

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----- ∞  ∞ -----



SOICT

BÁO CÁO LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Đề tài: Book Store Management

Giảng viên: **TS. Trịnh Thành Trung**

Nhóm sinh viên thực hiện: Nhóm 6 – Trịnh Thành Trung (PHD)

STT	Họ và tên	MSSV
1	Nguyễn Tài Hưng	20236034
2	Vũ Xuân Đức	20236025
3	Nguyễn Hoàng Tín	20236055
4	Đỗ Đức Minh Tuấn	20236057
5	Trần Chí Vũ	20236060
6	Phạm Văn Uy	20194716

Hà Nội, năm 2025

I. Phân tích yêu cầu đề bài

1.1 Đặt vấn đề

Hiệu sách không chỉ là nơi bán sách mà còn đóng vai trò quan trọng trong việc kết nối văn hóa đọc với cộng đồng. Đặc biệt, với sự đa dạng về thể loại sách và nhu cầu ngày càng tăng của khách hàng, việc quản lý hiệu sách trở nên phức tạp hơn, bao gồm các khía cạnh như quản lý thông tin sách, kho hàng, tài khoản khách hàng, và quy trình bán hàng. Những phương pháp quản lý truyền thống, dựa trên giấy tờ hoặc bảng tính, đã không còn phù hợp để đáp ứng các yêu cầu ngày càng cao về hiệu quả và độ chính xác.

Bên cạnh đó, sự phát triển của công nghệ thông tin mang đến cơ hội tối ưu hóa các quy trình quản lý hiệu sách. Việc xây dựng một hệ thống quản lý hiệu sách theo hướng đối tượng không chỉ hỗ trợ tổ chức thông tin một cách khoa học mà còn giúp tự động hóa các tác vụ, nâng cao trải nghiệm khách hàng, và tăng hiệu quả hoạt động. Một hệ thống như vậy cho phép người quản lý dễ dàng theo dõi hoạt động kinh doanh, kiểm soát kho hàng, và hỗ trợ khách hàng nhanh chóng hơn.

Tuy nhiên, để xây dựng một hệ thống quản lý hiệu sách hiệu quả, cần có sự phân tích chi tiết về các yêu cầu và áp dụng các nguyên lý của lập trình hướng đối tượng. Việc mô hình hóa các thực thể như sách, khách hàng, nhân viên, và quản lý cùng với các mối quan hệ giữa chúng là bước đầu tiên để đảm bảo rằng hệ thống có thể vận hành linh hoạt, mở rộng và nâng cấp trong tương lai.

Với mục tiêu đó, đề tài "Thiết kế hệ thống cho cửa hàng sách" được thực hiện, nhằm giải quyết các vấn đề cơ bản như quản lý thông tin sách, giỏ hàng, tài khoản khách hàng, và quy trình bán hàng. Đề tài không chỉ cung cấp một giải pháp kỹ thuật mà còn đóng góp vào việc nâng cao hiệu quả quản lý và phát triển bền vững ngành kinh doanh sách.

1.2 Phân tích hướng đối tượng

Yêu cầu mà hệ thống cần đáp ứng:

- Yêu cầu từ phía người dùng: Xem sách, thêm sách vào giỏ hàng, mua sách, đánh giá sách, xem lịch sử mua hàng, đăng nhập, đăng xuất, đổi mật khẩu, quản lý thông tin cá nhân.
- Yêu cầu từ phía quản lý: Quản lý sách, quản lý danh sách người dùng, xem lịch sử hoạt động của cửa hàng, quản lý các số liệu thống kê, đăng nhập, đăng xuất, đổi mật khẩu, quản lý thông tin cá nhân.
- Yêu cầu từ phía nhân viên: Quản lý sách, xem lịch sử hoạt động của cửa hàng, đăng nhập, đăng xuất, đổi mật khẩu, quản lý thông tin cá nhân.

Xác định các lớp chính của hệ thống:

- Các lớp sản phẩm:

Product
- productID: String - name: String - buyingPrice: double - sellingPrice: double - quantity: int
+ updateSellingPrice(double): boolean + updateQuantity(int): boolean + calculatePrice(int): double

Book
- isbn: String - title: String - author: String - publisher: String
+ getBookInfor(): void

Toy
- brand: String - type: String
+ getToyInfor(): void

Stationery
- brand: String - type: String
+ getStationeryInfor(): void

Book, Toy, Stationery đều thuộc chung loại Product

- Các lớp User:

Person
- ID: String - name: String - address: String - phone: String - email: String - username: String - password: String
+ getPersonInfor(): void

Customer
- cart: Cart
+ getCustomerInfo(): void

Employee
- salary: double
+ getEmployeeInfor(): void
+ processOrder(Order): boolean

Manager
- surbodinatees: List<Employee>
- salesRecords: List<Order>
+ calculateProfit(): double
+ updateSalary(double): boolean
+ getRevenueReport(start: Date, end: Date): void
+ recordExpense(double): void
+ removeEmployee(Employee): boolean
+ addEmployee(Employee): boolean

Mỗi Customer sẽ có một Cart của riêng mình.

- Các lớp quản lý và bán hàng:

OrderDetail
- orderlineID: String
- orderID: String
- product: Product
- quantity: int
+ getSubTotal(): double
+ getOrderlineInfo(): void

Order
- orderID: String
- customer: Customer
- orderlines: List<OrderDetail>
- orderDate: Date
- totalAmount: double
- paymentMethod: String
- status: String
+ getOrderInfo(): void

Mỗi OrderDetail sẽ có chỉ có 1 mã sản phẩm và số lượng sản phẩm đó. Một Order bao gồm nhiều OrderDetail.

Cart
- cartID: String - owner: Customer - items: List<CartItem>
+ getTotalAmount(): double + getTotalQuantity(): int + increaseCartQuantity(id: int, quantity: int): boolean + decreaseCartQuantity(id: int, quantity: int): boolean

CartItem
- product: Product - quantity: int
+ increaseQuantity(int): boolean + decreaseQuantity(int): boolean

- Các lớp quản lý kho:

Warehouse
- name: String - products: List<InventoryItem>
+ addProduct(p: Product): boolean + removeProduct(id:int): boolean + increaseInventory(id: int, quantity: int): boolean + decreaseInventory(id: int, quantity: int): boolean + updatePrice(id: int, buyingPrice: int, sellingPrice: int): boolean

InventoryItem
- product: Product - quantity: int - lastUpdated: Date
+ increaseInventory(quantity: int): boolean + decreaseInventory(quantity: int): boolean

1.3 Xác định tính kết tập của hệ thống

- Lớp Customer sẽ được kết tập bởi 1 object Cart. Tức là 1 khách hàng sẽ tương ứng với 1 giỏ hàng.
- Lớp Manager được kết tập bởi danh sách các Employee (các nhân viên dưới quyền) và danh sách các đơn hàng (để quản lý đơn hàng). Đây là mối quan hệ “has a”: Manager có một danh sách nhân viên dưới quyền và có một danh sách các đơn hàng bán ra.

- Employee (nhân viên) có thể tồn tại độc lập với Manager. Khi một Manager bị xóa, các Employee vẫn tiếp tục tồn tại và có thể được gán cho một Manager khác.
 - Order (đơn hàng) có thể tồn tại độc lập với Manager. Đơn hàng vẫn còn trong hệ thống ngay cả khi Manager không còn quản lý chúng nữa.
 - Manager không "sở hữu hoàn toàn" các Employee. Các nhân viên là đối tượng riêng biệt có thể thuộc về nhiều đơn vị khác trong tổ chức.
 - Manager không "sở hữu hoàn toàn" các Order. Đơn hàng có thể được xem, xử lý hoặc thậm chí chuyển cho người quản lý khác.
- Order được kết tập bởi danh sách các OrderDetail. Vì trong một đơn hàng, sẽ có nhiều sản phẩm, mỗi sản phẩm lại có số lượng khác nhau. OrderDetail chỉ chứa thông tin 1 loại sản phẩm và số lượng sản phẩm số trong Order.
 - Cart được kết tập bởi danh sách CartItem, Warehouse được kết tập bởi danh sách InventoryItem. Lý do tương tự như Order và OrderDetail. CartItem chứa thông tin của 1 loại sản phẩm và số lượng sản phẩm đó trong giỏ hàng. InventoryItem chứa thông tin của 1 loại sản phẩm và số lượng sản phẩm đó trong kho.
 - Ví dụ:

```

1 package main_class;
2
3 public class test {
4     public void main(String[] args){
5         Product book1 = new Book(1, "Sách lập trình Java", 150000);
6         Product book2 = new Book(2, "Clean Code", 200000);
7         Product pen = new Stationery(3, "Bút bi", 5000);
8         Product notebook = new Stationery(4, "Sổ ghi chép", 25000);
9         Customer cus1 = new Customer();
10        Cart cart = new Cart(cartID:"cart1", cus1);
11        CartItem newItem1 = new CartItem(book1, quantity:2);
12        cart.addCartItem(newItem1);
13        CartItem newItem2 = new CartItem(book2, quantity:2);
14        cart.addCartItem(newItem2);
15        CartItem newItem3 = new CartItem(pen, quantity:2);
16        cart.addCartItem(newItem3);
17        CartItem newItem4 = new CartItem(notebook, quantity:2);
18        cart.addCartItem(newItem4);
19    }
20 }

```

1.4 Xác định tính kế thừa của hệ thống

- Các lớp Book, Toy, Stationery là kế thừa của Product. Tức là các object của các lớp này cũng là các Product.
- Các lớp Customer, Employee, Manager là kế thừa của lớp Person vì các lớp này đều thuộc loại là con người, có các thông tin cá nhân cơ bản.
- Ví dụ :

```
1  package main_class;
2
3  public class Book extends Product{
4      private String isbn;
5      private String author;
6      private String publisher;
7      public Book(String isbn, String author, String publisher) {
8          this.isbn = isbn;
9          this.author = author;
10         this.publisher = publisher;
11     }
12 }
```