

ADVANCED COMPUTER PROGRAMMING

**PROJECT (MODULE-2)**

**RESTAURANT BILLING MANAGEMENT SYSTEM**

**GUI + DATABSE**

**SUBMITTED TO**

**MR. MUHAMMAD OSMAN**

**SUBMITTED BY**

**AHMAD ABU BAKR 44352**

**SHEHRYAR AHMAD 47435**

**M JUNAID RIAZ 43997**

# Restaurant Billing System - Documentation

## Introduction

The Restaurant Billing System is a Java Swing-based application designed to simplify and automate the billing process in a restaurant. This system allows users to input customer details, add items from the menu to an order, generate receipts, and save customer and order information to a database.

## Features

**GUI Design**: User-friendly interface built using Java Swing with organized panels for customer information, menu selection, and receipt generation.

**Order Management**: Add items from a predefined menu with quantities to create an order.

**Receipt Generation**: Automatically generate a detailed receipt displaying customer details, order summary, and the total bill.

**Database Integration**: Save customer and order details into a MySQL database.

**File Management**: Save the generated receipt to a text file for record-keeping.

**Validation**: Robust input validation for customer details to ensure accuracy.

## Concepts Used

**Java Swing**: For creating the graphical user interface.

**Event Handling**: To respond to user actions such as button clicks.

**JDBC**: Java Database Connectivity to interact with the MySQL database.

**File I/O**: For saving receipt information to a text file.

**OOP Principles**: Encapsulation and modular design for better code organization.

**Data Validation**: Ensuring correct input formats using regular expressions.

## Functionality

**1. Customer Information Input:** Users can enter the customer's name, phone number, age, and a unique customer ID. Input validation ensures data integrity.  
**2. Menu Selection:** A dropdown menu allows users to select items with prices. Quantity can be adjusted using a spinner.  
**3. Order Summary**: Displays the selected items with quantities and their total prices.  
**4. Receipt Generation:** Generates a receipt summarizing the order and customer details, displayed in a JTextArea.  
**5. Database Storage**: Saves customer and order information to a MySQL database.  
**6. File Saving:** Saves the generated receipt to a text file named 'receipt.txt'.  
**7. Form Clearing**: Provides an option to clear all inputs and start a new order.  
**8. Exit:** Closes the application.

## Source Code

Below is the complete source code for the Restaurant Billing System. The code demonstrates the integration of GUI design, database connectivity, and file handling in Java.

**CODE**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.io.File;**

**import java.io.FileWriter;**

**import java.io.IOException;**

**public class RestaurantBillingSystem extends JFrame {**

**private JTextField customerNameField, customerPhoneField, customerAgeField, customerIdField;**

**private JTextArea receiptArea;**

**private DefaultListModel<String> orderListModel;**

**private JComboBox<String> menuDropdown;**

**private JSpinner quantitySpinner;**

**private int totalBill = 0;**

**public RestaurantBillingSystem() {**

**// Set JFrame properties**

**setTitle("Restaurant Billing System");**

**setSize(1000, 800);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setLocationRelativeTo(null);**

**setLayout(new BorderLayout(20, 20)); // Add spacing between components**

**// Global Styles**

**Font titleFont = new Font("SansSerif", Font.BOLD, 30);**

**Font labelFont = new Font("SansSerif", Font.PLAIN, 18);**

**Font buttonFont = new Font("SansSerif", Font.BOLD, 16);**

**Color bgColor = new Color(245, 245, 245); // Light gray background**

**Color accentColor = new Color(30, 144, 255); // Dodger blue for highlights**

**Color buttonColor = new Color(0, 123, 255); // Bright blue for buttons**

**Color buttonTextColor = Color.BLACK; // White text for the button**

**getContentPane().setBackground(bgColor);**

**// Title Panel**

**JPanel titlePanel = new JPanel();**

**titlePanel.setBackground(bgColor);**

**JLabel titleLabel = new JLabel("Restaurant Billing System");**

**titleLabel.setFont(titleFont);**

**titleLabel.setForeground(accentColor);**

**titleLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**titlePanel.add(titleLabel);**

**add(titlePanel, BorderLayout.NORTH);**

**// Customer Information Panel**

**JPanel customerPanel = new JPanel(new GridLayout(4, 2, 15, 15)); // Increased spacing**

**customerPanel.setBackground(Color.WHITE);**

**customerPanel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(accentColor, 2), "Customer Information", 0, 0, labelFont, accentColor));**

**customerNameField = createStyledTextField();**

**customerPhoneField = createStyledTextField();**

**customerAgeField = createStyledTextField();**

**customerIdField = createStyledTextField();**

**customerPanel.add(createStyledLabel("Customer Name:"));**

**customerPanel.add(customerNameField);**

**customerPanel.add(createStyledLabel("Phone Number:"));**

**customerPanel.add(customerPhoneField);**

**customerPanel.add(createStyledLabel("Age:"));**

**customerPanel.add(customerAgeField);**

**customerPanel.add(createStyledLabel("Customer ID:"));**

**customerPanel.add(customerIdField);**

**// Menu and Order Panel**

**JPanel menuPanel = new JPanel(new BorderLayout(20, 20)); // Add spacing between sections**

**menuPanel.setBackground(Color.WHITE);**

**menuPanel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(accentColor, 2), "Menu and Orders", 0, 0, labelFont, accentColor));**

**// Menu Selection Panel**

**JPanel menuSelectionPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 15, 15)); // Increased spacing**

**menuSelectionPanel.setBackground(Color.WHITE);**

**String[] menuItems = {**

**"Burger - PKR 300", "Pizza - PKR 1200", "Cold Drink - PKR 50",**

**"Fried Chicken - PKR 500", "French Fries - PKR 100",**

**"Juices - PKR 150", "Biryani - PKR 500", "Samosas - PKR 100"**

**};**

**menuDropdown = new JComboBox<>(menuItems);**

**menuDropdown.setFont(labelFont);**

**quantitySpinner = new JSpinner(new SpinnerNumberModel(1, 1, 100, 1));**

**quantitySpinner.setFont(labelFont);**

**JButton addButton = createStyledButton("Add to Order", buttonColor, buttonTextColor, buttonFont);**

**menuSelectionPanel.add(createStyledLabel("Menu Item:"));**

**menuSelectionPanel.add(menuDropdown);**

**menuSelectionPanel.add(createStyledLabel("Quantity:"));**

**menuSelectionPanel.add(quantitySpinner);**

**menuSelectionPanel.add(addButton);**

**// Order List Panel**

**orderListModel = new DefaultListModel<>();**

**JList<String> orderList = new JList<>(orderListModel);**

**orderList.setFont(new Font("Monospaced", Font.PLAIN, 16));**

**orderList.setBackground(Color.WHITE);**

**JScrollPane orderScrollPane = new JScrollPane(orderList);**

**orderScrollPane.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(accentColor, 2), "Order Summary", 0, 0, labelFont, accentColor));**

**menuPanel.add(menuSelectionPanel, BorderLayout.NORTH);**

**menuPanel.add(orderScrollPane, BorderLayout.CENTER);**

**// Receipt and Buttons Panel**

**JPanel receiptPanel = new JPanel(new BorderLayout(10, 10)); // Added spacing**

**receiptPanel.setBackground(Color.WHITE);**

**receiptPanel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(accentColor, 2), "Receipt and Actions", 0, 0, labelFont, accentColor));**

**receiptArea = new JTextArea(15, 30);**

**receiptArea.setEditable(false);**

**receiptArea.setFont(new Font("Monospaced", Font.PLAIN, 14));**

**receiptArea.setBorder(BorderFactory.createLineBorder(accentColor, 1));**

**JScrollPane receiptScrollPane = new JScrollPane(receiptArea);**

**JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 20)); // Increased button spacing**

**buttonPanel.setBackground(Color.WHITE);**

**JButton generateReceiptButton = createStyledButton("Generate Receipt", buttonColor, buttonTextColor, buttonFont);**

**JButton clearButton = createStyledButton("Clear", new Color(220, 53, 69), Color.BLACK, buttonFont);**

**JButton exitButton = createStyledButton("Exit", Color.GRAY, Color.BLACK, buttonFont);**

**buttonPanel.add(generateReceiptButton);**

**buttonPanel.add(clearButton);**

**buttonPanel.add(exitButton);**

**receiptPanel.add(receiptScrollPane, BorderLayout.CENTER);**

**receiptPanel.add(buttonPanel, BorderLayout.SOUTH);**

**// Add Panels to Frame**

**JPanel mainPanel = new JPanel(new BorderLayout(20, 20)); // Add spacing between main panels**

**mainPanel.setBackground(bgColor);**

**mainPanel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30)); // Add spacing to the edges**

**mainPanel.add(customerPanel, BorderLayout.NORTH);**

**mainPanel.add(menuPanel, BorderLayout.CENTER);**

**mainPanel.add(receiptPanel, BorderLayout.SOUTH);**

**add(mainPanel, BorderLayout.CENTER);**

**// Button Actions**

**addButton.addActionListener(e -> addToOrder());**

**generateReceiptButton.addActionListener(e -> generateReceipt());**

**clearButton.addActionListener(e -> clearForm());**

**exitButton.addActionListener(e -> System.exit(0));**

**// Make the frame visible**

**setVisible(true);**

**}**

**private JLabel createStyledLabel(String text) {**

**JLabel label = new JLabel(text);**

**label.setFont(new Font("SansSerif", Font.PLAIN, 16));**

**label.setForeground(Color.DARK\_GRAY);**

**return label;**

**}**

**private JTextField createStyledTextField() {**

**JTextField textField = new JTextField();**

**textField.setFont(new Font("SansSerif", Font.PLAIN, 16));**

**return textField;**

**}**

**private JButton createStyledButton(String text, Color bgColor, Color fgColor, Font font) {**

**JButton button = new JButton(text);**

**button.setFont(font);**

**button.setBackground(bgColor);**

**button.setForeground(fgColor);**

**return button;**

**}**

**private void addToOrder() {**

**String selectedItem = menuDropdown.getSelectedItem().toString();**

**int quantity = (int) quantitySpinner.getValue();**

**String[] itemParts = selectedItem.split(" - PKR ");**

**int price = Integer.parseInt(itemParts[1]) \* quantity;**

**orderListModel.addElement(itemParts[0] + " x" + quantity + " - PKR " + price);**

**totalBill += price;**

**}**

**private void generateReceipt() {**

**String receipt = "Customer Name: " + customerNameField.getText() + "\n"**

**+ customerAgeField.getText() + "\n" + customerIdField.getText()**

**+ "Total Bill: PKR " + totalBill + "\nThank you for dining with us!";**

**receiptArea.setText(receipt);**

**saveReceiptToFile(receipt); // Save receipt to file**

**}**

**private void saveReceiptToFile(String receiptContent) {**

**try {**

**File receiptFile = new File("receipt.txt"); // File to save the receipt**

**if (!receiptFile.exists()) {**

**receiptFile.createNewFile(); // Create the file if it doesn't exist**

**}**

**FileWriter writer = new FileWriter(receiptFile, true); // Open file in append mode**

**writer.write(receiptContent + "\n\n"); // Write the receipt content**

**writer.close(); // Close the file**

**JOptionPane.showMessageDialog(this, "Receipt saved to receipt.txt", "Receipt Saved", JOptionPane.INFORMATION\_MESSAGE);**

**} catch (IOException e) {**

**JOptionPane.showMessageDialog(this, "Error saving receipt: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**private void clearForm() {**

**customerNameField.setText("");**

**customerPhoneField.setText("");**

**customerAgeField.setText("");**

**customerIdField.setText("");**

**orderListModel.clear();**

**receiptArea.setText("");**

**totalBill = 0;**

**}**

**public static void main(String[] args) {**

**new RestaurantBillingSystem();**

**}**

**}**