



COMSATS University Islamabad

Abbottabad campus

ASSIGNMENT#1

SUBJECT: Design Patterns

SUBMITTED TO: Sir Mukhtiar Zamin

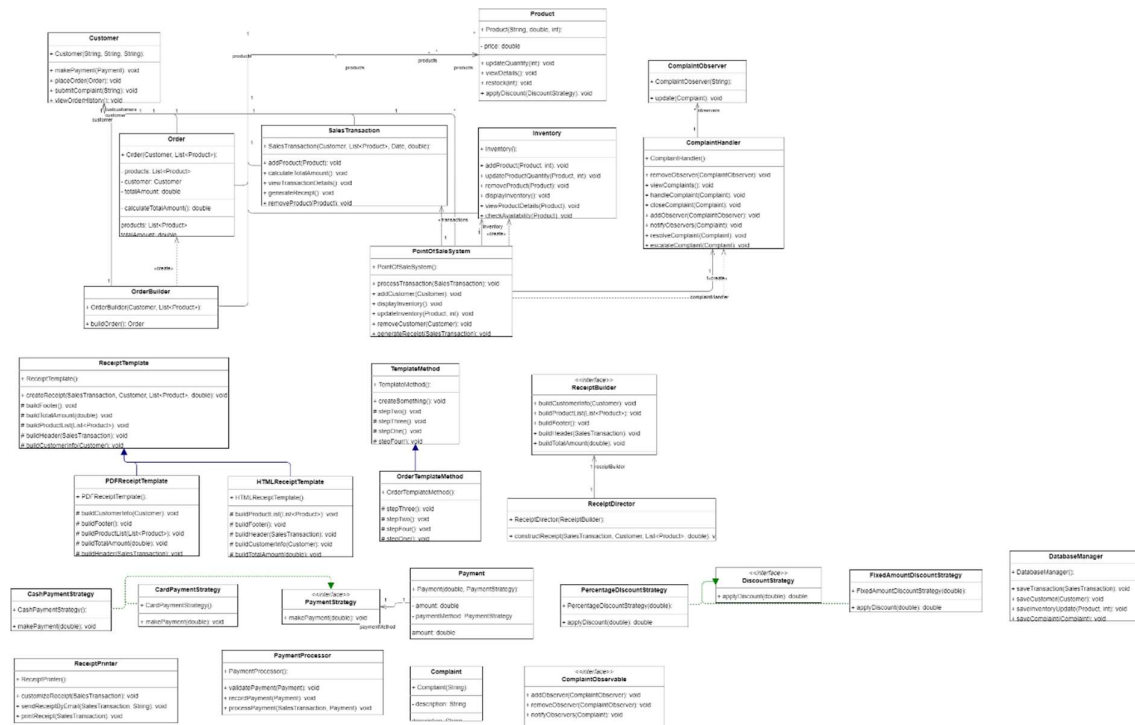
SUBMITTED BY:

- **Najeeb Said**
 - FA20-BSE-023
- **Muhammad Kashan**
 - FA20-BSE-008

DATE: Thursday, 26 October 2023

Project Name: POS System with Complain Handling

Diagram:



How are Patterns Used:

Observer Design Pattern:

Classes Using Observer:

- Customer (Subject)
- Inventory (Subject)
- ComplaintObservable (Subject)
- PaymentProcessor (Observer)

In these classes, the Observer pattern is used to establish a one-to-many dependency between subjects (like Customer, Inventory, ComplaintObservable) and observers (like PaymentProcessor).

Observers register themselves with the subject to receive notifications about any changes in state. For example, when a new payment is processed, the PaymentProcessor is notified and can react accordingly.

Builder Design Pattern:

Classes Using Builder:

- ReceiptBuilder (Builder)
- ReceiptDirector (Director)
- OrderBuilder (Builder)

The Builder pattern separates the construction of a complex object from its representation. It allows different representations to be created using the same construction process.

In this case, ReceiptBuilder and OrderBuilder are responsible for constructing complex receipts and orders, respectively. ReceiptDirector provides a high-level interface to control the construction process.

Strategy Design Pattern:

Classes Using Strategy:

- PaymentProcessor (Context)
- PaymentStrategy (Strategy)
- CashPaymentStrategy (Concrete Strategy)
- CardPaymentStrategy (Concrete Strategy)
- DiscountStrategy (Strategy)
- PercentageDiscountStrategy (Concrete Strategy)
- FixedAmountDiscountStrategy (Concrete Strategy)

The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. It allows the algorithm to vary independently from the client that uses it.

In this context, different payment strategies (CashPaymentStrategy, CardPaymentStrategy) and discount strategies (PercentageDiscountStrategy, FixedAmountDiscountStrategy) can be dynamically selected and applied by the PaymentProcessor.

Template Design Pattern:

Classes Using Template Method:

- ReceiptTemplate (Abstract Class)
- HTMLReceiptTemplate (Concrete Class)
- PDFReceiptTemplate (Concrete Class)
- OrderTemplateMethod (Abstract Class)

The Template Method pattern defines the skeleton of an algorithm in the superclass but let's subclasses override specific steps of the algorithm without changing its structure.

In this case, ReceiptTemplate defines the overall structure of generating receipts, but HTMLReceiptTemplate and PDFReceiptTemplate provide specific implementations. Similarly, OrderTemplateMethod outlines the process of order processing, with specific steps overridden in subclasses.