**BÁO CÁO OOP LAB 04**

**Họ tên: Trương Ngọc Mai – 20225879**

1. **Answer Question:**
   1. Which classes are aggregates of other classes(9)?

* Store aggregates Media (like DigitalVdDisc, CompactDisc, etc.)

Cart aggregates Media (like DigitalVdDisc, CompactDisc, etc.)

CompactDisc aggregates Track

Book aggregates authors (in the form of a list of strings)

* 1. If the passing object is not an instance of Media, what happens(10)?
* If the method is defined to accept a parameter of type Media (e.g., addMedia(Media media)), then only instances of Media or its subclasses can be passed. If you attempt to pass an object that is not a Media instance or a subclass of Media, a compilation error will occur. This is because Java enforces type safety at compile time.
  1. **(11)**
* What class should implement the Comparable interface?
* Media (or possibly subclasses of Media, such as DigitalVdDisc, CompactDisc, Book, etc.):
* In those classes, how should you implement the compareTo()method be to reflect the ordering that we want? (exg
* First compare by title: We use String's compareTo() method to compare the titles. If the titles are different, it will return a value indicating the order (< 0 if this.title is lexicographically less than other.title, > 0 if it is greater).
* If the titles are the same, it compares by cost using Float.compare(), which ensures the correct comparison of floating-point numbers.
* 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?
* Yes, we can have two ordering rules, but the Comparable interface only defines one natural ordering for the class.
* 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?
* If DVDs need to have a specific ordering rule (by title, then decreasing length, and then cost), we can override the compareTo() method in the DigitalVdDisc class (or any other class where this behavior is required).
* We can implement multiple ordering rules by defining different comparators for different sorting criteria (e.g., by title then cost, or by cost then title).
* For DigitalVdDisc, we can override the compareTo() method to implement a custom sorting order, like title, then decreasing length, then cost.
* Using Comparable or Comparator, we can achieve flexible sorting for our media objects based on different attributes.

1. **Source code:**

* *Book.java*

*A screenshot of a computer program

Description automatically generated*

* *Media.java*

*A screenshot of a computer program

Description automatically generated*

*A computer screen shot of colorful text

Description automatically generated*

* *CompactDisc.java*

*A computer screen shot of a program code

Description automatically generated* *A screen shot of a computer program

Description automatically generated* A computer code on a black background

Description automatically generated

* *Disc.java*

*A computer screen shot of text

Description automatically generated*

* *Track.java*

*A screen shot of a computer program

Description automatically generated* *A screen shot of a computer program

Description automatically generated*

* *Interface playable()*
* *updateCart*

*A screen shot of a computer program

Description automatically generated*

* *updateStore*

*A computer code on a black background

Description automatically generated*

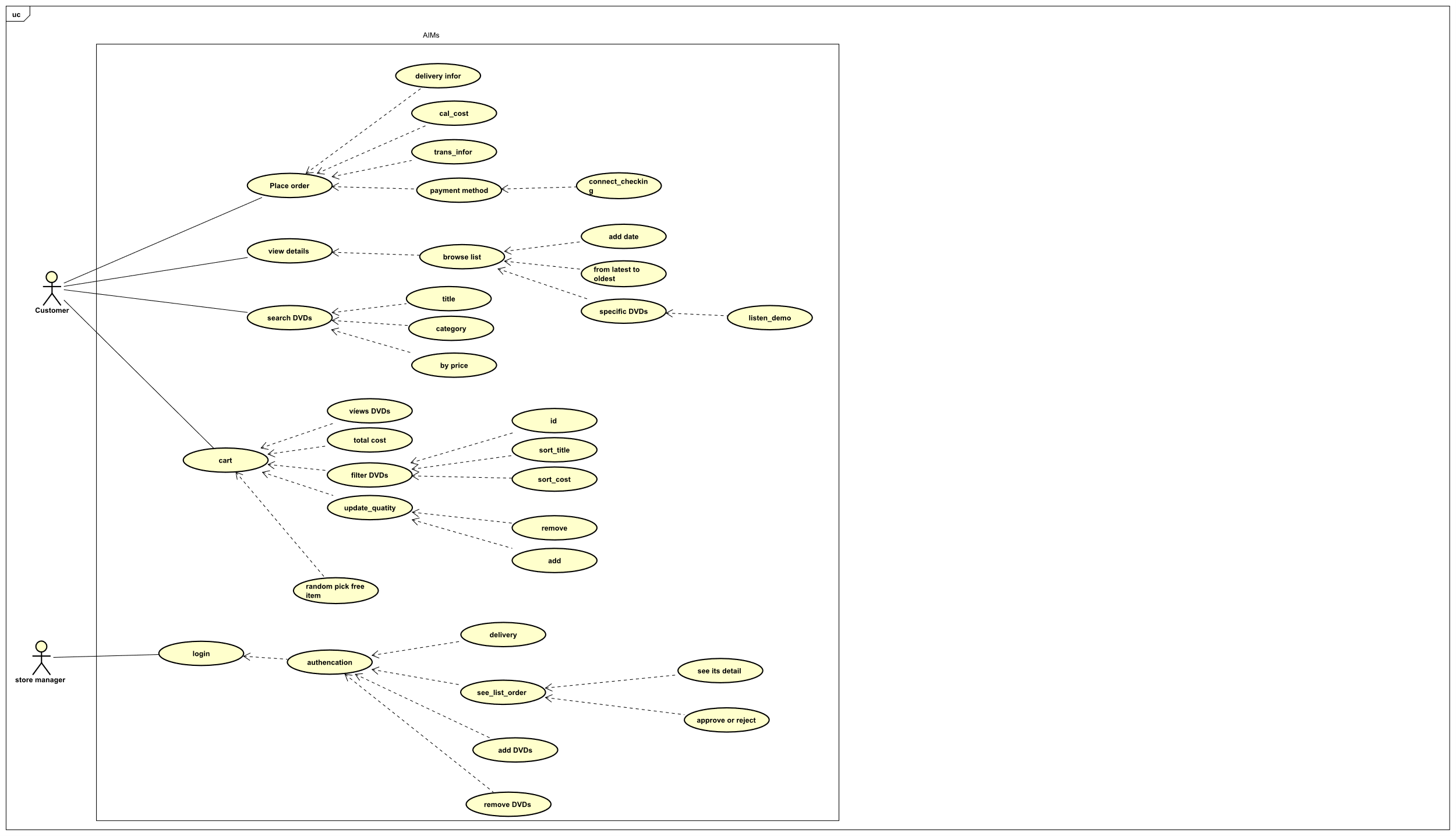
* *MediaCOmparator.java(cost & title)*

*A computer screen shot of text

Description automatically generated*

1. **Usecase Diagram and Class Diagram**

* **Usecase Diagram**



* **Class Diagram**

A screenshot of a computer

Description automatically generated