NYU School of Continuing and Professional Studies
Course Title: Object Oriented Analysis and Design
Fall 2011

Homework #8:
For this assignment, you will create an object-oriented design for an online store. In order to create the design, you will have to do the object-oriented analysis, based on the requirements.

System Requirements
You are to design a simple online (web-based) club that sells books, CD's and DVD's. The users are presented with your online catalog showing each product. You provide them with a virtual shopping cart that they use to keep their selections before "check-out", i.e. make a purchase.

Since your online store is a club, each customer will have to log-in before they can make a purchase. Each customer will have a unique customer number. They also have the option to review their all of their purchases to date. You will keep track of each customer's first name, last name, street, city, state, zip, work phone number and home phone number. In addition, each customer will have a login name and password. When they log-in, their supplied login and password are validated. Once validated, you'll display a welcoming message, or display an error message if their user id or password was incorrect. Your system will track all customer transactions, i.e. customer purchases. A customer may purchase any number of different products at one time. When the customer is completed, he/she logs off. Your system will keep track of each time the customer uses your site.

Your website will offer thousands of books, CD's and DVD's. There are many different CD's for sale. All are sold for $15.00. There are many different DVD's for sale. All are sold for $20.00. Likewise, there are many different books on sale. The books vary in price. You need to manage your inventory to be able to know when you have enough stock to fulfill a transaction. If you are out of stock, a notification should be given to the customer. Each product, i.e. individual book, CD or DVD, has a unique product id, title, price and description.

Once a user "checks-out", they will supply a credit-card number only. You will validate that the card is okay after requesting certain information from the user. This information may be stored so the user will not have to re-enter it again.

For your users, your system will provide a list of reports/searches:
Transaction history by date
List current contents of shopping cart
Search for any product by title
Search for book by title
Search for CD by title

Search for DVD by title

The searches will return the product's information, in addition to the number available.

You system will store all transaction, inventory and member data in a relational database. You must incorporate the relational database into your OO design.

Project Requirements

This is *not* an implementation exercise. This project assumes you are all familiar with how web pages look, not how they are implemented. There is no requirement for any HTML or Java etc. Likewise, issues such as detailed session management, security; etc. are to be ignored.

The description above is limited to the user interaction. You may ignore the details of how new inventory is added, how customers are added (or removed), etc. Assume there are methods to do these items (and any not implied in the requirements). Feel free to include such methods where appropriate, or ignore altogether. Whichever you choose should be clearly articulated in your design overview.

The object of the exercise is for you to do the analysis (based on the requirements) and create a design. As a result, you will have to make certain assumptions regarding the screen elements that would need to be present if you were actually going to implement this. Hint: define methods to be triggered by user selections. Assume that as a result of these methods, data is made available from the user interface to wherever it is required. Also, define methods that will present your user interface.

You may need to supplement the information in the requirements. Feel free to add information that will make your design more complete, without changing the requirements.

You should submit the following:

1. Class diagram(s): diagrams should show all relationships between the classes in your system
2. Any fields/additions/assumptions made