Analysis

We are creating a “cart” for an online vendor. We need to have an “adding” functionality, to add items to the cart, a “pricing” functionality to report the total price (tax and shipping included). We know the name (String), price (real number), quantity (non-negative whole number), and weight (whole pounds) of each purchase.

Some rules:

- Sales tax is 10% for clothes and electronics
  - Some items have no sales tax:
    - 0% for groceries
    - 0% for Electronics in TX, NM, VA, AZ, and AK
- Shipping is (20*weight)*quantity
  - Premium shipping is an additional 120% of the original shipping
    - required for perishable groceries
    - required for fragile electronics
    - not available for clothing
- Each item is shipped individually

Input is given as:

<operation> <category> <name> <price> <quantity> <weight> <optional 1> <optional 2>

Operations are:

insert
search
delete
update
print

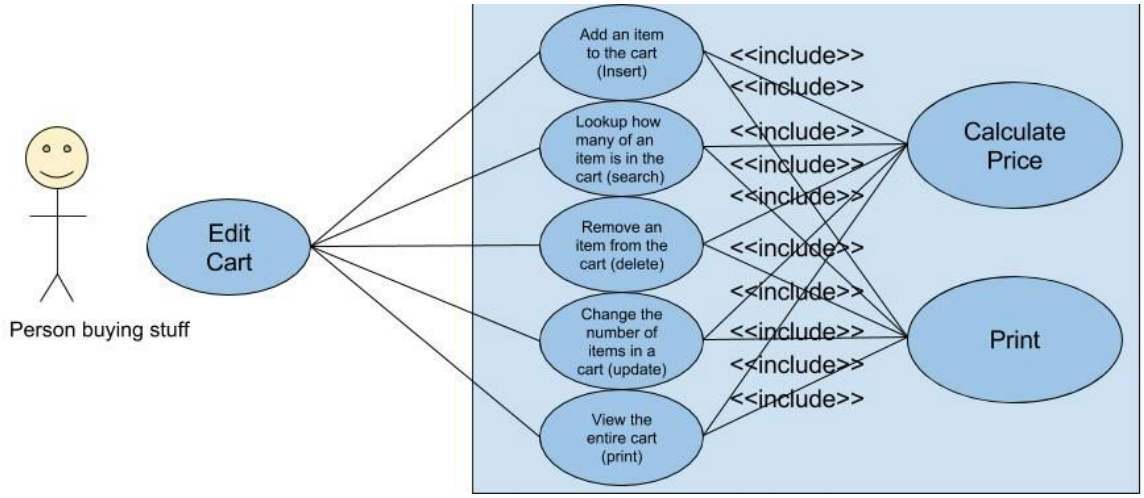
Categories are:

clothing
electronics
groceries

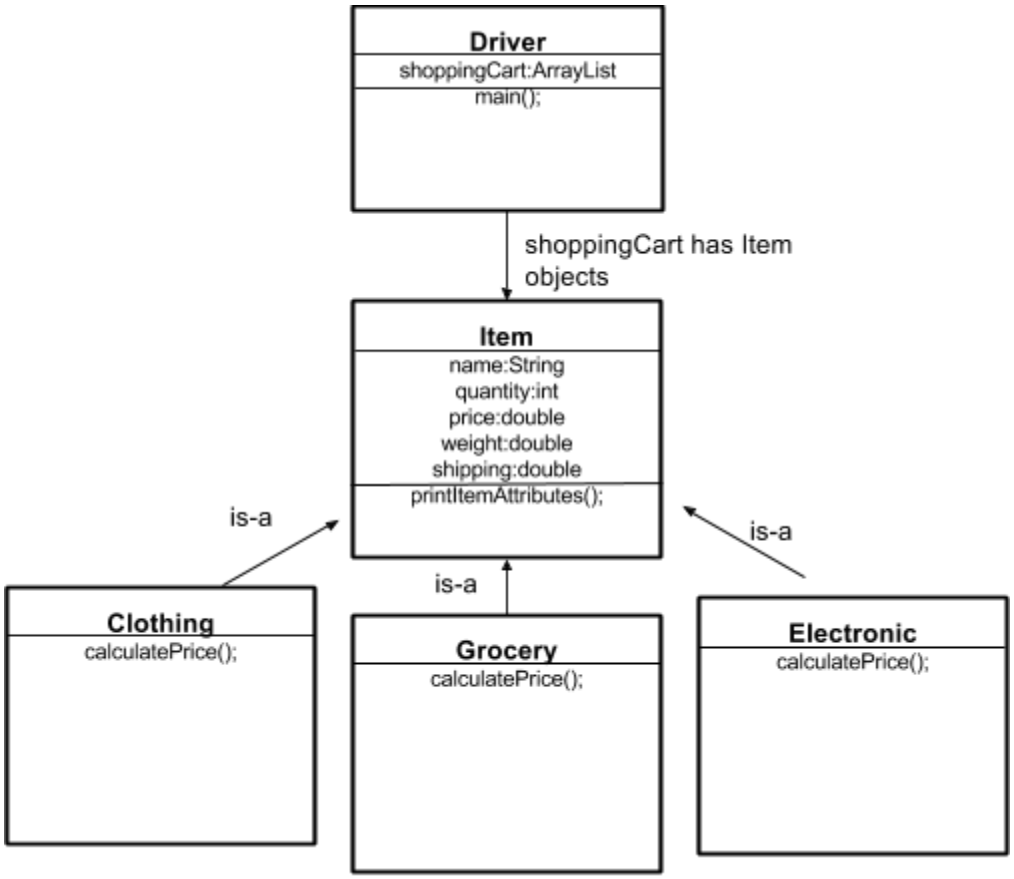
Names, prices, quantities, and weights are variables.

Designs

A System level use case diagram



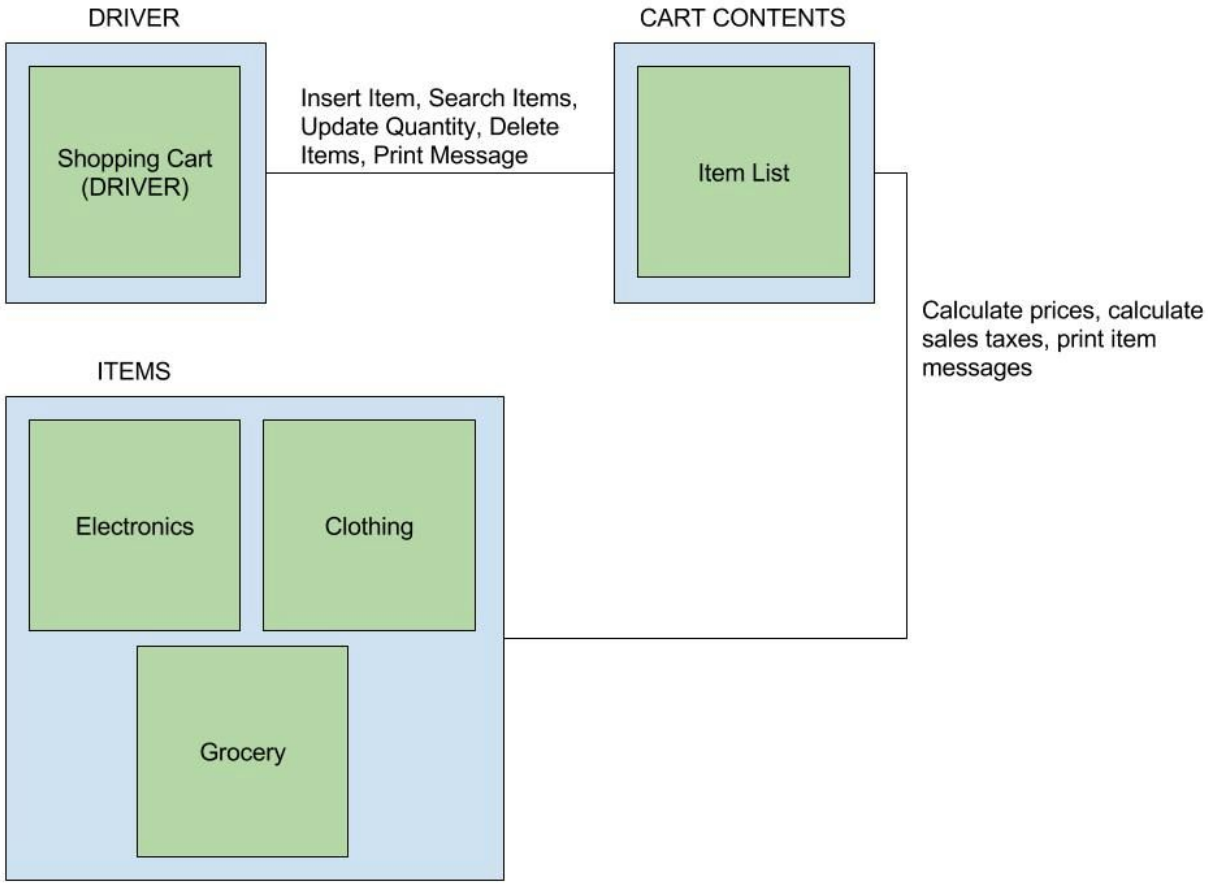
A UML model of the needed classes and their relationships



An ADT level description for each class that you create

We use the Arraylist to store our cart. We choose an arraylist so that we can dynamically insert and remove our Item objects and so that we can easily sort it. The constraints are that the inserted objects must be the same type of object. The arraylist is 1 smaller after removing something than it was previously, and is 1 larger after adding. Before and after sorting they should be the same size, but can be in a different order.

A functional block diagram showing the calling relationships between methods



- **The algorithm needed for the driver logic (main method)**
  - create shopping cart
  - read Input String
  - Input is checked for validity (if invalid do not crash, make a report)
  - Input String is broken into an array (Split on spaces)
  - Argument is processed based on what the first word (the operation) is
    - If the operation is insert, add the specified item to the cart arraylist
    - If the operation is delete, remove the all occurrences of the specified item from the cart arraylist and output the quantity removed

- If the operation is update, edit the amount of the first occurrence of the specified item
- If the operation is search, output the total quantity of items
- If the operation is print, print the contents of the cart