Advanced Software Engineering - Assignment 1

UML Class Diagrams
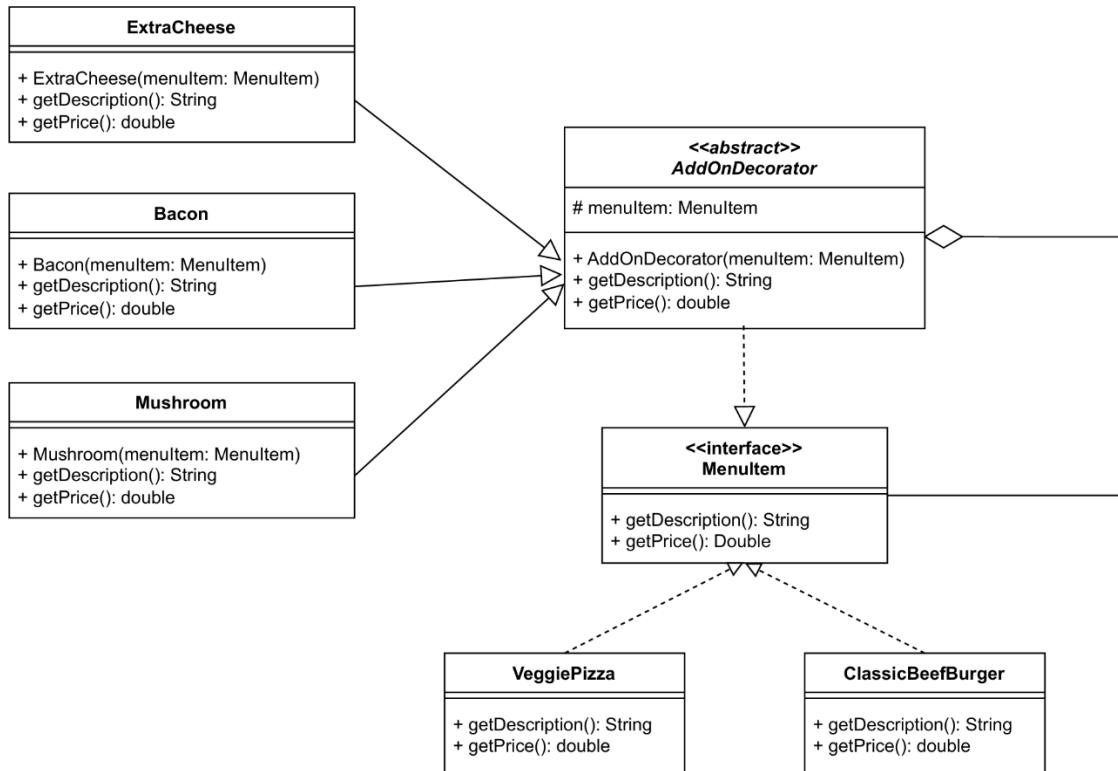
Course: Advanced Software Engineering 2025

**Student ID 1 (Eyad):** 20230074
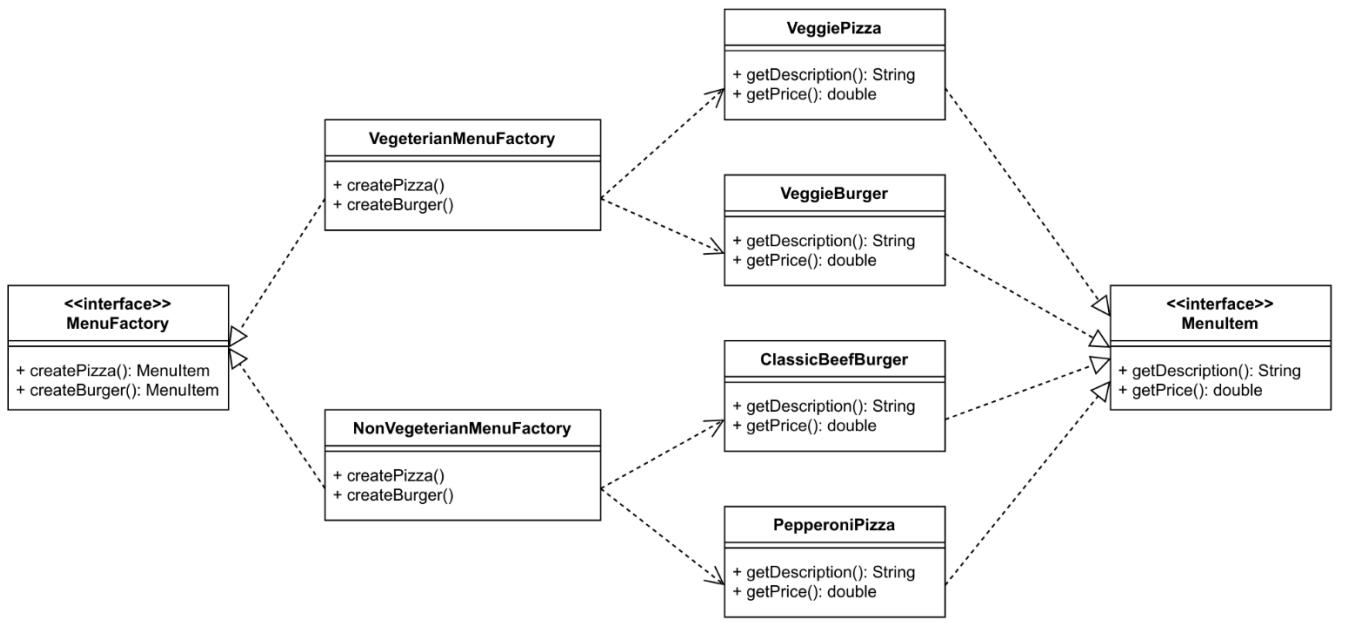
Student ID 2 (AlMoaid): 20210625

TA Name: Marwa Ahmed
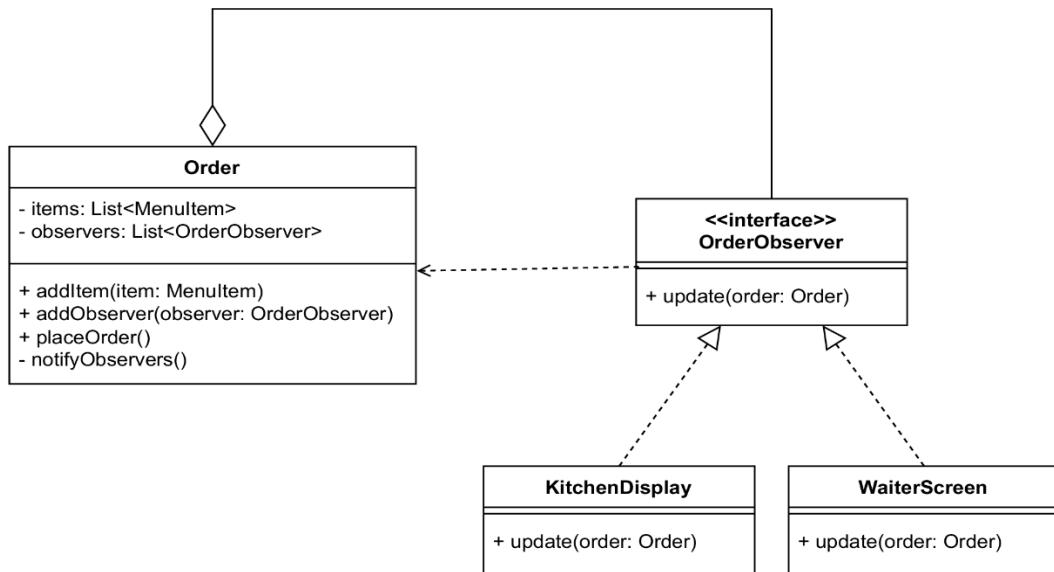
1. The Decorator Pattern Diagram
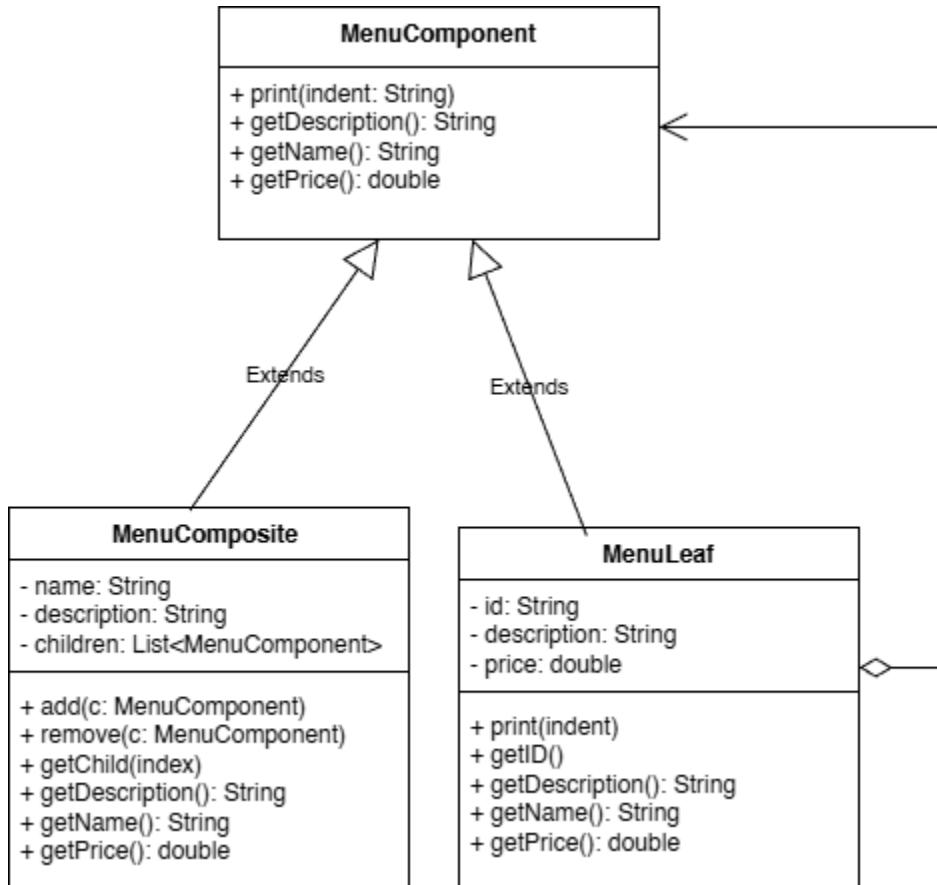
2. The Abstract Factory Pattern Diagram

VeggiePizza

+ getDescription(): String
+ getPrice(): double

VegeterianMenuFactory

+ createPizza()
+ createBurger()

VeggieBurger

+ getDescription(): String
+ getPrice(): double

<<interface>>
MenuFactory

+ createPizza(): MenuItem
+ createBurger(): MenuItem

ClassicBeefBurger

+ getDescription(): String
+ getPrice(): double

<<interface>>
MenuItem

+ getDescription(): String
+ getPrice(): double

NonVegeterianMenuFactory

+ createPizza()
+ createBurger()

PepperoniPizza

+ getDescription(): String
+ getPrice(): double

3. The Observer Pattern Diagram

Order

- items: List<MenuItem>
- observers: List<OrderObserver>

+ addItem(item: MenuItem)
+ addObserver(observer: OrderObserver)
+ placeOrder()
- notifyObservers()

<<interface>>
OrderObserver

+ update(order: Order)

KitchenDisplay

+ update(order: Order)

WaiterScreen

+ update(order: Order)
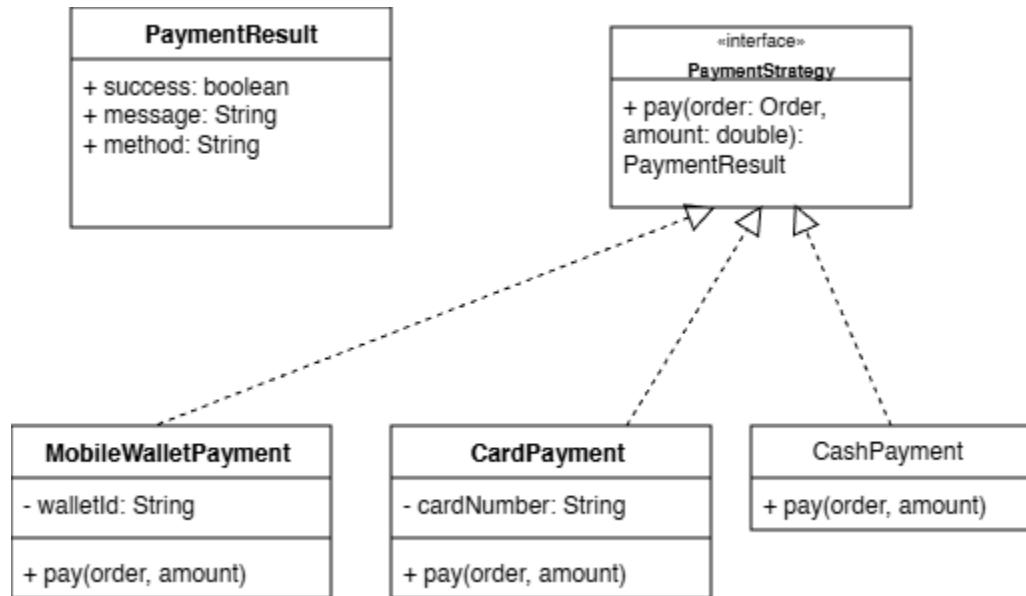
4. The Composite Pattern Diagram

MenuComponent

+ print(indent: String)
+ getDescription(): String
+ getName(): String
+ getPrice(): double

Extends

Extends

MenuComposite

- name: String
- description: String
- children: List<MenuComponent>

+ add(c: MenuComponent)
+ remove(c: MenuComponent)
+ getChild(index)
+ getDescription(): String
+ getName(): String
+ getPrice(): double

MenuLeaf

- id: String
- description: String
- price: double

+ print(indent)
+ getID()
+ getDescription(): String
+ getName(): String
+ getPrice(): double

5. The Strategy Pattern Diagram

A. Payment Strategy

```
PaymentResult
─────────────────────
+ success: boolean
+ message: String
+ method: String
```

```
«interface»
PaymentStrategy
─────────────────────
+ pay(order: Order,
amount: double):
PaymentResult
```

```
MobileWalletPayment
─────────────────────
- walletId: String
─────────────────────
+ pay(order, amount)
```

```
CardPayment
─────────────────────
- cardNumber: String
─────────────────────
+ pay(order, amount)
```

```
CashPayment
─────────────────────
+ pay(order, amount)
```

B. Discount Strategy

```
«interface»
DiscountStrategy
─────────────────────
+ calculateDiscount(order:
Order): double
```

```
NoDiscount
─────────────────────
+ calculateDiscount(order)
```

```
PizzaDiscount
─────────────────────
- percent: double
─────────────────────
+ calculateDiscount(order)
```

```
ComboDiscount
─────────────────────
- amount: double
─────────────────────
+ calculateDiscount(order)
```

6. The Facade Pattern Diagram

RestaurantFacade

- store: DataStorage
- publisher: OrderPublisher

+ createOrder(): Order
+buildMenusFromFactories()
+ addItemToOrder(order, leafId, qty)
+ addAddonToLastItem(order, addonType)
+showOrderItems(order)
+ checkout(order)
+ showMenu(name)
+ registerObserver(obs)
+ listAllOrders()
+finalizeOrder(order,PaymentStrategy)

Use → payment

Use → pricing

Use → factory

Use → billing

Use ↓ storage

Use ↓ decorators

Use ↓ notification

Use ↓ structure