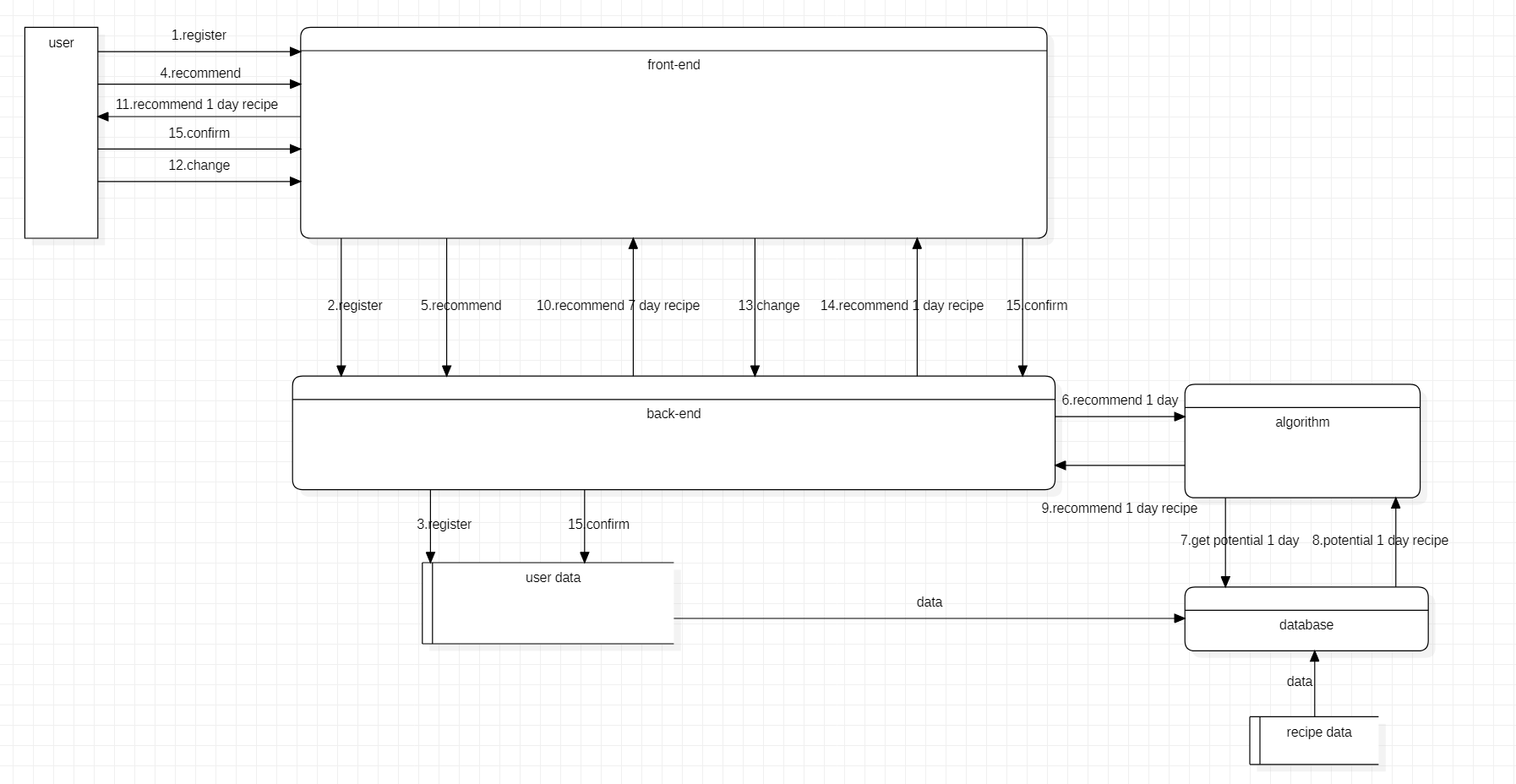
System Design Specification

****

## Database

Tables

1. food

Description: Store all the attributes of each food which is fetched from the Internet

Attributes:

（1）food\_id

type: INT

NULL: not NULL

（2）food\_name

type:VARCHAR(20)

Null: not NULL

（3）food\_kind

type:VARCHAR(20)

Null: not NULL

（4）heat

type: DOUBLE

Null: not NULL

（5）protein

type: DOUBLE

Null: not NULL

（6）carbohydrate

type: DOUBLE

Null: not NULL

（7）fat

type: DOUBLE

Null: not NULL

（8）weight\_loss\_suit

type: INT

NULL: not NUL

Primary Key: food\_id

Opertations:

（1）add\_food(self, food\_id, food\_name, food\_kind, heat, protein, carbohydrate, fat, weight\_loss\_suit)

（2）query\_food(self, food\_id)

1. User\_all

Description：Store all the user information

Attributes:

（1）user\_id

type: INT

NULL: not NULL

（2）user\_name

type:VARCHAR(20)

Null: not NULL

（3）user\_aim

type: INT

Null: could be NULL

（4）allergy

type: INT

Null: could be NULL

(5) user\_week(1)Opertations:

（1）add\_user(self, user\_id, user\_name, user\_aim,allergy)

（2）reset\_aim(self, aim)

(3) reset\_allergy(self,allergy)

1. UserInfo

Description：Store a specific user information

Attributes:

（1）user\_id

type: INT

NULL: not NULL

（2）food\_prefer\_1(1)

type:INT

Null: could be NULL

（3）food\_time\_1(0)

type: INT

Null: could be NULL

......

......

Opertations:

（1）reset\_food\_prefer (self,user\_id, food\_id, food\_prefer)

（2）reset\_food\_time (self,user\_id, food\_id, food\_time)

## API

Guan Mingfei

### API URL Address

http://server:port/api\_name

**#instance**

http://server:port/get\_recomd

http://server:port/login

### Test Method

curl -H 'content-type:application/json' -X POST -d 'JSON Input' http://server:port/api\_name

### API specifications

#### get\_recomd

For: Connect front-end interface and algorithm

Aim:

1. read recommendation parameters from front-end HTML form
2. call algorithm module
3. fill back result from algorithm to HTML result page and return.

Input **Format: HTML Form**

**Input Parameters:**

1. **Gender**
2. **Age**
3. **Height**
4. **Weight**
5. **Goal**

**0) increase muscle**

**1) lose weight**

**2) shaping**

1. **Target Weight**
2. **Activity Type**

**Output Format: Rendered Recommendation HTML**

**Output Parameters:**

1. **Caloeries Need**
2. **Protein Need**
3. **Carbohydate Need**
4. **Fat Need**
5. **Weeks Need**
6. **Recipe**

#### register

**For: Anomymous user to get a identity of the system**

**Aim:**

1. **Collect User information**
2. **Store in database**

Input **Format: HTML Form**

**Input Parameters:**

1. **username**
2. **password**

**Output Format: Rendered Recommendation HTML**

**Output Parameters:**

1. **Success**

#### change

**For:User to send a feedback to server that he doesn't like the recipe of the day**

**Aim:**

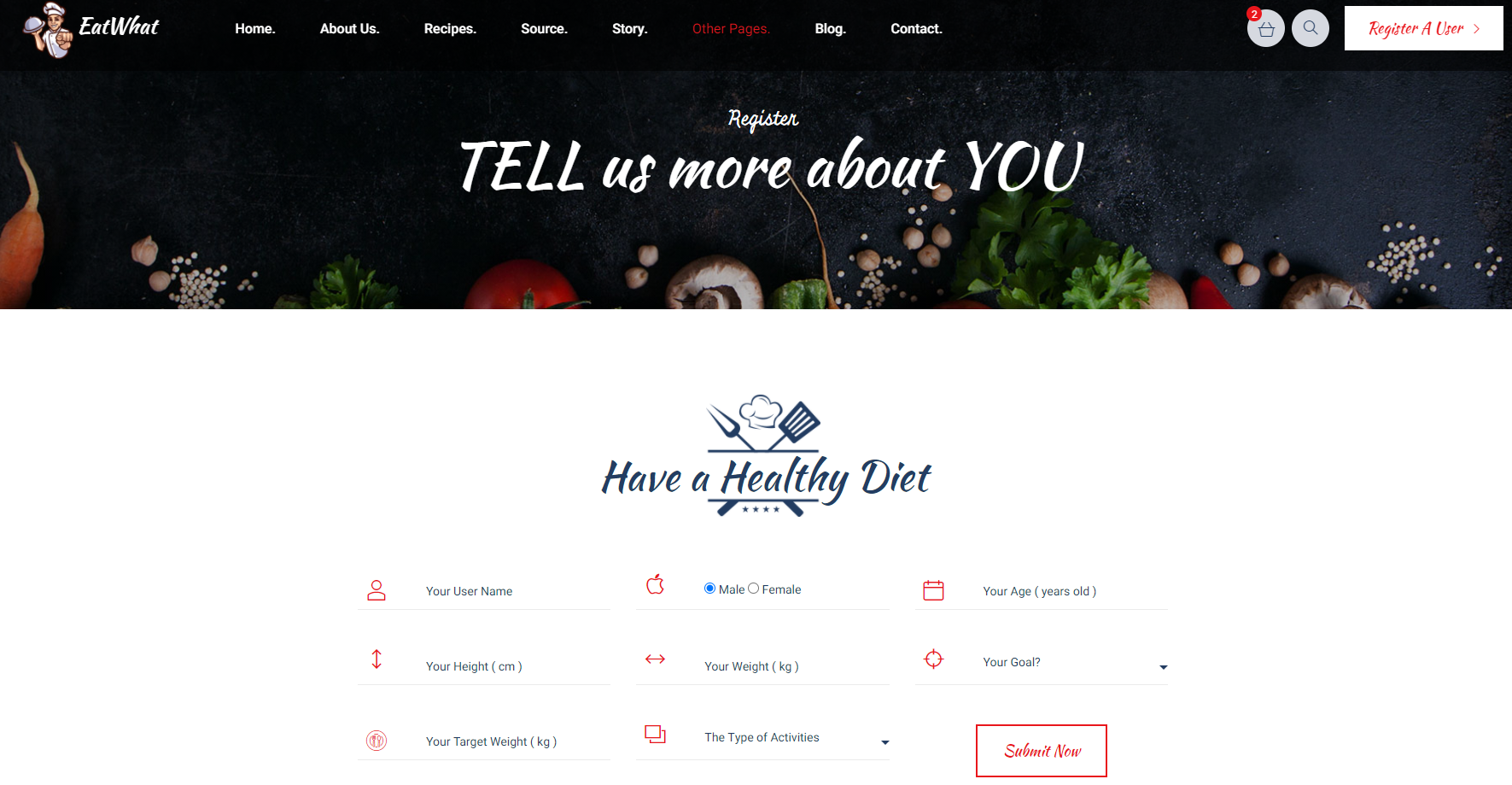
1. **Send feedback**
2. **Change the weight of the food**

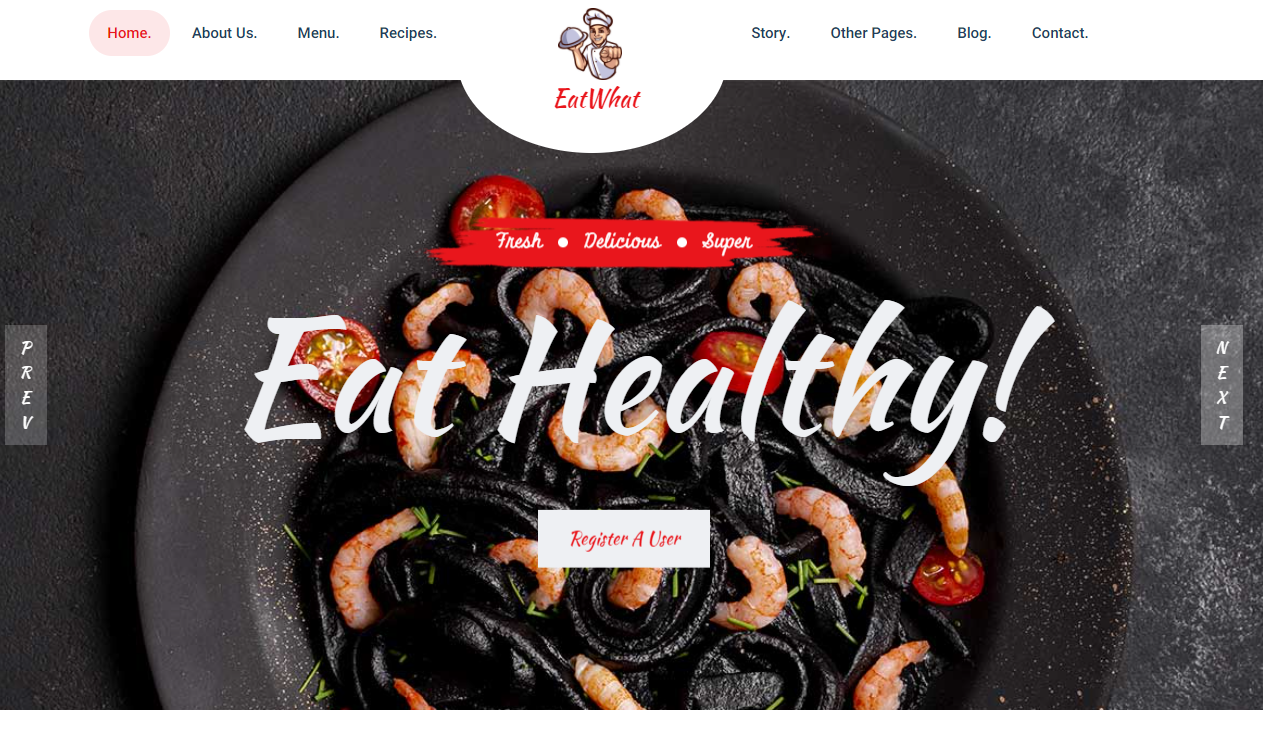
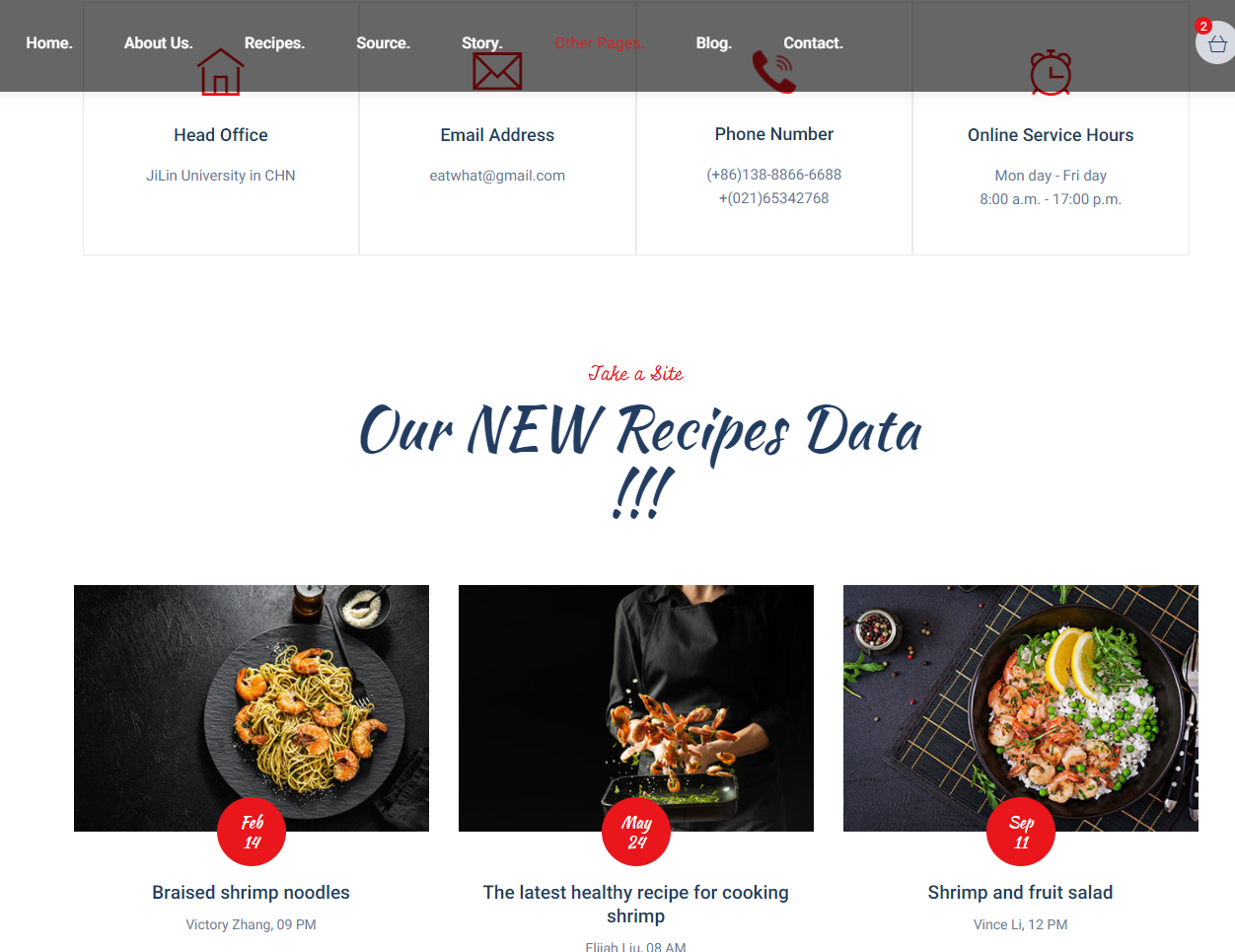
**Input Paramters:**

1. **Username**
2. **day**

**Output Paramters:None**

## Front End Design

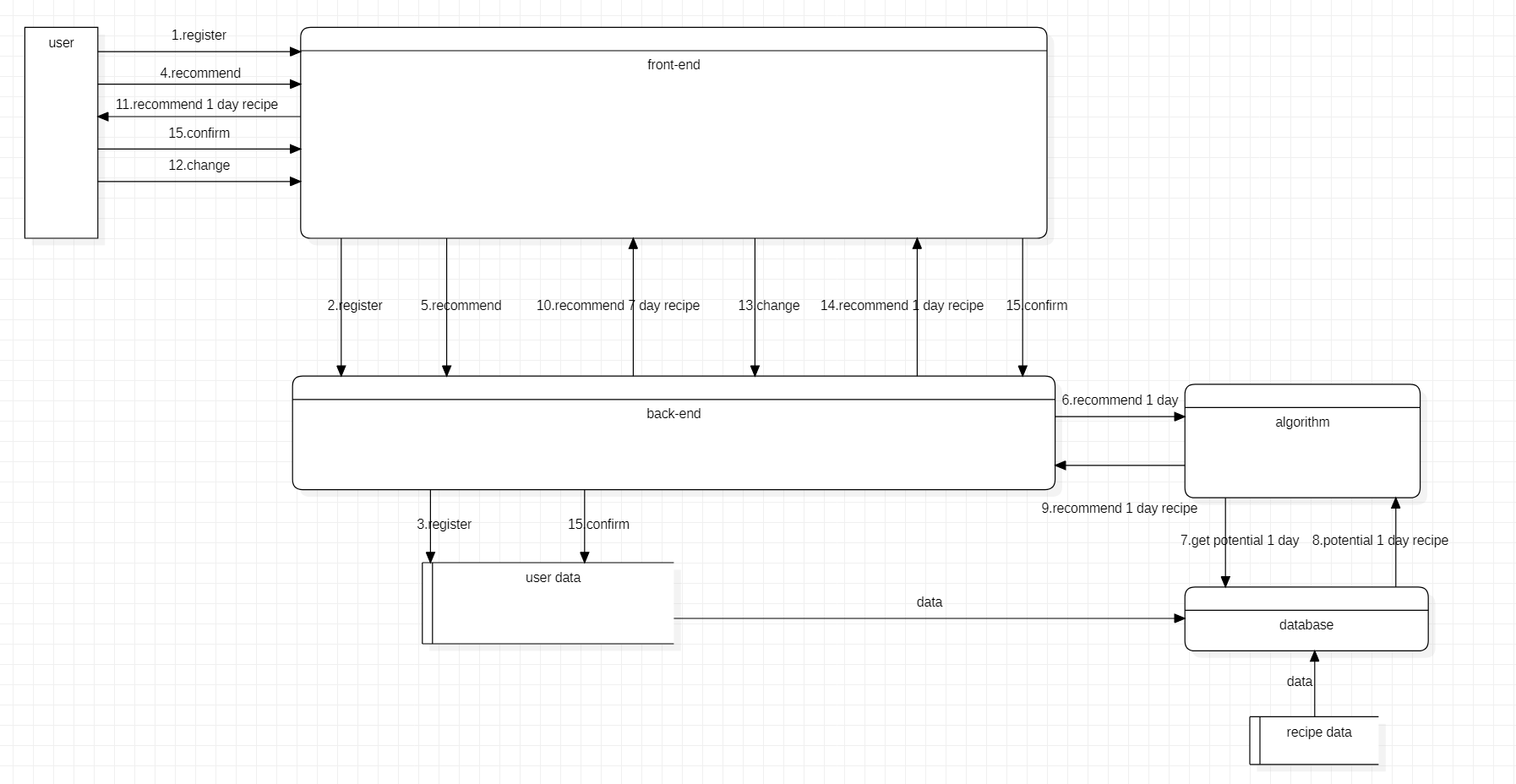
****

****

## 

## Algorithm Design

### Requirement Statement

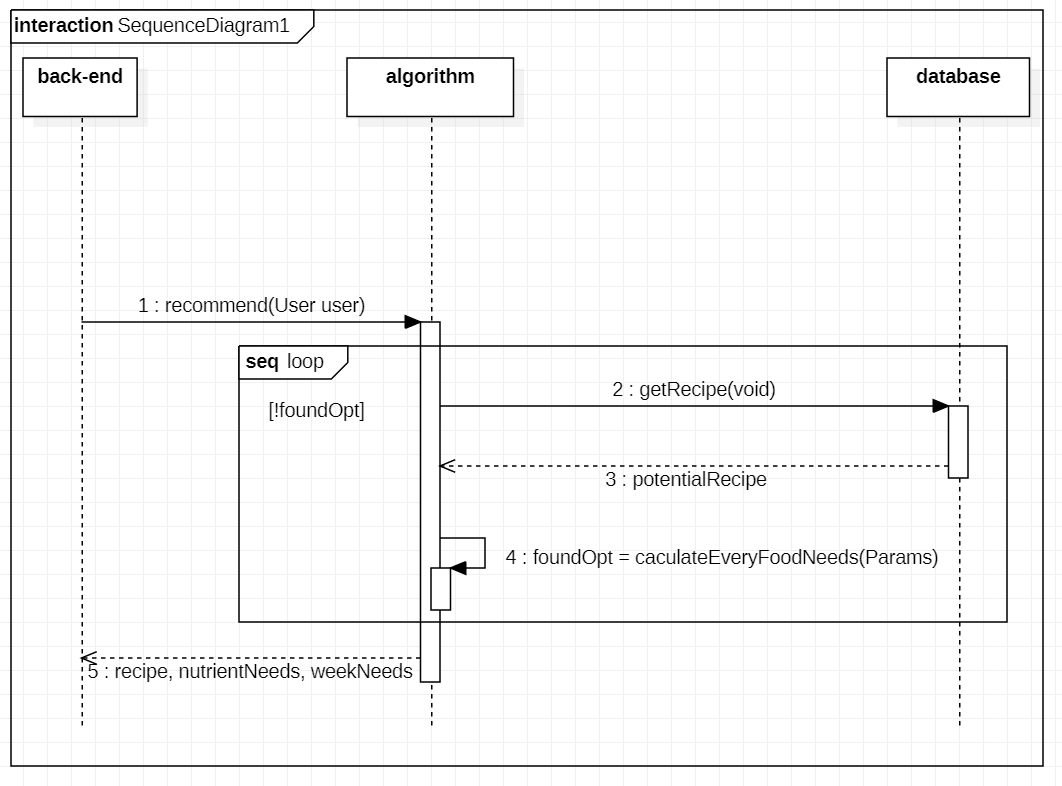
****

Algorithm module is called by the back-end module with parameters of userinfo like weight, height, age etc. Then it will call the database interface andget a potential recipe. Inside the algorithm module, we will evaluate whetherthis potential recipe will be recommended to the user by using the idea oflinear programing. If the potential recipe is good enough, we will recommend itwith the net content of each dish. Otherwise, we will try another potential recipe.

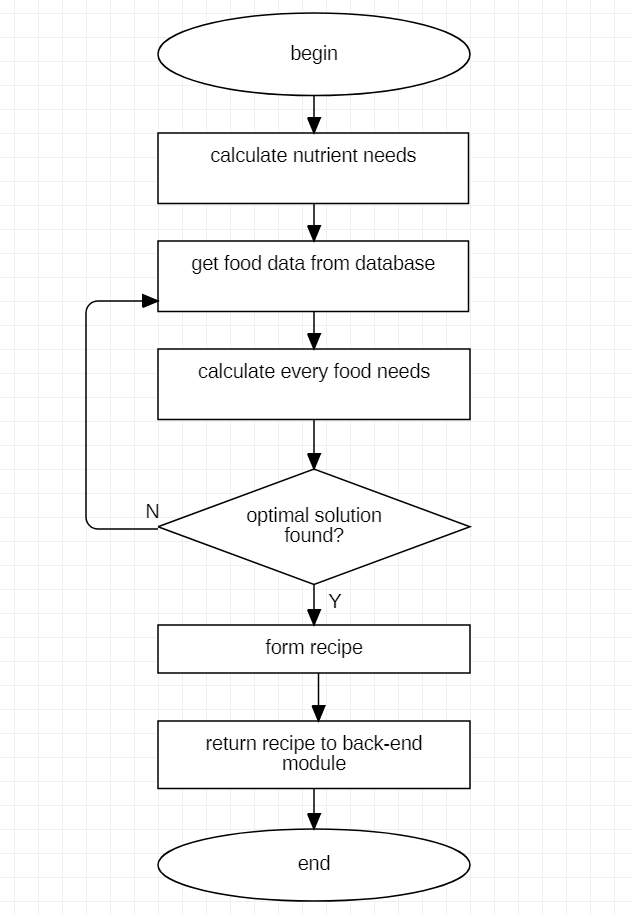
Generally, we provide an interface which recommend one day’s recipe. Theback-end module can get one-week recipes by calling it seven times. Besidesthat, it can also be used to change one day’s recipe.

#### 1.1 Sequence Diagram

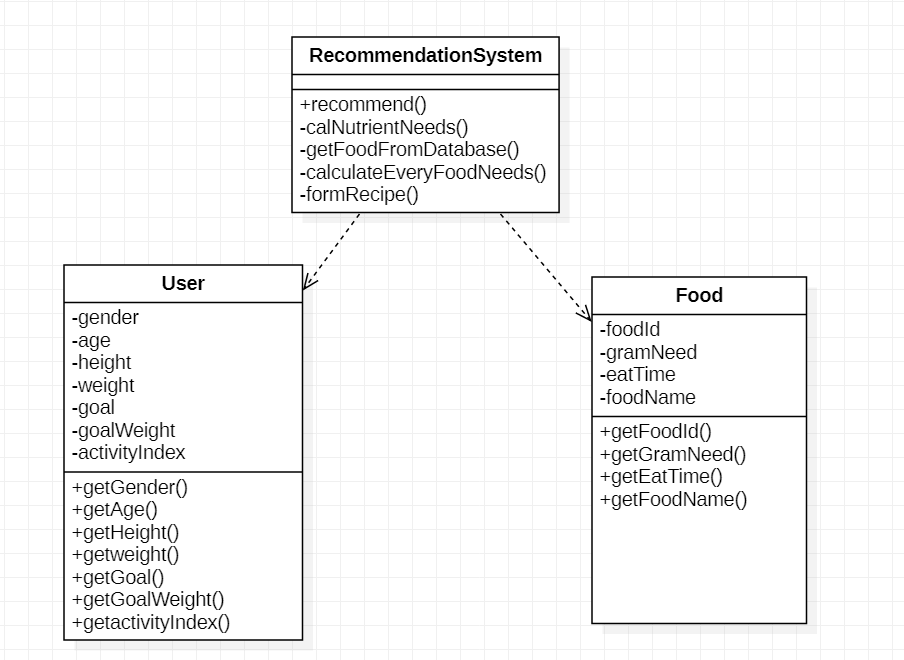
The above relations can be clearly shown by the Sequence Diagram:



#### 1.2 Flow Chart Diagram



#### 1.3 Class Diagram



### 2.Service Provided

|  |  |  |
| --- | --- | --- |
| Service | Provided  By | Tested  By |
| recommend | algorithm | Tester |

### 3.Access Method Effects

|  |  |
| --- | --- |
| **Access**  **Method** | **Description** |
| recommend | Generally speaking, This method takes user infomation as input and recommends a recipe back to the user. |

## Crawler

I use request and BeautifulSoup libraries to crawl food websites. And I faced to the following problems: if you visit too frequently, the crawler will be banned by the website, so I take the following measures:

1. Wait a while before visiting another webpage: avoid visiting URL too frequently and wait a while before crawling.
2. Change the IP: when it is identified and banned by the website, you can change the IP (such as using mobile phone hotspots), and then you can continue to crawl.

Data Cleaning

1.Zero Value, some food data all nutrients are 0, obviously abnormal data, for this kind of data, discarded .

2.Missing value: some foods are missing values in one or both nutritional indicators. For missing values, the strategy of filling the corresponding characteristics of a similar food is adopted.

3.Tag: the food website mixes vegetables, fruits, meat and eggs together, and because the scale of the data is not very large, it is processed by manual tagging.

4.For the two characteristics of "food type" and "whether it is recommended for people to lose weight", digital enumeration is used instead of text features.

In addition, I was involved in the development of the front-end page this week.

Data cleaning.

1.Zero Value, some food data all nutrients are 0, obviously abnormal data, for this kind of data, discarded .

2.Missing value: some foods are missing values in one or both nutritional indicators. For missing values, the strategy of filling the corresponding characteristics of a similar food is adopted.

1. Tag: the food website mixes vegetables, fruits, meat and eggs together, and because the scale of the data is not very large, it is processed by manual tagging.

4.For the two characteristics of "food type" and "whether it is recommended for people to lose weight", digital enumeration is used instead of text features.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Grain | Staple food | Fruit | Vegetable | Meat | Dairy Products |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Recommended for consumption | Eat in moderation | Eat carefully | Not suitable for use |