package restaurant;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class Restaurant {

MenuItem[] menu;

public Restaurant(int menuSize) {

this.menu = new MenuItem[menuSize];

}

public void addMenuItem(MenuItem menuItem, int index) {

this.menu[index] = menuItem;

}

}

class MenuItem {

private String name;

private double price;

public MenuItem(String name, double price) {

this.name = name;

this.price = price;

}

public String getName() {

return name;

}

public double getPrice() {

return price;

}

}

class Order {

private MenuItem[] items;

private int itemCount;

public Order(int orderSize) {

this.items = new MenuItem[orderSize];

this.itemCount = 0;

}

public void addItem(MenuItem item) {

if (itemCount < items.length) {

items[itemCount++] = item;

} else {

System.out.println("Order is full. Cannot add more items.");

}

}

public double calculateTotal() {

double total = 0;

for (int i = 0; i < itemCount; i++) {

total += items[i].getPrice();

}

return total;

}

public void resetOrder() {

items = new MenuItem[items.length];

itemCount = 0;

}

public int getItemCount() {

return itemCount;

}

public MenuItem getItem(int i) {

if (i >= 0 && i < itemCount) {

return items[i];

} else {

return null;

}

}

}

class RestaurantGUI extends JFrame {

private Restaurant restaurant;

private Order order;

private JTextArea textArea;

private JComboBox<String> menuComboBox;

public RestaurantGUI() {

this.restaurant = new Restaurant(10);

this.order = new Order(10);

restaurant.addMenuItem(new MenuItem("Ravioli", 230), 0);

restaurant.addMenuItem(new MenuItem("Pizza Margherita", 340), 1);

restaurant.addMenuItem(new MenuItem("Spaghetti", 250),2 );

restaurant.addMenuItem(new MenuItem("Truffles", 280), 3);

restaurant.addMenuItem(new MenuItem("Gelato", 250), 4);

restaurant.addMenuItem(new MenuItem("Bruschetta", 300), 5);

restaurant.addMenuItem(new MenuItem("Tiramisù", 380), 6);

restaurant.addMenuItem(new MenuItem("Cheesecake", 220), 7);

restaurant.addMenuItem(new MenuItem("Brownie", 100), 8);

restaurant.addMenuItem(new MenuItem("S'mores", 200), 9);

JPanel panel = new JPanel();

JButton displayMenuButton = new JButton("Display Menu");

JButton placeOrderButton = new JButton("Place Order");

JButton calculateBillButton = new JButton("Calculate Bill");

textArea = new JTextArea(10, 30);

textArea.setEditable(false);

// Action Listeners

displayMenuButton.addActionListener(e -> displayMenu());

placeOrderButton.addActionListener(e -> placeOrder());

calculateBillButton.addActionListener(e -> calculateBill());

// Adding components to the panel

panel.add(displayMenuButton);

panel.add(placeOrderButton);

panel.add(calculateBillButton);

// Adding panel and text area to the frame

add(panel, BorderLayout.NORTH);

add(new JScrollPane(textArea), BorderLayout.CENTER);

// Frame settings

setTitle("Restaurant Management System");

setSize(500, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

}

private void displayMenu() {

textArea.setText("Menu:\n");

for (MenuItem item : restaurant.menu) {

textArea.append(item.getName() + " - ₹" + item.getPrice() + "\n");

}

}

private void placeOrder() {

String[] menuItems = new String[restaurant.menu.length];

for (int i = 0; i < restaurant.menu.length; i++) {

menuItems[i] = restaurant.menu[i].getName();

}

menuComboBox = new JComboBox<>(menuItems);

int result = JOptionPane.showOptionDialog(

this,

menuComboBox,

"Select Item",

JOptionPane.OK\_CANCEL\_OPTION,

JOptionPane.QUESTION\_MESSAGE,

null,

null,

null

);

if (result == JOptionPane.OK\_OPTION) {

String selectedItemName = (String) menuComboBox.getSelectedItem();

MenuItem selectedMenuItem = null;

for (MenuItem item : restaurant.menu) {

if (item.getName().equalsIgnoreCase(selectedItemName)) {

selectedMenuItem = item;

break;

}

}

if (selectedMenuItem != null) {

order.addItem(selectedMenuItem);

JOptionPane.showMessageDialog(this, selectedItemName + " added to your order.");

} else {

JOptionPane.showMessageDialog(this, "Item not found in the menu.");

}

}

}

private void calculateBill() {

if (order.getItemCount() == 0) {

JOptionPane.showMessageDialog(this, "Your order is empty. Please add items before calculating the bill.");

return;

}

StringBuilder orderSummary = new StringBuilder();

double total = 0;

for (int i = 0; i < order.getItemCount(); i++) {

MenuItem item = order.getItem(i);

orderSummary.append(item.getName()).append(" - ₹").append(item.getPrice()).append("\n");

total += item.getPrice();

}

orderSummary.append("Total: ₹").append(total);

JOptionPane.showMessageDialog(this, orderSummary.toString());

order.resetOrder(); // Reset the order after calculating the bill

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

RestaurantGUI restaurantGUI = new RestaurantGUI();

restaurantGUI.setVisible(true);

});

}

}