BÁO CÁO THỰC HÀNH LAB 4
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Table of Contents

Table of Figures

1. Create the abstract Media class

Media.java

```java
package lab04.AimsProject.Media;

import java.util.Comparator;

1 usage
class MediaComparatorByTitle implements Comparator<Media> {
    @Override
    public int compare(Media media1, Media media2) { return media1.getTitle().compareTo(media2.getTitle());
}

1 usage
class MediaComparatorByCost implements Comparator<Media> {
    @Override
    public int compare(Media media1, Media media2) { return Double.compare(media1.getCost(), media2.getCost
}

4 inheritors
public abstract class Media{
    // Attribute
    4 usages
    private int id;
    6 usages
    private String title;
    3 usages
    private String category;
    3 usages
```

Media.java

```java
    private String category;
    3 usages
    private float cost;
    1 usage
    public static final Comparator<Media> COMPARE_BY_TITLE =
            new MediaComparatorByTitle();
    1 usage
    public static final Comparator<Media> COMPARE_BY_COST    =
            new MediaComparatorByCost();

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

    // Method to print a media
    3 overrides
    public void print() {
    }
```

Minh_Huyen > src > lab04 > AimsProject > Media > Media.java 1:33 CRLF UTF-8 4 spaces

```java
    public void print() {
    }

    // Method to finds out if the corresponding disk is a match given the title.
    2 usages
    public boolean isMatch(String title) { return title.equals(this.title); }

    @Override
    public boolean equals(Object o) {
        if (o instanceof Media media) {
            return title.equals(media.getTitle());
        }
        return false;
    }

    // Getter and Setter
    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getTitle() { return title; }

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

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

nh_Huyen > src > lab04 > AimsProject > Media > Media.java 1:33 CRLF UTF-8 4 spaces

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

    1 usage
    public void setCategory(String category) { this.category = category; }

    public float getCost() { return cost; }

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

Figure 1 Media abstract class code

2. Create the Disc class extending the Media class

```java
package lab04.AimsProject.Media;

2 usages  2 inheritors
public class Disc extends Media{
    // Attribute
    3 usages
    private String director;
    3 usages
    private int length;

    // Constructor
    1 usage
    public Disc(int id, String title) { super(id, title); }

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

    // Getter and Setter
    2 usages
    public String getDirector() { return director; }

    1 usage
    public void setDirector(String director) { this.director = director; }

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

    public void setLength(int length) { this.length = length; }
}
```

Figure 2 Disc class code

3. Create the Playable interface

```java
package lab04.AimsProject.Media;

3 usages  3 implementations
public interface Playable {
    // Method to play
    3 usages  3 implementations
    public void play();
}
```

Figure 3 Playable interface code

4. Create the Book class

```java
package lab04.AimsProject.Media;

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

13 usages
public class Book extends Media{
    // Attribute
    7 usages
    private List<String> authors = new ArrayList<~>();

    // Constructor
    9 usages
    public Book(int id, String title, String category, float cost) { super(id, title, category, cost); }

    // Method to add an author
    no usages
    public void addAuthor(String authorName) {
        int indexOfAuthor = authors.indexOf(authorName);
        if (indexOfAuthor == -1) {
            System.out.println("Author is already in the list");
            return;
        }
        authors.add(authorName);
        System.out.println("Added");
    }
```

```java
    }

    // Method to remove an author
    no usages
    public void removeAuthor(String authorName) {
        int indexOfAuthor = authors.indexOf(authorName);
        if (indexOfAuthor == -1) {
            System.out.println("Author is absent in the list");
            return;
        }
        authors.remove(indexOfAuthor);
        System.out.println("Removed");
    }

    // Method to print a book
    @Override
    public void print() {
        System.out.print(getId() + ". Book - "
                + getTitle() + " - "
                + getCategory() + " - ");
        for (String author: authors) {
            System.out.print(author + " - ");
        }
        System.out.println(getCost() + "$");
    }
```

```java
    }

    // Getter and Setter
    no usages
    public List<String> getAuthors() { return authors; }

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

Figure 4 Book class code

5. Create the Track class

```java
package lab04.AimsProject.Media;

7 usages
public class Track implements Playable{
    // Attribute
    5 usages
    private String title;
    5 usages
    private int length;

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

    // Method to play a track
    3 usages
    public void play() {
        System.out.println("Playing track: " + title);
        System.out.println("Track length : " + length);
    }

    @Override
    public boolean equals(Object o) {
```

```java
    @Override
    public boolean equals(Object o) {
        if (o instanceof Track track) {
            return title.equals(track.getTitle()) && length == track.getLength();
        }
        return false;
    }

    // Getter and Setter
    public String getTitle() { return title; }

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

    public int getLength() { return length; }

    public void setLength(int length) { this.length = length; }
}
```

Figure 5 Track class code

6. Create the CompactDisc class

```java
package lab04.AimsProject.Media;

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

10 usages
public class CompactDisc extends Disc implements Playable{
    // Attribute
    5 usages
    private String artist;
    6 usages
    private List<Track> tracks = new ArrayList<~>();

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

    // Method to add a track
    no usages
    public void addTrack(Track track) {
        int indexOfTrack = tracks.indexOf(track);
        if (indexOfTrack == -1) {
```

```java
        if (indexOfTrack == -1) {
            System.out.println("Track is already in the list");
            return;
        }
        tracks.add(track);
        System.out.println("Added");
    }

    // Method to remove a track
    no usages
    public void removeTrack(Track track) {
        int indexOfTrack = tracks.indexOf(track);
        if (indexOfTrack == -1) {
            System.out.println("Track is absent in the list");
            return;
        }
        tracks.remove(indexOfTrack);
        System.out.println("Removed");
    }

    // Method to get the length of CD
    @Override
    public int getLength() {
        int length = 0;
        for (Track track: tracks) {
```

```java
        for (Track track: tracks) {
            length += track.getLength();
        }
        setLength(length);
        return length;
    }

    // Method to play a track
    3 usages
    public void play() {
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("CD artist: " + artist);
        System.out.println("CD length: " + this.getLength());
        for (Track track: tracks) {
            track.play();
        }
    }

    // Method to print a cd
    @Override
    public void print() {
        System.out.println(getId() + ". CD - "
                + getTitle() + " - "
                + getCategory() + " - "
                + getDirector() + " - "
```

```java
        System.out.println(getId() + ". CD - "
                + getTitle() + " - "
                + getCategory() + " - "
                + getDirector() + " - "
                + artist + " - "
                + getLength() + ": "
                + getCost() + "$");
    }

    // Getter and Setter
    no usages
    public String getArtist() { return artist; }

    no usages
    public void setArtist(String artist) { this.artist = artist; }
}
```

Figure 6 CompactDisc class code

7. Update the DigitalVideoDisc

```java
package lab04.AimsProject.Media;

public class DigitalVideoDisc extends Disc implements Playable{
    // Constructor
    public DigitalVideoDisc(int id, String title) { super(id, title); }

    public DigitalVideoDisc(int id, String title, String category, float cost) {
        this(id, title);
        this.setCategory(category);
        this.setCost(cost);
    }

    public DigitalVideoDisc(int id, String title, String category, String director, float cost) {
        this(id, title, category, cost);
        this.setDirector(director);
    }

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

    // Method to print a dvd
    @Override
    public void print() {
        System.out.println(getId() + ". DVD - "
```

```java
    @Override
    public void print() {
        System.out.println(getId() + ". DVD - "
                + getTitle() + " - "
                + getCategory() + " - "
                + getDirector() + " - "
                + getLength() + ": "
                + getCost() + "$");
    }

    // Method to play a dvd
    3 usages
    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
}
```

Figure 7 DVD class code

8. Update the Cart class

```java
package lab04.AimsProject;

import lab04.AimsProject.Media.DigitalVideoDisc;
import lab04.AimsProject.Media.Media;
import java.util.ArrayList;
import java.util.List;

4 usages
public class Cart {
    // Attribute
    10 usages
    private List<Media> itemsOrdered = new ArrayList<~>();
    5 usages
    private int numOfDVDs;

    // Constructor
    2 usages
    public Cart() { numOfDVDs = 0; }

    // Method to add a new media
    2 usages
    public void addMedia(Media media) {
        if (itemsOrdered.contains(media)) {
            System.out.println("Media is already in the list");
            return;
        }
```

```java
        }
        itemsOrdered.add(media);
        if (media.getClass() == DigitalVideoDisc.class) {
            numOfDVDs++;
        }
        System.out.println("Added");
    }

    // Method to remove a media
    1 usage
    public void removeMedia(Media media) {
        // Search for media
        int indexOfRemoved = itemsOrdered.indexOf(media);

        // If not found
        if (indexOfRemoved == -1) {
            System.out.println("Not found");
            return;
        }

        // Remove
        itemsOrdered.remove(indexOfRemoved);
        if (media.getClass() == DigitalVideoDisc.class) {
            numOfDVDs--;
        }
```

```java
        }

        // Notify
        System.out.println("Removed");
    }

    // Method to calculate the total cost
    1 usage
    public double totalCost() {
        float cost = 0;
        for (Media media: itemsOrdered) {
            cost += media.getCost();
        }

        return Math.round(cost * 100.0) / 100.0;
    }

    // Method to print the list of ordered items of a cart,
    // the price of each item, and the total price
    4 usages
    public void printCart() {
        System.out.println("**********************CART**********************");
```

```java
    public void printCart() {
        System.out.println("**********************CART**********************");
        System.out.println("Ordered Items:");
        for (Media media : itemsOrdered) {
            media.print();
        }
        System.out.println("Total cost: " + totalCost());
        System.out.println("*************************************************");
    }

    // Method to search for media in the cart by ID and display the search results.
    1 usage
    public Media searchByID(int id) {
        for (Media media: itemsOrdered) {
            if (media.getId() == id) {
                return media;
            }
        }
        System.out.println("Not found!");
        return null;
    }

    // Method to search for media in the cart by title.
    3 usages
    public Media searchByTitle(String title) {
        for (Media media: itemsOrdered) {
```

```java
    // Method to search for media in the cart by title.
    3 usages
    public Media searchByTitle(String title) {
        for (Media media: itemsOrdered) {
            if (media.isMatch(title)) {
                return media;
            }
        }
        System.out.println("Not found!");
        return null;
    }

    // Method to sort by title and print
    1 usage
    public void sortByTitle() {
        itemsOrdered.sort(Media.COMPARE_BY_TITLE);
        printCart();
    }

    // Method to sort by cost and print
    1 usage
    public void sortByCost() {
        itemsOrdered.sort(Media.COMPARE_BY_COST);
        printCart();
    }

    // Getter and Setter

    1 usage
    public int getNumOfDVDs() { return numOfDVDs; }

    no usages
    public void setNumOfDVDs(int numOfDVDs) { this.numOfDVDs = numOfDVDs; }
}
```

Figure 8 Cart class code

9. Update the Store class

```java
package lab04.AimsProject;

import lab04.AimsProject.Media.Media;

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

4 usages
public class Store {
    // Attribute
    8 usages
    private List<Media> itemsInStore = new ArrayList<~>();

    // Constructor
    2 usages
    public Store() {
    }

    // Method to add a media
    20 usages
    public void addMedia(Media media) {
        if (itemsInStore.contains(media)) {
            System.out.println("Media is already in the list");
            return;
        }
        itemsInStore.add(media);
        System.out.println("Added");
    }

    // Method to remove a media
    1 usage
    public void removeMedia(Media media) {
        // Search for disc
        int indexOfRemoved = itemsInStore.indexOf(media);

        // If not found
        if (indexOfRemoved == -1) {
            System.out.println("Not found");
            return;
        }

        // Remove
        itemsInStore.remove(indexOfRemoved);

        // Notify
        System.out.println("Removed");
    }

    // Method to print all item in store
    1 usage
```

```java
    // Method to print all item in store
    1 usage
    public void printStore() {
        System.out.println("**********************STORE**********************");
        System.out.println("Items in store:");
        for (Media media : itemsInStore) {
            media.print();
        }
        System.out.println("*************************************************");
    }

    // Method to search for media in the store by title.
    4 usages
    public Media searchByTitle(String title) {
        for (Media media: itemsInStore) {
            if (media.isMatch(title)) {
                return media;
            }
        }
        System.out.println("Not found!");
        return null;
    }
```

h_Huyen > src > lab04 > AimsProject > © Store 3:38 CRLF UTF-8 4 spaces

```java

    // Getter and Setter
    no usages
    public List<Media> getItemsInStore() { return itemsInStore; }

    no usages
    public void setItemsInStore(List<Media> itemsInStore) { this.itemsInStore = itemsInStore; }
}
```

Figure 9 Store class code

10. Update the Aims class

```java
package lab04.AimsProject;

import lab04.AimsProject.Cart;
import lab04.AimsProject.Media.*;

import java.util.Scanner;

public class Aims {
    // Create a new store
    16 usages
    static Store store = new Store();
    // Create a new cart
    13 usages
    static Cart cart = new Cart();
    1 usage
    public static void showMenu() {
        int command;
        do {
            Scanner scanner = new Scanner(System.in);
            // Show store
            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");
```

```java
            System.out.println("0. Exit");
            System.out.println("--------------------------------");
            System.out.println("Please choose a number: 0-1-2-3");
            command = scanner.nextInt();

            // If chose View store
            if (command == 1) {
                store.printStore();
                storeMenu();
            }

            // If chose 2
            if (command == 2) {
                updateStoreMenu();
            }

            // If chose 3
            if (command == 3) {
                cart.printCart();
                cartMenu();
            }
        } while (command != 0);
    }

    1 usage
```

```java
1 usage
public static void updateStoreMenu() {
    Scanner scanner = new Scanner(System.in);
    // Show menu
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Add a media to store");
    System.out.println("2. Remove a media from cart");
    System.out.println("0. Back");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2");
    int command = scanner.nextInt();

    // If chose 1
    if (command == 1) {
        System.out.println("Added to store");
    }

    // If chose 2
    if (command == 2) {
        removeMediaFromStore();
    }
}

1 usage
```

```java
1 usage
public static void storeMenu() {
    Scanner scanner = new Scanner(System.in);
    // Show menu
    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");
    int command = scanner.nextInt();

    // If chose 1
    if (command == 1) {
        Media media;
        do {
            System.out.println("Enter the title of the media: ");
            scanner.nextLine();
            String title = scanner.nextLine();
            media = store.searchByTitle(title);
        } while (media == null);
        media.print();
```

```java
            media.print();
            mediaDetailsMenu(media);
        }

        // If chose 2
        if (command == 2) {
            addMediaToCart();
        }

        // If chose 3
        if (command == 3) {
            Media media;
            do {
                System.out.println("Enter the title of the media: ");
                scanner.nextLine();
                String title = scanner.nextLine();
                media = store.searchByTitle(title);
            } while (media == null);
            playAMedia(media);
        }

        // If chose 4
        if (command == 4) {
            cart.printCart();
            cartMenu();
```

```java
            cartMenu();
        }
    }

    1 usage
    public static void mediaDetailsMenu(Media media) {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Add to cart");
        if (!(media instanceof Book)) {
            System.out.println("2. Play");
        }
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.print("Please choose a number: 0-1");
        if (!(media instanceof Book)) {
            System.out.println("-2");
        }
        int command = scanner.nextInt();

        // If chose 1
        if (command == 1) {
            cart.addMedia(media);
```

```java
        if (command == 1) {
            cart.addMedia(media);
        }

        // If chose 2
        if (command == 2) {
            playAMedia(media);
        }
    }

    2 usages
    public static void cartMenu() {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        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("5. Place order");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
```

Figure 10 Aims class code

11. Demo Aims

```
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
1
**********************STORE*********************
Items in store:
1. DVD - Inception - Science Fiction - Christopher Nolan - 148: 19.99$
2. DVD - The Dark Knight - Action - Christopher Nolan - 152: 17.99$
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
3. CD - Random Access Memories - Electronic - Daft Punk - Daft Punk - 0: 15.99$
4. CD - 25 - Pop - Adele - Adele - 0: 14.99$
8. CD - Lover - Pop - Taylor Swift - Taylor Swift - 0: 17.99$
5. Book - The Silent Patient - Thriller - 14.95$
6. Book - Where the Crawdads Sing - Mystery - 12.99$
9. Book - Educated - Memoir - 16.95$
10. Book - Becoming - Autobiography - 22.99$
************************************************
```

```
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
1
Enter the title of the media:
Interstellar
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
1
Added
```

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

Process finished with exit code 0
```

Figure 11 Aims demo

12. Update the class diagram

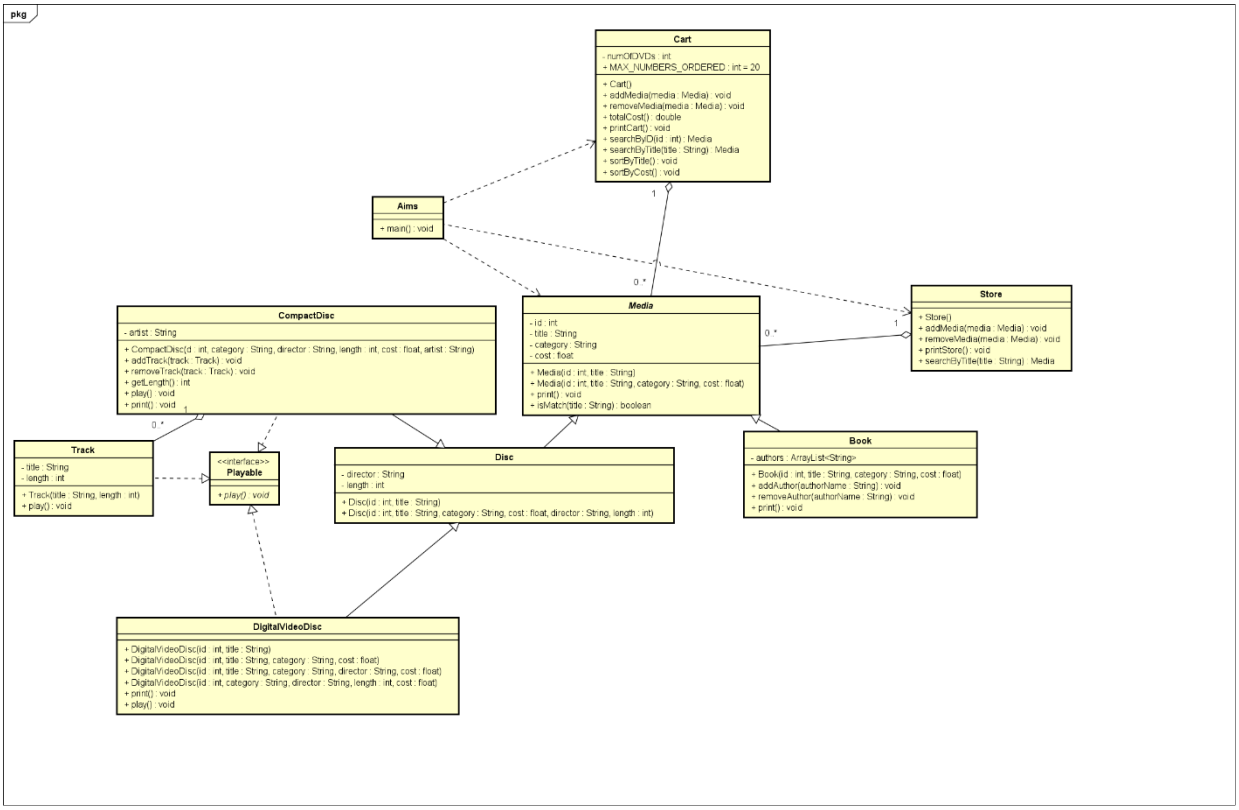

Figure 12 Class diagram

13. Update the Usecase diagram

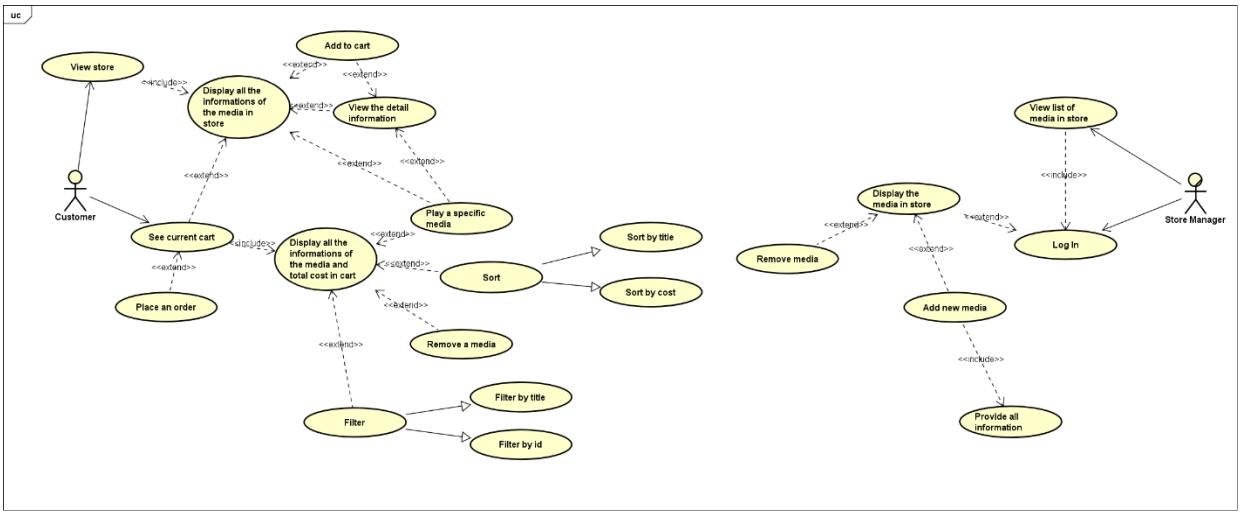

Figure 13 Usecase diagram

14. Answer the questions

- Trường hợp cần so sánh các Media với nhau bằng cách implement Comparable thay vì Comparator: Thay vì tạo class riêng cho từng Comparator, ta cần cho class Media implement interface Comparable.
- Ví dụ:

```java
public abstract class Media implements Comparable<Media> {
    new *
    @Override
    public int compareTo(Media otherMedia) {
        // Compare by title
        return this.title.compareTo(otherMedia.getTitle());
    }
}
```

Figure 14 Question code

- Có thể cài đặt như sau:

```java
public abstract class Media implements Comparable<Media> {
    new *
    @Override
    public int compareTo(Media otherMedia) {
    // Compare by title first
    int titleComparison = this.title.compareTo(otherMedia.getTitle());

    // If titles are equal, compare by cost
    return (titleComparison == 0) ? Float.compare(this.cost, otherMedia.getCost()) : titleComparison;
}
```

Figure 15 Question code

- Cài đặt như sau:

```java
public class DVD extends Media {
    // Override compareTo for DVDs
    new *
    @Override
    public int compareTo(Media otherMedia) {
        if (otherMedia instanceof DVD) {
            DVD otherDVD = (DVD) otherMedia;
            // Compare by title first
            int titleComparison = getTitle().compareTo(otherDVD.getTitle());
            // If titles are equal, compare by decreasing length
            if (titleComparison == 0) {
                int lengthComparison = Integer.compare(otherDVD.getLength(), getLength());

                // If lengths are equal, compare by cost
                return (lengthComparison == 0) ? Float.compare(getCost(), otherDVD.getCost())
                        : lengthComparison;
            }
            return titleComparison;
        } else {
            // For non-DVD media, use the default comparison (title then cost)
            return super.compareTo(otherMedia);
        }
    }
}
```

Figure 16 Question code