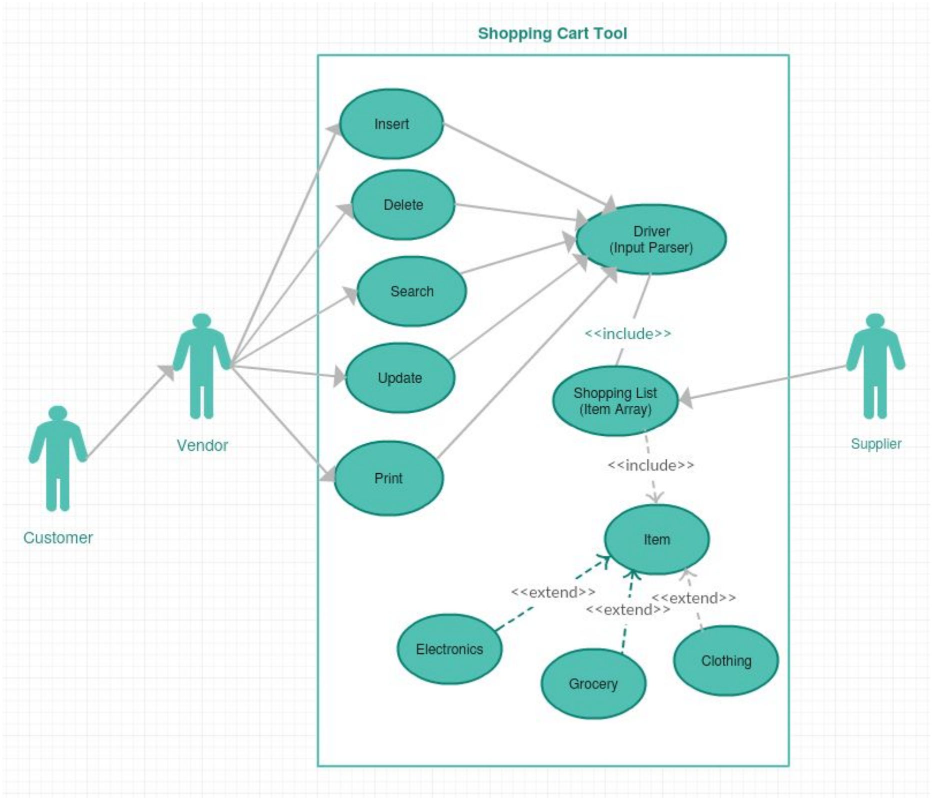
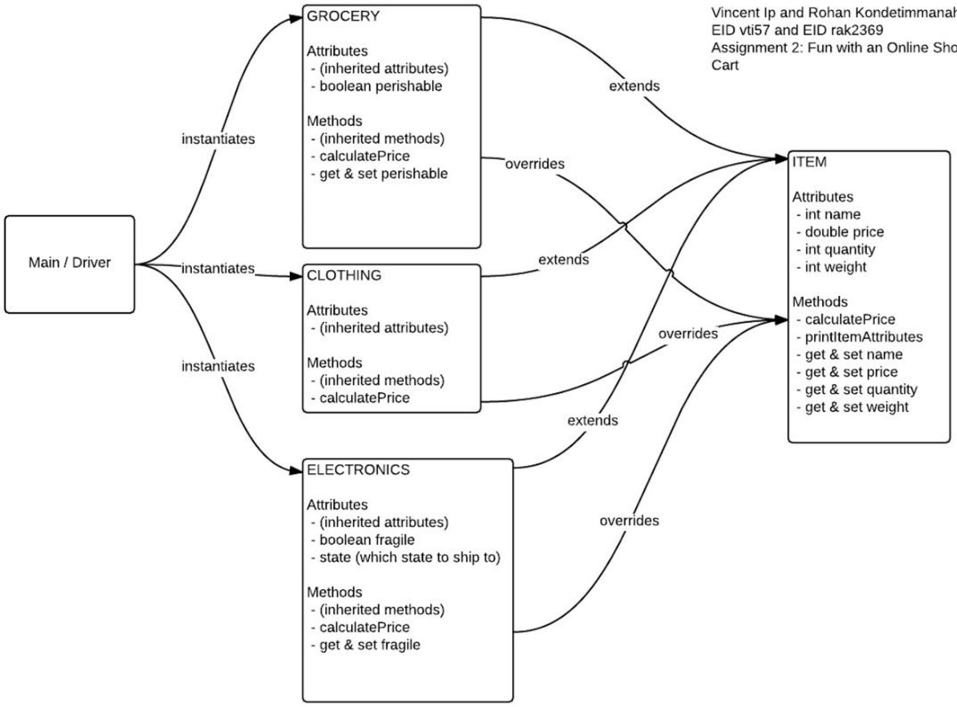
Design.pdf

Rohan Kondetimmanahalli (rak2369) Vincent Ip (vti57)

Use Case Diagram



UML Class Relationship Diagram



Vincent Ip and Rohan Kondetimmanahalli
EID vti57 and EID rak2369
Assignment 2: Fun with an Online Shopping Cart

Design.pdf

Rohan Kondetimmanahalli (rak2369) Vincent Ip (vti57)

ADT-Level Description of Classes

A3Driver.java

Main Method

Implements Driver algorithm parses input data and split each argument into its appropriate data type while ensuring input is valid. The algorithm checks for the correct number of arguments (if there are more or less than the expected number, the command will be skipped and an error message will be printed onto the screen). In addition, it makes sure that the correct data types are being used (if there is a floating point value where an integer is expected, the program will output a message and the specific command will not be processed). See algorithm of driver logic for more details.

Item.java

This is the superclass for Clothing.java, Electronics.java and Grocery.java. It handles all the generic attributes for all items including name, price, quantity, and weight. All the respective attributes in both the superclass and subclass come with helper functions to get and set each attribute.

Clothing.java

In addition to the attributes inherited from Item.java, this subclass modifies the calculatePrice() function to account for the lack of premium shipping.

Electronics.java

This subclass includes attributes for fragility and state which are used in its modified calculatePrice() method to account for varying sales tax in different states and required premium for fragile objects.

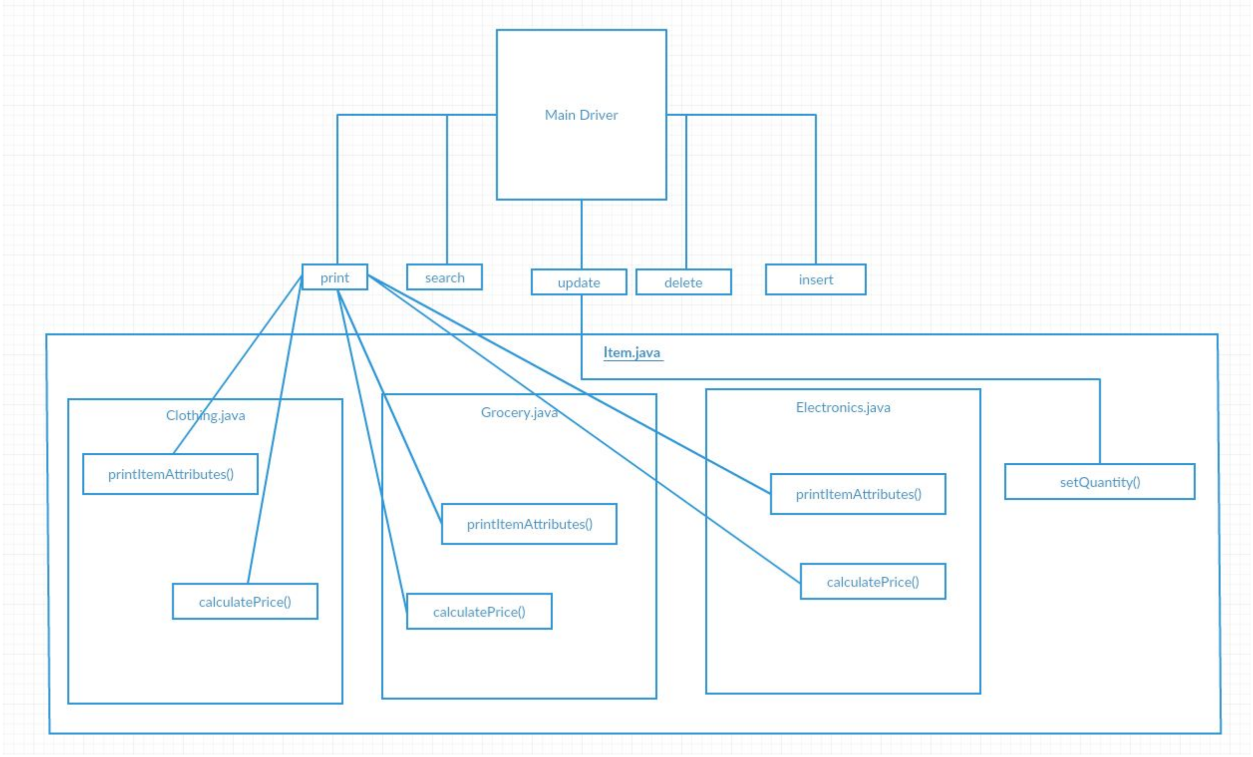
Grocery.java

This subclass includes an attribute to indicate if the item is perishable. This attribute is used in a modified calculatePrice() function to account for forced premium.

Design.pdf

Rohan Kondetimmanahalli (rak2369) Vincent Ip (vti57)

Functional Block Diagram



Driver Logic Algorithm

The driver class (which consists of the main method and implementation of the Item class and its subclasses). Focuses on parsing data from the inputFile while ensuring the inputs are valid.

1. Checks to see if file exists and if the command line input is correct and singular (try, catches invalid file name input).
2. Opens the indicated file and creates a scanner to prepare for reading file.
3. create an empty Item array.
4. Begins to read file.
    - a. Gets the next line in the file as a String object unless there is no next line then end program.
    - b. splits the string into words separated by spaces into a String array.
    - c. enters a switch statement to determine what the first word is. If it is not insert, search, delete, update, or print, then output an error and go back to step a.
    - d. If the first word is insert
        - i. places the first 6 arguments into either a String, int, or double, depending on which one it is. If there is a type mismatch, output an error and go step a.
        - ii. if the second word is clothing
            1. check to see if there are 6 words total, otherwise print error and go to a.
            2. create a new clothing object and add it to the array list and go to step a.
        - iii. if the second word is groceries
            1. check to see if there are 7 words total, otherwise print error and go to a.

```
                2. create a new grocery object and add it to the array list and go to step a.
        iv. if the second word is electronics
                1. check to see if there are 8 words total, otherwise print error and go to a.
                2. create a new electronics object and add it to the array list and go to step a.
        v. go to step a if not already done so (if the word was none of the above).
e. if the first word is search
        i. makes sure there are two arguments of the correct data type, else go to step a and
           print an error
        ii. count how many times the name is the same as the search term in the array
        iii. print the number of times the term appeared
        iv. go to step a.
f. if the first word is delete
        i. makes sure there are two arguments of the correct data type, else go to step a and
           print an error
        ii. traverse the array until the name equals the term to be deleted
        iii. remove the term
        iv. go to step a.
g. if the first word is update
        i. makes sure there are 3 arguments of the correct data type, else go to step a and
           print an error
        ii. traverse the array until the name is the term to be updated
        iii. change the quantity to the desired amount
        iv. go to step a
h. if the first word is print
        i. makes sure there is 1 argument of the correct data type, else go to step a and print
           an error
        ii. print all the items in the item array and their attributes.
        iii. go to step a.
i. if the first word is none of the above, print an error and go to step a.
```