

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



BÁO CÁO THỰC HÀNH TUẦN 2
Lập trình hướng đối tượng

Giảng viên hướng dẫn : Lê Thị Hoa
Sinh viên thực hiện : Nguyễn Thị Minh Châu
Mã số sinh viên : 20214997
Lớp : Việt Nhật 04- K66

Mục lục

1 Đề bài.....	3
2 Yêu cầu bài toán.....	3
3 Use case diagram.....	3
4 Class diagram.....	6
5 Mã nguồn của chương trình	6
5.1 Create Aims class	6
5.2 Create the DigitalVideoDisc class and its attributes	7
5.3 Create accessors and mutators for the class DigitalVideoDisc	7
5.4 Create Constructor method	8
5.5 Create the Cart class to work with DigitalVideoDisc.....	8
5.6 Create Method to support the class Cart.....	8
5.6.1 Create the method addDigitalVideoDisc(DigitalVideoDisc disc).....	8
5.6.2 Create the method removeDigitalVideoDisc(DigitalVideoDisc disc).....	9
5.6.3 Create the method totalCost()	9
6 Demo chương trình	9
6.1 Source code.....	9
6.2 The result	10

1 Đề bài

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

2 Yêu cầu bài toán

There might be a future that Tiki and Sendo be in talks over a potential merger to contend other e-commerce platforms and especially those who have foreign backers. The merger of these two firms would create a Ti-do company, where “Ti” is from Tiki, and “do” is from Sendo, which means a billion-dollar company in Vietnamese. That firm, Ti-do company, would like you to help them create a brand-new system for AIMS project (AIMS stands for An Internet Media Store). Currently, there is only one type of media: Digital Video Disc (DVD).

Customers can browse the list of DVDs available in the store, the display order is based on their added date, from latest to oldest. When a customer wants to search for DVDs to add to cart, he or she can choose one of three searching options. The software will display a list of all matches (latest DVDs first) with all their information. He or she can also choose to play a specific DVD. The software will play a DVD (a demo part). If a DVD has the length 0 or less, the system must notify the customer that the DVD cannot be played.

- When a customer searches for DVDs by title, he or she provides a string of keywords. If any DVD has the title containing any word in the string of keywords, it is counted as a match. Note that the comparison of words here is case-insensitive.
- When a customer searches for DVDs by category, he or she provides the category name. If any DVD has the matching category (case-insensitive), it is counted as a match.
- When a customer searches for DVDs by price, he or she provides either the minimum and maximum cost, or just the maximum cost.

Customers can view the detail information of a DVD from the list of DVDs. He/she can add a DVD to a cart from a list of DVDs or the detail screen.

When a customer wants to see the current cart, the system displays all the information of the DVDs, along with the total cost. Customers may listen to a DVD (a demo part) in the cart before confirming to place an order. Customers can sort all DVDs in the cart by title or by cost:

- Sort by title: the system displays all the DVDs in the alphabet sequence by title. In case they have the same title, the DVDs having the higher cost will be displayed first.
- Sort by cost: the system the system displays all the DVDs in decreasing cost order. In case they have the same cost, the DVDs will be ordered by increasing title.

Customers can update the quantity of a DVD in a cart or remove a DVD from a cart. To increase consumer demand for the product and grow sales, customers are allowed to have an item for free which is randomly picked out in the cart by the system. Customers can filter DVDs in the cart by providing either its ID or title. If the item is found, display information of the found item in the cart. Or else, notify the customer the item is not found in the current cart.

The customer can request to place order when they are seeing the current cart. For simplification, he/she does not need to log in to place an order. The application will prompt the customer to

enter the delivery information and delivery instructions. The software will then calculate the delivery fee based on the total mass of the order & the delivery location. Then, it will display to the customer the invoice including the DVD list, total cost before VAT, total cost after VAT, and the delivery fee. The customer can then proceed to pay for the order. Currently, only one payment method – i.e. credit card – is allowed by connecting to a card association system for checking the validation of the card or performing the pay transaction. After the transaction, the AIMS software will display all the detailed information such as transaction ID, card owner, transaction amount, transaction message, balance, transaction date to the customer. The order will be in pending state and the information of the order & the transaction will be sent to the customer's email.

The store manager needs to log in to the system to navigate to the management mode. He/she can see the list of pending orders, then can pick any order to see its detail to approve or reject the order. The store manager can add new DVDs to the store. He or she must provide all information of the new DVD, including its ID, title, category, director, length, and the cost. Additionally, the manager can also remove DVDs from the store.

3 Use case diagram

Từ yêu cầu của đề bài đưa ra, chúng ta sẽ phân tích để tạo biểu đồ use case như sau:

+, Xác định tác nhân (actor) sử dụng trong hệ thống: ở đây như trên đề bài sẽ có 2 tác nhân sẽ sử dụng đến hệ thống đó là khách hàng và quản lý. Vì vậy ta sẽ tạo ra 2 actors đó là Actor0 và StoreManger

+, Xác định các chức năng (use case) của hệ thống cung cấp cho tác nhân:

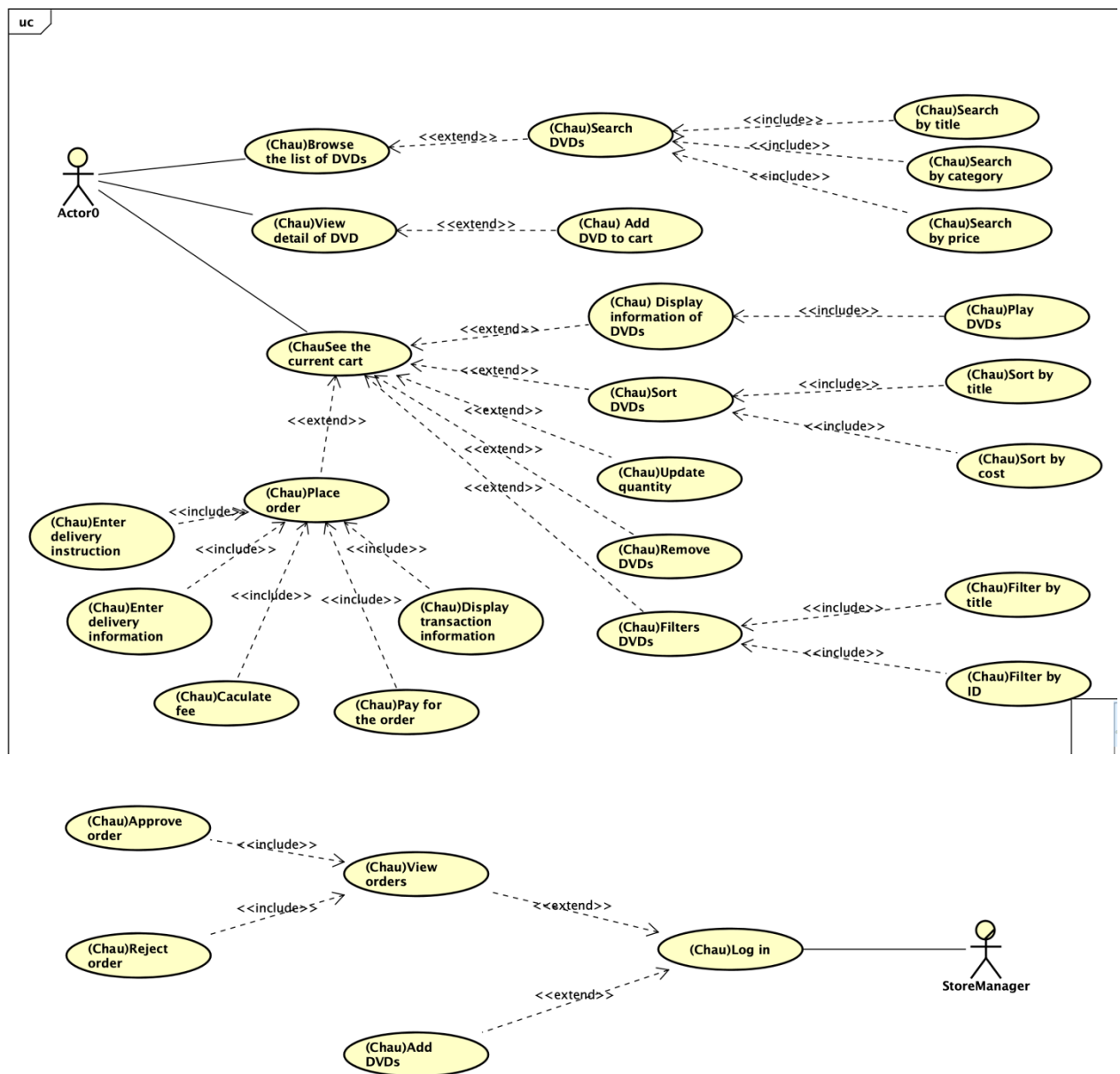
Đối với tác nhân Actor0:

- , Tác nhân này có thể xem danh sách các DVDs có sẵn nên chúng ta sẽ tạo ra 1 use case “Browse the list of DVDs”. Trong use case này, có thể mở rộng chức năng tìm kiếm DVDs bằng tiêu đề, thể loại hoặc là theo mức giá.
- , Tiếp theo, người dùng có thể xem chi tiết DVD từ list có sẵn để có quyết định có mua hay không. Do đó tạo thêm một use case “View detail information of DVD”. Sau khi xem, có thể mở rộng thêm chức năng thêm vào giỏ hàng
- , Người dùng này có thể xem giỏ hàng hiện thời nên thêm một use case “See curent cart”. Trong quá trình xem giỏ hàng hiện tại, người dùng có thể lựa chọn phát demo một bản DVD, hay là thêm bớt số lượng sản phẩm, hoặc là xóa sản phẩm ra khỏi giỏ hàng. Do đó tạo thêm các use case thực hiện chức năng đó. Thêm nữa, nếu đã hài lòng, người dùng có thể thanh toán luôn giỏ hàng đó.

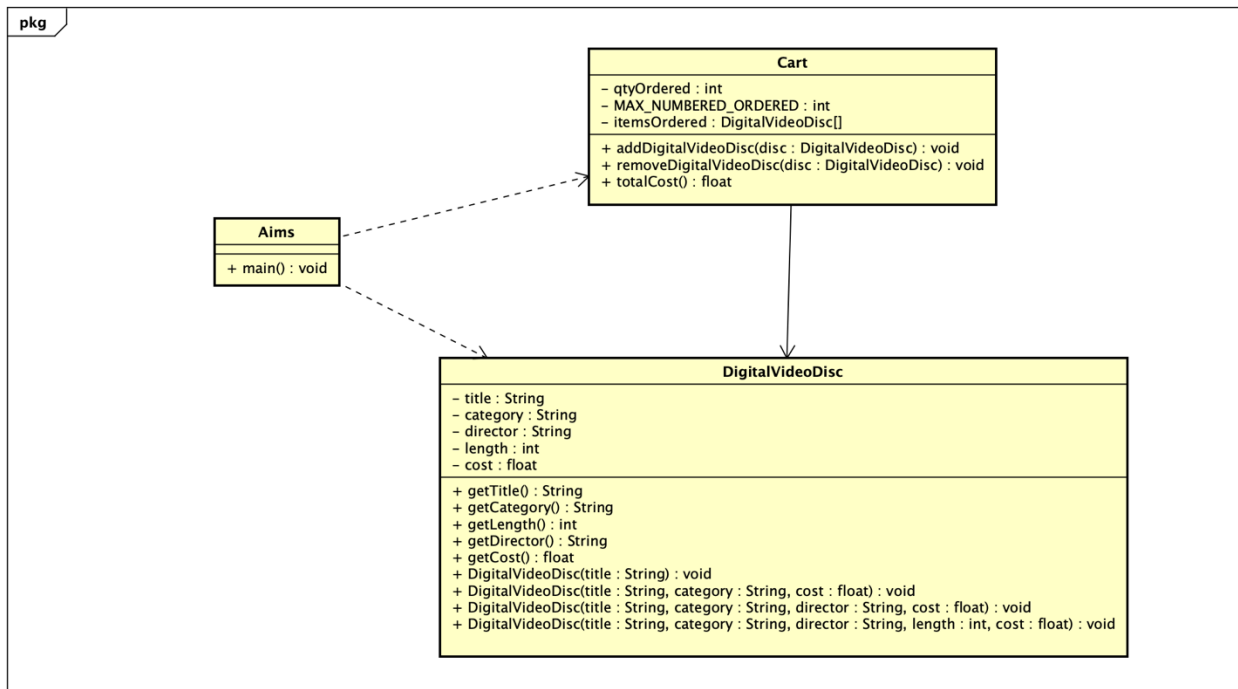
Đối với tác nhân Manager:

- , Tác nhân này có thể đăng nhập vào hệ thống để có thể quản lý cửa hàng. Do đó xây dựng use case “Login”. Trong quá trình login, tác nhân có thể xem danh sách các lệnh đang chờ xử lý, sau đó có thể chọn bất kỳ lệnh nào để xem chi tiết của lệnh đó để phê duyệt hoặc từ chối lệnh. Tác nhân có thể thêm DVD mới vào cửa hàng và cũng có thể xóa DVD khỏi cửa hàng. Từ đó ta thêm các use case tương ứng với các chức năng cần xây dựng

Ta có biểu đồ use case như hình dưới.



4 Class diagram



5 Mã nguồn của chương trình

5.1 Create Aims class

```

Lab02 > J Aims.java > Aims > main(String[])
1 | //Create Aims class
2 | public class Aims {
   | Run | Debug
3 | public static void main(String[] args) {
4 | | //TODO Auto-generated method stub
5 | |
  
```

5.2 Create the DigitalVideoDisc class and its attributes

```
Lab02 > J DigitalVideoDisc.java > DigitalVideoDisc
1 //create DigitalVideoDisc class
2 public class DigitalVideoDisc {
3     private String title;
4     private String category;
5     private String director;
6     private int length;
7     private float cost;
```

5.3 Create accessors and mutators for the class DigitalVideoDisc

```
Lab02 > J DigitalVideoDisc.java > DigitalVideoDisc
12 }
13 public String getTitle() {
14     return title;
15 }
16 public String getCategory() {
17     return category;
18 }
19 public String getDirector() {
20     return director;
21 }
22 public int getLength() {
23     return length;
24 }
25 public float getCost() {
26     return cost;
27 }
```

5.4 Create Constructor method

```
public DigitalVideoDisc(String title){
    this.id =Update();
    this.title = title;
}
public DigitalVideoDisc(String title , String category, float cost){
    this(title);
    this.category = category;
    this.cost = cost;
}
public DigitalVideoDisc(String title, String category, String director , float cost){
    this(title, category, cost);
    this.director = director;
}
public DigitalVideoDisc(String title, String category, String director , int length, float cost){
    this(title, category, director, cost);
    this.length = length;
}
```

5.5 Create the Cart class to work with DigitalVideoDisc

```
Lab02 > J Cart.java > Cart > itemOrdered
1 //create Cart class
2 public class Cart {
3     public static final int MAX_NUMBERS_ORDERED = 20;
4     //create a field as an array to store a list of DigitalVideoDisc
5     private DigitalVideoDisc itemOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
6     //create a field which stores how many DigitalVideoDiscs are in the cart
7     private int qtyOrdered = 0;
```

5.6 Create Method to support the class Cart

5.6.1 Create the method addDigitalVideoDisc(DigitalVideoDisc disc)

```
public void addDigitalVideoDisc(DigitalVideoDisc disc){
    if (qtyOrdered <= MAX_NUMBERS_ORDERED){
        itemOrdered[qtyOrdered] = disc;
        qtyOrdered++;
        System.out.println("The disc with name " + disc.getTitle() + " has been added");
    }
    else {
        System.out.println("The cart is almost full");
    }
}
```


5.6.2 Create the method removeDigitalVideoDisc(DigitalVideoDisc disc)

```
public void removeDigitalVideoDisc(DigitalVideoDisc disc){
    for (int i = 0 ; i < qtyOrdered ; i++){
        if (itemOrdered[i].equals(disc)){
            index = i;
        }
    }
    if ( index == -1 ){
        System.out.println(x: "Not found DVD in cart");
        return;
    }
    DigitalVideoDisc newArray[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
    System.arraycopy(itemOrdered, srcPos: 0 , newArray, destPos: 0 , index);
    System.arraycopy(itemOrdered, index+1 , newArray, index, qtyOrdered - index - 1);
    itemOrdered = newArray;
    qtyOrdered --;
    System.out.println(x: "The disc has been removed");
}
```

5.6.3 Create the method totalCost()

```
public void totalCost(){
    float sum = 0;
    for (int i = 0 ; i < qtyOrdered ; i++){
        System.out.println(itemOrdered[i].getId() + "\t" +
            itemOrdered[i].getTitle() + "\t" + itemOrdered[i].getCost());
        sum += itemOrdered[i].getCost();
    }
    System.out.println("Total cost : " + sum);
}
```

6 Demo chương trình

6.1 Source code

```
1 public class Aims {
2     Run | Debug
3     public static void main(String[] args) {
4         Cart anOrder = new Cart();
5         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title: "The Lion King", category: "Animation",
6             director: "Roger Allers", length: 87, cost: 19.95f);
7         anOrder.addDigitalVideoDisc(dvd1);
8
9         DigitalVideoDisc dvd2 = new DigitalVideoDisc(title: "Star Wars", category: "Science Fiction",
10             director: "George Lucas", length: 87, cost: 24.95f);
11         anOrder.addDigitalVideoDisc(dvd2);
12
13         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title: "Aladin", category: "Animation", cost: 18.99f);
14         anOrder.addDigitalVideoDisc(dvd3);
15
16         DigitalVideoDisc dvd4 = new DigitalVideoDisc(title: "Your name", category: "Romance", cost: 20.21f);
17         anOrder.addDigitalVideoDisc(dvd4);
18         anOrder.totalCost();
19     }
20 }
21
```

6.2 The result

```
The disc with name The Lion King has been added
The disc with name Star Wars has been added
The disc with name Aladin has been added
The disc with name Your name has been added
1      The Lion King    19.95
2      Star Wars       24.95
3      Aladin  18.99
4      Your name       20.21
Total cost : 84.1
```