



Cairo University



## **Faculty of Computers and Artificial Intelligence**

**Cairo University**

### **Assignment ASWE**

Submitted to

**Dr. Manar Elkady**

Prepared by

**Course Name: Advanced Software Engineering**

**Assignment: one**

**Section: IS\_S3&4**

**ID**

**NAME**

**EMAILS**

<b>20230650</b>	Ahmed Mahmoud Ibrahim	<a href="mailto:ahmedmahmoudibrahim7723@gmail.com">ahmedmahmoudibrahim7723@gmail.com</a>	
<b>20230368</b>	Mahmoud Ahmed Ibrahim	<a href="mailto:ma3549681@gmail.com">ma3549681@gmail.com</a>	

## 1. Project Overview

We have designed and implemented a modular, extensible, and fully functional Restaurant Ordering System that simulates the complete workflow from menu display to final billing. The system strictly follows SOLID principles and demonstrates clean usage of five major design patterns as required.

## 2. Design Patterns Applied

Pattern	Location	Purpose
Facade	RestaurantFacade()	Provides unified interface to complex subsystems (menu, order, payment)
Abstract Factory	VegMenuFactory() NonVegMenuFactory() KidsMenuFactory()	Creates families of related menu objects without specifying concrete classes
Decorator	ExtraCheese() ExtraSauce() MushroomTopping()	Dynamically adds responsibilities (add-ons) to menu items
Strategy	PizzaDiscount() KidsMenuDiscount() MeatDiscount() ChickenDiscount()	Encapsulates different discount algorithms and apply them at runtime
Observer	KitchenNotification() WaiterNotification()	Notifies kitchen and waiter instantly when a new order is placed

### **3. SOLID Principles Compliance**

- Single Responsibility: Each class has one clear responsibility
- Open/Closed: System is open for extension (new menus, add-ons, discounts) and closed for modification
- Liskov Substitution: All concrete menu items can replace IMenuItem safely
- Interface Segregation: Small, focused interfaces (IMenu, IPaymentMethod, etc.)
- Dependency Inversion: High-level modules depend on abstractions

### **4. Key Features Implemented**

- Unified interactive console interface
- Three menu types: Vegetarian, Non-Vegetarian, Kids (with 4 meals)
- Multiple items per category with realistic prices
- Customizable add-ons (Extra Cheese +15, Extra Sauce +10, Mushroom +20) – added only once per type
- Order types: Dine-in, Delivery, Takeaway
- Multiple payment methods: Cash, Credit Card, Mobile Wallet
- Automatic discount calculation based on order content:
  - 15% off when ordering 2 or more pizzas
  - 10% off on any Kids meal
  - EGP 25 off on Meat items (e.g., Pepperoni)
- Real-time notification to Kitchen and Waiter (Observer Pattern)
- Clean receipt with 14% tax and final total

### **5. Extensibility & Future Enhancements**

- New menu types (e.g., Seafood, Desserts) → create new Factory + Menu class
- New add-ons → create new Decorator class

- New discount strategies → implement IDiscountStrategy
- New payment methods → implement IPaymentMethod