

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

BÁO CÁO THỰC HÀNH  
**IT3103-744527-2024.1**

BÀI THỰC HÀNH 04

Họ và tên sv: **Trịnh Hữu An**

MSSV: **20225593**

Lớp: **Việt Nhật 03 – K67**

GVHD: **Lê Thị Hoa**

TA: **Đặng Mạnh Cường**

Hà Nội 12/2024

## Table of Contents

1.	Create the Book class.....	5
2.	Creating the abstract Media class .....	6
3.	Creating the CompactDisc class.....	8
3.1.	Create the Disc class extending the Media clas.....	8
3.2.	Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track.....	11
3.3.	Open the CompactDisc class .....	12
4.	Create the Playable interface.....	13
5.	Update the Cart class to work with Media.....	14
6.	Update the Store class to work with Media .....	16
7.	Constructors of whole classes and parent classes .....	17
8.	Unique item in a list .....	18
9.	Polymorphism with <code>toString()</code> method .....	19
10.	Sort media in the car .....	20
11.	Create a complete console application in the Aims class .....	22
11.1.	Người dùng chọn 1: View store .....	22
11.1.1.	Người dùng tiếp tục chọn 1. See a media's details.....	23
11.1.2.	Người dùng chọn 2: Add a media to the cart.....	23
11.1.3.	Người dùng chọn 3: Play a media.....	24
11.1.4.	Người dùng chọn 4: See current cart.....	24
11.2.	Người dùng chọn 2: Update store .....	24
11.2.1.	Người dùng chọn 1: Add a media to the store.....	25
11.2.2.	Người dùng chọn 2: Remove a media from the store .....	26
11.3.	Người dùng chọn 3: See current cart.....	27
11.3.1.	Người dùng chọn 1: Filter medias in cart.....	28
11.3.2.	Người dùng chọn 2: Sort medias in cart.....	29
11.3.3.	Người dùng chọn 3: Remove media from cart .....	30
11.3.4.	Người dùng chọn 4: Play a media .....	30
11.3.5.	Người dùng chọn 5: Place order .....	31
12.	Class Diagram .....	31
13.	Class Diagram .....	32
14.	Answer Questions .....	34

## Table of Figures

Figure 1: Book Class.....	5
Figure 2: Media Class .....	7
Figure 3: Disc Class .....	8
Figure 4: DigitalVideoDisc Class .....	10
Figure 5: CompactDisc Class.....	10
Figure 6: Track Class .....	12
Figure 7: CompactDisc Class.....	13
Figure 8: Playable interface .....	13
Figure 9: Method play() của DigitalVideoDisc .....	13
Figure 10: Method play() của Track .....	14
Figure 11: Method play() của CompactDisc .....	14
Figure 12: Cart Class .....	16
Figure 13: Store Class .....	17
Figure 14: Constructor Track Class .....	17
Figure 15: Constructor CompactDisc Class.....	18
Figure 16: Constructor Media Class .....	18
Figure 17: Constructor Disc Class .....	18
Figure 18: Override equals() in Media Class .....	18
Figure 19: Override Equals in Track Class .....	19
Figure 20: Code mô phỏng Polymorphism.....	19
Figure 21: Override toString() in Media Class .....	19
Figure 22: Result Demo .....	20
Figure 23: Add the comparators as attributes of the Media class .....	20
Figure 24: MediaComparatorByCostTitle Class.....	21
Figure 25: MediaComparatorByTitleCost Class.....	21
Figure 26: Màn hình chính.....	22
Figure 27: vào trang View Store .....	22
Figure 28: See a media's details .....	23
Figure 29: Add to cart .....	23
Figure 30: Thêm media vào cart .....	23
Figure 31: Play a media .....	24
Figure 32: See current cart after sort.....	24
Figure 33: Update Store .....	24
Figure 34: Add a media to store.....	25
Figure 35: Resul.....	26
Figure 36: Remove a media .....	26
Figure 37: Result.....	27
Figure 38: See current cart .....	27
Figure 39: Media in cart.....	27
Figure 40: Filter Cart by Id .....	28
Figure 41: Filter Cart by Title .....	29
Figure 42: Sort Cart by Title .....	29
Figure 43: Sort Cart by Cost .....	29
Figure 44: Remove media from Cart .....	30
Figure 45: Play a media .....	30
Figure 46: Result.....	30

Figure 47: Place Order .....	31
Figure 48: Result.....	31
Figure 49: Class Diagram.....	32
Figure 50: Use Case Diagram .....	33

## 1. Create the Book class

```

package hust.soict.hedspi.aims.media;

import java.util.ArrayList;
import java.util.List;
//Trịnh Hữu An 20225593
public class Book extends Media { 8 usages
    private List<String> authors = new ArrayList<>(); 6 usages

    public Book() { no usages
        super();
    }

    public Book(int id, String title, String category, float cost, List<String> authors) { no usages
        super(id, title, category, cost);
        this.authors = authors;
    }

    public Book(int id, String title, String category, float cost) { no usages
        super(id, title, category, cost);
    }

    public Book(String title, String category, float cost) { 4 usages
        super(nbMedia++, title, category, cost);
    }

    public void addAuthor(String authorName) { 4 usages
        if(!authors.contains(authorName)) {
            authors.add(authorName);
            System.out.println("Author added");
        }
        else System.out.println("Author is already in the list");
    }

    public void removeAuthor(String authorName) { no usages
        if(authors.contains(authorName)) {
            authors.remove(authorName);
            System.out.println("Author removed");
        }
        else System.out.println("Author is not in the list");
    }

    @Override
    public String toString() {
        return "Book: " +
            "id = " + getId() +
            " - title = '" + getTitle() + '\'' +
            " - category = '" + getCategory() + '\'' +
            " - cost = " + getCost() +
            " - (List of) authors = " + authors;
    }
}

```

Figure 1: Book Class

## 2. Creating the abstract Media class

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

import java.util.Comparator;
import java.util.Objects;
//Trịnh Hữu An 20225593
public abstract class Media implements Comparable<Media> { 4 inheritors
    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost(); 1 usage
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle(); 1 usage
    private int id; 4 usages
    private String title; 8 usages
    private String category; 4 usages
    private float cost; 6 usages

    public static int nbMedia = 1; 6 usages

    public Media() {

    }

    public Media(int id, String title, String category, float cost) {
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public int getId() { 6 usages
        return id;
    }

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

    public String getTitle() {
        return title;
    }

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

    public String getCategory() { 4 usages
        return category;
    }
}
```

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

public float getCost() { 11 usages
|   return cost;
}

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

@Override 4 overrides
public String toString() {
    return "Media: " +
        "id=" + id +
        " - title='\" + title + '\"' +
        " - category='\" + category + '\"' +
        " - cost=" + cost;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Media media = (Media) o;
    return Objects.equals(title, media.title);
}

@Override 1 override
public int compareTo(Media other) {
    int titleComparison = this.title.compareTo(other.title);
    if (titleComparison != 0) return titleComparison;

    return Float.compare(other.cost, this.cost);
}
```

Figure 2: Media Class

### 3. Creating the CompactDisc class

#### 3.1. Create the Disc class extending the Media clas

```

package hust.soict.hedspi.aims.media;
//Trinh Huu An 20225593
public class Disc extends Media { 2 usages 2 inheritors
    private int length; 3 usages
    private String director; 3 usages

    public Disc() { super(); }

    public Disc(int id, String title, String category, float cost, int length, String director) { 7 usages
        super(id, title, category, cost);
        this.length = length;
        this.director = director;
    }

    public int getLength() { 11 usages 1 override
        return length;
    }

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

    public String getDirector() { 3 usages
        return director;
    }

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

    @Override 2 overrides
    public String toString() {
        return "Disc: " +
            "id=" + getId() +
            " - title = '" + getTitle() + '\'' +
            " - category = '" + getCategory() + '\'' +
            " - cost = " + getCost() +
            " - length = " + getLength() +
            " - director = '" + getDirector() + '\'';
    }
}

```

Figure 3: Disc Class

```

package hust.soict.hedspi.aims.media;
//Trịnh Hữu An 20225593
public class DigitalVideoDisc extends Disc implements Playable { 36 usages

    private static int nbDigitalVideoDiscs = 1; 2 usages

    public DigitalVideoDisc(String title) { 3 usages
        super(nbMedia++, title, category: null, cost: 0, length: 0, director: null);
    }

    public DigitalVideoDisc(String title, String category, float cost) { no usages
        super(nbMedia++, title, category, cost, length: 0, director: null);
    }

    public DigitalVideoDisc(String title, String category, String director, float cost) { no usages
        super(nbMedia++, title, category, cost, length: 0, director);
    }

    public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 8 usages
        super(nbMedia++, title, category, cost, length, director);
    }

    public static int getNbDigitalVideoDiscs() { no usages
        return nbDigitalVideoDiscs;
    }

    public static void setNbDigitalVideoDiscs(int nbDigitalVideoDiscs) { no usages
        DigitalVideoDisc.nbDigitalVideoDiscs = nbDigitalVideoDiscs;
    }

    @Override
    public String toString() {
        return "DigitalVideoDisc: " +
            "id = " + getId() +
            " - title = '" + getTitle() + '\'' +
            " - category = '" + getCategory() + '\'' +
            " - cost = " + getCost() +
            " - length = " + getLength() +
            " - director = '" + getDirector() + '\'';
    }

    @Override 7 usages
    public void play() {
        System.out.println("Playing DVD: " + getTitle());

        if (getLength() == 0) System.out.println("DVD cannot be played.");
        else System.out.println("DVD length: " + getLength());
    }
}

```

```
@Override  
public int compareTo(Media other) {  
    if (!(other instanceof DigitalVideoDisc)) return super.compareTo(other);  
  
    DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;  
  
    int titleComparison = this.getTitle().compareTo(other.getTitle());  
    if (titleComparison != 0) return titleComparison;  
  
    if (this.getLength() == otherDVD.getLength()) return Float.compare(otherDVD.getCost(), this.getCost());  
  
    return Integer.compare(otherDVD.getLength(), this.getLength());  
}
```

Figure 4: DigitalVideoDisc Class

```
package hust.soict.hedspi.aims.media;  
//Trinh Huu An 20225593  
  
public class CompactDisc extends Disc { 17 usages 4 related problems  
}
```

Figure 5: CompactDisc Class

### 3.2. Create the Track class which models a track on a compact disc and will store information including the title and length of the track.

```

package hust.soict.hedspi.aims.media;
// Trịnh Hữu An 20225593
import java.util.Objects;

public class Track implements Playable { 26 usages
    private String title; 6 usages
    private int length; 6 usages

    public Track() { no usages
        super();
    }

    public Track(String title, int length) { 9 usages
        this.title = title;
        this.length = length;
    }

    public String getTitle() {
        return title;
    }

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

    public int getLength() { 3 usages
        return length;
    }

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

    @Override 7 usages
    public void play() {
        System.out.println("Playing track: " + getTitle());
        if (getLength() == 0) System.out.println("Track cannot be played.");
        else System.out.println("Track length: " + getLength());
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Track track = (Track) o;
        return length == track.length && Objects.equals(title, track.title);
    }
}

```

```

    @Override
    public String toString() {
        return "Track: " +
            "title = '" + title + '\'' +
            " - length = " + length;
    }
}

```

Figure 6: Track Class

### 3.3. Open the CompactDisc class

```

package hust.soict.hedspi.aims.media;
//Trịnh Hữu An 20225593
import java.util.ArrayList;

public class CompactDisc extends Disc { 17 usages
    private String artist; 5 usages
    private ArrayList<Track> tracks = new ArrayList<Track>(); 9 usages

    public CompactDisc() { no usages
        super();
    }

    public CompactDisc(int id, String title, String category, float cost, String director, String artist) { no usages
        super(id, title, category, cost, length: 0, director);
        this.artist = artist;
    }

    public CompactDisc(int id, String title, String category, float cost, String director, String artist, ArrayList<Track> tracks) {
        super(id, title, category, cost, length: 0, director);
        this.artist = artist;
        this.tracks = tracks;
    }

    public CompactDisc(String title, String category, String artist, float cost) { 4 usages
        super(nbMedia++, title, category, cost, length: 0, director: null);
        this.artist = artist;
        this.tracks = tracks;
    }

    public String getArtist() { no usages
        return artist;
    }

    public void addTrack(Track track) { 9 usages
        if (tracks.contains(track)) System.out.println("Track is already in the list");
        else {
            tracks.add(track);
            System.out.println("Track added");
        }
    }
}

```

```

public void removeTrack(Track track) { no usages
    if (tracks.contains(track)) {
        tracks.remove(track);
        System.out.println("Track removed");
    } else System.out.println("Track is not in the list");
}

public int getLength() { 9 usages
    int length = 0;
    for (Track track : tracks) {
        length += track.getLength();
    }
    return length;
}

@Override
public String toString() {
    return "CompactDisc: " +
        "id = " + getId() +
        " - title = '" + getTitle() + '\'' +
        " - category = '" + getCategory() + '\'' +
        " - cost = " + getCost() +
        " - length = " + getLength() +
        " - director = '" + getDirector() + '\'' +
        " - artist = '" + artist + '\'' +
        " - tracks = " + tracks;
}
}

```

Figure 7: CompactDisc Class

## 4. Create the Playable interface

```

package hust.soict.hedspi.aims.media;
// Trịnh Hữu An 20225593
public interface Playable { 3 usages 3 implementations
    public void play(); 7 usages 3 implementations
}

```

Figure 8: Playable interface

Implement play() cho các class DigitalVideoDisc, Track, CompactDisc

```

@Override 7 usages
public void play() {
    System.out.println("Playing DVD: " + getTitle());

    if (getLength() == 0) System.out.println("DVD cannot be played.");
    else System.out.println("DVD length: " + getLength());
}

```

Figure 9: Method play() của DigitalVideoDisc

```

@Override 7 usages
public void play() {
    System.out.println("Playing track: " + getTitle());
    if (getLength() == 0) System.out.println("Track cannot be played.");
    else System.out.println("Track length: " + getLength());
}

```

Figure 10: Method play() của Track

```

@Override 4 usages
public void play() {
    System.out.println("Playing CompactDisc: " + this.getTitle());

    if (getLength() == 0) System.out.println("CD cannot be played!");
    else {
        System.out.println("CD length: " + getLength());
        System.out.println("Artist: " + this.getArtist());

        for (Track track : tracks) track.play();
    }
}

```

Figure 11: Method play() của CompactDisc

## 5. Update the Cart class to work with Media

Lớp Cart bây giờ cần có khả năng tương tác với các đối tượng DVD, CD và Book. Vì các lớp DVD, CD và Book đều kế thừa từ lớp Media, nên thay vì làm việc trực tiếp với từng lớp con, lớp cart chỉ cần giao tiếp với lớp Media là có thể hoạt động được với tất cả.

```

package hust.soict.hedspi.aims.cart;
//Trinh Huu An 20225593
import hust.soict.hedspi.aims.media.Media;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;

public class Cart { 6 usages
    public static final int MAX_NUMBERS_ORDERED = 20; 1 usage
    private ArrayList<Media> itemsOrdered = new ArrayList<~>(); 15 usages

    public void addMedia(Media media) { 4 usages
        if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(media);
            System.out.println("The media has been added");
        } else {
            System.out.println("The media is almost full");
        }
    }
}

```

```

public float totalCost() { 1 usage
    float totalCost = 0;
    for (Media media : itemsOrdered) {
        totalCost += media.getCost();
    }

    return totalCost;
}

public void printCart() { 4 usages
    System.out.println("*****CART*****");
    System.out.println("Ordered Items:");
    for (Media media : itemsOrdered) {
        System.out.println(media.toString());
    }
    System.out.println("Total cost: " + totalCost() + "$");
    System.out.println("*****");
}

public boolean search(int id) { no usages
    for (Media media : itemsOrdered) {
        if (media.getId() == id)
            return true;
    }

    return false;
}

public boolean search(String title) { no usages
    for (Media media : itemsOrdered) {
        if (media.getTitle().equals(title))
            return true;
    }

    return false;
}

public Media searchByTitle(String title) { 3 usages
    for (Media media : itemsOrdered) {
        if (media.getTitle().equals(title))
            return media;
    }

    return null;
}

public Media searchById(int id) { 1 usage
    for (Media media : itemsOrdered) {
        if (media.getId() == id)
            return media;
    }

    return null;
}

```

```

public void sortByTitle() { 1 usage
    Collections.sort(itemsOrdered, Media.COMPARE_BY_TITLE_COST);
    Iterator<Media> iterator = itemsOrdered.iterator();

    while (iterator.hasNext()) {
        System.out.println(iterator.next());
    }
}

public void sortByCost() { 1 usage
    Collections.sort(itemsOrdered, Media.COMPARE_BY_COST_TITLE);
    Iterator<Media> iterator = itemsOrdered.iterator();

    while (iterator.hasNext()) System.out.println(iterator.next());
}

public void removeMedia(Media media) { 1 usage
    if (itemsOrdered.contains(media)) {
        itemsOrdered.remove(media);
        System.out.println("The media has been removed");
    } else System.out.println("The media is not in the cart");
}

public void empty() { itemsOrdered.clear(); }
}

```

Figure 12: Cart Class

## 6. Update the Store class to work with Media

```

package hust.soict.hedspi.aims.store;
// Trinh Huu An 20225593
import hust.soict.hedspi.aims.media.Media;

import java.util.ArrayList;

public class Store { 6 usages
    private ArrayList<Media> itemsInStore = new ArrayList<>(); 6 usages

    public void addMedia(Media media) { 15 usages
        boolean existed = false;
        for (Media item : itemsInStore) {
            if (item.getTitle().equals(media.getTitle())) {
                existed = true;
                break;
            }
        }

        if (!existed) {
            itemsInStore.add(media);
            System.out.println("The media has been added in Store.");
        } else {
            System.out.println("The media is already in the store.");
        }
    }
}

```

```

public void removeMedia(Media media) { 2 usages
    boolean existed = false;
    for (Media item : itemsInStore) {
        if (item.getTitle().equals(media.getTitle())) {
            itemsInStore.remove(item);
            System.out.println("The media has been removed from Store.");
            existed = true;
            break;
        }
    }

    if (!existed) {
        System.out.println("The media is not in the store.");
    }
}

public void printStore() { 7 usages
    for (Media media : itemsInStore) {
        System.out.println(media.toString());
    }
}

public Media searchByTitle(String title) { 4 usages
    for (Media media : itemsInStore) {
        if (media.getTitle().equals(title)) {
            return media;
        }
    }

    return null;
}
}

```

Figure 13: Store Class

## 7. Constructors of whole classes and parent classes

```

public Track(String title, int length) { 9 usages
    this.title = title;
    this.length = length;
}

```

Figure 14: Constructor Track Class

```

public CompactDisc() { no usages
    super();
}

public CompactDisc(int id, String title, String category, float cost, String director, String artist) { no usages
    super(id, title, category, cost, length: 0, director);
    this.artist = artist;
}

public CompactDisc(int id, String title, String category, float cost, String director, String artist, ArrayList<Track> tracks) {
    super(id, title, category, cost, length: 0, director);
    this.artist = artist;
    this.tracks = tracks;
}

public CompactDisc(String title, String category, String artist, float cost) { 4 usages
    super(nbMedia++, title, category, cost, length: 0, director: null);
    this.artist = artist;
    this.tracks = tracks;
}

```

Figure 15: Constructor CompactDisc Class

Lớp Disc kế thừa lớp Media, khi đó lớp Media là lớp cha, lớp Disc là lớp con.

```

public Media() {
    // Constructor không tham số
}

public Media(int id, String title, String category, float cost) {
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}

```

Figure 16: Constructor Media Class

```

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

public Disc(int id, String title, String category, float cost, int length, String director) { 7 usages
    super(id, title, category, cost);
    this.length = length;
    this.director = director;
}

```

Figure 17: Constructor Disc Class

## 8. Unique item in a list

```

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Media media = (Media) o;
    return Objects.equals(title, media.title);
}

```

Figure 18: Override equals() in Media Class

```

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Track track = (Track) o;
    return length == track.length && Objects.equals(title, track.title);
}

```

Figure 19: Override Equals in Track Class

## 9. Polymorphism with `toString()` method

```

List<Media> mediae = new ArrayList<Media>();
ArrayList<Track> tracks = new ArrayList<Track>();
tracks.add(new Track( title: "Track 1", length: 1));
tracks.add(new Track( title: "Track 2", length: 2));
Media cd = new CompactDisc( id: 7, title: "CD2", category: "Music", cost: 10.3f, director: "Director 2", artist: "Long", tracks);

Media dvd = new DigitalVideoDisc( title: "The Lion King",
                                    category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);

List<String> authors = new ArrayList<>();
authors.add("Author 1");
authors.add("Author 2");
Media book = new Book( id: 8, title: "Book2", category: "Horror", cost: 4.5f, authors);

mediae.add(cd);
mediae.add(dvd);
mediae.add(book);

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

```

Figure 20: Code mô phỏng Polymorphism

```

@Override 4 overrides
public String toString() {
    return "Media: " +
        "id=" + id +
        " - title=' " + title + '\'' +
        " - category=' " + category + '\'' +
        " - cost=" + cost;
}

```

Figure 21: Override `toString()` in Media Class

```
CompactDisc: id = 7 - title = 'CD2' - category = 'Music' - cost = 10.3 - length = 3 - director = 'Director 2' - artist = 'Long' - tracks = [Track: title = 'Track 1' - length = 1, Track: title = 'Track 2' - length = 2]
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Book: id = 8 - title = 'Book2' - category = 'Horror' - cost = 4.5 - (List of) authors = [Author 1, Author 2]

Process finished with exit code 0
```

Figure 22: Result Demo

- Lớp Media là lớp cha cơ bản, được các lớp con như CompactDisc, DigitalVideoDisc, và Book kế thừa để đại diện cho các loại phương tiện cụ thể hơn. Khi chúng ta tạo các đối tượng từ các lớp con như cd, dvd, hoặc book và gán chúng vào biến kiểu Media, điều này được gọi là **upcasting**. Upcasting cho phép một đối tượng lớp con được xử lý như thể nó thuộc về lớp cha.
- Một ví dụ tiêu biểu về tính **đa hình động** là khi chúng ta thêm các đối tượng này vào một danh sách List<Media> và duyệt qua danh sách để in thông tin của từng phần tử. Phương thức toString() được gọi trên mỗi đối tượng trong danh sách, nhưng thay vì luôn sử dụng phiên bản của lớp cha, Java tự động sử dụng phiên bản được ghi đè trong lớp con tương ứng.
- Điều này nghĩa là, dù các phần tử đều được xử lý như kiểu Media, kết quả thực tế của phương thức toString() sẽ khác nhau tùy thuộc vào đối tượng thuộc lớp con nào. Do đó, ta không cần kiểm tra kiểu cụ thể của từng đối tượng trong danh sách nhưng vẫn nhận được thông tin phù hợp với loại phương tiện thực tế.
- Dù cả ba phần tử trong danh sách đều thuộc kiểu Media, phương thức toString() được gọi sẽ là phiên bản tương ứng với lớp con thực sự của từng đối tượng (CompactDisc, DigitalVideoDisc, hoặc Book). Điều này minh họa tính đa hình, nơi hành vi của một phương thức thay đổi dựa trên đối tượng thực sự đang được sử dụng.

## 10. Sort media in the car

```
public abstract class Media implements Comparable<Media> { 4 inheritors
    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost(); 1 usage
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle(); 1 usage
```

Figure 23: Add the comparators as attributes of the Media class

```

package hust.soict.hedspi.aims.media;

import java.util.Comparator;
//Trinh Huu An 20225593
public class MediaComparatorByCostTitle implements Comparator<Media> { 1 usage

    @Override
    public int compare(Media o1, Media o2) {
        int costComparison = Float.compare(o2.getCost(), o1.getCost());
        if (costComparison != 0) {
            return costComparison;
        }

        return o1.getTitle().compareTo(o2.getTitle());
    }
}

```

Figure 24: *MediaComparatorByCostTitle Class*

```

package hust.soict.hedspi.aims.media;
//Trinh Huu An 20225593
import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> { 1 usage

    @Override
    public int compare(Media o1, Media o2) {
        int titleComparison = o1.getTitle().compareTo(o2.getTitle());
        if (titleComparison != 0) {
            return titleComparison;
        }

        return Float.compare(o2.getCost(), o1.getCost());
    }
}

```

Figure 25: *MediaComparatorByTitleCost Class*

## 11. Create a complete console application in the Aims class

```

AIMS:

-----
1. View store
2. Update store
3. See current cart
0. Exit

-----
Please choose a number: 0-1-2-3:

```

Figure 26: Màn hình chính

### 11.1. Người dùng chọn 1: View store

```

DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
DigitalVideoDisc: id = 2 - title = 'Star Wars' - category = 'Science Fiction' - cost = 24.95 - length = 87 - director = 'George Lucas'
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
CompactDisc: id = 4 - title = 'Thriller' - category = 'Pop' - cost = 9.99 - length = 819 - director = 'null' - artist = 'Michael Jackson'
CompactDisc: id = 5 - title = 'Kind of Blue' - category = 'Jazz' - cost = 15.5 - length = 970 - director = 'null' - artist = 'Miles Davis'
CompactDisc: id = 6 - title = 'Bohemian Rhapsody' - category = 'Rock' - cost = 11.99 - length = 856 - director = 'null' - artist = 'Queen'
Book: id = 7 - title = 'Hannibal' - category = 'Crime Thriller' - cost = 8.99 - (List of) authors = [Thomas Harris]
Book: id = 8 - title = 'Red Dragon' - category = 'Crime Thriller' - cost = 9.11 - (List of) authors = [Thomas Harris]
Book: id = 9 - title = 'The Martian' - category = 'Science Fiction' - cost = 8.97 - (List of) authors = [Andy Weir]
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4:

```

Figure 27: vào trang View Store

### 11.1.1. Người dùng tiếp tục chọn 1. See a media's details

```

1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4:
1
Enter the title of the media to view details (Press 0 to return): The Lion King
Details: DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2:
|
```

Figure 28: See a media's details

```

Please choose a number: 0-1-2-3-4:
1
Enter the title of the media to view details (Press 0 to return): The Lion King
Details: DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2:
1
The media has been added
Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2:
|
```

Figure 29: Add to cart

### 11.1.2. Người dùng chọn 2: Add a media to the cart

```

Enter the title of the media to view details (Press 0 to return): 0
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4:
2
Enter the title of the media to add to cart (Press 0 to return): Aladin
The media has been added
Enter the title of the media to add to cart (Press 0 to return): |
```

Figure 30: Thêm media vào cart

### 11.1.3. Người dùng chọn 3: Play a media

```
Please choose a number: 0-1-2-3-4:  
3  
Enter the title of the media to play (Press 0 to return): Star Wars  
  
Playing DVD: Star Wars  
DVD length: 87  
Enter the title of the media to play (Press 0 to return):
```

Figure 31: Play a media

### 11.1.4. Người dùng chọn 4: See current cart

```
*****CART*****  
Ordered Items:  
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'  
DigitalVideoDisc: id = 3 - title = 'Aladdin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'  
Total cost: 38.940002$  
*****  
Options:  
-----  
1. Filter medias in cart  
2. Sort medias in cart  
3. Remove media from cart  
4. Play a media  
5. Place order  
0. Back  
-----  
Please choose a number: 0-1-2-3-4-5:
```

Figure 32: See current cart after sort

## 11.2. Người dùng chọn 2: Update store

```
AIMS:  
-----  
1. View store  
2. Update store  
3. See current cart  
0. Exit  
-----  
Please choose a number: 0-1-2-3:  
2  
Options:  
-----  
1. Add a media  
2. Remove a media  
0. Back  
-----  
Please choose a number: 0-1-2:  
1
```

Figure 33: Update Store

### 11.2.1. Người dùng chọn 1: Add a media to the store

```
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3:
2
Options:
-----
1. Add a media
2. Remove a media
0. Back
-----
Please choose a number: 0-1-2:
1
Enter the category of the media:
(1) Book
(2) CD
(3) DVD
(0) Back
2

Enter the title of the CD:
Huu An
Enter the category of the CD:
Romance
Enter the artist of the CD:
Trinh Huu An
Enter the cost of the CD:
10.02
Do you want to add tracks to your CD?
(1) Yes (0) No:
1
Enter the number of tracks:
1
Enter the title of track 1:
Huu-An
Enter the length of track 1:
900
Track added
The media has been added in Store.
```

Figure 34: Add a media to store

```

Current store:
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
DigitalVideoDisc: id = 2 - title = 'Star Wars' - category = 'Science Fiction' - cost = 24.95 - length = 87 - director = 'George Lucas'
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
CompactDisc: id = 4 - title = 'Thriller' - category = 'Pop' - cost = 9.99 - length = 819 - director = 'null' - artist = 'Michael Jackson'
CompactDisc: id = 5 - title = 'Kind of Blue' - category = 'Jazz' - cost = 15.5 - length = 970 - director = 'null' - artist = 'Miles Davis'
CompactDisc: id = 6 - title = 'Bohemian Rhapsody' - category = 'Rock' - cost = 11.99 - length = 856 - director = 'null' - artist = 'Queen'
Book: id = 7 - title = 'Hannibal' - category = 'Crime Thriller' - cost = 8.99 - (List of) authors = [Thomas Harris]
Book: id = 8 - title = 'Red Dragon' - category = 'Crime Thriller' - cost = 9.11 - (List of) authors = [Thomas Harris]
Book: id = 9 - title = 'The Martian' - category = 'Science Fiction' - cost = 8.97 - (List of) authors = [Andy Weir]
CompactDisc: id = 10 - title = 'Huu An' - category = 'Romance' - cost = 10.02 - length = 900 - director = 'null' - artist = 'Trinh Huu An'

Enter the category of the media:
(1) Book
(2) CD
(3) DVD
(0) Back

```

ory = 'Romance' - cost = 10.02 - length = 900 - director = 'null' - artist = 'Trinh Huu An' - tracks = [Track: title = 'Huu-An' - length = 900]

Figure 35: Result

### 11.2.2. Người dùng chọn 2: Remove a media from the store

```

Options:
-----
1. Add a media
2. Remove a media
0. Back
-----
Please choose a number: 0-1-2:
2

```

```

Enter the title of the media to remove from store (Press 0 to return): Huu An
The media has been removed from Store.
Huu An has been successfully removed from the store.

```

Figure 36: Remove a media

```

Current store:
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
DigitalVideoDisc: id = 2 - title = 'Star Wars' - category = 'Science Fiction' - cost = 24.95 - length = 87 - director = 'George Lucas'
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
CompactDisc: id = 4 - title = 'Thriller' - category = 'Pop' - cost = 9.99 - length = 819 - director = 'null' - artist = 'Michael Jackson'
CompactDisc: id = 5 - title = 'Kind of Blue' - category = 'Jazz' - cost = 15.5 - length = 970 - director = 'null' - artist = 'Miles Davis'
CompactDisc: id = 6 - title = 'Bohemian Rhapsody' - category = 'Rock' - cost = 11.99 - length = 856 - director = 'null' - artist = 'Queen'
Book: id = 7 - title = 'Hannibal' - category = 'Crime Thriller' - cost = 8.99 - (List of) authors = [Thomas Harris]
Book: id = 8 - title = 'Red Dragon' - category = 'Crime Thriller' - cost = 9.11 - (List of) authors = [Thomas Harris]
Book: id = 9 - title = 'The Martian' - category = 'Science Fiction' - cost = 8.97 - (List of) authors = [Andy Weir]

Enter the title of the media to remove from store (Press 0 to return):

```

Figure 37: Result

### 11.3. Người dùng chọn 3: See current cart

```

AIMS:
-----
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3:
3

```

Figure 38: See current cart

Giả sử trong cart lúc này sẽ có các media sau:

```

*****CART*****
Ordered Items:
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
Total cost: 38.940002$
*****
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:

```

Figure 39: Media in cart

### 11.3.1. Người dùng chọn 1: Filter medias in cart

```
*****
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
1
Filter by:
(1) Title
(2) Id

2
Enter the ID of the media to filter (press 0 to stop): 1
```

Result: DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'  
Enter the ID of the media to filter (press 0 to stop): |

Figure 40: Filter Cart by Id

```
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
1
Filter by:
(1) Title
(2) Id

1
Enter the title of the media to filter (press 0 to stop): The Lion King
```

```
Result: DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Enter the title of the media to filter (press 0 to stop): |
```

Figure 41: Filter Cart by Title

### 11.3.2. Người dùng chọn 2: Sort medias in cart

```
Result: DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Enter the title of the media to filter (press 0 to stop): 0
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
2
Sort medias in cart by:
(1) title
(2) cost
1
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
```

Figure 42: Sort Cart by Title

```
Please choose a number: 0-1-2-3-4-5:
2
Sort medias in cart by:
(1) title
(2) cost
2
DigitalVideoDisc: id = 1 - title = 'The Lion King' - category = 'Animation' - cost = 19.95 - length = 87 - director = 'Roger Allers'
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
```

Figure 43: Sort Cart by Cost

### 11.3.3. Người dùng chọn 3: Remove media from cart

```
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
3
Enter the title of the media to remove from cart (Press 0 to return): The Lion King
The media has been removed
The Lion King has been successfully removed from the cart.
Current cart:
*****CART*****
Ordered Items:
DigitalVideoDisc: id = 3 - title = 'Aladin' - category = 'Animation' - cost = 18.99 - length = 89 - director = 'Guy Ritchie'
Total cost: 18.99$
*****
Enter the title of the media to remove from cart (Press 0 to return): |
```

Figure 44: Remove media from Cart

### 11.3.4. Người dùng chọn 4: Play a media

```
*****
Enter the title of the media to remove from cart (Press 0 to return): 0
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5:
4
Enter the title of the media to play (Press 0 to return): Aladin
```

Figure 45: Play a media

```
Playing DVD: Aladin
DVD length: 89
Enter the title of the media to play (Press 0 to return): |
```

Figure 46: Result

### 11.3.5. Người dùng chọn 5: Place order

```
Options:  
-----  
1. Filter medias in cart  
2. Sort medias in cart  
3. Remove media from cart  
4. Play a media  
5. Place order  
0. Back  
-----  
Please choose a number: 0-1-2-3-4-5:  
5
```

Figure 47: Place Order

```
The order has successfully created.  
Your cart is now empty.  
Options:  
-----  
1. Filter medias in cart  
2. Sort medias in cart  
3. Remove media from cart  
4. Play a media  
5. Place order  
0. Back  
-----  
Please choose a number: 0-1-2-3-4-5:
```

Figure 48: Result

## 12. Class Diagram

## 13. Class Diagram

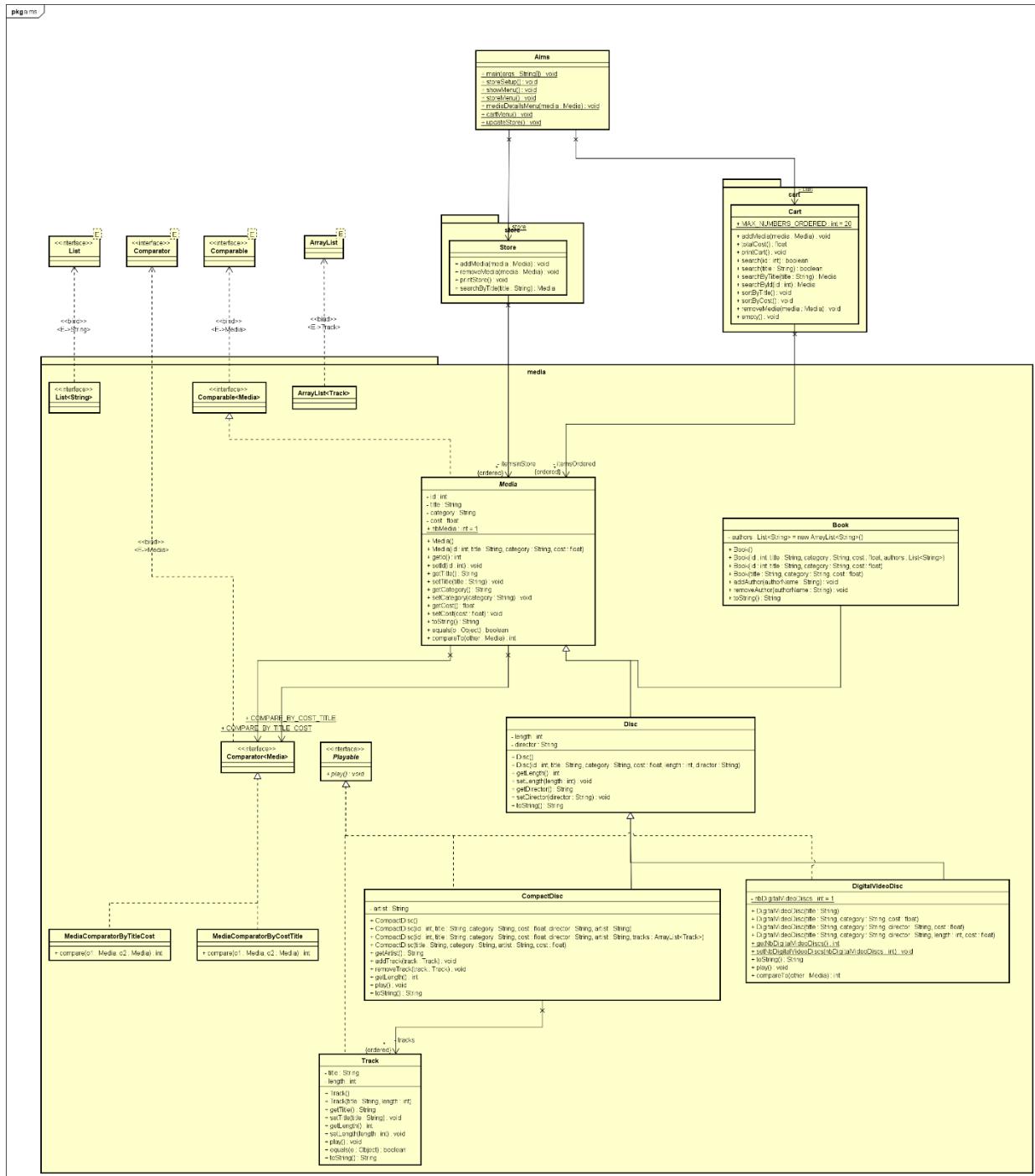


Figure 49: Class Diagram

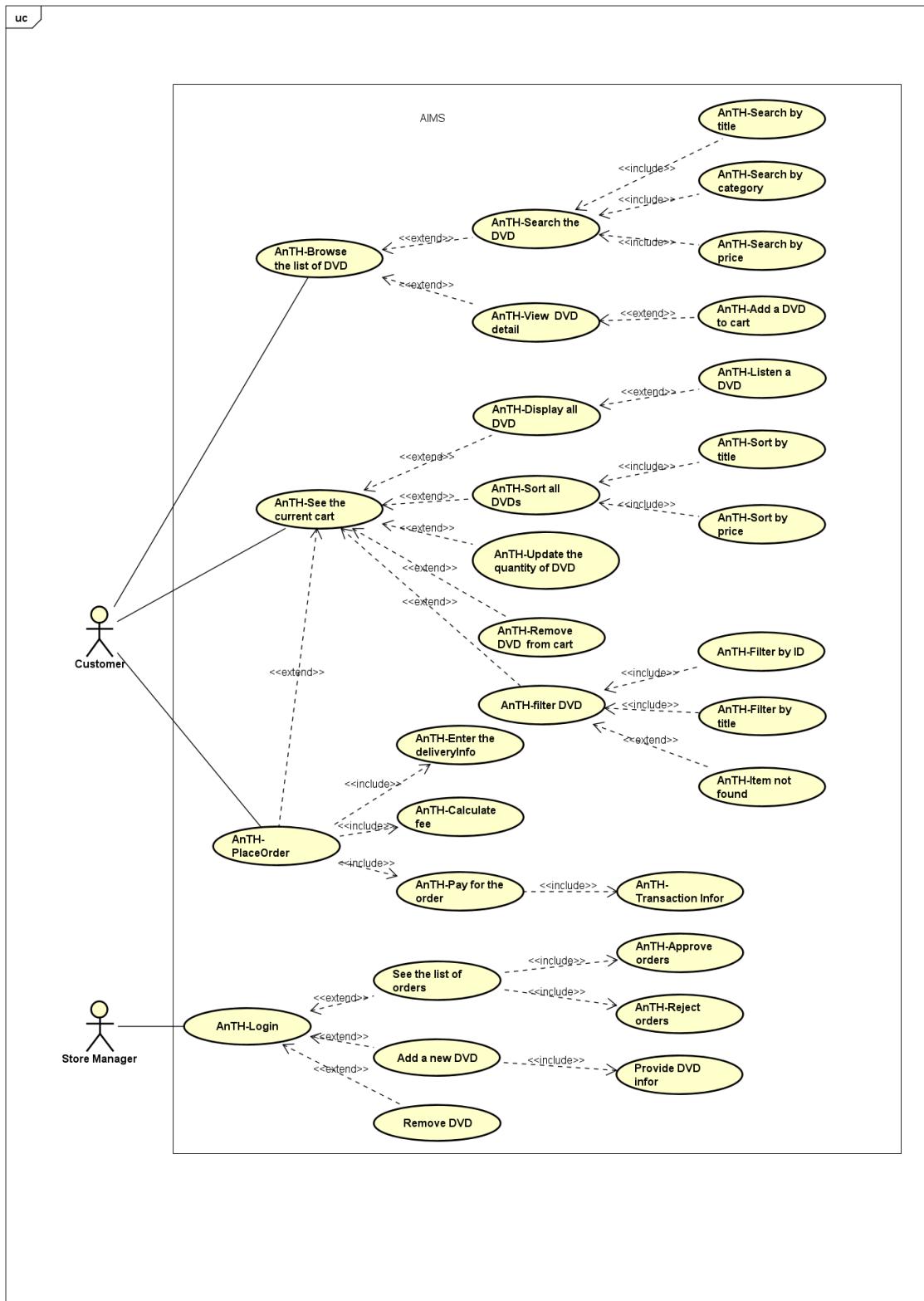


Figure 50: Use Case Diagram

## 14. Answer Questions

### 10. Unique Item in a List

Nếu đối tượng được truyền vào không phải là instance của lớp Media, việc so sánh sẽ không thể thực hiện được. Tuy nhiên, nếu đối tượng đó là instance của một lớp con kế thừa từ Media, phép so sánh vẫn hoạt động như bình thường.

Ví dụ minh họa trong lớp Aims:

```
Media book1 = new Book(5, "Book1", "Horror", 4.5f);
Media book2 = new Book(6, "Book1", "Scientific", 5.5f);
if (book1.equals(book2)) {
    System.out.println("Two objects are equal");
} else {
    System.out.println("Two objects are not equal");
}
```

**Kết quả:** Two objects are equal, vì cả hai đối tượng đều có tiêu đề (title) giống nhau, và phép so sánh chỉ dựa trên thuộc tính này.

### 12. Sort Media in the Cart

#### a. What class should implement the Comparable interface?

Lớp đại diện cho các đối tượng cần so sánh, chẳng hạn như Media hoặc các lớp con của nó (ví dụ DigitalVideoDisc, Book, CompactDisc), sẽ cần triển khai giao diện Comparable.

#### b. In those classes, how should you implement the compareTo() method be to reflect the ordering that we want?

Phương thức compareTo() cần được triển khai để phản ánh quy tắc sắp xếp mong muốn. Ví dụ, trong lớp Media, có thể sắp xếp theo tiêu đề trước, sau đó đến giá:

@Override

```
public int compareTo(Media other) {
    int titleComparison = this.title.compareTo(other.title);
    if (titleComparison != 0) {
        return titleComparison;
    }
}
```

```

    }
    return Float.compare(other.cost, this.cost);
}

```

**c. Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this Comparable interface approach?**

Không, giao diện Comparable chỉ cho phép định nghĩa một quy tắc sắp xếp duy nhất qua phương thức compareTo(). Nếu cần nhiều quy tắc sắp xếp, cần sử dụng Comparator, chẳng hạn:

- MediaComparatorByTitleCost
- MediaComparatorByCostTitle.

**d. Suppose the DVDs has a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?**

Có thể ghi đè phương thức compareTo() trong lớp DigitalVideoDisc:

`@Override`

```

public int compareTo(Media other) {
    if (!(other instanceof DigitalVideoDisc)) {
        return super.compareTo(other);
    }
}

```

DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;

```

int titleComparison = this.getTitle().compareTo(other.getTitle());
if (titleComparison != 0) {
    return titleComparison;
}

if (this.getLength() == otherDVD.getLength()) {
    return Float.compare(otherDVD.getCost(), this.getCost());
}

return Integer.compare(otherDVD.getLength(), this.getLength());
}

```

Ví dụ kiểm tra:

ArrayList<DigitalVideoDisc> DVDList = new ArrayList<>();

```
DVDList.add(new DigitalVideoDisc("The Lion King", "Animation", "Roger  
Allers", 87, 19.95f));  
DVDList.add(new DigitalVideoDisc("The Lion King", "Animation", "Roger  
Allers", 87, 25f));  
DVDList.add(new DigitalVideoDisc("Aladdin", "Animation", "Roger Allers", 90,  
12.0f));
```

```
System.out.println("Before sorting:");  
for (DigitalVideoDisc dvd : DVDList) {  
    System.out.println(dvd.getTitle() + " " + dvd.getLength() + " " + dvd.getCost());  
}
```

```
Collections.sort(DVDList);
```

```
System.out.println("\nAfter sorting:");  
for (DigitalVideoDisc dvd : DVDList) {  
    System.out.println(dvd.getTitle() + " " + dvd.getLength() + " " + dvd.getCost());  
}
```

Kết quả:

**Before sorting:**

The Lion King 87 19.95  
The Lion King 87 25.0  
Aladdin 90 12.0

**After sorting:**

Aladdin 90 12.0  
The Lion King 87 25.0  
The Lion King 87 19.95

Quy tắc sắp xếp được đảm bảo theo thứ tự: **Tiêu đề → Độ dài giảm dần → Giá.**