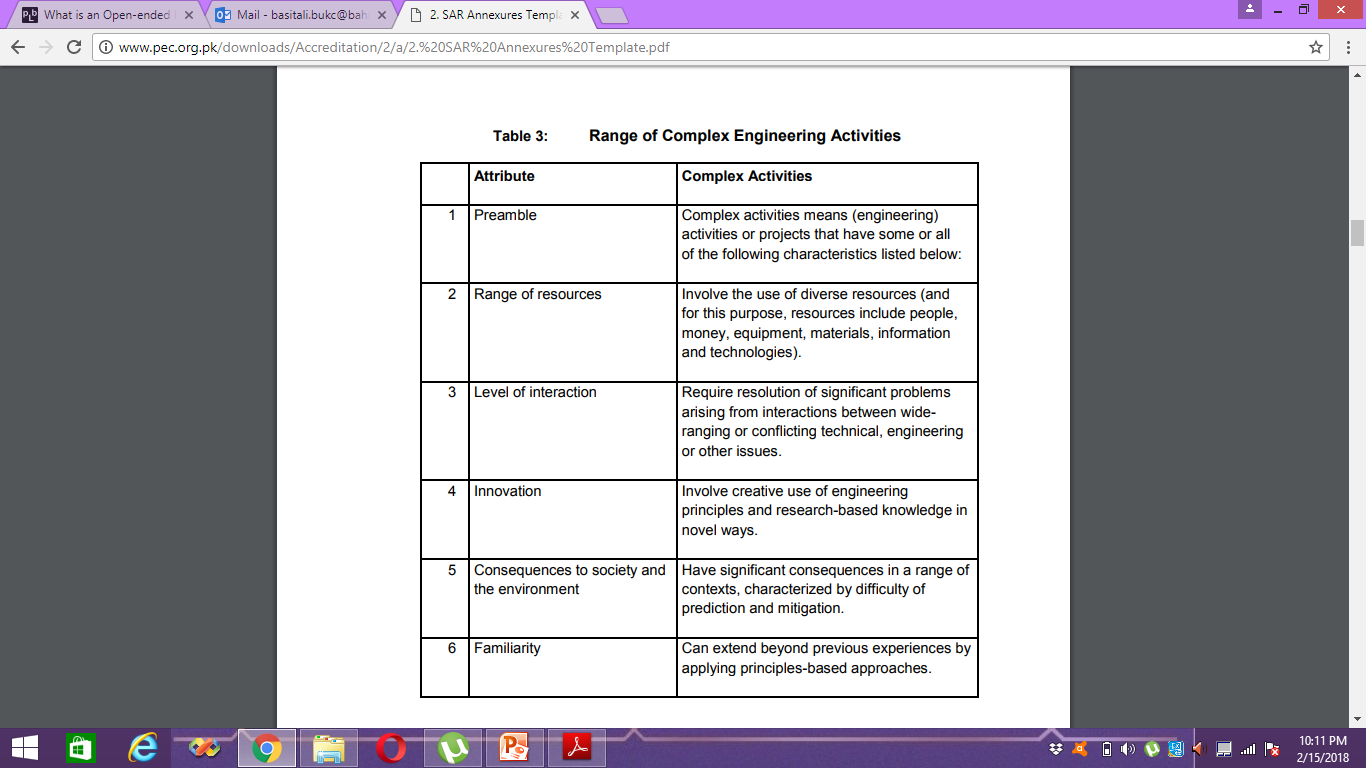
3. Backgroung of the project 9

4. Oop concept used in the project . 10

5. Description and code 11

6. Output and Interfaces 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name :OBJECT ORIENTED PROGRAMMING**  **Course code:** | | **Preamble** | **Range of Resources** | **Level of Interaction** | **Innovation** | **Consequences to society** | **Familiarity** |
| **Sr.**  1. | **Project description**  The project we design is restaurant management system ,and it consist of many  Boxes with different options we put and with 4 buttons ,each with a different option . This all is achieved by using JAVA SWING. |  |  |  |  |  |  |

****

**ABSTRACT (SUMMARY)**

In our project we have designed a restaurant management system.. In our project we have use core concepts of Object Oriented Programming. We are making our project using all latest OOP Concepts of Java and to use JAVA we have use **net beans software.** We have build our project with back-end as well as front-end (GUI), for back-end we are using **Java language** and for front-end (GUI) we are using **Java Swing**.

The project gives the multiple options to the customer and to the staff .There is a separate option for each item ,for example we have a specific box if meal and multiple food options are there to choose ,in a same way there is an option of drinks, then the edible we consume help the managing staff of restaurant to generate a bill along with all taxes visible on receipt to satisfy the customer that no extra money is charge to them .

**INTRODUCTION**

Swing also known as JAVA SWING is a part of Java Foundation classes (JFC), the other parts of Java Foundation classes are java2D and Abstract window toolkit (AWT). Abstract window toolkit, Swing & Java 2D are used for building graphical user interfaces (GUIs) in java. As we our giving our introduction ,and telling about the depth and details of our project we will mainly discuss about Swing API which is used for building GUIs on the top of AWT and are much more light-weight compared to AWT .Our project is entirely based on the JAVA SWING to achieve goal .It is a desktop application that we have created .

In our project we have used several components. Let’s discuss a bit about them first so it will be easy for the reader that why we use them in the project and they will be aware of there use then we will move further ahead discussing other details .

**JFrame** – A frame is an instance of JFrame. Frame is a window that can have title, border, menu, buttons, text fields and several other components. A Swing application must have a frame to have the components added to it.  
 **JPanel** – A panel is an instance of JPanel. A frame can have more than one panels and each panel can have several components. You can also call them parts of Frame. Panels are useful for grouping components and placing them to appropriate locations in a frame.

**JLabel** – A label is an instance of JLabel class. A label is unselectable text and images. If you want to display a string or an image on a frame, you can do so by using labels. In the above example we wanted to display texts “User” & “Password” just before the text fields , we did this by creating and adding labels to the appropriate positions.

**JTextField** – Used for capturing user inputs, these are the text boxes where user enters the data…

**JButton** – A button is an instance of JButton class. In the above example we have a button “Login”.

***We have not use two other features but they are helpful, if we kept a security lock in our project then we must use***

**JPasswordField** – Similar to text fields but the entered data gets hidden and displayed as dots on GUI.

***“ BACKGRROUND OF THE PROJECT ”***

Formally in our country in some local restaurant we don’t have a proper billing system means the management of the restaurants are still working on the traditional old system ,where the waiters take orders of the customers and the food the consume the waiter just simply add its cost and the manager make a final bill by using calculator manually .There is a major drawback in this system firstly many times the staff overbill the customers either by will or by intentions to earn extra money and secondly there is no proper tax system which put burden on the government as there are so many restaurant and out of many only few pay the bill while rest simply don’t pay it and as this is a commercial business there is a need to regulate them . They must pay the tax and the customer should also pay the tax on the food .

Purpose of this application is vast means it is beneficent for the customer ,staff of the restaurant ,government and finally for the common citizen.

1) Customer is charged for what they eat means they are not overbilled and they can cross check it against there receipt because everything is now mentioned in the receipt.

2) Staff is not blamed for anything and they can do there jobs freely .

3) It create legal and the basic source of income for the government so they can earn extra money which is simply the basic right of the government . Due to this our government tax collection increase.

4) Common citizen just got benefit from this because food chain is the largest business in our country and only few pays the tax ,but now as they will pay tax government will more spend on the development .

***“OOP CONCEPTS USED IN THE PROJECT”***

It is hard to say that any concept of ***OOP*** is left or not but we try our best to cover all the things we fell necessary to cover in this , and all those concepts which are useful to give our project a good look .

First of all the basic thing we do in our , or we can say that we used the basic concept of OOP that is we first make ***CLASSES*** then we create ***STATES*** and ***METHODS***  in the class means we have done ***ENCAPSULATION*** as inside the class there are both the thing methods and states declared .We also created ***CONSTRUCTOR*** to initialize the objects.

Secondly we create ***OBJECTS*** to call methods and states and to set variables of the states if they were set public previously , know moving ahead our group also used the ***ACCESS MODIFIERS*** (public private ,protected ,static). We all performed ***COMPOSITION*** .

Moreover we also use the concept of **INHERITENCE** that is we create **PARENT CLASS , SUPERCLASS** and inherit **a CHILD CLASS , DERIVED CLASS** from it . For inheritance we used the key word ***extends***

So that all the features of parent class are now inherited into child class and we don’t have to write each and every method of it again and again and this help in the ***REUSABILITY*** of the code .

Method of ***GETTER*** and ***SETTER*** were also used when in J FRAME we set price of the edible items .

Moving to the J FRAME we do multiple things first of all we create a panel and through and then choose text field to put the heading of restaurant management system. Then we do multiple things like we create BUTTONS with options of ***RESET , TOTAL , EXIT*** .

We import libraries to make the buttons functionals effective so the task we want to perform with the help of our button we can do it simply and we also do multiple task to do things which make our project effective.

Lastly , the thing which is of importance is that we import ***JTATTOO***.

JTattoo consists of several different Look and Feels for Swing applications. All of them enables developers to improve their application with an excellent user interface. So JTattoo opens desktop applications the door to end users who are unfortunate with the Look and Feels shipped with the standard JDK.

***“CODES DESCRIPTION”***

**LOGIN :**

package Login;

import javax.swing.JOptionPane;

import java.awt.Toolkit;

import java.awt.\*;

import java.awt.event.WindowEvent;

import Login.RestaurantSystem;

import javax.swing.ImageIcon;

public class Login extends javax.swing.JFrame {

public Login() {

initComponents();

setLocationRelativeTo(null);

ImageIcon myimage = new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("lock1.png")));

Image img1 = myimage.getImage();

Image img2 = img1.getScaledInstance(Lable.getWidth(),Lable.getHeight(), Image.SCALE\_SMOOTH);

ImageIcon i = new ImageIcon(img2);

Lable.setIcon(i);

}

@SuppressWarnings("unchecked")

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

// TODO add your handling code here:

System.exit(0);

}

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

User.setText(null);

Password.setText(null);

}

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

String password = Password.getText();

String user = User.getText();

if (user.contains("Tesla") && password.contains("12345")){

User.setText(null);

Password.setText(null);

systemExit();

RestaurantSystem Info = new RestaurantSystem("HUM Restaurant System");

Info.setVisible(true);

}

else{

JOptionPane.showMessageDialog(null,"Invalid Login Details","Login Error",JOptionPane.ERROR\_MESSAGE);

User.setText(null);

Password.setText(null);

}

}

public static void main(String args[]) {

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

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Exit;

private javax.swing.JLabel Lable;

private javax.swing.JButton Login;

private javax.swing.JPasswordField Password;

private javax.swing.JButton Reset;

private javax.swing.JTextField User;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

// End of variables declaration

private void systemExit(){

WindowEvent winCloseing = new WindowEvent(this,WindowEvent.WINDOW\_CLOSING);

}

}

**RestaurentSystem:**

package Login;

import java.util.\*;

import java.util.Calendar;

import java.text.SimpleDateFormat;

import java.awt.\*;

import java.awt.TextField;

import javax.swing.\*;

public class RestaurantSystem extends javax.swing.JFrame {

/\*\*

\* Creates new form RestaurantSystem

\*/

public RestaurantSystem(String title) {

super( title );

initComponents();

setLocationRelativeTo(null);

ImageIcon myimage = new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("r.png")));

Image img1 = myimage.getImage();

Image img2 = img1.getScaledInstance(lable1.getWidth(),lable1.getHeight(), Image.SCALE\_SMOOTH);

ImageIcon i = new ImageIcon(img2);

lable1.setIcon(i);

ImageIcon myimage1 = new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("r.png")));

Image img12 = myimage1.getImage();

Image img21 = img12.getScaledInstance(lable3.getWidth(),lable3.getHeight(), Image.SCALE\_SMOOTH);

ImageIcon i1 = new ImageIcon(img21);

lable3.setIcon(i1);

}

/\*\*

\* 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")

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

// TODO add your handling code here:

}

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

SubClass ItemCost = new SubClass();

double iTax, iSubTotal, iTotal;

ItemCost.ChickenBurger = ItemCost.pChickenBurger \* Double.parseDouble(jtxtChickenBurger.getText());

ItemCost.BeefRoll = ItemCost.pBeefRoll \* Double.parseDouble(jtxtBeefRoll.getText());

ItemCost.Pizza = ItemCost.pPizza \* Double.parseDouble(jtxtPizza.getText());

ItemCost.ChickenTikka = ItemCost.pChickenTikka \* Double.parseDouble(jtxtChickenTikka.getText());

ItemCost.ChickenKarahi = ItemCost.pChickenKarahi \* Double.parseDouble(jtxtChickenKarahi.getText());

ItemCost.ColdDrink = ItemCost.pColdDrink \* Double.parseDouble(jtxtColdDrink.getText());

ItemCost.MilkShake = ItemCost.pMilkShake \* Double.parseDouble(jtxtMilkShake.getText());

ItemCost.Coffee = ItemCost.pCoffee \* Double.parseDouble(jtxtCoffee.getText());

ItemCost.Tea = ItemCost.pTea \* Double.parseDouble(jtxtTea.getText());

ItemCost.Juice = ItemCost.pJuice \* Double.parseDouble(jtxtJuice.getText());

iSubTotal = ItemCost.GetAmount();

iTax =ItemCost.cFindTax(iSubTotal);

iTotal = iSubTotal + iTax;

String SubTotal =String.format("Rs%.2f", iSubTotal);

jtxtSubTotal.setText(SubTotal);

String Tax =String.format("Rs%.2f", iTax);

jtxtTax.setText(Tax);

String Total =String.format("Rs%.2f", iTotal);

jtxttotal.setText(Total);

String Meal =String.format("Rs%.2f", ItemCost.Meals);

jtxtMealCost.setText(Meal);

String Drink =String.format("Rs%.2f", ItemCost.Drinks);

jtxtDrinkCost.setText(Drink);

String TotalCost =String.format("Rs%.2f", ItemCost.Total);

jtxtTotalCost.setText(TotalCost);

}

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

JTextField cleartext = null;

for (Component c:jP1.getComponents()){

if(c.getClass().toString().contains("javax.swing.JTextField")){

cleartext = (JTextField)c;

cleartext.setText("0");

}

}

for (Component c:jP2.getComponents()){

if(c.getClass().toString().contains("javax.swing.JTextField")){

cleartext = (JTextField)c;

cleartext.setText("0");

}

}

for (Component c:jP4.getComponents()){

if(c.getClass().toString().contains("javax.swing.JTextField")){

cleartext = (JTextField)c;

cleartext.setText("");

}

}

for (Component c:jP3.getComponents()){

if(c.getClass().toString().contains("javax.swing.JTextField")){

cleartext = (JTextField)c;

cleartext.setText("");

}

}

jTxtRecipte.setText(null);

}

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

int refs= 1325+(int)(Math.random()\*4238);

Calendar timer=Calendar.getInstance();

timer.getTime();

SimpleDateFormat tTime=new SimpleDateFormat("HH:mm:ss");

tTime.format(timer.getTime());

SimpleDateFormat Tdate=new SimpleDateFormat("dd-MM-YYYY");

Tdate.format(timer.getTime());

jTxtRecipte.append("\tRestaurant Management System\n" +

"Reference:\t\t\t" + refs +

"\n=======================================================\n\n" +

"Date:" +Tdate.format(timer.getTime()) +

"\nTime:" +tTime.format(timer.getTime()) +

"\n======================================================\n" +

"Meals:\t\t" + jtxtMealCost.getText() + "\n\n" +

"Drinks:\t\t" + jtxtDrinkCost.getText() + "\n\n" +

"Total Cost :\t\t" + jtxtTotalCost.getText() + "\n\n" +

"========================================================\n\n" +

"Tax:\t\t" + jtxtTax.getText() + "\n\n" +

"SubTotal:\t\t" + jtxtSubTotal.getText() + "\n\n" +

"Total :\t\t" + jtxttotal.getText() + "\n\n" +

"\n=======================================================\n" +

"---------------------------------------Thank You-------------------------------------------\n" +

"=======================================================\n");

}

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

SubClass iExit = new SubClass();

iExit.iExistSystem();

}

/\*\*

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new RestaurantSystem("HUM Restaurant System").setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Exit;

private javax.swing.JLabel Subtotal1;

private javax.swing.JLabel Subtotal3;

private javax.swing.JLabel Subtotal4;

private javax.swing.JLabel Subtotal5;

private javax.swing.JLabel Subtotal7;

private javax.swing.JLabel Tax1;

private javax.swing.JLabel Tax3;

private javax.swing.JLabel Tax4;

private javax.swing.JLabel Tax5;

private javax.swing.JLabel Tax7;

private javax.swing.JButton Total;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jP1;

private javax.swing.JPanel jP2;

private javax.swing.JPanel jP3;

private javax.swing.JPanel jP4;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel6;

private javax.swing.JPanel jPanel7;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTxtRecipte;

private javax.swing.JTextField jtxtBeefRoll;

private javax.swing.JTextField jtxtChickenBurger;

private javax.swing.JTextField jtxtChickenKarahi;

private javax.swing.JTextField jtxtChickenTikka;

private javax.swing.JTextField jtxtCoffee;

private javax.swing.JTextField jtxtColdDrink;

private javax.swing.JTextField jtxtDrinkCost;

private javax.swing.JTextField jtxtJuice;

private javax.swing.JTextField jtxtMealCost;

private javax.swing.JTextField jtxtMilkShake;

private javax.swing.JTextField jtxtPizza;

private javax.swing.JButton jtxtRecipte;

private javax.swing.JButton jtxtReset;

private javax.swing.JTextField jtxtSubTotal;

private javax.swing.JTextField jtxtTax;

private javax.swing.JTextField jtxtTea;

private javax.swing.JTextField jtxtTotalCost;

private javax.swing.JTextField jtxttotal;

private javax.swing.JLabel lable1;

private javax.swing.JLabel lable2;

private javax.swing.JLabel lable3;

private javax.swing.JLabel s;

private javax.swing.JLabel t;

private javax.swing.JLabel tt;

// End of variables declaration

}

**SuperClass:**

package Login;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

public class SuperClass {

public double ChickenBurger;

public double BeefRoll;

public double Pizza;

public double ChickenTikka;

public double ChickenKarahi;

public double ColdDrink;

public double MilkShake;

public double Coffee;

public double Tea;

public double Juice;

public double Meals;

public double Drinks;

public double Total;

public double AllTotal;

public double GetAmount(){

Meals = ChickenBurger + BeefRoll + Pizza + ChickenTikka + ChickenKarahi;

Drinks = ColdDrink + MilkShake + Coffee + Tea + Juice;

Total = Meals + Drinks;

AllTotal = Total;

return AllTotal;

}

private JFrame frame;

public void iExistSystem(){

frame = new JFrame("Exist");

if(JOptionPane.showConfirmDialog(frame, "Confirm if you want to exist","Restaurant Managment System",

JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_NO\_OPTION){

System.exit(0);

}

}

//============================Price==============================

public double pChickenBurger = 180;

public double pBeefRoll = 120;

public double pPizza = 800;

public double pChickenTikka = 160;

public double pChickenKarahi = 950;

public double pColdDrink = 50;

public double pMilkShake = 120;

public double pCoffee = 150;

public double pTea = 60;

public double pJuice = 80;

//================================================================

public double mcTax = 0.13;

public double cFindTax(double cAmount){

double FindTax = (cAmount \* mcTax);

return FindTax;

}

}

**SubClass:**

package Login;

public class SubClass extends SuperClass {

}

**“*Output and Interfaces*”**

