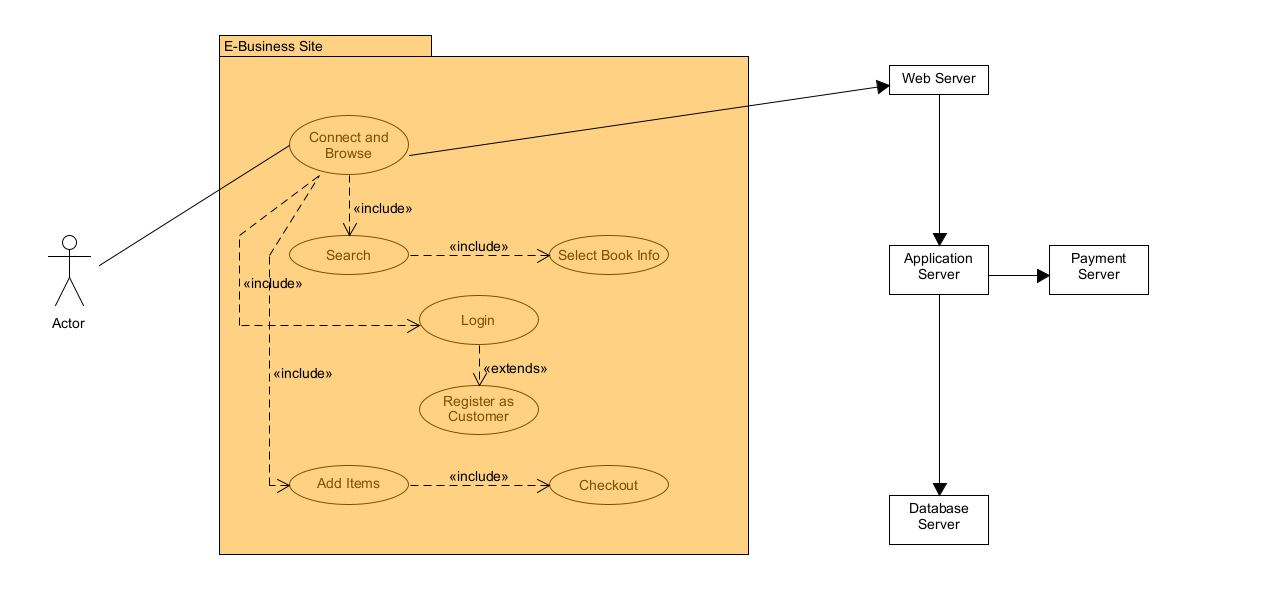
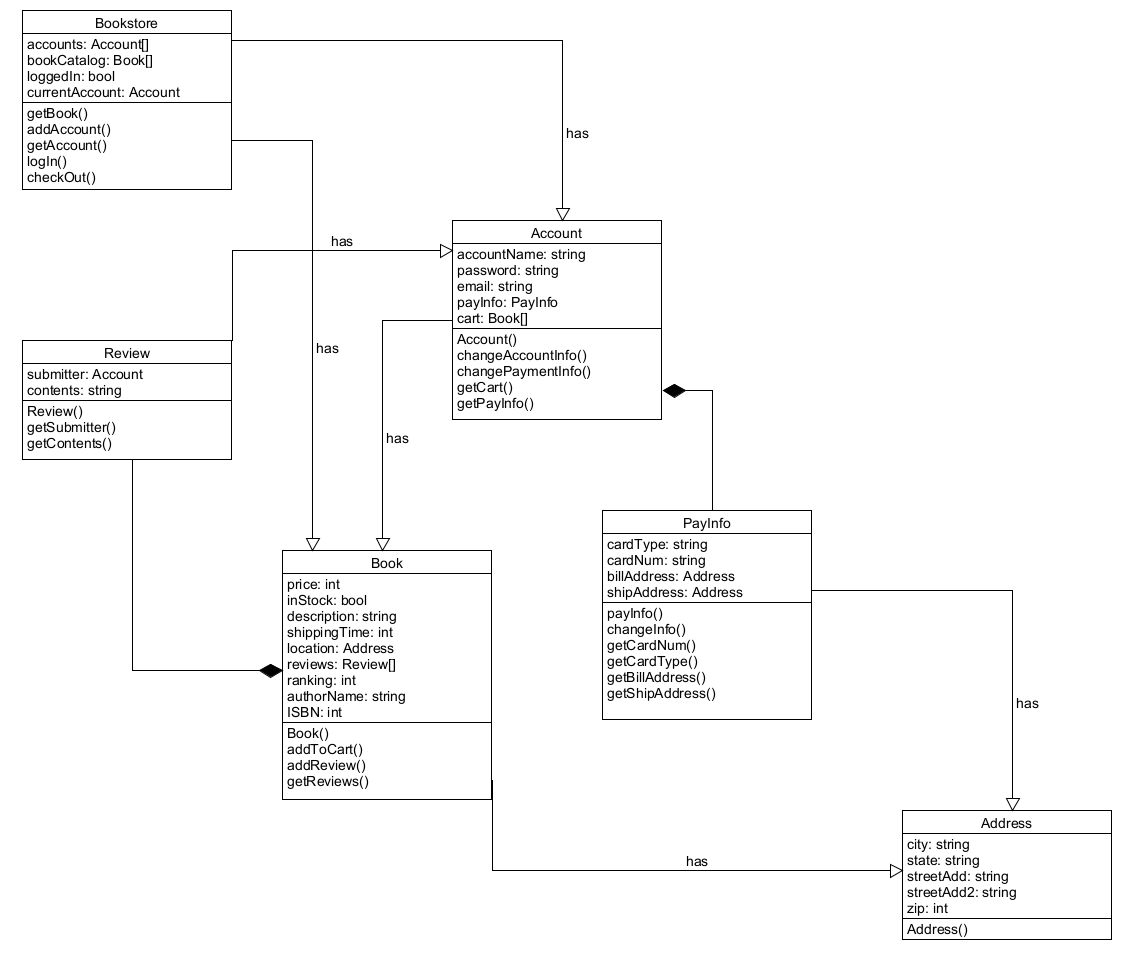
I chose to put the site as one package with the actions within it serving as a connection between the outside users and the servers, and I represented the connections between the servers on the outside of it, as it’s not the part the user directly interacts with. As I understood it, all the other actions are dependent on the basic “connect and browse” action, thus the <<includes>> arrows indicating said dependence and the user and server infrastructure solely being connected to the main function. This function, and by extension all the others, are served directly by the web server, executing the commands of the application server, which reads and writes data to the payment and database servers if the function requires it. I put the registration function as an extension of the login function because in most websites registration is an optional path when on the login page and lacking an account. 



I reasoned that as a storefront much of the experience revolves around your account and its interactions. To this end the account has attributes like its username and password, and some attributes that are objects of their own like payment info, addresses and a list of books in the cart. Depending on the language I’d have used dictionaries for some of these but I’m going to use objects for clarity, and because in languages like javascript they are indistinguishable from each other. The book object has all of the info a user could want about it, and contains the functions to add to the user’s cart and to review it. Reviews are also their own object to contain info about the submitter. These are contained at the highest level within the Bookstore object, which contains the catalog of books for sale, information about whether a user is logged in and if so, into which account, and the log in, check out, and book retrieval options. Some of these functions could and likely would be broken down into sub functions when actually coding a project (like finding a book by ISBN, by title, by author, etc), but I figured since these diagrams are supposed to show high level functionality I’d gloss over that a bit. I would do the total price calculation after retrieving the cart list from the account, within the checkOut function, which is why my following sequence diagram may look a bit simple.

