

**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 <code>toString()</code> 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

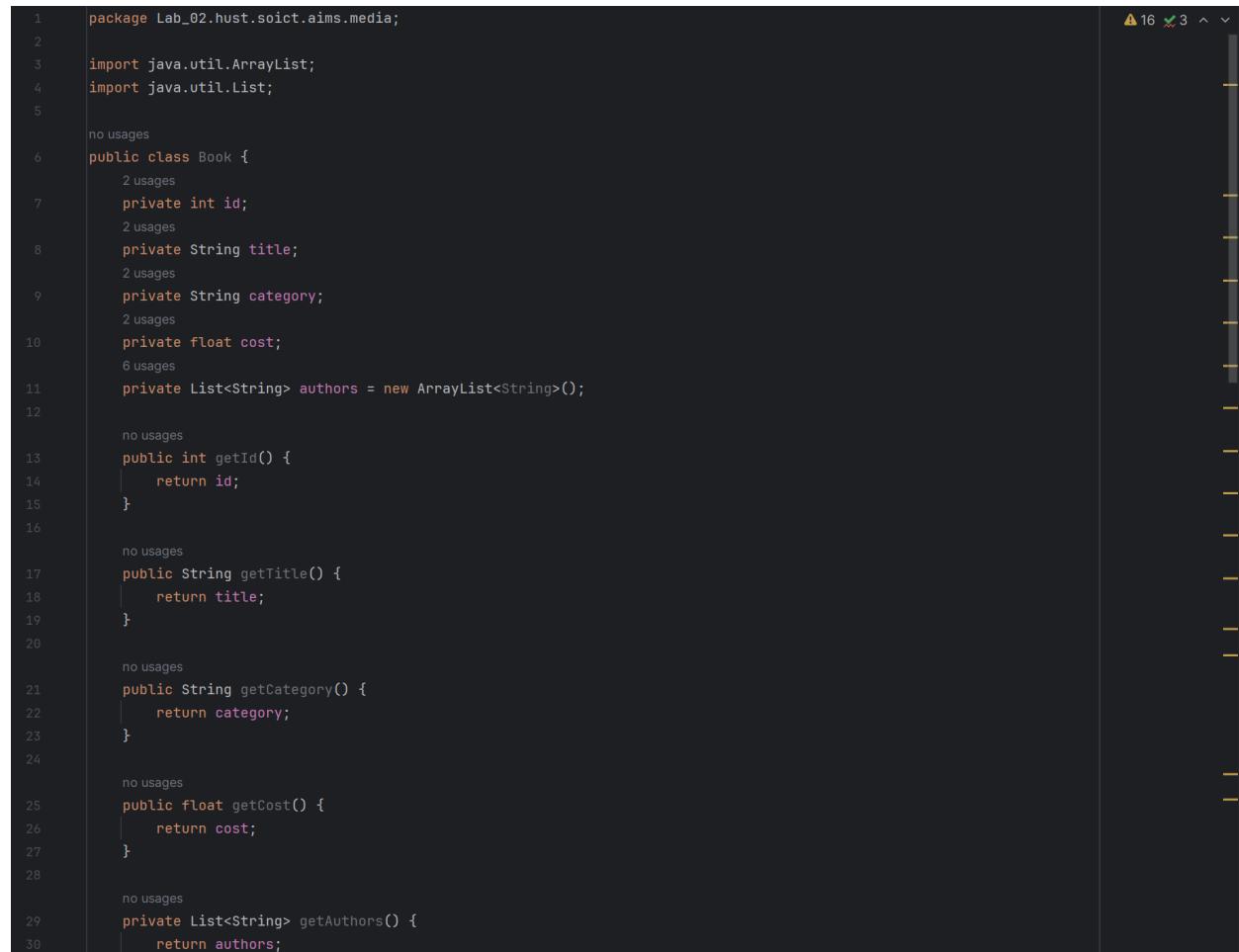
## Table of Figures

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:



The screenshot shows a code editor window with a dark theme. The code is written in Java and defines a class named Book. The class has private fields for id, title, category, and cost, and a private field for authors (a List of strings). It includes four public getter methods: getId, getTitle, getCategory, and getCost, and one private method getAuthors. The code is annotated with usage counts for each line, indicating where it is used in the codebase.

```
1 package Lab_02.hust.soict.aims.media;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 no usages
7 public class Book {
8     2 usages
9         private int id;
10    2 usages
11        private String title;
12    2 usages
13        private String category;
14    2 usages
15        private float cost;
16    6 usages
17        private List<String> authors = new ArrayList<String>();
18
19 no usages
20
21     public int getId() {
22         return id;
23     }
24
25     no usages
26     public String getTitle() {
27         return title;
28     }
29
30     no usages
31     public String getCategory() {
32         return category;
33     }
34
35     no usages
36     public float getCost() {
37         return cost;
38     }
39
40     no usages
41     private List<String> getAuthors() {
42         return authors;
43     }
44 }
```

Figure 1: Book Class 1

```

29     no usages
30     private List<String> getAuthors() {
31         return authors;
32     }
33
34     no usages
35     public void setId(int id) {
36         this.id = id;
37     }
38
39     no usages
40     public void setTitle(String title) {
41         this.title = title;
42     }
43
44     no usages
45     public void setCategory(String category) {
46         this.category = category;
47     }
48
49     no usages
50     public void setCost(float cost) {
51         this.cost = cost;
52     }
53

```

Figure 2: Book Class 2

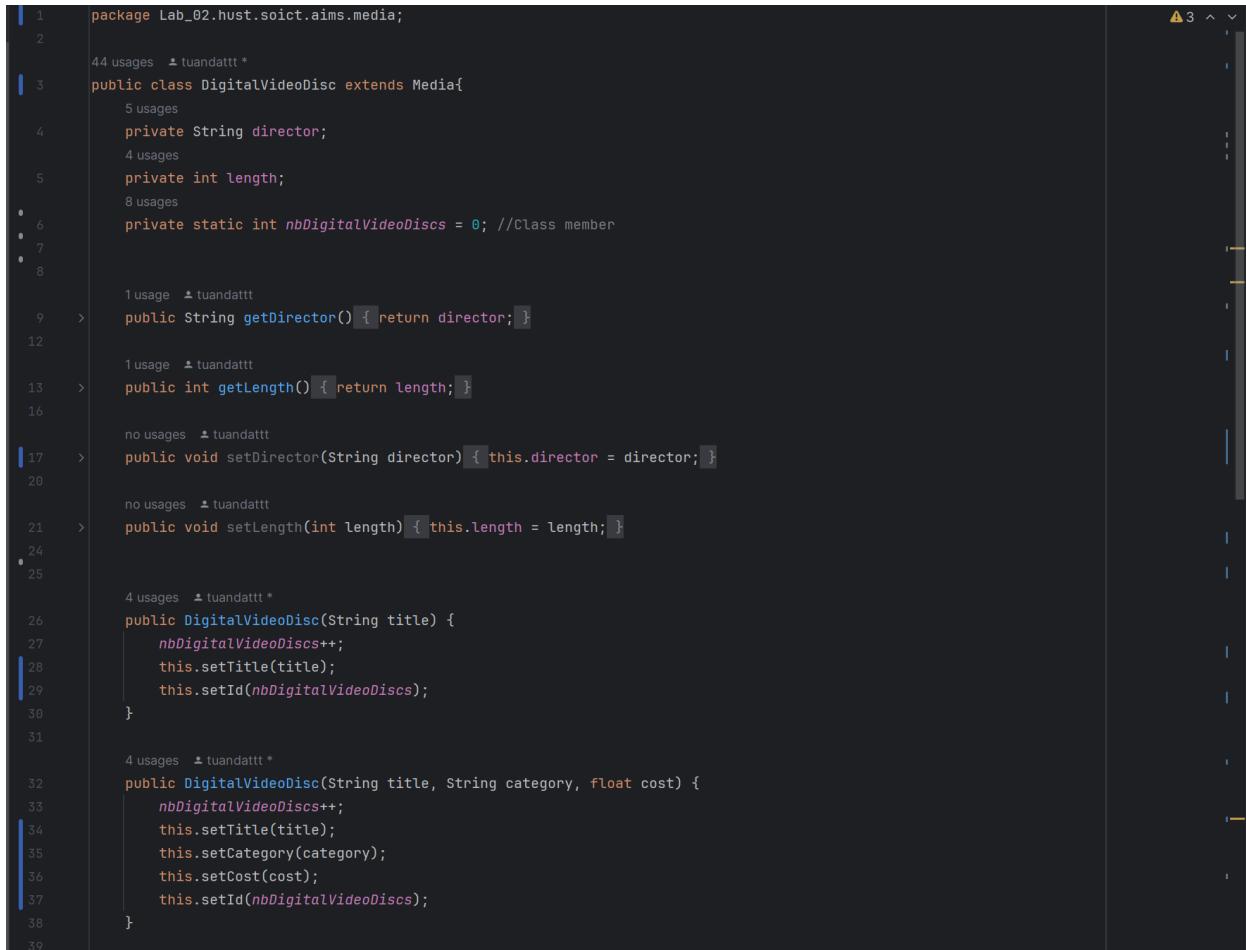
```

54     no usages
55     public void addAuthor(String authorName){
56         boolean found = false;
57         for(String author: authors){
58             if(author.equals(authorName)){ //kiem tra ton tai author
59                 found = true;
60             }
61         }
62         if(found){
63             System.out.println("Author has already existed"); //In thong bao
64         }
65         else{
66             authors.add(authorName); //Them author vao list
67             System.out.println("Added new author");
68         }
69
70     no usages
71     public void removeAuthor(String authorName){
72         boolean found = false;
73         for(String author: authors){
74             if(author.equals(authorName)){ //kiem tra ton tai author
75                 found = true;
76             }
77         }
78         if(!found){
79             System.out.println("Author do not exist"); //In thong bao
80         }
81         else{
82             authors.remove(authorName); //Xoa author vao list
83             System.out.println("Removed new author");
84         }
85     }

```

Figure 3: Book Class 3

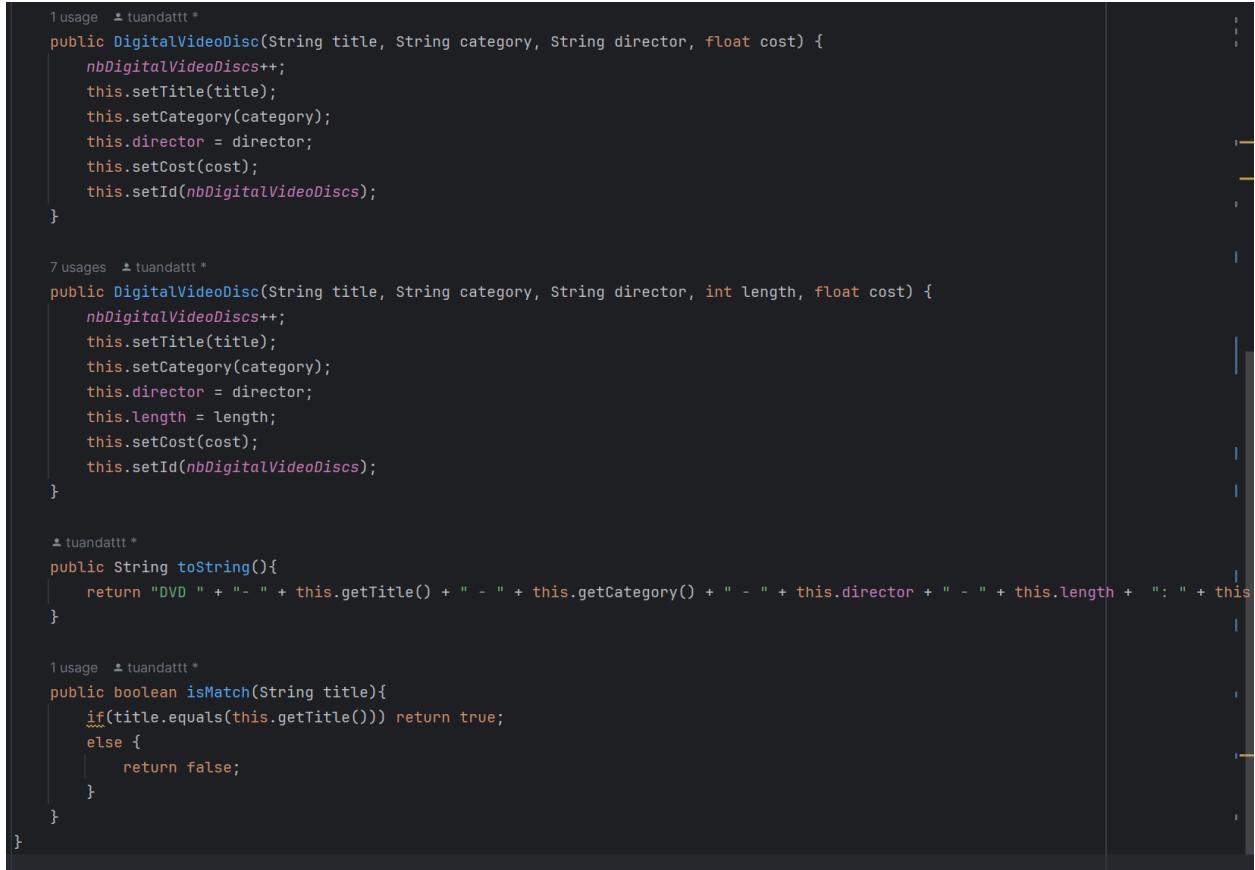
## 2. Create the abstract Media Class:



```
1 package Lab_02.hust.soict.aims.media;
2
3 44 usages ▲ tuandattt *
4
5 public class DigitalVideoDisc extends Media{
6     5 usages
7     private String director;
8     4 usages
9     private int length;
10    8 usages
11    private static int nbDigitalVideoDiscs = 0; //Class member
12
13    1 usage ▲ tuandattt
14    > public String getDirector() { return director; }
15
16    1 usage ▲ tuandattt
17    > public int getLength() { return length; }
18
19    no usages ▲ tuandattt
20    > public void setDirector(String director) { this.director = director; }
21
22    no usages ▲ tuandattt
23    > public void setLength(int length) { this.length = length; }
24
25    4 usages ▲ tuandattt *
26    public DigitalVideoDisc(String title) {
27        nbDigitalVideoDiscs++;
28        this.setTitle(title);
29        this.setId(nbDigitalVideoDiscs);
30    }
31
32    4 usages ▲ tuandattt *
33    public DigitalVideoDisc(String title, String category, float cost) {
34        nbDigitalVideoDiscs++;
35        this.setTitle(title);
36        this.setCategory(category);
37        this.setCost(cost);
38        this.setId(nbDigitalVideoDiscs);
39    }

```

Figure 4: Cập nhật DigitalVideoDisc 1



```
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



```
1 package Lab_02.hust.soict.aims.media;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 no usages
7 public class Book extends Media{
8     6 usages
9     private List<String> authors = new ArrayList<String>();
10
11     no usages
12     private List<String> getAuthors() {
13         return authors;
14     }
15
16     no usages
17     private void setAuthors(List<String> authors) {
18         this.authors = authors;
19     }
20 }
```

Figure 6: Cập nhật Book Class

```

1  package Lab_02.hust.soict.aims.media;
2
3  public abstract class Media {
4      private int id;
5      private String title;
6      private String category;
7      private float cost;
8
9      public int getId() {
10         return id;
11     }
12
13     public String getTitle() {
14         return title;
15     }
16
17     public String getCategory() {
18         return category;
19     }
20
21     public float getCost() {
22         return cost;
23     }
24

```

Figure 7: Media Class 2

```

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;
}

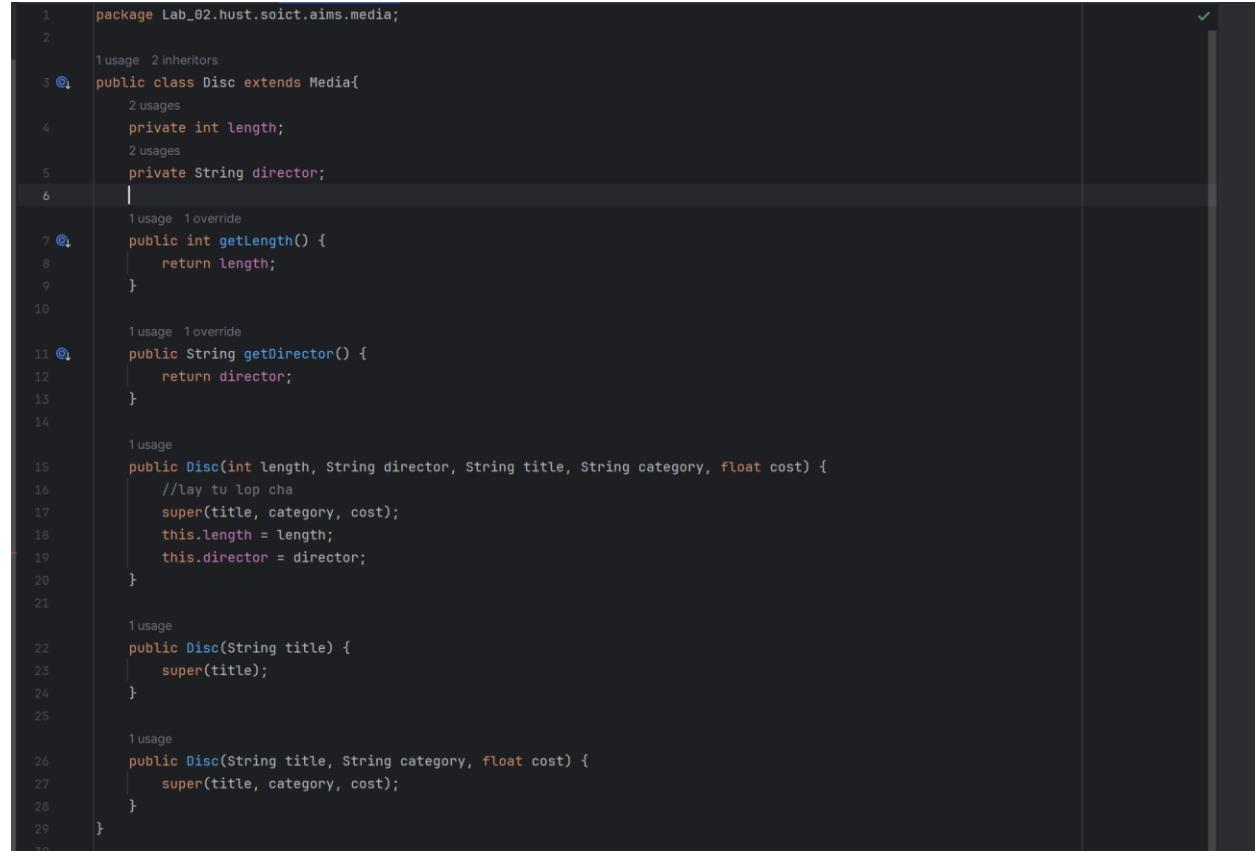
}

```

Figure 8: Media Class 2

### 3. Creating CompactDisc class:

#### 1. Create the Disc class extending the Media Class:



```
1 package Lab_02.hust.soict.aims.media;
2
3 @4 public class Disc extends Media{
4     2 usages
5     private int length;
6     2 usages
7     private String director;
8
9     1 usage 1 override
10    public int getLength() {
11        return length;
12    }
13
14    1 usage 1 override
15    public String getDirector() {
16        return director;
17    }
18
19    1 usage
20    public Disc(int length, String director, String title, String category, float cost) {
21        //lay tu lop cha
22        super(title, category, cost);
23        this.length = length;
24        this.director = director;
25    }
26
27    1 usage
28    public Disc(String title) {
29        super(title);
30    }
31
32    1 usage
33    public Disc(String title, String category, float cost) {
34        super(title, category, cost);
35    }
36}
```

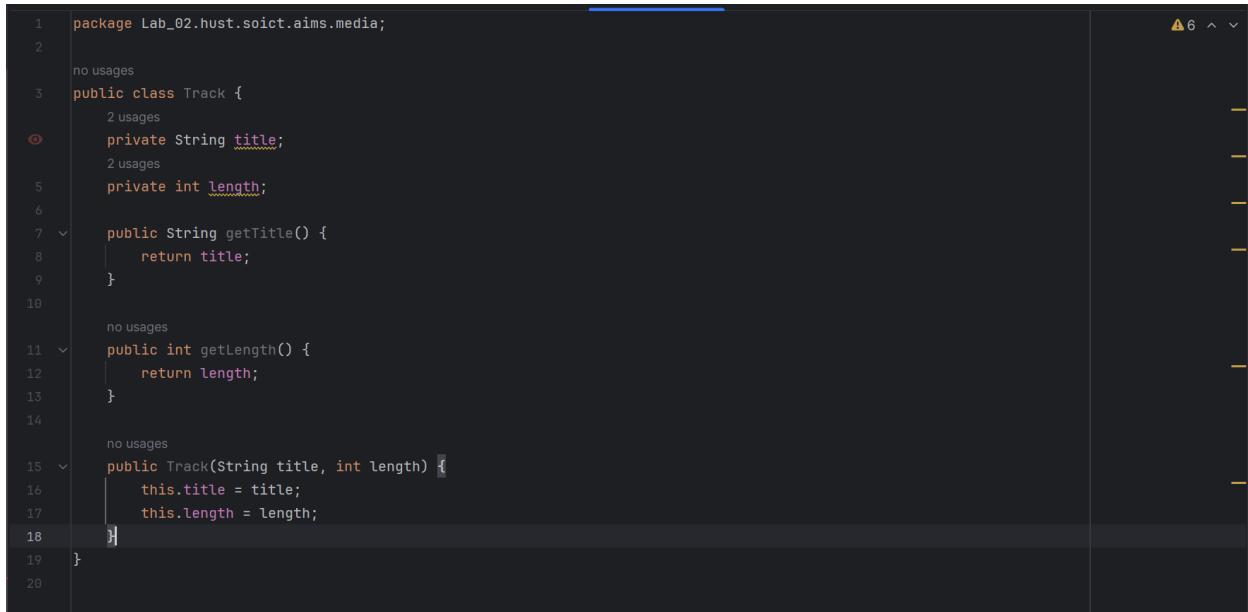
Figure 9: Disc Class

The screenshot shows a Java code editor with the following code:

```
1 package Lab_02.hust.soict.aims.media;
2
3 @public class DigitalVideoDisc extends Disc{
4     private String director;
5     private int length;
6     private static int nbDigitalVideoDiscs = 0; //Class member
7
8
9 @  >     public String getDirector() { return director; }
10
11    public int getLength() { return length; }
12
13 @  >    public void setDirector(String director) { this.director = director; }
14
15    public void setLength(int length) { this.length = length; }
16
17
18    public DigitalVideoDisc(String title) {
19        super(title);
20    }
21
22
23    public DigitalVideoDisc(String title, String category, float cost) {
24        super(title, category, cost);
25    }
26
27
28    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
29        super(length, director, title, category, cost);
30        nbDigitalVideoDiscs++;
31        this.setId(nbDigitalVideoDiscs);
32    }
33
34
35
36
37 }
```

Figure 10: Cập nhật DigitalVideoDisc Class

## 2. Create Track Class:



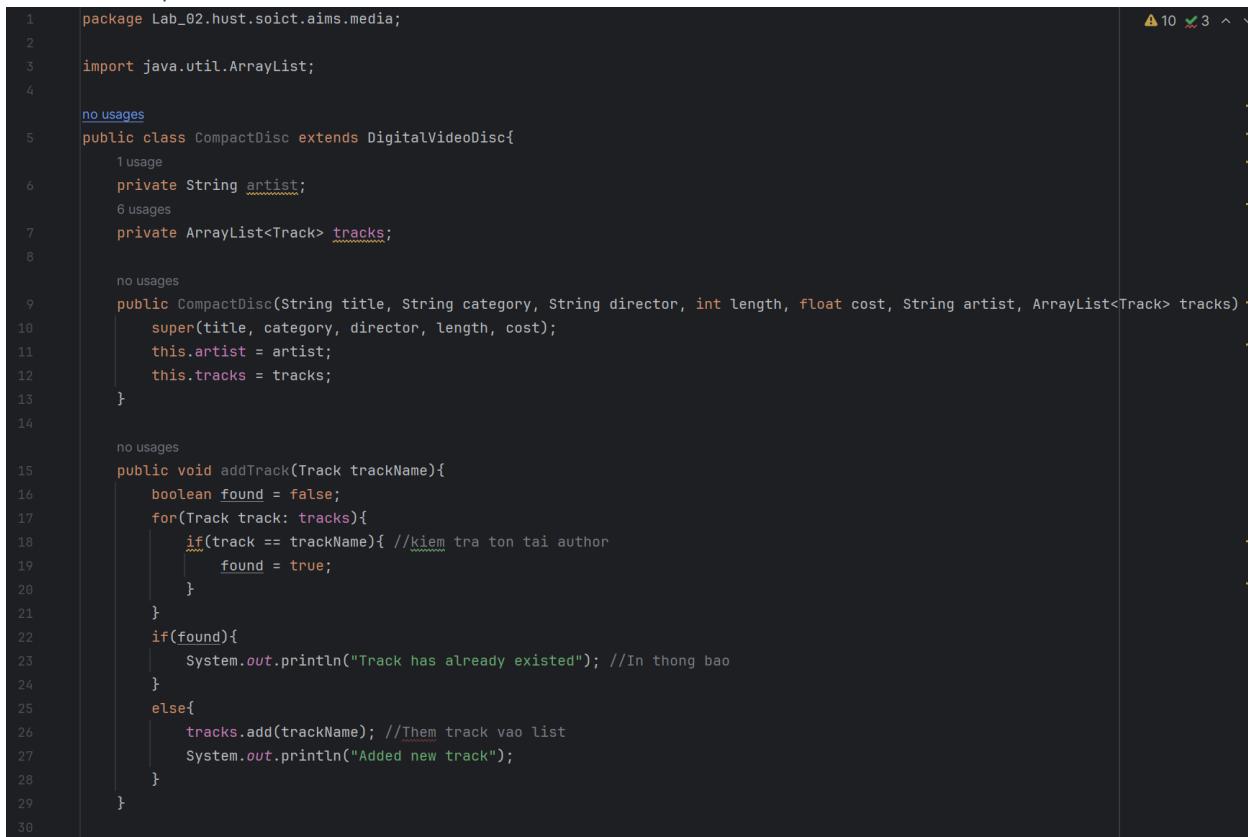
```

1 package Lab_02.hust.soict.aims.media;
2
3 no usages
4
5 public class Track {
6     2 usages
7     private String title;
8     2 usages
9     private int length;
10
11    public String getTitle() {
12        return title;
13    }
14
15    no usages
16    public int getLength() {
17        return length;
18    }
19
20    no usages
21    public Track(String title, int length) {
22        this.title = title;
23        this.length = length;
24    }
25
26}

```

Figure 11: Track Class

## 3. CompactDisc Class:



```

1 package Lab_02.hust.soict.aims.media;
2
3 import java.util.ArrayList;
4
5 no usages
6
7 public class CompactDisc extends DigitalVideoDisc{
8     1 usage
9     private String artist;
10    6 usages
11    private ArrayList<Track> tracks;
12
13    no usages
14    public CompactDisc(String title, String category, String director, int length, float cost, String artist, ArrayList<Track> tracks) {
15        super(title, category, director, length, cost);
16        this.artist = artist;
17        this.tracks = tracks;
18    }
19
20    no usages
21    public void addTrack(Track trackName){
22        boolean found = false;
23        for(Track track: tracks){
24            if(track == trackName){ //kiem tra ton tai author
25                found = true;
26            }
27        }
28        if(found){
29            System.out.println("Track has already existed"); //In thong bao
30        }
31        else{
32            tracks.add(trackName); //Them track vao list
33            System.out.println("Added new track");
34        }
35    }
36
37}

```

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

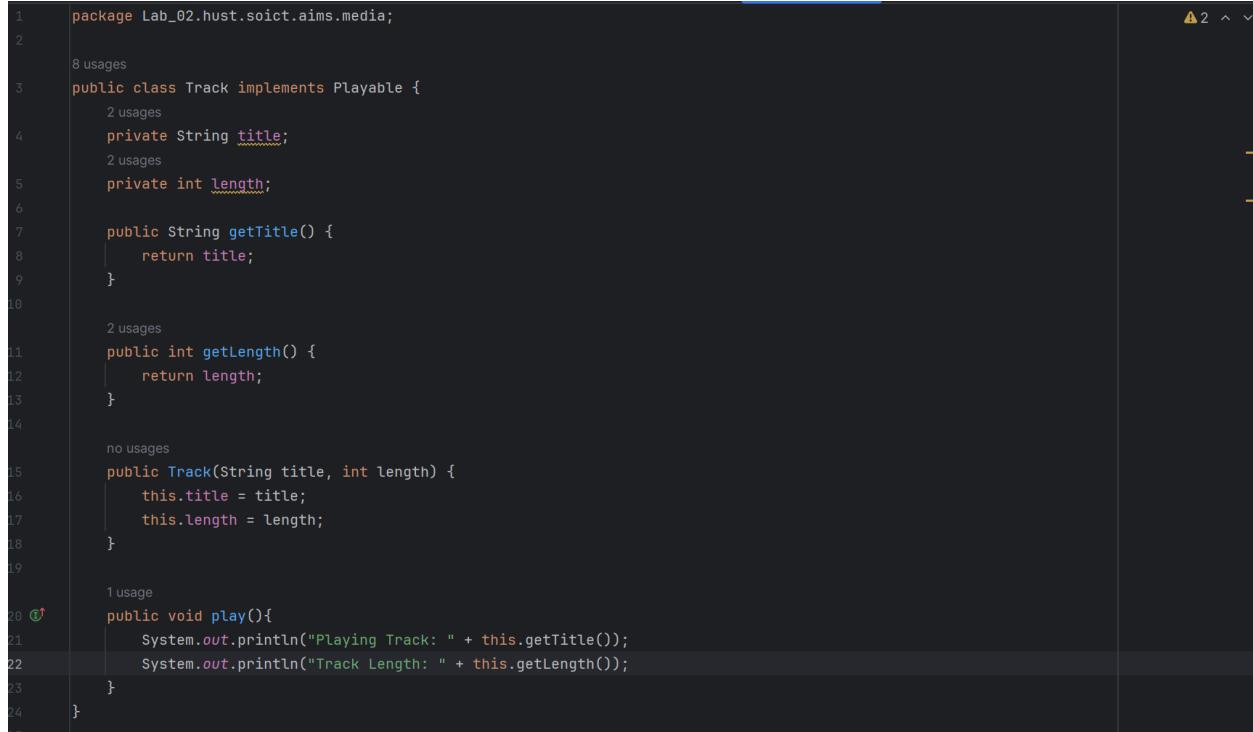
#### 4. Create the Playable interface:

```

1 package Lab_02.hust.soict.aims.media;
2
3 3 usages 3 implementations
4 @interface Playable {
5     1 usage 3 implementations
6     public void play();
7 }

```

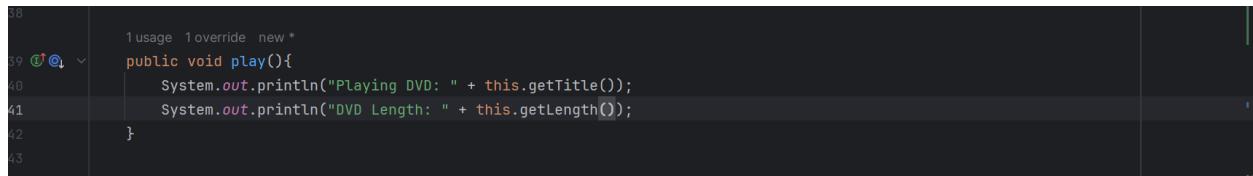
Figure 14: Interface Playable



```

1 package Lab_02.hust.soict.aims.media;
2
3 public class Track implements Playable {
4     private String title;
5     private int length;
6
7     public String getTitle() {
8         return title;
9     }
10
11    public int getLength() {
12        return length;
13    }
14
15    public Track(String title, int length) {
16        this.title = title;
17        this.length = length;
18    }
19
20    public void play(){
21        System.out.println("Playing Track: " + this.getTitle());
22        System.out.println("Track Length: " + this.getLength());
23    }
24 }
```

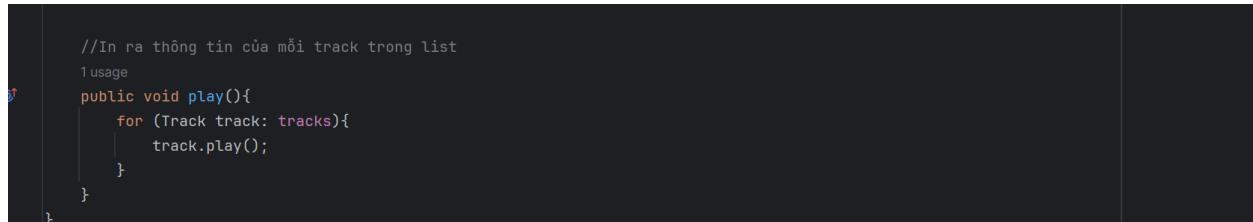
Figure 15: Cập nhật Track Class



```

39 public void play(){
40     System.out.println("Playing DVD: " + this.getTitle());
41     System.out.println("DVD Length: " + this.getLength());
42 }
```

Figure 16: Cập nhật DigitalVideoDisc Class



```

1 //In ra thông tin của mỗi track trong list
2 public void play(){
3     for (Track track: tracks){
4         track.play();
5     }
6 }
```

Figure 17: Cập nhật CompactDisc Class

## 5. Update the Cart Class to work with Media:

The screenshot shows a Java code editor with the following code:

```
1 package Lab_02.hust.soict.aims.cart;
2
3 import Lab_02.hust.soict.aims.media.DigitalVideoDisc;
4 import Lab_02.hust.soict.aims.media.Media;
5 import Lab_02.hust.soict.aims.media.Track;
6
7 import java.util.ArrayList;
8
9 6 usages ▾ tuandattt *
10 public class Cart {
11     //Số lượng đĩa nhiều nhất có thể mua
12     no usages
13     public static final int MAX_NUMBERS_ORDERED = 20;
14     //tạo biến lưu thông tin đĩa cho vào giỏ hàng
15     8 usages
16     private ArrayList<Media> itemsOrdered = new ArrayList<>();
17
18     //Thêm hàng vào giỏ
19     no usages ▾ tuandattt *
20     public void addMedia(Media mediaName){
21         boolean found = false;
22         for(Media item: itemsOrdered){
23             if(item == mediaName){ //kiểm tra tồn tại
24                 found = true;
25             }
26         }
27         if(found){
28             System.out.println("Item has already existed"); //In thông báo
29         }
30     }
31 }
```

The code defines a `Cart` class that manages a list of `Media` objects. It includes methods for adding media to the cart and checking if a specific item already exists.

Figure 18: Cập nhật Cart Class 1

```
//Xoa hàng trong gio
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

```

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

```

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

```

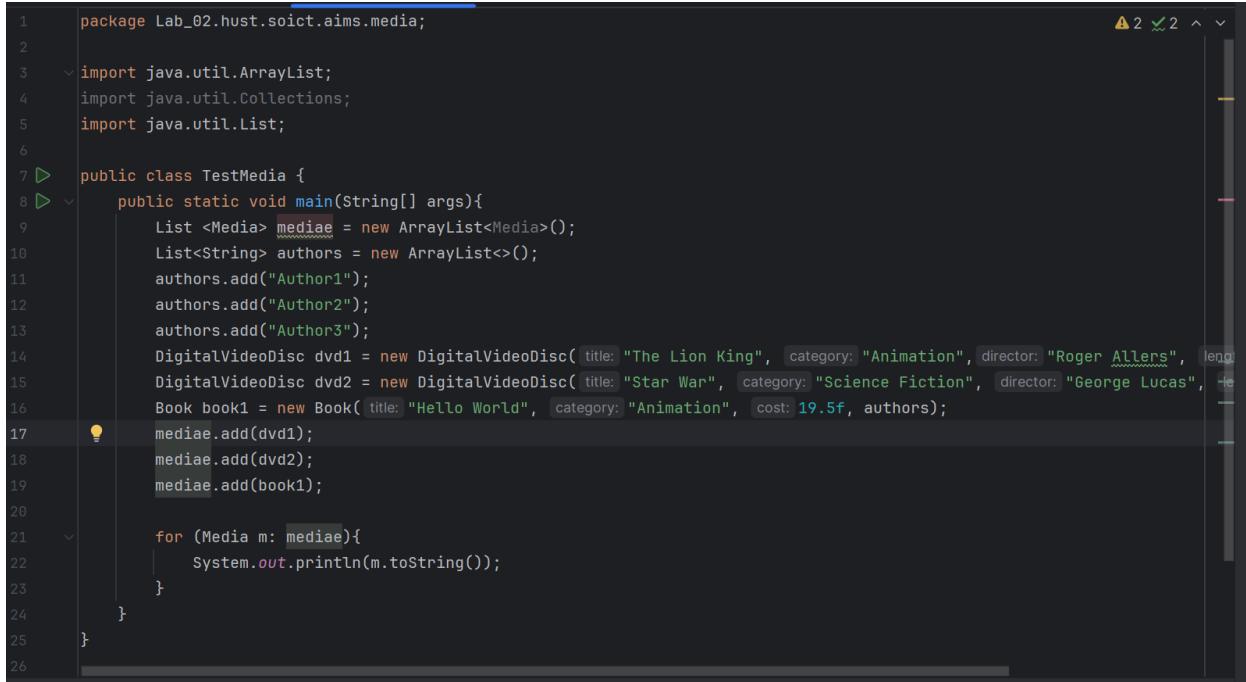
new *
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (obj == null || getClass() != obj.getClass()) return false;

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

```

Figure 22: Overriding equals() Track Class

## 7. Polymorphism with toString() method

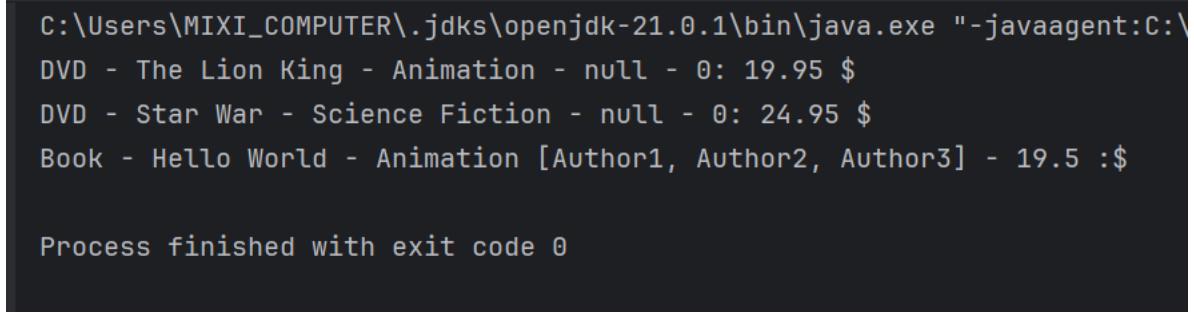


```

1 package Lab_02.hust.soict.aims.media;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.List;
6
7 public class TestMedia {
8     public static void main(String[] args){
9         List<Media> mediae = new ArrayList<Media>();
10        List<String> authors = new ArrayList<String>();
11        authors.add("Author1");
12        authors.add("Author2");
13        authors.add("Author3");
14        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers", length: 120 );
15        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War", category: "Science Fiction", director: "George Lucas", length: 150 );
16        Book book1 = new Book( title: "Hello World", category: "Animation", cost: 19.5f, authors );
17        mediae.add(dvd1);
18        mediae.add(dvd2);
19        mediae.add(book1);
20
21        for (Media m: mediae){
22            System.out.println(m.toString());
23        }
24    }
25 }

```

Figure 23: TestMedia Class



```

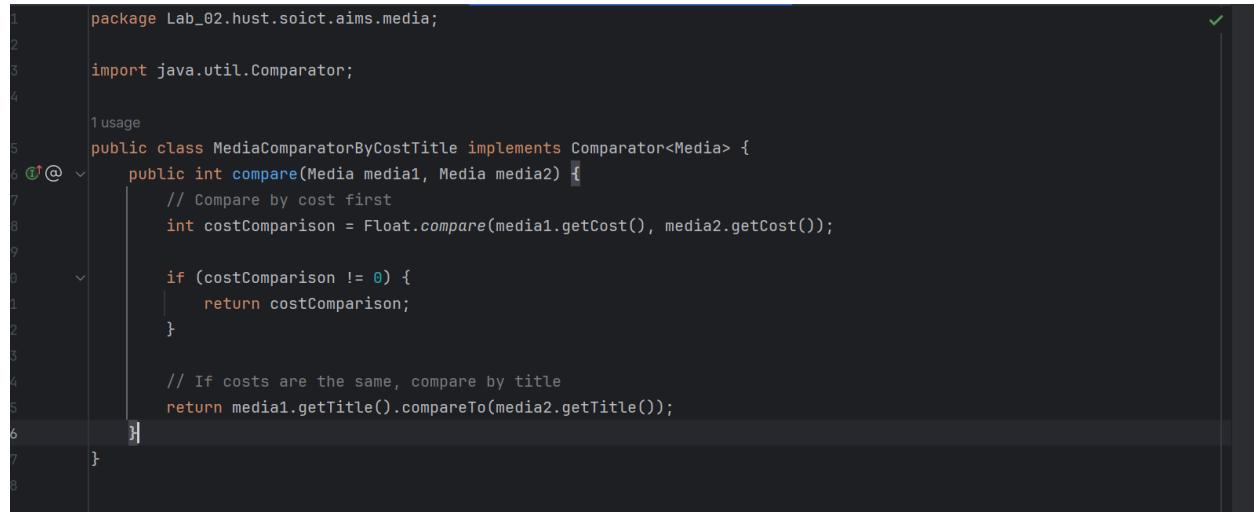
C:\Users\ MIXI_COMPUTER\.jdks\openjdk-21.0.1\bin\java.exe "-javaagent:C:\\\
DVD - The Lion King - Animation - null - 0: 19.95 $
DVD - Star War - Science Fiction - null - 0: 24.95 $
Book - Hello World - Animation [Author1, Author2, Author3] - 19.5 :$"

Process finished with exit code 0

```

Figure 24: Kết quả TestMedia Class

## 8. Sort Media in the cart



```

1 package Lab_02.hust.soict.aims.media;
2
3 import java.util.Comparator;
4
5 1 usage
6 public class MediaComparatorByCostTitle implements Comparator<Media> {
7     @ 6 public int compare(Media media1, Media media2) {
8         // Compare by cost first
9         int costComparison = Float.compare(media1.getCost(), media2.getCost());
10
11        if (costComparison != 0) {
12            return costComparison;
13        }
14
15        // If costs are the same, compare by title
16        return media1.getTitle().compareTo(media2.getTitle());
17    }
18 }

```

Figure 25: MediaComparatorByCostTitle Class

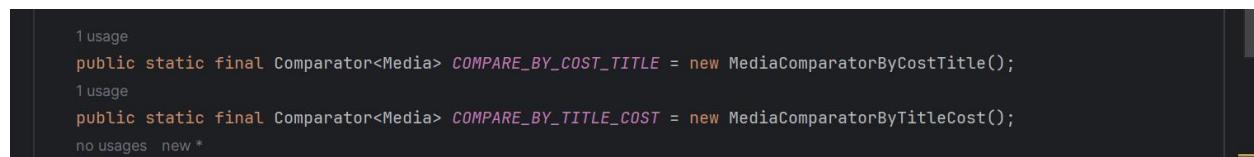


```

1 usage
2 public class MediaComparatorByTitleCost implements Comparator<Media> {
3     public int compare(Media media1, Media media2) {
4         // Compare by title first
5         int titleComparison = media1.getTitle().compareTo(media2.getTitle());
6
7         if (titleComparison != 0) {
8             return titleComparison;
9         }
10
11        // If titles are the same, compare by cost
12        return Float.compare(media1.getCost(), media2.getCost());
13    }
14 }

```

Figure 26: MediaComparatorByTitleCost



```

1 usage
2 public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
3 1 usage
4 public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
5 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

```
1 package Lab_02.hust.soict.aims;
2 import Lab_02.hust.soict.aims.media.*;
3 import Lab_02.hust.soict.aims.store.Store;
4 import Lab_02.hust.soict.aims.cart.Cart;
5
6 import java.util.ArrayList;
7 import java.util.Scanner;
8
9 // unknown +1
9 ▷ public class Aims {
10     19 usages
11     static Store store = new Store();
12     9 usages
13     static Cart cart = new Cart();
14     // unknown +1
15     public static void main(String[] args){
16         ArrayList<Track> track1 = new ArrayList<Track>();
17         track1.add(new Track( title: "Hello", length: 120));
18         track1.add(new Track( title: "World", length: 120));
19         CompactDisc cd1 = new CompactDisc( title: "CD Title 1", category: "Pop", director: "Jamie", length: 120, cost: 15.99f, artist: "Artist 1");
20         CompactDisc cd2 = new CompactDisc( title: "CD Title 2", category: "Rock", director: "Jane", length: 90, cost: 12.99f, artist: "Artist 2", t
21         CompactDisc cd3 = new CompactDisc( title: "CD Title 3", category: "Classical", director: "Harry", length: 150, cost: 20.99f, artist: "Arti
22         DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "DVD Title 1", category: "Action", director: "Director 1", length: 120, cost: 19.99
23         DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "DVD Title 2", category: "Comedy", director: "Director 2", length: 130, cost: 16.99f
24         DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "DVD Title 3", category: "Drama", director: "Director 3", length: 140, cost: 22.99
25
26         ArrayList<String> authorName = new ArrayList<String>();
27         authorName.add("James");
28         authorName.add("Jane");
29         Book book1 = new Book( title: "Book Title 1", category: "Fiction", cost: 29.99f, authorName);
30         Book book2 = new Book( title: "Book Title 2", category: "Non-fiction", cost: 24.99f, authorName);
31         Book book3 = new Book( title: "Book Title 3", category: "Mystery", cost: 19.99f, authorName);
32
33         store.addMedia(cd1);
34         store.addMedia(cd2);
35         store.addMedia(cd3);
36         store.addMedia(dvd1);
37         store.addMedia(dvd2);
38         store.addMedia(dvd3);
39         store.addMedia(book1);
40         store.addMedia(book2);
```

```
2      1 usage  ▲ unknown
3      public static void viewStore(){
4          store.print();
5          storeMenu();
6      }
7
8      4 usages  ▲ unknown
9      public static void showMenu(){
10         System.out.println("AIMS: ");
11         System.out.println("-----");
12         System.out.println("1. View store");
13         System.out.println("2. Update store");
14         System.out.println("3. See current cart");
15         System.out.println("0. Exit");
16         System.out.println("-----");
17         System.out.println("Please choose a number: 0-1-2-3");
18
19         Scanner scanner = new Scanner(System.in);
20         int choice;
21         do {
22             choice = scanner.nextInt();
23             switch (choice) {
24                 case 1:
25                     viewStore();
26                     break;
27                 case 2:
28                     updateStore();
29                     break;
30                 case 3:
31                     currentCart();
32                     break;
33                 case 0:
34                     System.out.println("Exiting the application. Goodbye!");
35                     break;
36                 default:
37                     System.out.println("Invalid choice. Please choose again.");
38             }
39         } while (choice != 0);
40     }
```

```
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);
}
```

```
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:
                playMedia(item);
                break;
            case 0:
                return;
        }
    } while (choice != 0);
}
```

```
1 public static void playMedia(Media item){  
2     if(item instanceof CompactDisc){  
3         ((CompactDisc) item).play();  
4     } else if (item instanceof DigitalVideoDisc) {  
5         ((DigitalVideoDisc) item).play();  
6     } else {  
7         System.out.println("Cannot play media");  
8     }  
9 }  
10  
11 usage: ± unknown  
12 public static void cartMenu() {  
13     System.out.println("Options: ");  
14     System.out.println("-----");  
15     System.out.println("1. Filter medias in cart");  
16     System.out.println("2. Sort medias in cart");  
17     System.out.println("3. Remove media from cart");  
18     System.out.println("4. Play a media");  
19     System.out.println("0. Back");  
20     System.out.println("-----");  
21     System.out.println("Please choose a number: 0-1-2-3-4-5");  
22     Scanner scanner = new Scanner(System.in);  
23     int choice;  
24     do {  
25         showMenu();  
26         choice = scanner.nextInt();  
27         switch (choice) {  
28             case 1:  
29                 filterMedia();  
30                 break;  
31             case 2:  
32                 sortMedia();  
33                 break;  
34             case 3:  
35                 removeMediaFromCart();  
36                 break;  
37             case 4:  
38                 playMediaFromCart();  
39                 break;  
40             case 0:  
41                 System.out.println("Return to Store Menu");  
42         }  
43     } while (choice != 0);  
44 }
```

```
        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");
    }
}
```

```
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

```
*****CART*****
Compact Disc: CD Title 1 - Pop - Artist 1[Hello, World]: 15.99 $
Compact Disc: CD Title 2 - Rock - Artist 2[Hello, World]: 12.99 $
Compact Disc: CD Title 3 - Classical - Artist 3[Hello, World]: 20.99 $
DVD - DVD Title 1 - Action - null - 0: 19.99 $
DVD - DVD Title 2 - Comedy - null - 0: 16.99 $
DVD - DVD Title 3 - Drama - null - 0: 22.99 $
Book - Book Title 1 - Fiction [James, Jane] - 29.99 :$ 
Book - Book Title 2 - Non-fiction [James, Jane] - 24.99 :$ 
Book - Book Title 3 - Mystery [James, Jane] - 19.99 :$ 
*****
```

Figure 31: Kết quả

## 10. Update Class diagram

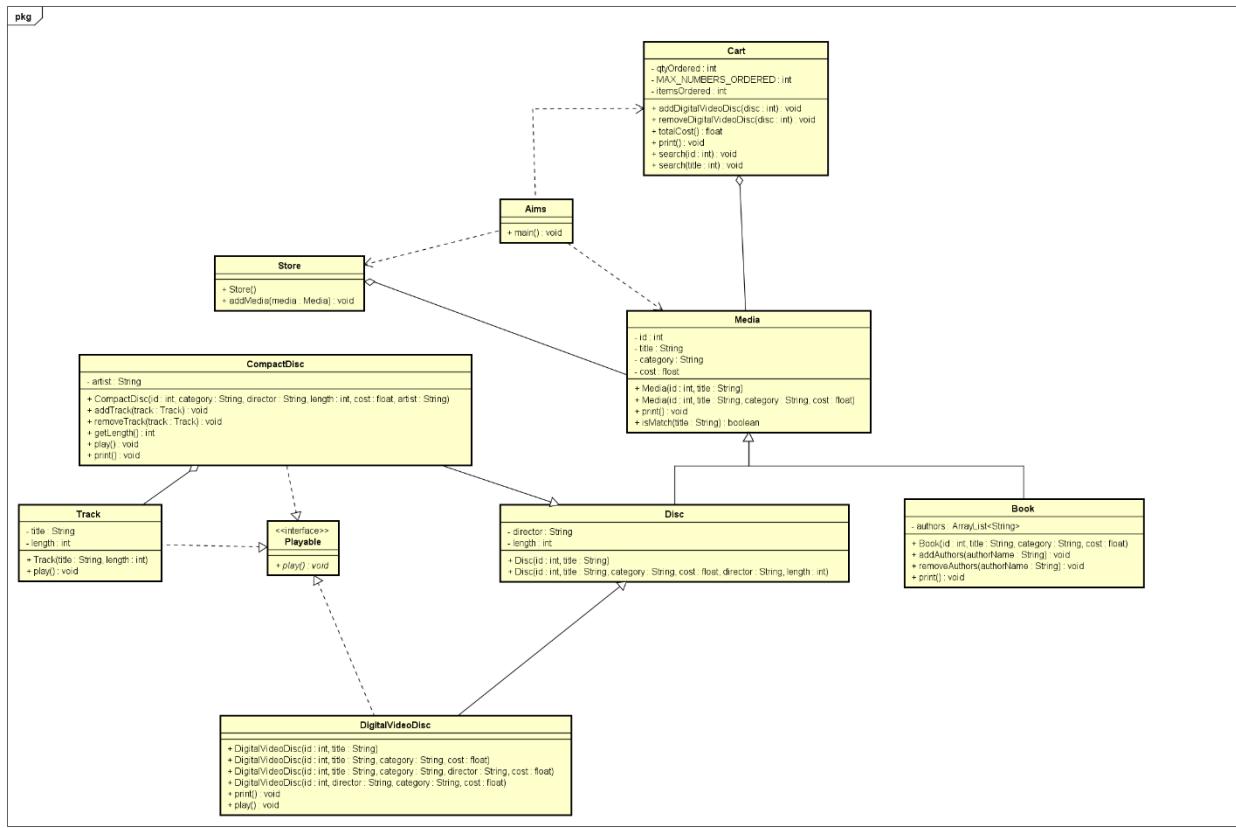


Figure 32: Class Diagram

## 11. Update user diagram

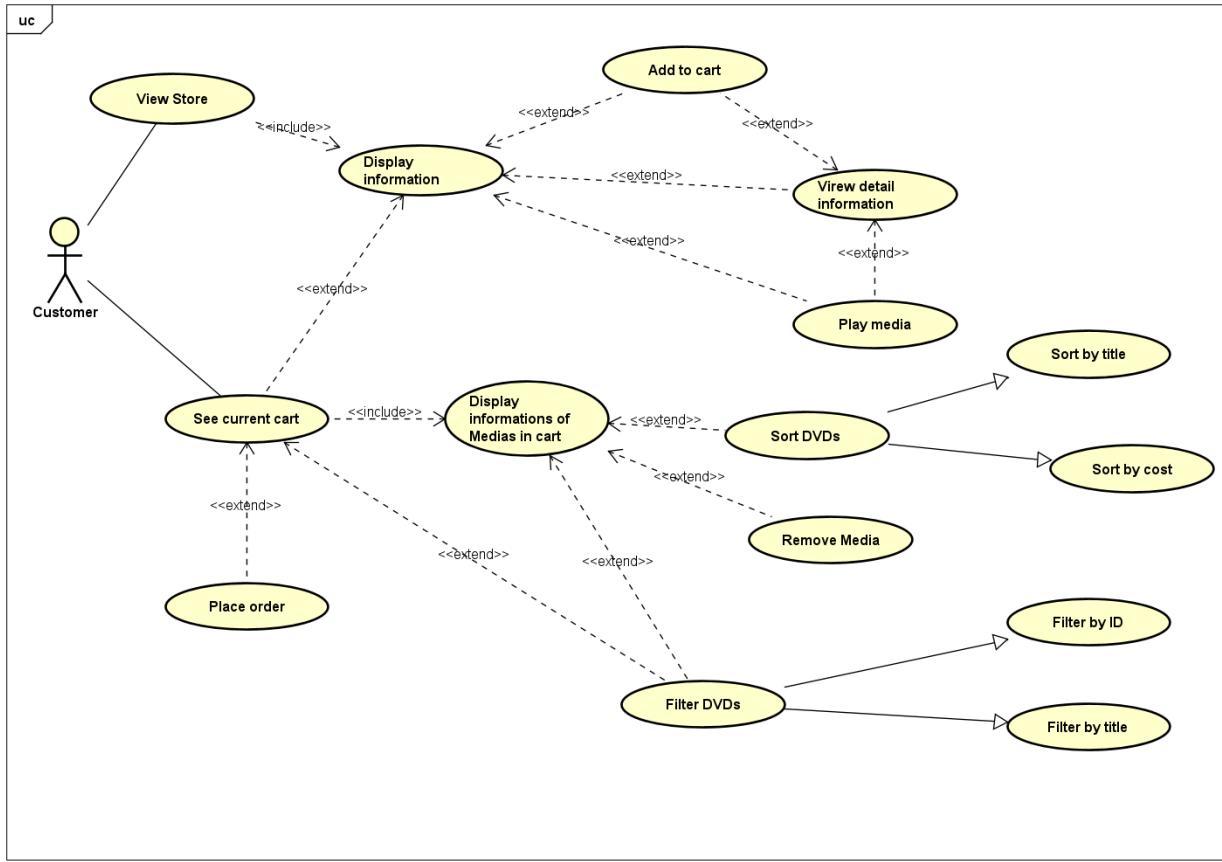


Figure 33: Customer Usecase Diagram

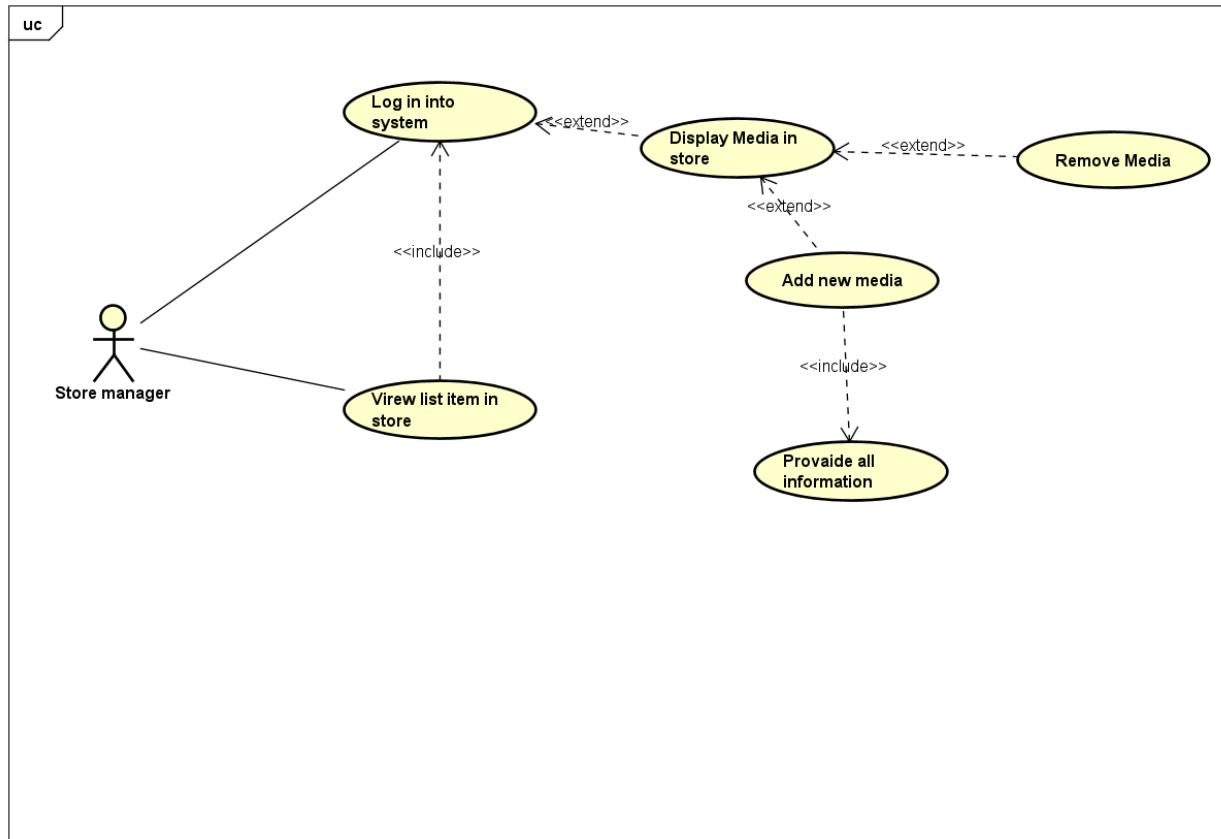


Figure 34: Manager Usecase Diagram