

# POINT OF SALES SYSTEM

F R A P P E   S T O R E

Object Oriented  
Programming

IT 201

## MEMBERS:

**Baria, Brittany Haley**  
**Evans, Elaisa Nicole**  
**Jimenez, Luis Victor**  
**Ramirez, Leoj**  
**Valdez, Daniel**

# GANNT CHART

# POWERPOINT

## SLIDES

# POINT OF SALES SYSTEM

## FRAPPE STORE

Object Oriented Programming  
IT 201

Canva

IT 201

>>>

## Members Profile



**Brittany Haley  
Baria**



**Elaisa Nicole Evans**



**Luis Victor Jimenez**



**Leoj Ramirez**



**Daniel Valdez**



## Introduction

A point of sale (POS) system is a software and hardware solution used in businesses to process sales transactions, record important data like items, quantities, prices, and payment methods. It aids in inventory management, financial reporting, and customer relationship management.

## What is the importance of the system that you created?

01

### Efficient transactions

POS systems streamline payment processing, reducing wait times and ensuring accurate calculations and receipts.

02

### Sales analysis

POS systems generate reports on sales patterns, popular items, and customer preferences, aiding decision-making and marketing strategies.

03

### Inventory management

Real-time tracking of stock levels helps prevent shortages and optimize inventory management.

04

### Improved customer service

POS systems provide customer information, enabling personalized interactions, loyalty programs, and targeted marketing.

05

### Security and Stability

It is important in POS system to secure the sensitive customer and transaction data with its built-in features for secure coding, encryption and access control.

## What is the importance of the system that you created?

05

### Streamlined accounting

POS systems automate sales and financial data recording, reducing errors and facilitating better financial management.

06

### Integration capabilities

POS systems can integrate with other software, enhancing operational efficiency and data flow.

07

### Scalability and flexibility

POS systems adapt to business growth, accommodating more products, multiple locations, and diverse payment methods.

## How does it differ from the unique or existing?

### Proof of purchase and documentation

Proof of purchase and documentation are vital in school canteens especially in Adamson University where there is lack in utilization of this system. POS is also crucial for accountability, food safety compliance, allergen management, efficient operations, audits, supplier validation, and dispute resolution.

### Efficient Order processing and Payment

POS system program implemented in school's canteen can streamline the flow of orders and payments that leads to increased the efficiency and decreased the waiting times for students. These capabilities enhance the transaction process, resulting in an improved dining experience.

### Integration and Scalability

It can enable an efficient data sharing, synchronization, and it can enhance operational efficiency that reduces manual data entry. This is designed to scale and adapt to the changing needs of the school's canteen, it can handle the growing of transaction volumes and it can adjust to future changes and expansions.

## Why do you choose this topic?



01

### To promote efficient customer service

Efficient customer service in food stores is important for customer satisfaction, speed and convenience, customer retention, positive word-of-mouth, reputation management, and brand loyalty.

02

### To secure customers safety

Proof of purchase and documentation in food stores are important for ensuring quality, compliance, customer confidence, recalls, supplier accountability, inventory management, and financial and tax compliance.

03

### To understand the functions of the program

POS system program, it enables the developers to explore more distinctive features, and it creates integrations that will address a specific business needs by offering a competitive edge through innovative problem-solving.

# DEMO PROGRAM SIMULATION

ENHANCE CUSTOMER  
SERVICE

Adamson SV Canteen - Frap...

**Sign In**

BSIT FRAPPE

USERNAME

PASSWORD

**LOGIN**

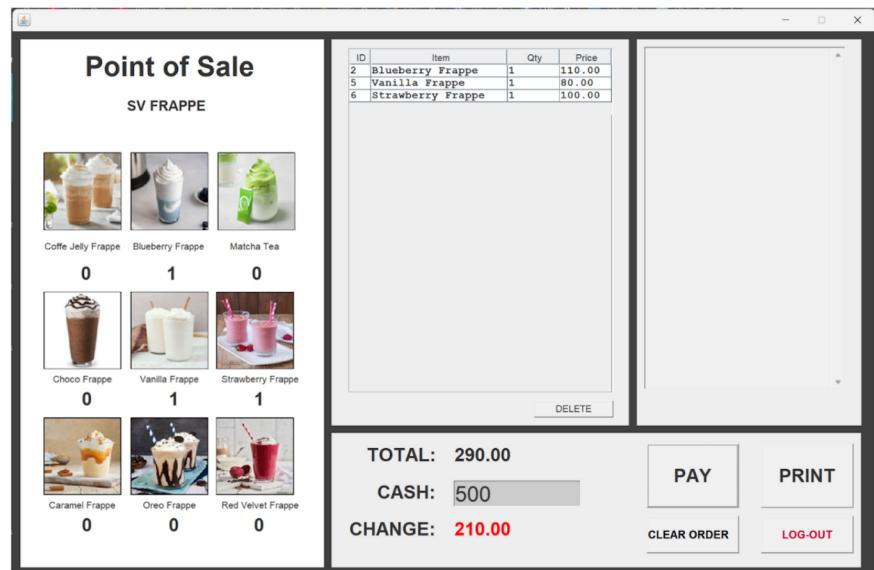
CONTACT ITC

# DEMO

## PROGRAM

### SIMULATION

ENHANCE CUSTOMER  
SERVICE



Point of Sales System

# THANK YOU!

# CODE SNIPPETS

# LOG IN FORM

## Importing Modules

```
1 package FinalProjectFile;
2 [-] import java.awt.Color;
3 import javax.swing.JFrame;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import javax.swing.JOptionPane;
```

## Public Class of Log In Form using JFrame with Customization

```
11 public class PointOfSales extends javax.swing.JFrame {
12
13 [-] public PointOfSales() {
14     this.setTitle( title:" Adamson SV Canteen - Frappe");
15     this.setDefaultCloseOperation( operation:JFrame.EXIT_ON_CLOSE);
16     this.getContentPane().setBackground( c:Color.WHITE);
17     this.setLocationRelativeTo( c:null); //Center Form
18 }
```

## Initializing Components and Adding Authentication Code

```
19 [-] initComponents();
20 PasswordField.setEchoChar( c:'*');
21 [-] LoginBtn.addActionListener( new ActionListener() {
22     public void actionPerformed(ActionEvent e){
23         String username = UsernameField.getText();
24         String password = PasswordField.getText();
25         if(username.equals( anObject:"Adamson") && password.equals( anObject:"Adul123")){
26             dispose();
27             AdminPortal adminportal = new AdminPortal();
28             adminportal.setVisible( b:true);
29         } else if(username.equals( anObject:"") || password.equals( anObject:"")){
30             JOptionPane.showMessageDialog( parentComponent:PointOfSales.this,
31                                         message:"Complete the Field, Please Try Again");
32             UsernameField.setText( t:"");
33             PasswordField.setText( t:"");
34         } else {
35             JOptionPane.showMessageDialog( parentComponent:PointOfSales.this,
36                                         message:"Incorrect Username or Password, Please Try Again");
37             UsernameField.setText( t:"");
38             PasswordField.setText( t:"");
39         }
40     });
41 });
42 }
```

# LOG IN FORM

This part of code, instructs the compiler to suppress the warning related to unchecked operations for the specific code element it is applied to, such as a method, a class, or a variable. It tells the compiler to ignore the unchecked warning for that particular code element and not issue it.

```
@SuppressWarnings("unchecked")  
Generated Code
```

Dialog Box for ITC Contact Button

```
188 | private void ITCLabelMouseClicked(java.awt.event.MouseEvent evt) {  
189 |     JOptionPane.showMessageDialog(parentComponent: this, "ITC Email: itc@adamson.edu.ph"  
190 |     + "\nITC Contact #: +63 123 5633");
```

Main Method

```
193 | public static void main(String args[]) {  
194 |     Look and feel setting code (optional)  
195 |     //</editor-fold>  
196 |  
197 |     /* Create and display the form */  
198 |     java.awt.EventQueue.invokeLater(new Runnable() {  
199 |         @  
200 |         public void run() {  
201 |             new PointOfSales().setVisible( b: true);  
202 |         }  
203 |     });  
204 | }
```

# LOG IN FORM

## Variables Declaration (GUI)

```
225     // Variables declaration - do not modify
226     private java.awt.Label ITCLabel;
227     private java.awt.Button LoginBtn;
228     private java.awt.TextField PasswordField;
229     private java.awt.TextField UsernameField;
230     private java.awt.Label label2;
231     private java.awt.Label label3;
232     private java.awt.Label label4;
233     private java.awt.Label label5;
234     private java.awt.Panel panel1;
235     // End of variables declaration
236 }
```

# POS GUI FORM

## Importing Modules

```
1 package FinalProjectFile;
2 import javax.swing.table.DefaultTableModel;
3 import java.util.Vector;
4 import java.text.DecimalFormat;
5 import java.awt.*;
6 import javax.swing.JOptionPane;
7 import java.util.Random;
```

It Inherits the Basic Window Functionality and can Utilize it Directly

```
10 public class AdminPortal extends javax.swing.JFrame
11 {
```

Initializing Components and Customizing the Width of the Item and ID Column

```
13     public AdminPortal()
14     {
15         initComponents();
16
17         getContentPane().setBackground( c.Color.DARK_GRAY );
18         jTable2.getColumnModel().getColumn( columnIndex: 0 ).setPreferredWidth( preferredWidth: 30 );
19         jTable2.getColumnModel().getColumn( columnIndex: 1 ).setPreferredWidth( preferredWidth: 200 );
20     }
```

# POS GUI FORM

## Order Table Code for Adding Items from User Interactions

```
22     public void addTable(int id, String name, int qty, Double price)
23     {
24
25         DefaultTableModel dt = (DefaultTableModel) jTable2.getModel();
26
27         DecimalFormat df = new DecimalFormat(pattern: "00.00");
28         double Total = price * Double.valueOf(d:qty);
29         String TotalPrice = df.format(number:Total);
30
31         for(int row = 0; row < jTable2.getRowCount(); row++)
32         {
33             if(name == jTable2.getValueAt(row, column:1))
34             {
35                 dt.removeRow(row:jTable2.convertRowIndexToModel(viewRowIndex:row));
36             }
37         }
38
39         Vector v = new Vector();
40
41         v.add(e:id);
42         v.add(e:name);
43         v.add(e:qty);
44         v.add(e:TotalPrice);
45
46         dt.addRow(rowData:v);
47     }
48 }
```

## Item Calculation Method for Price Based on Quantity

```
50     public void cal()
51     {
52         int rowNum = jTable2.getRowCount();
53         double total = 0.0;
54
55         for(int i = 0; i < rowNum; i++)
56         {
57             double value = Double.valueOf(s:jTable2.getValueAt(row:i, column:3).toString());
58             total += value;
59         }
60
61         DecimalFormat df = new DecimalFormat(pattern: "00.00");
62         TotalLabel.setText(text:df.format(number:total));
63     }
}
```

# POS GUI FORM

JButton Code for Frappe Products that will be Added to the Parameters of Order Table(JTable2)

```
691  private void CJFActionPerformed(java.awt.event.ActionEvent evt) {  
692      int q = Integer.valueOf(s:CJLabel.getText());  
693      q++;  
694      CJLabel.setText( text:String.valueOf( i:q ));  
695  
696      addTable( id:1, name:"CJ Frappe\t", qty:q, price: 90.000);  
697      cal();  
698  }  
699  
700  private void BlueberryActionPerformed(java.awt.event.ActionEvent evt) {  
701      int q = Integer.valueOf(s:BlueLabel.getText());  
702      q++;  
703      BlueLabel.setText( text:String.valueOf( i:q ));  
704  
705      addTable( id:2, name:"Blueberry Frappe ", qty:q, price: 110.00 );  
706      cal();  
707  }  
708  
709  private void MatchaActionPerformed(java.awt.event.ActionEvent evt) {  
710      int q = Integer.valueOf(s:MatchaLabel.getText());  
711      q++;  
712      MatchaLabel.setText( text:String.valueOf( i:q ));  
713  
714      addTable( id:3, name:"Matcha Tea           ", qty:q, price: 100.00 );  
715      cal();  
716  }  
717  
718  private void VanillaFrappeActionPerformed(java.awt.event.ActionEvent evt) {  
719      int q = Integer.valueOf(s:VanillaLabel.getText());  
720      q++;  
721      VanillaLabel.setText( text:String.valueOf( i:q ));  
722  
723      addTable( id:5, name:"Vanilla Frappe     ", qty:q, price: 80.000);  
724      cal();  
725  }  
726  
727  private void ChocoFrappeActionPerformed(java.awt.event.ActionEvent evt) {  
728      int q = Integer.valueOf(s:ChocoLabel.getText());  
729      q++;  
730      ChocoLabel.setText( text:String.valueOf( i:q ));  
731  
732      addTable( id:4, name:"Chocolate Frappe ", qty:q, price: 100.00 );  
733      cal();  
734  }  
735  
736  private void StrawberryActionPerformed(java.awt.event.ActionEvent evt) {  
737      int q = Integer.valueOf(s:StrawberryLabel.getText());  
738      q++;  
739      StrawberryLabel.setText( text:String.valueOf( i:q ));  
740  
741      addTable( id:6, name:"Strawberry Frappe ", qty:q, price: 100.00 );  
742      cal();  
743  }  
744  
745  
```

# POS GUI FORM

## Continuation

```
751 [-] private void CaramelActionPerformed(java.awt.event.ActionEvent evt) {  
752     int q = Integer.valueOf( s:CaramelLabel.getText());  
753     q++;  
754     CaramelLabel.setText( text:String.valueOf( i:q));  
755  
756  
757     addTable( id:7, name:"Caramel Frappe", qty:q, price:120.00);  
758     cal();  
759 }  
760  
761 [-] private void OreoActionPerformed(java.awt.event.ActionEvent evt) {  
762     int q = Integer.valueOf( s:OreoLabel.getText());  
763     q++;  
764     OreoLabel.setText( text:String.valueOf( i:q));  
765  
766  
767     addTable( id:8, name:"Oreo Frappe", qty:q, price:100.00);  
768     cal();  
769 }  
770  
771 [-] private void RedVelvetActionPerformed(java.awt.event.ActionEvent evt) {  
772     int q = Integer.valueOf( s:RedLabel.getText());  
773     q++;  
774     RedLabel.setText( text:String.valueOf( i:q));  
775  
776  
777     addTable( id:9, name:"Red Velvet Frappe", qty:q, price:140.00);  
778     cal();  
779 }
```

## Delete Button Function to Delete Items in Order Table

```
781 [-] private void DeleteBtnActionPerformed(java.awt.event.ActionEvent evt) {  
782     try  
783     {  
784         DefaultTableModel dt = (DefaultTableModel) jTable2.getModel();  
785         String r = dt.getValueAt( row:jTable2.getSelectedRow(), column:0).toString();  
786  
787         //Product Removal (It will Select the Product to be Removed)  
788         int row = jTable2.getSelectedRow();  
789         dt.removeRow(row);  
790     }
```

# POS GUI FORM

Delete Method - Resetting Quantity Label when the Item is Deleted from the Table

```
792     //Reset Quantity
793     switch(r)
794     {
795         case "1":
796             CJFLabel.setText( text:"0");
797             break;
798         case "2":
799             BlueLabel.setText( text:"0");
800             break;
801         case "3":
802             MatchaLabel.setText( text:"0");
803             break;
804         case "4":
805             ChocoLabel.setText( text:"0");
806             break;
807         case "5":
808             VanillaLabel.setText( text:"0");
809             break;
810         case "6":
811             StrawberryLabel.setText( text:"0");
812             break;
813         case "7":
814             CaramelLabel.setText( text:"0");
815             break;
816         case "8":
817             OreoLabel.setText( text:"0");
818             break;
819         case "9":
820             RedLabel.setText( text:"0");
821             break;
822         default:
823             System.out.println( x:"0");
824     }
```

# POS GUI FORM

## Delete Method - Updating the Total Price Label Based on Existing Items in the Order Table

```
826     int rowNum = jTable2.getRowCount();
827     double total = 0.0;
828
829     for(int i = 0; i < rowNum; i++)
830     {
831         double value = Double.valueOf(s:jTable2.getValueAt( row:i, column:3).toString());
832         total += value;
833     }
834
835     DecimalFormat df = new DecimalFormat(pattern:"00.00");
836     TotalLabel.setText(text:df.format(number:total));
837 }
```

## Exception in Case the Admin is Clicking the Delete Button without Selected Items (Avoiding System Crash)

```
838     catch(Exception e)
839     {
840         JOptionPane.showMessageDialog(parentComponent: this, message: "Select an Item to Delete");
841     }
842 }
```

## Payment Button Function with Exception (Numeric Values Only)

```
844     private void PayBtnActionPerformed(java.awt.event.ActionEvent evt) {
845
846         try
847         {
848             double total = Double.valueOf(s:TotalLabel.getText());
849             double cash = Double.valueOf(s:CashTextField.getText());
850             double change = cash - total;
851
852             DecimalFormat df = new DecimalFormat(pattern:"00.00");
853             ChangeLabel.setForeground(c:Color.red);
854             ChangeLabel.setText(text:String.valueOf(obj:df.format(number:change)));
855
856         }
857         catch (Exception e)
858         {
859             JOptionPane.showMessageDialog(parentComponent: this, message: "Add Value/Numeric Input Only");
860         }
861     }
```

# POS GUI FORM

## Printing Receipt Based on Total Price, Cash Given, Change, and the Orders on the Order Table

```
private void PrintBtnActionPerformed(java.awt.event.ActionEvent evt) {
try
{
    Bill.setText(t:"\\n\\t      SV Frappe Store \\n");
    Bill.setText(Bill.getText() + "\\tSan Marcelino, Ermita Manila \\n");
    Bill.setText(Bill.getText() + "\\t      +63 1234 5678 901 \\n");
    Bill.setText(Bill.getText() + "\\t      SV Frappe Store \\n\\n");
    Bill.setText(Bill.getText() + " *****\\n\\n");
    Bill.setText(Bill.getText() + "     ITEM          QTY          PRICE \\n\\n");

    //Getting the Products from Our Order Table
    DefaultTableModel df = (DefaultTableModel) jTable2.getModel();
    for(int i = 0; i < jTable2.getRowCount(); i++)
    {
        String Name = df.getValueAt( row:i, column:1).toString();
        String Qty = df.getValueAt( row:i, column:2).toString();
        String Price = df.getValueAt( row:i, column:3).toString();

        Bill.setText(Bill.getText() + "   " + Name + "\\t" + Qty + "\\t" + Price + "\\n");
    }

    Random r = new Random();           Random Number Generator as Order Number
    int random = r.nextInt();          Reference

    Bill.setText(Bill.getText() + "\\n\\n");
    Bill.setText(Bill.getText() + " *****\\n\\n");
    Bill.setText(Bill.getText() + " Total : P" + TotalLabel.getText() + "\\n");
    Bill.setText(Bill.getText() + " Cash : P" + CashTextFieldId.getText() + "\\n");
    Bill.setText(Bill.getText() + " Change: P" + ChangeLabel.getText() + "\\n\\n\\n");
    Bill.setText(Bill.getText() + " Thank you! Come again ^=^ \\n\\n");
    Bill.setText(Bill.getText() + " *****\\n\\n");
    Bill.setText(Bill.getText() + " REFERENCE #: " + random);

    //Clear Table Content & Computations
    DefaultTableModel dt = (DefaultTableModel) jTable2.getModel();
    dt.setRowCount( rowCount:0 );

    TotalLabel.setText( text:String.valueOf( i:0 ) );
    CashTextField.setText( t:String.valueOf( obj: "" ) );
    ChangeLabel.setForeground( c:Color.BLACK );
    ChangeLabel.setText( text:String.valueOf( i:0 ) );

    CJLabel.setText( text:"0" );
    BlueLabel.setText( text:"0" );
    MatchaLabel.setText( text:"0" );
    ChocoLabel.setText( text:"0" );
    VanillaLabel.setText( text:"0" );
    StrawberryLabel.setText( text:"0" );
    CaramelLabel.setText( text:"0" );
    OreoLabel.setText( text:"0" );
    RedLabel.setText( text:"0" );
}
}
```

# POS GUI FORM

Exception in case the Program will Crash

```
915     catch (Exception e)
916     {
917         JOptionPane.showMessageDialog( parentComponent: this, "System Error, "
918             + "Try Again or Contact ITC in Log In Form");
919     }
```

Log Out Button, Going Back to Log In Form

```
923     private void BackBtnActionPerformed(java.awt.event.ActionEvent evt) {
924         dispose();
925         PointOfSales POS = new PointOfSales();
926         POS.setVisible( b: true );
927     }
```

Clear Button/Reset Order Session

```
929     private void ClearBtnActionPerformed(java.awt.event.ActionEvent evt) {
930         DefaultTableModel dt = (DefaultTableModel) jTable2.getModel();
931         dt.setRowCount( rowCount: 0 );
932
933         TotalLabel.setText( text: String.valueOf( i: 0 ) );
934         CashTextField.setText( t: String.valueOf( obj: "" ) );
935         ChangeLabel.setForeground( c: Color.BLACK );
936         ChangeLabel.setText( text: String.valueOf( i: 0 ) );
937
938         CJFLabel.setText( text: "0" );
939         BlueLabel.setText( text: "0" );
940         MatchaLabel.setText( text: "0" );
941         ChocoLabel.setText( text: "0" );
942         VanillaLabel.setText( text: "0" );
943         StrawberryLabel.setText( text: "0" );
944         CaramelLabel.setText( text: "0" );
945         OreoLabel.setText( text: "0" );
946         RedLabel.setText( text: "0" );
947
948         Bill.setText( t: "" );
949     }
```

# POS GUI FORM

## Main Method

```
951  public static void main(String args[]) {  
952      ...  
953      ...  
954      ...  
955      java.awt.EventQueue.invokeLater(new Runnable() {  
956          public void run() {  
957              new AdminPortal().setVisible(true);  
958          }  
959      });  
960  }
```

## Variables Declaration

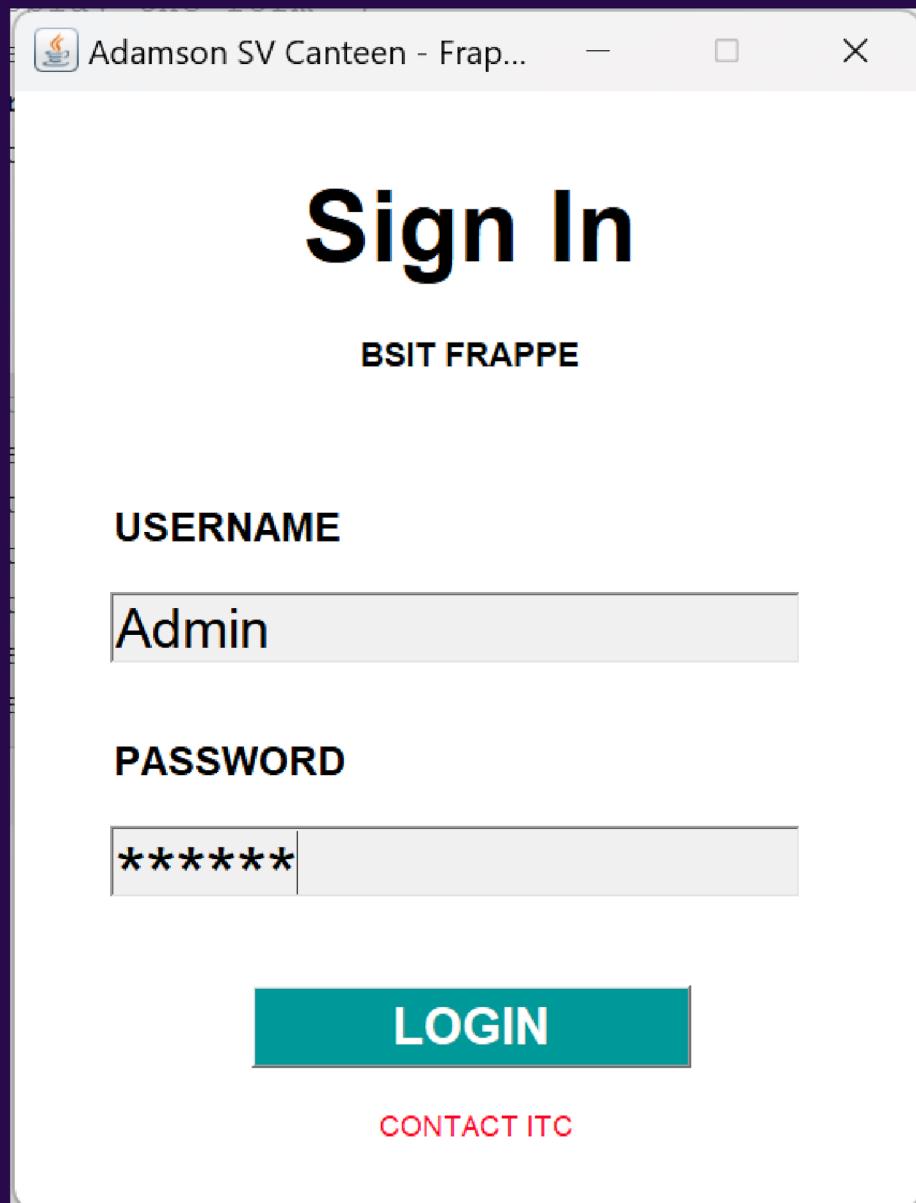
```
960  // Variables declaration - do not modify  
961  private java.awt.Button BackBtn;  
962  private java.awt.TextArea Bill;  
963  private java.awt.Label BlueLabel;  
964  private javax.swing.JButton Blueberry;  
965  private javax.swing.JButton CJF;  
966  private java.awt.Label CJFLabel;  
967  private javax.swing.JButton Caramel;  
968  private java.awt.Label CaramelLabel;  
969  private java.awt.TextField CashTextField;  
970  private java.awt.Label ChangeLabel;  
971  private javax.swing.JButton ChocoFrappe;  
972  private java.awt.Label ChocoLabel;  
973  private java.awt.Button ClearBtn;  
974  private java.awt.Button DeleteBtn;  
975  private javax.swing.JButton Matcha;  
976  private java.awt.Label MatchaLabel;  
977  private javax.swing.JButton Oreo;  
978  private java.awt.Label OreoLabel;  
979  private java.awt.Button PayBtn;  
980  private java.awt.Button PrintBtn;  
981  private java.awt.Label RedLabel;  
982  private javax.swing.JButton RedVelvet;  
983  private javax.swing.JButton Strawberry;  
984  private java.awt.Label StrawberryLabel;  
985  private java.awt.Label TotalLabel;  
986  private javax.swing.JButton VanillaFrappe;  
987  private java.awt.Label VanillaLabel;  
988  private javax.swing.JScrollPane jScrollPane1;  
989  private javax.swing.JScrollPane jScrollPane2;
```

# POS GUI FORM

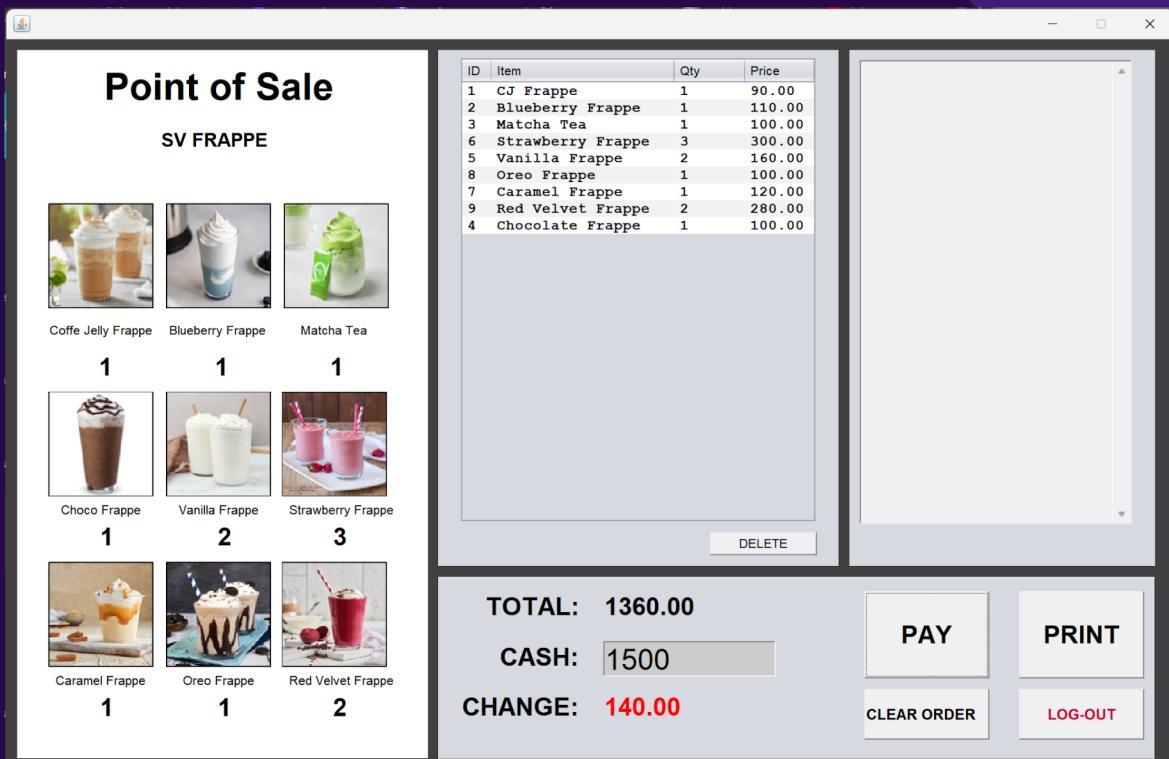
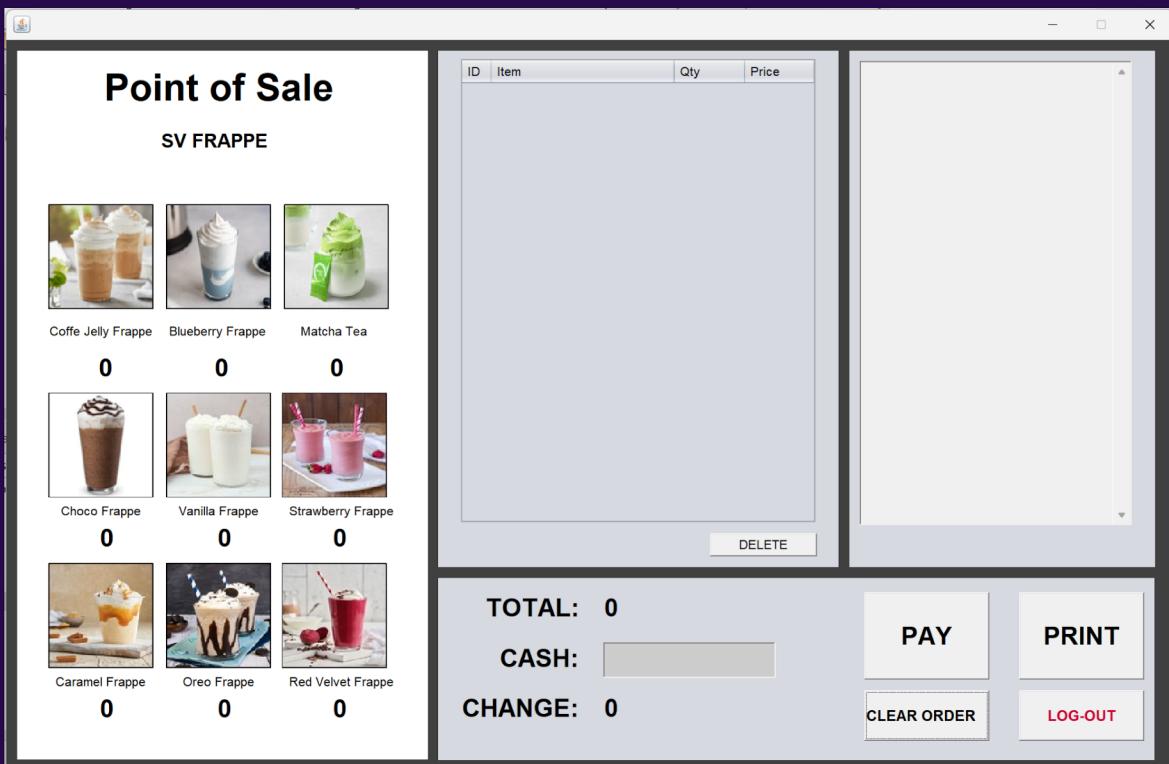
## Variable s Declaration Continuation

```
991     private javax.swing.JTable jTable2;
992     private java.awt.Label label1;
993     private java.awt.Label label10;
994     private java.awt.Label label11;
995     private java.awt.Label label12;
996     private java.awt.Label label13;
997     private java.awt.Label label14;
998     private java.awt.Label label15;
999     private java.awt.Label label16;
1000    private java.awt.Label label2;
1001    private java.awt.Label label3;
1002    private java.awt.Label label4;
1003    private java.awt.Label label5;
1004    private java.awt.Label label6;
1005    private java.awt.Label label7;
1006    private java.awt.Label label8;
1007    private java.awt.Label label9;
1008    private java.awt.Panel panel1;
1009    private java.awt.Panel panel2;
1010    private java.awt.Panel panel3;
1011    private java.awt.Panel panel4;
1012    private java.awt.TextArea textArea1;
1013    // End of variables declaration
1014 }
1015 }
```

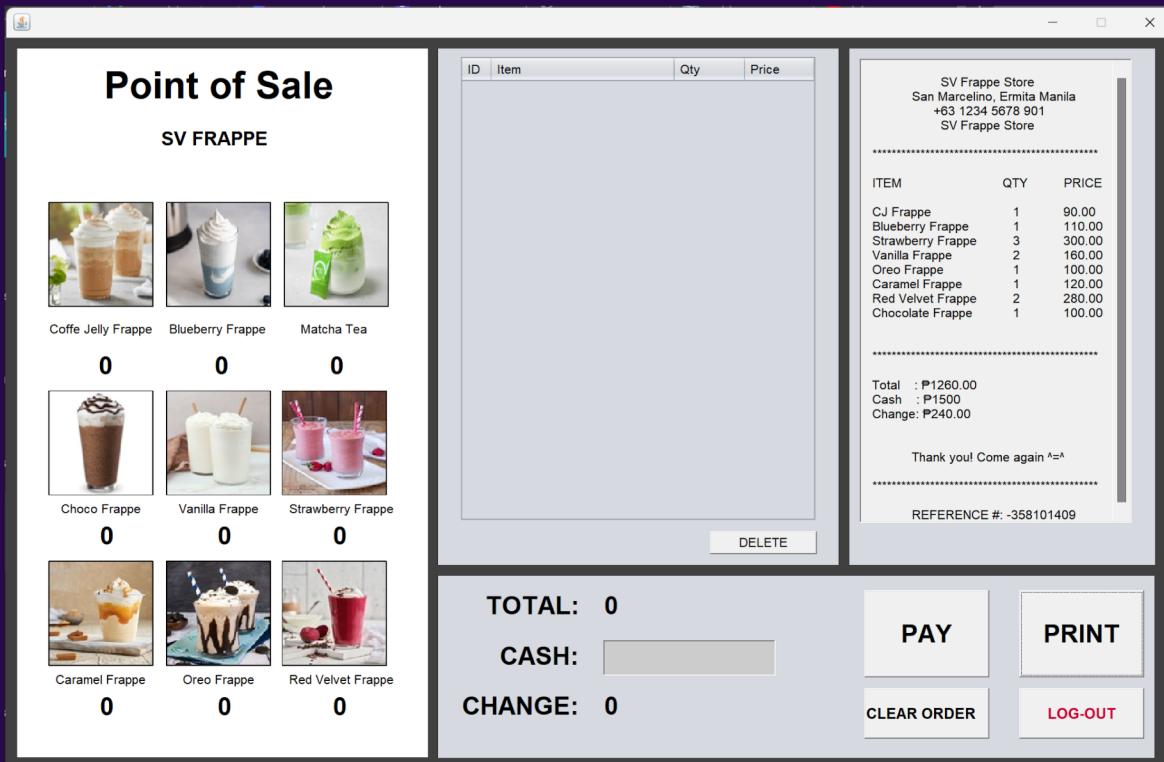
# LOG IN INTERFACE



# POS INTERFACE



# POS INTERFACE





# OBJECT ORIENTED PROGRAMMING

# JAVA

## PROJECT DETAILS

**Submitted To:** Ma'am Felnita Tan

**Submitted By:** BSIT 201 (POS Project)

**Project:** Basic Point of Sales Based in Adamson  
University Canteen Stores

**Date Presented:** May 19, 2023