

DB Model & DB Design Report (Due 7/5/2020)

Summer 2020 | CS 157A

GNT-Market

Grocery Nutrient Tracker

Team 11

Tracy Ho & Inhee Park

Functional Requirements

User Interaction with the App:

Main theme of this web app is to help users to get a convenient and healthy selection for the grocery list. Users can generate their grocery list by selecting a dish, which internally converts a dish to ingredient food items and/or by adding food items directly. User customized dietary restrictions and preferences are taken into consideration when generating a grocery list. This app also provides quantitative nutritional distribution for their grocery list as well as qualitative beneficial nutrient information for body systems.

1. [Dish \Rightarrow Grocery List \Rightarrow Nutrient/Calorie Information]

User inputs a desired dish from a menu in the app.

The app will provide a list of grocery items with nutritional distribution (with exceeding/deficient nutrition) and the calorie information of each item.

2. [Grocery List \Rightarrow Nutrient Information]

User inputs a list of grocery items in the app.

Then the app outputs cumulative nutritional distribution (with exceeding/deficient nutrition).

3. [Grocery Browser]

Users can browse food categories corresponding to section names of grocery aisles. Which users can then add food items into their grocery list.

4. [Dietary Restriction]

Users can set their dietary restrictions (e.g. food allergy) to the app.

Then if the user selects a grocery item that conflicts with their diet restriction, the app gives a warning.

5. [Browse Beneficial Food]

Users can browse a paired list of food and their respective beneficiary body systems, so they may tailor food items for their specific health needs.

6. [User Creation]

A user will be able to register their own account to use the web application with an email and password.

Then they will be able to use the application for their own usage.

7. [Nutrient Suggestions]

A user who may want to get more of a particular nutrient in their diet may look into suggestions of what is rich in those nutrients, so they can include it in their list of groceries to buy.

This will display a list of food items that contain a specific nutrient, giving the user an idea of what nutrients they are taking in their diet.

8. [Favorite Food Items]

Users are able to mark which items as favorites so they may be able to access the items easier, whether it is to look up more details of the items or to re-add the items in their grocery list.

Functions:

- **Frequently Purchased Grocery Items:** Keeping track of the number of times an item has been purchased, and store those items as frequently purchased items.
- **List of Dietary Restrictions:** Set dietary restrictions by adding specific food items to be avoided.
- **List of Beneficial Food:** Selects a body system from a dropdown menu that a user is concerned about, then the app will suggest healthy foods that can benefit that body system.
- **Create/Delete a Grocery Shopping List Card:** Users can create a card that will contain all of the grocery items that they want to purchase. Items can be added to this card, deleted, or updated accordingly.
- **Duplicate Grocery List:** Users that wish to reuse a previously made grocery list may be able to copy what they previously created.
- **Browse Grocery Categories,** such as:
 1. Beverages – coffee/tea, juice, soda
 2. Bread/Bakery – sandwich loaves, dinner rolls, tortillas, bagels
 3. Canned/Jarred Goods – vegetables, spaghetti sauce, ketchup
 4. Dairy – cheeses, eggs, milk, yogurt, butter
 5. Dry/Baking Goods – cereals, flour, sugar, pasta, mixes
 6. Frozen Foods – waffles, vegetables, individual meals, ice cream
 7. Meat – lunch meat, poultry, beef, pork
 8. Produce – fruits, vegetables

Entity Description

E1. User:

The User entity represents the User of GNT-Market. A user would have a user_id, email, and password.

E2. Dish:

The Dish entity represents the dish that is made up of several different foods. It contains a food_id and name that says that it is, and a short description of it.

E3. Food:

The Food entity represents a single food item that may be used in a relation to other entities.

E4. BodySystem:

This entity represents the different body systems. The entity set has body_sys_id, name, description to keep track of the different body systems.

E5. GroceryList:

GroceryList is an entity to represent the food list that the user creates by selecting a dish (then converting to ingredient food list) or by adding a food list directly. This contains list_id and date (in datetime type).

E6. Nutrient:

Nutrient is an entity to represent individual nutrients. This contains nutrient_id and name to keep track of the different nutrients.

E7. Category:

Category is an entity to represent a group of similar types of food. It will be retried later in the implementation stage to provide users with browning food by category. This contains category_id and category name.

Relationships

R1. User_creates_GroceryList: This relationship connects a grocery list to a user. Since a user creates a grocery list, a grocery list may only belong to one user.

R2. User_marks_Food: This relationship connects a user to a food item with conditions tied to the food item. The user may mark the food item as a Favorite and/or Restricted. A user may put conditions on multiple food items, and a food item may have conditions from multiple users.

R3. User_adds_Food: This relationship connects a user list to food. The user adds the food item to consider to buy. This also works for deleting food items.

R4. User_selects_Dish: This relationship connects a user list to dish. The user selects the dish item, which would be internally converted to a list of food items. This also works for de-selecting dish items.

R5. Dish_has_Food: This relationship represents what dish contains what food items. It connects dishes with food. A dish can have multiple food items, and a food item can be part of multiple dishes.

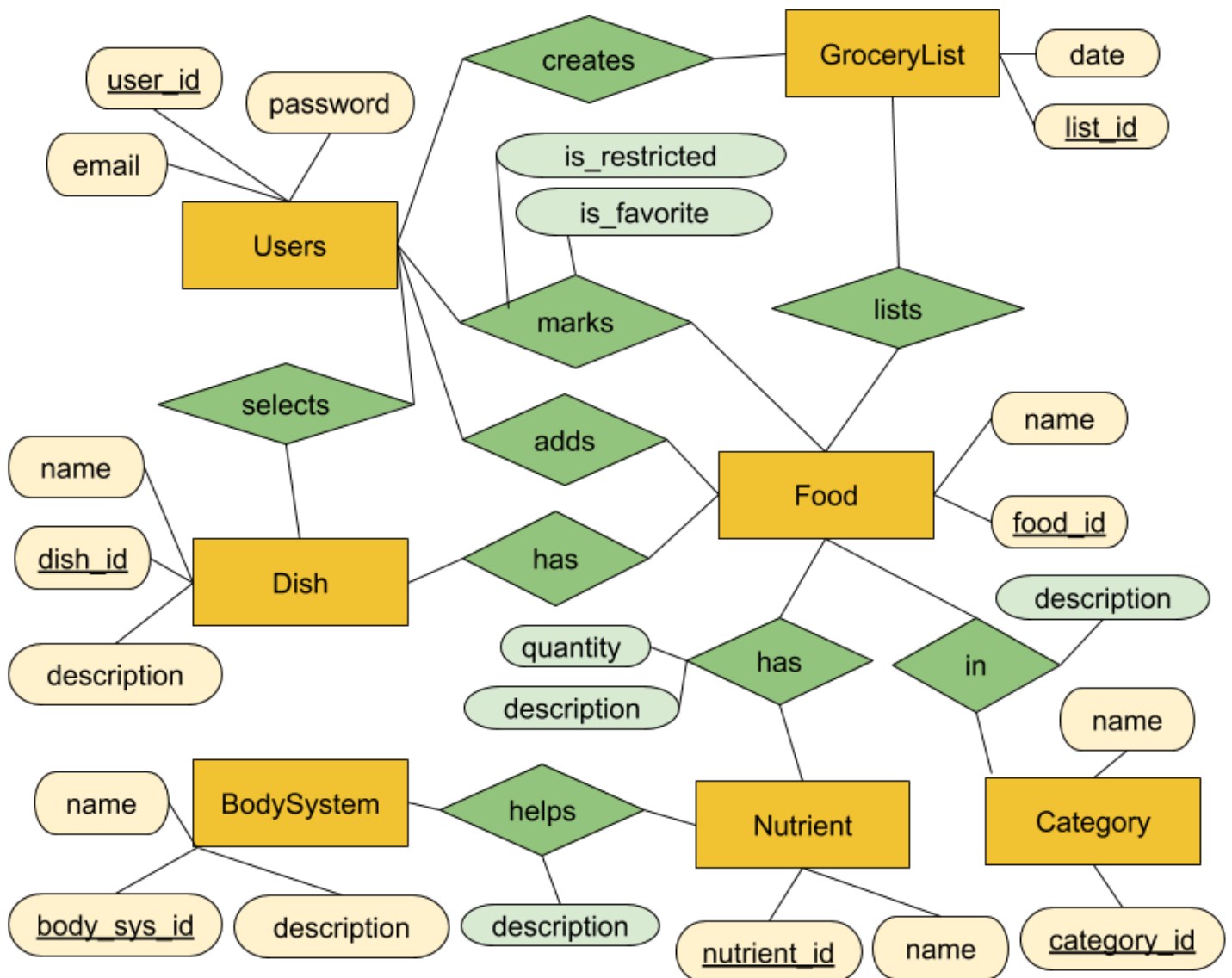
R6. Nutrient_helps_BodySystem: This relationship connects nutrients with body systems. It is to show which nutrients can benefit what body systems. A nutrient can benefit multiple body systems, and a body system can be benefited by multiple nutrients.

R7. Food_has_Nutrient: This relationship connects food with nutrients. A food item can have multiple nutrients and a nutrient can be found in many different food items. This will also be where the nutrient quantity for a food item will be stored.

R8. Food_lists_GroceryList: This relationship connects food with the user's grocery list. From the selected food items to the grocery list, we may suggest users to add their favorite food or give warning on the restricted food item if it's selected.

R9. Food_in_Category: This relationship connects a food to a category. Each food belongs to a category, which will be retrieved later in the implementation stage for providing users with browsing food by category option.

Entity-Relationship Diagram



Schemas for Entity Sets

- E1. Users(user_id, password, email)
- E2. Dish(dish_id, name, description)
- E3. Food(food_id, name)
- E4. BodySystem(body_sys_id, name, description)
- E5. GroceryList(list_id, date)
- E6. Nutrient(nutrient_id, name)
- E7. Category(category_id, name)

Schemas for Relationships

- R1. User_creates_GroceryList(user_id, list_id)
- R2. User_marks_Food(user_id, food_id, is_restricted, is_favorite)
- R3. User_adds_Food(user_id, food_id)
- R4. User_selects_Dish(user_id, dish_id)
- R5. Dish_has_Food(dish_id, food_id)
- R6. Nutrient_helps_BodySystem(nutrient_id, body_sys_id, description)
- R7. Food_has_Nutrient(food_id, nutrient_id, quantity, description)
- R8. Food_lists_GroceryList(food_id, list_id)
- R9. Food_in_Category(food_id, category_id)

MySQL Workbench Tables

E1. Users(user_id, password, email)

user_id	email	password
1	testuser1@example.com	passtest123
2	testuser2@example.com	pass90
3	testuser3@example.com	sqlop90
4	testuser4@example.com	f4fopi5
5	testuser5@example.com	fiver
6	testuser6@example.com	qrghbdsed52
7	testuser7@example.com	septem
8	testuser8@example.com	passtest123
9	testuser9@example.com	sjsucs157a
10	testuser10@example.com	encrypted
11	testuser11@example.com	password
12	testuser12@example.com	defend
13	testuser13@example.com	the
14	testuser14@example.com	east
15	testuser15@example.com	wall!

E2. Dish(dish_id, name, description)

dish_id	name	description
1	Basic Salad	Assorted vegetables tossed together
2	Hamburger	Basic American Burger
3	Stir-fry Veggies	Assorted Grilled Vegetables
4	Mashed Califlower	Califlower mashed together
5	Vegetable Juice	Blended Vegetable Juice
6	Pasta	Basic Pasta Dish
7	Roasted Chicken	Baked Chicken
8	Ham & Eggs	Ham with Eggs
9	Carrot Juice	Blended Carrot Juice
10	Grape Juice	Blended Grape Juice
11	Vegetable Soup	Assorted veggie soup
12	Cherry Pie	Baked pie with cherries
13	Lemonade	Classic Lemon Drink
14	Steamed Asparagus	Steamed up asparagus
15	Beef & Broccoli	Cooked beef and broccoli

E3. Food(food_id, name)

food_id	name
30001	Asparagus
30002	Broccoli
30003	Carrots
30004	Cauliflower
30005	Celery
30006	Cherries
30007	Grapefruit
30008	Grapes
30009	Kiwis
30010	Lemons / Limes
30011	Bacon / Sausage
30012	Beef
30013	Chicken
30014	Ground beef / Turkey
30015	Ham / Pork

E4. BodySystem(body_sys_id, name, description)

body_sys_id	name
401	respiratory system
402	digestive system
403	cardiovascular system
404	urinary system
405	blood cells
406	skeletal system
407	muscular system
408	endocrine system
409	lymphatic system
410	nervous system
411	reproductive system
412	immune system
413	eyes
414	skin/hair/nails
415	heart
NULL	NULL

Table: BodySystem

Columns:

Column Name	Data Type
<u>body_sys_id</u>	int PK
name	varchar(45)

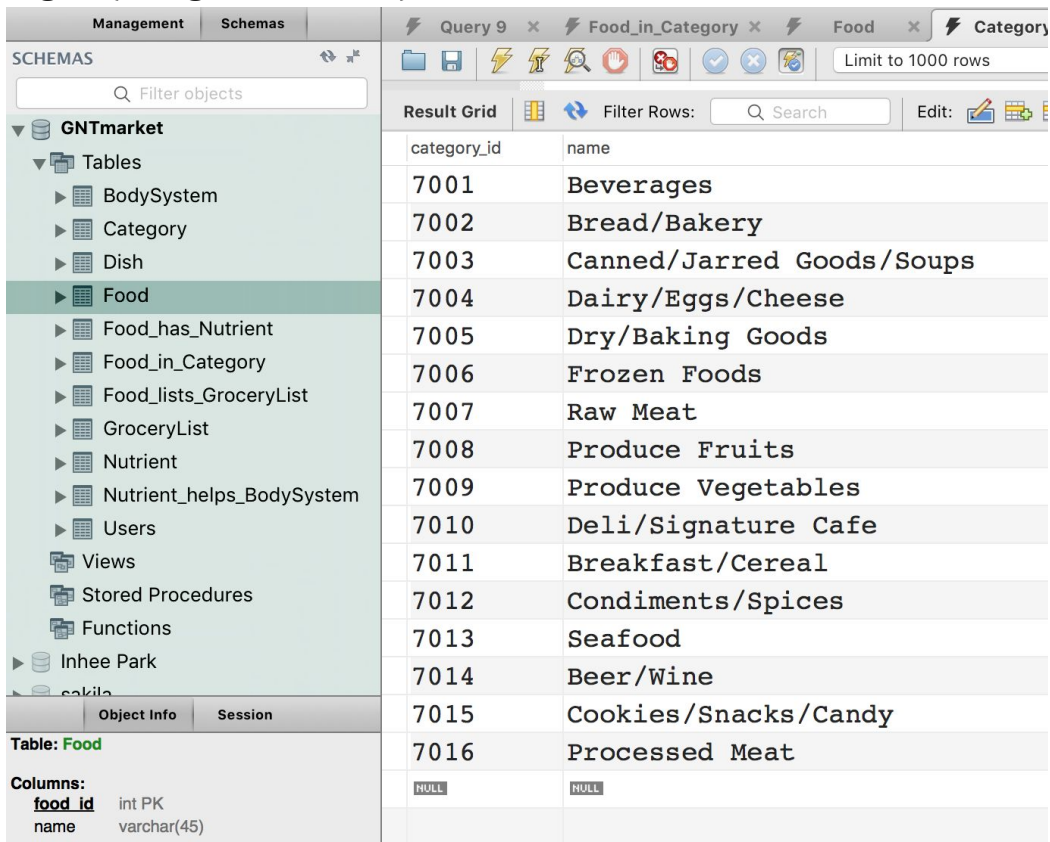
E5. GroceryList(list_id, date)

list_id	date
1	2020-07-05 16:12:36
2	2020-07-05 16:12:41
3	2020-07-05 16:12:45
4	2020-07-05 16:12:48
5	2020-07-05 16:12:53
6	2020-07-05 16:12:56
7	2020-07-05 16:12:59
8	2020-07-05 16:13:02
9	2020-07-05 16:13:09
10	2020-07-05 16:13:11
11	2020-07-05 16:13:14
12	2020-07-05 16:13:17
13	2020-07-05 16:13:20
14	2020-07-05 16:13:25
15	2020-07-05 16:13:31

E6. Nutrient(nutrient_id, name)

Management Schemas		Query 28 x	Nutrient x
SCHEMAS		Limit to 1000 rows	
Filter objects		Filter Rows: Search	
GNTmarket		Result Grid	
Tables		Edit: [Icons]	
BodySystem		nutrient_id name	
Category		6001 Carbohydrate	
Dish		6002 Polyunsaturated fat	
Food		6003 Protein	
Food_has_Nutrient		6004 Saturated Fat	
Food_in_Category		6005 Total Dietary Fiber	
Food_lists_GroceryList		6006 Calcium	
GroceryList		6007 Fluoride	
Nutrient		6008 Iron	
Nutrient_helps_BodySystem		6009 Magnesium	
Users		6010 Manganese	
Views		6011 Phosphorus	
Stored Procedures		6012 Potassium	
Functions		6013 Selenium	
Inhee Park		6014 Sodium	
sakila		6015 Zinc	
Object Info Session		6016 Caffeine	
Table: Nutrient		6017 Cholesterol	
Columns:		6018 Omega-3 Fatty Acid	
nutrient_id int PK		6019 Vitamin C	
name varchar(45)		6020 Vitamin B	
		6021 Vitamin D	
		6023 Vitamin E	
		6024 Water	
		6025 Vitamin A	

E7. Category(category_id, name)



Management Schemas Query 9 x Food_in_Category x Food x Category

Limit to 1000 rows

Result Grid Filter Rows: Search Edit:

category_id	name
7001	Beverages
7002	Bread/Bakery
7003	Canned/Jarred Goods/Soups
7004	Dairy/Eggs/Cheese
7005	Dry/Baking Goods
7006	Frozen Foods
7007	Raw Meat
7008	Produce Fruits
7009	Produce Vegetables
7010	Deli/Signature Cafe
7011	Breakfast/Cereal
7012	Condiments/Spices
7013	Seafood
7014	Beer/Wine
7015	Cookies/Snacks/Candy
7016	Processed Meat
NULL	NULL

Table: Food

Columns:

- food_id int PK
- name varchar(45)

R1. User_creates_GroceryList(user_id, list_id)

user_id	list_id
1	1
2	2
3	3
4	4
5	5
6	6
7	15
8	14
9	13
10	12
11	11
12	10
13	9
14	8
15	7

R2. User_marks_Food(user_id, food_id, is_restricted, is_favorite)

user_id	food_id	is_restricted	is_favorite
1	1	0	1
2	1	0	0
3	3	1	0
1	11	1	0
4	5	0	1
5	5	0	1
7	6	0	1
9	4	0	1
9	5	0	1
10	9	0	1
11	12	1	0
4	8	1	0
12	12	0	1
13	11	1	0
15	15	0	1

R3. User_adds_Food(user_id, food_id)

user_id	food_id
1	1
4	5
5	5
7	6
9	4
9	5
10	9
12	12
15	15
11	2
13	3
14	7
12	8
15	5
2	2

R4. User_selects_Dish(user_id, dish_id)

user_id	dish_id
1	1
1	2
2	2
3	5
6	4
4	4
8	3
9	11
14	12
12	10
13	13
11	9
13	10
14	14
15	15

R5. Dish_has_Food(dish_id, food_id)

dish_id	food_id
1	2
1	3
1	4
3	1
3	2
3	3
2	14
4	4
14	1
6	11
6	14
12	6
9	3
15	2
15	12

R6. Nutrient_helps_BodySystem(nutrient_id, body_sys_id, description)

Management Schemas		
Query 28 x Nutrient_helps_BodySystem x		
Limit to 1000 rows		
Result Grid Filter Rows: Search Edit: Export/Imp		
nutrient_id	body_sys_id	description
6004	414	protein helps skin/hair/nails
6005	402	fiber helps digestive system
6006	406	calcium helps skeletal system
6006	407	calcium helps muscle system
6006	414	calcium helps skin/hair/nails
6007	406	fluoride helps skeletal system
6008	405	iron helps blood cells
6012	401	potassium helps nervous system
6012	407	potassium helps muscle system
6015	412	zinc helps immune system
6019	412	vitamin C helps immune system
6020	412	vitamin D helps immune system
6021	412	vitamin E helps immune system
6024	415	water helps circulatory system
6025	413	vitamin A helps eyes
NULL	NULL	NULL

Object Info Session		
Table: Nutrient_helps_BodySystem		
Columns:		
nutrient_id	int PK	
body_sys_id	int PK	
description	varchar(45)	

R7. Food_has_Nutrient(food_id, nutrient_id, quantity, description)

The screenshot shows a database management interface with a left sidebar containing a schema tree. The 'Food_has_Nutrient' table is selected. The main area displays the table's structure and data.

Table: Food_has_Nutrient

Columns:

- food_id int PK
- nutrient_id int PK
- quantity decimal(5,2)
- description varchar(45)

Result Grid:

food_id	nutrient_id	quantity	description
30001	6001	2.60	asparagus has 2.6g carbohydrates
30001	6003	1.50	asparagus has 1.5g protein
30001	6004	0.10	asparagus has 1.5g fat
30001	6005	1.40	asparagus has 1.4g fiber
30001	6019	0.00	asparagus has 0.38mg vitamin C
30006	6001	19.00	cherries have 19g carbohydrates
30006	6003	1.60	cherries have 1.6g protein
30006	6005	2.50	cherries have 2.5g fiber
30006	6012	0.27	cherries have 268mg potassium
30009	6001	10.00	kiwi has 10g carbohydrates
30009	6004	0.40	kiwi has 0.4g fat
30009	6012	0.22	kiwi has 215mg potassium
30011	6003	2.90	bacon has 2.9g protein
30011	6004	3.50	bacon has 3.5g fat
30011	6012	0.04	bacon has 44mg potassium
30011	6014	0.18	bacon has 178mg sodium
30011	6017	0.01	bacon has 9mg cholesterol
NULL	NULL	NULL	NULL

R8. Food_lists_GroceryList(food_id, list_id)

The screenshot shows a database management interface with a left sidebar containing a schema tree. The 'Food_lists_GroceryList' table is selected. The main area displays the table's structure and data.

Table: Food_lists_GroceryList

Columns:

- food_id int PK
- list_id int PK

Result Grid:

food_id	list_id
30001	12
30002	12
30003	12
30009	12
30013	12
30011	12
30001	13
30008	13
30012	13
30004	13
30003	14
30005	14
30007	14
30010	14
30011	14
30015	14
NULL	NULL

Proposal (6/15/2020 submitted)

Summer 2020 | CS 157A

GNT-Market

Grocery Nutrient Tracker, Calorie Calculator

Team 11

Tracy Ho & Inhee Park

Project Description

- **Goal & Motivation :**

With the on-going unprecedented pandemic situation, one essential activity that still needs to be done is grocery shopping. It is often advised to be a quick process with a healthy selection due to the need of minimizing time spent inside a store, along with the need to have proper nutrients from a well-balanced diet given how sedentary people's lives have become from staying at home. To help people's dietary needs, we propose the "GNT-Market" web app as a **Grocery Nutrient Tracker** as well as a Calorie Calculator. This can assist people track their grocery buying along with food consumption to have a better view of their diet.

- **Stakeholders :** consumers

- **Application Domain :** health, food, nutrient

- **Benefits to Users :** Quick and healthy tracker for users' grocery shopping list

System Environment

- **Structure of the system (3-tiered architecture) :**

Layer/Tier	Front-end	Middle-ware	Back-end
Role	Web Client	Web Server	DB Server
Software	Bootstrap [1-2]	TomCat [3-4]	MySQL [5]
Application Language	HTML, CSS, JS	Java, JSP	SQL

- **HW/SW used :** macOS Sierra 10.12.6

- **RDBMS used :** MySQL Ver 8.0.19; MySQL Workbench 6.0.10

- **Application languages :** Java, SQL, JSP, HTML, CSS, JS

Functional Requirements (DB manipulation activities)

A list of detailed descriptions of users and how users interact with your application

- **User Interaction with the App:**

There are 9 features to “GNT-Market”, which will be incorporated in a web application. Most of the data source will come from FoodData Central (FDC) of the U.S. DEPARTMENT OF AGRICULTURE. [6]

1. **[Dish ⇒ Grocery List ⇒ Nutrient/Calorie Information]**

User inputs a desired dish from a menu in the app.

The app will provide a list of grocery items with nutritional distribution (with exceeding/deficient nutrition) and the calorie information of each item.

2. **[Grocery List ⇒ Nutrient/Calorie Information]**

User inputs a list of grocery items in the app.

Then the app outputs cumulative nutritional distribution (with exceeding/deficient nutrition) and the calorie information of each item.

3. **[Grocery Browser]**

Users can browse food categories corresponding to section names of grocery aisles.

Which users can then add food items into their grocery list.

4. **[Search By Branded Grocery Item / Food]**

Users can type in grocery items to search for and add to their grocery list. Both branded items and general food name are accepted.

5. **[Dietary Restriction]**

Users can set their dietary restrictions (e.g. food allergy) to the app.

Then if the user selects a grocery item that conflicts with their diet restriction, the app gives a warning.

6. **[Browse Beneficial Food]**

Users can browse a paired list of food and their respective beneficiary body systems, so they may tailor food items for their specific health needs.

7. [User Creation]

A user will be able to register their own account to use the web application with an email and password.

Then they will be able to use the application for their own usage.

8. [Nutrient Suggestions]

A user who may want to get more of a particular nutrient in their diet may look into suggestions of what is rich in those nutrients, so they can include it in their list of groceries to buy.

This will display a list of food items that contain a specific nutrient, giving the user an idea of what nutrients they are taking in their diet.

9. [Favorite Food Items]

Users are able to mark which items as favorites so they may be able to access the items easier, whether it is to look up more details of the items or to re-add the items in their grocery list.

Describe each individual function/feature, functional process and I/O.

○ Functions:

1. **Frequently Purchased Grocery Items:** Keeping track of the number of times an item has been purchased, and store those items as frequently purchased items.
2. **List of Dietary Restrictions:** Set dietary restrictions by adding specific food items to be avoided.
3. **List of Beneficial Food:** Selects a body system from a dropdown menu that a user is concerned about, then the app will suggest healthy foods that can benefit that body system.
4. **Create/Delete a Grocery Shopping List Card:** Users can create a card that will contain all of the grocery items that they want to purchase. Items can be added to this card, deleted, or updated accordingly.
5. **Duplicate Grocery List:** Users that wish to reuse a previously made grocery list may be able to copy what they previously created.
6. **Browse Grocery Categories,** such as:
 - ✓ Beverages – coffee/tea, juice, soda

- ✓ Bread/Bakery – sandwich loaves, dinner rolls, tortillas, bagels
- ✓ Canned/Jarred Goods – vegetables, spaghetti sauce, ketchup
- ✓ Dairy – cheeses, eggs, milk, yogurt, butter
- ✓ Dry/Baking Goods – cereals, flour, sugar, pasta, mixes
- ✓ Frozen Foods – waffles, vegetables, individual meals, ice cream
- ✓ Meat – lunch meat, poultry, beef, pork
- ✓ Produce – fruits, vegetables

7. **Calorie Tracking:** Provides different categories to keep track of calories. Users will use the food that they already have added to keep a count of their calories.

Non-functional issues

Detailed descriptions of Graphical User Interface...

The web application will use Bootstrap to incorporate the following GUI concept into the web interface. Bootstrap is a collection of templates in HTML/CSS/Javascript for most GUI components (Tables, Buttons, Progress Bars, Pagination Panels, Dropdowns, Forms, Inputs, Carousel, and etc.) including Grid System, Themes.

Below is a concept sketch of the “GNT-Market” app. The main component is a Grocery List, and each item in the list can be populated with 3 different methods (by search, browse, or dish menu). The results will show Nutrition and Calorie Information at the bottom. Users may add their dietary restriction, so that the app will warn them when they add an avoided item to their grocery list. Development will be based on this conceptual GUI draft.

GNT-Market					User Add (+)	
Grocery List			By Search Food <u>Type Food Name</u>	Beneficial Food List <u>Learn More</u>		
+	2% Milk	x		ok	toList	
+	Cream Cheese	x	By Browse <u>Fruit Veggie Meat ...</u>	ok	toList	
+	Oranges	x		ok	toList	
+	Peanut Butter (!)	x		Food Restriction List		
+		x		no	Peanut	
			By Dish Menu <u>Type Dish Name</u>	no		
				no		
Nutrition Info			Calorie			
50% Carb	30% Protein	10% Fat	1,000 kcal/mol			

Detailed descriptions of Security...

Security of users will be protected by an email and password to identify the user. The username and password will be securely stored in the database, which the password will be encrypted, so even the web administration will not be able to see the password. Users must use their own email and password to login to be able to use the application. If they forget their password, they can provide their email address for a password change request. The web application itself will use HTTPS to have encrypted connections.

Detailed descriptions of Access Control...

For access control, users will only be able to have access to their own data. They will not be able to retrieve nor modify data from other users. There will be checks in place to make sure that only the user sees their own data. Administrators will be able to have access to all data, including user data and backend data that is only available to admin-level users.

References

- [1] [Add Bootstrap to JSP Page](#)
- [2] [Build fast, responsive sites with Bootstrap](#)
- [3] [Apache Tomcat](#)
- [4] [Apache Tomcat Troubleshoot](#)
- [5] [MySQL Workbench](#)
- [6] [FoodData Central Download Data](#) (All Data Types: [April 2020 version 2* \(CSV – 85M\)](#))