Hamp Crafts is a family-owned craft store that has decided to create an online presence to help increase revenue. A Unified Modeling Language (UML) diagram has been created to help create the online storefront that Hamp Crafts desires. Hamp Crafts has several requirements for their online storefront; the ability for customers to create an account with their shipping, billing, and contact information, the ability to accept credit and debit cards for payment, and the capability for customers to receive notifications for their order status and confirmation. Customers should also be able to check the status of their order at any time. Administratively, Hamp Crafts needs to receive an alert or notification of a new transaction, customer support functionality, and updates to customer information on the website.

* What are the different functions of the online storefront? How are they represented in this type of model?

When analyzing the UML diagram, many functions can be identified. Functions are in the bottom third of each class. All the classes have public functions annotated by a ‘+’ sign in front of the function, allowing for access outside of the class by using an instance of the class. Private variables would be annotated by a ‘-’ sign in front of the variable and are not accessible outside of a class or object. An example would be the Shopping Cart class, which has four public functions: addCartItem, updateQuantity, viewCartDetails, and checkOut.

* What are the different classes of “users” represented by this object model? What are the associations between these classes?
* How would the objects “use” their respective variables and functions?

There are two subclasses of the superclass user, customer, and administrator. Customer and administrator act as subclasses in this UML diagram. Superclasses are also known as parent classes and subclasses are also known as child classes. These are fundamental concepts that facilitate code reusability, inheritance, and polymorphism. The Customer and Administrator subclasses inherit userid, password, loginStatus, and registerDate from the User parent class. The classes can perform tasks by accessing their functions. Private variables may not be accessed by other classes and can only be altered by the class to which they are assigned.

* Does this object model capture all of Hamp Crafts’ desired functionality? Why or why not?

Although the UML diagram does include a majority of the functionality that Hamp Crafts has asked for, several aspects are missing. One of the most important functions of this online website is for Hamp Crafts to be able to get payments deposited into their company’s bank account. While the functionality of this may exist inside of an already existing function, it is not easily discernable from this diagram. Similarly, the ability for backend administrative support seems limited based on this diagram alone. As stated previously, the functionality of this may be included in certain functions but isn’t clear in this diagram.

* 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?

The UML diagram uses a solid diamond shape to represent a composition relationship between the classes. Composition is a strong form of association where one class contains or is composed of other classes. Composition links show that a class has an exclusive relationship over the other classes, which helps create the parent-child relationship. The dependency in this relationship is strong, and if the parent class is deleted, the child classes are also deleted. The composition relationship is necessary in this instance because the relationship between these classes should be independent of those class relationships and is not necessarily applicable to the other classes.

* 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?

The process model does a good job of representing the development process, how tasks are organized, and how the programming and system are built over time. It makes the above-mentioned information easier to understand for all parties involved. While this model is more generic, it helps provide a complete overview of what needs to be done. Because the process model is vaguer than the object model, it does not help individuals understand how tasks are going to be accomplished or implemented, only that they will exist and perform specific tasks.

* + 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?

The object model does a good job of representing how the system will behave and how interactions occur by identifying objects, classes, and relationships. The object model makes understanding system interaction more comprehensible and intuitive for all parties, especially developers. The object model does make it difficult to understand certain aspects of the design of the system. As previously stated, certain aspects that were asked for by Hamp Crafts seem to be missing. These may be included functionally into aspects of the functions or variables but aren’t necessarily clear based on the diagram. Also, the ability to know how these classes, objects, and functions work and interact is very generic.