## Step 1: Role Description for each class in the project folder

**Node**

* Object class which is the template for the Node objects used in the link list
* When Node is instantiated, it constructs the object
* Holds the quantity of the item
* Holds the item name
* Provides accessor methods, mutator methods, and a to string method

**SystemProcess**

* System process is the processor for the program which interacts with the ShoppingList class
* Reads the text file to ensure that the file is valid for both the meal and inventory files
* Scans the file and creates Node objects for valid information gathered from the text file
* Eliminates nodes from the list if there is no need for the Node object
* Updates the state of the Nodes if required
* Writes the information to file
* Invokes methods from the ShoppingList class

**ShoppingList**

* Implements the GroceryList class
* Provides an addItem method to increase quantities in existing nodes, or create new nodes to the list
* Provides an algorithm to immediately sort new Nodes into proper lexicographic order
* Provides a removeItem method to reduce quantities in existing nodes, or remove them from the list
* Provides a contains method which returns true if the object exists in the list
* Provides a getQuantity method which returns an integer quantity from a specific node
* Provides a size method which counts the number of Nodes in the list and returns an integer value
* Provides a to string method which builds the information used to write to the file

**Client**

* Main method which prints the messages for file name requests to the screen
* Invokes the methods used to read the files in the SystemProcess class
* Ensures the program stops if any files are unread or maximum attempts are reached

## Step 2: Fill out one CRC card (for each class you found)

|  |  |
| --- | --- |
| Name: ShoppingList | |
| **Knows:**  Starting point of list if exists  Current position of node found  Quantity of  **Responsibilities:**  Creates objects for list through parameters  Adds items to list  Updates items in list  Removes items in list  Verifies if list contains element  Returns quantity | **Collaborations:**  ShoppingList calls contains from within  SystemProcess sends arguments |
| Name: SystemProcess | |
| **Knows:**  State of bad file boolean  When file is found  **Responsibilities:**  Read file names  Read files  Validate file information  Pass arguments to ShoppingList  Write to file | **Collaborations:**  Client requests Boolean  ShoppingList  Keyboard  Scanner  PrintWriter |
| Name: Client | |
| **Knows:**  na  **Responsibilities:**  Main method calls | **Collaborations:**  SystemProcess |
| Name: Node | |
| **Knows:**  Quantity  Item (name)  Next node position  **Responsibilities:**  Constructs the Node object | **Collaborations:**  ShoppingList |

## Step 3: Visually draw the collaborations/relationships

**ShoppingList**

Creates objects for list through parameters ShoppingList

Adds items to list

Updates items in list SystemProcess

Removes items in list

Verifies if list contains element

Returns quantity

**SystemProcess**

Read File Names ShoppingList

Read files Client

Validate file information Keyboard

Pass arguments read from file Scanner

Write to file PrintWriter

**Node**

Constructs Node object ShoppingList

**Client**

Main method calls SystemProcess