Restaurant Ordering & Billing System Assignment

1. Description

Design and implement a modular and extensible Restaurant Ordering & Billing System that enables customers to browse menus, customize meals, place orders, process payments, and generate bills. The system must adhere to SOLID design principles and apply all the required object-oriented design patterns. The goal is to demonstrate clean architecture, flexibility for future extensions, and maintainability through proper design abstraction.

2. System Overview

The system should simulate the complete workflow of a restaurant, from ordering to billing. The following sections describe the required components

Ordering Workflow

- Menu Display & Order Creation:
  - Provide a unified interface for managing customer interactions such as displaying menus, creating orders, customer billing, ...

- Multiple Menu Types:
  - The restaurant may have several types of menus such as Vegetarian, Non- Vegetarian, and Kids Menu. Encapsulate the creation of related menu families without specifying their concrete classes.

- Menu Item Variations:
  - Each menu category can contain multiple variants, for example, Italian Pizza, Eastern Pizza, or Classic Burger. Prepare specific menu items dynamically based on the order request.

- Customizable Add-ons:
  - Allow customers to customize their meals with add-ons like extra cheese, sauces, or toppings.

- Order Notification to Kitchen:
  - When an order is placed, the kitchen and waiter systems should be acknowledged with the new order.

- Payment & Billing:
  - Support multiple payment methods such as Cash, Credit Card, and Mobile Wallets.

- Discount Handling:
  - Create various discount strategies based on item categories (e.g., Chicken, Meat, Pizza discounts). Each discount method should define its own calculation logic and apply automatically during checkout.

- Ordering Workflow Definition:
  - The sequence includes displaying the menu, selecting items and add-ons, choosing order type (dine-in, delivery, takeaway), calculating totals, applying taxes, discounts, and generating the final receipt before placing the order.

3. Evaluation Criteria

Criteria	Weight
Application of SOLID Principles	20%
Appropriate and correct use of Design Patterns	50%
Code Structure, Modularity, and Extensibility	10%
UML Diagram, and Clarity	20%

4. Deliverables

• Source Code – fully functional, well-organized, and properly commented.
• UML Class Diagram – clearly showing class relationships and applied patterns.
• Document file - contains an explanation of your design decisions.
• README File – instructions for running the project, including example test cases and discount scenarios.

5. Submission Rules

- Each assignment must be submitted **by a team of two students** who belong to the **same lab group or have the same TA**.
- The submission deliverables must be **compressed into a single ZIP file** named using the following format:
  Ass1_ID1_ID2_TAname.zip
  (Example: Ass1_20201234_20204567_AhmedSamir.zip)
- **Academic Integrity:** Any form of plagiarism, including the use of **AI tools** or **identical submissions** between teams, will result in a **deduction of 5 marks** from the other coursework grade in addition to **zero in this assignment** and may be subject to further disciplinary action.