Mini Project

Roll No. : N002	Name: Anmaya Agarwal
Program: MBA Tech (Computer)	Division: B
Semester: 3	Batch : B1

Program 1: Mini Project for developing a GUI to order and customize a pizza

Source Code:

Class pizzaShop –

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;


public class pizzaShop
{
   JFrame screen;
   public double sum,diff,product,division;


   pizzaShop (JFrame screen)
   {
      this.screen = screen;
      screen.setVisible(true);


      JLabel heading = new JLabel("Pizza Shop");
      heading.setFont(new Font("Verdana", Font.BOLD, 18));
```

```
heading.setForeground(Color.RED);
heading.setBounds(570,0,150,30);
screen.add(heading);

JLabel l1 = new JLabel("Name = ");
l1.setBounds(70,75,80,20);
l1.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l1);
JTextField t1 = new JTextField("");
t1.setBounds(160,70,200,30);
t1.setFont(new Font("Verdana", Font.PLAIN, 14));
screen.add(t1);

JLabel l2 = new JLabel("Phone Number = ");
l2.setBounds(500,75,150,20);
l2.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l2);
JTextField t2 = new JTextField(10);
t2.setBounds(660,70,200,30);
t2.setFont(new Font("Verdana", Font.PLAIN, 14));
screen.add(t2);

JLabel l3 = new JLabel("Pizza Type = ");
l3.setBounds(70,140,150,20);
l3.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l3);
```

```
JRadioButton b1 = new JRadioButton("Margherita(100)");
JRadioButton b2 = new JRadioButton("Chicken Tikka(305)");
JRadioButton b3 = new JRadioButton("Farmhouse(215)");
b1.setFont(new Font("Verdana", Font.BOLD, 14));
b2.setFont(new Font("Verdana", Font.BOLD, 14));
b3.setFont(new Font("Verdana", Font.BOLD, 14));
b1.setBackground(Color.CYAN);
b2.setBackground(Color.CYAN);
b3.setBackground(Color.CYAN);
b1.setBounds(230,140,200,20);
b2.setBounds(440,140,200,20);
b3.setBounds(650,140,200,20);
b1.setSelected(true);
screen.add(b1);
screen.add(b2);
screen.add(b3);
ButtonGroup bg1 = new ButtonGroup();
bg1.add(b1);
bg1.add(b2);
bg1.add(b3);

JLabel l4 = new JLabel("Crust = ");
l4.setBounds(70,205,150,20);
l4.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l4);
JRadioButton b4 = new JRadioButton("Hand Tossed");
```

```
JRadioButton b5 = new JRadioButton("Pan Pizza(40)");
JRadioButton b6 = new JRadioButton("Cheese Burst(100)");
b4.setFont(new Font("Verdana", Font.BOLD, 14));
b5.setFont(new Font("Verdana", Font.BOLD, 14));
b6.setFont(new Font("Verdana", Font.BOLD, 14));
b4.setBackground(Color.CYAN);
b5.setBackground(Color.CYAN);
b6.setBackground(Color.CYAN);
b4.setBounds(230,205,200,20);
b5.setBounds(440,205,200,20);
b6.setBounds(650,205,200,20);
b4.setSelected(true);
screen.add(b4);
screen.add(b5);
screen.add(b6);
ButtonGroup bg2 = new ButtonGroup();
bg2.add(b4);
bg2.add(b5);
bg2.add(b6);

JLabel l5 = new JLabel("Size = ");
l5.setBounds(70,270,150,20);
l5.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l5);
JRadioButton b7 = new JRadioButton("Regular");
JRadioButton b8 = new JRadioButton("Medium(150)");
```

```
JRadioButton b9 = new JRadioButton("Large(250)");
b7.setFont(new Font("Verdana", Font.BOLD, 14));
b8.setFont(new Font("Verdana", Font.BOLD, 14));
b9.setFont(new Font("Verdana", Font.BOLD, 14));
b7.setBackground(Color.CYAN);
b8.setBackground(Color.CYAN);
b9.setBackground(Color.CYAN);
b7.setBounds(230,270,200,20);
b8.setBounds(440,270,200,20);
b9.setBounds(650,270,200,20);
b7.setSelected(true);
screen.add(b7);
screen.add(b8);
screen.add(b9);
ButtonGroup bg3 = new ButtonGroup();
bg3.add(b7);
bg3.add(b8);
bg3.add(b9);

JLabel l6 = new JLabel("Toppings = ");
l6.setBounds(70,335,150,20);
l6.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l6);
JCheckBox cb1 = new JCheckBox("Extra Cheese(75)");
JCheckBox cb2 = new JCheckBox("Jalapeno(60)");
JCheckBox cb3 = new JCheckBox("Pepperoni(75)");
```

```
cb1.setFont(new Font("Verdana", Font.BOLD, 14));
cb2.setFont(new Font("Verdana", Font.BOLD, 14));
cb3.setFont(new Font("Verdana", Font.BOLD, 14));
cb1.setBackground(Color.CYAN);
cb2.setBackground(Color.CYAN);
cb3.setBackground(Color.CYAN);
cb1.setBounds(230,335,200,20);
cb2.setBounds(440,335,300,20);
cb3.setBounds(650,335,300,20);
screen.add(cb1);
screen.add(cb2);
screen.add(cb3);

JLabel l7 = new JLabel("Quantity = ");
l7.setBounds(70,400,100,20);
l7.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(l7);
JTextField t3 = new JTextField("1");
t3.setBounds(230,395,200,30);
t3.setFont(new Font("Verdana", Font.PLAIN, 14));
screen.add(t3);

JTextArea bill = new JTextArea("");
bill.setBounds(70,465,850,150);
bill.setFont(new Font("Verdana", Font.PLAIN, 18));
screen.add(bill);
```

```
JButton button1 = new JButton("Generate Bill");
button1.setBounds(650,395,150,30);
button1.setFont(new Font("Verdana", Font.BOLD, 14));
screen.add(button1);
button1.addActionListener(new ActionListener()
{
   public void actionPerformed(ActionEvent e)
   {
      String s1 = t1.getText();
      String s2 = t2.getText();
      String s3 = t3.getText();
      int quantity = Integer.parseInt(s3);
      if(s1.isEmpty() || s2.isEmpty() || quantity <= 0)
      {
        JOptionPane.showMessageDialog(null, "Please out the details correctly", "Invalid
TextFields", JOptionPane.ERROR_MESSAGE);
        return;
      }
      else
      {
        double total=0;
        String ans,size="",crust="",pizza="",toppings="";
        if(b1.isSelected())
        {
          total+=100;
```

```
  pizza = "Margherita";
}
else if(b2.isSelected())
{
  total+=305;
  pizza = "Chicken Tikka";
}
else if(b3.isSelected())
{
  total+=215;
  pizza = "Farmhouse";
}
if(b4.isSelected())
{
  total+=0;
  crust = "Hand Tossed";
}
else if(b5.isSelected())
{
  total+=40;
  crust = "Pan";
}
else if(b6.isSelected())
{
  total+=100;
  crust = "Cheese Burst";
```

```
}
if(b7.isSelected())
{
  total+=0;
  size = "Regular";
}
else if(b8.isSelected())
{
  total+=150;
  size = "Medium";
}
else if(b9.isSelected())
{
  total+=250;
  size = "Large";
}
if(cb1.isSelected())
{
  total+=75;
  toppings += " Extra Cheese,";
}
if(cb2.isSelected())
{
  total+=60;
  toppings += " Jalapeno,";
}
```

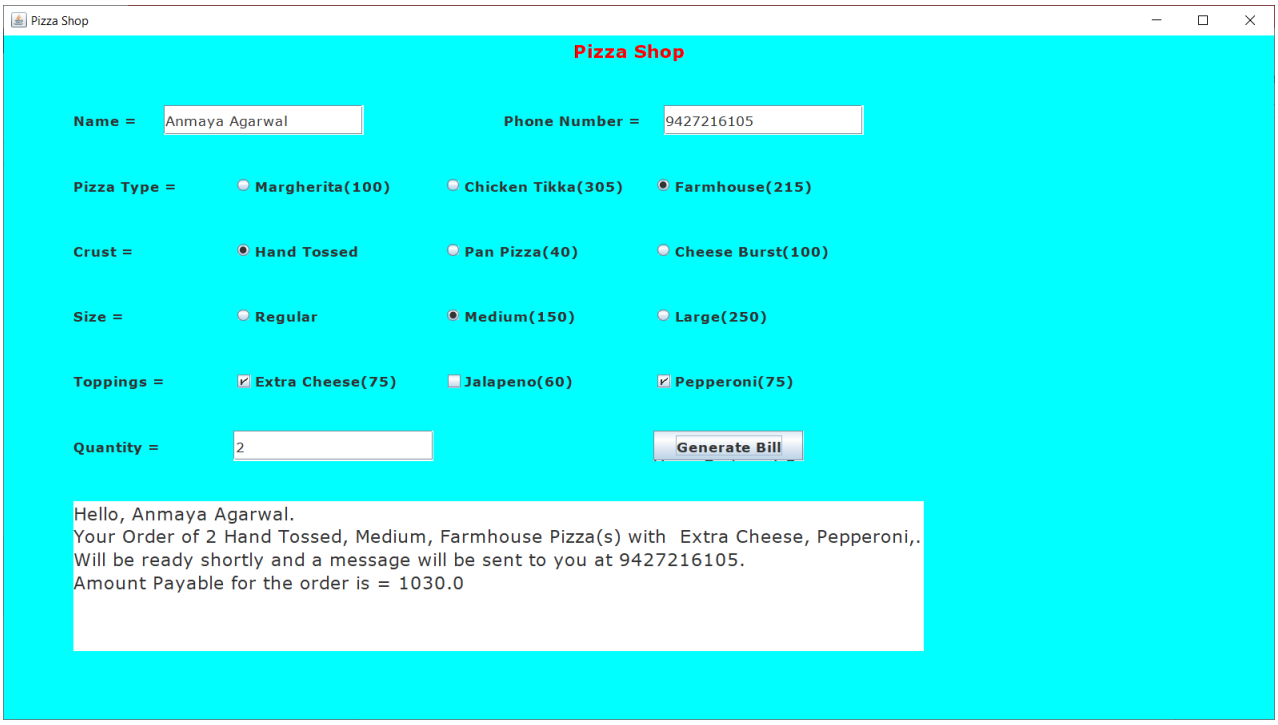
```
            if(cb3.isSelected())
            {
              total+=75;
              toppings += " Pepperoni,";
            }
            total = total * quantity;
            ans = "Hello, " +s1+ ". \nYour Order of " +quantity+ " " +crust+ ", " +size+ ", "
+pizza+ " Pizza(s) with " +toppings+ ". \nWill be ready shortly and a message will be sent to you
at " +s2+ ". \nAmount Payable for the order is = " +total;
            bill.setText(ans);
          }
        }
      });
    }
}
```

Main Class (testPizzaShop) –

```
import javax.swing.*;
import java.awt.*;


public class testPizzaShop
{
   public static void main(String[] args)
   {
      JFrame screen = new JFrame ("Pizza Shop");
      screen.setSize (1280,720);
      screen.setLayout(null);
      screen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      screen.setVisible(true);
      screen.getContentPane().setBackground(Color.CYAN);
      new pizzaShop(screen);
   }
}
```

Output:



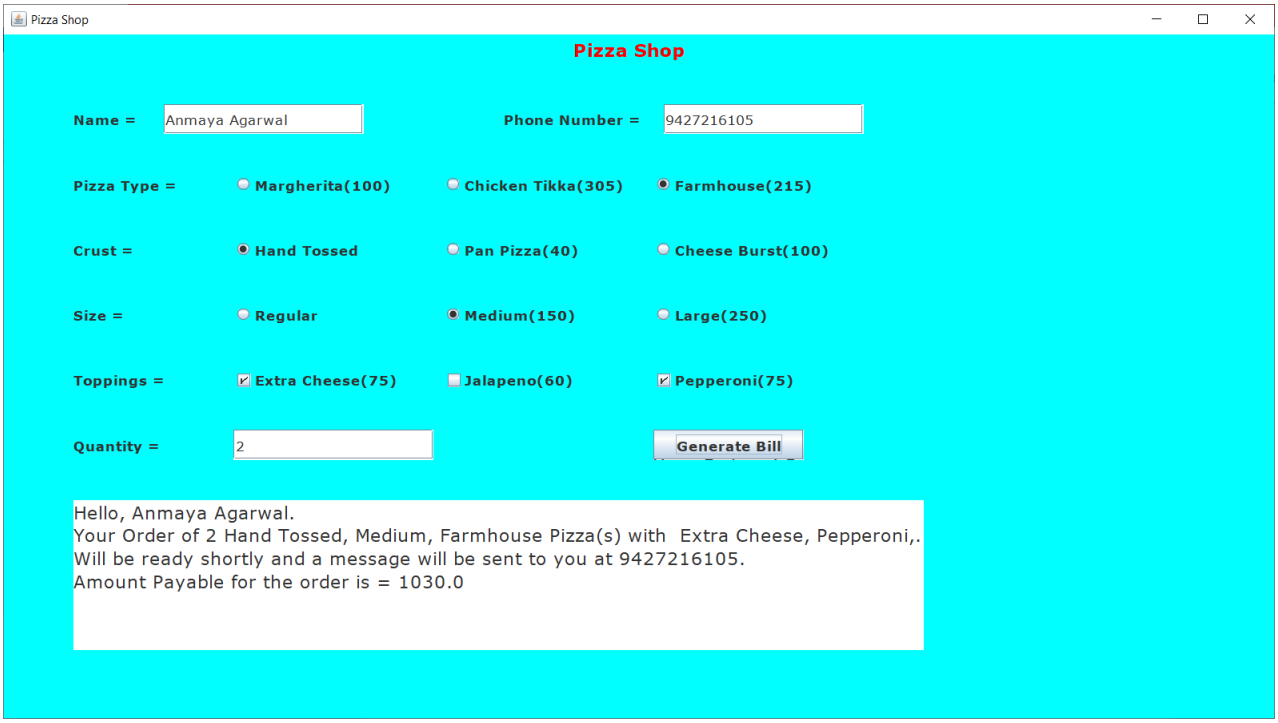
B.3 Conclusion:

We have created a GUI program for ordering and customizing your pizza with the type of crust, toppings. We learnt how to implement separate GUI elements like radio buttons and checkboxes in a single program

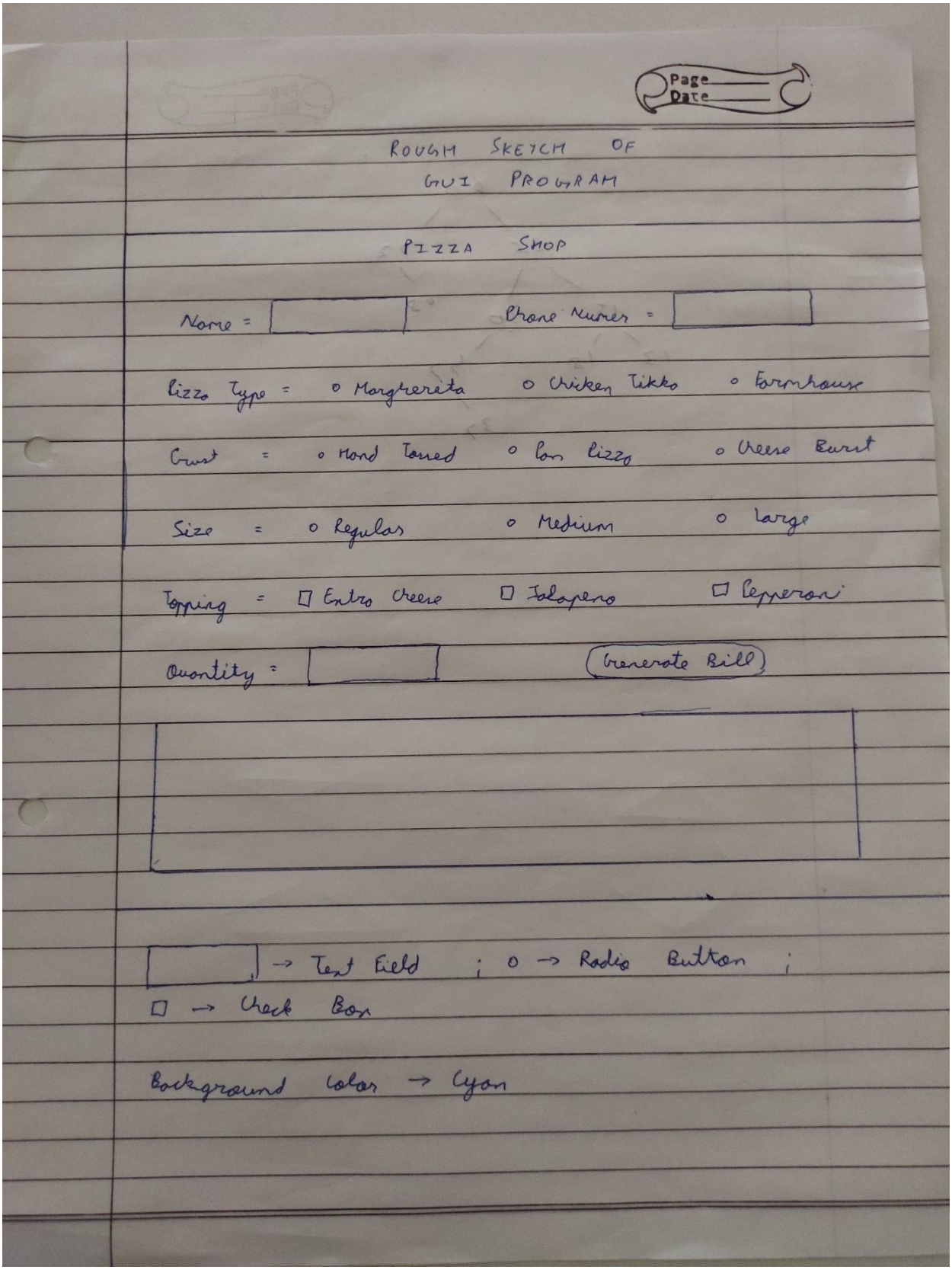
TESTING PLAN

Aim: Test your GUI code from Mini Project Pizza Shop

Real Output -



Rough Sketch of Output –



Test Plan –

Test No.	Test Category	Purpose of Test	Expected Result	Actual Result	Outcome
1)	Dimensions and alignments of input elements	Testing the size, position, width, height of input name.	The text field should be of dimensions 200x30 and it should be at a position of 160, 70 according to the GUI coordinate system.	The size of the element is as expected and it is positioned correctly according to the GUI coordinate system	Everything is as expected. No actions required.
		Testing the size, position, width, height of input phone number.	The text field should be of dimensions 200x30 and it should be at a position of 660, 70 according to the GUI coordinate system.	The size of the element is as expected and it is positioned correctly according to the GUI coordinate system	
		Testing the size, position, width, height of input pizza type.	The radio buttons should be of dimensions 200x20 and they should be at distance of 140 from the top of the screen and each should have an increment of 210 for the x-coordinate starting from 230	The size of the elements is as expected and it is positioned correctly as expected	

		Testing the size, position, width, height of input crust.	The radio buttons should be of dimensions 200x20 and they should be at distance of 205 from the top of the screen and each should have an increment of 210 for the x-coordinate starting from 230	The size of the elements is as expected and it is positioned correctly as expected	
		Testing the size, position, width, height of input size.	The radio buttons should be of dimensions 200x20 and they should be at distance of 270 from the top of the screen and each should have an increment of 210 for the x-coordinate starting from 230	The size of the elements is as expected and it is positioned correctly as expected	
		Testing the size, position, width, height of input toppings.	The checkbox buttons should be of dimensions 200x20 and they should be at distance of 335 from the top of the screen and each should have an increment of 210 for the x-coordinate starting from 230	The size of the elements is as expected and it is positioned correctly as expected	

		Testing the size, position, width, height of input quantity.	The text field should be of dimensions 100x20 and it should be at a position of 70, 400 according to the GUI coordinate system.	The size of the element is as expected and it is positioned correctly according to the GUI coordinate system	
		Testing the size, position, width, height of output result.	The text area should be of dimensions 850x150 and it should be at a position of 70, 465 according to the GUI coordinate system.	The size of the element is as expected and it is positioned correctly according to the GUI coordinate system	
		Testing the size, position, width, height of "generate bill" button.	The button should be of dimensions 150x30 and it should be at a position of 650, 395 according to the GUI coordinate system.	The size of the element is as expected and it is positioned correctly according to the GUI coordinate system	
2)	Error Messages	Testing the error message that should be displayed if the user doesn't enter a name	If the user does not enter a name display an error message saying "Please fill out the details correctly".	When the user leaves the name input as blank an error message saying "Please fill out the details correctly" is displayed.	Everything is as expected. No actions required.

		Testing the error message that should be displayed if the user doesn't enter a phone number	If the user does not enter a phone number display an error message saying "Please fill out the details correctly".	When the user leaves the phone number input as blank an error message saying "Please fill out the details correctly" is displayed.	
		Testing the error message that should be displayed if the user enters a negative quantity	If the user enters a negative quantity display an error message saying "Please fill out the details correctly".	When the user enter the quantity as negative an error message saying "Please fill out the details correctly" is displayed.	
3)	Font Testing	Testing whether the font is readable or not, the font size and font face of every label in the GUI – Input name, phone number, pizza type, crust, size, toppings, quantity and generate bill button.	The font should be readable and should be visible clearly. The font size and font face of every label in the GUI – Input name, phone number, pizza type, crust, size, toppings, quantity and generate bill button should be 14 and "Verdana" respectively	The font is readable and is visible clearly The font size and font face of every label in the GUI – Input name, phone number, pizza type, crust, size, toppings, quantity and generate bill button is 14 and "Verdana" respectively	Everything is as expected. No actions required.
		Testing whether the font is readable or not, the font size and font face of the heading label "Pizza Shop"	The font should be readable and should be visible clearly. The font size and font face of the label "Pizza Shop" should be 18 and "Verdana" respectively.	The font is readable and is visible clearly The font size and font face of the label "Pizza Shop" is 18 and "Verdana" respectively.	

		Testing whether the font is readable or not, the font size and font face of the output result	The font should be readable and should be visible clearly. The font size and font face of the output should be 18 and "Verdana" respectively.	The font is readable and is visible clearly The font size and font face of the output is 18 and "Verdana" respectively.	
4)	Spelling test	Testing whether the spellings of each and every heading and label is correct.	The spelling should be correct for every word in the GUI program.	The spellings are correct for every word, heading and label in the GUI.	Everything is as expected. No action required.
5)	Design test	Testing the background colors and overall design of the program. Testing whether the program is user friendly or not.	The background color should be cyan and overall design should be as shown in the rough sketch. The program should not be complicated and should be user friendly	The background color is cyan and the overall design is as shown in the rough sketch. The program is intuitive and not complicated and is quite user friendly	Everything is as expected. No actions required.

B.3 Conclusion:

We had created a GUI program for ordering and customizing your pizza with the type of crust, toppings for out mini project. We have now implemented a testing for the same GUI program where we have extensively tested the GUI program and made sure that there are no errors and the overall design and usage is user friendly.