Justification for component decomposition and Interactions:

I saw in our system that there are 4 major components. Those being the system the user can interact with, the inner subsystems that the bulk of our code is using, the warehouse component that includes our online and physical inventory, and the payment system when someone wants to buy something. The inner components of each are the main components that are needed to do such actions. For the App system, the first component is the authorization component, which ensures that the user logs in correctly as any type of user. The other two components are the online item component and the physical item component, which components need an authorised user to be able to take out or buy an online item or physical item. So the authorization component is being used as the provider interface and the online item and physical item are the consumer interfaces. For the inner system subsystem, the main components here are the customers, online order and physical order. Here authorization is the provider for the customers, and vice versa with the online item and physical item. The customer doesn't provide for anything else because the customer component encapsulates all the things customers can change with their account, being password, email etc. The online order and physical order act as providers for the payment component and the warehouse subsystem. This is because when an online or physical order is sent, we need to check the inventory to see if it's available and of course we need to get the item's price and information to make a payment.