

BÁO CÁO THỰC HÀNH LAB 2

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Table of Contents

Table of Figures	1
1. Bài toán đặt ra	2
2. Yêu cầu hệ thống.....	2
2.1. Đối với Customer:.....	2
2.2. Đối với Store Manager:.....	2
3. Use Case Diagram.....	3
4. Class Diagram	4
5. Source Code	5
5.1. Aims Class	5
5.2. DigitalVideoDisc Class	6
5.3. Cart Class	8
6. Kết quả demo	10
7. Reading Assignment	11
8. Answer the question	11

Table of Figures

Figure 1: Use Case Diagram	3
Figure 2: Class Diagram.....	4
Figure 3: Aims Class	5
Figure 4: DigitalVideoDisc Class 1.....	6
Figure 5: Cart Class.....	8
Figure 6: Code Demo	10
Figure 7: Result	11
Figure 8: Getter and Setter Methods	12

1. Bài toán đặt ra

Thiết kế hệ thống mới cho dự án AIMS (Hiện tại chỉ có 1 loại phương tiện DVD)

2. Yêu cầu hệ thống

2.1. Đối với Customer:

Có thể sẽ có một tương lai mà Tiki và Sendo sẽ thảo luận về một cuộc sáp nhập tiềm năng để cạnh tranh với các nền tảng thương mại điện tử khác, đặc biệt là những nền tảng có nhà đầu tư nước ngoài. Cuộc sáp nhập giữa hai công ty này sẽ tạo ra một công ty Ti-do, trong đó “Ti” là từ Tiki, và “do” là từ Sendo, có nghĩa là công ty trị giá tỷ đô la tại Việt Nam. Công ty Ti-do muốn bạn giúp họ tạo ra một hệ thống mới cho dự án AIMS (AIMS viết tắt cho An Internet Media Store). Hiện tại, chỉ có một loại phương tiện: Đĩa Video Kỹ thuật số (DVD).

Khách hàng có thể duyệt danh sách các DVD có sẵn trong cửa hàng, thứ tự hiển thị dựa trên ngày thêm vào, từ mới nhất đến cũ nhất. Khi một khách hàng muốn tìm kiếm DVD để thêm vào giỏ hàng, họ có thể chọn một trong ba tùy chọn tìm kiếm. Phần mềm sẽ hiển thị danh sách tất cả các DVD phù hợp (DVD mới nhất trước) cùng với tất cả thông tin của chúng. Họ cũng có thể chọn để phát một DVD cụ thể. Phần mềm sẽ phát một DVD (một phần demo). Nếu một DVD có độ dài 0 hoặc thấp hơn, hệ thống sẽ thông báo cho khách hàng rằng DVD không thể được phát.

- Khi một khách hàng tìm kiếm DVD theo tiêu đề, họ cung cấp một chuỗi từ khóa. Nếu bất kỳ DVD nào có tiêu đề chứa bất kỳ từ nào trong chuỗi từ khóa, nó sẽ được tính là một sự khớp. Lưu ý rằng việc so sánh từ ở đây không phân biệt chữ hoa chữ thường.
- Khi một khách hàng tìm kiếm DVD theo thể loại, họ cung cấp tên thể loại. Nếu bất kỳ DVD nào có thể loại khớp (không phân biệt chữ hoa chữ thường), nó sẽ được tính là một sự khớp.
- Khi một khách hàng tìm kiếm DVD theo giá, họ cung cấp hoặc mức giá tối thiểu và tối đa, hoặc chỉ mức giá tối đa.

Khách hàng có thể xem thông tin chi tiết của một DVD từ danh sách các DVD. Họ có thể thêm một DVD vào giỏ hàng từ danh sách các DVD hoặc từ màn hình chi tiết.

Khi một khách hàng muốn xem giỏ hàng hiện tại, hệ thống sẽ hiển thị tất cả thông

tin của các DVD, cùng với tổng chi phí. Khách hàng có thể nghe một DVD (một phần demo) trong giỏ hàng trước khi xác nhận để đặt hàng. Khách hàng có thể sắp xếp tất cả các DVD trong giỏ hàng theo tiêu đề hoặc theo chi phí:

- Sắp xếp theo tiêu đề: hệ thống hiển thị tất cả các DVD theo thứ tự chữ cái theo tiêu đề. Trong trường hợp có cùng tiêu đề, các DVD có chi phí cao hơn sẽ được hiển thị trước.
- Sắp xếp theo chi phí: hệ thống sẽ hiển thị tất cả các DVD theo thứ tự giảm dần của chi phí. Trong trường hợp có cùng chi phí, các DVD sẽ được sắp xếp theo thứ tự tăng dần của tiêu đề.

Khách hàng có thể cập nhật số lượng của một DVD trong giỏ hàng hoặc xóa một DVD khỏi giỏ hàng. Để tăng nhu cầu tiêu dùng cho sản phẩm và tăng doanh số bán hàng, khách hàng được phép có một món hàng miễn phí được hệ thống chọn ngẫu nhiên trong giỏ hàng. Khách hàng có thể lọc DVD trong giỏ hàng bằng cách cung cấp ID hoặc tiêu đề của nó. Nếu món hàng được tìm thấy, thông tin của món hàng đó sẽ được hiển thị trong giỏ hàng. Nếu không, hệ thống sẽ thông báo cho khách hàng rằng món hàng không được tìm thấy trong giỏ hàng hiện tại.

Khách hàng có thể yêu cầu đặt hàng khi họ đang xem giỏ hàng hiện tại. Để đơn giản hóa, họ không cần đăng nhập để đặt hàng. Ứng dụng sẽ nhắc khách hàng nhập thông tin giao hàng và hướng dẫn giao hàng. Phần mềm sẽ tính toán phí giao hàng dựa trên tổng khối lượng của đơn hàng và địa điểm giao hàng. Sau đó, nó sẽ hiển thị cho khách hàng hóa đơn bao gồm danh sách DVD, tổng chi phí trước VAT, tổng chi phí sau VAT và phí giao hàng. Khách hàng sau đó có thể tiến hành thanh toán cho đơn hàng. Hiện tại, chỉ có một phương thức thanh toán – tức là thẻ tín dụng – được phép bằng cách kết nối với hệ thống hiệp hội thẻ để kiểm tra tính hợp lệ của thẻ hoặc thực hiện giao dịch thanh toán. Sau khi giao dịch, phần mềm AIMS sẽ hiển thị tất cả thông tin chi tiết như ID giao dịch, chủ thẻ, số tiền giao dịch, thông điệp giao dịch, số dư, ngày giao dịch cho khách hàng. Đơn hàng sẽ ở trạng thái đang chờ và thông tin của đơn hàng và giao dịch sẽ được gửi đến email của khách hàng.

2.2. Đối với Store Manager:

Người quản lý cửa hàng cần đăng nhập vào hệ thống để truy cập vào chế độ quản lý. Họ có thể thấy danh sách các đơn hàng đang chờ xử lý, sau đó có thể chọn bất kỳ đơn hàng nào để xem chi tiết và phê duyệt hoặc từ chối đơn hàng đó. Người quản lý cửa hàng có thể thêm DVD mới vào cửa hàng. Họ phải cung cấp tất cả thông tin của

DVD mới, bao gồm ID, tiêu đề, thể loại, đạo diễn, độ dài và chi phí. Ngoài ra, người quản lý cũng có thể xóa DVD khỏi cửa hàng.

3. Use Case Diagram

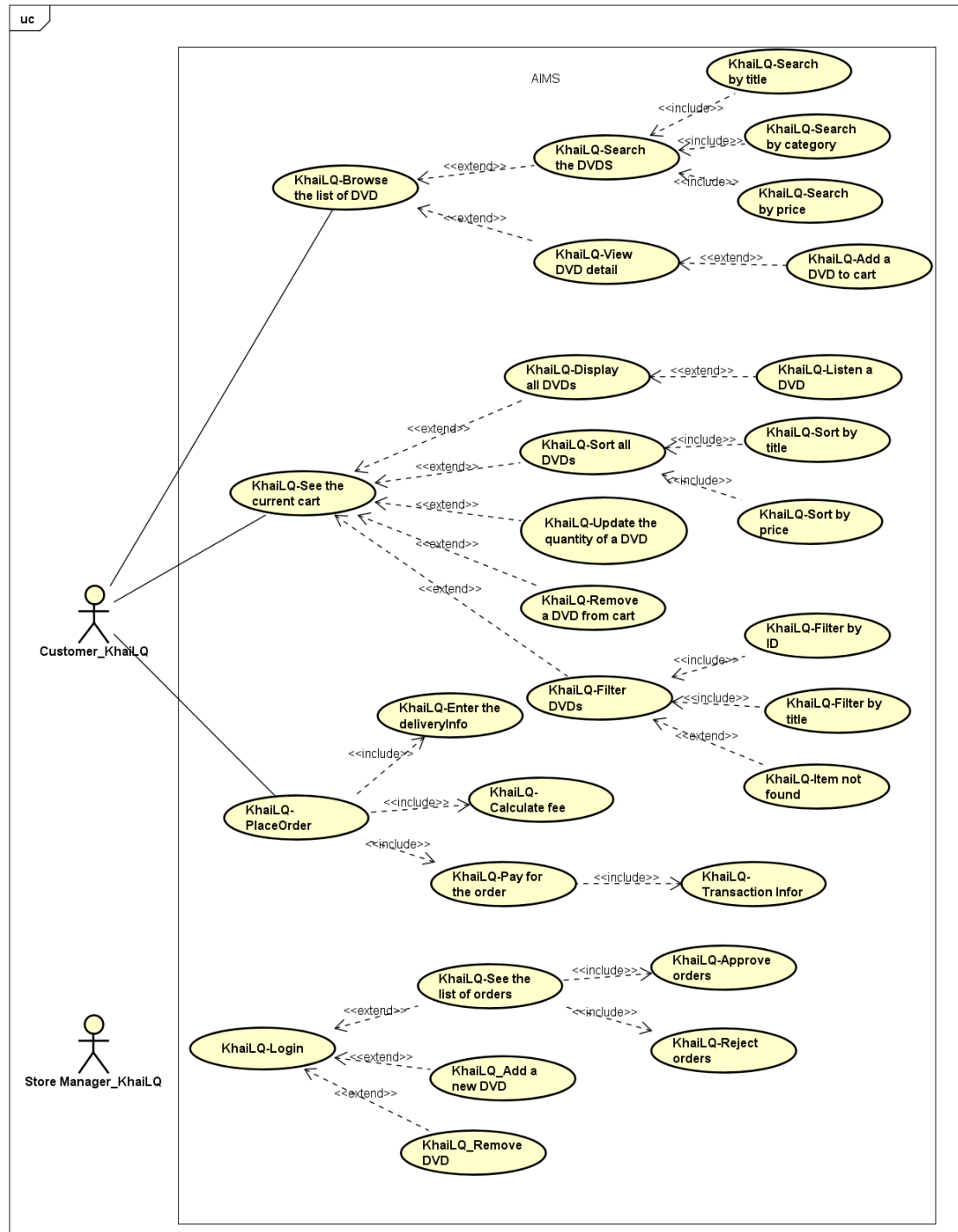


Figure 1: Use Case Diagram

4. Class Diagram

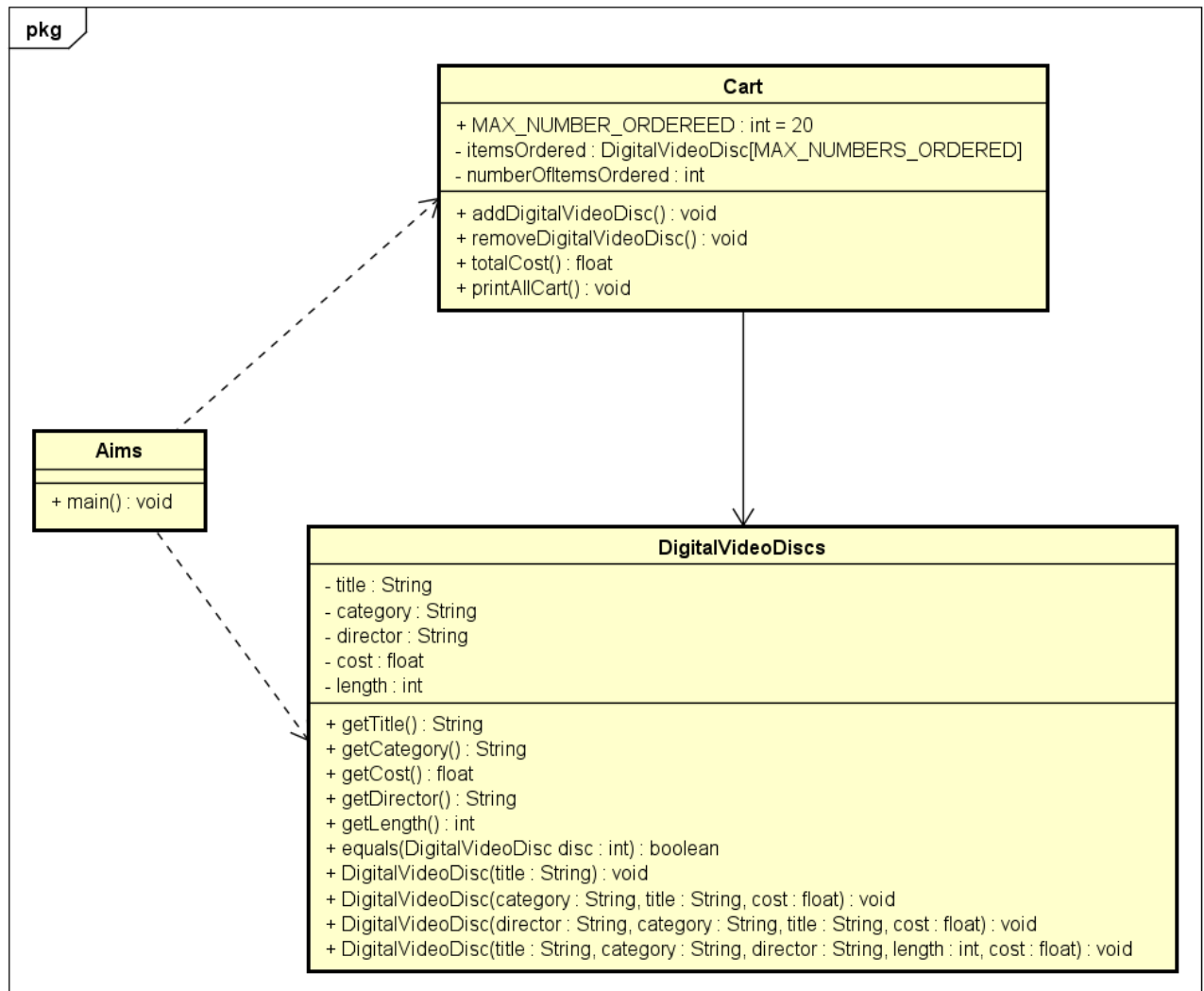


Figure 2: Class Diagram

5. Source Code

5.1. Aims Class

```
public class Aims {  
    public static void main(String[] args) {  
        System.out.println("Le Quang Khai - 20225638");  
        Cart newOrder = new Cart();  
  
        // Create new dvd objects and add them to the cart  
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "Your Name", category: "Animation",  
            director: "Shinkai Makoto", length: 110, cost: 19.95f);  
        newOrder.addDigitalVideoDisc(dvd1);  
  
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Avengers: End Game", category: "Science Fiction",  
            director: "Joe Russo", length: 180, cost: 24.95f);  
        newOrder.addDigitalVideoDisc(dvd2);  
  
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( director: "Inside Out 2",  
            category: "Animation", title: "Kelsey Mann", cost: 19.99f);  
        newOrder.addDigitalVideoDisc(dvd3);  
  
        // print cart after add 3 disc  
        System.out.println("LQK - 20225638 - Print cart after add 3 disc");  
        newOrder.printAllDisc();  
        // Remove the Disc "Your Name"  
        newOrder.removeDigitalVideoDisc(dvd1);  
        // print cart after remove disc "Your Name"  
        System.out.println("LQK - 20225638 - Print cart after remove disc Your Name");  
        newOrder.printAllDisc();  
    }  
}
```

Figure 3: Aims Class

5.2. DigitalVideoDisc Class

```

public class DigitalVideoDisc { 11 usages
    private String title; 5 usages
    private String category; 4 usages
    private String director; 3 usages
    private int length; 2 usages
    private float cost; 4 usages

    > public String getTitle() { return title; }
    > public String getCategory() { return category; }
    > public String getDirector() { return director; }
    > public int getLength() { return length; }
    > public float getCost() { return cost; }
    > public DigitalVideoDisc(String title) { this.title = title; }
    ✓ public DigitalVideoDisc(String category, String title, float cost) { no usages
        this.category = category;
        this.title = title;
        this.cost = cost;
    }

    ✓ public DigitalVideoDisc(String director, String category, String title, float cost) { 1 usage
        this.director = director;
        this.category = category;
        this.title = title;
        this.cost = cost;
    }

    ✓ public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 2 usages
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
    }

    > public boolean equals(DigitalVideoDisc disc) { return this.getTitle() == disc.getTitle(); }
    }

```

Figure 4: DigitalVideoDisc Class

5.3. Cart Class

```

public class Cart { 2 usages
    public static final int MAX_NUMBERS_ORDERED = 20; 2 usages
    private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED]; 7 usages
    public int numberOfItemsOrdered = 0; 12 usages
    // thêm 1 đĩa mới vào cart
    public void addDigitalVideoDisc(DigitalVideoDisc disc) { 3 usages
        if(this.numberOfItemsOrdered == 0) {
            this.itemsOrdered[0] = disc;
            this.numberOfItemsOrdered++;
            System.out.println("Le Quang Khai - 20225638 - The disc: " + disc.getTitle() + " has been added");
        }
        else if(this.numberOfItemsOrdered >= MAX_NUMBERS_ORDERED) {
            System.out.println("Le Quang Khai - 20225638 - The cart is full");
        }
        else{
            this.itemsOrdered[this.numberOfItemsOrdered] = disc;
            this.numberOfItemsOrdered++;
            System.out.println("Le Quang Khai - 20225638 - The disc: " + disc.getTitle() + " has been added");
        }
    }

    // xoá 1 đĩa trong cart
    public void removeDigitalVideoDisc(DigitalVideoDisc disc) { 1 usage
        if (this.numberOfItemsOrdered == 0) {
            System.out.println("Le Quang Khai - 20225638 - The cart is empty now so cannot remove any discs");
        } else {
            // xoá 1 đĩa trong cart
            public void removeDigitalVideoDisc(DigitalVideoDisc disc) { 1 usage
                if (this.numberOfItemsOrdered == 0) {
                    System.out.println("Le Quang Khai - 20225638 - The cart is empty now so cannot remove any discs");
                } else {
                    // Kiểm tra disc có tồn tại k
                    for (int i = 0; i < this.numberOfItemsOrdered; ++i) {
                        // nếu tồn tại thì xoá và in tbao
                        if (this.itemsOrdered[i].equals(disc)) {
                            for (int j = i; j < this.numberOfItemsOrdered; ++j) {
                                this.itemsOrdered[j] = this.itemsOrdered[j + 1];
                            }
                            this.numberOfItemsOrdered--;
                            System.out.println("The disc " + disc.getTitle() + " has been removed from the cart");
                            break;
                        }
                    }
                    if(i==this.numberOfItemsOrdered-1) {
                        //Neu khong co dia can xoa thi in ra thong bao
                        System.out.println("The disc " + disc.getTitle() + " doesn't exist in the cart");
                    }
                }
            }
        }
    }
}

```



```
// Tính tổng giá trị cart
public float totalCost() { 1 usage
    float cost = 0f;
    for (int i = 0; i < this.numberOfItemsOrdered; ++i) {
        cost += this.itemsOrdered[i].getCost();
    }
    return cost;
}

// In ra toàn bộ thông tin về cart gồm các địa trong cart và tổng tiền
public void printAllDisc() { 2 usages
    System.out.println("Le Quang Khai - 20225638 ");
    System.out.println("\n-----CART-----");
    System.out.println("Ordered Items:");
    for (int i = 0; i < this.numberOfItemsOrdered ; i++) {
        System.out.println((i + 1) + ". " + this.itemsOrdered[i].getTitle());
    }
    System.out.println("Total cost: " + totalCost() + " $");
    System.out.println("-----");
}
}
```

Figure 5: Cart Class

6. Kết quả demo

```
Le Quang Khai - 20225638
Le Quang Khai - 20225638 - The disc: Your Name has been added
Le Quang Khai - 20225638 - The disc: Avengers: End Game has been added
Le Quang Khai - 20225638 - The disc: Kelsey Mann has been added
LQK - 20225638 - Print cart after add 3 disc
Le Quang Khai - 20225638

-----CART-----
Ordered Items:
1. Your Name
2. Avengers: End Game
3. Kelsey Mann
Total cost: 64.89 $

-----
The disc Your Name has been removed from the cart
LQK - 20225638 - Print cart after remove disc Your Name
Le Quang Khai - 20225638

-----CART-----
Ordered Items:
1. Avengers: End Game
2. Kelsey Mann
Total cost: 44.940002 $
```

Figure 6: Code Demo

7. Reading Assignment

8. Answer the question

- Khi tạo một đối tượng DigitalVideoDisc với một tham số kiểu String, Java sẽ không biết mình muốn sử dụng constructor nào, vì cả hai đều nhận một tham số kiểu String. Điều này dẫn đến **sự mơ hồ** và Java sẽ báo lỗi khi biên dịch do không thể xác định constructor nào sẽ được gọi.

```
public DigitalVideoDisc(String title) { this.title = title; }
```

```
public DigitalVideoDisc(String category) { this.category = category; }
```

'DigitalVideoDisc(String)' is already defined in 'DigitalVideoDisc'