HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



School of Information and Communications Technology

Software Requirement Specification

AIMS – An Internet Media Store ITSS Software Development

Nhóm 20

Design Pattern

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Vai trò |
| Nguyễn Duy Tấn | 20215478 | Nhóm trưởng |
| Nguyễn Văn Tấn | 20215479 | Thành viên |
| Lù Mạnh Thắng | 20194167 | Thành viên |
| Lưu Trọng Tấn | 20215477 | Thành viên |
| thanongsith thavisack | 20180288 | Thành viên |

*Hanoi, January 2025*

# Vấn đề 1: thêm sản phẩm mới AudioBook

Đối với yêu cầu 1: Thêm mặt hàng Media mới: AudioBook

Thông tin về loại mặt hàng mới như sau:

AudioBook

+ String author, ví dụ: Paulo Coelho

+ String format, ví dụ: mp3

+ String language, ví dụ: English

+ String accent, ví dụ: Male - North America

+ int lengthInMinutes, ví dụ: 226

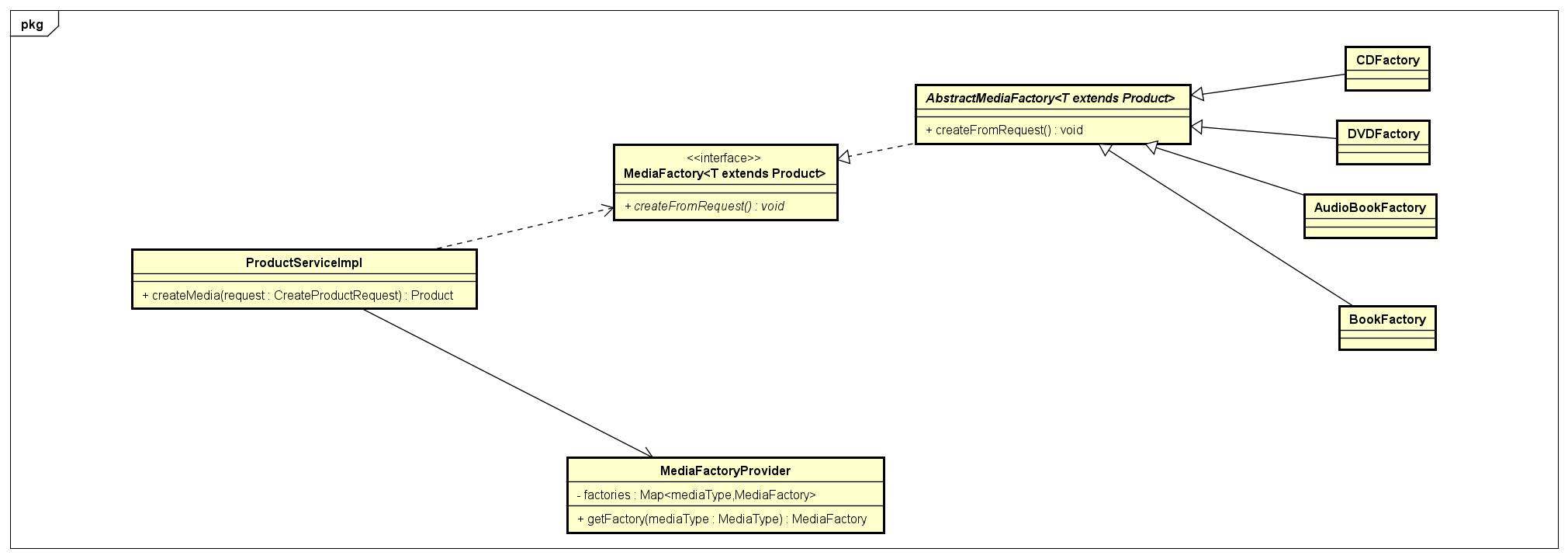
**Vấn đề**

Cơ sở dữ liệu của nhóm đang sử dụng MongoDB lưu tất cả các loại sản phẩm vào trong Document Product. Việc thêm một sản phẩm mới trong cơ sở dữ liệu sẽ gây ra control coupling, đồng thời vi phạm tính Open-Close Principle trong SOLID minh họa trong đoạn code sau

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

**Để giải quyết** vấn đề trên Nhóm đã áp dụng factory Method như sau.



Triển khai cụ thể

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Đối với mỗi request để tạo loại sản phẩm mới, mỗi loại sản phẩm có các thuộc tính riêng được gửi lên sử dụng extend để kế thừa các thuộc tính chung.

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Phương thức createMedia được gọi có triển khai như sau

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

Dựa vào type trong product sẽ lấy được các factory như AudioFactory, AudioFactory sẽ sử dụng phương thức createFromRequest() được kế thừa từ AbstractMediaFactory , và trong lớp AbstractMediaFactory phương thức createFromRequest() đã được implement. Tạo 1 sản phẩm mới từ request dựa trên request được gửi.

Triển khai của các factory như sau

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

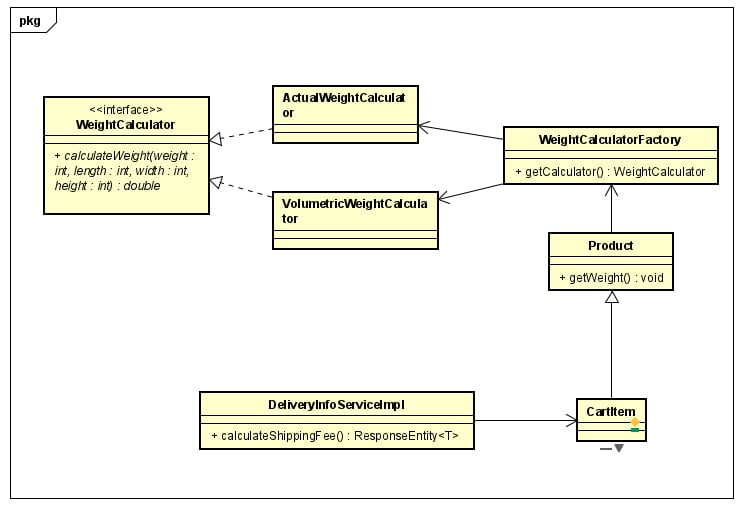
# Vấn đề 2: Thay đổi cách tính phí vận chuyển

**Vấn đề:**

Trong hệ thống hiện tại, phí vận chuyển được tính dựa trên khối lượng thực tế của kiện hàng. Tuy nhiên, yêu cầu mới đòi hỏi phải xem xét thêm khối lượng quy đổi (dựa trên kích thước) và tính phí dựa trên giá trị lớn hơn giữa hai khối lượng. Việc tích hợp logic này vào các phương thức tính phí có thể gây ra "control coupling", làm hệ thống khó bảo trì.

**Giải pháp:**

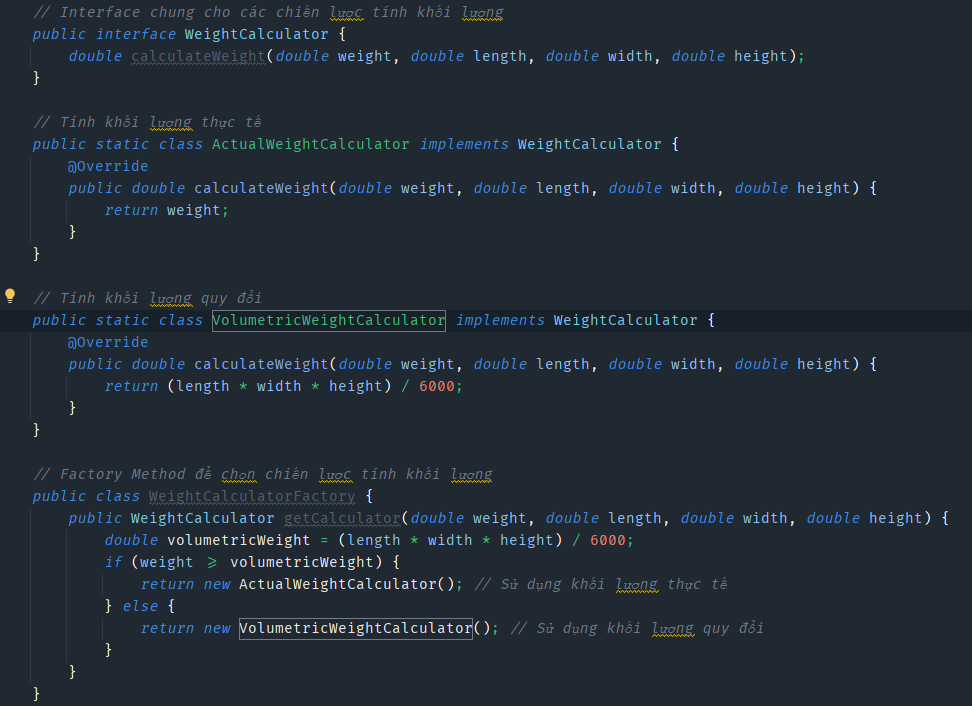
Nhóm đã áp dụng Factory Method để tách biệt việc quyết định cách tính khối lượng và logic tính phí. Factory Method được sử dụng để tạo các đối tượng thực hiện tính khối lượng phù hợp dựa trên dữ liệu đầu vào.



**Triển khai:**

1. Factory Method để chọn chiến lược tính khối lượng:

Phương pháp này quyết định sử dụng cách tính khối lượng dựa trên trọng lượng thực tế hoặc khối lượng quy đổi bằng cách tạo các đối tượng chiến lược.



1. Hàm getWeight trong lớp Product:

Để đơn giản hóa việc truy xuất khối lượng, phương thức getWeight sử dụng WeightCalculatorFactory để tạo chiến lược và tính toán khối lượng phù hợp.

A screen shot of a computer code

Description automatically generated

1. Hàm calculateShippingFee:

Hàm này tính phí vận chuyển dựa trên các tham số đầu vào như ID giỏ hàng, tỉnh thành, và yêu cầu giao gấp. Logic bao gồm:

* Xác định khối lượng tổng: Sử dụng phương thức getWeight() cho từng sản phẩm trong giỏ hàng.
* Tính phí giao hàng nội tỉnh hoặc ngoại tỉnh: Phí giao hàng được tính theo quy tắc có sẵn, có tính đến khối lượng tổng và các ngưỡng phụ phí.
* Tính phí giao gấp: Thêm phí giao gấp dựa trên số lượng sản phẩm.
* Miễn phí vận chuyển: Giảm trừ phí giao hàng nếu tổng giá trị giỏ hàng đạt ngưỡng quy định.

A screen shot of a computer program

Description automatically generated

**Kết quả:**

Thiết kế này giúp tích hợp logic tính khối lượng và phí vận chuyển một cách tách biệt và linh hoạt. Việc áp dụng Factory Method đảm bảo rằng việc tạo và chọn chiến lược tính toán khối lượng được thực hiện chính xác và dễ bảo trì.

# Vấn đề 3 Thêm phương thức thanh toán mới

Đối với vấn đề

3. Thêm phương thức thanh toán mới: Thẻ nội địa (Domestic Card)

Khách hàng có thể lựa chọn phương thức thanh toán thẻ tín dụng (credit card) hoặc

thẻ nội địa (domestic card). Thông tin về loại thẻ thanh toán mới như sau:

+ Type: Domestic Debit Card

+ Issuing Bank, ví dụ VietinBank

+ Card number 16 kí tự là chữ số

+ Valid-from date, ví dụ: 12/33

+ Cardholder’s name, ví dụ: DO MINH HIEU

Với thẻ nội địa, chương trình vẫn kết nối với API của Interbank, chỉ thay đổi

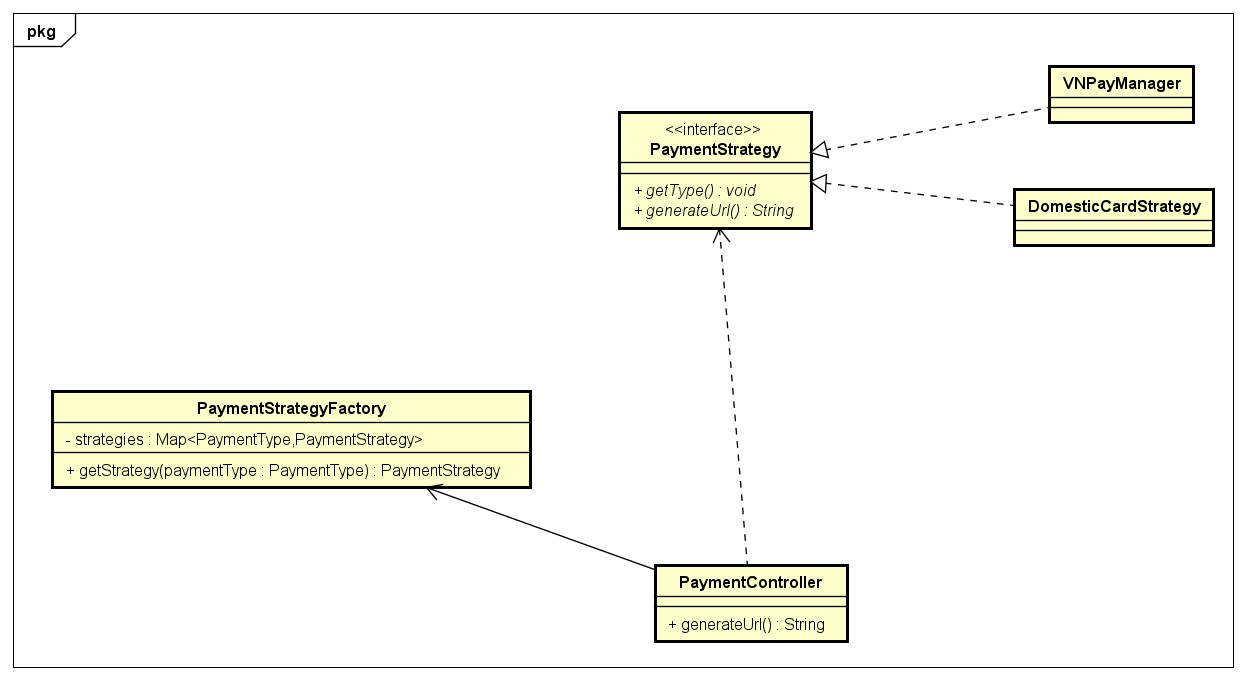
thông tin của phương thức thanh toán.

**Vấn đề:** control coupling và đồng thời vi phạm tính Open-Close Principle tiếp tục có thể xảy ra minh họa với đoạn code sau:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

**Để giải quyết** vấn đề trên nhóm đã áp dụng strategy pattern.



Đối với mỗi yêu cầu thanh toán

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Trong đó triển khai các strategy như sau:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

VNPayManager và DomesticCard sẽ implements phương thức generateUrl() , chi tiết triển khai và các thuật toán mã hóa hoặc xác thực cụ thể sẽ được implement trả về đường dẫn trực tiếp đến cổng VNPAY hoặc trang thanh toán của thẻ nội địa.

# Các design pattern bổ sung

## Singleton

Nhóm đã triển khai Singleton bên frontend để đảm bảo mỗi lần vào trang web người dùng chỉ làm việc với 1 cart duy nhất. Đồng thời tránh các vấn đề về common coupling khi delivery và order đều có quan hệ với cart và sử dụng các trường thông tin trong cart để sử dụng

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

## Builder Pattern

Ngoài ra nhóm cũng sử dụng builder pattern trong project để cung cấp một cách linh hoạt để tạo đối tượng mà không cần phải sử dụng nhiều constructor (tránh "constructor telescoping").

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, ảnh chụp màn hình

Mô tả được tạo tự động