**Analysis:**

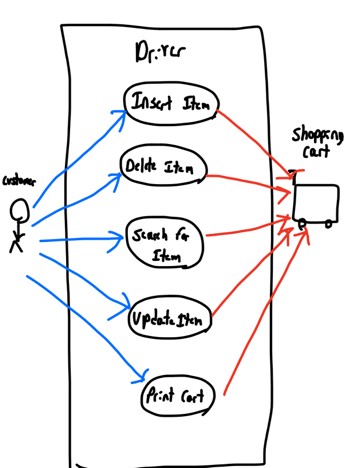
Questions:

1. Do we print both the total price and individual price of an item?
2. Should we print all attributes for an item?
3. When you delete, do you show the quantity of that item that was deleted or the number of object deleted?
4. When you search, do you add up all the quantities of that item and just show the quantity and item name?
5. What exactly is the Functional Block Diagram supposed to look like? When I google it give me more of a hardware Functional Block Diagram, so I’m unsure of what its suppose to look like.
6. When you do the algorithm for the driver class, what if you create an object Transaction that handles most of the work? Are you supposed to write the algorithm for that too or just the driver class?
7. Will we be penalized for not setting/using github correctly?

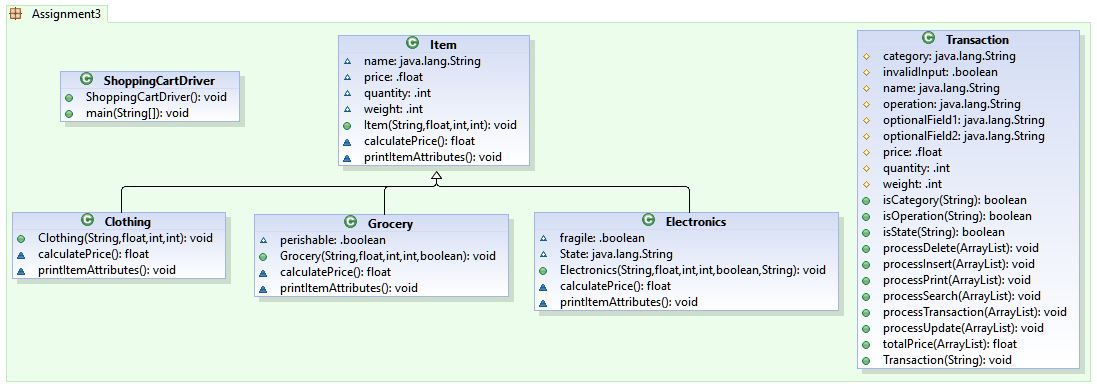
Assumptions:

1. The user can and will input whatever they want. The program with output that the input is invalid and continue to the next line in the input file.
2. The user can input decimals for the price, weight, and quantity but the program will make them into the proper types that are needed for the attributes.
3. If the user uses capitals for the operation and category they will still be consider correct.
4. An item names Pants and pants are two different items.
5. The items should be organized in an ArrayList.
6. Printing the attributes can be formatted however the programmer decides
7. We can use the arraylist sorting method to sort the arraylist as long as uppercase comes before lowercase.
8. Can have a Transaction class to take care of error handling and processing the input.
9. We can show the invalid input to the user so that the user may try to figure out why its invalid.

**Design:**

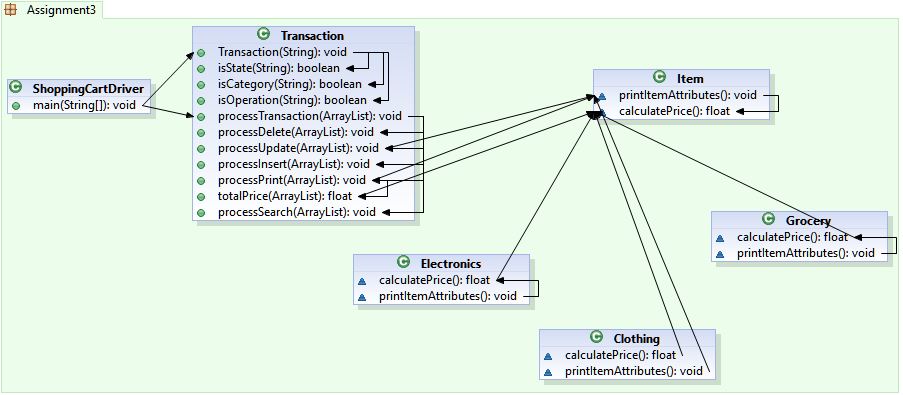


**Use Case Diagram:**

 **UML:**

**ADT Level Description:**

* **Item** - Base class, includes name, price, quantity, and weight of an item. Has methods for calculating price, printing item attributes, updating the quantity, and get methods for each variable.
* **Grocery (extends item)** - contains Boolean to determine perishable or not, overrides price calculation to include premium shipping if perishable, overrides print method to include Boolean perish
* **Clothing (extends item)** - Overrides price calculation to include tax
* **Electronics (extends item)** - contains Boolean to represent fragility and String state to represent state it is purchased from, overrides price calculation to include premium shipping if fragile and tax if from a taxed state, overrides print to include fragility and state.

**Functional Block Diagram**

**Algorithm for driver logic:**

Check if there is a file

If not end program

Open and Read the file

Create an Array List of items for Shopping Cart

For every line in file

Create a transaction object that checks if the input is valid and stores the different attributes of the input

If the input is a valid transaction

Process the transaction

Else

Tell the user the input is not valid and show them the input