RecipeNShopping.java

RecipeNShopping.java is an application that makes a shopping list based on the recipes you plan to cook in a week plus the staples purchased each week for your family. To do this, the user would enter a list of recipes to be cooked in this format. The user should type these with the keyboard (so use Scanner with System.in) until entering QUIT (sentinel):

Chicken Soup
Tacos
Dijon Yogurt Roast Salmon
Malaysian Spring Rolls
Black and White Cookies
Asparagus
Bacon and Herbs Mac N Cheese
Chickpea Salad
Frozen Pizza

The user could type in a little or a lot, and the software should be able to produce an appropriate shopping list each time.

Upon receipt of this list, the software would review the recipes for each item, putting the ingredients for each recipe together in a file called ShoppingList.txt. All recipes are together in a file called Recipes.txt. This can be systematically done, because the format of all recipes in Recipes.txt is as follows:

Recipe: Title – serving size
Rating – notes

Ingredients:
Ingredient List with volumes next to each item

Directions:
Narrative of how to cook the food

For example:

Recipe: Chicken Soup – 8 servings
5 stars – highly customizable and extra yummy and nutritious if bone broth is used

Ingredients:
celery (3 stalks)
carrot (2)
onion (1)
…

Directions:

Finely chop or dice the celery, carrots, and onion. To facilitate the process, you can roughly chop each, then use a food processor. …

In Recipe.txt, the term "Recipe:" does not exist except in the title of each recipe. Recipes.txt has more recipes in it than just the ones the user will enter. Recipe.txt may also be missing things that the user enters (see below on how to deal with this).

Ingredients and volumes will be retrieved from the Recipe.txt file and used to populate ShoppingList.txt. This means if Chicken Soup is in the recipe list entered by the user, the ShoppingList.txt would be generated using Recipe.txt content and include the lines:

celery (3 stalks)
carrot (2)
onion (1)

If there is no recipe for an item in the recipe list (e.g. Asparagus), then that item should be listed in the Shopping List just as it was entered in the recipe list. In this case, there may not be a volume next to it, so if no volume is added, add (1) next to it in the ShoppingList.txt. In other words, all lines in ShoppingList.txt have a number in parenthesis and for the example above, your list would have the line:

Asparagus (1)

Each week staples should be purchased. They are in a file called Staples.txt. You should populate your Staples.txt file yourself (you can just type it yourself and save it as Staples.txt), based on what needs to be purchased every week for your family. For instance, in my family, Staples.txt would have:

tomatoes (4)
cow's milk (1/2 gallon)
coconut milk (1/2 gallon)
eggs (2 dozen)
bacon (1 pkg)

Be sure to put all of the staples in your Shopping List.

Don't worry about repeating list items, unless you want to do the extra credit (see below).

Extra Credit: In order to make the Shopping List as concise as possible, common ingredients should be combined as a single list item. Volumes can be combined using the '+' character. For example, if 1 onion is needed for Chicken Soup (above) and 1 onion is needed for tacos, then the Shopping List might have one of its lines:

onion (1 + 1)

Don't worry about singular or plural terms (as for onion, above) – you can assume an ingredient is always spelled the same, although you should consider ignoring case (.equalsIgnoreCase()). You will have to explain how you did the extra credit, here, to earn the extra credit. Be prepared to describe iteration, control structures, and objects instantiated, manipulated, and closed. You are welcome to use 2-D Arrays along with files. Please do not use structures not taught in class.