Assignment #4

Practice with Object-oriented analysis & design, and GRASP

Sample answers to questions A-D.

These are based on the Online Inventory system described in Assignment 2. I will treat "Process Sale" as a use case for the Fulfillment Specialist actor. Note: I am not going to consider "Boundary Objects": I am going to ignore the User Interface, and assume that the actor interacts with the Controller Object directly.

A. Write the textual description for the **process sale** use case, including Participating Actors, Flow of Events, Exceptional Flow of Events, and Entry and Exit Conditions.

Use Case Name	Process Sale
Participating Actors	Initiated by Fulfillment Specialist
Flow of Events	The Fulfillment Specialist activates the "Process Sale" function. The Fulfillment Specialist enters the following values: the sku of the sold item, the quantity of that item that were sold, and the cost to ship all of the items. The system finds the Product with the given sku in the Inventory. The system decreases the quantity of the Product by the given quantity that were sold. The system computes the total price, shipping credit, commission, and profit, and outputs these values to the FulfillmentSpecialist.
Exceptional Flow of Events	If there is no product in the inventory with the given sku, the System outputs an error message and aborts the operation. If the quantity of the Product in the inventory is not greater than the quantity sold, the System outputs an error message and aborts the operation.
Entry Condition	The Fulfillment Specialist has started the system.
Exit Condition	The Fulfillment Specialist has received the computed statistics, and the quantity of the Product has been decreased by the quantity sold.

B. Identify the entity, boundary, and control objects from your use case. Be sure to establish a control object for the use case.
   - Entity objects:

Product	The item that was sold
Inventory	The list of Products sold by the company

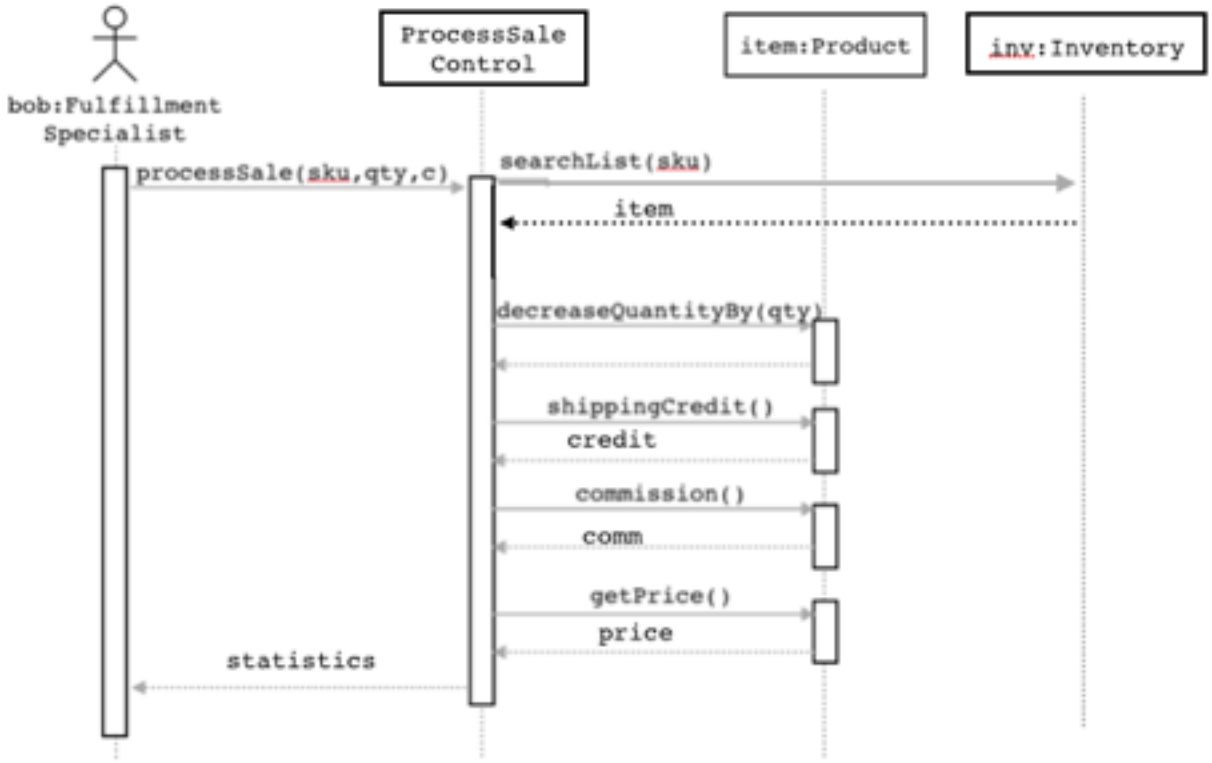
   - Boundary objects: Ignoring these for this assignment.

   - Control objects (Note we did not have this in Assignment 2):

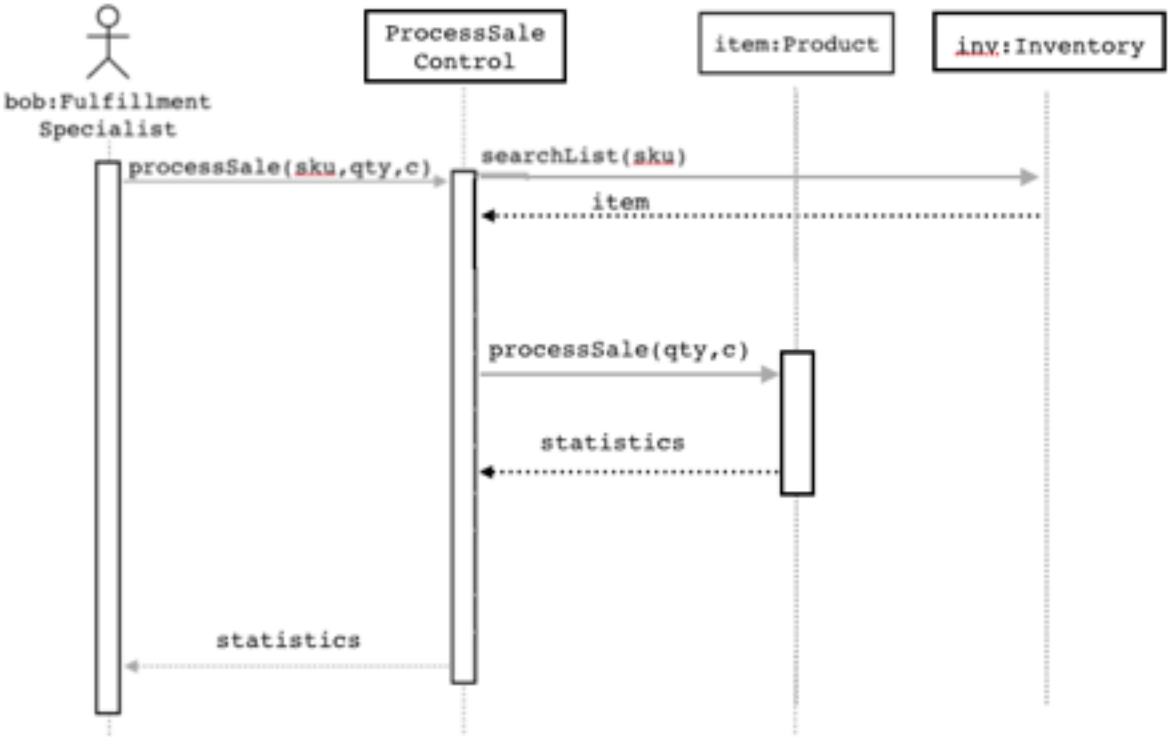
ProcessSaleControl	Manages the processSale reporting function. This object will coordinate the work done by the system.

C. Draw a sequence diagram that describes the process sale use case. Annotate as many messages as possible with a note indicating the GRASP (Expert, Creator, and so on) and/or any other pattern or reason that justifies it.

   I'm going to use two diagrams and then discuss them with respect to GRASP principles. The first one is similar to the answer to Assignment 3 Question 3, but I have a controller object now, so I'm having it do the work instead of the Inventory, and the searchList function is now public and returns the found Product. I'm not reverse engineering now, I'm forwards engineering, not looking at the code.



So the ProcessSaleControl communicates with the user and dispatches work to Inventory and Product. If I evaluate this with respect to the GRASP principles, it looks like ProcessSaleControl is doing a lot of work to compute the statistics (Low Coherence?). And the Product has all the information to compute the statistics, so maybe the Product should calculate the statistics by Information Expert:



Since the Product has all the info to produce the statistics, I don't need to specify the details here of how that is done (though I could add a bunch of self-calls if I wanted to make it more clear).

Another thing to consider is the Coupling; ProcessSaleControl communicates with Inventory AND Product. We could put a processSale method in Inventory (similar to the original assignment solution) that finds the Product item but then calls the new Product.processSale(qty,c) method on the Product. Then we would have less coupling, but the Inventory would have appear to have more responsibility since it knows about processing sales (lower coherence). So this diagram should do fine.

D. Draw a class diagram capturing the classes involved in the process sale use case. Include associations, attributes, and operations. Note that you may need classes and attributes for things not mentioned in the Problem Domain description. [You do not need to include the control object in this diagram].

See next page. I could add more information, especially in Product, about the

commission and shippingCredit, or the shippingCredit and commission methods, but they are not in the sequence diagram.