

**FOODWHALE**

**Software Design Specification**

– Hanoi, August 2019 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 1/12/2021 | A | Ha | SWP391-AppDevProject\_Desgin |
| 1/12/2021 | M | Ha | Overview |
| 1/12/2021 | A | Ha | View list user, edit Account, add Account |
| 1/12/2021 | A | HoangAnh | User Profile |
| 1/12/2021 | A | Quoc | Code Design : Login and Register |
| 3/12/2021 | A | Quoc | Code Design : Reset Password |
| 3/12/2021 | A | Minh | View list recipe, edit recipe, add recipe |
| 3/12/2021 | A | Nghia | View Homepage, Recipe, Ingredient |
| 3/12/2021 | A | Ha | View list order, Order detail |
| 8/12/2021 | A | Quoc | Code Design: Post Recipe, View liked Recipe |
| 8/12/2021 | A | Hoanganh | Add to Cart |
| 8/12/2021 | A | Ha | Ingredient, Category list and detail |
| 13/12/2021 | M | Nghia | Change Add to Cart to Cart Detail |
| 13/12/2021 | A | Nghia | View Recipe Detail, View Ingredient Detail |
| 13/12/2021 | A | Ha | Recipe Ingredient Link |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) 6

[1. Code Packages](#_heading=h.30j0zll) 6

[2. Database Design](#_heading=h.1fob9te) 8

[a. Database Schema](#_heading=h.3znysh7) 8

[b. Table Description](#_heading=h.2et92p0) 8

[II. Code Designs](#_heading=h.tyjcwt) 10

1[. Homepage](#_heading=h.3dy6vkm) 10

[a. Class Diagram](#_heading=h.1t3h5sf) 10

[b. Class Specifications](#_heading=h.4d34og8) 10

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 11

[d. Database queries](#_heading=h.17dp8vu) 12

2[. Login](#_heading=h.3dy6vkm) 13

[a. Class Diagram](#_heading=h.1t3h5sf) 13

[b. Class Specifications](#_heading=h.4d34og8) 13

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 14

[d. Database queries](#_heading=h.17dp8vu) 14

3[. User Profile](#_heading=h.3dy6vkm) 14

[a. Class Diagram](#_heading=h.1t3h5sf) 14

[b. Class Specifications](#_heading=h.4d34og8) 15

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 16

[d. Database queries](#_heading=h.17dp8vu) 17

4[. View list user](#_heading=h.3dy6vkm) 17

[a. Class Diagram](#_heading=h.1t3h5sf) 17

[b. Class Specifications](#_heading=h.4d34og8) 18

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 18

[d. Database queries](#_heading=h.17dp8vu) 19

5[. Register](#_heading=h.3dy6vkm) 19

[a. Class Diagram](#_heading=h.1t3h5sf) 19

[b. Class Specifications](#_heading=h.4d34og8) 20

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 20

[d. Database queries](#_heading=h.17dp8vu) 21

6[. Account Detail](#_heading=h.3dy6vkm) 21

[a. Class Diagram](#_heading=h.1t3h5sf) 21

[b. Class Specifications](#_heading=h.4d34og8) 21

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 22

[d. Database queries](#_heading=h.17dp8vu) 24

7[. Reset Password 25](#_heading=h.3dy6vkm)

[a. Class Diagram 25](#_heading=h.1t3h5sf)

[b. Class Specifications 25](#_heading=h.4d34og8)

[c. Sequence Diagram(s) 26](#_heading=h.2s8eyo1)

[d. Database queries 26](#_heading=h.17dp8vu)

8[. View Recipe](#_heading=h.3dy6vkm) 27

[a. Class Diagram](#_heading=h.1t3h5sf) 27

[b. Class Specifications](#_heading=h.4d34og8) 27

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 28

[d. Database queries](#_heading=h.17dp8vu) 28

9[. View Ingredient](#_heading=h.3dy6vkm) 29

[a. Class Diagram](#_heading=h.1t3h5sf) 29

[b. Class Specifications](#_heading=h.4d34og8) 29

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 30

[d. Database queries](#_heading=h.17dp8vu) 30

10[. View List Order](#_heading=h.3dy6vkm) 30

[a. Class Diagram](#_heading=h.1t3h5sf) 30

[b. Class Specifications](#_heading=h.4d34og8) 31

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 31

[d. Database queries](#_heading=h.17dp8vu) 31

11[. Order detail](#_heading=h.3dy6vkm) 32

[a. Class Diagram](#_heading=h.1t3h5sf) 32

[b. Class Specifications](#_heading=h.4d34og8) 32

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 32

[d. Database queries](#_heading=h.17dp8vu) 34

12[. View List Recipe](#_heading=h.3dy6vkm) 35

[a. Class Diagram](#_heading=h.1t3h5sf) 35

[b. Class Specifications](#_heading=h.4d34og8) 35

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 36

[d. Database queries](#_heading=h.17dp8vu) 36

13[. Post Recipe](#_heading=h.3dy6vkm) 37

[a. Class Diagram](#_heading=h.1t3h5sf) 37

[b. Class Specifications](#_heading=h.4d34og8) 37

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 38

[d. Database queries](#_heading=h.17dp8vu) 38

14[. View Liked Recipe](#_heading=h.3dy6vkm) 39

[a. Class Diagram](#_heading=h.1t3h5sf) 39

[b. Class Specifications](#_heading=h.4d34og8) 39

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 40

[d. Database queries](#_heading=h.17dp8vu) 40

15[. View List Ingredient](#_heading=h.3dy6vkm) 41

[a. Class Diagram](#_heading=h.1t3h5sf) 41

[b. Class Specifications](#_heading=h.4d34og8) 41

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 41

[d. Database queries](#_heading=h.17dp8vu) 42

16[. Ingredient detail](#_heading=h.3dy6vkm) 42

[a. Class Diagram](#_heading=h.1t3h5sf) 42

[b. Class Specifications](#_heading=h.4d34og8) 42

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 43

[d. Database queries](#_heading=h.17dp8vu) 45

17[. View List Category](#_heading=h.3dy6vkm) 46

[a. Class Diagram](#_heading=h.1t3h5sf) 46

[b. Class Specifications](#_heading=h.4d34og8) 46

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 46

[d. Database queries](#_heading=h.17dp8vu) 47

18[. Category detail](#_heading=h.3dy6vkm) 47

[a. Class Diagram](#_heading=h.1t3h5sf) 47

[b. Class Specifications](#_heading=h.4d34og8) 47

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 48

[d. Database queries](#_heading=h.17dp8vu) 50

19[. Cart Detail 5](#_heading=h.3dy6vkm)1

[a. Class Diagram 5](#_heading=h.1t3h5sf)1

[b. Class Specifications 5](#_heading=h.4d34og8)1

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 52

d. Database queries 55

20[. Recipe Ingredient](#_heading=h.3dy6vkm) 55

[a. Class Diagram 55](#_heading=h.1t3h5sf)

[b. Class Specifications 55](#_heading=h.4d34og8)

[c. Sequence Diagram(s) 56](#_heading=h.2s8eyo1)

d. Database queries 58

21[. View Recipe Detail 5](#_heading=h.3dy6vkm)9

[a. Class Diagram 5](#_heading=h.1t3h5sf)9

[b. Class Specifications 5](#_heading=h.4d34og8)9

[c. Sequence Diagram(s) 6](#_heading=h.2s8eyo1)0

[d. Database queries 6](#_heading=h.17dp8vu)2

22[. View Ingredient Detail](#_heading=h.3dy6vkm) 63

[a. Class Diagram](#_heading=h.1t3h5sf) 63

[b. Class Specifications](#_heading=h.4d34og8) 63

[c. Sequence Diagram(s) 6](#_heading=h.2s8eyo1)4

[d. Database queries 6](#_heading=h.17dp8vu)7

23[. Favourite Recipe management](#_heading=h.3dy6vkm) 67

[a. Class Diagram](#_heading=h.1t3h5sf) 67

[b. Class Specifications](#_heading=h.4d34og8) 68

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 68

[d. Database queries](#_heading=h.17dp8vu) 69

24[. Users Order History](#_heading=h.3dy6vkm) 70

[a. Class Diagram](#_heading=h.1t3h5sf) 70

[b. Class Specifications](#_heading=h.4d34og8) 70

[c. Sequence Diagram(s)](#_heading=h.2s8eyo1) 71

[d. Database queries](#_heading=h.17dp8vu) 71

# I. Overview

## 1. Code Packages

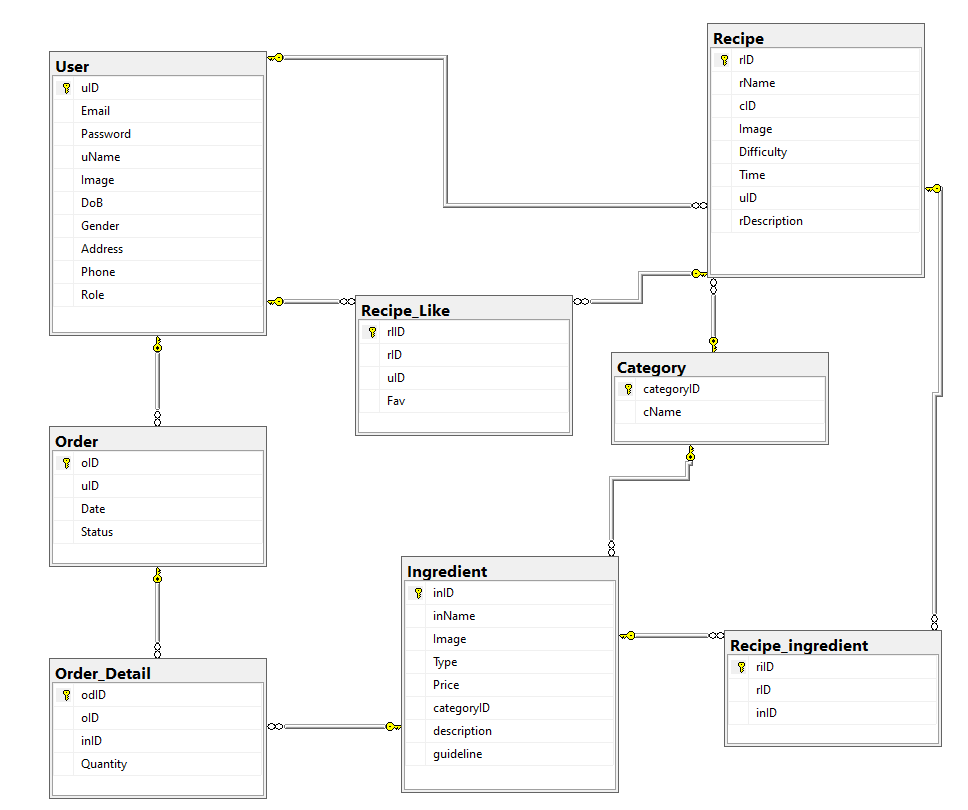


***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *View* | *Provide interaction with user,*  *GUI Java Class Naming Convention* |
| *02* | *Controller* | *Control the data flow between View and Model (DAO), Control the logic of the system Java Class Naming Convention* |
| *03* | *DAO* | *Provide the interface for the DAO to implement* |
| *04* | *Filter* | *Control the redirection between Pages in View and from View to Controller Java Class Naming Convention* |
| *05* |  |  |

## 2. Database Design

### a. Database Schema



### 

### 

### 

### b. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | User | uID int IDENTITY(1,1) NOT NULL  Email varchar](50) NOT NULL  Password varchar(50) NOT NULL  uName varchar(30) NOT NULL  Image varchar(255) NOT NULL  DoB date NULL  Gender char(1) NULL  Address varchar(50) NULL  Phone varchar(50) NULL  Role varchar(5) NULL,  - Primary keys: uID |
| *02* | Order | oID int IDENTITY(1,1) NOT NULL  uID int NOT NULL  Date datetime NOT NULL  Status bit NULL  - Primary keys: oID  - Foreign keys: uID |
| 03 | Order\_Detail | odID int IDENTITY(1,1) NOT NULL  oID int NOT NULL  inID int NOT NULL  Quantity int NOT NULL  - Primary keys: odID  - Foreign keys: oID, inID |
| 04 | Recipe | rID int IDENTITY(1,1) NOT NULL  rName nvarchar(100) NOT NULL  cID int NOT NULL  Image varchar(255) NOT NULL  Difficulty varchar(6) NOT NULL  Time int NOT NULL  uID int NOT NULL  rDescription nvarchar(max) NULL,  - Primary keys: rID  - Foreign keys: cID, uID |
| 05 | Recipe\_ingredient | riID int IDENTITY(1,1) NOT NULL  rID int NOT NULL  inID int NOT NULL  - Primary keys: riID  - Foreign keys: rID, inID |
| 06 | Recipe\_Like | rlID int IDENTITY(1,1) NOT NULL  rID int not null  uID int NOT NULL  Fav bit NULL  - Primary keys: rlID  - Foreign keys: rID, uID |
| 07 | Category | categoryID int IDENTITY(1,1) NOT NULL  cName nvarchar(30) NOT NULL  - Primary keys: categoryID |
| 08 | Ingredient | inID int IDENTITY(1,1) NOT NULL  inName nvarchar(50) NOT NULL  Image varchar(255) NOT NULL  Type varchar(30) NOT NULL  Price money NOT NULL  categoryID int NOT NULL  description nvarchar(50) NOT NULL  guideline nvarchar(max) NOT NULL  - Primary keys: inID  - Foreign keys: categoryID |

# 

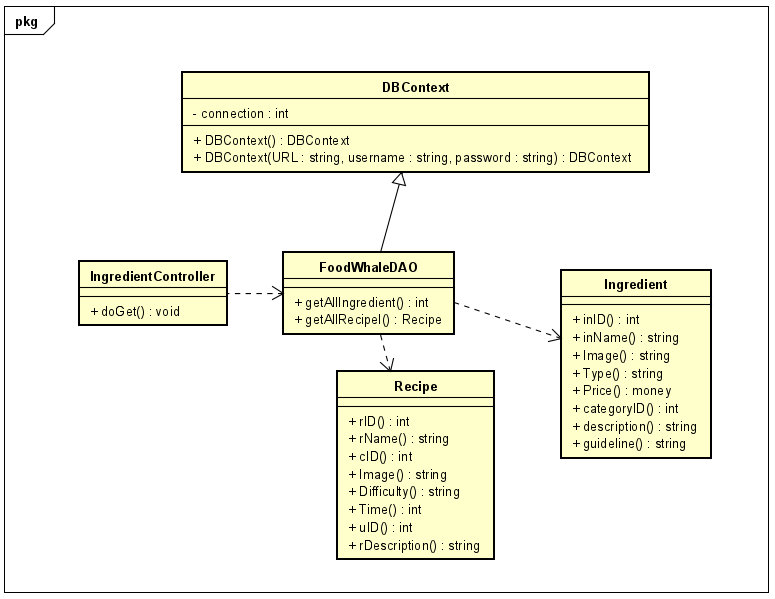
# 

# 

# II. Code Designs

## 1. Homepage

### a. Class Diagram



### b. Class Specifications

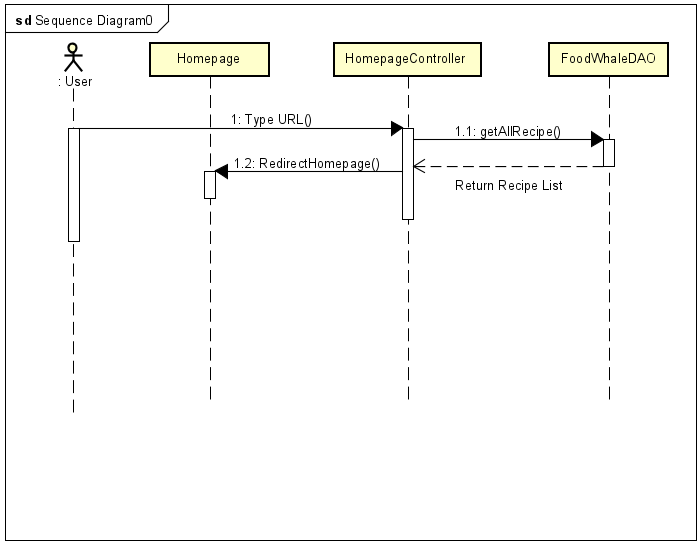
#### HomepageController

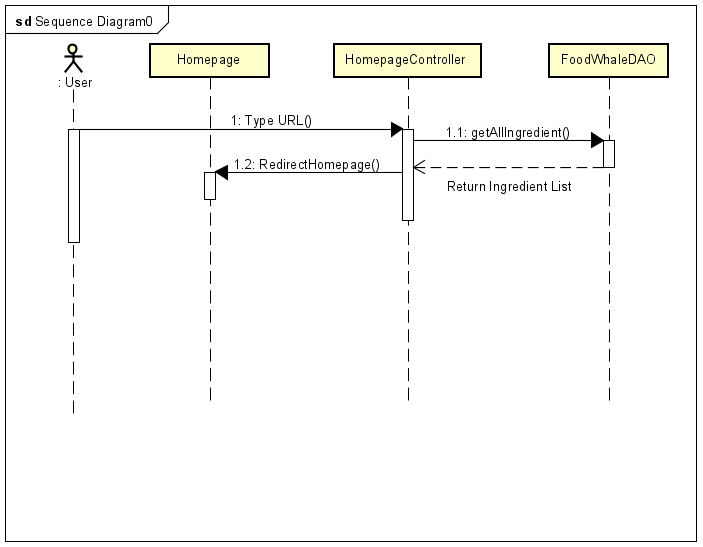
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doGet | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllRecipe() | get all Recipe |
| 2 | getAllIngredient() | get all Ingredient |

### c. Sequence Diagram(s)

**

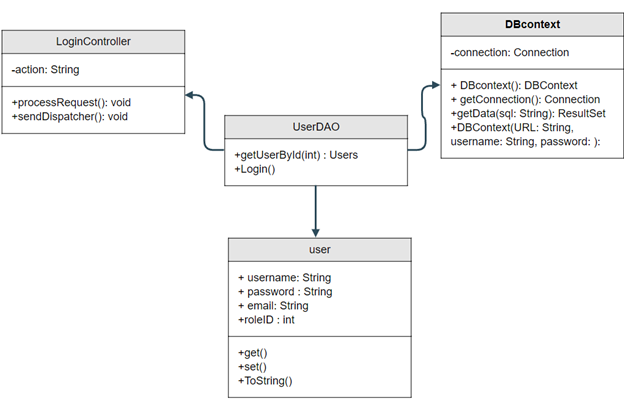
****

### d. Database queries

* select \* from foodwhale.recipe
* select \* from foodwhale.ingredient

## 2. Login

### a. Class Diagram

****

### b. Class Specifications

#### Login

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doPost() | Send a user session interval |

#### DAOauth

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | AdAuth() | Authorization |

#### DAOAdmin

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | loginAdmin() | Login as admin |

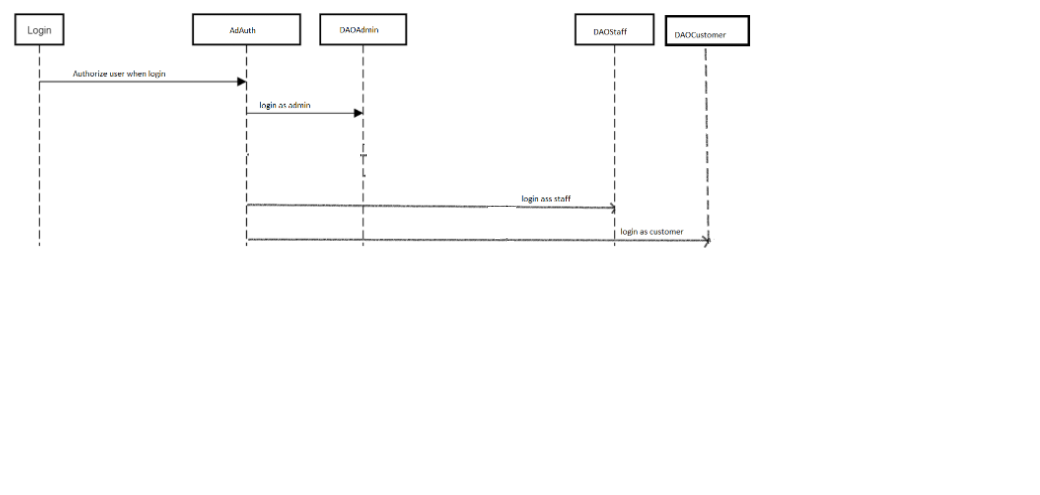
***DAOCustomer***

| No | Method | Description |
| --- | --- | --- |
| 1 | loginCustomer() | Login as customer |

***DAOStaff***

| No | Method | Description |
| --- | --- | --- |
| 1 | loginStaff() | Login as staff |

### c. Sequence Diagram(s)

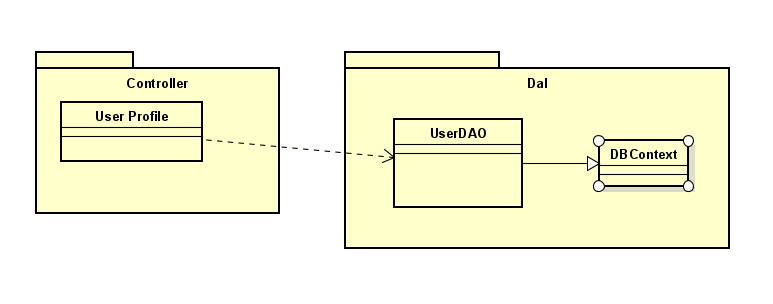


### d. Database queries

* select \* from foodwhale.user where uName = ? and Password = ?

## 3.User Profile

### a. Class Diagram:



### b. Class Specifications

#### UserProfileServlet Class

### 

| No | Method | Description |
| --- | --- | --- |
| 01 | processRequest | - input:  + HttpServletRequest request  + HttpServletResponse response  - output: N.A  - use: isNullOrEmpty, createUser, addAttributeWithExceptParam |

### 

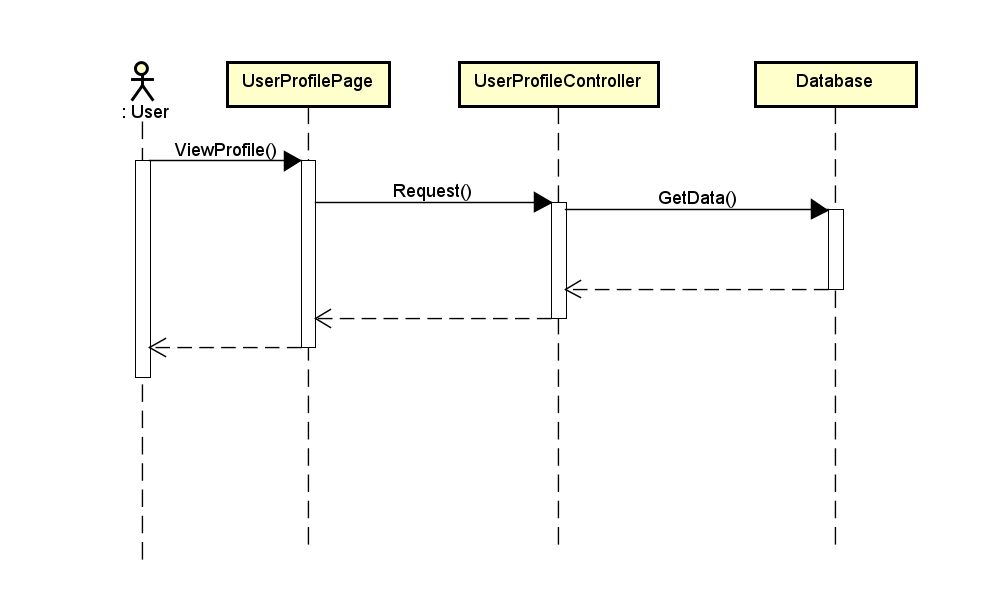
#### UserDaoImpl Class

| No | Method | Description |
| --- | --- | --- |
| 01 | GetUserByEmail | - input:  + String userId  - output: User with specified Email  - use: getConnection, PreparedStatement, Connection |

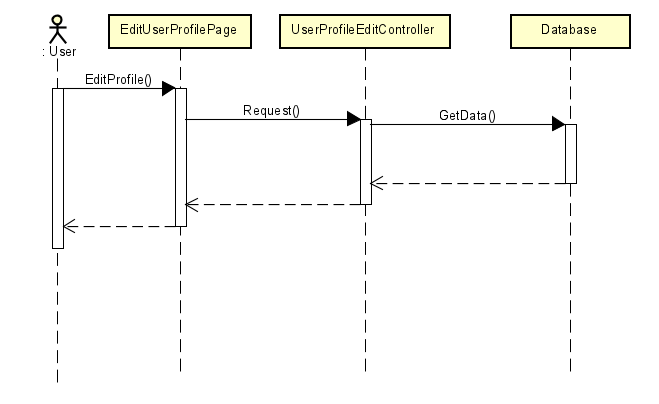
### 

### c. Sequence Diagram(s)

1. UserProfile



1. Edit UserProfile

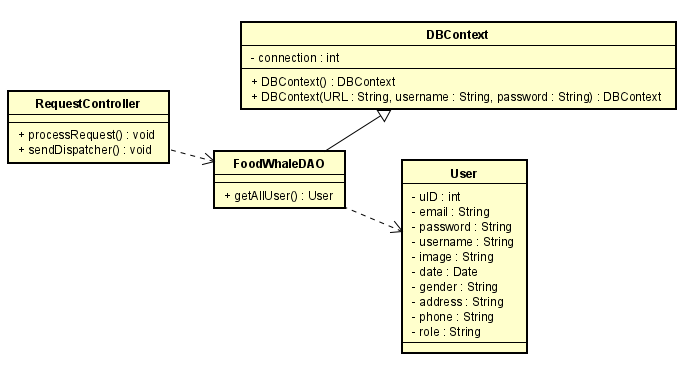


### d. Database queries

SELECT \* FROM dbo.user WHERE email = ? AND password = ? AND Status = 'True'

## 4. View list user

### a. Class Diagram



### b. Class Specifications

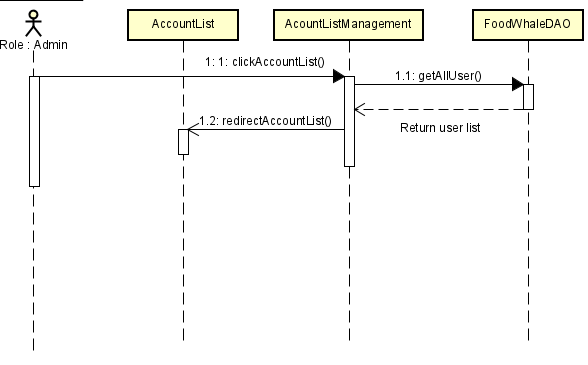
#### AccountListManagement

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllUser() | list of all user in database |

### c. Sequence Diagram(s)

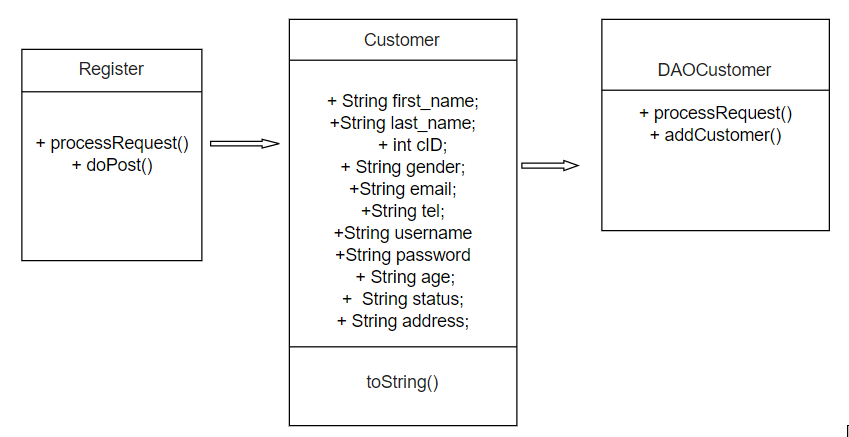


### d. Database queries

select \* from foodwhale.user

## 5. Register

### a. Class Diagram

****

### b. Class Specifications

#### Register

| No | Method | Description |
| --- | --- | --- |
| 1 | doPost() | Send a user session interval |

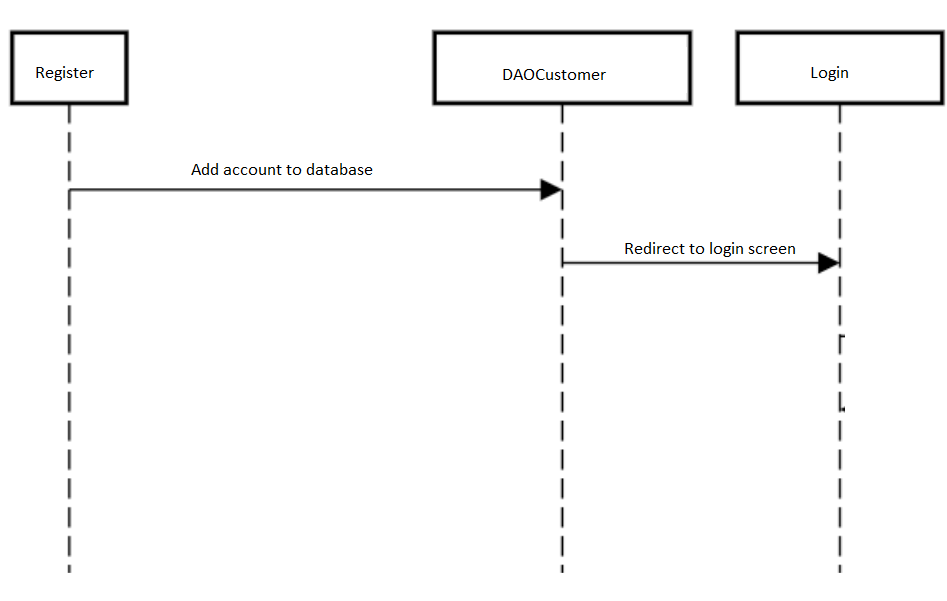
**Entity User**

| No | Method | Description |
| --- | --- | --- |
| 1 | +data field  + toString() | Take data |

DAOCustomer

| No | Method | Description |
| --- | --- | --- |
| 1 | addCustomer() | Insert account to database |

### c. Sequence Diagram(s)

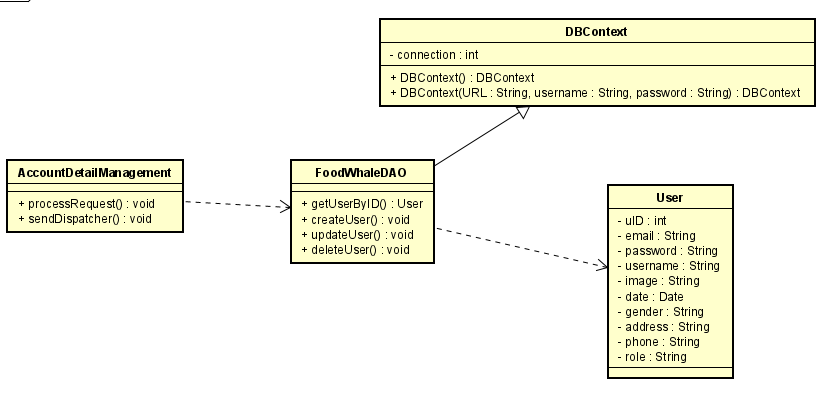
****

### d. Database queries

* insert into foodwhale.user(email, password, uName, image, DoB, gender,address,phone) values(?, ?, ?, ?, ?, ?, ?, ?)

## 6. Account detail

### a. Class Diagram



### b. Class Specifications

#### 

#### AccountListManagement

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getUserByID() | get user information from database by that ID |
| 2 | createUser() | create new user to database |
| 3 | updateUser() | update user information of that user to database |
| 4 | deleteUser() | delete user from database by that ID |

### c. Sequence Diagram(s)

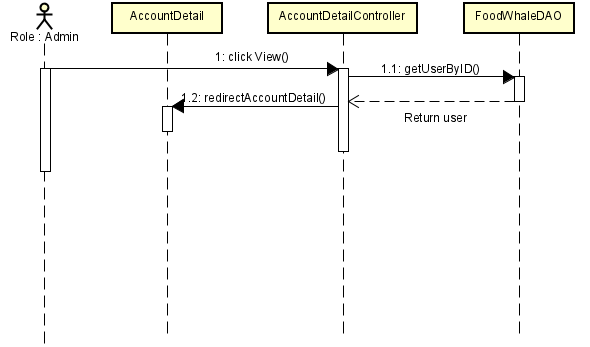


Diagram 1-1: View Account Detail

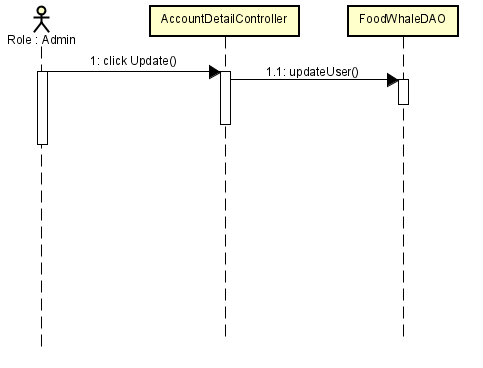


Diagram 1-2: Update user information

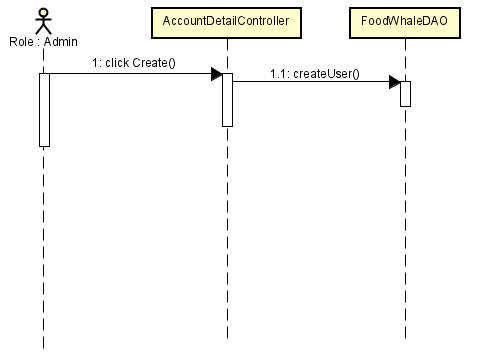


Diagram 1-3: Create new user

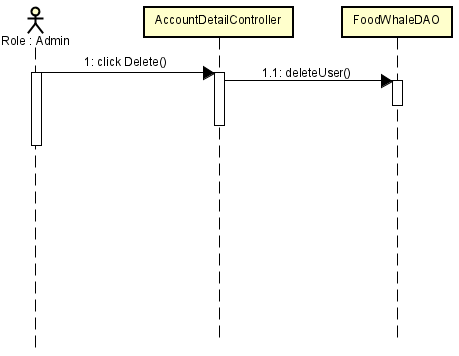


Diagram 1-4: delete User

### d. Database queries

* View

“select \* from foodwhale.user where uID= ?”

* Create

"INSERT INTO foodwhale.user(email, password, username, image, date, gender, address, phone, role) VALUES (?, ?, ?,?, ?, ?,?, ?, ?)"

* Update

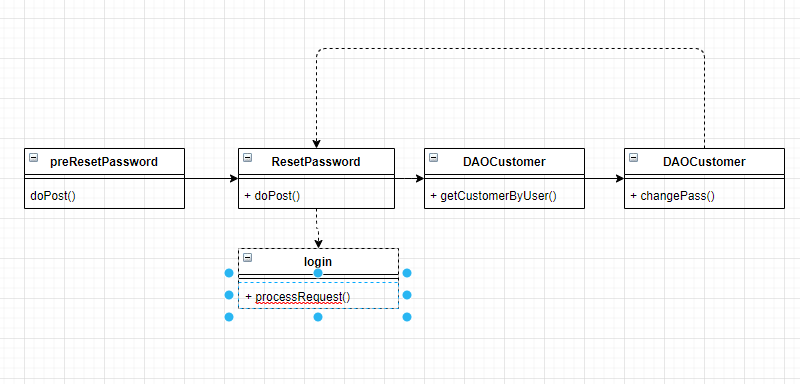
"update foodwhale.user set Email=?, Password=?, uName=?, Image=?, DoB=?, Gender=?, Address=?, Phone=?, Role=? where uID=?"

* Delete

“delete from foodwhale.user where Uid=?”

## 7. Reset Password

### Class Diagram

****

### Class specifications

preResetResetPassword

| No | Method | Description |
| --- | --- | --- |
| 1 | doPost() | Send a form to user and set session interval, send Email with reset password url to user |

ResetPassword

| No | Method | Description |
| --- | --- | --- |
| 1 | doPost() | Take username and new password |

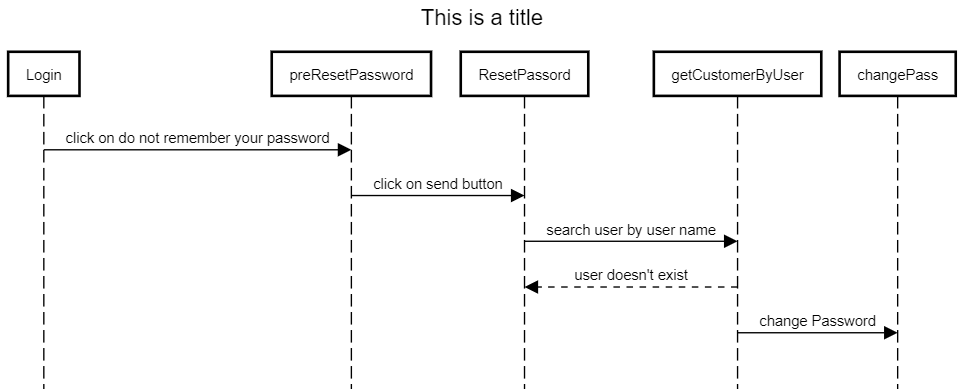
DAOCustomer

| No | Method | Description |
| --- | --- | --- |
| 1 | getCustomerByUser() | Search customer in database by their username |
| **2** | **ChangePass** | **Change password of specified user** |

**Login**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **processRequet** | **Display Login page** |

### Sequence Diagram(s)

****

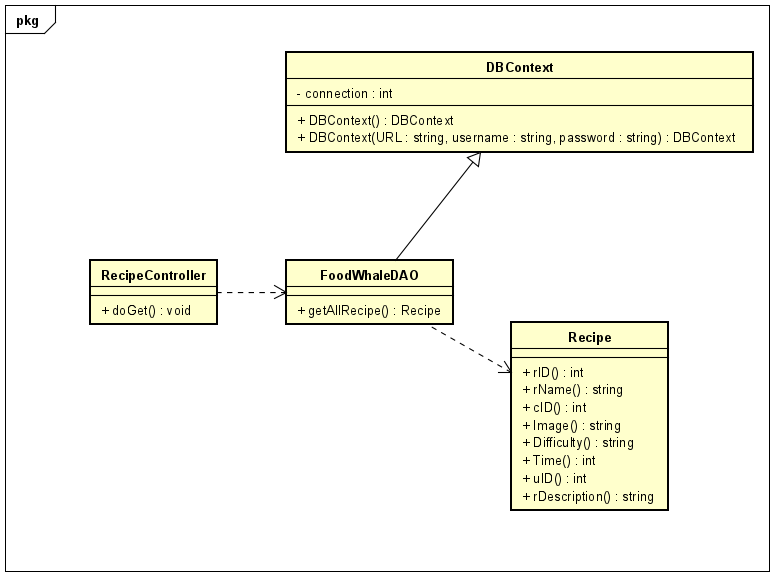
### Database queries

select uName from foodwhale.user where uName = ?

* update foodwhale.user set Password = ? where uName = ?

## 8. View Recipe

### a. Class Diagram



### b. Class Specifications

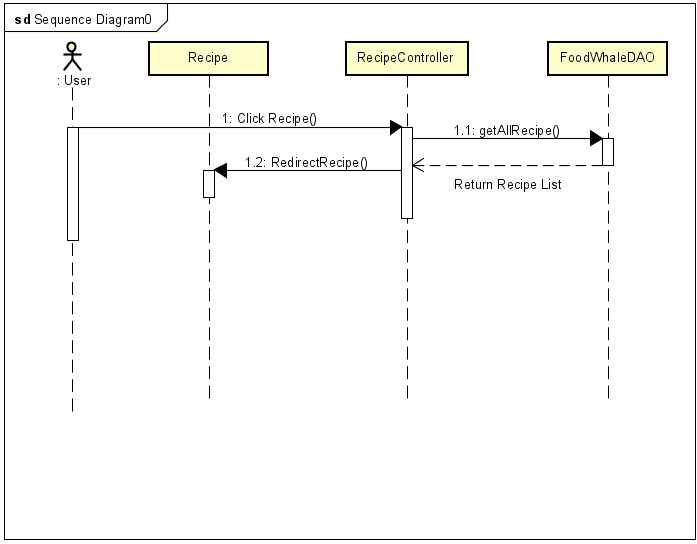
#### RecipeController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doGet | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllRecipe() | get all Recipe |

### c. Sequence Diagram(s)

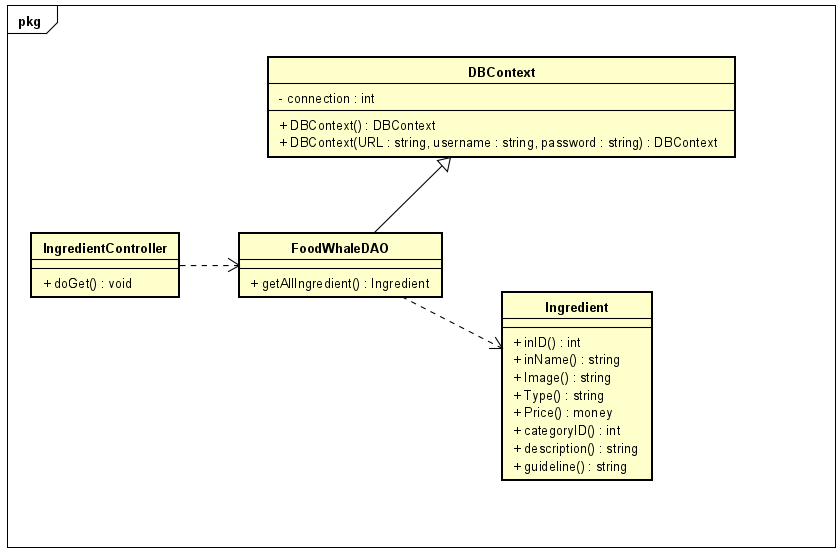
****

### d. Database queries

* select \* from foodwhale.recipe

## 9. View Ingredient

### a. Class Diagram



### b. Class Specifications

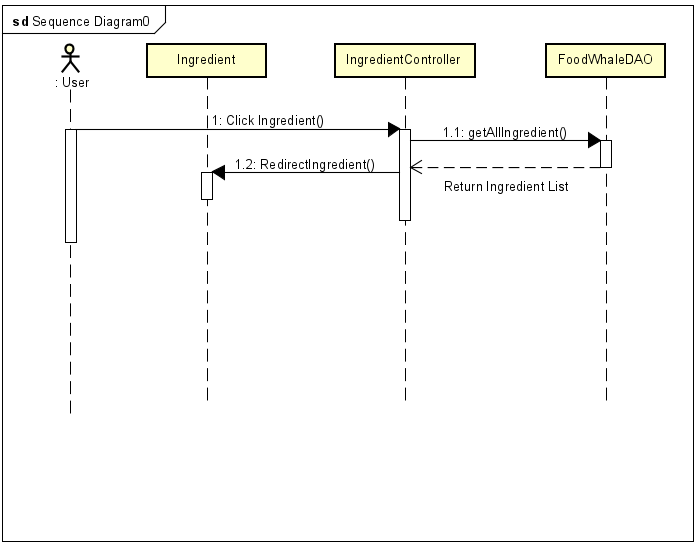
#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doGet | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllIngredient() | get all Ingredient |

### c. Sequence Diagram(s)

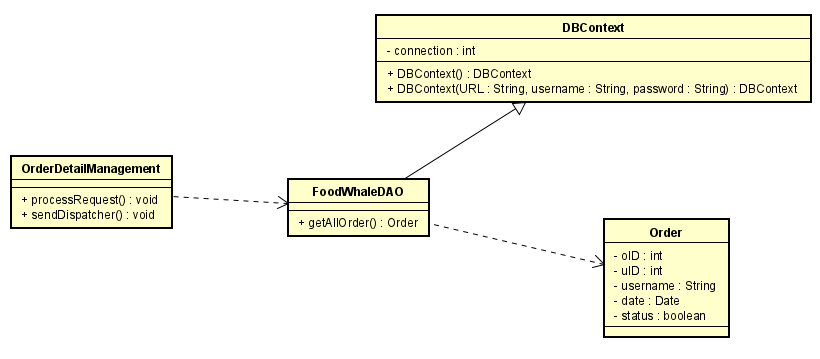
****

### d. Database queries

* select \* from foodwhale.ingredient

## 10. View list order

### a. Class Diagram



### b. Class Specifications

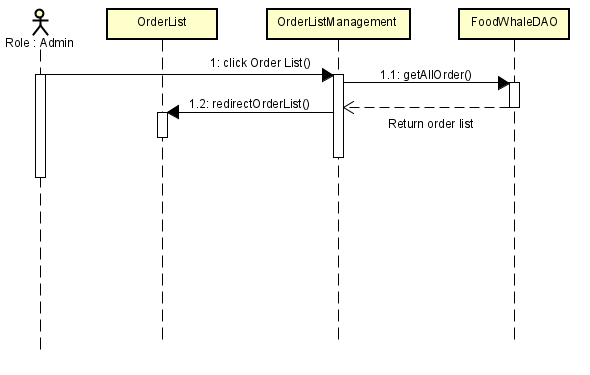
#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllOrder() | get all Order from database |

### c. Sequence Diagram

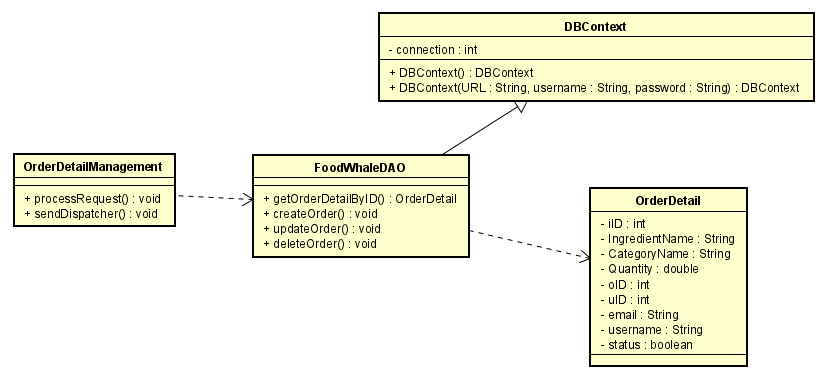
****

### d. Database queries

* select \* from foodwhale.order

## 11. Order detail

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doGet | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getOrderDetailByID() | get Order information from database by that ID |
| 2 | createOrder() | create new Order to database |
| 3 | updateOrder() | update Order Information of that Order to database |
| 4 | deleteOrder() | delete Order from database by that ID |

### c. Sequence Diagram

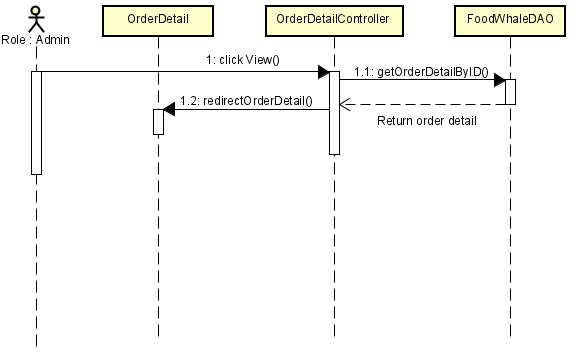


Diagram 1-1: View Order Detail

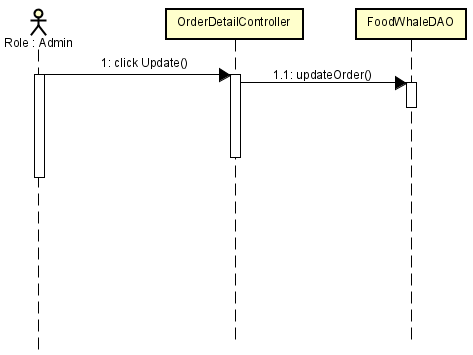


Diagram 1-2: Update Order information

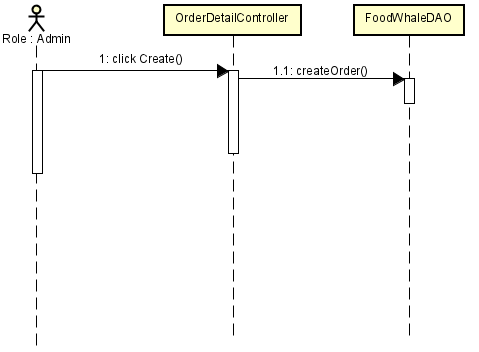


Diagram 1-3: Create new Order

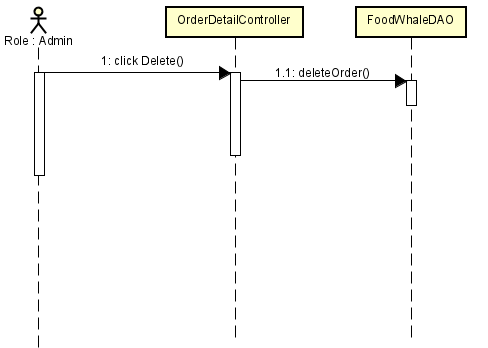


Diagram 1-4: delete Order

### d. Database queries

* View

“select \* from foodwhale.order\_detail where oID= ?”

* Create

"INSERT INTO foodwhale.order(oID, uID, date, status) VALUES (?, ?, ?,?)"

* Update

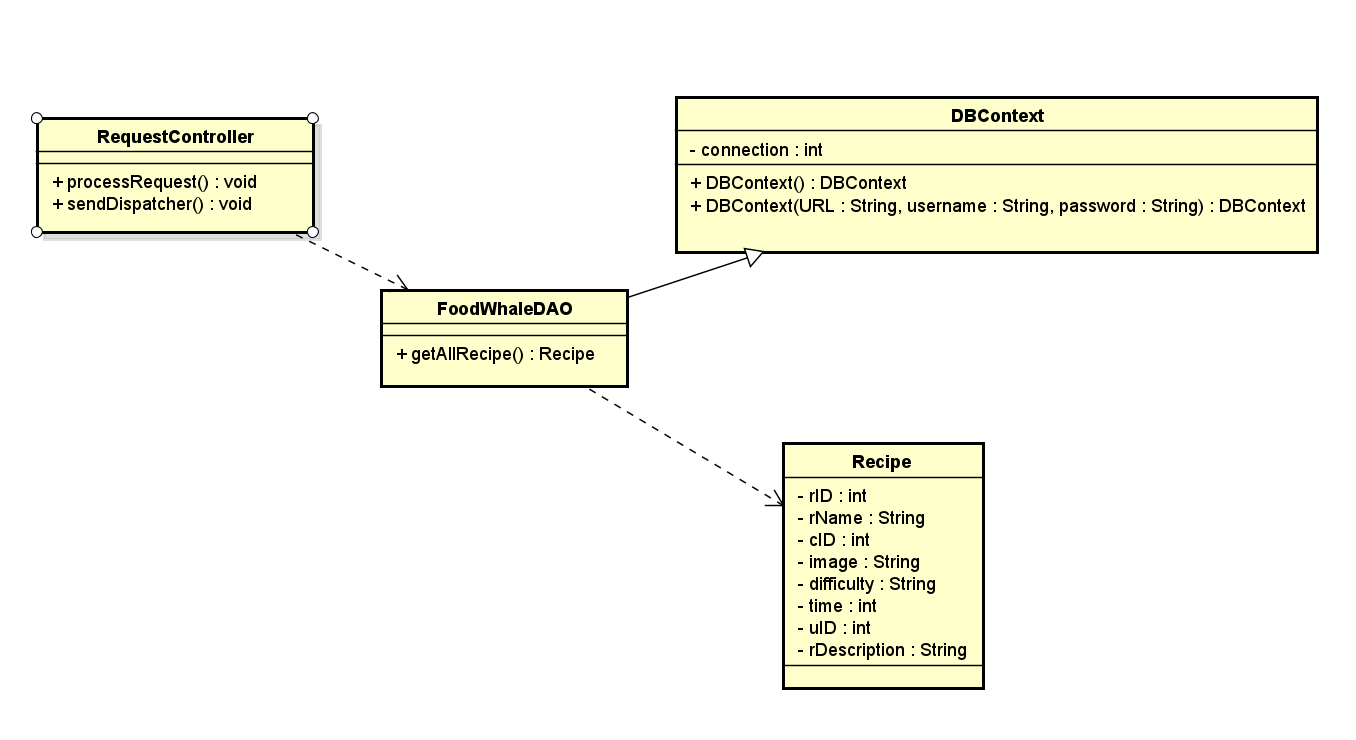
"update foodwhale.order\_detail set inID=?, oID=?, Quantity=? where odID=?"

* Delete

“delete from foodwhale.order\_detail where odID=?”

## 12. View list recipe

### a. Class Diagram

****

### b. Class Specifications

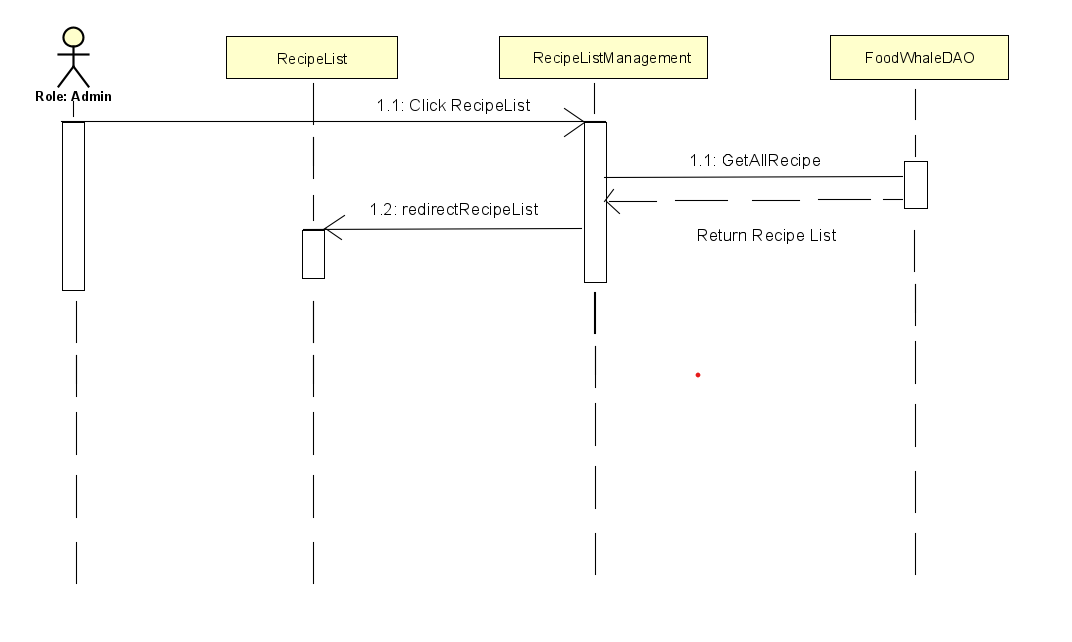
#### RecipeListManagement

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllRecipe() | list of all recipe in database |

### c. Sequence Diagram(s)

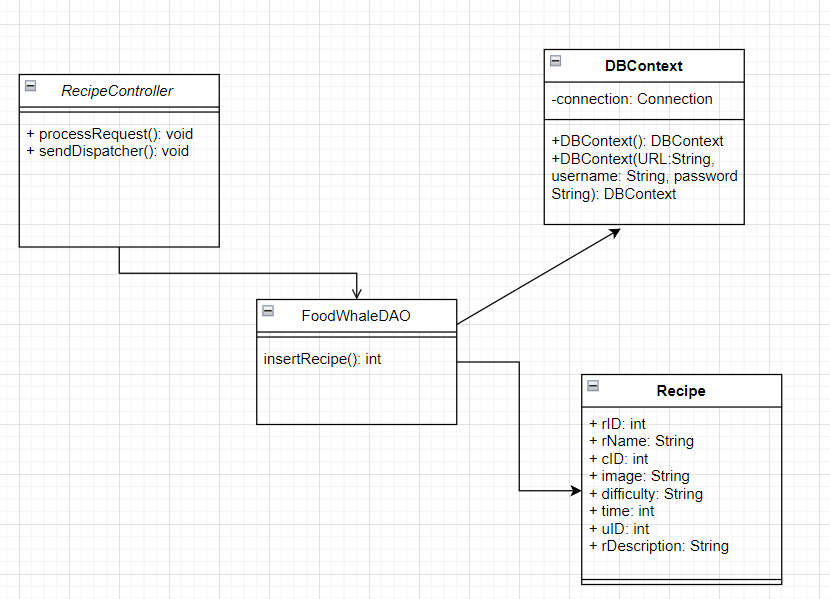


### d. Database queries

select \* from foodwhale.recipe

## 13. Post Recipe

### **a.Class diagram**



### b. Class Specifications

***RecipeController***

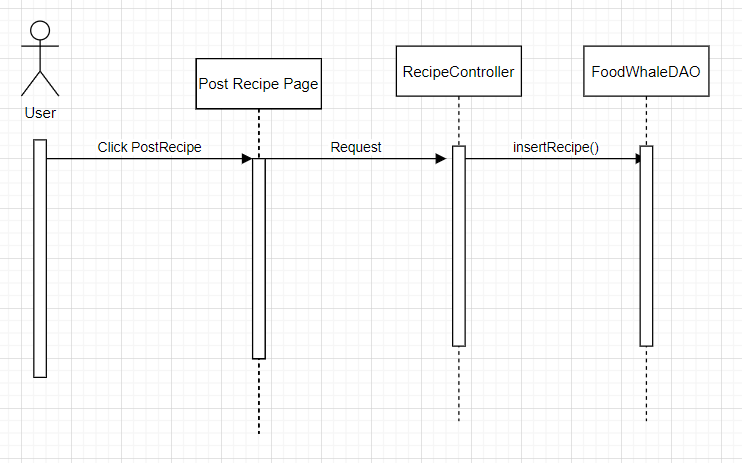
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | insertRecipe() | insert recipe into database |

### 

### c.Sequence Diagram(s)



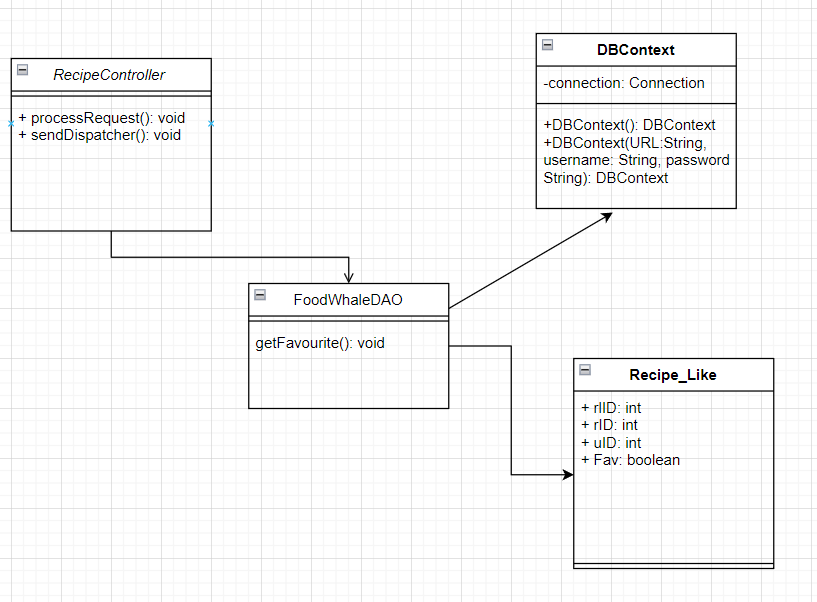
### d.Database queries

"INSERT INTO foodwhale.recipe(rName, cID, Image, Difficulty, Time,uID,rDescription)

VALUES (?, ?, ?,?,?,?,?)"

## 14. View liked Recipe

### Class diagram



### b.Class Specifications

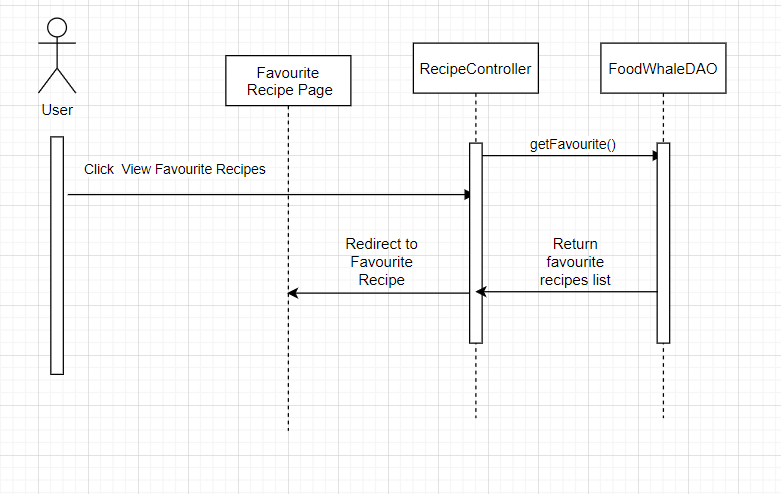
RecipeController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getFavourite() | get All Favourite recipes |

### c. Sequence Diagram

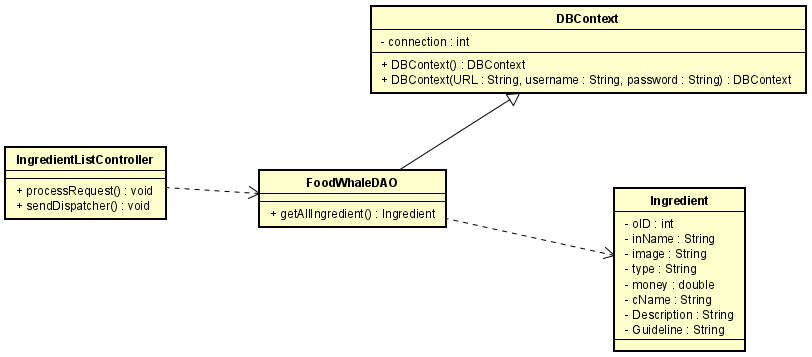


### d. Database queries

Select \* from foodwhale.recipe\_like where uID= ?

## 15. View list Ingredients

### a. Class Diagram



### b. Class Specifications

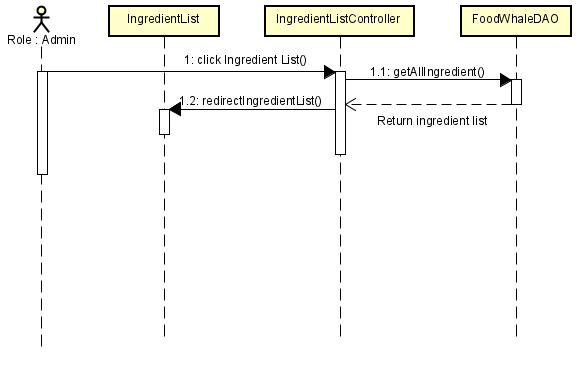
#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllIngredient() | get all Ingredients from database |

### c. Sequence Diagram

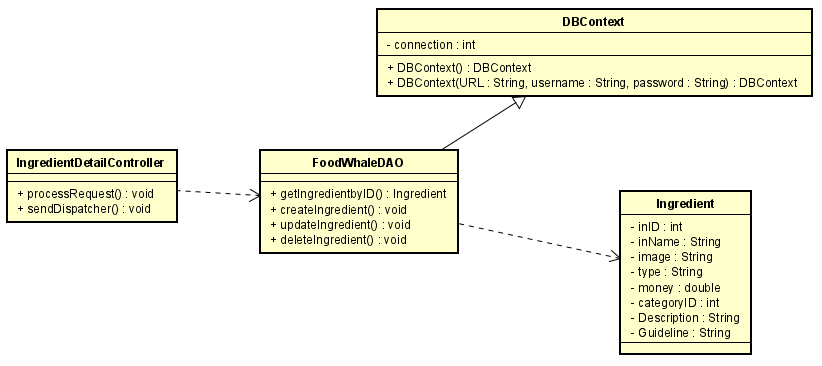
****

### d. Database queries

* select i.inID, i.inName , i.Type, i.Price, c.cName from foodwhale.ingredient i inner join foodwhale.category c on i.categoryID = c.categoryID

## 16. Ingredient Detail

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getIngredientByID() | get Ingredient information from database by that ID |
| 2 | createIngredient() | create new Ingredient to database |
| 3 | updateIngredient() | update Ingredient Information of that Ingredient To database |
| 4 | deleteIngredient() | delete Ingredient From database by that ID |

### c. Sequence Diagram

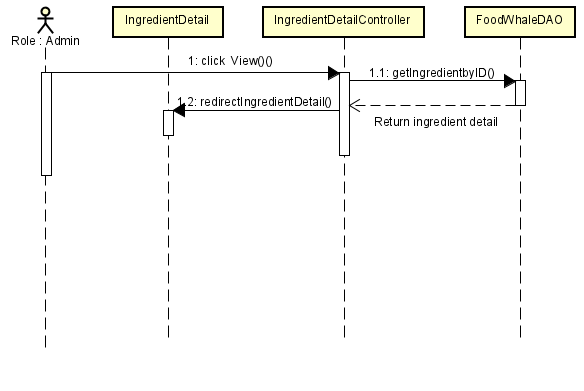
****

Diagram 1-1: View Ingredient Detail

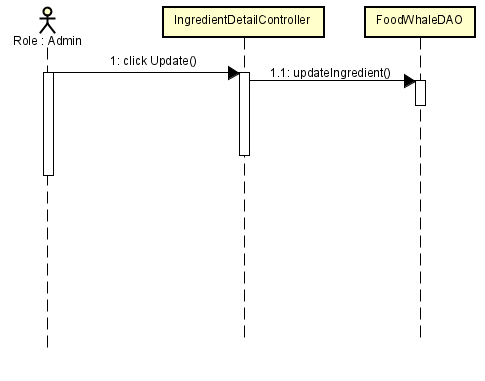


Diagram 1-2: Update Ingredient information

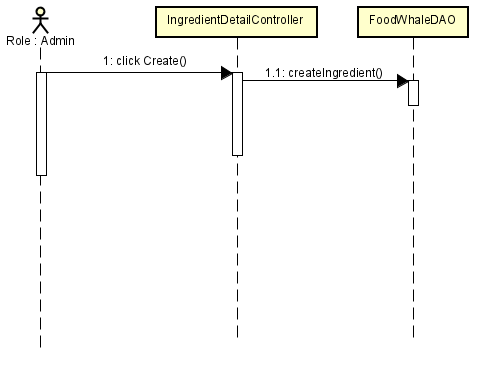


Diagram 1-3: Create new Ingredient

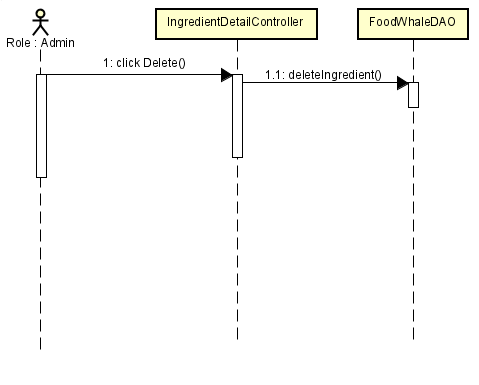


Diagram 1-4: delete Ingredient

### d. Database queries

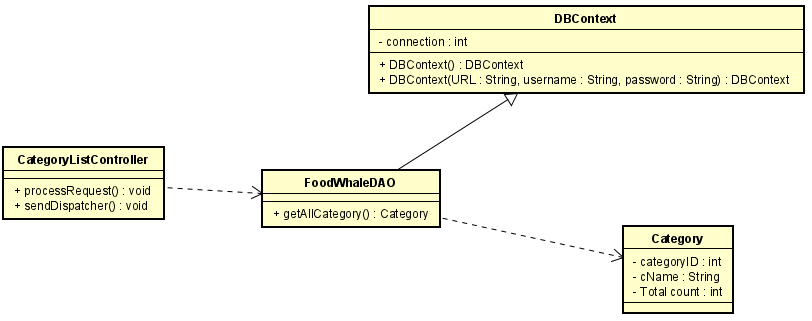
* select inID, inName , Type, Price, categoryID, Description, Guideline from foodwhale.ingredient
* update inName = ?, Type = ?, Price = ?, categoryID = ?, Description = ?, Guideline = ? where inID = ?
* insert into Ingredient(inName, Image, Type, Price, categoryID, Description, Guideline)

values (?,?,?,?,?,?,?)

* delete from foodwhale.ingredient where inID=?

## 17. View list Category

### a. Class Diagram



### b. Class Specifications

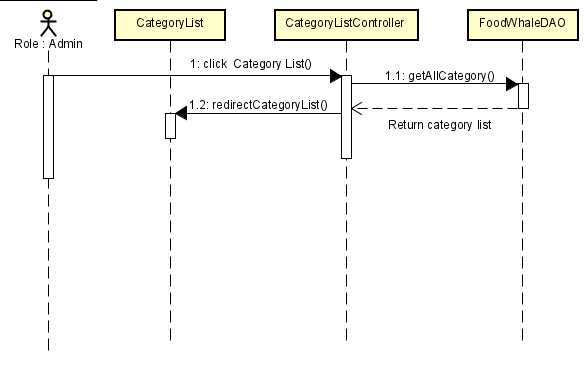
#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getAllCategory() | get all Category from database |

### c. Sequence Diagram

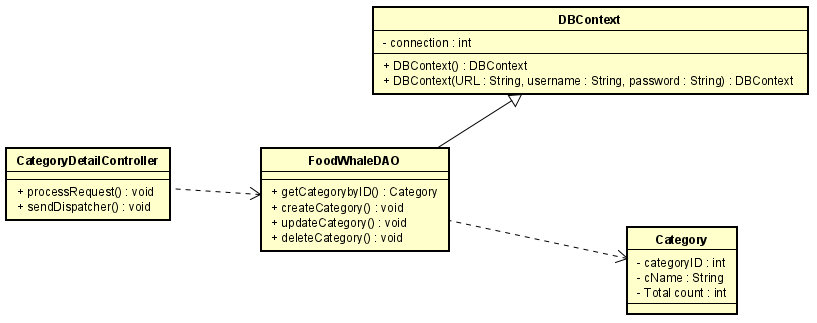
****

### d. Database queries

* select \* from foodwhale.category

## 18. Category Detail

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getCategoryByID() | get Category information from database by that ID |
| 2 | createCategory() | create new Category to database |
| 3 | updateCategory() | update Category Information of that Category To database |
| 4 | deleteCategory() | delete Category From database by that ID |

### c. Sequence Diagram

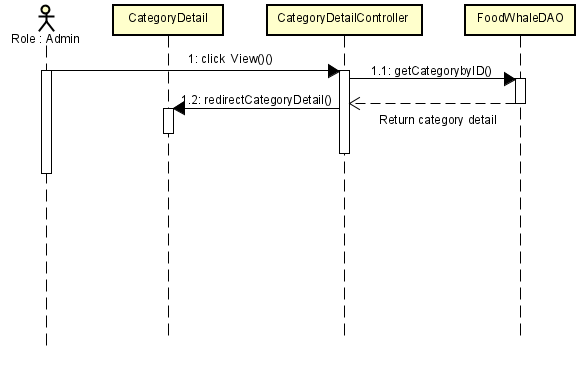
****

Diagram 1-1: View Category Detail

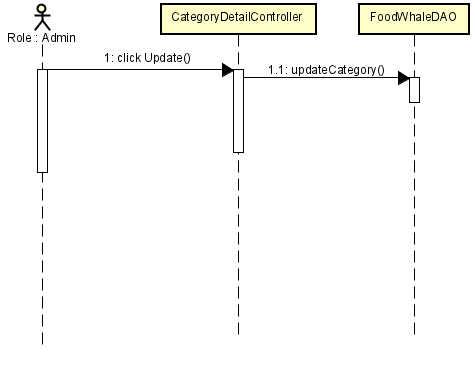


Diagram 1-2: Update Category information

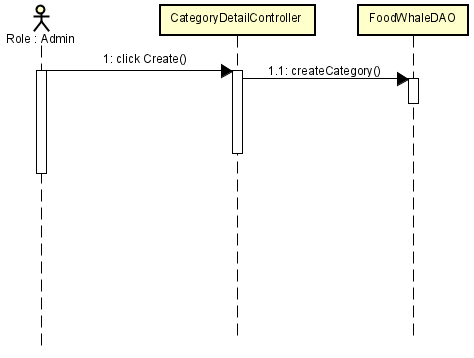


Diagram 1-3: Create new Category

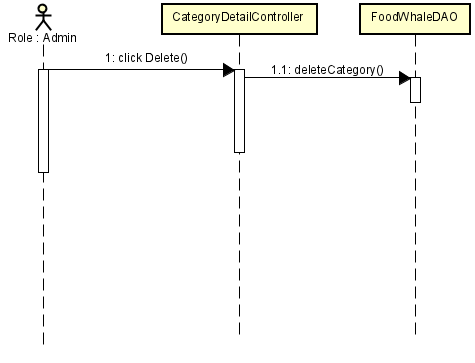


Diagram 1-4: delete Category

### d. Database queries

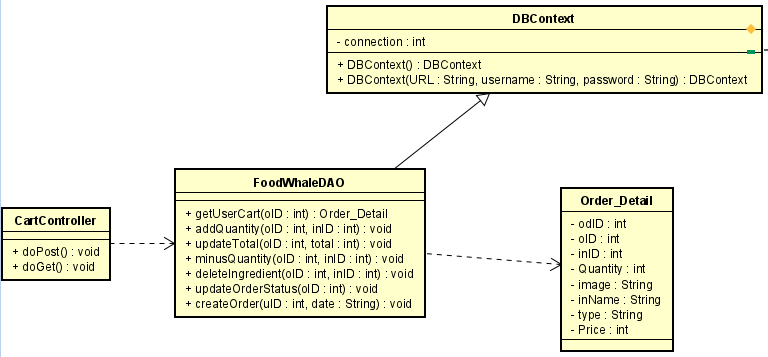
* select \* from foodwhale.category
* update foodwhale.category set cName=? where categoryID = ?
* insert into foodwhale.ingredient(inName)

values (?)

* delete from foodwhale.ingredient where categoryID =?

## 19. Cart Detail

### a. Class Diagram



### b. Class Specifications

#### CartController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getUserCart  (oID: int): Order\_Detail | get User’s Order\_Detail by User’s Order ID |
| 2 | addQuantity  (oID : int, inID : int) : void | plus one quantity for an Ingredient in User’s Order\_Detail by User’s Order ID and that Ingredient ID |
| 3 | updateTotal(oID : int, total : int) : void | update Total Price for User’s Order in Order by User’s Order ID and the Total Price |
| 4 | minusQuantity(oID : int, inID : int) : void | minus one quantity for an Ingredient in User’s Order\_Detail by User’s Order ID and that Ingredient ID |
| 5 | deleteIngredient(oID : int, inID : int) : void | delete an Ingredient in User’s Order\_Detail by User’s Order ID and that Ingredient ID |
| 6 | checkOrderEmpty(oID : int) : boolean | check if User’s Order\_Detail is Empty or not by User’s Order ID |
| 7 | updateOrderStatus(oID : int) : void | change Status of User’s Order by User’s Order ID |
| 8 | createOrder(uID : int, date : String) : void | create new Order for User with User’s ID and current Date |

### c. Sequence Diagram

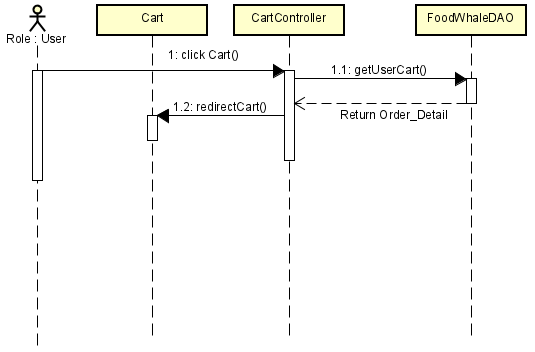
****

Diagram 1-1: View Cart

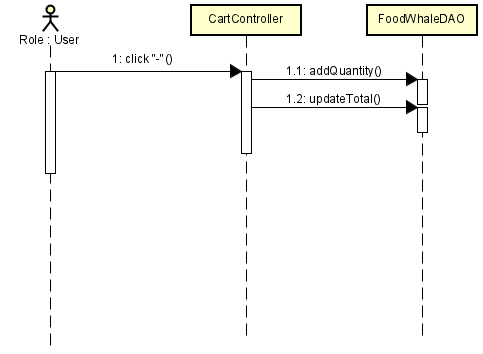


Diagram 1-2: Add Quantity, update Total

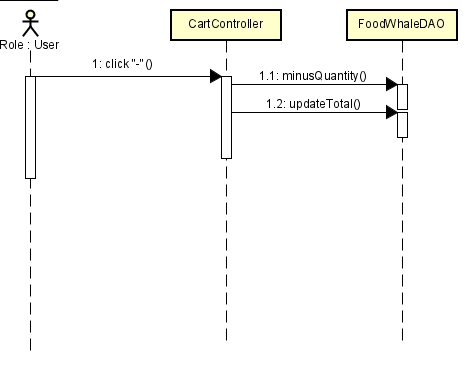


Diagram 1-3: Minus Quantity, update Total

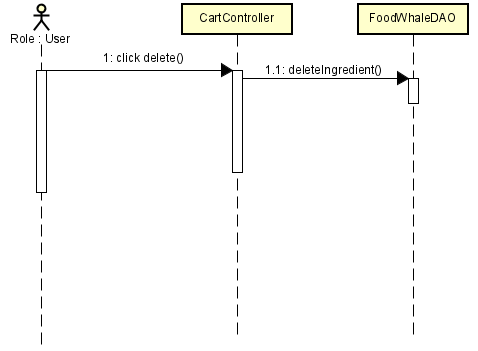


Diagram 1-4: delete Ingredient

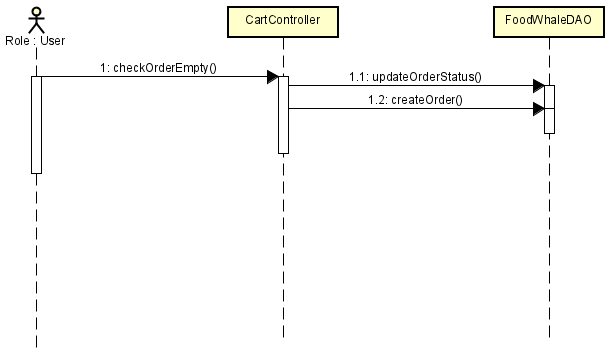


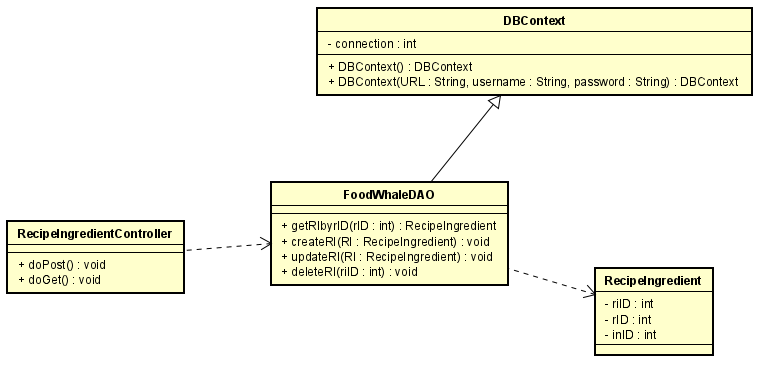
Diagram 1-5: CheckOrderEmpty, updateOrderStatus, createOrder

### d. Database queries

* select \* from foodwhale.order\_detail od inner join foodwhale.ingredient i on od.inID = i.inID where od.oID = (?)
* update foodwhale.order\_detail SET Quantity = Quantity + 1 where oID = (?) and inID (?)
* update foodwhale.order set Total = (?) where oID = (?)
* update foodwhale.order\_detail SET Quantity = Quantity - 1 where oID = (?) and inID = (?)
* delete from foodwhale.order\_detail where oID = (?) and inID = (?)
* select count(\*) 'count' from foodwhale.order\_detail where oID = (?)
* update foodwhale.order set Status = 'Waiting' where oID = (?)
* insert into foodwhale.order(uID, Date, Total) values (?, ?, 0)

## 20. Recipe Ingredient

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doPost() | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |
| 2 | doGet() | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getRIbyrID(rID: int):  RecipeIngredient | get RecipeIngredient information from database by rID |
| 2 | createRI(RI: RecipeIngredient): void | create new RecipeIngredient to database |
| 3 | updateRI(RI:  RecipeIngredient):  void | update RecipeIngredient Information of that RecipeIngredient to database |
| 4 | deleteRI(rID: int): void | delete RecipeIngredient from database by that rID |

### c. Sequence Diagram

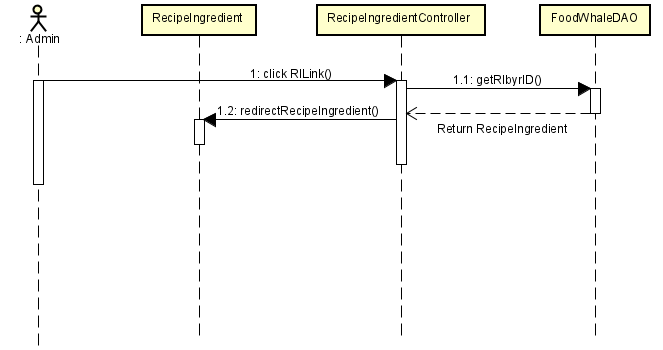
****

Diagram 1-1: View RecipeIngredient

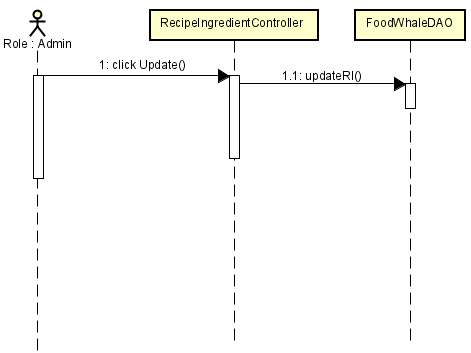


Diagram 1-2: Update RecipeIngredient

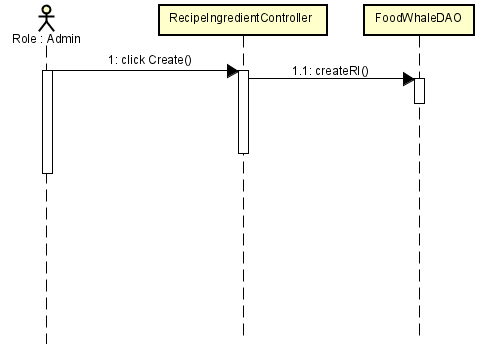


Diagram 1-3: Create new RecipeIngredient

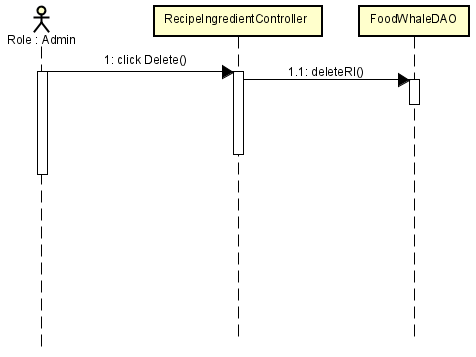


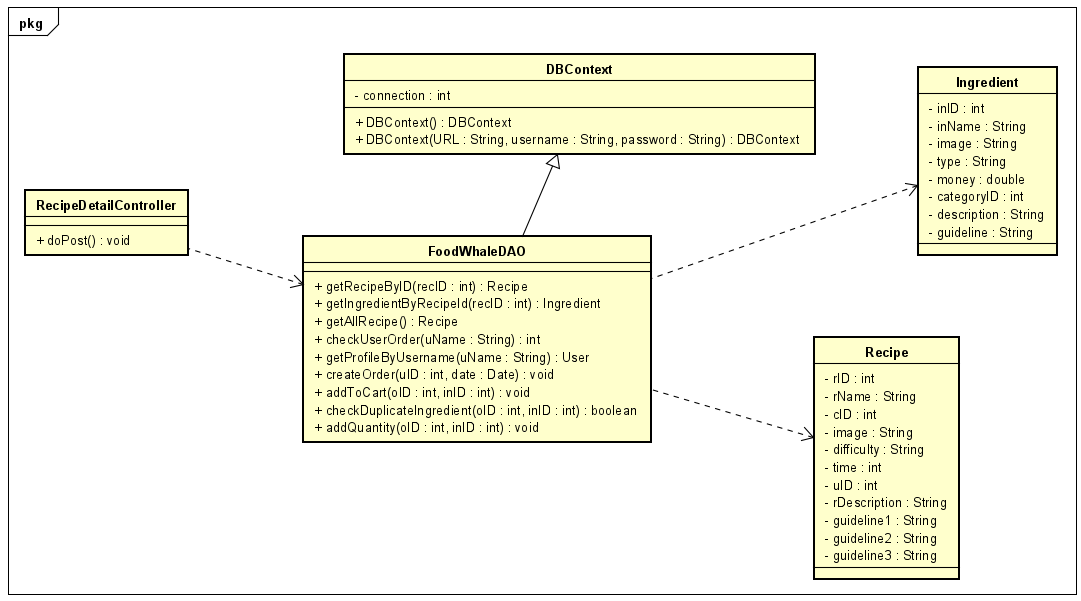
Diagram 1-4: delete RecipeIngredient

### d. Database queries

* select \* from foodwhale.recipe\_ingredient
* update foodwhale.recipe\_ingredient set rID=?, inID = ? where riID = ?
* insert into foodwhale.recipe\_ingredient(rID, inID) values (?,?)
* delete from foodwhale.recipe\_ingredient where riID =?

## 21. View Recipe Detail

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | doPost | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

#### FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getRecipeByID(recID : int) : Recipe | get Recipe information from database by Recipe ID |
| 2 | getIngredientByRecipeId(recID : int) : Ingredient | get Ingredient information of that Recipe by Recipe ID |
| 3 | getAllRecipe() : Recipe | get all Recipe information from database |
| 4 | checkUserOrder(uName : String) : int | get User’s Order number by Username |
| 5 | getProfileByUsername(uName : String) : User | get User’s Profile information by Username |
| 6 | createOrder(uID : int, date : Date) : void | create new Order for User with User’s ID and current Datetime |
| 7 | addToCart(oID : int, inID : int) : void | add Ingredient to Cart by Order ID and Ingredient ID |
| 8 | checkDuplicateIngredient(oID : int, inID : int) : boolean | check if a Ingredient has existed in User’s Order or not |
| 9 | addQuantity(oID : int, inID : int) : void | add one more Quantity to User’s Cart by Order ID and Ingredient ID |

### c. Sequence Diagram

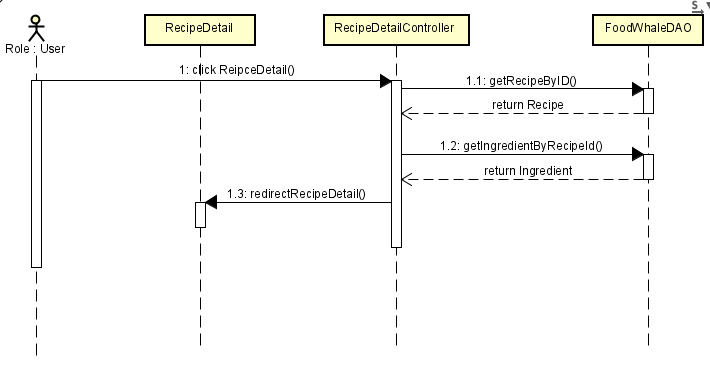
****

Diagram 1-1: View Recipe Detail

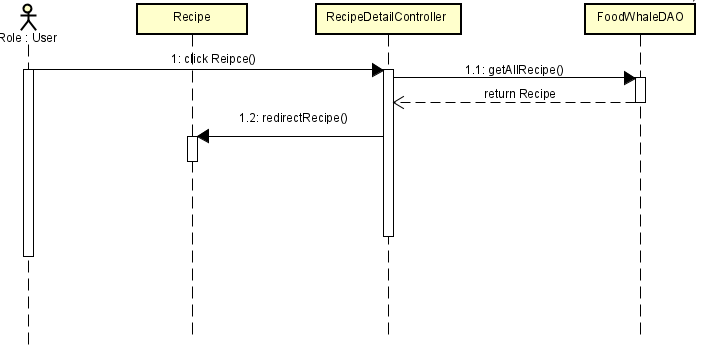


Diagram 1-2: View RecipeList ìf rID=null

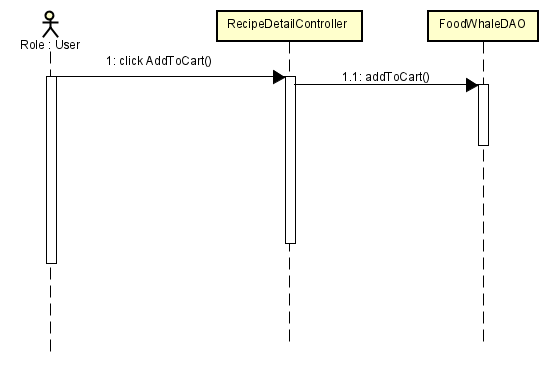


Diagram 1-3: Add To Cart

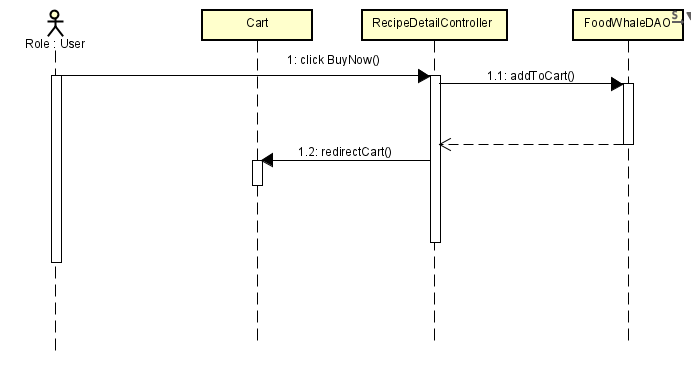


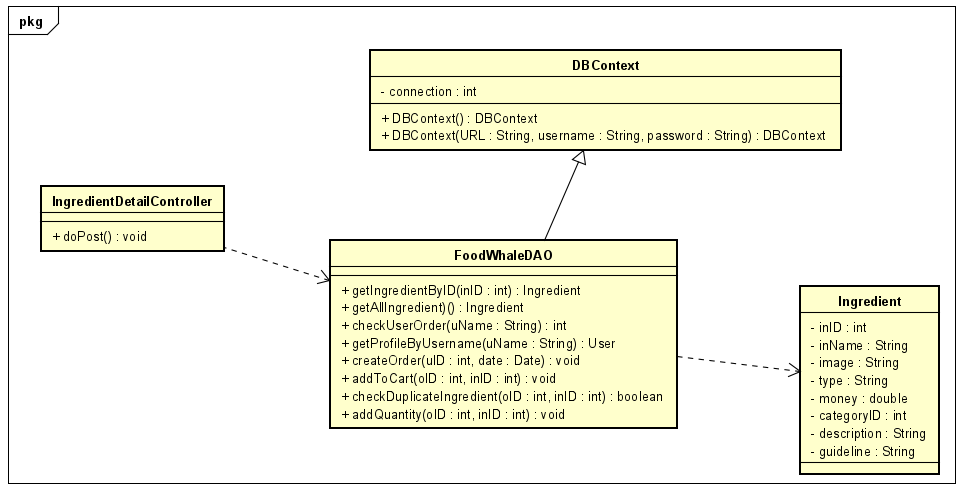
Diagram 1-4: Buy Now

### d. Database queries

* select r.rID, r.rName, c.cName, r.Image, r.Difficulty, r.Time, r.uID, r.rDescription, r.Guideline1, r.Guideline2, r.Guideline3 from foodwhale.recipe r inner join foodwhale.category c on r.cID = c.categoryID where r.rID = (?)
* select i.inID, i.inName, i.Image, i.Type, i.Price, i.categoryID, i.description, i.guideline from foodwhale.recipe\_ingredient ri inner join foodwhale.ingredient i on ri.inID = i.inID where ri.rID = (?)
* select \* from foodwhale.recipe order by rID desc
* select MAX(oID) as 'orderNo' from foodwhale.order o inner join foodwhale.user u on o.uID = u.uID where u.uName = (?)
* select \* from foodwhale.user where uName = (?)
* insert into foodwhale.order(uID, Date, Total) values (?, ?, 0)
* insert into foodwhale.order\_detail(oID, inID, Quantity) values (?, ?, 1)
* select count(\*) 'count' from foodwhale.order\_detail where oID = (?) and inID = (?)
* update foodwhale.order\_detail SET Quantity = Quantity + 1 where oID = (?) and (?)

## 22. View Ingredient Detail

### a. Class Diagram



### b. Class Specifications

#### IngredientController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getIngredientByID(inID : int) : Recipe | get Ingredient information from database by Ingredient ID |
| 2 | getAllIngredient() : Ingredient | get all Ingredient information from database |
| 3 | checkUserOrder(uName : String) : int | get User’s Order number by Username |
| 4 | getProfileByUsername(uName : String) : User | get User’s Profile information by Username |
| 5 | createOrder(uID : int, date : Date) : void | create new Order for User with User’s ID and current Datetime |
| 6 | addToCart(oID : int, inID : int) : void | add Ingredient to Cart by Order ID and Ingredient ID |
| 7 | checkDuplicateIngredient(oID : int, inID : int) : boolean | check if a Ingredient has existed in User’s Order or not |
| 8 | addQuantity(oID : int, inID : int) : void | add one more Quantity to User’s Cart by Order ID and Ingredient ID |

### c. Sequence Diagram

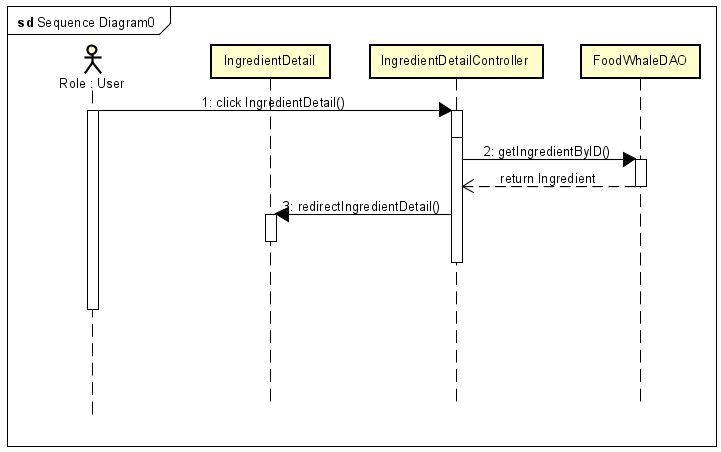
****

Diagram 1-1: View Ingredient Detail

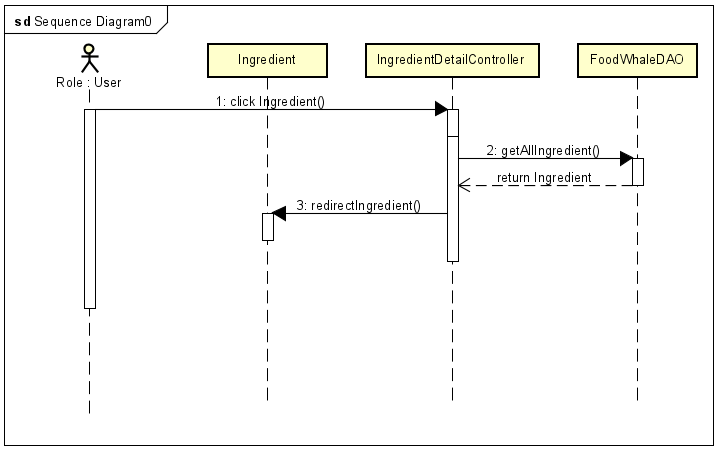


Diagram 1-2: View IngredientList if inID = null

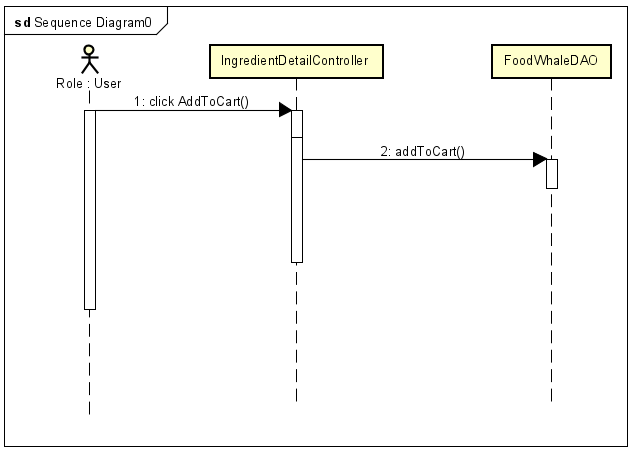


Diagram 1-3: Add to Cart

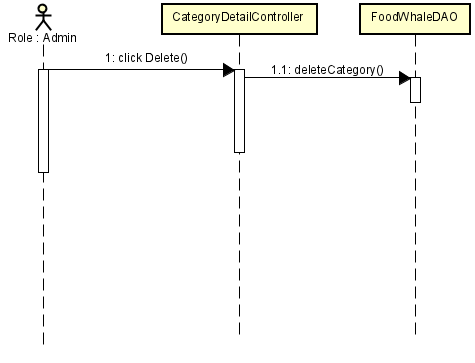


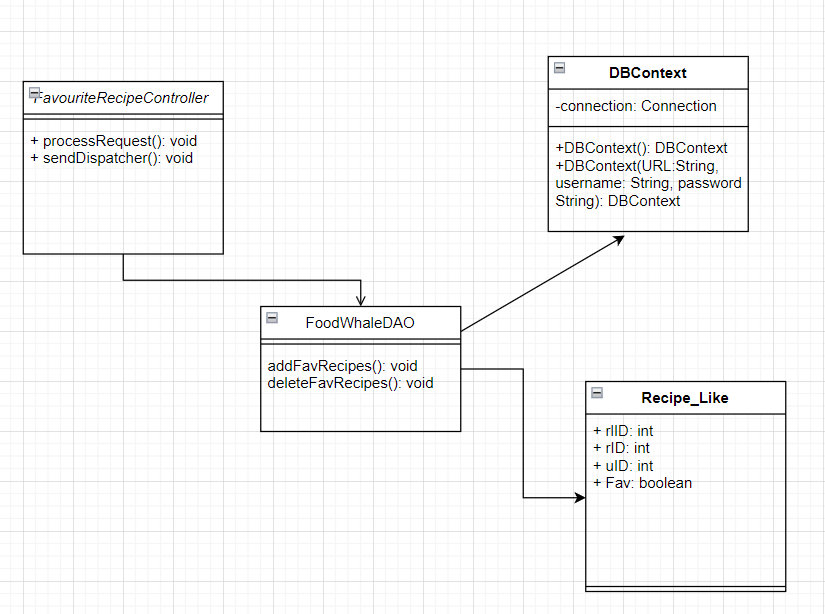
Diagram 1-4: delete Category

### d. Database queries

* select i.inID, i.inName,i.Image, i.Type, i.Price, c.cName, i.description, i.Guideline from foodwhale.ingredient i inner join foodwhale.category c on i.categoryID = c.categoryID where i.inID = (?)
* select \* from foodwhale.ingredient order by rID desc
* select MAX(oID) as 'orderNo' from foodwhale.order o inner join foodwhale.user u on o.uID = u.uID where u.uName = (?)
* select \* from foodwhale.user where uName = (?)
* insert into foodwhale.order(uID, Date, Total) values (?, ?, 0)
* insert into foodwhale.order\_detail(oID, inID, Quantity) values (?, ?, 1)
* select count(\*) 'count' from foodwhale.order\_detail where oID = (?) and inID = (?)
* update foodwhale.order\_detail SET Quantity = Quantity + 1 where oID = (?) and (?)

## 23. Favourite Recipe management

### a.Class Diagram



### b.Class Specification

FavouriteRecipeController

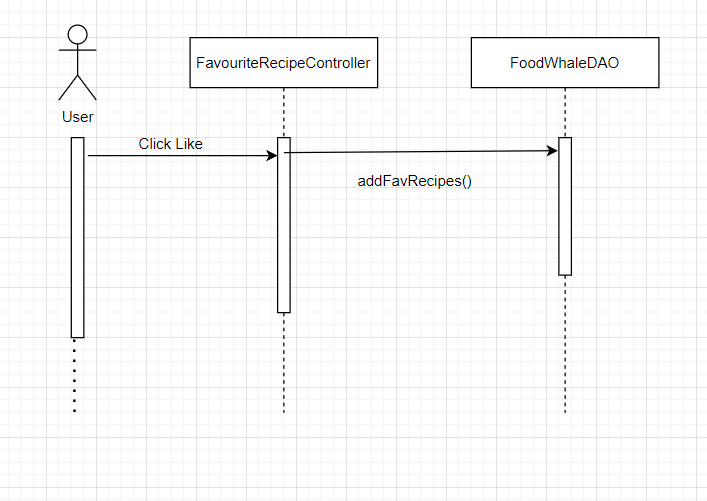
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

FoodWhaleDAO

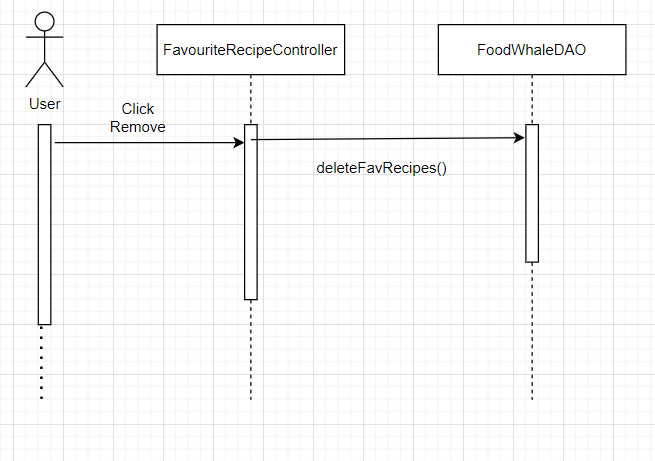
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | addFavRecipes() | add selected recipe to favourite list |
| 2 | deleteFavRecipes() | remove selected recipe from favourite list |

### c.Sequence Diagram

-Add:



-Remove:

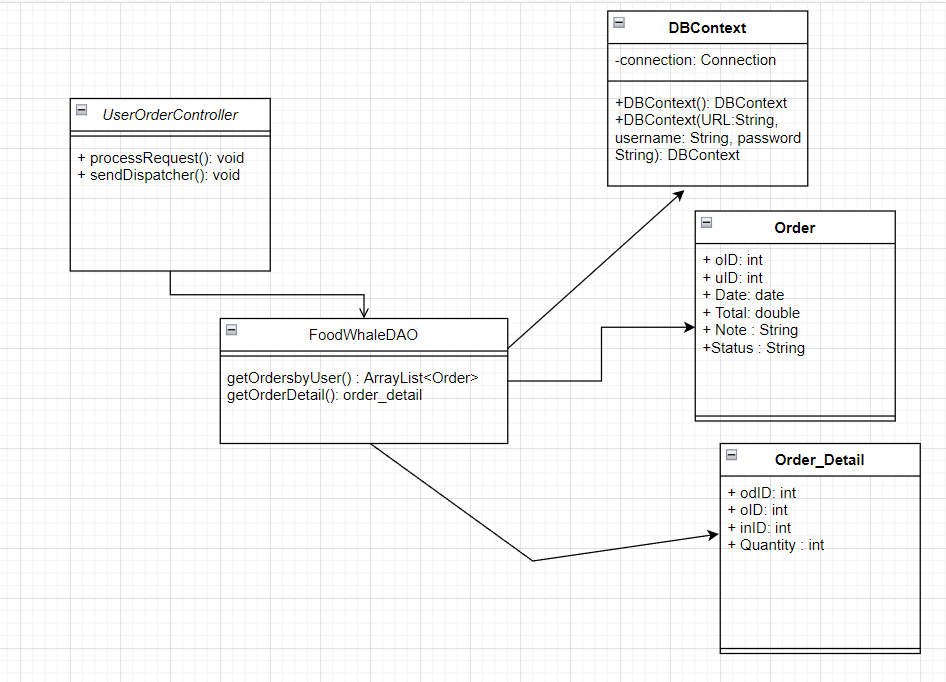


### ***d.Database queries***

* insert into recipe\_like (rId,uId) values (?,?)
* delete from recipe\_like where rId= ? and uId=?

## 24. Users Order History

### a.Class Diagram



### b.Class Specifications

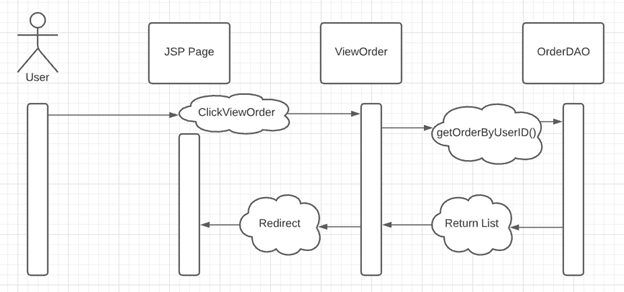
UserOrdersController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | processRequest | * Return: Null * Parameter: HttpServletRequest request, HttpServletResponse response |

FoodWhaleDAO

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | getOrdersbyUser() | get all orders by user ID |
| 2 | getOrderDetail() | get details of selected order |
| 3 | getIngredientbyID() | get all ingredients in user’s order |

### c.Sequence Diagram



### d.Database queries

+ SELECT \* from order where uID= ?

+ Select \* from order\_detail where oID=?