THE UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

NAME: CHAABILO LUBOBYA

COMPUTER NUMBER: 2021418235

COURSE: ADVANCED SOFTWARE ENGINEERING

COURSE CODE: 4630

LAB 3

APPLYING GRASP GoF DESIGN PATTERNS

Problem Statement

Design an Online Learning Platform with these features:

1. Multiple Course Formats: Video, Text, Live (e.g., Zoom-based).
2. Flexible Payment Processing: Support multiple payment gateways.
3. User Notifications: Alerts on enrollment/completion.
4. Dynamic Course Enhancements: Add features like certificates/badges post-enrollment.

1. Domain Modeling & Problem Analysis

Core Objects:

- Course
- User
- PaymentProcessor
- NotificationService
- Certificate / Badge

Pain Points & Pattern Triggers

Pain Point	Why it's a Problem	Pattern(s) to Consider
Course Instantiation	Different course formats (Video, Text, Live)	**Factory Method** or **Abstract Factory**
Payment Processing	Multiple gateways needed (e.g., PayPal, Stripe)	**Strategy**
User Notifications	Must decouple alerts (e.g., Email, SMS)	**Observer**
Add Certificates/Badges at Runtime	Post-enrollment features, avoid subclass bloat	**Decorator**

2. Pattern Discovery & Application

Pattern 1: Factory Method

Used for: Course creation (video/text/live).

Justification: Creation logic varies; applying Creator and Low Coupling.

Pattern 2: Strategy

Used for: Payment gateway switching.

Justification: Behavior varies independently; Polymorphism, Low Coupling.

Pattern 3: Observer

Used for: Sending notifications on events (enrollment/completion).

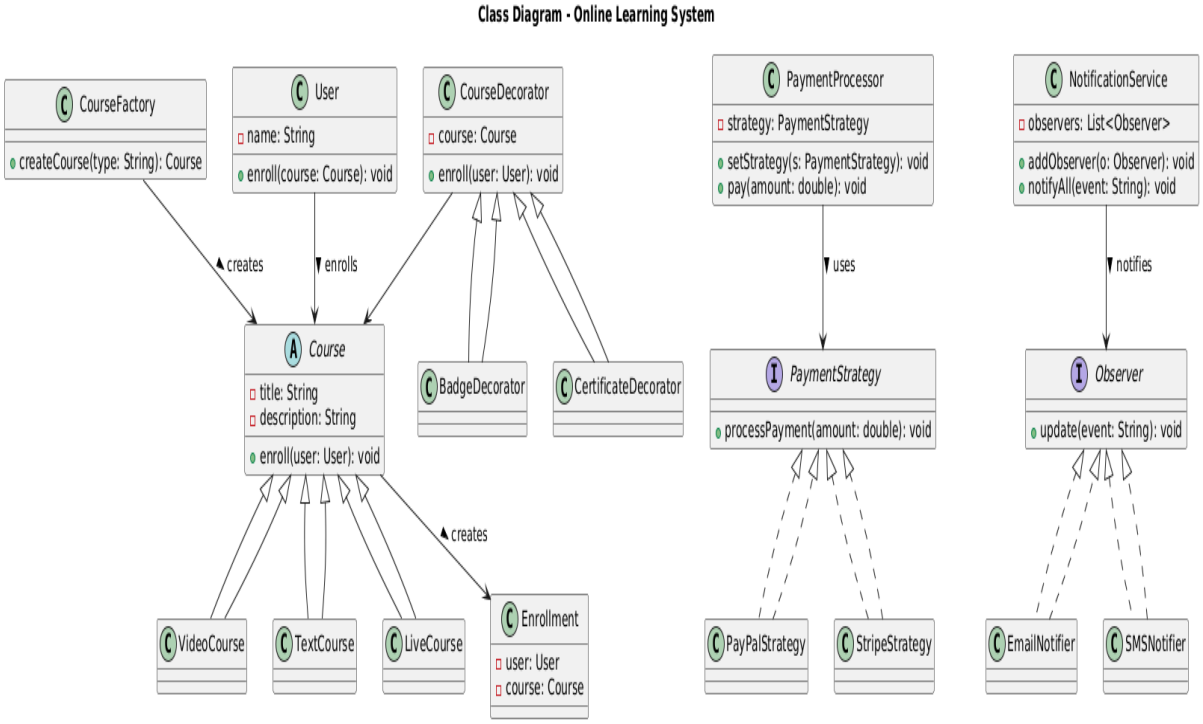
Justification*:* Indirection and Low Coupling avoid hardwiring all outputs.

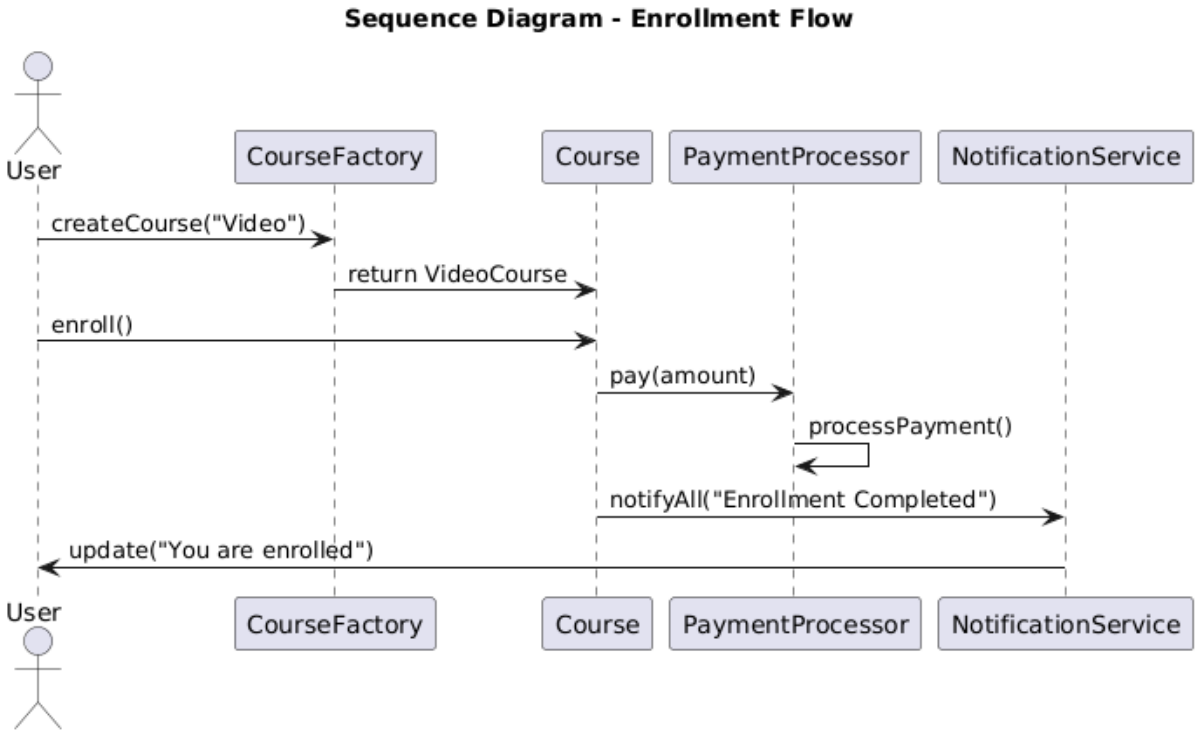
Pattern 4: Decorator

Used for*:* Dynamically adding certificates or badges.

Justification: Avoid subclassing every feature combination. Use High Cohesion, Open-Closed Principle.

3. UML Modeling





Design Justification

Pattern	Applied To	GRASP Justification
Factory Method	Course creation	Creator, Low Coupling
Strategy	Payment processing	Polymorphism, Low Coupling
Observer	Notifications	Indirection, Low Coupling
Decorator	Feature enhancement (e.g., Certificates)	Open-Closed Principle, High Cohesion