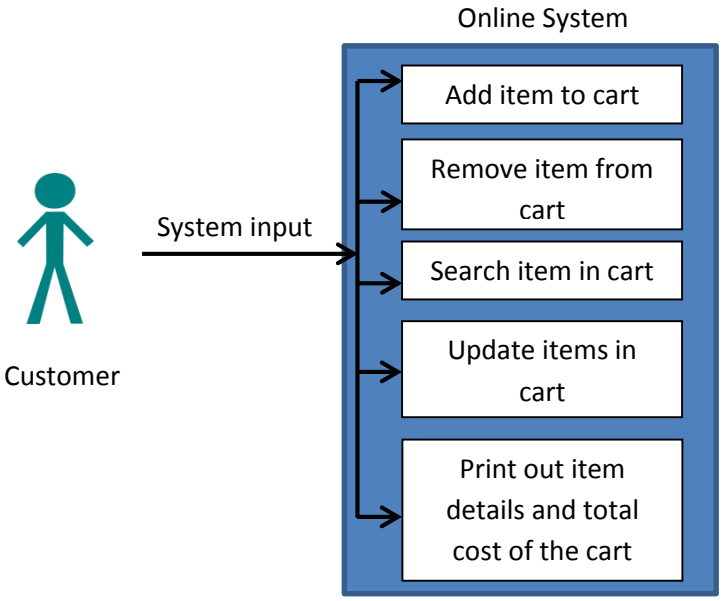
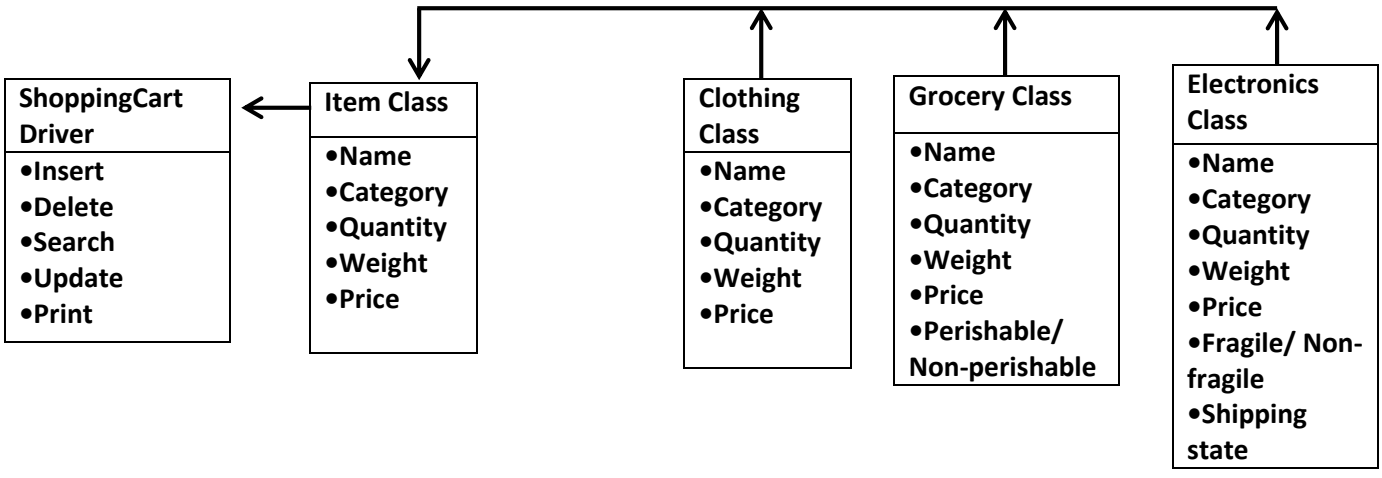
Lab 3 Analysis and Diagrams

System Level Use Case Diagram:



UML Diagram:



Description of Classes:

•**Shopping Cart Driver-** The driver will create an array that all the items are stored on. It then takes the input string and separates the word from the string. The driver will then decide the operation it must perform (insert, delete, search, update, print) and follow through with the stated method.

•**Item Class-** Items contain attributes that can be modified. Each item will have a name, category, quantity, price, and weight. The item class also contains methods that can retrieve each variable from any item, as well as a method to calculate price, tax, shipping and total costs.

•**Clothing Class-** An extension of the Item class. No extra attributes are added.

•**Grocery Class-** An extension of the Item class. A Boolean perishable/nonperishable is added. Contains a special method for calculating groceries based on the perishable/nonperishable Boolean. Also contains a method that prints out a grocery item and its' attributes.

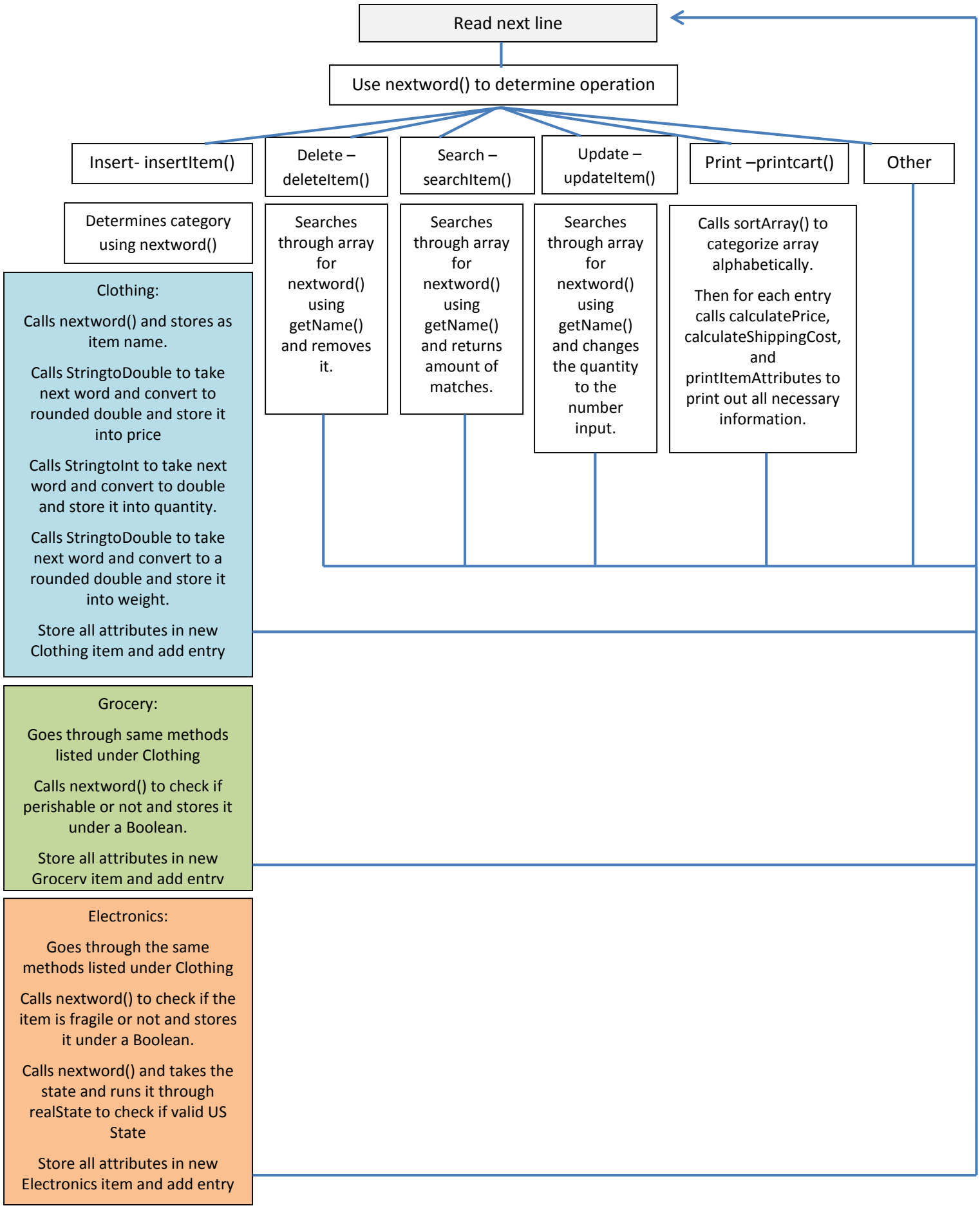
•**Electronics Class-** An extension of the Item class. A Boolean fragile/nonfragile is added. The class also includes a method that checks if the state selected for shipping is one of the US states. Contains a special method for calculating electronics based on fragile/nonfragile Boolean and the shipping state. Also contains a method that prints out a electronics item and its' attributes.

•**Conversion Class-** An extra class used to parse the String input to double or int as required, and also has a method for rounding numbers to a certain decimal point for calculations.

Main Method Algorithm- Used the same reader and main as Assignment 1, where the main would keep the reader running as long as there were no errors in reading the document. The reader will read each line and send it to processCart to process the transaction. After the transaction is completed, a string will be returned, either a message confirming the transaction was processed, or an error message. The reader will then print out the string and proceed on with the next line.

Analysis: The project problem was to create a shopping system where a user can input a line that would include a transaction type and an item and its attributes that are required for that transaction type. Transaction types include insert, delete, search, update, print. A question with the update function would be wondering does updating a quantity of an item to 0 remove it from the shopping cart array list? We addressed that question in the program by keeping the item in the array but just changing the quantity attribute to 0. Another common question about sorting the cart alphabetically is what would happen if an item had the same name? Or an exact match even? The sorter organizes the cart in a way that uppercase values would come first, and even if the item had the exact same name it would still create a new entry however the first inserted entry would be used for other transactions such as update, but all entries of that exact name will be deleted when the delete transaction is called.

Assumptions are that the quantity input will be a whole integer, as well as weight. However weight can be entered in as a decimal number; however the rounding method will be used to convert it to a whole number so the transaction will still go through. The same rounding idea will apply to price if the price is a number that goes further than 2 decimal places. Shipping should stay within the U.S.


Read next line
Use nextword() to determine operation
Insert- insertItem()
Delete – deleteItem()
Search – searchItem()
Update – updateItem()
Print –printcart()
Other
Determines category using nextword()
Searches through array for nextword() using getName() and removes it.
Searches through array for nextword() using getName() and returns amount of matches.
Searches through array for nextword() using getName() and changes the quantity to the number input.
Calls sortArray() to categorize array alphabetically.
Then for each entry calls calculatePrice, calculateShippingCost, and printItemAttributes to print out all necessary information.
Clothing:
Calls nextword() and stores as item name.
Calls StringtoDouble to take next word and convert to rounded double and store it into price
Calls StringtoInt to take next word and convert to double and store it into quantity.
Calls StringtoDouble to take next word and convert to a rounded double and store it into weight.
Store all attributes in new Clothing item and add entry
Grocery:
Goes through same methods listed under Clothing
Calls nextword() to check if perishable or not and stores it under a Boolean.
Store all attributes in new Grocery item and add entry
Electronics:
Goes through the same methods listed under Clothing
Calls nextword() to check if the item is fragile or not and stores it under a Boolean.
Calls nextword() and takes the state and runs it through realState to check if valid US State
Store all attributes in new Electronics item and add entry