USER

+ changeEmail (String newEmail)
    - Can remove
+ changePhone (String newPhone)
    - Can remove
+ newGroceryList (String listName)
    - Requirement#10: The application must support multiple lists at a time (e.g., “weekly grocery list”, “monthly farmer’s market list”). Therefore, the application must provide the users with the ability to create, (re)name, select, and delete lists.
+ removeGroceryList (groceryList)
    - Requirement#10: The application must support multiple lists at a time (e.g., “weekly grocery list”, “monthly farmer’s market list”). Therefore, the application must provide the users with the ability to create, (re)name, select, and delete lists.
+ renameList (groceryList, String newName)
    - Requirement#10: The application must support multiple lists at a time (e.g., “weekly grocery list”, “monthly farmer’s market list”). Therefore, the application must provide the users with the ability to create, (re)name, select, and delete lists.
+ search (availableItemList)
    - Requirement#9: The application must present the items in a list grouped by type, so as to allow users to shop for a specific type of products at once (i.e., without having to go back and forth between aisles)
    - This is basically just allowing the user to access the database with a “browse” sort of purpose, to view the list of all items separated by list
+ searchType (availableItemList, String type)
    - Can remove
    - Not a requirement, but I threw this in there just cuz it seemed like a logical decision – being able to filter for a specific itemType and look through those items.
+ searchItem (availableItemList, String itemName)
    - Requirement#4: Users must also be able to specify an item by typing its name. In this case, the application must look in its DB for items with similar names and ask the users, for each of them, whether that is the item they intended to add
    - Having this function would essentially give users the option to search by just typing in an item name. And then they would have an additional, separate option to search by itemType as well
+ addToList (availableItemsList, item_id, groceryList)
    - Requirement#1: The application must allow users to add items to a list, delete items from a list, and change the quantity of items in the list
+ removeFromList(groceryList, item_id)
    - Requirement#1: The application must allow users to add items to a list, delete items from a list, and change the quantity of items in the list
+ changeQuantity(groceryList, item_id, newQuantity)
    - Requirement#1: The application must allow users to add items to a list, delete items from a list, and change the quantity of items in the list
+ selectItems(availableItemsList)
    - Can remove, unnecessary additional functionality to select multiple items from the full availableItemsList
+ addSelectedItems(availableItemsList)
    - Can remove, unnecessary additional functionality to add the selected items from the full availableItemsList

+ checkOff_Item(groceryList, item_id)
    - Requirement#6: Users must be able to check off items in a list (without deleting them).
+ clearChecked(groceryList)
    - Requirement#7: Users must also be able to clear all the check-off marks in a list at once.
+ selectItems(groceryList)
    - Can remove, unnecessary additional functionality to select multiple items from the user's groceryList
+ clearSelected(groceryList)
    - Can remove, unnecessary additional functionality to remove the selected items from the user's groceryList
+ saveList(groceryList)
    - Can remove; unnecessary additional functionality to allow the user to manually save a list, but this is irrelevant since autosave would be enabled since it is a requirement

availableItemList

- newItemType(String typeName)
    - Requirement4: Users must also be able to specify an item by typing its name. In this case, the application must look in its DB for items with similar names and ask the users, for each of them, whether that is the item they intended to add. If a match cannot be found, the application must ask the user to select a type for the item and then save the new item, together with its type, in its DB.
    - Doesn't necessarily say to create a new itemType, but it makes sense if you are adding a new item and the itemType doesn't exist yet
    - Can Remove
- removeItemType(itemType_id)
    - Requirement4: Users must also be able to specify an item by typing its name. In this case, the application must look in its DB for items with similar names and ask the users, for each of them, whether that is the item they intended to add. If a match cannot be found, the application must ask the user to select a type for the item and then save the new item, together with its type, in its DB.
    - SAME REASING AS FUNCTION ABOVE: Doesn't necessarily require us to create/remove a new itemType, but it makes sense if you are adding a new item and the itemType doesn't exist yet
    - Can Remove
- filterForType (String userInput)
    - Can remove; but would work together with "searchType( )" function under USER.
    - Unnecessary additional functionality
- filterForItem (String userInput)
    - Requirement#4: Users must also be able to specify an item by typing its name. In this case, the application must look in its DB for items with similar names and ask the users, for each of them, whether that is the item they intended to add. If a match cannot be found, the application must ask the user to select a type for the item and then save the new item, together with its type, in its DB.
    - User would call USER.searchItem(availableItemList, String input)
        - The function is made up of the this class's function: availableItemList.filterForItem(input)
        - Then it would show those with matches to the user's input, or allow them to create a new item.
    - Should make this a + instead of a – in order for it to be called by the USER I guess?
- requestNewItem (user, itemName, itemType)
    - Can remove; didn't realize the user will just have all the power lol I thought they would have to request for an item to become an item. But I guess the item they want will just automatically be added to the availableItems

itemType

- newItem(String itemName)
  - Requirement#4: Needed to add a new item to this itemType
- removeItem(item_id)
  - Can remove; unnecessary functionality to remove an item from an itemType
- filterForItem (String userInput)
  - Can remove; additional functionality
  - After searching for a specific itemType, you can search for an item within it….but I guess this is redundant since you could just search for the item directly from the beginning

ITEM

- increaseQuantity( )
  - Can remove; unnecessary functionality to order more if store inventory is low on this item
- decreaseQuantity( )
  - Can remove; unnecessary functionality to order more if store inventory is high on this item, or to account for perishable items that weren't sold. Either way, inventory management wasn't a requirement for us.
- isSelected( )
  - Functionality to add more than one item to your ilst
  - Can remove
- select( )
  - Functionality to add more than one item to your ilst
  - Can remove
- unselect( )
  - Functionality to add more than one item to your ilst
  - Can remove

groceryList

- addNewItem(item_id)
  - Main requirement: will just add the itemID to the User's grocerylList. That's really all you need in the grocerylist – with the itemID, you can access the itemName and other details
  - When user calls USER. addToList (availableItemsList, item_id, groceryList), it runs groceryList.addNewItem(item_id)
  - Should probably make this a + instead of a -
- removeItem (item_id)
  - Main requirement: will remove the itemID from the User's grocerylList. That's really all you need in the grocerylist – with the itemID, you can access the itemName and other details
  - When user calls USER. removeFromList (groceryList, item_id), it runs groceryList.removeItem(item_id)
  - Should probably make this a + instead of a -

- changeQuantity (item_id, newQuantity)
  - Main requirement: will change the quantity of the itemID from the User's grocerylList.
  - When user calls USER. changeQuantity (groceryList, item_id, int newQuantity), it runs groceryList.changeQuantity(item_id, newQuantity)
  - Should probably make this a + instead of a -
- checkOff (item_id)
  - Requirement#6: Users must be able to check off items in a list (without deleting them).

- User calls this function inside its own USER.checkOff (groceryList, itemID) function
- Should probably make this a + instad of a -

- selectItem(item_id)
  - Can remove; unnecessary functionality to select multiple items and remove them from a list
- clearChecked( )
  - Requirement#7: Users must also be able to clear all the check-off marks in a list at once.
  - User calls this function from within its own User.clearChecked(groceryList) function

itemAdded

- increaseQuantity( )
  - Main Requirement: must be able to edit the quantity of an item that is on the user's list
  - Called within groceryList function: groceryList.changeQuantity(item_id, newQuantity)
- decreaseQuantity( )
  - Main Requirement: must be able to edit the quantity of an item that is on the user's list
  - Called within groceryList function: groceryList.changeQuantity(item_id, newQuantity)
  - Now that I look at this, we could just make an editQuantity( ) function instead of increase and decrease quantity
- isCheckedOff( )
  - Necessary boolean validation to run clearCheckedOff( )
- checkOff( )
  - Necessary boolean validation to run clearCheckedOff( )
- unCheck( )
  - Necessary boolean validation to run clearCheckedOff( )
- isSelected( )
  - Can remove, unnecessary functionality to select multiple items that are on a groceryList and remove/edit their quantities
- select( )
  - Can remove, unnecessary functionality to select multiple items that are on a groceryList and remove/edit their quantities
- unselect( )
  - Can remove, unnecessary functionality to select multiple items that are on a groceryList and remove/edit their quantities