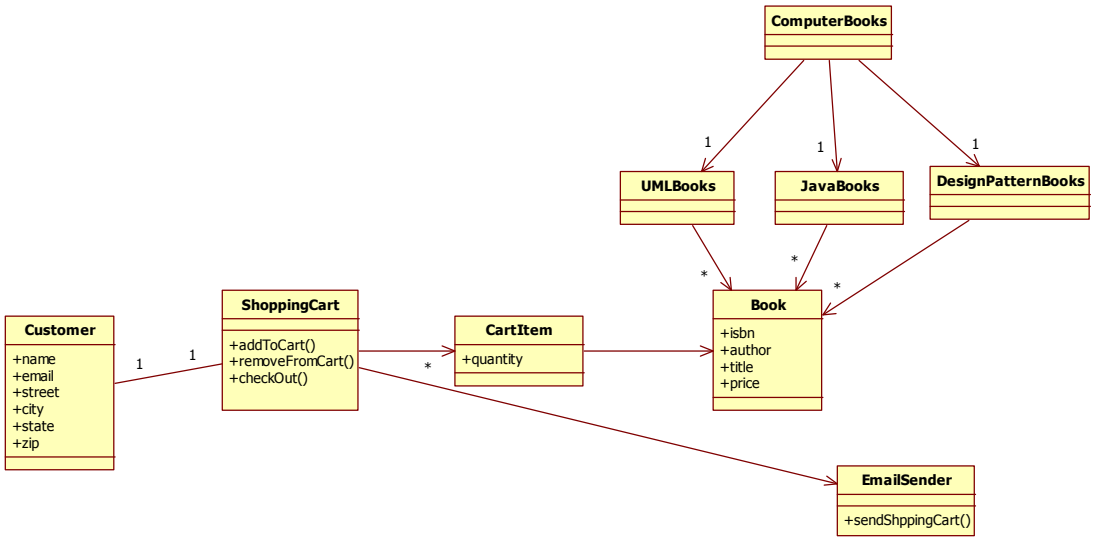
Webshop Framework UML Lab

Two of our clients have asked us to build a webshop application. We have designed and implemented a webshop for selling computer books , and another webshop for selling DVD's. Now a third client asks us to build another webshop for selling TV's, so we decided to design a general webshop framework .

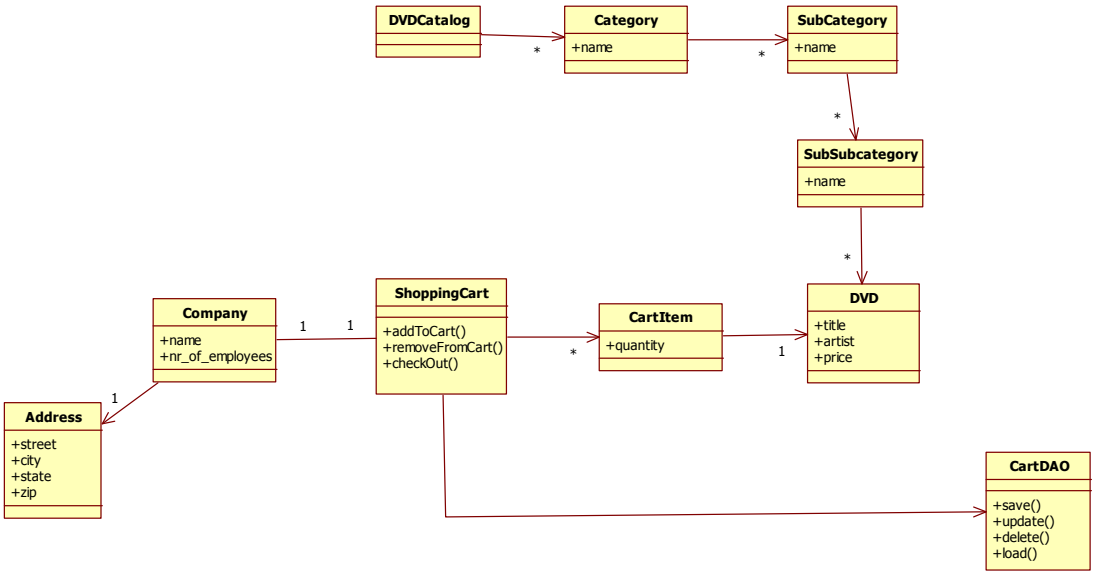
Below are given the designs of the 2 webshops we have already build:

Computer books webshop:



This webshop sells 3 types of books: UML books, Java books and Design Pattern books. We can browse through all the available books, and then add them to the shoppingcart. When we checkout the shoppingcart, the shoppingcart is send by the EmailSender to an external Order system.

DVD webshop:



At the DVD webshop only companies can buy DVD's. The webshop contains a catalog of DVD's that contains many Categories, and then Categories contain SubCategories and then SubCategories contain SubSubcategories. SubSubCategories contain DVD's. So we can browse through the DVDCatalog to find DVD's. We have for example the main categories Action, Drama and Children. The category Children has sub-categories like Walt-Disney, New Releases, Pre-school, etc. The sub-category Walt-Disney has sub-sub-categories like Animated and Non-animated.

When you checkOut() the ShoppingCart, the ShoppingCart will be saved in the database with the CartDAO object. The CartDAO object knows how to save the ShoppingCart in the database.

Design a framework for webshop applications with the following requirements:

1. Customers can be companies or individual persons.
2. We can sell different types of products
3. We can have hierarchies of categories and products of any level deep
4. We want undo/redo functionality for the addToCart() and removeFromCart() methods on the ShoppingCart
5. If you call checkOut() on the ShoppingCart, then all kind of things can happen. For example, the ShoppingCart can be saved in the database, it can be send by email, it can be written to a log file, etc. Apply the **Observer** pattern such that the ShoppingCart does not need to know what needs to happen when the checkOut() method is called.

Design the general webshop framework. Draw the class diagram of the DVD webshop using the framework. Show clearly which objects are within the framework, and which objects are outside the framework.