

Foodfinder

Software Architecture Document

Team

Arul Kiruthika Raghupathi	40200225
Akshaya Barat Bushan	40220139

Table of Contents

1	Foodfinder - Application Overview	3
2	Class Diagram	4
3	Object Relational Mapping	5
4	Data Source Architecture and Implementation	6
5	Sequence Diagram	7
6	Database Design	8
7	API Details	11
8	Queries Used	12
9	Deployment	14
10	Source Code Components	15
11	Refactoring Strategies	16
12	Sample Execution	17
13	System Size	18
14	GitHub	18

Foodfinder

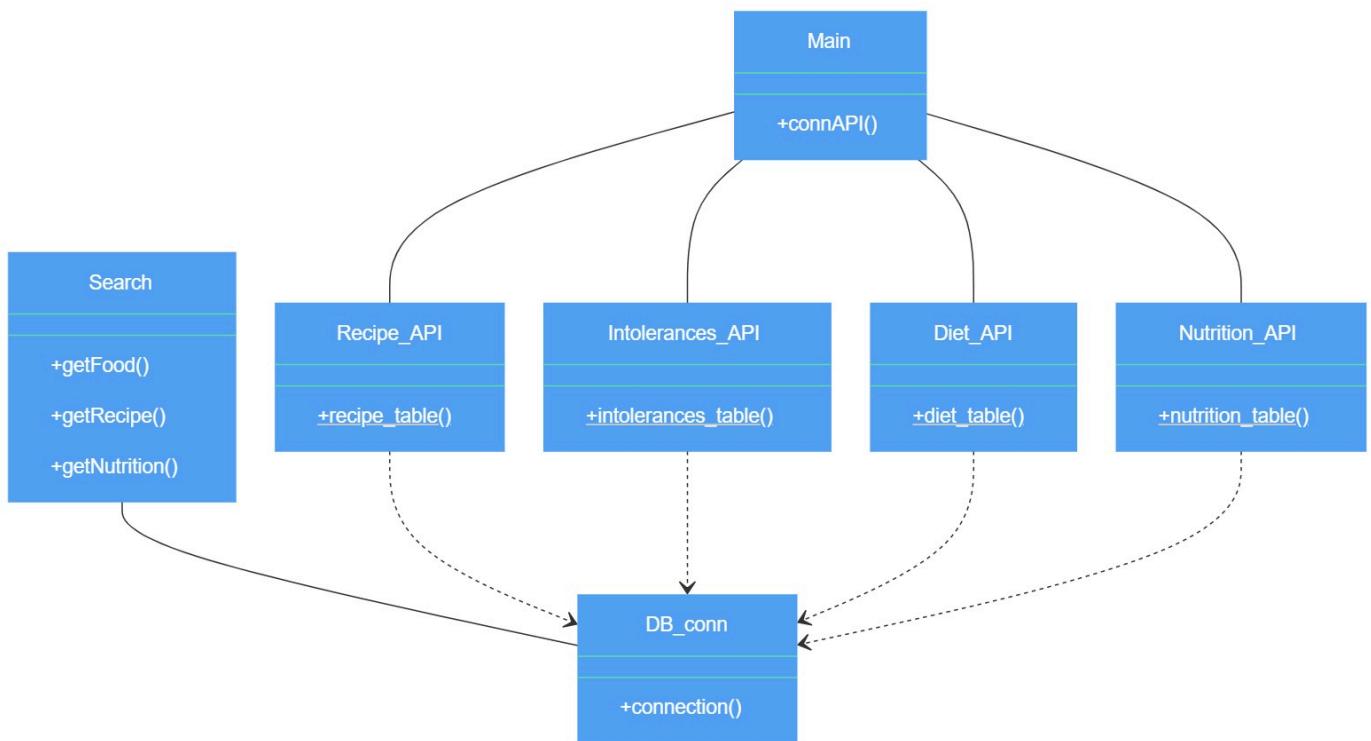
Application Overview

The Foodfinder is a web based application that allows you to find out the recipes based on your choices. The application filters recipes to check for dishes that contain common allergens, such as wheat, dairy, eggs, soy, nuts, etc., and also determine whether a recipe is vegan, vegetarian, Paleo friendly, Whole30 compliant, and more. It also fetches food with the nutritional information that satisfy users' dietary goals. If needed, the application provides the recipe card for the selected food.

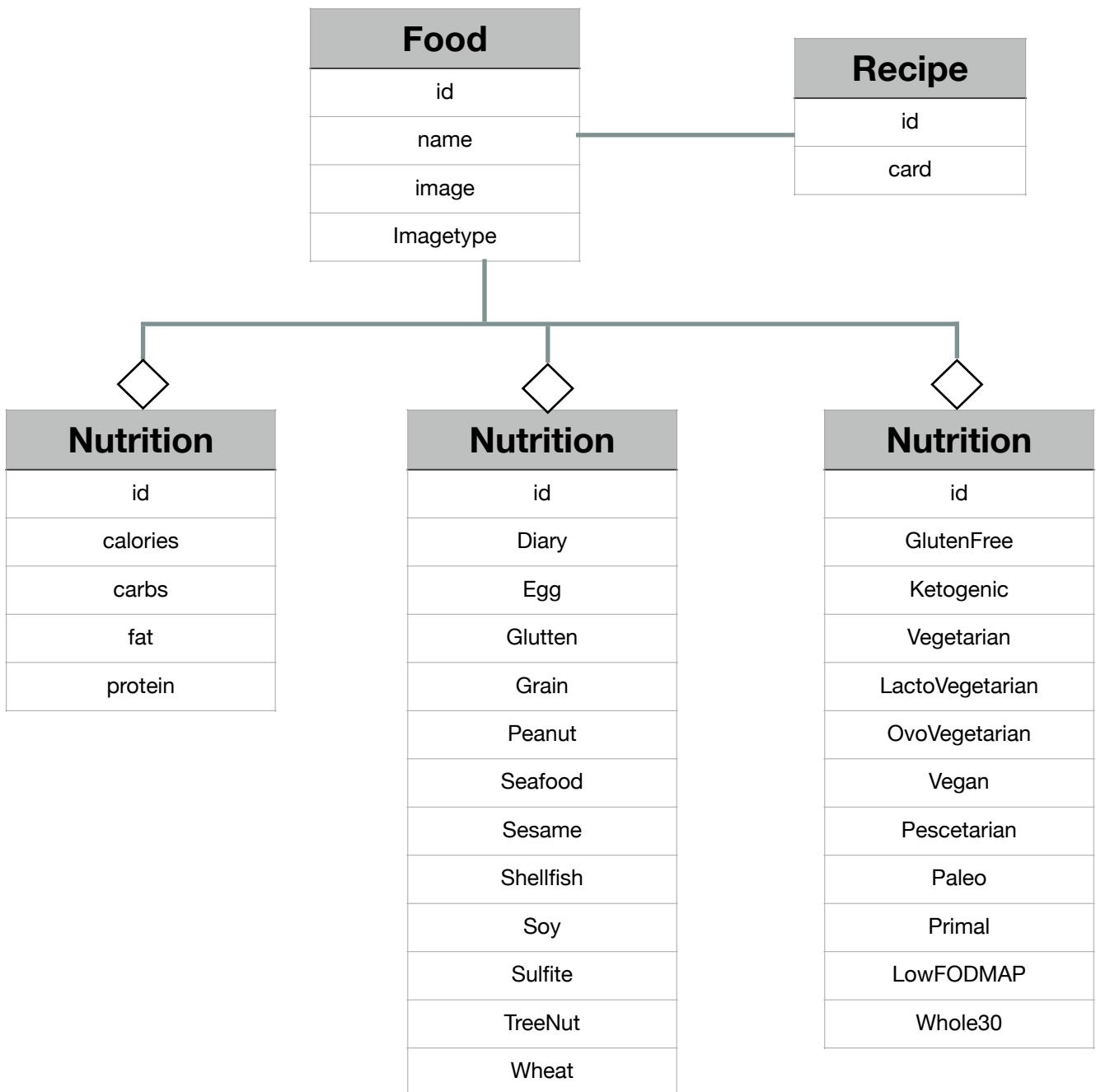
Why Foodfinder?

Tired of eating the same food when you have intolerances and dietary restrictions? Use Foodfinder to find yummy recipes that satisfy all your requirements!

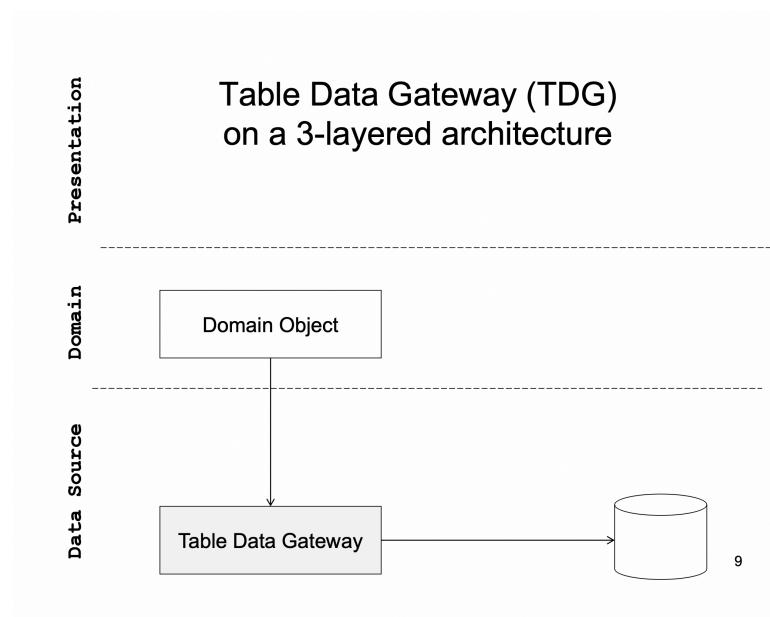
Class Diagram



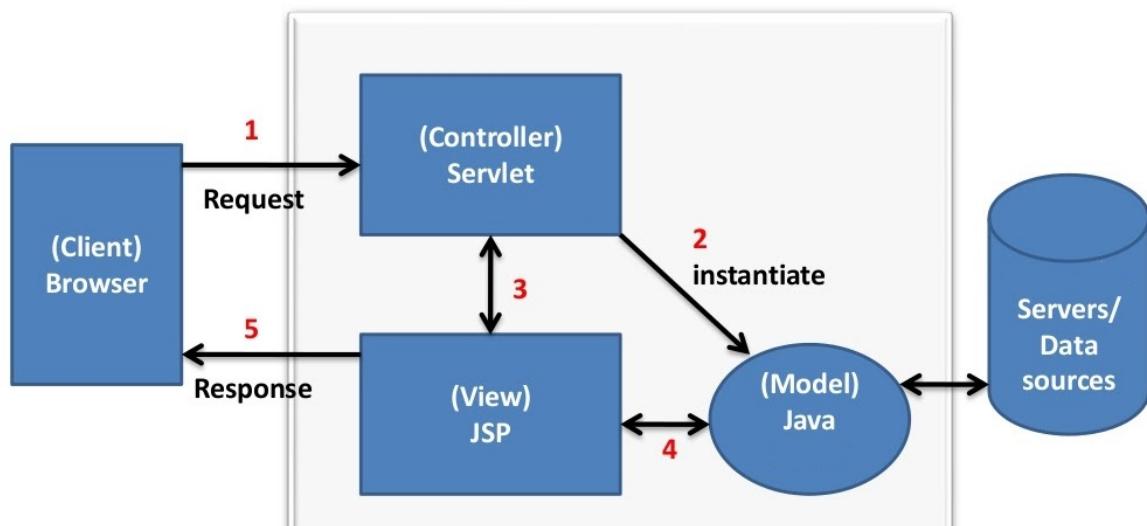
Object Relational Mapping



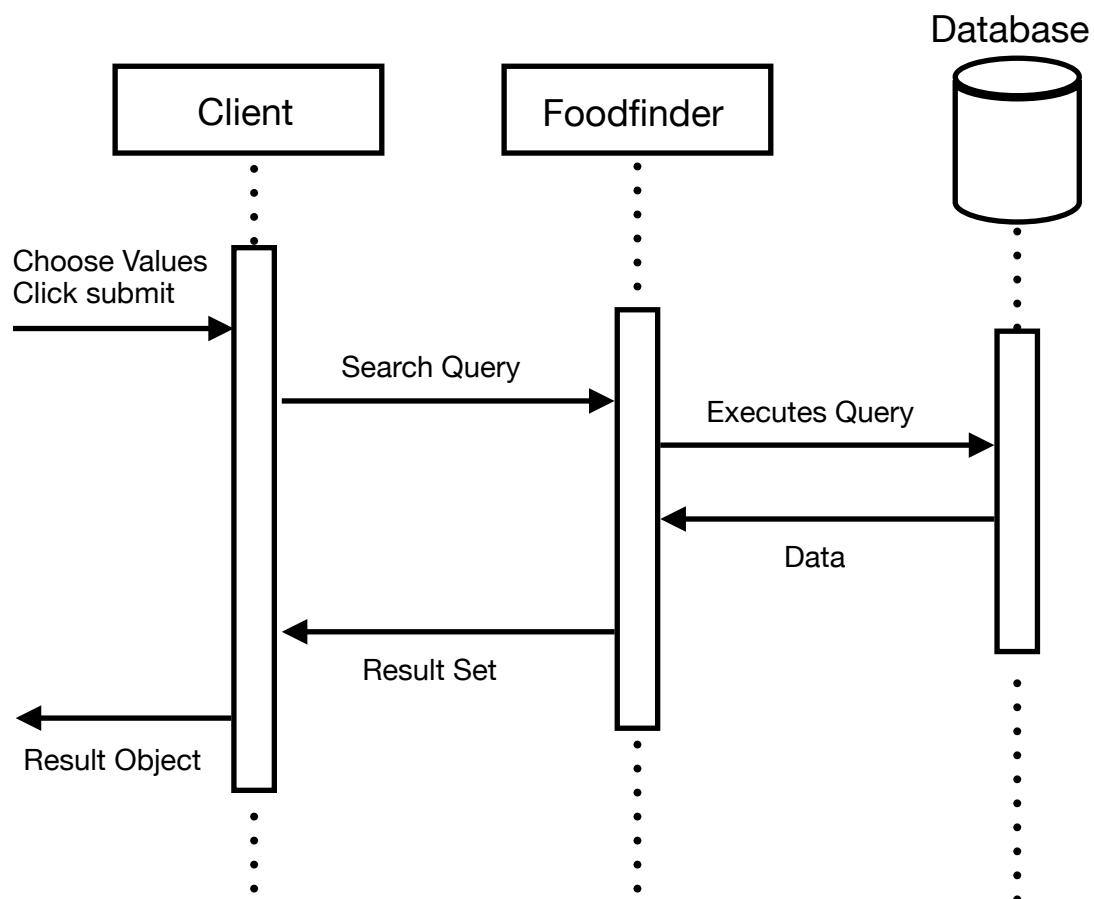
Data Source Architecture



Implementation



Sequence Diagram



Database Design

Database - sample

```
[mysql] > show tables;
+-----+
| Tables_in_sample |
+-----+
| diet
| Food
| intolerances
| nutrition
| recipe
+-----+
5 rows in set (0.00 sec)
```

Table - Food:

```
[mysql] > describe Food;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int       | NO   | PRI | NULL    |        |
| title | varchar(100) | NO   |     | NULL    |        |
| imagepath | varchar(300) | YES  |     | NULL    |        |
| imagetype | varchar(10)  | YES  |     | NULL    |        |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.08 sec)
```

Table - diet:

```
[mysql] > describe diet;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
GlutenFree	varchar(10)	YES		no	
Ketogenic	varchar(10)	YES		no	
Vegetarian	varchar(10)	YES		no	
LactoVegetarian	varchar(10)	YES		no	
OvoVegetarian	varchar(10)	YES		no	
Vegan	varchar(10)	YES		no	
Pescetarian	varchar(10)	YES		no	
Paleo	varchar(10)	YES		no	
Primal	varchar(10)	YES		no	
LowFODMAP	varchar(10)	YES		no	
Whole30	varchar(10)	YES		no	

12 rows in set (0.01 sec)

Table - nutrition:

```
[mysql] > describe nutrition;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
calories	int	YES		NULL	
carbs	varchar(10)	YES		NULL	
fat	varchar(10)	YES		NULL	
protein	varchar(10)	YES		NULL	

5 rows in set (0.00 sec)

Table - intolerances:

```
[mysql] > describe intolerances;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
Diary	varchar(10)	YES		no	
Egg	varchar(10)	YES		no	
Glutten	varchar(10)	YES		no	
Grain	varchar(10)	YES		no	
Peanut	varchar(10)	YES		no	
Seafood	varchar(10)	YES		no	
Sesame	varchar(10)	YES		no	
Shellfish	varchar(10)	YES		no	
Soy	varchar(10)	YES		no	
Sulfite	varchar(10)	YES		no	
TreeNut	varchar(10)	YES		no	
Wheat	varchar(10)	YES		no	

```
13 rows in set (0.01 sec)
```

Table - recipe:

```
[mysql] > describe recipe;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
card	varchar(300)	YES		NULL	

```
2 rows in set (0.00 sec)
```

API Details

Diet API:

```
value = {"GlutenFree", "Ketogenic", "Vegetarian", "LactoVegetarian", "OvoVegetarian",  
"Vegan", "Pescetarian", "Paleo", "Primal", "LowFODMAP", "Whole30"}
```

```
https://api.spoonacular.com/recipes/complexSearch?  
diet=value&apiKey=your\_key
```

Intolerances API:

```
value = {"Diary", "Egg", "Gluten", "Grain", "Peanut", "Seafood", "Sesame", "Shellfish",  
"Soy", "Sulfite", "TreeNut", "Wheat"}
```

```
https://api.spoonacular.com/recipes/complexSearch?  
intolerances=value&apiKey=your\_key
```

Nutrition API:

Id refer to the food id.

```
https://api.spoonacular.com/recipes/  
id/nutritionWidget.json?  
apiKey=your\_key
```

Recipe API:

Id refer to the food id.

```
https://api.spoonacular.com/recipes/  
id/card?  
backgroundImage=none&apiKey=your\_key
```

Queries Used

API Queries:

Table - Food:

```
create table Food(id int NOT NULL PRIMARY KEY, title varchar(100) NOT NULL,  
imagepath varchar(300), imagetype varchar(10));
```

```
insert into food(id, title, imagepath, imagetype) values( 1095693, "Raspberry Arugula  
Side Salad", "https://spoonacular.com/recipeImages/1095693-312x231.jpg", "jpg");
```

```
select id from food;
```

Table - diet:

```
create table diet(id int NOT NULL PRIMARY KEY, GlutenFree varchar(10) DEFAULT  
'no', Ketogenic varchar(10) DEFAULT 'no', Vegetarian varchar(10) DEFAULT 'no',  
LactoVegetarian varchar(10) DEFAULT 'no', OvoVegetarian varchar(10) DEFAULT 'no',  
Vegan varchar(10) DEFAULT 'no', Pescetarian varchar(10) DEFAULT 'no', Paleo  
varchar(10) DEFAULT 'no', Primal varchar(10) DEFAULT 'no', LowFODMAP varchar(10)  
DEFAULT 'no', Whole30 varchar(10) DEFAULT 'no');
```

```
insert into diet(id, Vegan) values(644387, "yes");
```

```
select * from diet where id=644387;
```

```
update diet set Vegetarian="yes" where id=644387;
```

Table - nutrition:

```
create table nutrition(id int NOT NULL PRIMARY KEY, calories int, carbs varchar(10), fat  
varchar(10), protein varchar(10));
```

```
select * from nutrition where id=644387;
```

```
insert into nutrition(id,calories,carbs,fat,protein) values(1095693,142,8g,11g,2g);
```

Table - intolerances:

```
create table intolerances(id int NOT NULL PRIMARY KEY, Diary varchar(10) DEFAULT 'no', Egg varchar(10) DEFAULT 'no', Gluten varchar(10) DEFAULT 'no', Grain varchar(10) DEFAULT 'no', Peanut varchar(10) DEFAULT 'no', Seafood varchar(10) DEFAULT 'no', Sesame varchar(10) DEFAULT 'no', Shellfish varchar(10) DEFAULT 'no', Soy varchar(10) DEFAULT 'no', Sulphite varchar(10) DEFAULT 'no', TreeNut varchar(10) DEFAULT 'no', Wheat varchar(10) DEFAULT 'no');
```

```
insert into intolerances(id, Peanut) values(644387, "yes");
```

```
select * from intolerances where id=644387;
```

```
update intolerances set Soy="yes" where id=644387;
```

Table - recipe:

```
create table recipe(id int NOT NULL PRIMARY KEY, card varchar(300));
```

```
insert into recipe(id, card) values(644387, "https://spoonacular.com/recipeCardImages/recipeCard-1668069421983.png");
```

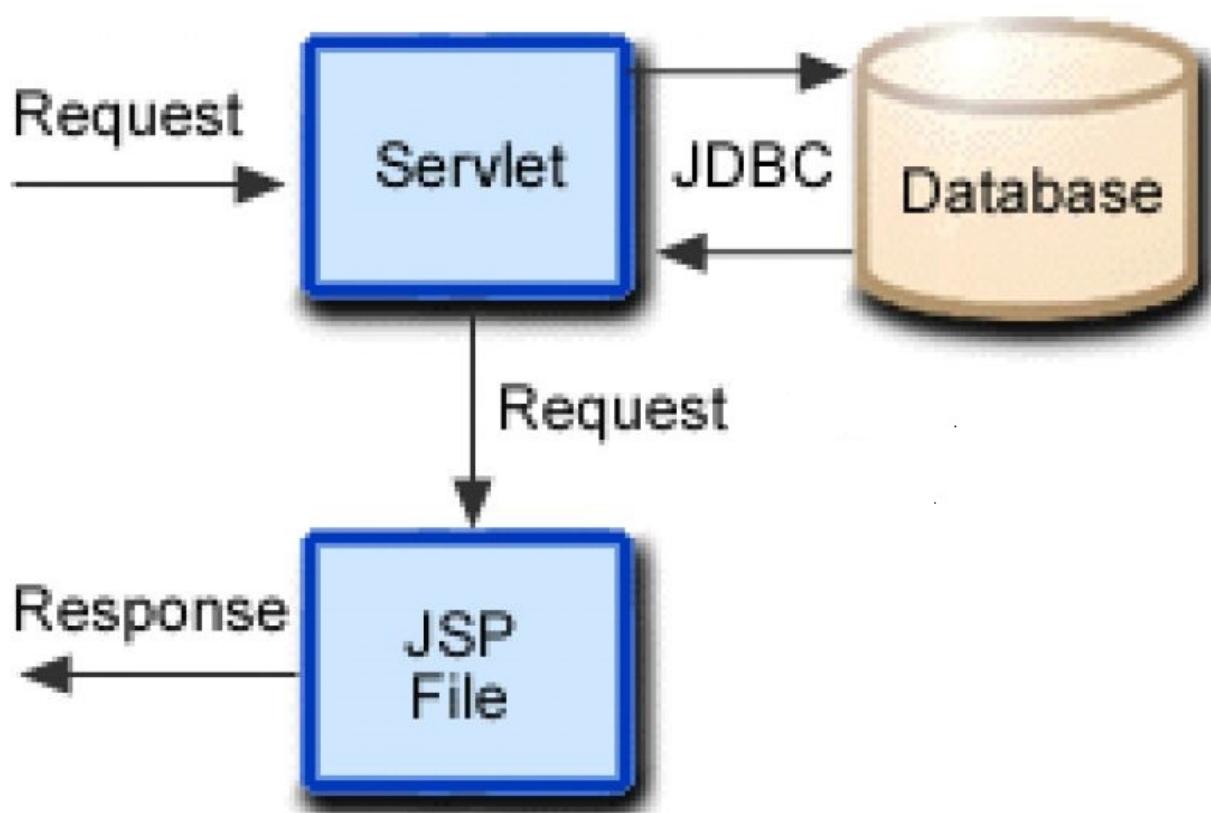
Search Queries:

```
select * from food where id in(select id from nutrition where id in (select d.id from intolerances as i inner join diet as d where Primal="yes" and Egg="yes") and calories>0 and calories<=250);
```

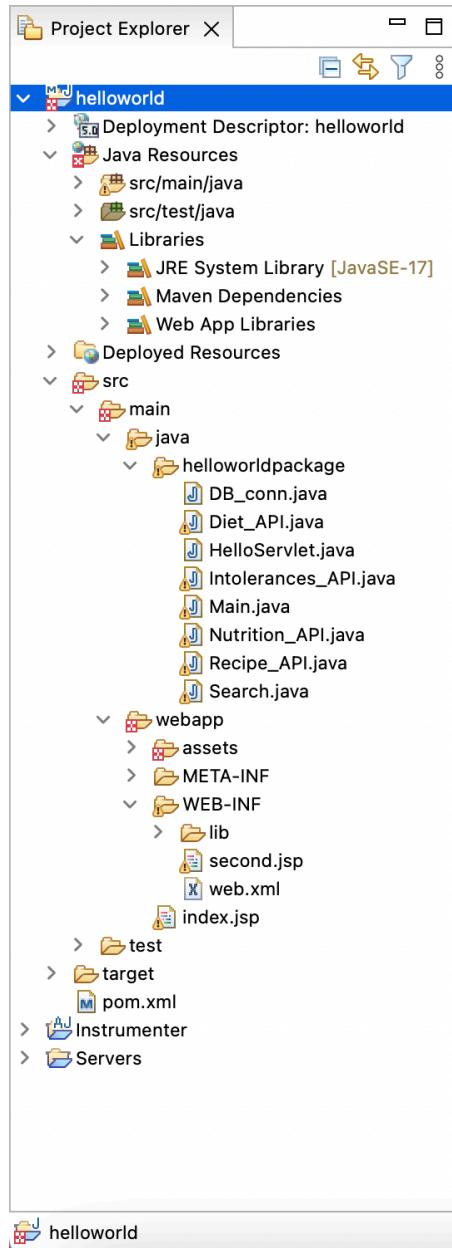
```
select card from recipe where id in (select id from food where id in (select id from nutrition where id in (select d.id from intolerances as i inner join diet as d where Primal="yes" and Egg="yes") and calories>0 and calories<=250)));
```

```
select calories,carbs,fat,protein from nutrition where id in (select d.id from intolerances as i inner join diet as d where Primal="yes" and Egg="yes") and calories>0 and calories<=250;
```

Deployment



Source Code Components



Refactoring Strategies

1. **Extract Methods:** A code fragment inside a method performs a cohesive task. Encapsulate the fragment into its own method. When the fragment modifies a local variable: If the variable is used only within the extracted code, then move it into the extracted code. If the variable is used outside (after) the extracted code we need to make the extracted code return the new value of the variable.
2. **Inline Temp:** A temporary variable is assigned the result of a simple expression. Replace the references to the variable with the expression itself.
3. **Split temporary variable:** A local variable is used to store various intermediate values. Use different variables for different values. Each variable should be responsible for only one thing.
4. **Extract Class:** We have a class that does two unrelated tasks. Create a new class, thus separating the tasks into two classes.
5. **Change Bidirectional Association to Unidirectional:** We have bidirectional association between two classes, but only one needs to maintain visibility over the other. Remove the unnecessary association.

Sample Execution

FIND OUT THE TASTY RECIPES!



Under 250 kcalories

Primal

Egg

Find yummy Recipes!

Output:

Food	Calories	Carbs	Fat	Protein
Crunchy Brussels Sprouts Side Dish	231	19g	15g	7g
Garlicky Kale	178	10g	14g	3g
Vegetable Dip	152	20g	3g	12g
Summer Berry Salad	97	22g	1g	2g
Homemade Guacamole	170	10g	14g	2g
Chilled Cucumber Avocado Soup with Yogurt and Kefir	188	24g	7g	8g

System Size

The Foodfinder application's size is described with the following items:

- Dependencies on external components: 5
- Lines of Java code: approximately 1000
- Java source files: 8
- JSP files: 2

GITHUB

Public Repository

Link: [***https://github.com/Arul210/SOEN-6441-Project***](https://github.com/Arul210/SOEN-6441-Project)