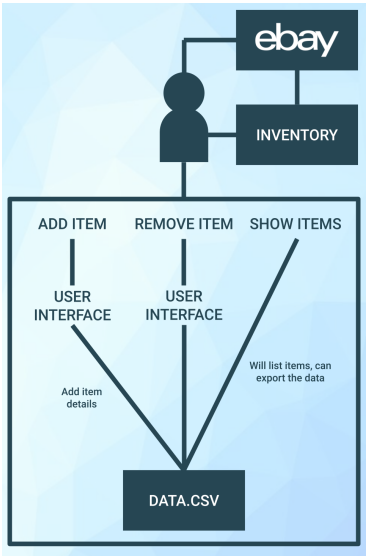
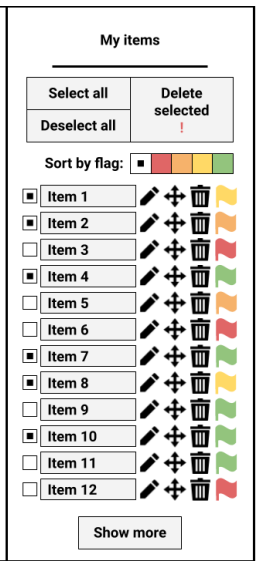
Design overview:

Use case diagram and screen sketches for adding an item & showing items. Along with this there's a code planning document for the Item class. These were the initial versions, and much about the project changed. Even though the final project is so different, it was important having these diagrams so that I could keep the initial vision for the project.

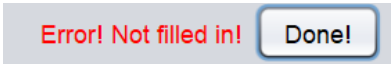
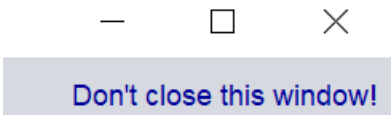
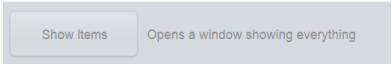




Item class: Item objects hold data about an individual item. It will be able to return its instance variables and edit its instance variables.
Local variables: • index - int • name - String • price - double • description - String • amount - integer • images - ??? • status: Sold/Coming/Here - Enum • flagColor - Enum
Constructor: Creates Item objects. It takes the variables above as parameters. Maye it should have a separate constructor for an item that doesn't have images. New objects should be added to an ArrayList of Items.
Accessors: GetItem(); Returns all of the private instance variables of the Item. GetIndex(); Returns the index of the item. GetName(); Returns the name of the item. GetPrice(); Returns the price of the item. GetAmount(); Returns the amount of the item. GetImages(); Returns the images of the item. GetStatus(); Returns the status of the item. GetFlagColor(); Returns the flag color of the item.
EditItem(); Will have the user interface open up a window allowing the user to edit the item. This function has no parameters because it prompts the user to input the edits in a different place.
DuplicateItem(); Will create a duplicate of the item on the ArrayList of items. The index of the clone is 1 more than the original.
DeleteItem(); Will remove the object from the ArrayList of Items... Should this function be in the class storing the ArrayList?

Testing plan:

To test the functionality of the project, I had to ensure a couple of things. I had already outlined the success criteria, but there were other areas of functionality to address.

Action/edgecase to test	Testing method	Solution
Price is not a number	Adding a new item with a price that is “Banana”	Added special conditional statement to prevent this problem
Item amount is negative, or not an integer	Adding a new item with an amount that is “Banana” or -1	Changed the Amount field to not allow the problem to happen 
Product with name “\n” or similar will cause errors with formatting	Naming a product with “\n” or other Java special characters	No solution needed - this does not cause a problem
New item does not have a name	Adding an item without a name	Added a message for when this happens 
Edit button pressed while no items selected or no items exist	Removing all items and pressing the edit button	Added special conditional statement to prevent this problem
Closing the ShowItems window causes problems	Continuously opening and closing ShowItems to see what errors come up	Added a message reminding the user not to close the window, and made it impossible to close(Close MainGUI instead) 
ShowItems can be opened multiple times, causing extreme issues	Clicking the Show Items button 10 times and then trying to interact with the program normally	The button will be greyed out to signify that the user cannot press it again 
Closing AddItem or EditItem terminates entire program	N/A	Editing the JFrame functionality to not allow this