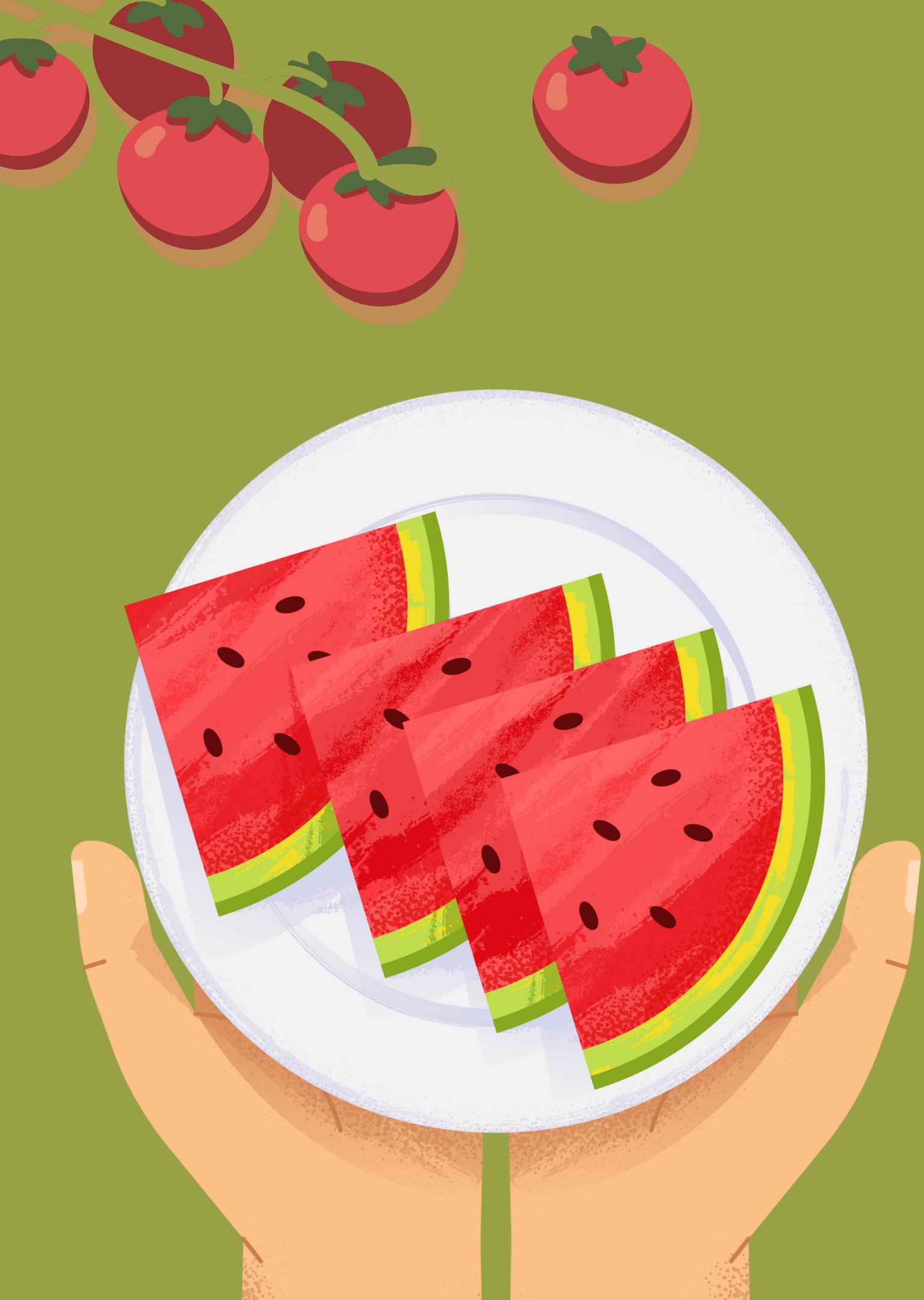


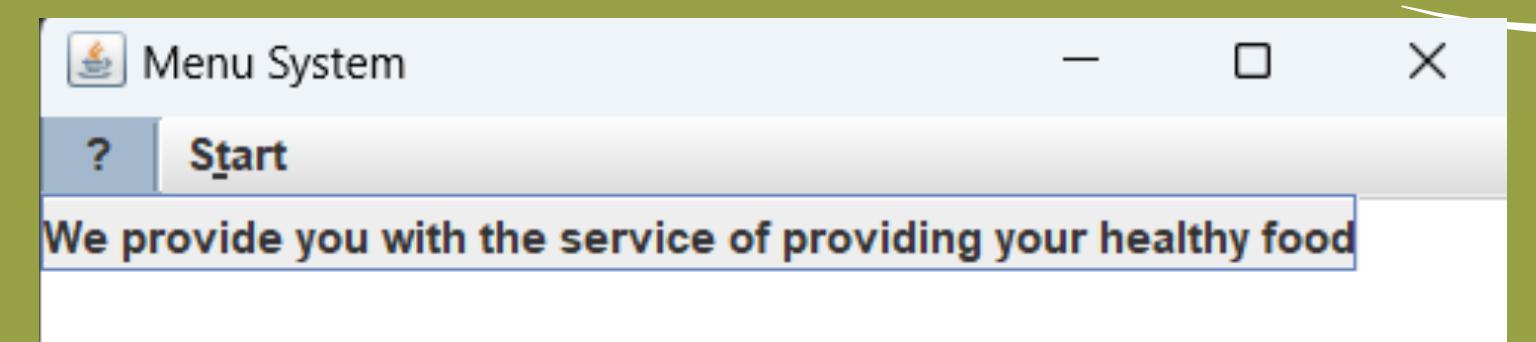
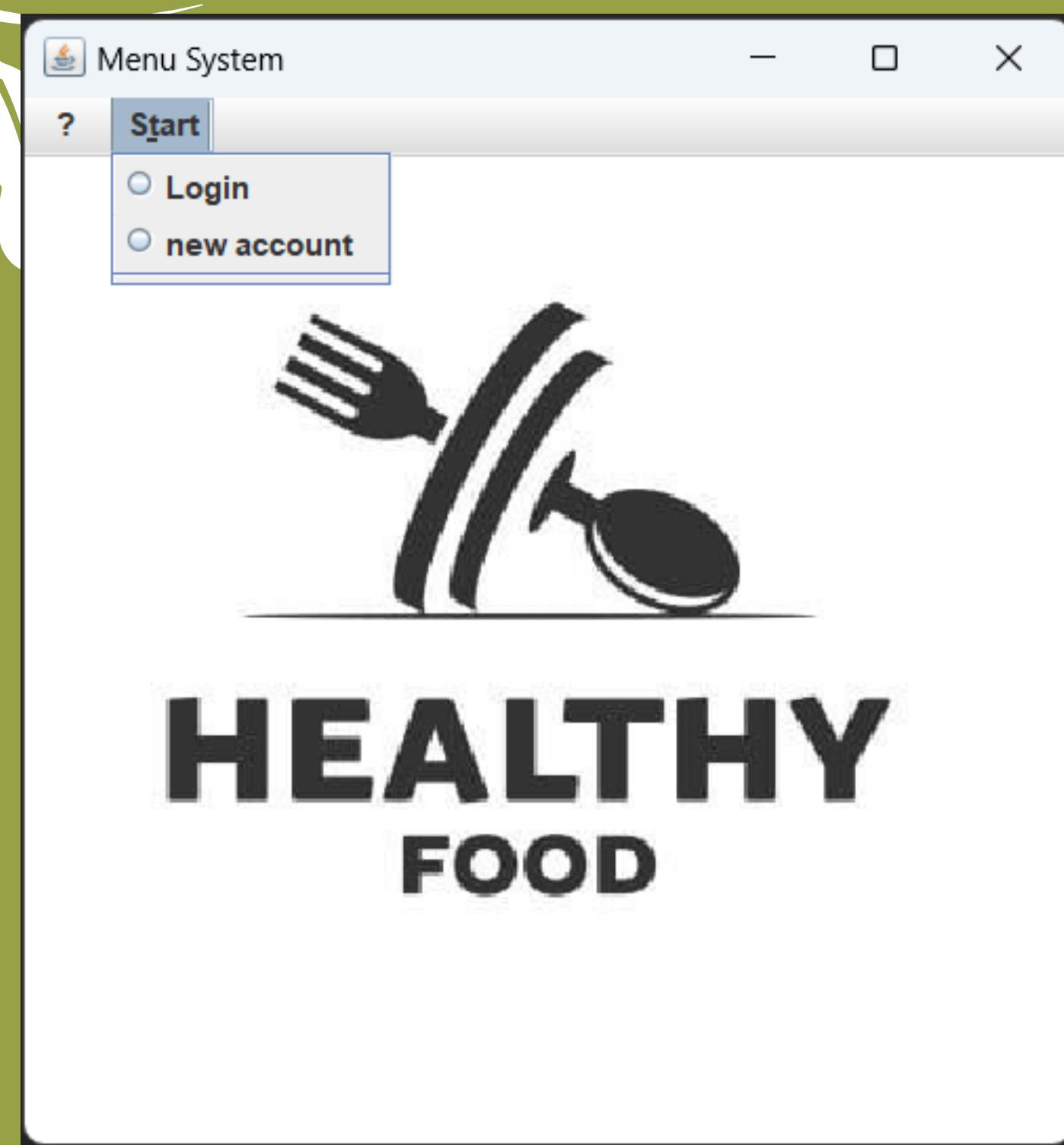


healthy meals application

live healthy



- 
- Khalid Ibrahim Nimri ID:
 - Naif Hussain Alayaid ID :
 - Mohammed Loay Allagany ID:
 - Naif Mersal Al-mehmadi ID:



Homepage



```
6 usages
public class menu extends JFrame {

    3 usages
    private JLabel messageLabel;
    1 usage
    private final int LABEL_WIDTH = 399;
    1 usage
    private final int LABEL_HEIGHT = 375;

    4 usages
    private JMenuBar menuBar;
    4 usages
    private JMenu fMenu;
    6 usages
    private JMenu tMenu;

    2 usages
    private JLabel exitItem;
    6 usages
    private JRadioButtonMenuItem Login;
    6 usages
    private JRadioButtonMenuItem newRegistration;
    1 usage
    ImageIcon Img = new ImageIcon( filename: "HealthyFood2.jpeg");
    2 usages
    private JLabel imgLabel = new JLabel(Img);
    private JPanel imgPanel;
```



3 usages

```
public menu()
{
    setTitle(" Menu System");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    imgLabel.setBounds( x: 0, y: 0, width: 399, height: 375);
    add(imgLabel);
    add(messageLabel);
    buildMenuBar();
    setJMenuBar(menuBar);
    pack();
    setLocationRelativeTo(null);
    setVisible(true);
}

1 usage
```

```
private void buildMenuBar()
{
    menuBar = new JMenuBar();

    buildFileMenu();
    buildTextMenu();

    menuBar.add(fMenu);
    menuBar.add(tMenu);

}
```

```
1 usage
private void buildFileMenu()
{
    exitItem = new JLabel( text: "We provide you with the service of providing your healthy food");
    fMenu = new JMenu( s: " ? ");
    fMenu.setMnemonic(KeyEvent.VK_F);
    fMenu.add(exitItem);
}
```



```
private void buildTextMenu()
{
    Login = new JRadioButtonMenuItem( text: "Login");
    Login.setMnemonic(KeyEvent.VK_B);
    Login.addActionListener(new Listener());

    newRegistration = new JRadioButtonMenuItem( text: "new account");
    newRegistration.setMnemonic(KeyEvent.VK_R);
    newRegistration.addActionListener(new Listener());

    ButtonGroup group = new ButtonGroup();
    group.add(Login);
    group.add(newRegistration);

    tMenu = new JMenu( s: "Start");
    tMenu.setMnemonic(KeyEvent.VK_T);

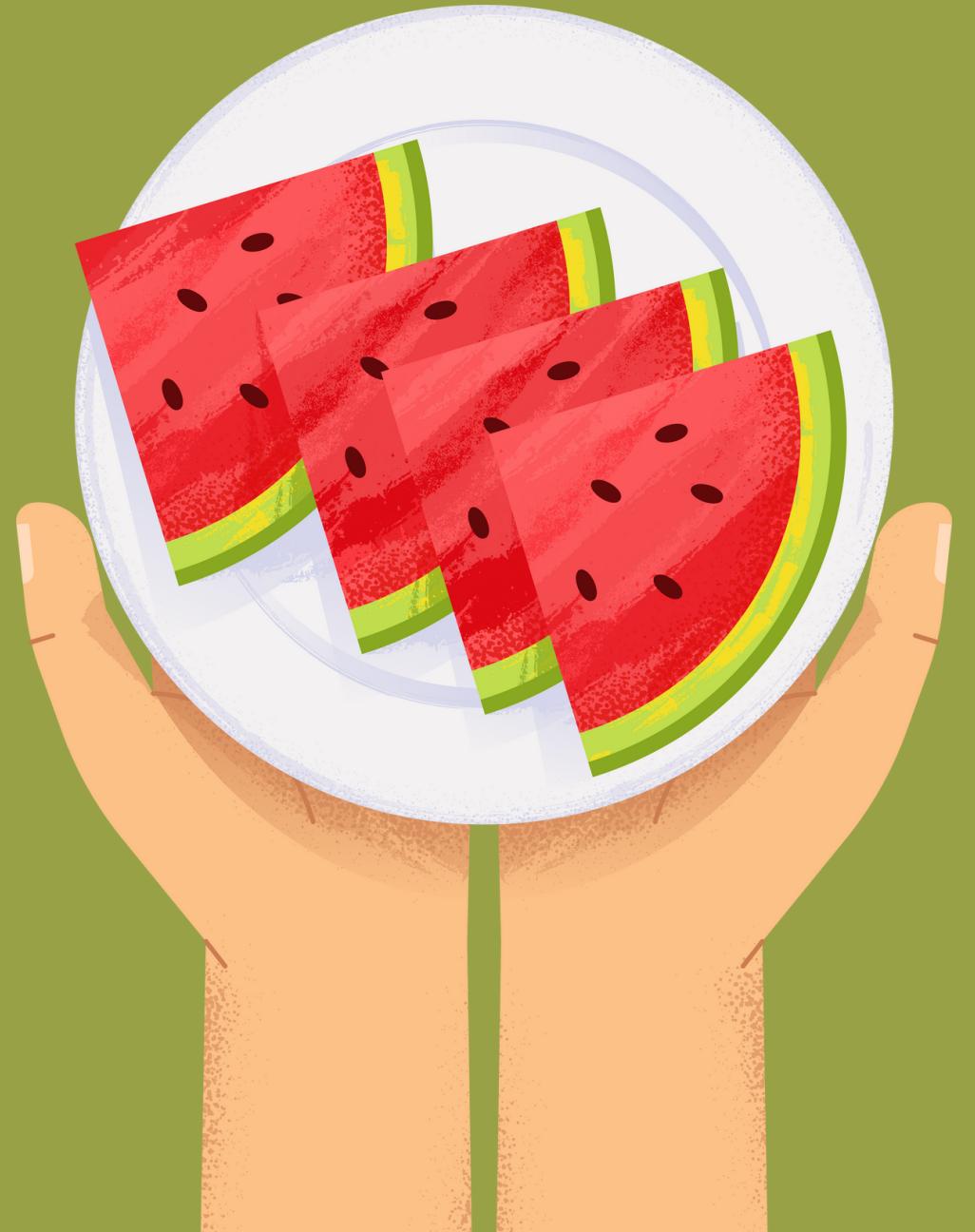
    tMenu.add(Login);
    tMenu.add(newRegistration);
    tMenu.addSeparator();
}
```



2 usages

```
private class Listener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if (Login.isSelected()){
            dispose();
            LogIn newWi = new LogIn();

        }
        else if (newRegistration.isSelected()) {
            dispose();
            NewCs NC = new NewCs();
        }
    }
}
```



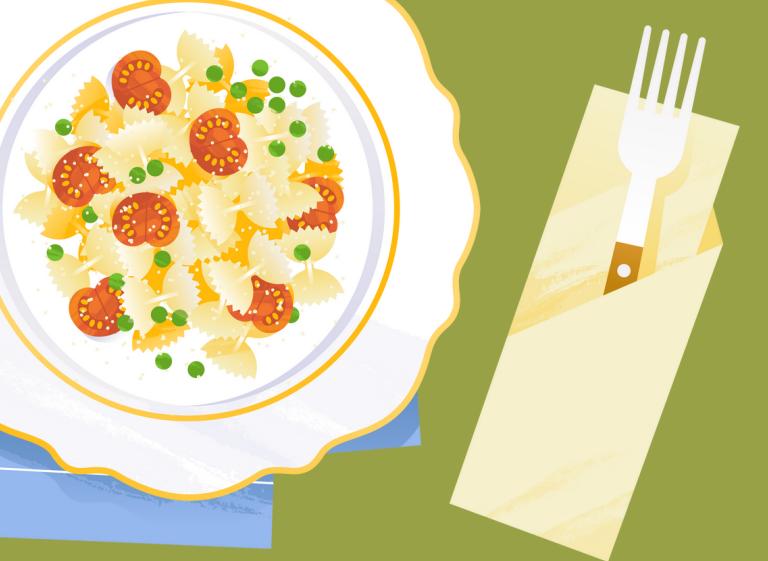
Login page

login page

Enter your user name

Enter your phone

Cancel **ok**



```
public class LogIn extends JFrame {
```

4 usages

```
    private JPanel panel1;
```

4 usages

```
    private JPanel panel2;
```

4 usages

```
    private JPanel panel3;
```

2 usages

```
    private JLabel label2;
```

2 usages

```
    private JLabel label1;
```

2 usages

```
    private JTextField text;
```

2 usages

```
    private JTextField text2;
```

3 usages

```
    private JButton button1;
```

3 usages

```
    private JButton button2;
```

```
public LogIn() {  
  
    setTitle("login page");  
    setLayout(new GridLayout( rows: 3 , cols: 1));  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setSize( width: 400 , height: 200);  
    setLocationRelativeTo(null);  
    setVisible(true);  
  
  
  
  
    panel1 = new JPanel();  
    panel2 = new JPanel();  
    panel3 = new JPanel();  
  
  
  
  
    label1 = new JLabel( text: "Enter your user name");  
    panel1.add(label1);  
    text = NewCs.text1 = new JTextField( columns: 10);  
    panel1.add(text);  
    label2 = new JLabel( text: "Enter your phone");  
    panel2.add(label2);  
    text2 = NewCs.text3 = new JTextField( columns: 10);  
    panel2.add(text2);  
    button1 = new JButton( text: "ok");  
    button1.addActionListener(new ActionListener() {
```



```
public  
void actionPerformed(ActionEvent e) {  
    setVisible(false);  
    o1 = new Order();  
}  
});  
button2 = new JButton( text: "Cancel");  
button2.addActionListener(new ActionListener() {  
    public  
    void actionPerformed(ActionEvent e) {  
        setVisible(false);  
        menu a = new menu();  
    }  
});  
panel3.add(button2);  
panel3.add(button1);  
  
add(panel1); //BorderLayout.NORTH  
add(panel2); // BorderLayout.CENTER  
add(panel3); // BorderLayout.SOUTH  
  
setVisible(true);
```



New Account

new account

Enter the username

Enter your full name

Enter your phone

Enter your City

Enter your address



```
20 usages
public class NewCs extends JFrame {
    5 usages
    private JPanel panel1;
    4 usages
    private JPanel panel2;
    4 usages
    private JPanel panel3;
    4 usages
    private JPanel panelT1;
    4 usages
    private JPanel panelT2;
    4 usages
    private JPanel panelT3;
    4 usages
    private JPanel panelT4;
    4 usages
    private JPanel panelT5;
    2 usages
    private JLabel label1;
    2 usages
    private JLabel label2;
```

```
private JLabel label3;
2 usages
private JLabel label4;
2 usages
private JLabel label5;
5 usages
static JTextField text1;
4 usages
static JTextField text2;
5 usages
static JTextField text3;
4 usages
static JTextField text4;
4 usages
static JTextField text5;
3 usages
private JButton button1;
3 usages
private JButton button2;
1 usage
ImageIcon Img = new ImageIcon( filename: "HealthyFood2.jpeg");
private JLabel imgLabel = new JLabel(Img , JLabel.CENTER);
private JPanel imgPanel;
1 usage
static Order 01;
```

```
4 usages
public NewCs() {

    setTitle("new account");
    setLayout(new BorderLayout());
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setSize( width: 400, height: 250);
    setLocationRelativeTo(null);
    setVisible(true);

    panel1 = new JPanel(new BorderLayout());
    panel2 = new JPanel(new BorderLayout());
    panel3 = new JPanel();

    |
    panelT1 =new JPanel();
    panelT2 =new JPanel();
    panelT3 =new JPanel();
    panelT4 =new JPanel();
    panelT5 =new JPanel();

    label1 = new JLabel( text: "Enter the username");
    text1 = new JTextField( columns: 10);

    label2 = new JLabel( text: "Enter your full name");
    text2 = new JTextField( columns: 10);
    label3 = new JLabel( text: "Enter your phone");
    text3 = new JTextField( columns: 10);
```

```
label4 = new JLabel( text: "Enter your City");
text4 = new JTextField( columns: 10);
label5 = new JLabel( text: "Enter your address");
text5 = new JTextField( columns: 10);

panelT1.add(label1);
panelT1.add(text1);

panelT2.add(label2);
panelT2.add(text2);

panelT3.add(label3);
panelT3.add(text3);

panelT4.add(label4);
panelT4.add(text4);

panelT5.add(label5);
panelT5.add(text5);

panel1.add(panelT1, BorderLayout.NORTH);
panel1.add(panelT2 , BorderLayout.CENTER);
panel1.add(panelT3 , BorderLayout.SOUTH);

panel2.add(panelT4,BorderLayout.NORTH);
panel2.add(panelT5,BorderLayout.CENTER);
```

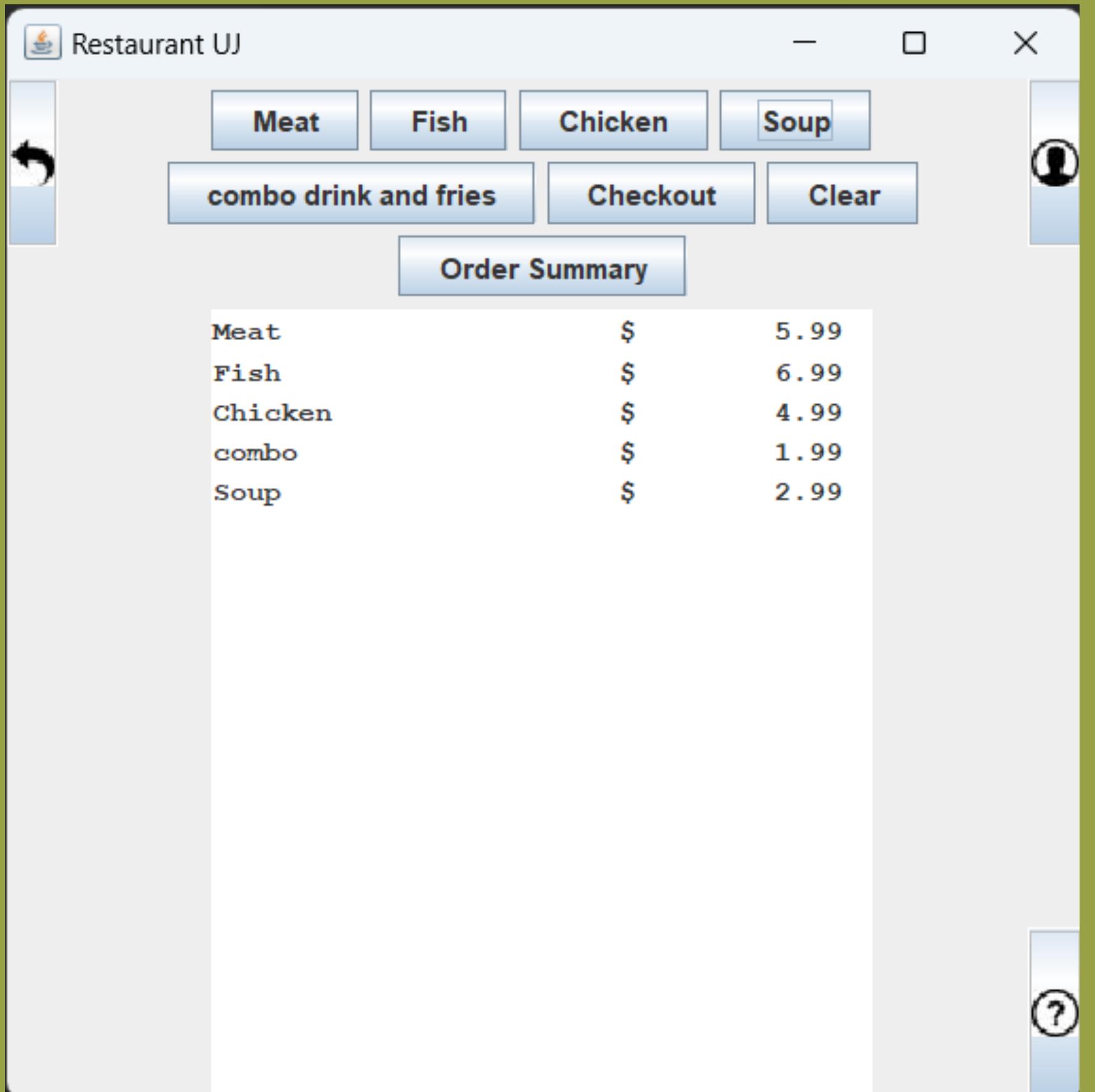
```
button1 = new JButton( text: "OK");
button1.addActionListener(new ActionListener() {
    public
    void actionPerformed(ActionEvent e) {
        setVisible(false);
        OI = new Order();
    }
});

button2= new JButton( text: "Cancel");
button2.addActionListener(new ActionListener() {
    public
    void actionPerformed(ActionEvent e) {
        setVisible(false);
        menu a = new menu();
    }
});

panel3.add(button2);
panel3.add(button1);

add(panel1 ,BorderLayout.NORTH);
add(panel2 , BorderLayout.CENTER);
add(panel3, BorderLayout.SOUTH);
```





List of orders



```
public class Order extends JFrame {  
  
    4 usages  
    private JButton btnMeat;  
    4 usages  
    private JButton btnFish;  
    4 usages  
    private JButton btnChicken;  
    4 usages  
    private JButton btnSoup;  
    4 usages  
    private JButton btnCombo;  
    4 usages  
    private JButton btnTotal;  
    4 usages  
    private JButton btnClear;  
    2 usages  
    private JButton Backbtn;  
    3 usages  
    private JButton Customerbtn;  
    4 usages  
    private JButton btnCustomerPanel;  
    12 usages  
    static JTextArea billArea;  
    4 usages  
    static int counterFish = 0;  
    4 usages  
    static int counterMeat = 0;  
    4 usages  
    static int counterSoup = 0;  
  
    4 usages  
    static int counterCombo = 0;  
    4 usages  
    static int counterChicken = 0;  
    5 usages  
    JPanel rightPanel=new JPanel();  
    4 usages  
    JPanel leftPanel=new JPanel();  
    11 usages  
    JPanel centerPanel;  
    1 usage  
    ImageIcon helpImg = new ImageIcon( filename: "help.jpeg");  
    1 usage  
    ImageIcon returnImg = new ImageIcon( filename: "return.jpeg");  
    1 usage  
    ImageIcon userImg = new ImageIcon( filename: "user.jpeg");  
    3 usages  
    JButton returnBtn =new JButton(returnImg);  
    3 usages  
    JButton helpBtn =new JButton(helpImg);  
    3 usages  
    JButton userBtn =new JButton(userImg);  
    2 usages  
    static String typeMeat;  
    2 usages  
    static String typeFish;  
    2 usages  
    static String typeChicken;  
    2 usages  
    static String typeSoup;
```

4 usages

```
public Order() {  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setTitle("Restaurant UJ");  
    setSize( width: 470, height: 470);  
    setLocationRelativeTo(null);  
    setVisible(true);  
    Container cp = getContentPane();  
    cp.setLayout(new FlowLayout());  
  
    centerPanel = new JPanel();  
  
    btnMeat = new JButton( text: "Meat");  
    btnFish = new JButton( text: "Fish");  
    btnChicken = new JButton( text: "Chicken");  
    btnSoup = new JButton( text: "Soup");  
    btnCombo = new JButton( text: "combo drink and fries");  
    btnTotal = new JButton( text: "Checkout");  
    btnClear = new JButton( text: "Clear");  
    btnCustomerPanel = new JButton( text: "Order Summary");  
    billArea = new JTextArea( rows: 25, columns: 40);  
    billArea.setEditable(false);  
    Font font = new Font( name: "Monospaced", Font.BOLD, size: 12);  
    billArea.setFont(font);  
    Backbtn = new JButton( text: "Back");  
  
    Backbtn.addActionListener(new ActionListener() {
```

```
        public  
        void actionPerformed(ActionEvent e) {  
            setVisible(false);  
            LogIn newWi = new LogIn();  
        }  
});  
Customerbtn = new JButton( text: "Account");  
Customerbtn.addActionListener(new ActionListener() {  
  
        public  
        void actionPerformed(ActionEvent e) {  
            setVisible(false);  
            CustomerPanel a = new CustomerPanel();  
        }  
});  
  
cp.add(btnMeat);  
cp.add(btnFish);  
cp.add(btnChicken);  
cp.add(btnSoup);  
cp.add(Customerbtn);  
cp.add(btnCombo);  
cp.add(btnTotal);  
cp.add(btnClear);  
cp.add(btnCustomerPanel);  
cp.add(billArea);
```

```
centerPanel.add(btnMeat);
centerPanel.add(btnFish);
centerPanel.add(btnChicken);
centerPanel.add(btnSoup);
centerPanel.add(btnCombo);
centerPanel.add(btnTotal);
centerPanel.add(btnClear);
centerPanel.add(btnCustomerPanel);
centerPanel.add(billArea);
setLayout(new BorderLayout());
buildLeftPanel();
buildRightPanel();

btnCustomerPanel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        CustomerPanel r = new CustomerPanel();
    }
});
```

```
btnMeat.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        ++counterMeat;
        billArea.append("Meat\t\t\t$\t 5.99\n");
        total += 5.99;
        typeMeat = String.valueOf( obj: "Meat\t$" + String.valueOf( d: 5.99));
        CustomerPanel.orders[MainCounter++]=typeMeat;
    }
});

btnFish.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        ++counterFish;
        billArea.append("Fish\t\t\t$\t 6.99\n");
        total += 6.99;
        typeFish = String.valueOf( obj: "Fish\t$" + String.valueOf( d: 6.99));
        CustomerPanel.orders[MainCounter++]=typeFish;
    }
});

btnChicken.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        ++counterChicken;
```

```
public void actionPerformed(ActionEvent e) {
    billArea.setText("");
    total = 0;
    counterChicken=0;
    counterCombo=0;
    counterFish=0;
    counterMeat=0;
    counterSoup=0;
    String []orders=new String[100];
    orders[0] = null;
    orders[1] = null;
    orders[2] = null;
    orders[3] = null;
    orders[4] = null;
    orders[5] = null;
    orders[6] = null;
    orders[7] = null;
    orders[8] = null;
    orders[9] = null;
    orders[10] = null;
    orders[11] = null;
    orders[12] = null;
    orders[13] = null;
    orders[14] = null;
    orders[15] = null;
    orders[16] = null;
    orders[17] = null;
    orders[18] = null;
    orders[19] = null;
```

```
1 usage
public void buildLeftPanel(){
    leftPanel.setLayout(new GridLayout( rows: 6, cols: 1));
    returnBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    returnBtn.addActionListener(new ActionListener() {
        public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            LogIn newWi = new LogIn();
        }
    });
    leftPanel.add(returnBtn);
    for(int i=0;i<3;i++){
        leftPanel.add(new JPanel());
    }
}
1 usage
public void buildRightPanel(){
    rightPanel.setLayout(new GridLayout( rows: 6, cols: 1));
    userBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    userBtn.addActionListener(new ActionListener() {

        public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            NewCs o = new NewCs();
            o.setTitle("Edit information");
        }
    });
}
```

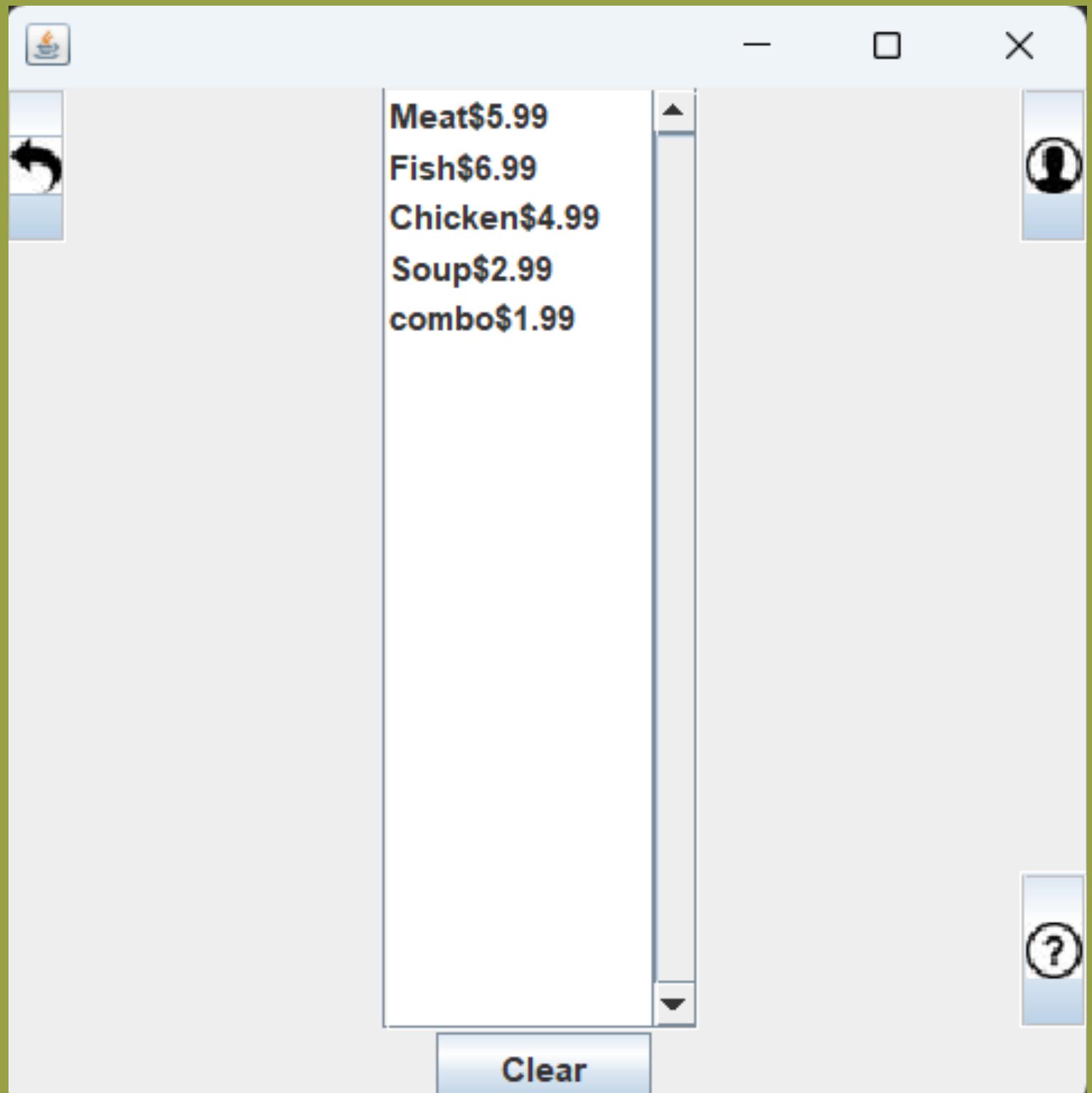


```
        });

    rightPanel.add(userBtn);
    for(int i=0;i<4;i++){
        rightPanel.add(new JPanel());
    }
    helpBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    helpBtn.addActionListener(new ActionListener() {
        public
        void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog( parentComponent: null, message: "In this page you can add the order to the basket.");
        }
    });
    rightPanel.add(helpBtn);
    add(centerPanel, BorderLayout.CENTER);
    add(leftPanel, BorderLayout.WEST);
    add(rightPanel, BorderLayout.EAST);
}
```



Order Summary



```
5 usages
 JPanel rightPanel=new JPanel();
4 usages
 JPanel leftPanel=new JPanel();
5 usages
 JPanel bottomPanel=new JPanel();
5 usages
 JPanel centerPanel=new JPanel();
1 usage
 JPanel orderPanel=new JPanel();
3 usages
 JList orderList=new JList();
3 usages
 JScrollPane pane;
1 usage
 ImageIcon helpImg = new ImageIcon( filename: "help.jpeg");
1 usage
 ImageIcon returnImg = new ImageIcon( filename: "return.jpeg");
1 usage
 ImageIcon userImg = new ImageIcon( filename: "user.jpeg");
2 usages
 static JButton clearBtn =new JButton( text: "Clear");
3 usages
 JButton returnBtn =new JButton(returnImg);
3 usages
 JButton helpBtn =new JButton(helpImg);
3 usages
 JButton userBtn =new JButton(userImg);
52 usages
 static Object orders[] = new Object[20];
```

```
2 usages
public CustomerPanel(){
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize( width: 400, height: 400);
    setLocationRelativeTo(null);
    setLayout(new BorderLayout());
    buildLeftPanel();
    buildRightPanel();
    buildBottomPanel();
    buildCenterPanel();
    add(centerPanel,BorderLayout.CENTER);
    add(bottomPanel,BorderLayout.SOUTH);
    add(leftPanel,BorderLayout.WEST);
    add(rightPanel,BorderLayout.EAST);
    setVisible(true);

    clearBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            orders[0] = null;
            orders[1] = null;
            orders[2] = null;
            orders[3] = null;
            orders[4] = null;
            orders[5] = null;
            orders[6] = null;
```



```
1 usage
public void buildLeftPanel(){
    leftPanel.setLayout(new GridLayout( rows: 6, cols: 1));
    returnBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    returnBtn.addActionListener(new ActionListener() {
        public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            new Order();
            Order.counterChicken=0;
            Order.counterCombo=0;
            Order.counterFish=0;
            Order.counterMeat=0;
            Order.counterSoup=0;
        }
    });
    leftPanel.add(returnBtn);
    for(int i=0;i<3;i++){
        leftPanel.add(new JPanel());
    }
}
1 usage
public void buildRightPanel(){
    rightPanel.setLayout(new GridLayout( rows: 6, cols: 1));
    userBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    userBtn.addActionListener(new ActionListener() {
```



```
1 usage
public void buildRightPanel(){
    rightPanel.setLayout(new GridLayout( rows: 6 , cols: 1));
    userBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    userBtn.addActionListener(new ActionListener() {

        public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            NewCs o = new NewCs();
            o.setTitle("Edit information");
        }
    });

    rightPanel.add(userBtn);
    for(int i=0;i<4;i++){
        rightPanel.add(new JPanel());
    }
    helpBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray));
    helpBtn.addActionListener(new ActionListener() {
        public
        void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog( parentComponent: null, message: "In this page you can edit your information.");
        }
    });
    rightPanel.add(helpBtn);
}
1 usage
public void buildCenterPanel(){
    orderPanel.setLayout(new GridLayout( rows: 0 , cols: 1));
```

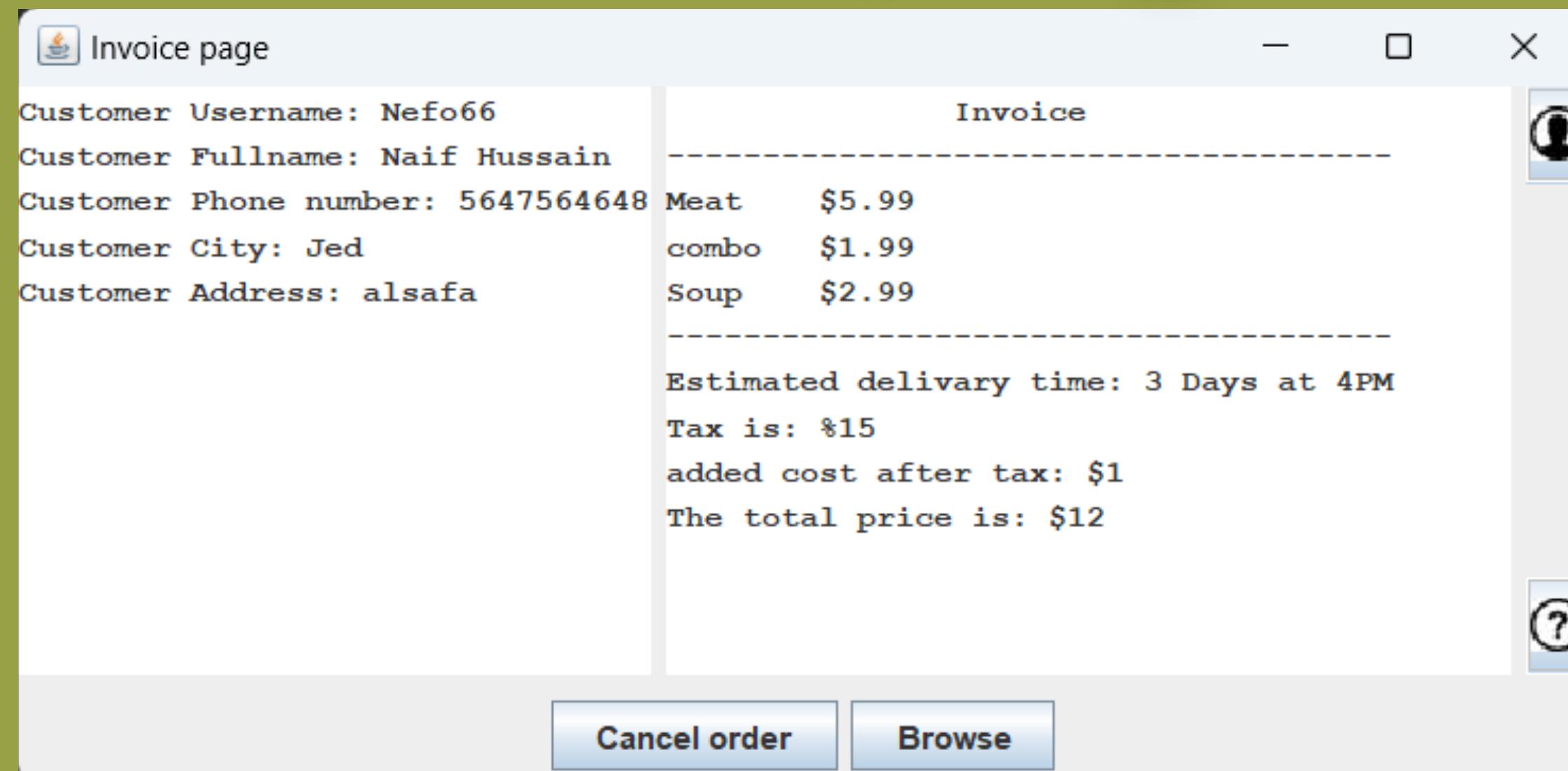




```
1 usage
public void buildCenterPanel(){
    orderPanel.setLayout(new GridLayout( rows: 0 , cols: 1));
    centerPanel.setLayout(new GridLayout( rows: 1 , cols: 3));
    centerPanel.add(new JPanel());
    orderList.setListData(orders);
    orderList.setVisibleRowCount(5);
    pane=new JScrollPane(orderList);
    pane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    centerPanel.add(pane);
    centerPanel.add(new JPanel());

}
1 usage
public void buildBottomPanel(){
    bottomPanel.setLayout(new GridLayout( rows: 1 , cols: 6));
    for(int i=0;i<2;i++) bottomPanel.add(new JPanel());
    bottomPanel.add(clearBtn);
    for(int i=0;i<2;i++) bottomPanel.add(new JPanel());
}
```

Invoice page



```
int status;
4 usages
int RandomDay;
4 usages
int RandomTime;
//////////////////////////// Double variables declaration
2 usages
double tax = 0.15;
2 usages
double PriceAddedAfterTax = Order.total*tax;
3 usages
double TotalAfterTax = 0;
////////////////// Random Object declaration
2 usages
Random Day = new Random();
2 usages
Random Time = new Random();
////////////////

1 usage
public Invoice(){                                // Invoice Constructor
    /////////////////////////////////
    setSize( width: 525, height: 300);
    setTitle("Invoice page");
    setLocationRelativeTo(null);                  // JFrame Properties
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout( hgap: 5, vgap: 5));
    setBackground(Color.gray);
    ///////////////////////////////
    buildCustomerArea();
    buildRightPanel();
}
```

```
buildRightPanel();
buildOrderSummaryArea();                         // Method
buildbottomPanel();
///////////////////////////////
add(InvoiceArea1, BorderLayout.WEST);            // Add
add(InvoiceArea2, BorderLayout.CENTER);           // Add
add(rightPanel, BorderLayout.EAST);
///////////////////////////////
InvoiceArea1.setFont(font);                    // Sett
InvoiceArea2.setFont(font);
///////////////////////////////
setVisible(true);                            // Sett
///////////////////////////////
}

1 usage
public void buildCustomerArea(){               // Built
    /////////////////////////////////
    if(NewCs.text1 != null){
        CustomerUserName = NewCs.text1.getText(); // <   <   < // T
    }
    else {
        CustomerUserName = null;                // <   <   < // I
    }
    if(NewCs.text2 != null){
        CustomerFullName = NewCs.text2.getText();
    }
    else {
        CustomerFullName = null;
    }
}
```

```
if(NewCs.text3 != null){ // Same operations ...
CustomerPhoneNumber = NewCs.text3.getText();
}
else{
    CustomerPhoneNumber = null;
}
if(NewCs.text4 != null){ // Same operations ...
CustomerCity = NewCs.text4.getText();
}
else {
    CustomerCity = null;
}
if(NewCs.text5 != null){ // Same operations ...
CustomerAddress = NewCs.text5.getText();
}
else {
    CustomerAddress = null;
}
//////////////////////////// Adding the user's information to the JTextArea
InvoiceArea1.append(String.valueOf( obj: "Customer Username: "+CustomerUserName) +'\n');
InvoiceArea1.append(String.valueOf( obj: "Customer Fullname: "+CustomerFullName) +'\n');
InvoiceArea1.append(String.valueOf( obj: "Customer Phone number: "+CustomerPhoneNumber) +'\n');
InvoiceArea1.append(String.valueOf( obj: "Customer City: "+CustomerCity) +'\n');
InvoiceArea1.append(String.valueOf( obj: "Customer Address: "+CustomerAddress) +'\n');
////////////////////////////
InvoiceArea1.setEditable(false); // Making the JTextArea not editable
////////////////////////////
FileChooserButton.setToolTipText("Click here to choose a file to save in \"Txt\""); // Hint for the button
////////////////////////////
setVisible(true); // Setting the visibility true
```



```
public void buildbottomPanel(){ // BuildbottomPanel method
    bottom_Cancelorder_FileChooser_panel.setSize( width: 20, height: 20); // Setting the bottom panel's size
    CancelOrderbtn.setToolTipText("Click here to cancel the order!"); // Hint for the cancel button
    CancelOrderbtn.addActionListener(new ActionListener() { // Adding ActionListener to the cancel button
    /////////////////////////////////
    public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            Order.billArea.setText("");
            int total = 0;
            int counterChicken = 0;
            int counterCombo=0;
            int counterFish=0;
            int counterMeat=0;
            int counterSoup=0;
            String []orders=new String[100];
            orders[0] = null;
            orders[1] = null;
            orders[2] = null;
            orders[3] = null;
            orders[4] = null;
            orders[5] = null; // If the button got clicked it will clear
            orders[6] = null; // And reset everything.
            orders[7] = null;
            orders[8] = null;
            orders[9] = null;
            orders[10] = null;
            orders[11] = null;
            orders[12] = null;
            orders[13] = null;
```



```
FileChooserButton.addActionListener(new ActionListener() { // Adding ActionListener to the FileChooser ▲ 45 ✘ 14 ^  
    public // Beginning of the ActionListener  
    void actionPerformed(ActionEvent e) { // Start of the ActionListener  
        FileChooser = new JFileChooser(); // Making the FileChooser Object  
        FileChooser.showSaveDialog(FileChooserButton); // Showing the save dialog window to browse  
  
        if (status == JFileChooser.APPROVE_OPTION){ // If the user clicked save then the following  
            File selectedFile = FileChooser.getSelectedFile(); // Will happen...  
            String FileName = selectedFile.getPath();  
            try // Try statement to check if there is Exceptions  
            {  
                FileWriter fw = new FileWriter(FileName);  
                PrintWriter OutPutFile = new PrintWriter(fw);  
                OutPutFile.println("");  
                OutPutFile.append("\t\t Invoice"); // Writing to the file chosen  
                OutPutFile.println("");  
                OutPutFile.append("-----");  
                OutPutFile.println("");  
                OutPutFile.append("Customer Username: "+CustomerUserName);  
                OutPutFile.println("");  
                OutPutFile.append("Customer Fullname: "+CustomerFullName);  
                OutPutFile.println("");  
                OutPutFile.append("Customer Phone number: "+CustomerPhoneNumber);  
                OutPutFile.println("");  
                OutPutFile.append("Customer City: "+CustomerCity);  
                OutPutFile.println("");  
                OutPutFile.append("Customer Address: "+CustomerAddress);  
                OutPutFile.println("");  
                OutPutFile.append("-----");  
                OutPutFile.println("");  
                if(CustomerPanel.orders[0] != null){ // If there's no order in Index 0 then don't write anything
```



```
        }

        RandomDay = Day.nextInt( origin: 1, bound: 4); // Making a random number between 1-4
        RandomTime = Time.nextInt( origin: 1, bound: 12); // Making a random number between 1-12
        OutPutFile.append("-----\n"); // proceed writing to the file
        OutPutFile.append("Estimated delivery time: "+String.valueOf(RandomDay)+" Days "
        +"at "+String.valueOf(RandomTime)+"PM\n");
        OutPutFile.append("Tax is: %15\n"+ "added cost after tax: $" +(int)PriceAddedAfterTax+'\n');
        OutPutFile.append("The total price is: $" +(int)TotalAfterTax+'\n');
        OutPutFile.append("-----\n");
        OutPutFile.close(); // Closing the file.
        System.exit( status: 0); // Closing the application after saving the bill to
    }
    catch (IOException ex) // Catch the Exception if found and do what is needed.
    {
        Logger.getLogger(Invoice.class.getName()).
            log(Level.SEVERE, msg: null, ex);
    }
}
}

});

// Ending of the ActionListener
bottom_Cancelorder_FileChooser_panel.add(CancelOrderbtn); // Adding the cancel button to the panel
bottom_Cancelorder_FileChooser_panel.add(FileChooserButton); // Adding the FileChooser button to the panel
add(bottom_Cancelorder_FileChooser_panel,BorderLayout.SOUTH); // Adding the panel to the JFrame
}

1 usage
```



```
usage
public void buildOrderSummaryArea(){                                // buildOrderSummaryArea method
    int Counter = 0;
    InvoiceArea2.append("-----\n");
    while(CustomerPanel.orders.length > Counter){

        if (CustomerPanel.orders[Counter] == null){                  // If Index 0 is empty break the loop.
            break;
        }
        InvoiceArea2.append(String.valueOf(CustomerPanel.orders[Counter++]) + '\n');           // if not then add the order to the JTextArea
        InvoiceArea2.append("-----\n");

        TotalAfterTax = Order.total+(Order.total*tax);             // Calculations..
        RandomDay = Day.nextInt( origin: 1, bound: 4);
        RandomTime = Time.nextInt( origin: 1, bound: 12);
        InvoiceArea2.append("Estimated delivery time: "+String.valueOf(RandomDay)+" Days "
        +"at "+String.valueOf(RandomTime)+"PM\n");                // Adding the delivery time to the JTextArea
        InvoiceArea2.append("Tax is: %15\n"+ "added cost after tax: $" +(int)PriceAddedAfterTax+'\n');
        InvoiceArea2.append("The total price is: $" +(int)TotalAfterTax);          // Adding the price to the JTextArea
        /////////////////////////////////
        setVisible(true);                                         // Setting the visibility true
        ///////////////////////////////
    }
    ///////////////////////////////
```



```
1 usage
public void buildRightPanel(){                                // buildRightPanel method
    rightPanel.setLayout(new GridLayout( rows: 6 , cols: 1));           // Setting the layout to Grid "6 Rows & 1 Column"
    userBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray)); // Adding border
    userBtn.setToolTipText("Click here to edit your information!");          // Hint fot the button
    userBtn.addActionListener(new ActionListener() {           // Adding ActionListener to the button
////////////////////////////////////////////////////////////////// Beginning of the ActionListener
        public
        void actionPerformed(ActionEvent e) {
            setVisible(false);
            NewCs o = new NewCs();
            o.setTitle("Edit information");
        }
    });
////////////////////////////////////////////////////////////////// Ending of the ActionListener
    rightPanel.add(userBtn);                                // Adding the button to the panel
    for(int i=0;i<4;i++){
        rightPanel.add(new JPanel());                      // Adding 4 more Panels to the current Panel
    }
    helpBtn.setBorder(BorderFactory.createBevelBorder( type: 1, Color.lightGray, Color.lightGray)); // Adding border
    helpBtn.setToolTipText("what is this page?");           // Hint for the button
    helpBtn.addActionListener(new ActionListener() {           // Adding ActionListener to the button
////////////////////////////////////////////////////////////////// Beginning of the ActionListener
        public
        void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog( parentComponent: null, message: "In this page you are checking out the order");
        }
    });
////////////////////////////////////////////////////////////////// Ending of the ActionListener
    rightPanel.add(helpBtn);                                // Adding the help button to the panel.
```



*Thank
you!*

