Class Summary:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| Interface | BPTreeADT.java | A generic B+ Tree interface. |
| Interface | FoodDataADT.java | A generic Food Data interface. |
| Class | BPTree.java | This class is used to implement B+ tree to access different indexes of a large data set. |
| Class | FoodData.java | The class is used to load and process food item data. |
| Class | FoodItem.java | The class is used to represent the food item with all its properties. |
| Class | FilterButton.java | Check all the rules(range of the nutrients’ values, name) created by user, show foods that meet the criteria in the food list when CheckBoxButton is checked |
| Class | ClearButton.java | Clear all rules entered by the user |
| Class | CheckBoxButton.java | Click to check on/off |
| Class | RemoveButton.java | When clicked, this class calls the removeFromMealList() method that remove the selected foodItem entry from the meal list. |
| Class | AnalyzeButton.java | When clicked, this class calls the analyze() method that generates a new meal list result and pops up a new window showing the results. |
| Class | SaveButton.java | Save the existing foods to a data file containing current food list information. |
| Class | LoadButton.java | Load new foods from a data file containing new food list information and replace the existing list. |
| Class | AddNewFoodButton.java | Add new foods in the food list. After clicking the button, a new window pops up. The user put food’s name and the nutritions so that a new food is on the list |
| Class | AddToMeal.java | Select the food and add these food that are selected to the meal list |

### Class diagram

BPTree:

|  |  |  |  |
| --- | --- | --- | --- |
| Return type | Method name | Parameter List | Description |
| void | insert | K key; V value | Insert the key- value pair into the tree |
| List<V> | rangeSearch | K key; String comparator | Get the values that satisfy the given range search arguments |
| String | toString | none | Represent the tree with a String |

FoodData:

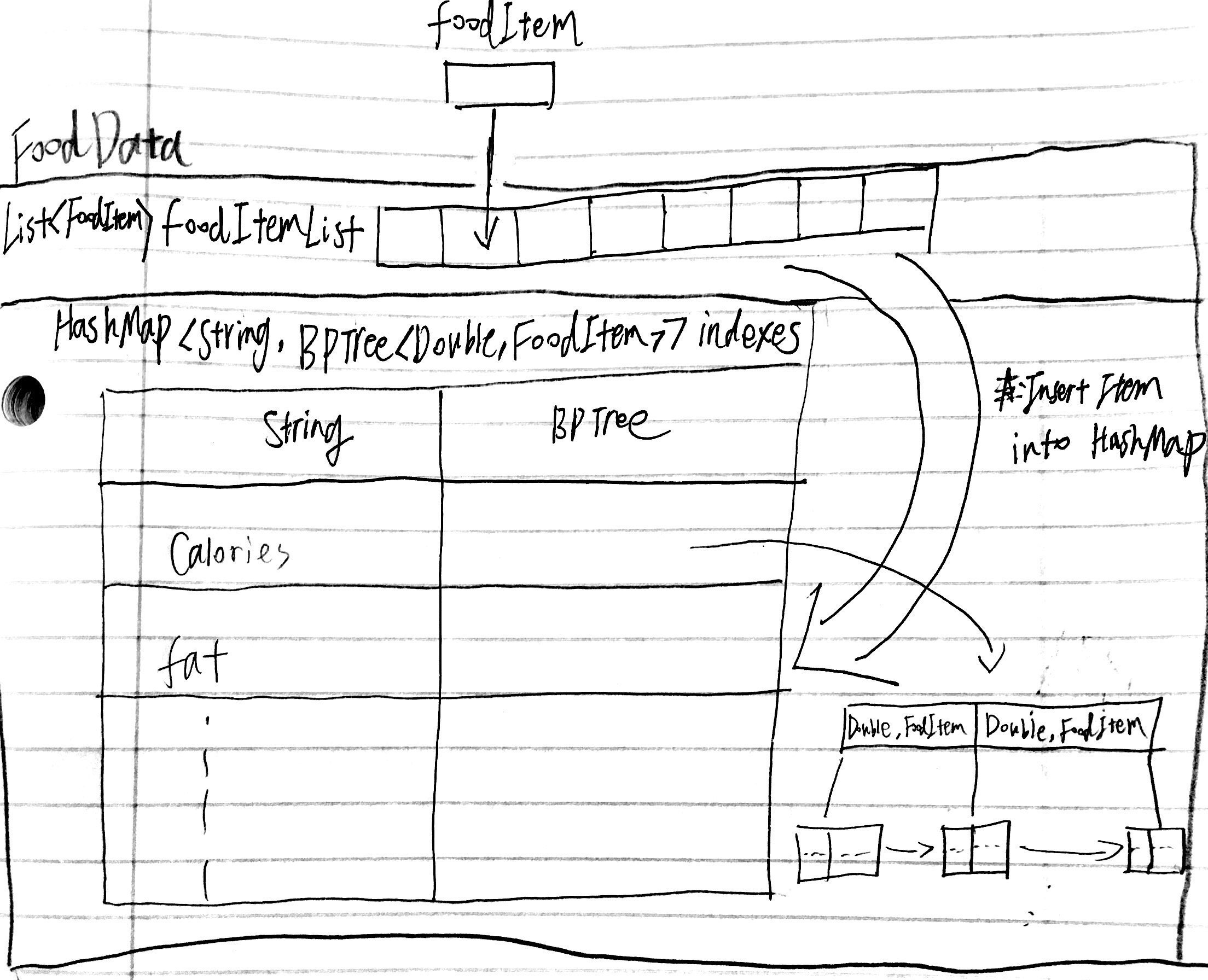
|  |  |  |  |
| --- | --- | --- | --- |
| Return type | Method name | Parameter List | Description |
| void | loadFoodItems | String filePath | Load food data from the file |
| List<FoodItem> | filterByName | String substring | Show only food with name containing the substring |
| List<FoodItem> | filterByNutrients | List<String> rules | List the food that fulfill the provided rule |
| void | addFoodItem | FoodItem foodItem | Add a food item to the loaded data |
| List<FoodItem> | getAllFoodItems | none | Get the list of all food items |
| void | saveFoodItems | String filename | Save the list of food items in ascending order |

FoodItem:

|  |  |  |  |
| --- | --- | --- | --- |
| **Return type** | **Method name** | **Parameter List** | **Description** |
| String | getName | none | Return the name of the food |
| HashMap<String, Double> | getNutrients | none | Return all the nutrient info |
| void | addNutrient | String name, double value | Adds a nutrient and its value to this food. If nutrient already exists, updates its value. |
| double | getNutrientValue | String name | Returns the value of the given nutrient for this food item. If not present, then returns 0. |

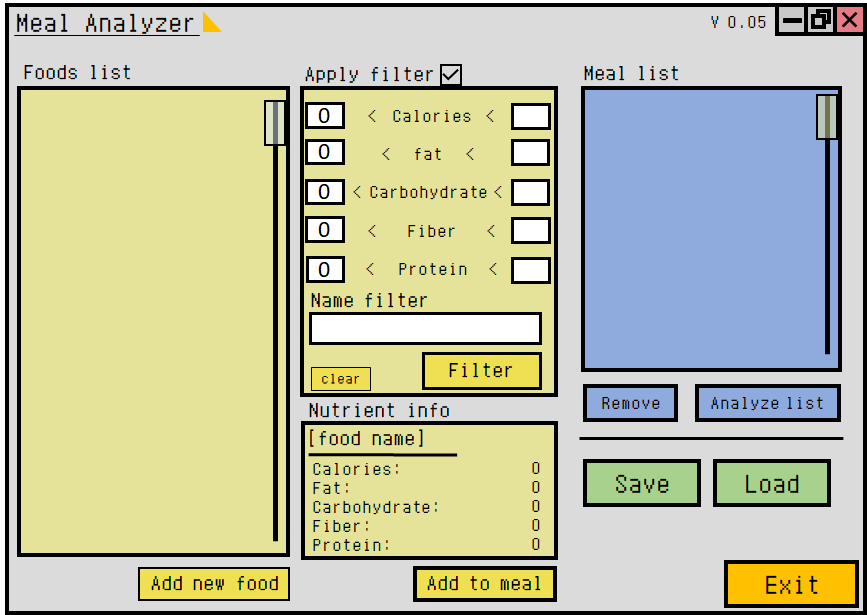
### Object Diagram - a sketch or list the object instances that exist when program first is first launched

* + show what instances (objects) exist when the program's main GUI page has been created
  + show relationships between objects with edges and labels describing the connections.
  + you are not required to additional objects that will be created when program is being used
  + (just show those that exist at start)
  + label your figure and explain any abbreviations or symbols you use



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instances |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### GUI Layout Sketch - an image saved in your design document



## **3. Submit your design document here, as a single pdf file named design.pdf**