JAVA PROJECT REPORT

(Project Term January-May 2023)

(*Food Service Payment Processing Software*)

Submitted by

(Irla.Likhitha Sivadurga) **Registration Number : 12103228**

Course Code: CSE310

Under the Guidance of

(Dr. A. Ranjith Kumar sir)

School of Computer Science and Engineering

TABLE OF CONTENTS

1. INTRODUCTION

A Food Service Payment Processing Software project using Java Swing J Frames would involve creating a graphical user interface (GUI) for the billing system using Java Swing components such as J Frame, J Label, J Text Field, J Button, and J Table.

The project would involve designing a user-friendly interface for restaurant staff to use, including features such as menu management, order placement, order tracking, and payment processing. The application would need to be able to handle multiple orders simultaneously, manage inventory, and generate reports for business analysis.

To develop this project, Java programming language would be used along with the Java Swing library for creating the user interface. The project would also require knowledge of database management and design principles to store and retrieve data related to orders, inventory, and payments.

The GUI would consist of multiple J Frames, each containing different functionalities such as placing an order, viewing the menu, processing payments, and generating reports. The components such as J Labels, J Text Fields, J Buttons, and J Tables would be added to these J Frames to create the required user interface.

The Food Service Payment Processing Software project using J Frames would require a good understanding of Java programming concepts, Swing components, and event handling. The project would also require knowledge of database design, data structures, and algorithms to ensure that the application runs efficiently and is easy to maintain.

.

1.1 Graphical user interface in java-j-frames*:*

GUI stands for Graphical User Interface, and it refers to the visual elements that allow users to interact with a software application. A GUI is typically composed of graphical components such as buttons, text fields, labels, menus, and windows, which enable users to perform different actions within the application.

Java Swing is a popular GUI toolkit for building graphical user interfaces in Java. J Frames are one of the core components of the Java Swing library and provide a container for GUI components to be added. A J Frame is a top-level window with a title bar, menu bar, and border, which can contain other GUI components such as buttons, labels, text fields, and tables.

To create a J Frame in Java, you can use the J Frame class from the javax.swing package. You can add various components to a JFrame using methods such as add() or setContentPane(), and set their layout using methods such as setLayout() or setPreferredSize(). You can also set properties such as size, title, location, and visibility of a J Frame using various methods provided by the J Frame class.

2. PROPOSED TECHNIQUE:

To implement billing according to food items in Food Service Payment Processing Software using JFrames, these modules were used:

2.1 Module 1*:*

1. Create a menu for the restaurant: Create a menu that includes all the food items offered by the restaurant along with their prices. Store this information in a data structure such as a database table or a collection.

2.2 Module 2*:*

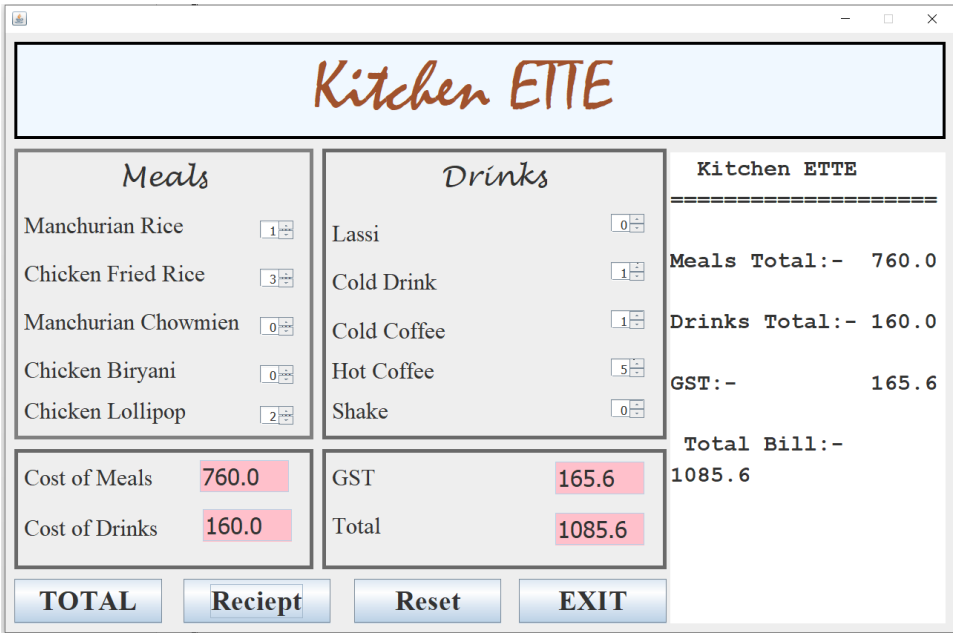
2. Add the food items to the billing interface: Create a JFrame that will serve as the billing interface. Add JLabels and JTextFields for displaying the food items and their prices. You can also use a JTable to display the menu items and their prices.

2.3 Module 3*:*

3. Implement an ActionListener for the food items: Add an ActionListener to each food item that will be added to the bill. When a food item is selected, retrieve its price from the menu and display it in the corresponding JTextField or cell of the JTable.

2.4 Module 4*:*

4. Calculate the total amount: Add a button to the billing interface that will calculate the total amount of the bill. When this button is clicked, retrieve the prices of all the selected food items and add them up to get the total bill amount. Display the total amount in a separate JTextField or JLabel.



3. SAMPLE CODE:

```
import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JPasswordField;
```

```
import javax.swing.JButton;
import javax.swing.JSeparator;
import java.awt.Color;
import javax.swing.border.EtchedBorder;
import javax.swing.JCheckBox;
import java.awt.event.ActionListener;
import java.sql.Connection;
Import java.sql.ResultSet;
import java.sql.Statement;
import java.awt.event.ActionEvent;
import deod.ConnectionProvider1;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
public Login() {
        try {
            cn=ConnectionProvider1.getConnection();
            st=cn.createStatement();

        }catch(Exception e) {
            JOptionPane.showMessageDialog(null,e);
        }

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 769, 546);
        setResizable(false);
        contentPane = new JPanel();
```

```
                contentPane.setBackground(new Color(250, 128, 114));
                contentPane.setBorder(new EtchedBorder(EtchedBorder.LOWERED,
Color.BLACK, null));
                setContentPane(contentPane);
                contentPane.setLayout(null);

                JLabel lblNewLabel = new JLabel("Employee ID");
                lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 30));
                lblNewLabel.setBounds(155, 136, 169, 56);
                contentPane.add(lblNewLabel);

                textField = new JTextField();
                textField.setForeground(Color.BLACK);
                textField.setFont(new Font("Times New Roman", Font.PLAIN, 30));
                textField.setBounds(396, 143, 183, 44);
                contentPane.add(textField);
                textField.setColumns(10);

                JLabel lblNewLabel_1 = new JLabel("Password");
                lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 30));
                lblNewLabel_1.setBounds(155, 234, 156, 35);
                 contentPane.add(lblNewLabel_1); passwordField = new
JPasswordField();
                passwordField.setEchoChar('*');
                passwordField.setFont(new Font("Tahoma", Font.PLAIN, 15));
                passwordField.setBounds(396, 238, 183, 35);
                contentPane.add(passwordField);
```

```
JButton btnNewButton = new JButton("Close");
btnNewButton.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
		setVisible(false);
		System.exit(0);
	}
});
btnNewButton.setForeground(new Color(139, 0, 0));
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 30));
btnNewButton.setBounds(142, 343, 113, 44);
contentPane.add(btnNewButton);

JButton btnSignIn = new JButton("Sign In");
btnSignIn.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
	   try {
		String user=textField.getText();
			String pass=passwordField.getText();
		  String query="select * from records where EmpId =
'"+user+"'";
			ResultSet rs=st.executeQuery(query);
			if(rs.next()) {
				if(pass.contains(rs.getString(3))){
					Application.main(null);
				}
				else
```

```
            JOptionPane.showMessageDialog(null,"Wrong UserName/Password");
                                    }
                                } catch (Exception e1) {
                                    // TODO Auto-generated catch block
                                    JOptionPane.showMessageDialog(null,e1);
                                }

                            }
                        });
        JLabel lblNewLabel_2 = new JLabel("WELCOME to Kitchen ETTE");
                        lblNewLabel_2.setForeground(new Color(178, 34, 34));
                        lblNewLabel_2.setFont(new Font("Lucida Calligraphy", Font.PLAIN, 40));
                        lblNewLabel_2.setBounds(54, 47, 608, 44);
                        contentPane.add(lblNewLabel_2);
        public Application() {
                        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                        setBounds(100, 100, 1038, 684);
                        setResizable(false);
                        contentPane = new JPanel();
                        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
                        setContentPane(contentPane);
                        contentPane.setLayout(null);

                        String initial="  Kitchen ETTE  ==================== Meals : -
Manchurian Rice  60 Chic. Fried Rice 100 Manchurian Chow. 80 Chicken Lollipop 200
Chicken Biryani  120 Drinks : -     Lassi          30  Cold Drink      20  Cold Coffee     40  Hot
Coffee      20  Shakes         40          +18% GST";
```

```
JTextArea textArea = new JTextArea(initial);
textArea.setBackground(SystemColor.text);
textArea.setTabSize(6);
textArea.setFont(new Font("Monospaced", Font.BOLD, 24));
textArea.setWrapStyleWord(true);
textArea.setLineWrap(true);
textArea.setBounds(717, 129, 297, 508);
contentPane.add(textArea); ================== Meals : -
Manchurian Rice  60 Chic. Fried Rice 100 Manchurian Chow. 80 Chicken Lollipop 200
Chicken Biryani  120 Drinks : -     Lassi          30  Cold Drink      20  Cold Coffee      40  Hot
Coffee      20  Shakes          40          +18% GST");
        }
});
btnReset.setFont(new Font("Times New Roman", Font.BOLD, 30));
btnReset.setBounds(376, 589, 159, 48);
contentPane.add(btnReset);

JButton btnExit = new JButton("EXIT");
btnExit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                setVisible(false);
                System.exit(0);
        }
});
btnExit.setFont(new Font("Times New Roman", Font.BOLD, 30));
btnExit.setBounds(554, 589, 159, 48);
contentPane.add(btnExit);
```

```
        }
}# Auto detect text files and perform LF normalization
* text=auto
```

3. CONCLUSION:

In conclusion, a Food Service Payment Processing Software system is an essential tool for any organization that wants to streamline its billing processes, improve efficiency, and reduce errors. By automating billing tasks, organizations can save time and resources, increase accuracy, and enhance customer satisfaction. A well-designed billing management system can provide real-time insights into financial performance and help organizations make informed decisions about revenue generation and expense management. With the right system in place, organizations can better manage their cash flow, reduce billing-related disputes, and ultimately improve their bottom line. Overall, a billing management system is a smart investment for any organization looking to optimize its financial operations and stay competitive in today's fast-paced business environment. Overall, there are many enhancements that can be made to a billing management system to improve its functionality and provide a better user experience. By implementing these enhancements, organizations can optimize their billing processes, increase customer satisfaction, and ultimately improve their bottom line.