

Work Designation:

- CRC: All
- Scenario Walkthrough: All
- Code:
 - Ingredients - Yuanyue Qiao
 - Tag - Yeming
 - Contents - Emily Hu
 - Dishes - Yuxuan Yang
 - Subclasses of Dishes - Emily Hu
 - ChouTuanMaster - Yuxuan Liu
 - Command Line Interface - Yuxuan Liu
 - Favorites - Ruiyang Xu

Specifications:

Our program (ChouTuanEATS) is supposed to be a cuisine search & recommendation system that enables users to take various actions. The program will have the following functions; A search system for cuisines, which allows the user to search for either the name or the “tags” of the cuisine; A favorite system for cuisines, which allows the users to store and remove their preferred cuisines in a favorite list. Such a list can be sorted differently according to the user’s preference; A system for calculating and displaying each cuisine’s calories and cooking instructions; A cmd system for processing all the user input.

CRC

| Ingredients (Entity) | |
|---|---|
| <ul style="list-style-type: none">- generate names of ingredients- generate calories of ingredients- generate weight of ingredients | Dishes Tag Contents cmdInterface |

Next steps for ingredients:

We may want to include a database which stores all data instead of using tag if possible. Therefore, the categories of ingredients (i.e. meat, vegetables, eggs, etc) will be specified in the ingredient class.

| Tag (Entity) | |
|---|-------------------------|
| <ul style="list-style-type: none"> - Generate tags for categories of ingredients - Getters for names of ingredients - Add ingredients and ingredients' categories to Map | Ingredients Contents |

| Contents (Use Case) | |
|--|------------------------------|
| <p>Add:</p> <ul style="list-style-type: none"> - Add dish instances to storage by category of cooking method. <p>Storage:</p> <ul style="list-style-type: none"> - Stores for list of fried dishes, boiled dishes, steamed dishes, grilled dishes <p>Getters:</p> <ul style="list-style-type: none"> - Getters for list of fried dishes - Getters for list of Boiled dishes - Getters for list of Steamed dishes - Getters for list of Grilled dishes <p>Search:</p> <ul style="list-style-type: none"> - Returns list of existing dishes (in storage) according to dish and/or tag searched. <p>TODO:</p> <ul style="list-style-type: none"> - Add ingredients, organised and stored by tag (?) - Getters for tags and ingredients - Storage tags and ingredients | Tag Dishes Ingredients |

Next Steps for Content:

- Move everything to the database if possible.

- Improve search method:
 - Enable search by ingredient and tag
 - Not only generating a list of dishes that fully matches the input

| Dishes (Entity) Subclass: Fry, Steam, Boil, Grilled | |
|--|--|
| <ul style="list-style-type: none"> - Getter for name of the dish (get_dish_name) - Getter for calories of this dish (get_calories) - Getter for instructions of how to cook this dish (get_instructions) - Getter for ingredients of this dish (get_ingredients) - Generate name, total calories, instructions, and ingredient_list of the dish (initialize them in the constructor) <p>*Users add instructions in addNewDish ex. tomato 500g, egg 500g, fried * 2 total calories = (500*tomato calories + 500 *egg calories) * 2 (research)</p> | Ingredients Fry Steam Boil Grilled |

| Fried (Entity) Parent: Dishes | |
|--|--------|
| <ul style="list-style-type: none"> - Override total calorie attribute to generate total calories. | Dishes |

| | |
|--|----------------|
| Steamed (Entity) | Parent: Dishes |
| - Override total calorie attribute to generate total calories. | Dishes |
| Boiled (Entity) | Parent: Dishes |
| - Override total calorie attribute to generate total calories. | Dishes |

| | |
|--|----------------|
| Grilled (Entity) | Parent: Dishes |
| - Override total calorie attribute to generate total calories. | Dishes |

Next Steps for Dish Subclasses (Fried, Boiled, Steamed, Grilled)

- Improve accuracy of calorie calculation by doing a bit of research.
- Apply a constant variable to the total_calories attribute under each respective subclass instead of an integer.

| | |
|--|------------------------|
| ChouTuanMaster (Presenter) | |
| <ul style="list-style-type: none">- Stores main method- Present Welcome and Ending words. | Command line Interface |

| Command line Interface(Presenter & Controller) | |
|--|---|
| <ul style="list-style-type: none"> - Present instruction and check user's input for basic three functions(Add new dish, search for dishes and manage favourite list). - Present dishes list returned from Content and Favourites class's method. | ChouTuanMaster Content Favourites |

| Favourites (Use Case Class) | |
|---|----------|
| <p>Getters:</p> <ul style="list-style-type: none"> - getNames - getCalories - getfavouriteRank - getFavouriteList <p>Use Case methods:</p> <ul style="list-style-type: none"> - addFavourite - removeFavourite - containDish | Contents |

Next Steps for Favourites:

The next steps are to implement the getFavouriteRank method, which is the core of the favourite class; allow the users to sort their favourite list according to their preferences.

Scenario Walkthrough

Adding new dishes:

Users clicking on “Add new dishes” button

a page shown “ingredients, dishes name, cooking method, instructions” (users need to fill in)

users clicking on “Finish” and the dish will be added

Searching for an existing dish:

Search for name of ingredients or tags or name of dishes (either from dishes added from users or initial database)

Select the dishes users would like to see

Show total calories, ingredients, and cooking instructions

Users can add to favourites

Return to searching engine (main page)

Digging “Favourites”:

Users can click on “Favourites”

All favourite dishes will be shown

Dishes can be removed from “Favourites”

Users have three options by starting the program which are adding new dishes, searching for an existing dish, and digging their favorite lists. First of all, in order to add a new dish, users will need to choose the “Add new dishes” option. Then the page will show a list of blanks about the name of the dish, total calories of the dish, required ingredients and corresponding calories per gram as well as required weight for the dish. A new dish will be successfully added.

Considering searching for a new dish, users will need to search for the name of the dishes. A list of dishes will be shown. By Choosing the interested dishes, a page about the dish will be shown. Contents will be on total calories, ingredients as well as the cooking instructions. On that page, users can choose to add the dish into their favourite lists.

Last but not least, users can view their favourite lists by simply clicking on the “Favourite” button. Then a list of favourite dishes will be shown. Users can remove dishes from the favourite list.