UML Diagram

The diagram below shows how our objects interact with each other. We used "Extend" arrows to show inheritance and "Use" arrows to show dependencies. In addition, we've specified various aggregation and composition relationships with relevant arrows, and with field types. Setters and getters are assumed.

IngredientList

+ ingredients: list of **Ingredient**

+ addIngredient(ingredient: Ingredient): void
+ deleteIngredient(ingredient: Ingredient): boolean

Ingredient

+ unit: String
+ amount: int
+ name: String
+ expDate: Date
+ warningThreshold: int

Extends Extends Extends

RecipeGenerator

+ account: **Account**
+ useDigitalPantry: boolean
+ usingIngredients: list of **Ingredient**

+ showOptions(): void
+ getDigitalPantryIngredients(): void
+ getRecipes: list of **Recipe**

GroceryList

+ storeFinder: **StoreFinder**

DigitalPantry

+ closestExpiration: DateTime
+ lastUpdated: DateTime
+ account: **Account**

+ showWarnings(): void
+ showingOptions(): void
+ showIngredients(): void

THHandler

+ showWarnings(ing: list of **Ingredient**): list of **Ingredient**
+ queryWarnings(): list of **Ingredient**

Use

RecipeFilter

+ sortBy(attribute: String): list of **Recipe**

SpoonacularHandler

- apiURL: String

+ generateRecipes(ing: list of **Ingredient**,
             diet: list of String): list of **Recipe**

StoreFinder

+ list: **GroceryList**

+ findStore(): **GroceryStore**

Use

Use   Use

Recipe

+ url: String
+ name: String
+ usedIngredients: int
+ missingIngredients: int

IngChecker

+ checkSpoonacular(): list of **Ingredient**
- dropDownIngredients(): list of String

GroceryStore

+ name: String
+ location: String
+ apiURL: String

+ calcDistance(address: String): double

Use

Account

~ password: String
+ diet_pref: list of String
+ email: String
+ city: String
+ state: String

+ get_diet_pref(): list of string
+ login(username: String, pass: String)

StoreIngredientFilter

+ sortBy: int

+ getStores(sF: **StoreFinder**): **GroceryStore**

Use   Use

Use

DBHandler

- account: **Account**

+ getIngredients(): list of **Ingredient**
+ updateIngredient(ing: **Ingredient**): void
+ queryWarnings(): list of **Ingredient**