

E-commerce Product Management System – Final Report

1. Object-Oriented Analysis (OOA)

◆ Objects (Nouns)

- **Product:** đại diện cho một mặt hàng
- **Electronics:** sản phẩm điện tử kế thừa từ Product
- **ShoppingCart:** giỏ hàng của người dùng
- **InventoryList:** danh sách sản phẩm trong kho
- **Order** (tiềm năng mở rộng): đơn hàng sau khi thanh toán

◆ Attributes (Descriptive Nouns)

- Product: id, name, price, stock
- Electronics: warrantyMonths
- ShoppingCart: cartItems (map), total
- InventoryList: items (vector)

◆ Methods (Verbs)

- addItem(), removeItem(), searchItem(), displayAll()
- updateStock(), applyDiscount(), checkout()
- Operator overloading: +=, ==

◆ Inheritance Relationships

- Electronics kế thừa từ Product
- Product kế thừa từ interface Discountable
- InventoryList<T> là template class quản lý danh sách sản phẩm

2. Class Design Overview

◆ Inheritance

- Electronics : public Product
- Product : public Discountable (interface)

◆ Interface

cpp

```
class Discountable {  
  
public:  
  
    virtual double applyDiscount(double rate) = 0;  
  
};
```

→ Được triển khai trong Product và sử dụng trong ShoppingCart.

◆ Template Class

cpp

```
template <typename T>  
  
class InventoryList {  
  
    vector<T> items;  
  
    ...  
  
};
```

→ Cho phép quản lý danh sách sản phẩm một cách tổng quát.

◆ Operator Overloading

- Product::operator== → so sánh theo id
- ShoppingCart::operator+= → thêm sản phẩm vào giỏ hàng

3. Code Walkthrough

- **Product**: lớp cơ sở với thông tin sản phẩm và phương thức giảm giá
- **Electronics**: override updateStock() để cộng thêm phí
- **ShoppingCart**: dùng map<string, pair<Product*, int>> để lưu sản phẩm và số lượng
- **InventoryList**: template class giúp thêm, tìm, xóa sản phẩm
- **Main menu**: cho phép người dùng tương tác với hệ thống

4. Test Results

◆ Tình huống kiểm thử

- Thêm sản phẩm thường và điện tử
- Hiển thị danh sách sản phẩm
- Thêm nhiều sản phẩm vào giỏ cùng lúc: P001 P002 P003
- Áp dụng giảm giá: nhập 0.1 → giảm 10%
- Thanh toán: chọn Cash hoặc Card

◆ Kết quả đầu ra (console)

Mã

=== Shopping Cart ===

P001 x2 → \$200

P002 x1 → \$150

Total after discount: \$315

Payment successful via Card.

Thank you for your purchase!

5. UML Diagrams

◆ Class Diagram

- Product → Electronics (generalization)
- Product → Discountable (realization)
- ShoppingCart sử dụng Product*
- InventoryList<T> quản lý Product*

(Vẽ bằng Lucidchart hoặc Draw.io, chèn ảnh vào PDF)

◆ Sequence Diagram (Add to Cart)

1. User nhập danh sách ID
2. Hệ thống gọi searchItem(id) cho từng ID
3. Nếu tìm thấy → gọi operator+=
4. Cập nhật giỏ hàng và giảm stock

6. Use of LLM AI Model

Tôi đã sử dụng Microsoft Copilot để:

- Gợi ý thiết kế lớp InventoryList<T> và cách dùng template
- Hướng dẫn cách xử lý nhập nhiều sản phẩm cùng lúc
- Tối ưu hóa logic thanh toán và kiểm tra giỏ hàng
- Viết lại code theo chuẩn OOP và tạo báo cáo PDF

Tôi đã tự viết lại toàn bộ mã nguồn và kiểm thử thủ công để đảm bảo tính chính xác.

Kết luận

Hệ thống đã đáp ứng đầy đủ yêu cầu:

- Sử dụng kế thừa, interface, template, operator overload
- Tương tác qua menu
- Kiểm thử đầy đủ các chức năng chính