The reason that we picked the observer design pattern was because we have a model-view-controller architectural pattern.  It allows for an object’s state to change and then automatically update all the objects that are presentations of the original object to change accordingly.  We need this for user’s listings.  If an Authorized user selling a textbook wants to change the price or add a picture to their listing the presentations of the listing being viewed at by other authorized users need to be updated accordingly.  It also allows for alternate displays.  When a user is looking at their own listing, they will be able to see notation allowing them to add a picture.  When a different authorized user is viewing the presentation of the listing, they should not see the notation, they should only see the Name, Listing, and Price.  Being able to separate the presentation of the object from the object itself is vital for CSwap to work successfully.

Diagram

Description automatically generated