Elliot Putnam

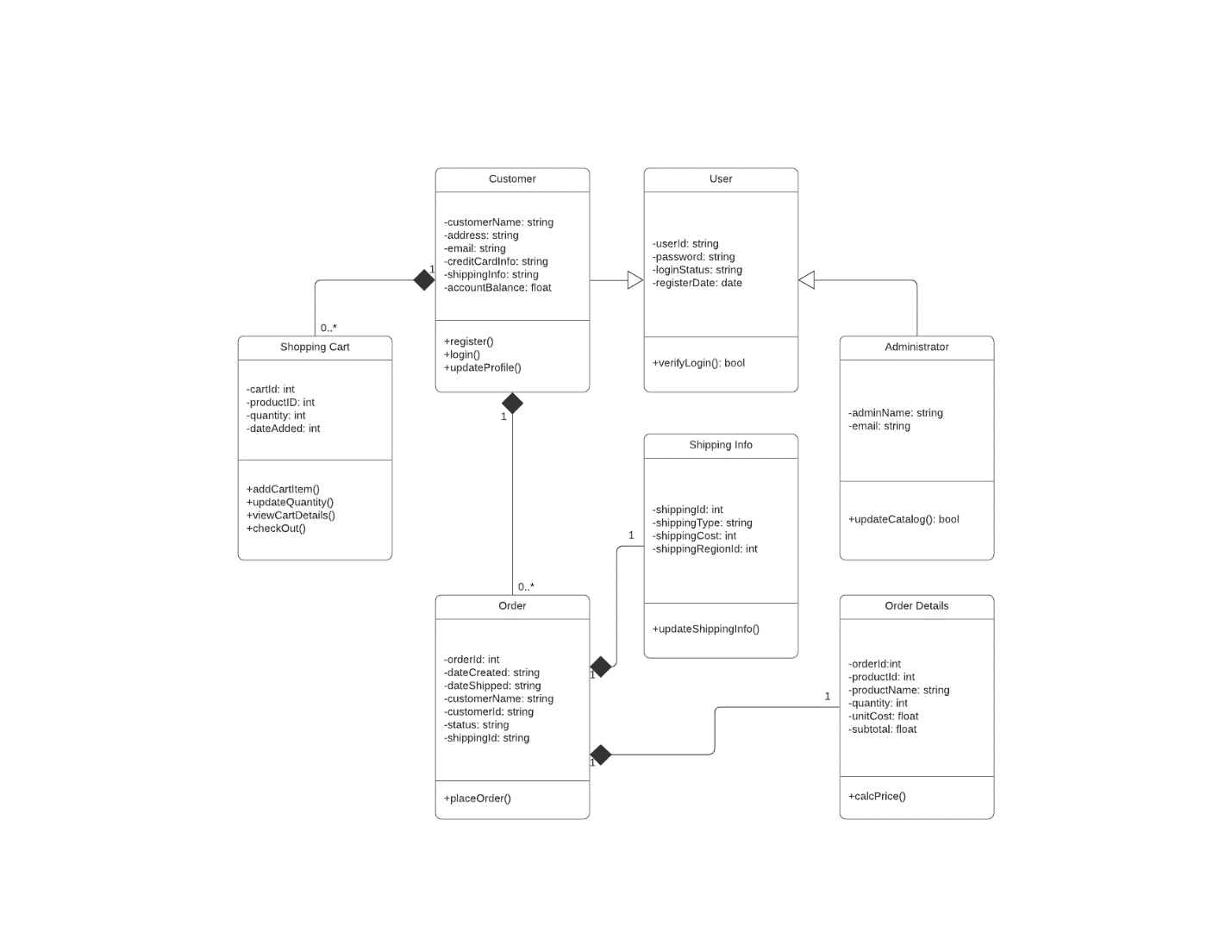
CS-255 System Analysis and Design

Object Model Evaluation

11/17/23

Hamp Crafts is a family-owned craft store that has been in business for decades. Its current operations are limited to its physical (brick-and-mortar) storefront. Recently, the owners have determined that one of the best ways to gain additional revenue is through online sales. You have reassured Hamp Crafts’ owners that adding in an online storefront will work with their current processes. In this assignment, you will review an object model, specifically a UML class diagram for the new online store. You will interpret the diagram and reflect on how well it meets Hamp Crafts’ needs.

1. First, review Hamp Crafts’ needs for their online storefront. Then review the UML class diagram for the new online storefront.   
     
   Hamp Crafts would like customers to be able to create an account with their shipping, billing, and contact information. For customer orders, Hamp Crafts would like to accept credit and debit cards for transactions. Hamp Crafts plans on using an established credit card vendor service (e.g., Square, Shopify) to receive customer payments. Once a transaction is complete, the customer should receive a notification based on the information in their personal profile regarding order status and confirmation. On the administrative side of the online storefront, Hamp Crafts should receive an alert of the transaction. Customers should be able to check the status of their order any time online from their personal account profile under order history. The business owners also need an administrative back end for customer support and updates to customer information and the website.
2. **Interpret the object model** for the new online storefront by responding to the following prompts:
   1. What are the different functions of the online storefront? How are they represented in this type of model?
   2. What are the different classes of “users” represented by this object model? What are the associations between these classes?
   3. How would the objects “use” their respective variables and functions?
   4. Does this object model capture all of Hamp Crafts’ desired functionality? Why or why not?
   5. The above diagram uses a solid diamond shape to represent a form of aggregation. What type of aggregation does this represent? What does it imply about the relationship between the classes? Why is a solid diamond the appropriate choice here?
3. Finally, think through the two different models you’ve explored for Hamp Crafts’ systems: a process model and an object model. Then **compare these models** by responding to the following prompts:
   1. How well do you think a process model describes the system? What information does it make easier to understand? What aspects of the system are more difficult to understand or are not represented?
   2. How well do you think an object model describes the system? What information does it make easier to understand? What aspects of the system are more difficult to understand or are not represented?



All the functions of the object model are:

* addCartItem – this adds an item to the cart
* updateQuantity – changes the count for the quantity attached to the item
* viewCartDetails – this is used to view what the user has in their cart
* checkOut – used to move to the checkout portion of the website
* placeOrder – receives shipping info and order details
* updateShippingInfo – fills in shipping fields
* calcPrice – creates price based on items, count, and pricing
* updateCatalog – updates inventory on website
* verifyLogin – checks entered credentials
* register – takes new user details and creates account
* login – takes returning user details and logs in
* updateProfile – allows user to change user details and saves changes

All these functions are represented as public functions with no needed parameters to pass in. There are two types of users that can be created from the User class, Customer, and Administrator. The administrator can access the updateCatalog function, while the customer can create a shopping cart, place and order, receive shipping information.

The objects use their variables and functions within their class. A single customer can have any amount of shopping carts and orders. The orders themselves, have a 1:1 relationship with the shipping info and order details.

The object model does not capture every function that Hamp Crafts would like to implement. What is missing:

* The customer needs a way to see previous orders and orders that are in progress. So, we must implement a function in the Customer class called orderHistory, along with a private list field called orders.
* The customer and Hamp Crafts need to be notified when an order is processed. Create a function within the Order class called notifyEntities.
* Within the Shopping Cart class, a processPayment function must be added to incorporate the processing of payments via the third party payment portal. This can be called during checkout, with private paymentVerified bool field.
* The Administrator class needs ability to access basic customer credentials such as name, address, shipping address, and email address for modification. This can be called from an updateCustomerInfo function.

The composition identifiers (black diamonds), represent the relationship between the classes. Simply put, when using a composition relationship, the associated objects cannot work without the object they relate to. In this instance, the Customer object must exist for any of its child objects to be created/remain. When the customer gets removed, so do the other objects associated with that object.

The process model described the working steps needed to complete an order from start to finish. Being the *process*, it is telling the analyst what needs to happen, and the rules around how it can happen.

The object model shows what functional elements the program needs in order to be built to maintain a functional relationship with the process models steps.

Each model missed some elements required to be fully complete. However, they did capture the essence of what was trying to be achieved, which was automating a system that was already in place to give better user experience to the customer, along with the administrative users of the company.