Detailed Design

for

Brew Day!

Version 1.0 approved

Prepared by

Atanasoff

Guo Rui 1630013011

Ji Jia 1630003023

Xie Qizhou 1630003056

Chen Mingxuan 1630003002

Atanasoff

20:30 10/04/2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
GUO Rui, JI Jia, Chen Mingxuan, Xie Qizhou.	2019/04/10	First version	Initial Version

1. Overview

1.1 Project description

"Brew Day!" is an application that allows home brewers to maintain an organized database of their beer recipes. The application allows users to create, store and modify recipes, and later on delete them, if the user wishes to do so. And users can write note after using a specific recipe to brew beer.

1.2 References

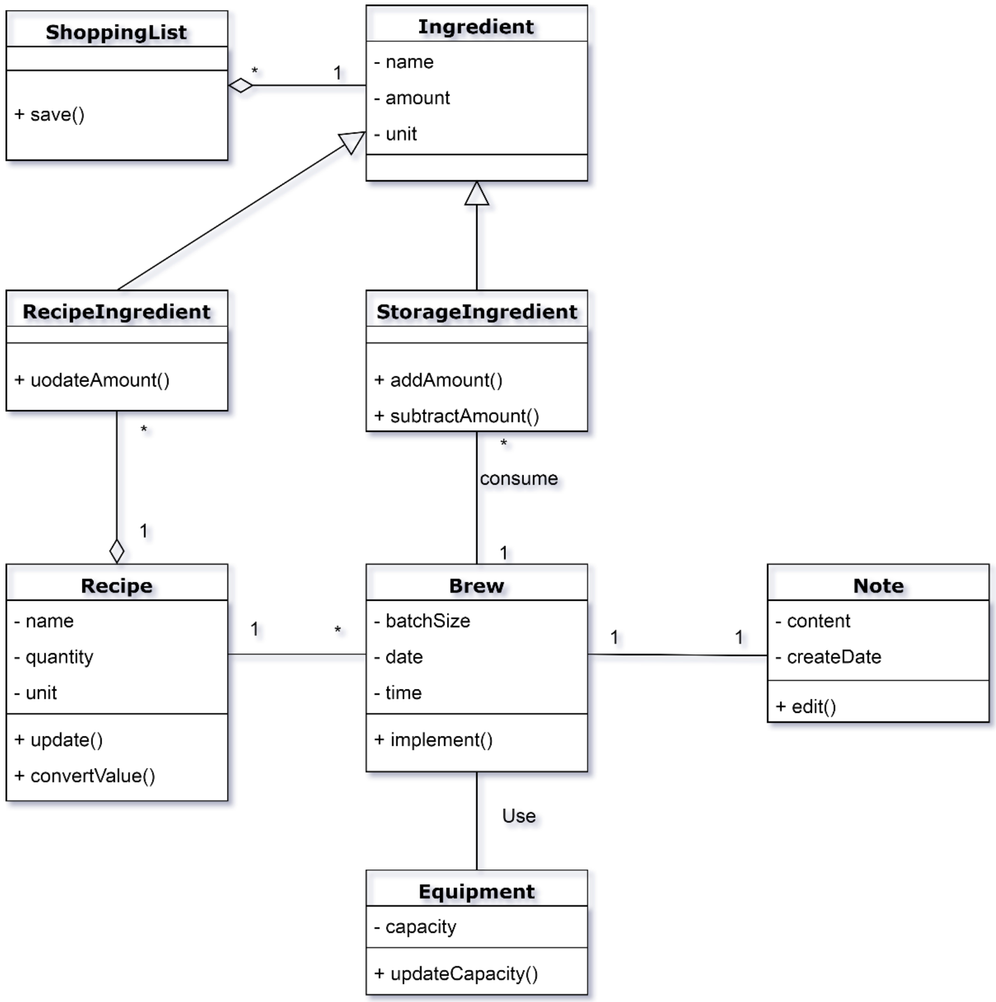
1. Guo Rui; Chen Mingxuan; Xie QiZhou; Ji Jia, "SRS_Atanasoff_v2.1", Atanasoff Company, 2019.
2. Guo Rui; Chen Mingxuan; Xie QiZhou; Ji Jia, "Architecture Design_Atanasoff_v1.0", Atanasoff Company, 2019.

1.3 Design purpose

The architecture design document is to divide the system into several parts and let users work together in a more efficient way.

2. Overall description

2.1 Class diagram

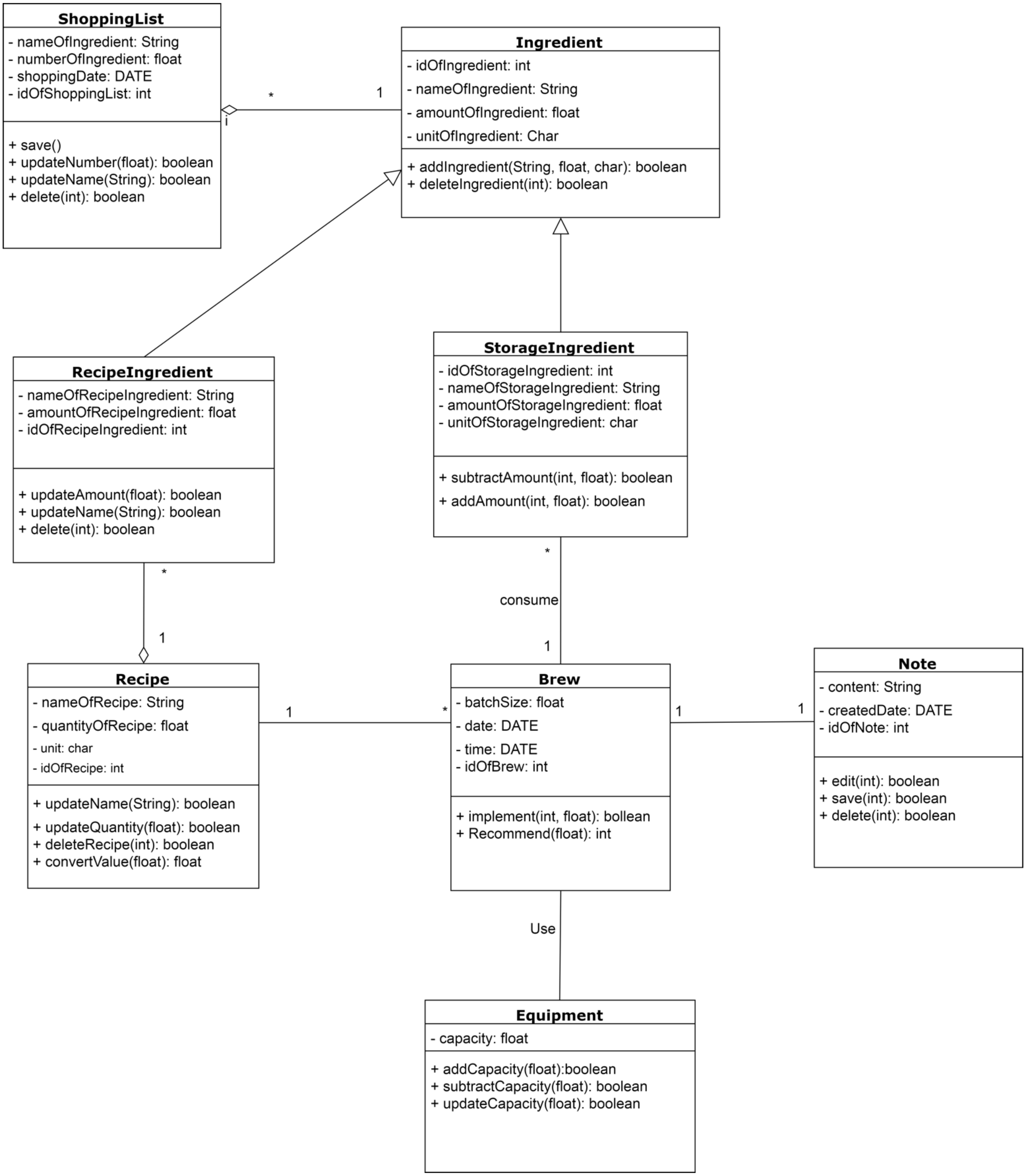


2.2 Refinements

In the restructuring class diagram, added a lot of detail about methods and attributes, refined the type of the attribute, the return value of the function, and the data type read.

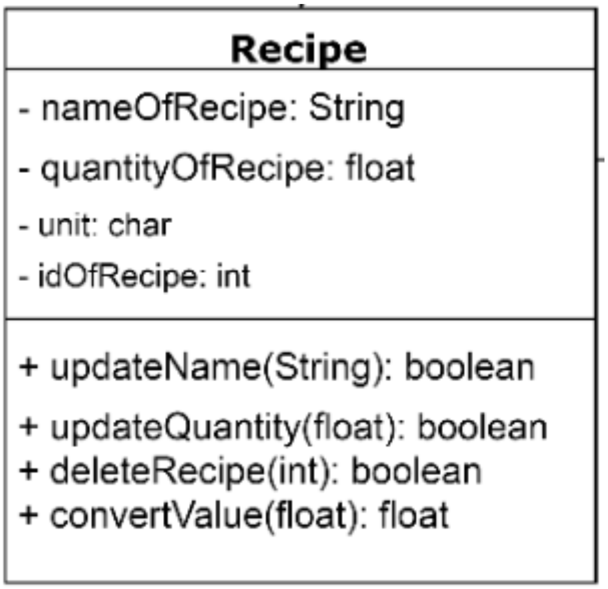
3. Detailed design

3.1 Class diagram


ShoppingList
- nameOfIngredient: String
- numberOfIngredient: float
- shoppingDate: DATE
- idOfShoppingList: int
+ save()
+ updateNumber(float): boolean
+ updateName(String): boolean
+ delete(int): boolean
*
1
i
Ingredient
- idOfIngredient: int
- nameOfIngredient: String
- amountOfIngredient: float
- unitOfIngredient: Char
+ addIngredient(String, float, char): boolean
+ deleteIngredient(int): boolean
RecipeIngredient
- nameOfRecipeIngredient: String
- amountOfRecipeIngredient: float
- idOfRecipeIngredient: int
+ updateAmount(float): boolean
+ updateName(String): boolean
+ delete(int): boolean
StorageIngredient
- idOfStorageIngredient: int
- nameOfStorageIngredient: String
- amountOfStorageIngredient: float
- unitOfStorageIngredient: char
+ subtractAmount(int, float): boolean
+ addAmount(int, float): boolean
*
1
*
consume
1
Recipe
- nameOfRecipe: String
- quantityOfRecipe: float
- unit: char
- idOfRecipe: int
+ updateName(String): boolean
+ updateQuantity(float): boolean
+ deleteRecipe(int): boolean
+ convertValue(float): float
1
*
Brew
- batchSize: float
- date: DATE
- time: DATE
- idOfBrew: int
+ implement(int, float): bollean
+ Recommend(float): int
1
1
Note
- content: String
- createdDate: DATE
- idOfNote: int
+ edit(int): boolean
+ save(int): boolean
+ delete(int): boolean
Use
Equipment
- capacity: float
+ addCapacity(float):boolean
+ subtractCapacity(float): boolean
+ updateCapacity(float): boolean

3.2 Classes

3.2.1 Recipe



Recipe

- nameOfRecipe: String
- quantityOfRecipe: float
- unit: char
- idOfRecipe: int

+ updateName(String): boolean
+ updateQuantity(float): boolean
+ deleteRecipe(int): boolean
+ convertValue(float): float

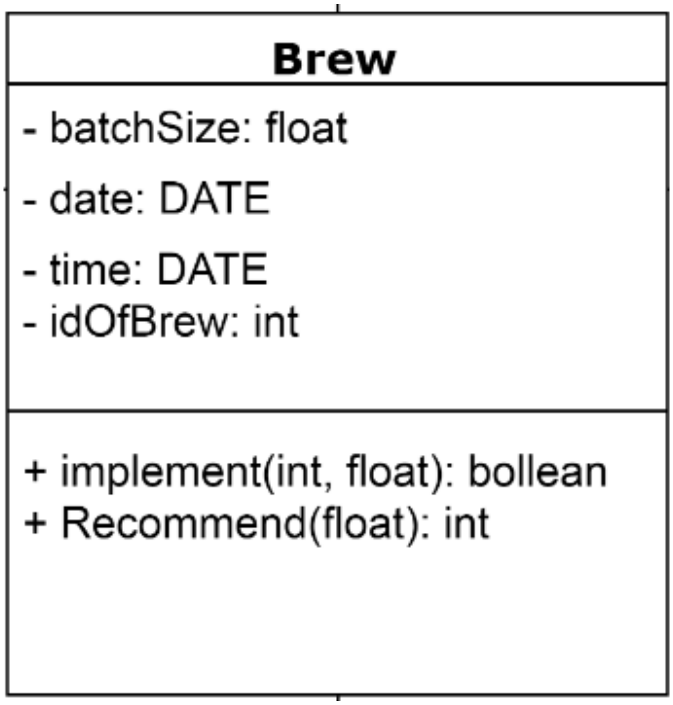
Explanations

Comparing to the original class “Recipe”, the new “Recipe” class give the type of the certain attributes. For example, the nameOfRecipe is assigned to a string and the quantityOfRecipe is assigned to a float number. Also, we add a new attribute idOfRecipe into the Recipe class. This id number can help us to find the recipe in the database more easily. What’s more, the update function is now becoming updateName and updateQuantity, this can allow the user to make more changes. Lastly, the deleteRecipe function is also added, this can allow the user to delete the recipe when they want. All the attributes are hidden and all the function is public.

Constraints (optional)

N/A

3.2.2 Brew



Explanations

In this “Brew” class, we add one new attribute idOfBrew and set the type to be integer. All the attributes in this class are set to private. For the implement function, there are two input values: an integer and a float number. The integer is the attribute idOfBrew and the float number is the batchSize. And this function will return a Boolean value to show that the brew process is success or not. Moreover, a new function “Recommend” is also added, this will recommend a brewing recipe to the user.

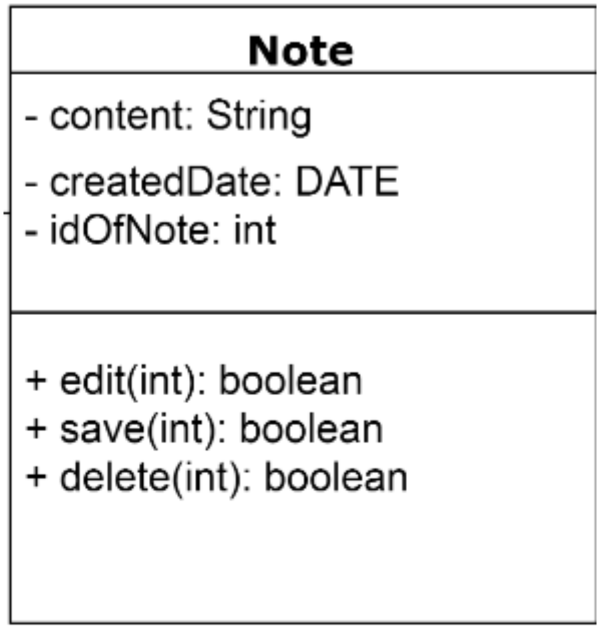
Constraints (optional)

Boolean = {NO_ENOUGH_INGREDIENT (0), OK (1)}
Boolean implement (int, float);

Pre-condition: the integer and the float number are not null.
Pose-condition: the return value shows that the brew process success or not.

1. If the ingredient(int) cannot support the amount of brewing number(float), that means, the ingredient is not enough for this brew action, return NO_ENOUGH_INGREDIENT (0).
2. Otherwise, return OK (1).

3.2.3 Note

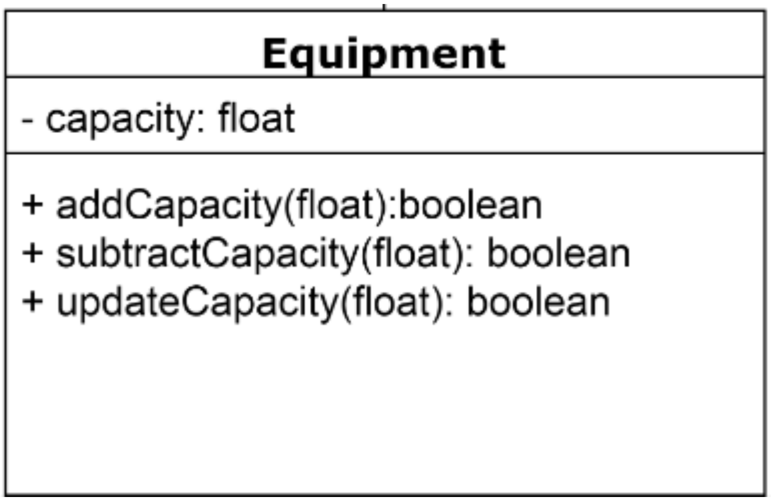


Explanations

In this Note class, there are totally 3 attributes. The content is assigned to string type and the createdDate is assigned into DATE type. Since Java didn't provide the type DATE, we need to define the DATE type by ourselves. What's more, a new attribute idOfNote in integer type is also added. Just like the previous ids', this can help in managing the database. Besides edit, two new functions save and delete is also added, this allow user to manage the note by themselves. All the three functions will need a integer as input and it will return a Boolean value to show that the function is successfully finished.

Constraints (optional)
N/A

3.2.4 Equipment



Equipment

- capacity: float

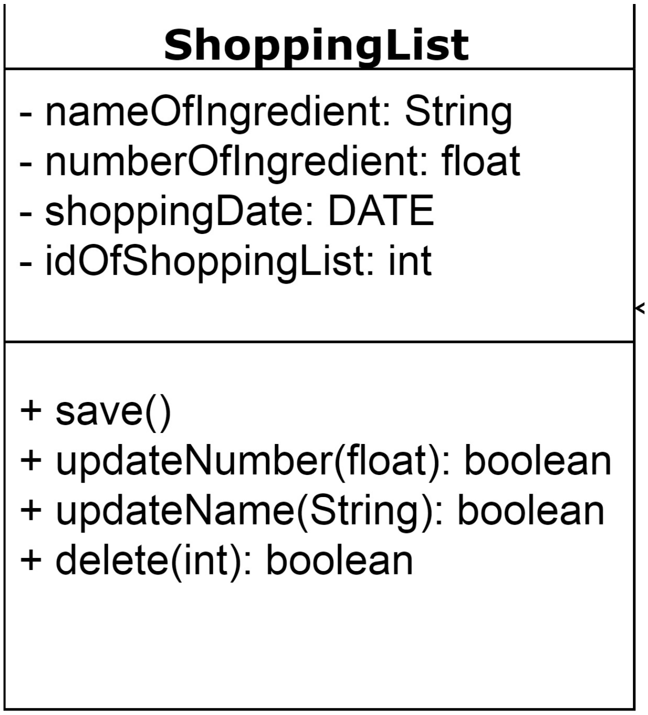
+ addCapacity(float):boolean
+ subtractCapacity(float): boolean
+ updateCapacity(float): boolean

Explanations

In this Equipment class, we didn't add any new attributes to the class. The only attribute is a float number capacity. And we add two more functions addCapacity and subtractCapacity. All these three functions will need a float number as an input value and it will return a Boolean value to show that the process is successfully finished or not.

Constraints (optional)
N/A

3.2.5 ShoppingList



ShoppingList

- nameOfIngredient: String
- numberOfIngredient: float
- shoppingDate: DATE
- idOfShoppingList: int

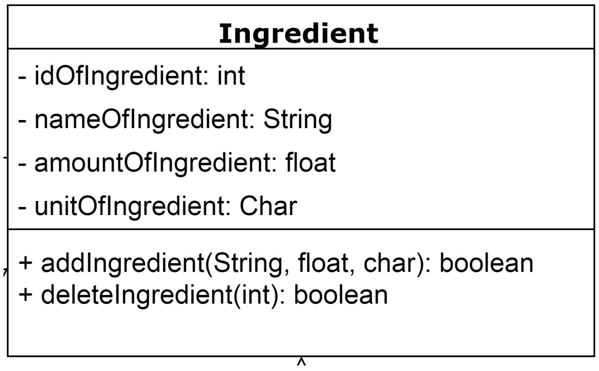
+ save()
+ updateNumber(float): boolean
+ updateName(String): boolean
+ delete(int): boolean

Explanations

In this ShoppingList class, compare to the same class in section 2.1, there are 4 new attributes: nameOfIngredient which is String type; numberOfIngredient which is float type; shoppintDate which is DATE type; and idOfShoppingList which is int type. There are 3 new methods: updateNumber function which use float type data and return with boolean type; updateName function which use String type data and return with boolean type; and delete function which use int type data and return with boolean type.

Constraints (optional)
N/A

3.2.6 Ingredient



Ingredient

- idOfIngredient: int
- nameOfIngredient: String
- amountOfIngredient: float
- unitOfIngredient: Char

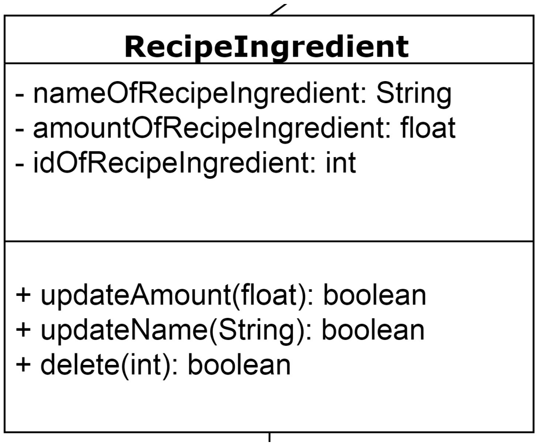
+ addIngredient(String, float, char): boolean
+ deleteIngredient(int): boolean

Explanations

In this Ingredient class, compare to the same class in section 2.1, there are 1 new attributes and 3 changes with origin attributes: idOfIngredient which is int type; nameOfIngredient which is String type; amountOfIngredient which is float type; and unitOfIngredient which is char type. There are 2 new methods: addIngredient function which use String, float, char type data and return with boolean type; deleteIngredient function which use int type data and return with boolean type.

Constraints (optional)
N/A

3.2.7 RecipeIngredient



RecipeIngredient

- nameOfRecipeIngredient: String
- amountOfRecipeIngredient: float
- idOfRecipeIngredient: int

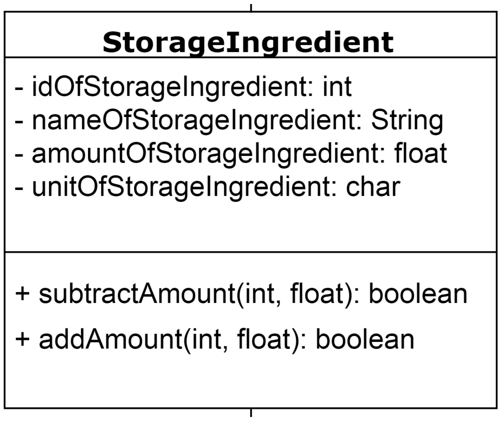
+ updateAmount(float): boolean
+ updateName(String): boolean
+ delete(int): boolean

Explanations

In this RecipeIngredient class, compare to the same class in section 2.1, there are 3 new attributes: nameOfRecipeIngredient which is String type; amountOfRecipeIngredient which is float type; idOfRecipeIngredient which is int type. There are 2 new methods and 1 change with origin methods: updateAmount function which use float type data and return with boolean type; updateName function which use String type data and return with boolean type; delete function which use int type data and return with boolean type.

Constraints (optional)
N/A

3.2.8 StorageIngredient



StorageIngredient

- idOfStorageIngredient: int
- nameOfStorageIngredient: String
- amountOfStorageIngredient: float
- unitOfStorageIngredient: char

+ subtractAmount(int, float): boolean
+ addAmount(int, float): boolean

Explanations

In this StorageIngredient class, compare to the same class in section 2.1, there are 4 new attributes: idOfStorageIngredient which is int type; nameOfStorageIngredient which is String type; amountOfStorageIngredient which is float type; unitOfStorageIngredient which is char type. There are 2 change with origin methods: subtractAmount function which use int, float type data and return with boolean type; addAmount function which use int, float type data and return with boolean type.

Constraints (optional)
N/A

4. Alternative detailed design (Optional)

N/A

5. More considerations

This diagram could be changed according to the situation, when reading this document, please contact us if you cannot understand well.