Here's are my classes so far:

LibraryMembers - Users for the library system. Could be librarians or customers. Librarians have greater priveleges than customers.

Attributes:

-memberID - Unique identifier for a library member. The library member could either be a librarian (who is an administrator) or a Customer.

-name - Name of the Library Member

-contact - contact information of the Library Member

Librarians (derived from LibraryMembers class)

Attributes:

-adminID - Unique identifier for librarian

Methods:

- createAccount() - Has the ability to create a Customer Account

- adminCheckout() - checks out items to library members and record the date it was borrowed and its due date

-viewCustomer() - enables librarian to view the list of customers, the items they own, and their due dates

CustomerAccount (Derived LibraryMember)

Attributes:

-customerID - unique identifier for customer

Methods:

- listItems() - Lists all items available

- checkout() - adds item from the list of items available to cart to be checked out (must be approved by librarian). Once the item is successfully checkout out, it goes to the personal inventory of the customer, which contains all owned items.

- search () - allows users to find items by title, author or type. It should also display the availability of each of the items found.

-returnItem() - enables customers to return books they own (this would be available within the Inventory section)

LibraryItem (this has a relationship to Librarians and CustomerAccount but I don't know what kind of relationship there is.)

Attributes:

itemID – unique identifier for the item

title – title of the item

author – author of the item

pubDate – publication date of the item

status – status of the item, whether it is available or not

Book, Magazine, Journal, DigitalMedia (All derive from the LibraryItem Class)

Cart – This class is meant to store the books added by the customer to be checked out. The items must be approved by the librarian in order to be checked out. (Please help me identify the relationship between cart and the other classes)

Inventory – This class is meant to contain the items the customer owns. (Please help me identify the relationship between Inventory and the other classes)

Solution:  
  
Feedback and suggestions on the class relationships:

LibraryMembers:

This class represents the base class for both librarians and customers. It should contain common attributes and methods that both types of members share. You can have a polymorphic relationship here, meaning you can have a vector of LibraryMembers to hold both librarians and customers.

Librarians (derived from LibraryMembers class):

This class can inherit from LibraryMembers, which indicates an "is-a" relationship, i.e., a librarian is a type of library member.

CustomerAccount (Derived from LibraryMember):

Similar to librarians, CustomerAccount can inherit from LibraryMember. Customers are a specialized form of library members.

LibraryItem:

LibraryItem can have a composition relationship with Cart and Inventory. Each LibraryItem may be a part of a Cart when a customer is checking it out, and it can also be part of the Inventory when a customer owns it.

Book, Magazine, Journal, DigitalMedia (All derive from the LibraryItem Class):

These derived classes represent specific types of library items, and they should inherit from the base LibraryItem class. This inheritance represents an "is-a" relationship.

Cart:

You can have a composition relationship between Cart and LibraryItem. A Cart can contain multiple LibraryItems when a customer is in the process of checking them out.

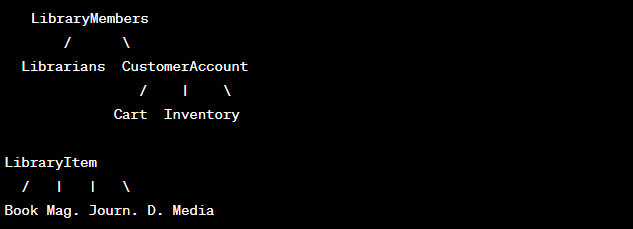
Inventory:

Similarly, Inventory can have a composition relationship with LibraryItem. It represents the items a customer owns. When a customer checks out items, they get added to their inventory.

Library (not explicitly mentioned but would be useful):

You might want to consider a Library class that contains a collection of LibraryItem objects, allowing you to manage the library's inventory collectively. This class can also include the search functionality.

Here's a simplified class diagram to help you visualize the relationships:



Relationships

The relationship between CustomerAccount, Cart, and Inventory can be described as follows:

Composition: A composition relationship represents a "whole-part" relationship, where one class (the whole) is composed of or contains other classes (the parts). In your case:

CustomerAccount may have a composition relationship with Cart and Inventory. This means that a CustomerAccount is composed of a Cart and an Inventory, and the existence of a CustomerAccount is closely tied to the existence of its Cart and Inventory. When a CustomerAccount is created, it would typically include an associated Cart and an Inventory.

The Cart and Inventory belong to and are managed by a CustomerAccount. When a customer creates an account, they are given a Cart to hold items they want to check out, and an Inventory to store items they own. The Cart and Inventory are conceptually part of the CustomerAccount.

This relationship signifies that the lifetime of the Cart and Inventory is dependent on the existence of a CustomerAccount. When the CustomerAccount is deleted or no longer exists, the associated Cart and Inventory would typically be deleted or cease to exist as well.