Kennedy Uzoho

SNHU CS-255 (Evaluating an Object Model)

1/28/2022

Hamp Craft’s Object Model Evaluation

The different functions of Hamp Craft’s storefront includes register(), addCartItem(), updateQuantity(), updateShippingInfo(), updateCatalog(), viewCartDetails(), updateProfile(), checkOut(), placeOrder(), calcPrice(), register(), login(), and verifyLogin(). The listed functions are represented as either private or public function. The different classes of “users” represented by the object model include the users (“customers”), the maintainers (“Admins”), and a visitor (“regular user”). The associations between the two classes are as follows: Name string, address string, and email address string.

There are so many ways objects use their variables and functions, in this project different model consists of the order, the shopping cart, the customer, the user, the administrator, the shipping information, and the order’s information. The expected users’ “User”, “Administrator”, and “Customer” must be identified by “name”, “ID”, “Physical Address”, “Email address”, then logins must be verified and authenticated. Every type of user will be able to >register >sign in, and update profile and account as needed. The shopping cart will relate to its variables in a way that “customer”, “user”, or “Administrator/employees” with a shopping cart needs to be first identified and the items inside the shopping cart will be identified as well, the item needs to be counted, the date that the item was added needs to be present. All users need to be able to add, update, review items in the shopping cart when needed. There should be a checkout window of the shopping cart where after adding, updating, and reviewing items, the user will checkout via this window. Each user’s order will be tagged with an ID that consists of the creation’s date, shipping information, status like received, has shipped/not yet shipped. The order’s status will be shared with the user, including updates. The transaction must be completed by the user, except in the presence of some exceptions like errors with a credit card and IDs not matching. The shipping information will include an Identification Number, type, costs, carrier, and carrier’s information. Shipping information should be able to be updated and the order’s information needs to be presented and consists of the user’s information, products information, count of the product in the order. The subtotal and unit cost will be displayed. After adding and updating, the order’s total will be calculated and displayed.

The object model may or may not capture all the desired functionality because not everything can be implemented, however, the storefront could use cancel, return, order again functions to better optimize their storefront for their users’. Looking closely, you can see that these classes have “aspects” that contain another or another aspect of objects. Is like an embedded process. To be precise, let’s take the user “customer” as an example, >customer >each customer’s instance contains >shopping cart and order, each order’s instance contains> order’s information and shipping information.

To better distinguish between process model and object model, the process model is like the skeleton of the entire program, it includes operators and structural aspects of the program. The object model includes further information with classes and their variables and functions. When I think of the process model, I think of a protocol that makes any process(anything) be easily understood, it is like an engine of a car with so many parts, each part is just the basics of the contents and how they operate. The object model described the system better as each part of the system is decomposed or broken down into several variables and functions, which allows for a better understanding of the system. The decomposition of the variables and functions helped me understand what is happening in the system and how they happen. The object model presented the organization and corporation of other aspects or classes in the system.