BÁO CÁO THỰC HÀNH LAB 4

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

I. Mã nguồn chương trình: 3
1. Create Book class: 3
2. Create the abstract Media Class: 5
3. Creating CompactDisc class: 8
1. Create the Disc class extending the Media Class: 8
2. Create Track Class: 10
3. CompactDisc Class: 10
4. Create the Playable interface: 11
5. Update the Cart Class to work with Media: 13
6. Unique item in a list 15
7. Polymorphism with toString() method 16
8. Sort Media in the cart 17
9. Create a complete console application in the Aims Class 19
10. Update Class diagram 31
11. Update user diagram 32

Table of Figures

Figure 1: Book Class 1 3
Figure 2: Book Class 2 4
Figure 3: Book Class 3 4
Figure 4: Cập nhật DigitalVideoDisc 1 5
Figure 5: Cập nhật DigitalVideoDisc 2 6
Figure 6: Cập nhật Book Class 6
Figure 7: Media Class 2 7
Figure 8: Media Class 2 7
Figure 9: Disc Class 8
Figure 10: Cập nhật DigitalVideoDisc Class 9
Figure 11: Track Class 10
Figure 12: CompactDisc Class 10
Figure 13: CompactDisc Class 2 11
Figure 14: Interface Playable 11
Figure 15: Cập nhật Track Class 12
Figure 16: Cập nhật DigitalVideoDisc Class 12

Figure 17: Cập nhật CompactDisc Class 12
Figure 18: Cập nhật Cart Class 1 13
Figure 19: Cập nhật Cart Class 2 14
Figure 20: Cập nhật Cart Class 3 15
Figure 21: Overriding equals() Media Class 15
Figure 22: Overriding equals() Track Class 16
Figure 23: TestMedia Class 16
Figure 24: Kết quả TestMedia Class 16
Figure 25:MediaComparatorByCostTitle Class 17
Figure 26: MediaComparatorByTitleCost 17
Figure 27: Media Class 17
Figure 28: TestMedia Class 18
Figure 29: Kết quả TestMedia Class 18
Figure 30: Menu 30
Figure 31: Kết quả 30
Figure 32: Class Diagram 31
Figure 33: Customer Usecase Diagram 32
Figure 34: Manager Usecase Diagram 33

I. Mã nguồn chương trình:

1. Create Book class:

```java
package Lab_02.hust.soict.aims.media;

import java.util.ArrayList;
import java.util.List;

no usages
public class Book {
    2 usages
    private int id;
    2 usages
    private String title;
    2 usages
    private String category;
    2 usages
    private float cost;
    6 usages
    private List<String> authors = new ArrayList<String>();

    no usages
    public int getId() {
        return id;
    }

    no usages
    public String getTitle() {
        return title;
    }

    no usages
    public String getCategory() {
        return category;
    }

    no usages
    public float getCost() {
        return cost;
    }

    no usages
    private List<String> getAuthors() {
        return authors;
```

Figure 1: Book Class 1

```
    no usages
    private List<String> getAuthors() {
        return authors;
    }

    no usages
    public void setId(int id) {
        this.id = id;
    }

    no usages
    public void setTitle(String title) {
        this.title = title;
    }

    no usages
    public void setCategory(String category) {
        this.category = category;
    }

    no usages
    public void setCost(float cost) {
        this.cost = cost;
    }

    no usages
    private void setAuthors(List<String> authors) {
        this.authors = authors;
    }
```

Figure 2: Book Class 2

```
    no usages
    public void addAuthor(String authorName){
        boolean found = false;
        for(String author: authors){
            if(author.equals(authorName)){ //kiem tra ton tai author
                found = true;
            }
        }
        if(found){
            System.out.println("Author has already existed"); //In thong bao
        }
        else{
            authors.add(authorName); //Them author vao list
            System.out.println("Added new author");
        }
    }

    no usages
    public void removeAuthor(String authorName){
        boolean found = false;
        for(String author: authors){
            if(author.equals(authorName)){ //kiem tra ton tai author
                found = true;
            }
        }
        if(!found){
            System.out.println("Author do not exist"); //In thong bao
        }
        else{
            authors.remove(authorName); //Xoa author vao list
            System.out.println("Removed new author");
        }
    }
}
```

Figure 3: Book Class 3

2. Create the abstract Media Class:

```java
package Lab_02.hust.soict.aims.media;

44 usages  tuandattt *
public class DigitalVideoDisc extends Media{
    5 usages
    private String director;
    4 usages
    private int length;
    8 usages
    private static int nbDigitalVideoDiscs = 0; //Class member


    1 usage  tuandattt
    public String getDirector() { return director; }

    1 usage  tuandattt
    public int getLength() { return length; }

    no usages  tuandattt
    public void setDirector(String director) { this.director = director; }

    no usages  tuandattt
    public void setLength(int length) { this.length = length; }


    4 usages  tuandattt *
    public DigitalVideoDisc(String title) {
        nbDigitalVideoDiscs++;
        this.setTitle(title);
        this.setId(nbDigitalVideoDiscs);
    }

    4 usages  tuandattt *
    public DigitalVideoDisc(String title, String category, float cost) {
        nbDigitalVideoDiscs++;
        this.setTitle(title);
        this.setCategory(category);
        this.setCost(cost);
        this.setId(nbDigitalVideoDiscs);
    }
```

Figure 4: Cập nhật DigitalVideoDisc 1

```java
1 usage  tuandattt *
public DigitalVideoDisc(String title, String category, String director, float cost) {
    nbDigitalVideoDiscs++;
    this.setTitle(title);
    this.setCategory(category);
    this.director = director;
    this.setCost(cost);
    this.setId(nbDigitalVideoDiscs);
}

7 usages  tuandattt *
public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
    nbDigitalVideoDiscs++;
    this.setTitle(title);
    this.setCategory(category);
    this.director = director;
    this.length = length;
    this.setCost(cost);
    this.setId(nbDigitalVideoDiscs);
}

tuandattt *
public String toString(){
    return "DVD " + "- " + this.getTitle() + " - " + this.getCategory() + " - " + this.director + " - " + this.length +  ": " + this
}

1 usage  tuandattt *
public boolean isMatch(String title){
    if(title.equals(this.getTitle())) return true;
    else {
        return false;
    }
}
}
```

Figure 5: Cập nhật DigitalVideoDisc 2

```java
package Lab_02.hust.soict.aims.media;

import java.util.ArrayList;
import java.util.List;

no usages
public class Book extends Media{
    6 usages
    private List<String> authors = new ArrayList<String>();

    no usages
    private List<String> getAuthors() {
        return authors;
    }

    no usages
    private void setAuthors(List<String> authors) {
        this.authors = authors;
    }
```

Figure 6: Cập nhật Book Class

```java
package Lab_02.hust.soict.aims.media;

2 usages  2 inheritors
public abstract class Media {
    2 usages
    private int id;
    2 usages
    private String title;
    2 usages
    private String category;
    2 usages
    private float cost;

    1 usage
    public int getId() {
        return id;
    }

    18 usages
    public String getTitle() {
        return title;
    }

    2 usages
    public String getCategory() {
        return category;
    }

    3 usages
    public float getCost() {
        return cost;
    }
```

Figure 7: Media Class 2

```java
    4 usages
    public void setId(int id) {
        this.id = id;
    }

    7 usages
    public void setTitle(String title) {
        this.title = title;
    }

    3 usages
    public void setCategory(String category) {
        this.category = category;
    }

    3 usages
    public void setCost(float cost) {
        this.cost = cost;
    }

}
```

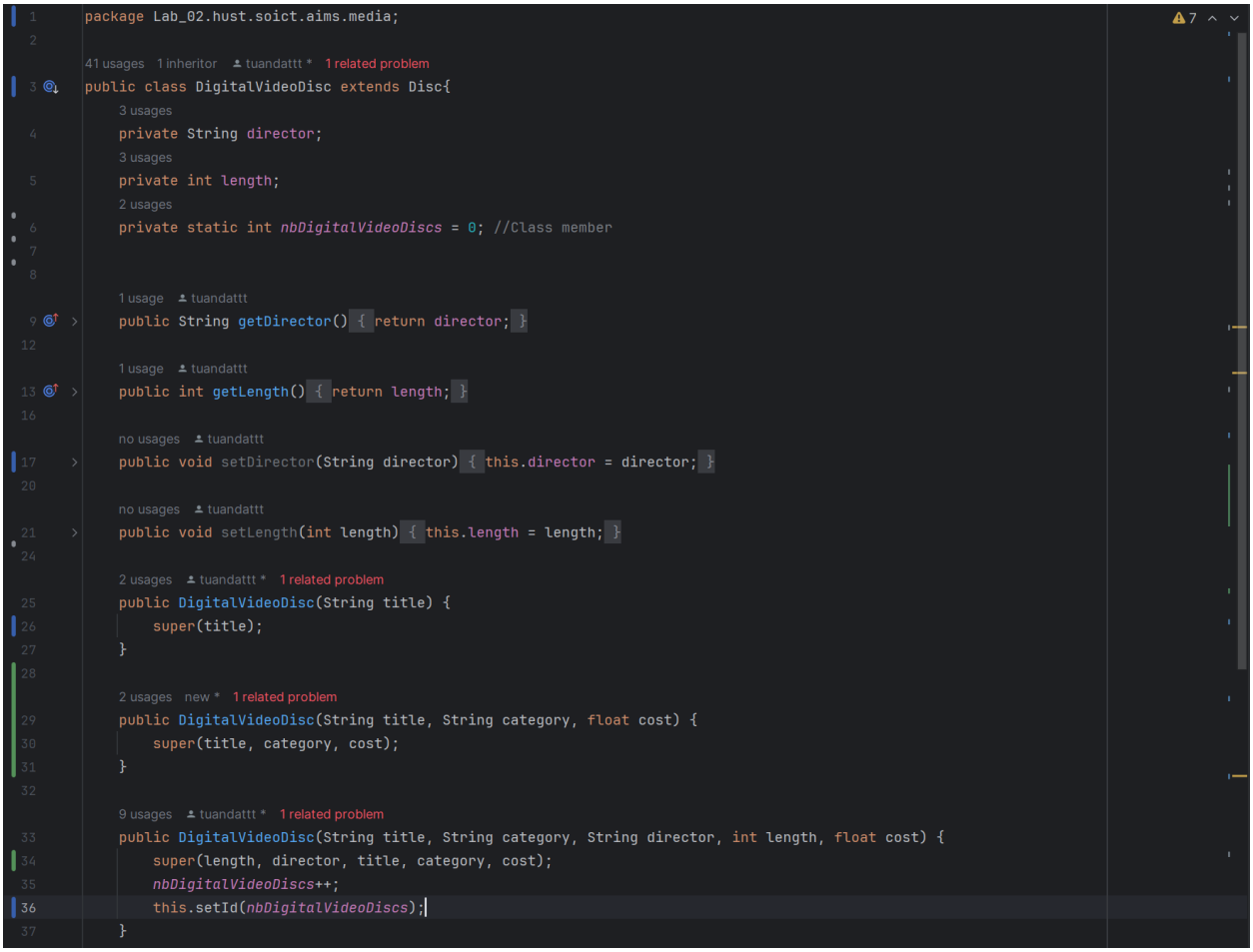
Figure 8: Media Class 2

3. Creating CompactDisc class:

1. Create the Disc class extending the Media Class:

```java
package Lab_02.hust.soict.aims.media;

1 usage  2 inheritors
public class Disc extends Media{
    2 usages
    private int length;
    2 usages
    private String director;

    1 usage  1 override
    public int getLength() {
        return length;
    }

    1 usage  1 override
    public String getDirector() {
        return director;
    }

    1 usage
    public Disc(int length, String director, String title, String category, float cost) {
        //lay tu lop cha
        super(title, category, cost);
        this.length = length;
        this.director = director;
    }

    1 usage
    public Disc(String title) {
        super(title);
    }

    1 usage
    public Disc(String title, String category, float cost) {
        super(title, category, cost);
    }
}
```

Figure 9: Disc Class



```java
package Lab_02.hust.soict.aims.media;

41 usages  1 inheritor  tuandattt *  1 related problem
public class DigitalVideoDisc extends Disc{
    3 usages
    private String director;
    3 usages
    private int length;
    2 usages
    private static int nbDigitalVideoDiscs = 0; //Class member


    1 usage  tuandattt
    public String getDirector() { return director; }

    1 usage  tuandattt
    public int getLength() { return length; }

    no usages  tuandattt
    public void setDirector(String director) { this.director = director; }

    no usages  tuandattt
    public void setLength(int length) { this.length = length; }

    2 usages  tuandattt *  1 related problem
    public DigitalVideoDisc(String title) {
        super(title);
    }

    2 usages  new *  1 related problem
    public DigitalVideoDisc(String title, String category, float cost) {
        super(title, category, cost);
    }

    9 usages  tuandattt *  1 related problem
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
        super(length, director, title, category, cost);
        nbDigitalVideoDiscs++;
        this.setId(nbDigitalVideoDiscs);
    }
```

Figure 10: Cập nhật DigitalVideoDisc Class

2. Create Track Class:

```java
package Lab_02.hust.soict.aims.media;

no usages
public class Track {
    2 usages
    private String title;
    2 usages
    private int length;

    public String getTitle() {
        return title;
    }

    no usages
    public int getLength() {
        return length;
    }

    no usages
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }
}
```

Figure 11: Track Class

3. CompactDisc Class:

```java
package Lab_02.hust.soict.aims.media;

import java.util.ArrayList;

no usages
public class CompactDisc extends DigitalVideoDisc{
    1 usage
    private String artist;
    6 usages
    private ArrayList<Track> tracks;

    no usages
    public CompactDisc(String title, String category, String director, int length, float cost, String artist, ArrayList<Track> tracks) {
        super(title, category, director, length, cost);
        this.artist = artist;
        this.tracks = tracks;
    }

    no usages
    public void addTrack(Track trackName){
        boolean found = false;
        for(Track track: tracks){
            if(track == trackName){ //kiem tra ton tai author
                found = true;
            }
        }
        if(found){
            System.out.println("Track has already existed"); //In thong bao
        }
        else{
            tracks.add(trackName); //Them track vao list
            System.out.println("Added new track");
        }
    }
```

Figure 12: CompactDisc Class

```
    no usages
    public void removeTrack(Track trackName){
        boolean found = false;
        for(Track track: tracks){
            if(track == trackName){ //kiem tra ton tai track
                found = true;
            }
        }
        if(!found){
            System.out.println("Track does not exist"); //In thong bao
        }
        else{
            tracks.remove(trackName); //Xoa track trong list
            System.out.println("Removed new author");
        }
    }

    1 usage
    public int getLength(){
        int sum = 0;
        for(Track track: tracks){
            sum += track.getLength();
        }
        return sum;
    }
}
```

Figure 13: CompactDisc Class 2

4. Create the Playable interface:

```
1   package Lab_02.hust.soict.aims.media;
2
    3 usages  3 implementations
3   interface Playable {
        1 usage  3 implementations
4       public void play();
5   }
```

Figure 14: Interface Playable

```java
package Lab_02.hust.soict.aims.media;

8 usages
public class Track implements Playable {
    2 usages
    private String title;
    2 usages
    private int length;

    public String getTitle() {
        return title;
    }

    2 usages
    public int getLength() {
        return length;
    }

    no usages
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

    1 usage
    public void play(){
        System.out.println("Playing Track: " + this.getTitle());
        System.out.println("Track Length: " + this.getLength());
    }
}
```

Figure 15: Cập nhật Track Class

```java
    1 usage  1 override  new *
    public void play(){
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD Length: " + this.getLength());
    }
```

Figure 16: Cập nhật DigitalVideoDisc Class

```java
    //In ra thông tin của mỗi track trong list
    1 usage
    public void play(){
        for (Track track: tracks){
            track.play();
        }
    }
}
```

Figure 17: Cập nhật CompactDisc Class

5. Update the Cart Class to work with Media:

```java
package Lab_02.hust.soict.aims.cart;

import Lab_02.hust.soict.aims.media.DigitalVideoDisc;
import Lab_02.hust.soict.aims.media.Media;
import Lab_02.hust.soict.aims.media.Track;

import java.util.ArrayList;

6 usages  tuandattt *
public class Cart {
    //Số lượng đĩa nhiều nhất có thể mua
    no usages
    public static final int MAX_NUMBERS_ORDERED = 20;
    //tạo biến lưu thông tin đĩa cho vào giỏ hàng
    8 usages
    private ArrayList<Media> itemsOrdered = new ArrayList<~>();

    //Thêm hàng vào giỏ
    no usages  tuandattt *
    public void addMedia(Media mediaName){
        boolean found = false;
        for(Media item: itemsOrdered){
            if(item == mediaName){ //kiem tra ton tai
                found = true;
            }
        }
        if(found){
            System.out.println("Item has already existed"); //In thong bao
        }
        else{
            itemsOrdered.remove(mediaName); //Thêm item vào list
            System.out.println("Removed new author");
        }
    }
```

Figure 18: Cập nhật Cart Class 1

```java
//Xoá hàng trong giỏ
no usages  tuandattt *
public void removeMedia(Media mediaName){
    boolean found = false;
    for(Media item: itemsOrdered){
        if(item == mediaName){ //kiem tra ton tai
            found = true;
        }
    }
    if(!found){
        System.out.println("Item does not exist"); //In thong bao
    }
    else{
        itemsOrdered.remove(mediaName); //Xoa item trong list
        System.out.println("Removed new author");
    }
}

//Phuong thuc print
1 usage  tuandattt *
public void print(){
    System.out.println("**********CART**********");
    for(Media item: itemsOrdered){
        System.out.println("DVD " + "- " + item.getId() + " " + item.getTitle() + " - " + item.getCategory() +
                ": " + item.getCost() + " $");

    }
    System.out.println("Total Cost: " + totalCost());
    System.out.println("**************************************************");
}
```

Figure 19: Cập nhật Cart Class 2

```java
1 usage  tuandattt *
public void search(int id){
    boolean found = false;
    for(Media item: itemsOrdered){
        if(item.getId() == id) {
            System.out.println(item);
            found = true;
        }
    }
    if(!found){
        System.out.println("No match is found");
    }
}

1 usage  tuandattt *
public void search(String title){
    boolean found = false;
    for(Media item: itemsOrdered){
        if(item.isMatch(title)){
            System.out.println(item);
            found = true;
        }
    }
    if(!found){
        System.out.println("No match is found");
    }
}

2 usages  tuandattt *
public float totalCost(){
    float sum = 0;
    for(Media item: itemsOrdered){
        sum = sum + item.getCost();
    }
    return sum;
}
```

Figure 20: Cập nhật Cart Class 3

6. Unique item in a list

```java
    new *
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;

        Media media = (Media) obj;
        return Objects.equals(title, media.title);
    }
}
```

Figure 21: Overriding equals() Media Class

```
    new *
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;

        Track track = (Track) obj;
        return Objects.equals(title, track.title) && length == track.length;
    }
}
```

Figure 22: Overriding equals() Track Class

7. Polymorphism with toString() method

```
package Lab_02.hust.soict.aims.media;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class TestMedia {
    public static void main(String[] args){
        List <Media> mediae = new ArrayList<Media>();
        List<String> authors = new ArrayList<>();
        authors.add("Author1");
        authors.add("Author2");
        authors.add("Author3");
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers",
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War", category: "Science Fiction", director: "George Lucas",
        Book book1 = new Book( title: "Hello World", category: "Animation", cost: 19.5f, authors);
        mediae.add(dvd1);
        mediae.add(dvd2);
        mediae.add(book1);

        for (Media m: mediae){
            System.out.println(m.toString());
        }
    }
}
```

Figure 23: TestMedia Class

```
C:\Users\MIXI_COMPUTER\.jdks\openjdk-21.0.1\bin\java.exe "-javaagent:C:\
DVD - The Lion King - Animation - null - 0: 19.95 $
DVD - Star War - Science Fiction - null - 0: 24.95 $
Book - Hello World - Animation [Author1, Author2, Author3] - 19.5 :$

Process finished with exit code 0
```

Figure 24: Kết quả TestMedia Class

8. Sort Media in the cart

```java
package Lab_02.hust.soict.aims.media;

import java.util.Comparator;

1 usage
public class MediaComparatorByCostTitle implements Comparator<Media> {
    public int compare(Media media1, Media media2) {
        // Compare by cost first
        int costComparison = Float.compare(media1.getCost(), media2.getCost());

        if (costComparison != 0) {
            return costComparison;
        }

        // If costs are the same, compare by title
        return media1.getTitle().compareTo(media2.getTitle());
    }
}
```

Figure 25:MediaComparatorByCostTitle Class

```java
1 usage
public class MediaComparatorByTitleCost implements Comparator<Media> {
    public int compare(Media media1, Media media2) {
        // Compare by title first
        int titleComparison = media1.getTitle().compareTo(media2.getTitle());

        if (titleComparison != 0) {
            return titleComparison;
        }

        // If titles are the same, compare by cost
        return Float.compare(media1.getCost(), media2.getCost());
    }
}
```

Figure 26: MediaComparatorByTitleCost

```java
1 usage
public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
1 usage
public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
no usages  new *
```

Figure 27: Media Class

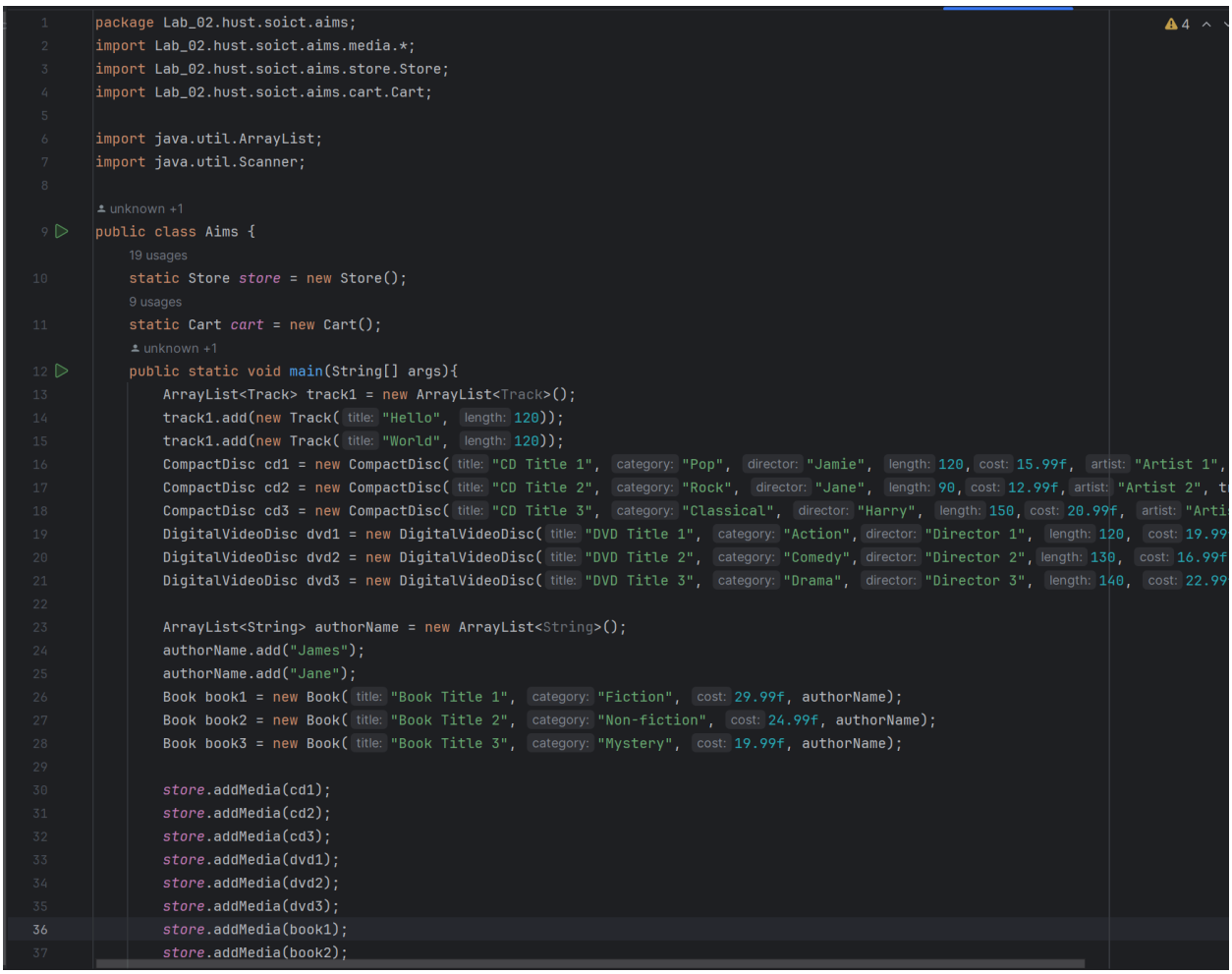
```
        Collections.sort(mediae, Media.COMPARE_BY_COST_TITLE);
        System.out.println("Sort by Cost");
        for (Media m: mediae){
            System.out.println(m.toString());
        }

        // OR Sorting by title then cost
        Collections.sort(mediae, Media.COMPARE_BY_TITLE_COST);
        System.out.println("Sort by Title");
        for (Media m: mediae){
            System.out.println(m.toString());
        }
    }
```

Figure 28: TestMedia Class

```
Sort by Cost
Book - Hello World - Animation [Author1, Author2, Author3] - 19.5 :$
DVD - The Lion King - Animation - null - 0: 19.95 $
DVD - Star War - Science Fiction - null - 0: 24.95 $
Sort by Title
Book - Hello World - Animation [Author1, Author2, Author3] - 19.5 :$
DVD - Star War - Science Fiction - null - 0: 24.95 $
DVD - The Lion King - Animation - null - 0: 19.95 $
```

Figure 29: Kết quả TestMedia Class

9. Create a complete console application in the Aims Class



```java
package Lab_02.hust.soict.aims;
import Lab_02.hust.soict.aims.media.*;
import Lab_02.hust.soict.aims.store.Store;
import Lab_02.hust.soict.aims.cart.Cart;

import java.util.ArrayList;
import java.util.Scanner;

unknown +1
public class Aims {
    19 usages
    static Store store = new Store();
    9 usages
    static Cart cart = new Cart();
    unknown +1
    public static void main(String[] args){
        ArrayList<Track> track1 = new ArrayList<Track>();
        track1.add(new Track( title: "Hello", length: 120));
        track1.add(new Track( title: "World", length: 120));
        CompactDisc cd1 = new CompactDisc( title: "CD Title 1", category: "Pop", director: "Jamie", length: 120, cost: 15.99f, artist: "Artist 1",
        CompactDisc cd2 = new CompactDisc( title: "CD Title 2", category: "Rock", director: "Jane", length: 90, cost: 12.99f, artist: "Artist 2", t
        CompactDisc cd3 = new CompactDisc( title: "CD Title 3", category: "Classical", director: "Harry", length: 150, cost: 20.99f, artist: "Arti
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "DVD Title 1", category: "Action", director: "Director 1", length: 120, cost: 19.99
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "DVD Title 2", category: "Comedy", director: "Director 2", length: 130, cost: 16.99f
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "DVD Title 3", category: "Drama", director: "Director 3", length: 140, cost: 22.99

        ArrayList<String> authorName = new ArrayList<String>();
        authorName.add("James");
        authorName.add("Jane");
        Book book1 = new Book( title: "Book Title 1", category: "Fiction", cost: 29.99f, authorName);
        Book book2 = new Book( title: "Book Title 2", category: "Non-fiction", cost: 24.99f, authorName);
        Book book3 = new Book( title: "Book Title 3", category: "Mystery", cost: 19.99f, authorName);

        store.addMedia(cd1);
        store.addMedia(cd2);
        store.addMedia(cd3);
        store.addMedia(dvd1);
        store.addMedia(dvd2);
        store.addMedia(dvd3);
        store.addMedia(book1);
        store.addMedia(book2);
```

```java
1 usage  unknown
public static void viewStore(){
    store.print();
    storeMenu();
}

4 usages  unknown
public static void showMenu(){
    System.out.println("AIMS: ");
    System.out.println("--------------------------------");
    System.out.println("1. View store");
    System.out.println("2. Update store");
    System.out.println("3. See current cart");
    System.out.println("0. Exit");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2-3");

    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        choice = scanner.nextInt();
        switch (choice) {
            case 1:
                viewStore();
                break;
            case 2:
                updateStore();
                break;
            case 3:
                currentCart();
                break;
            case 0:
                System.out.println("Exiting the application. Goodbye!");
                break;
            default:
                System.out.println("Invalid choice. Please choose again.");
        }

    } while (choice != 0);
}
```

```java
1 usage  unknown
public static void updateStore(){
    Scanner scanner = new Scanner(System.in);
    int option;
    do {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Add a media to the store");
        System.out.println("2. Remove a media from the store");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
        System.out.print("Enter option: ");
        option = scanner.nextInt();
        scanner.nextLine(); // consume newline character

        switch (option) {
            case 1:
                addMediaToStore();
                break;
            case 2:
                removeMediaFromStore();
                break;
            case 0:
                System.out.println("Going back to the main menu...");
                break;
            default:
                System.out.println("Invalid option. Please try again.");
        }
    } while (option != 0);
}

1 usage  unknown
public static void removeMediaFromStore(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = store.findMedia(title);
    store.removeMedia(item);
}
```

```java
public static void addMediaToStore(){
    Scanner scanner = new Scanner(System.in);
    System.out.println("---------------------------------------");
    System.out.println("Enter media type (CD/DVD/Book): ");
    String mediaType = scanner.nextLine();

    System.out.println("Enter media title: ");
    String title = scanner.nextLine();

    System.out.println("Enter media category: ");
    String category = scanner.nextLine();

    System.out.println("Enter media cost: ");
    float cost = scanner.nextFloat();
    scanner.nextLine();

    switch (mediaType.toUpperCase()) {
        case "CD":
            System.out.println("Enter media director: ");
            String directorCD = scanner.nextLine();

            System.out.println("Enter media length: ");
            int lengthCD = scanner.nextInt();

            System.out.println("Enter media artist: ");
            String artist = scanner.nextLine();

            System.out.println("Enter number of tracks: ");
            ArrayList<Track> trackName = new ArrayList<Track>();
            int numTrack = scanner.nextInt();
            for(int i = 0; i < numTrack; i++){
                String nameOfTrack = scanner.nextLine();
                int length = scanner.nextInt();
                Track track = new Track(nameOfTrack, length);
                trackName.add(track);
            }

            CompactDisc cd = new CompactDisc(title,category,directorCD,lengthCD,cost,artist,trackName);
            store.addMedia(cd);
            break;
```

```
            break;

        case "DVD":
            System.out.println("Enter media director: ");
            String directorDVD = scanner.nextLine();

            System.out.println("Enter media length: ");
            int lengthDVD = scanner.nextInt();

            DigitalVideoDisc dvd = new DigitalVideoDisc(title,category,directorDVD,lengthDVD,cost);
            store.addMedia(dvd);
            break;

        case "BOOK":
            System.out.println("Enter number of authors: ");
            ArrayList<String> name = new ArrayList<String>();
            int numAuthor = scanner.nextInt();
            for(int i = 0; i < numAuthor; i++){
                String author = scanner.nextLine();
                name.add(author);
            }
            Book book = new Book(title,category,cost, name);
            store.addMedia(book);
            break;
        default:
            System.out.println("Invalid media type.");
    }
}

1 usage  unknown
public static void storeMenu() {
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. See a media's details");
    System.out.println("2. Add a media to cart");
    System.out.println("3. Play a media");
    System.out.println("4. See current cart");
    System.out.println("0. Back");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2-3-4");
```

```
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            showMenu();
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    seeMediaDetail();
                    break;
                case 2:
                    addMedia();
                    break;
                case 3:
                    playAnotherMedia();
                    break;
                case 4:
                    currentCart();
                    break;
                case 0:
                    System.out.println("Return to main menu");
                    break;
                default:
                    System.out.println("Invalid choice. Please choose again.");
            }

        } while (choice != 0);
    }
```

```
1 usage  unknown +1
public static void seeMediaDetail(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = store.findMedia(title);
    if(item == null){
        System.out.println("Cannot find");
    }
    else {
        System.out.println(item);
        mediaDetailsMenu(item);
    }
}

1 usage  unknown
public static void addMedia(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = store.findMedia(title);
    if(item == null){
        System.out.println("Cannot find");
    }
    else {
        cart.addMedia(item);
    }
}
```

```
1 usage  unknown
public static void playAnotherMedia(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = store.findMedia(title);
    if(item == null){
        System.out.println("Cannot find");
    }
    else {
        playMedia(item);
    }
}

2 usages  unknown
public static void currentCart(){
    cart.print();
    cartMenu();
}

1 usage  unknown
public static void mediaDetailsMenu(Media item) {
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Add to cart");
    System.out.println("2. Play");
    System.out.println("0. Back");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2");

    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        showMenu();
        choice = scanner.nextInt();
        switch (choice) {
            case 1:
                cart.addMedia(item);
                break;
            case 2:
```

```java
public static void playMedia(Media item){
    if(item instanceof CompactDisc){
        ((CompactDisc) item).play();
    } else if (item instanceof  DigitalVideoDisc) {
        ((DigitalVideoDisc) item).play();
    } else {
        System.out.println("Cannot play media");
    }
}

1 usage  unknown
public static void cartMenu() {
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Filter medias in cart");
    System.out.println("2. Sort medias in cart");
    System.out.println("3. Remove media from cart");
    System.out.println("4. Play a media");
    System.out.println("0. Back");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2-3-4-5");
    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        showMenu();
        choice = scanner.nextInt();
        switch (choice) {
            case 1:
                filterMedia();
                break;
            case 2:
                sortMedia();
                break;
            case 3:
                removeMediaFromCart();
                break;
            case 4:
                playMediaFromCart();
                break;
            case 0:
                System.out.println("Return to Store Menu");
```

```
            case 0:
                System.out.println("Return to Store Menu");
                break;
            default:
                System.out.println("Invalid choice. Please choose again.");
        }

    } while (choice != 0);
}

1 usage  unknown
public static void filterMedia(){
    Scanner scanner = new Scanner(System.in);
    int option;
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Filter medias in cart by id");
    System.out.println("2. Filter medias in cart by title");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 1-2");
    option = scanner.nextInt();
    scanner.nextLine();
    if (option == 1){
        System.out.println("Enter the id:");
        int inputId = scanner.nextInt();
        cart.search(inputId);
    } else if (option == 2) {
        System.out.println("Enter the title:");
        String inputTitle = scanner.nextLine();
        cart.search(inputTitle);
    } else {
        System.out.println("Invalid choice, back to Cart menu");
    }
}
```

```java
public static void sortMedia(){
    Scanner scanner = new Scanner(System.in);
    int option;
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Sort medias in cart by cost");
    System.out.println("2. Sort medias in cart by title");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 1-2");
    option = scanner.nextInt();
    scanner.nextLine();
    if (option == 1){
        cart.getItemsOrdered().sort(Media.COMPARE_BY_COST_TITLE);
    } else if (option == 2) {
        cart.getItemsOrdered().sort(Media.COMPARE_BY_TITLE_COST);
    } else {
        System.out.println("Invalid choice");
    }
}

1 usage  unknown
public static void removeMediaFromCart(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = store.findMedia(title);
    cart.removeMedia(item);
}
```

```
1 usage  unknown
public static void playMediaFromCart(){
    System.out.print("Enter Name of the media: ");
    Scanner scanner = new Scanner(System.in);
    String title = scanner.nextLine();
    Media item = cart.findMedia(title);
    if(item == null){
        System.out.println("Cannot find");
    }
    else {
        playMedia(item);
    }
}

}
```

Figure 30: Menu

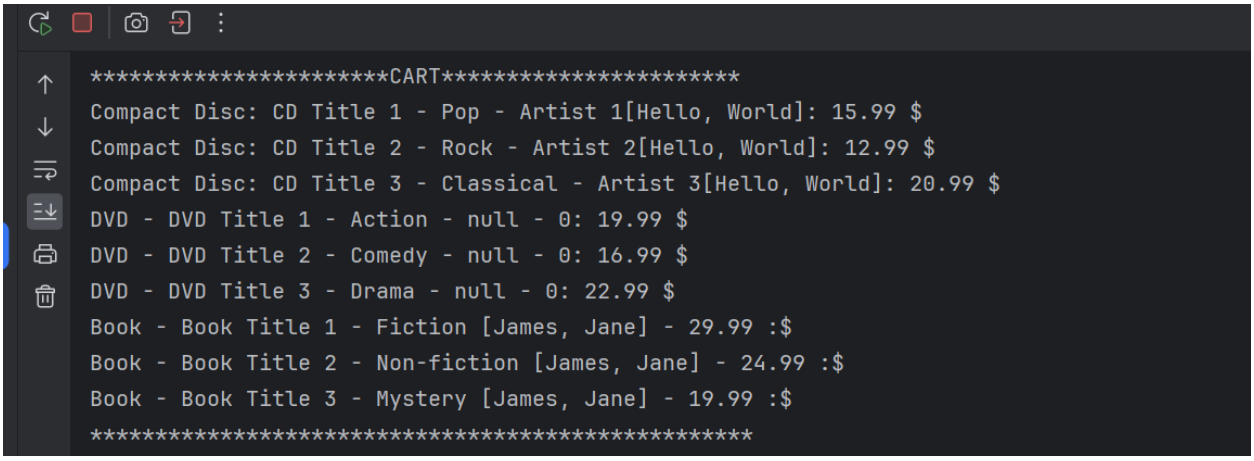


Figure 31: Kết quả

10. Update Class diagram

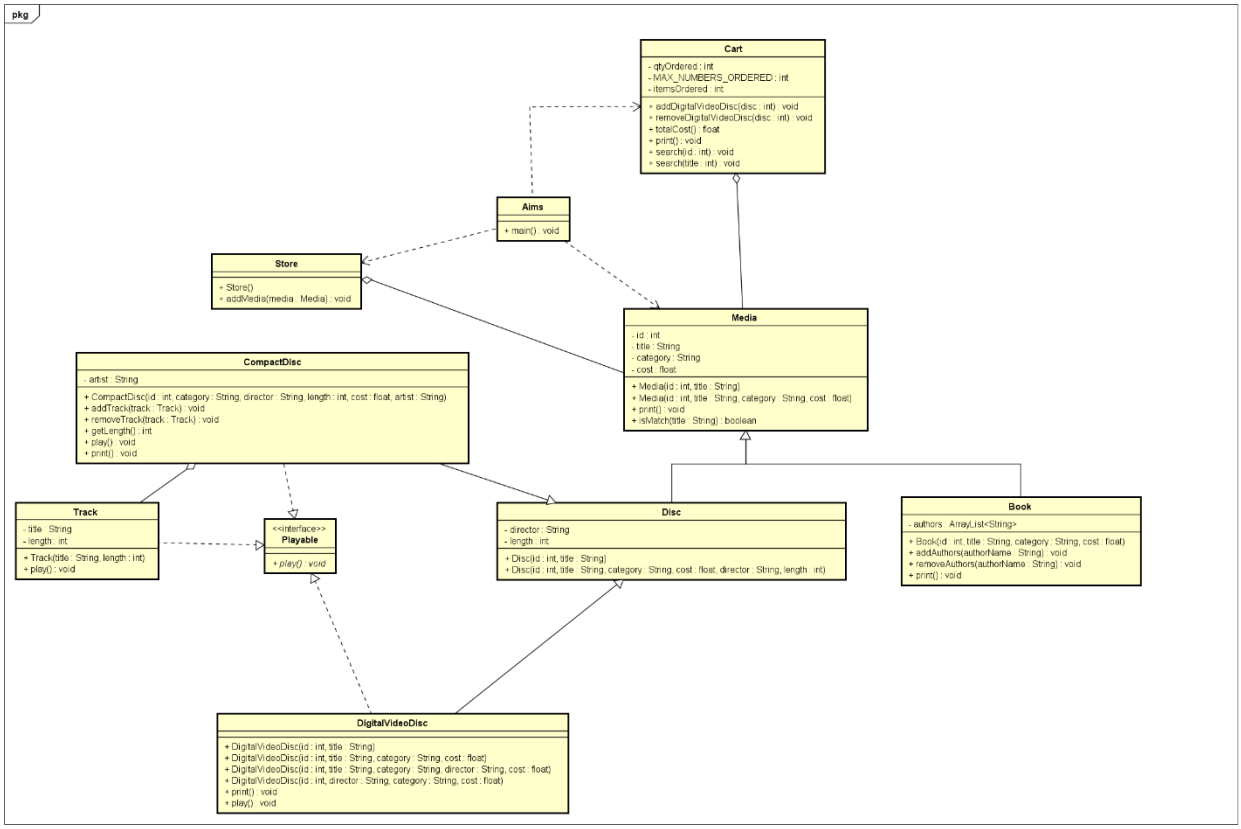


Figure 32: Class Diagram

11. Update user diagram

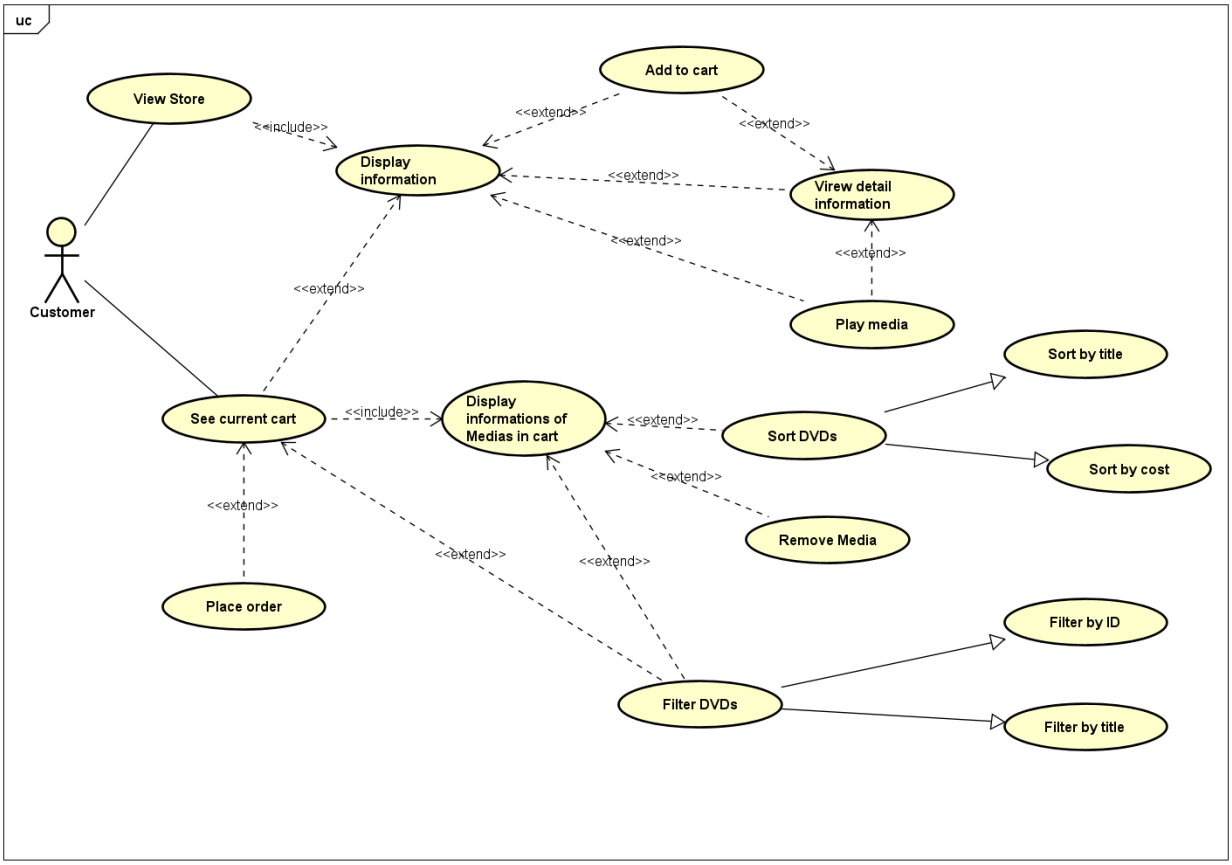


Figure 33: Customer Usecase Diagram

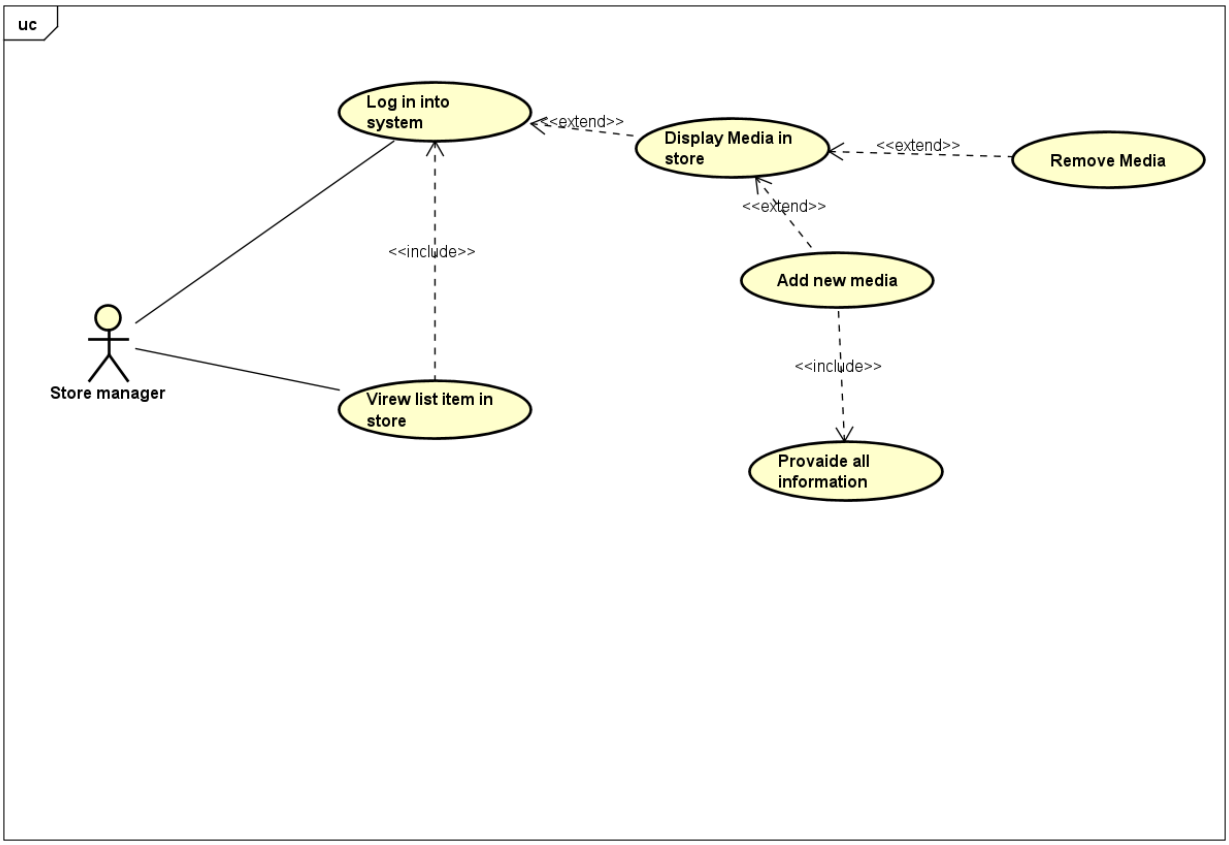


Figure 34: Manager Usecase Diagram