**Final Report**

**for**

**ThinkFood**

**By**

**ICS Group 06**

Contents

[1. Project Overview 4](#_Toc5808504)

[2. Project Specifications 5](#_Toc5808505)

[2.1 Function Requirements 5](#_Toc5808506)

[2.2 Non-Function Requirements 5](#_Toc5808507)

[2.3 Group Roles 6](#_Toc5808508)

[3. Usability Guide 7](#_Toc5808509)

[3.1 Customer 7](#_Toc5808510)

[3.2 Restaurant 7](#_Toc5808511)

[3.3 Administrator 16](#_Toc5808512)

[4. Problems and Challenges 18](#_Toc5808513)

[4.1 Managing tasks on schedule 18](#_Toc5808514)

[4.2 Standards 18](#_Toc5808515)

[4.3 GitLab 18](#_Toc5808516)

[4.4 Analying requirements 18](#_Toc5808517)

[5. Skill Learned 18](#_Toc5808518)

[5.1 Cooperation 18](#_Toc5808519)

[5.2 GitLab 18](#_Toc5808520)

[5.3 MySQL and Workbench 18](#_Toc5808521)

[5.4 PHP, HTML, CSS 18](#_Toc5808522)

[6. Proposed Changes 19](#_Toc5808523)

[6.1 Time management 19](#_Toc5808524)

[6.2 Dedicated role assign 19](#_Toc5808525)

[6.3 Avoid duplicate statements 19](#_Toc5808526)

[6.4 Avoid duplicate statements 19](#_Toc5808527)

[6.5 Avoid duplicate statements 19](#_Toc5808528)

[7. Code Snippets 19](#_Toc5808529)

[7.1 Batch for recommend products 19](#_Toc5808530)

[7.2 Add product to cartt 19](#_Toc5808531)

[7.3 ppppppp 20](#_Toc5808532)

**Table of Figures**

[Figure 1. System Architecture 4](#_Toc5814433)

[Figure 2. Privacy Term Accpet 7](#_Toc5814434)

[Figure 3. Login 7](#_Toc5814435)

[Figure 4. Login 8](#_Toc5814436)

[Figure 5. Product 8](#_Toc5814437)

[Figure 6. Add to cart 9](#_Toc5814438)

[Figure 7. Product Review 9](#_Toc5814439)

[Figure 8. Order History 10](#_Toc5814440)

[Figure 9. Review 10](#_Toc5814441)

[Figure 10. Cart 11](#_Toc5814442)

[Figure 11. Cart 11](#_Toc5814443)

[Figure 12. Update a cart 12](#_Toc5814444)

[Figure 14. Order 13](#_Toc5814445)

[Figure 15. Payment 13](#_Toc5814446)

[Figure 16. Order Detail 14](#_Toc5814447)

[Figure 17. Order History 14](#_Toc5814448)

[Figure 18. Restaurant manage 15](#_Toc5814449)

[Figure 19. Administrator Page 16](#_Toc5814450)

[Figure 20. Enroll a restaurant 16](#_Toc5814451)

[Figure 21. Restaurant Page 17](#_Toc5814452)

## Project Overview

ThinkFood is a web application where customer can order foods online. Instead of googling each restaurant

to order foods, with ThinkFood, it lets you easily find and order foods in one website with out much hustle.

The feature for the website is to have a menu for each category by foods and restaurant. Make purchases,

view order history and much more. For the administrator, they can add and remove menu from the

website, and as for restaurant owner they receive orders from the administrator.

The objective is to design a website where customers can add, remove, rate, comment, make purchases, and view

history of their orders. As for the Owner users, they can add and remove product, view the order of their customers,

and deliver the food to the customer. For the Restaurant users they can view the orders they received. We will

implement a database and have it all integrate into the website.

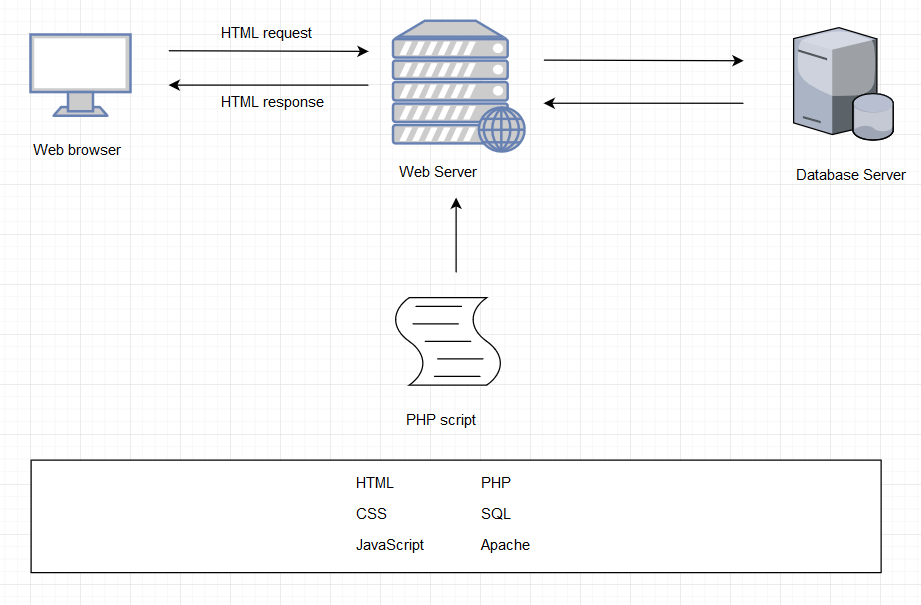


Figure 1. System Architecture

## Project Specifications

### 2.1 Function Requirements

* R01 – The client must be able to log in to an admin account
* R02 – The client must be able to add products to the web site
* R03 – Must be able to display products by categories
* R04 - Customers must register to place orders
* Must provide a fully functioning shopping cart utility where a customer can:
  + R05 – display the current items in the cart
  + R06 - add selected products to the cart
  + R07 - delete products (individually or all) from their cart
  + R08 - adjust the quantity of a selected product currently in the cart
  + R09 - preserve the contents of the cart if the user’s session is disconnected
  + R10 - have the cart accessible regardless from any machine the customer is logging in
  + R11 - Create a unique file for each order that contains the order confirmation information
  + R12 - include a secure, online payment process
* R13 - Allow customers to view their order history

Extra:

* R20 – Allow customers to provide reviews for any of the products they have purchased
* R21 – Must be able to display the evaluation reviews for any products
* R22 – Provide customers with appropriate product recommendations when they log in
* R23 – Provide customers with any product recommendations when they look at a particular product
* R24 – Ask customers to agree with the term condition on their first log in

### 2.2 Non-Function Requirements

* + R12 - include a secure, online payment process
  + R15 - Use free open source software
  + R16 - Have the system fully tested and up and running in 8 weeks

### 2.3 Group Roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **ChangSin Park** | **Qi Li** | **Rolando Pacho** | **Quang Pham** | **Jonghyun Choi** |
| 02 | DBA/Back-End developer | Front-End developer | Middle-ware developer | Front-End developer | Team Lead  Technical Writer |
| 03 | DBA/Back-End developer | Team Lead  Front-End developer | Middle-ware developer | Front-End developer | Technical Writer  DBA/Back-End developer |
| 04 | Front-End developer | DBA/Back-End developer | Team Lead  Front-End developer | Middle-ware developer | Technical Writer  DBA/Back-End developer |
| 05 | Team Lead  Middle-ware developer | DBA/Back-End developer | Technical Writer  Front-End developer | Front-End developer | Middle-ware developer |
| 06 | DBA/Back-End developer | DBA/Back-End developer  Technical Writer | Front-End developer | Team Lead  Front-End developer | Middle-ware developer |
| 07 | Front-End developer | Middle-ware developer | DBA/Back-End developer  Technical Writer | DBA/Back-End developer | Team Lead  Front-End developer |
| 08 | Middle-ware developer | Front-End developer | Front-End developer | Team Lead | Technical Writer |

Table 1. Weekly Roles

## Usability Guide

### Customer

### Register Step 1. Go to register page, and accept or reject the privacy policy agreement.

Figure 2. Privacy Term Accpet

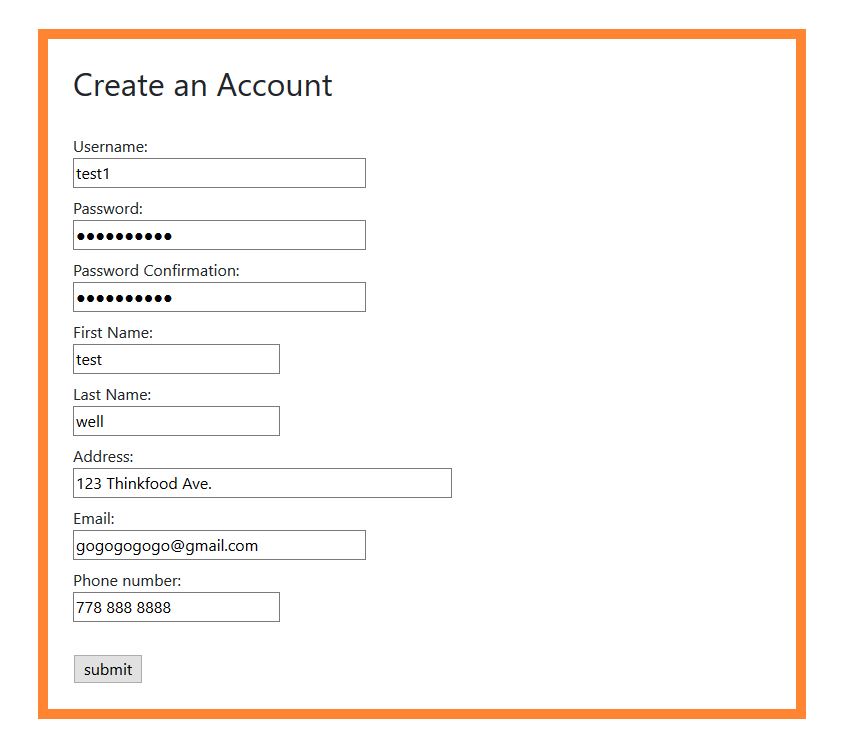
Step 2. fill the form and submit  


Figure 3. Login

### Login Step 1. Go to login page. Step 2. Input username and password, and login.

Figure 4. Login

* + 1. roduct choose

Step 1. go to product page

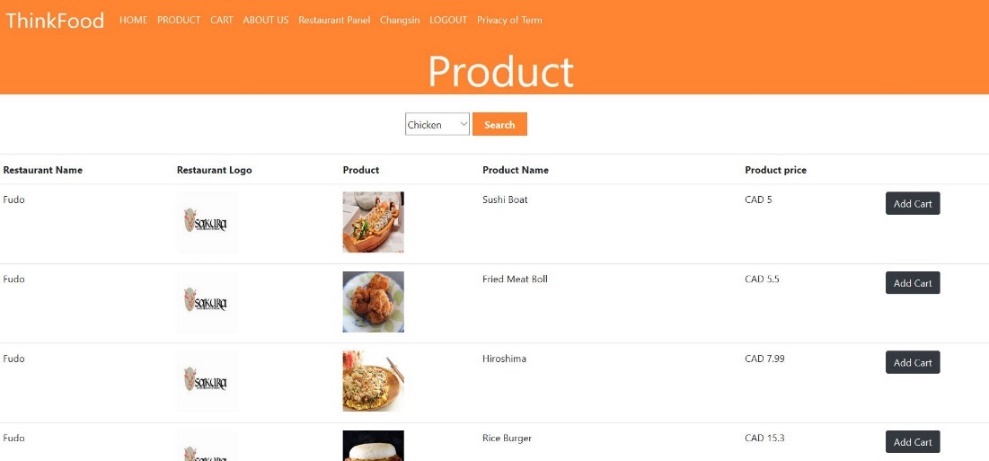


Figure 5. Product

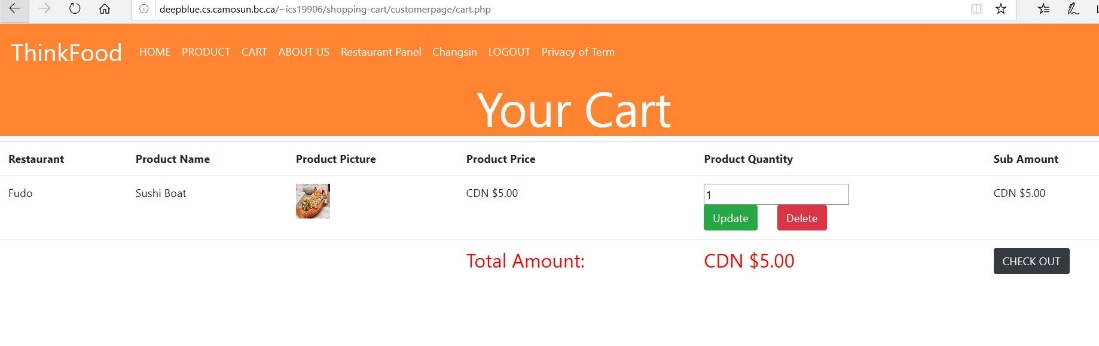
Step 2. add items into cart  
  


Figure 6. Add to cart

* + 1. Review

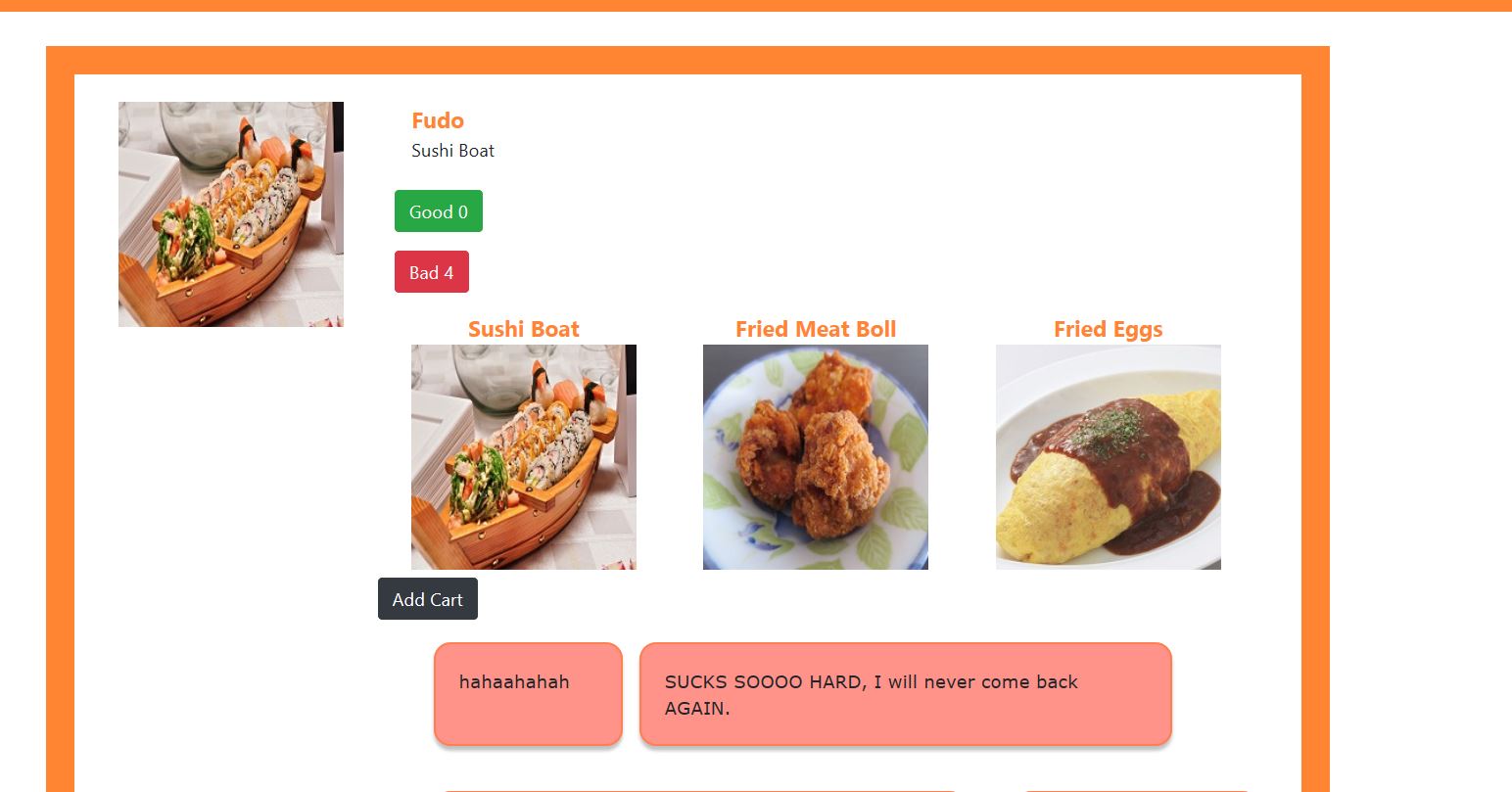
Step 1. Click product image to see the review of any product's reputation.  
  


Figure 7. Product Review

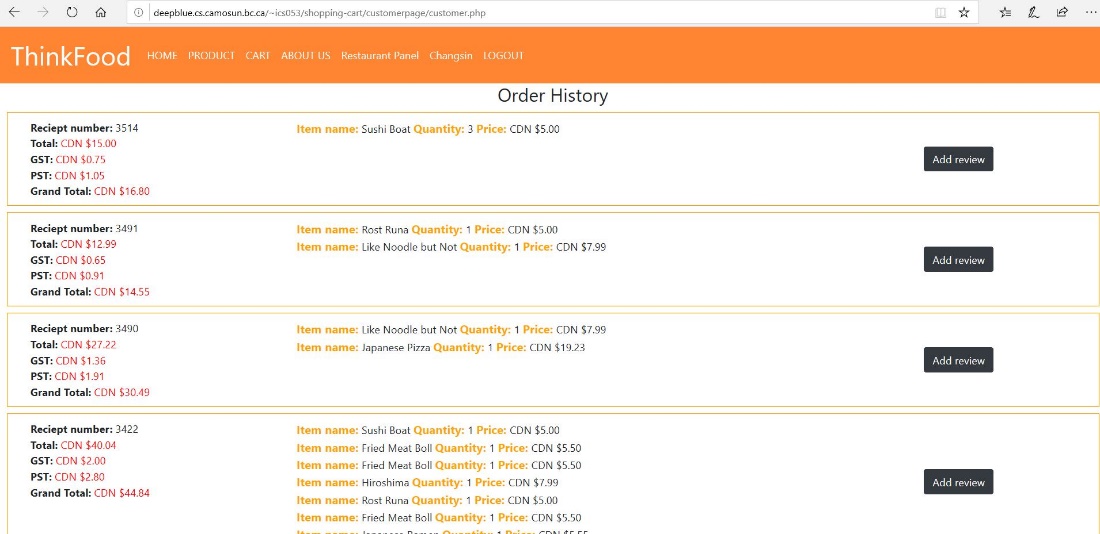
Step 2. After checkout user allowed to go the food what they ordered in their account and rating and reviewing the items.  


Figure 8. Order History



Figure 9. Review

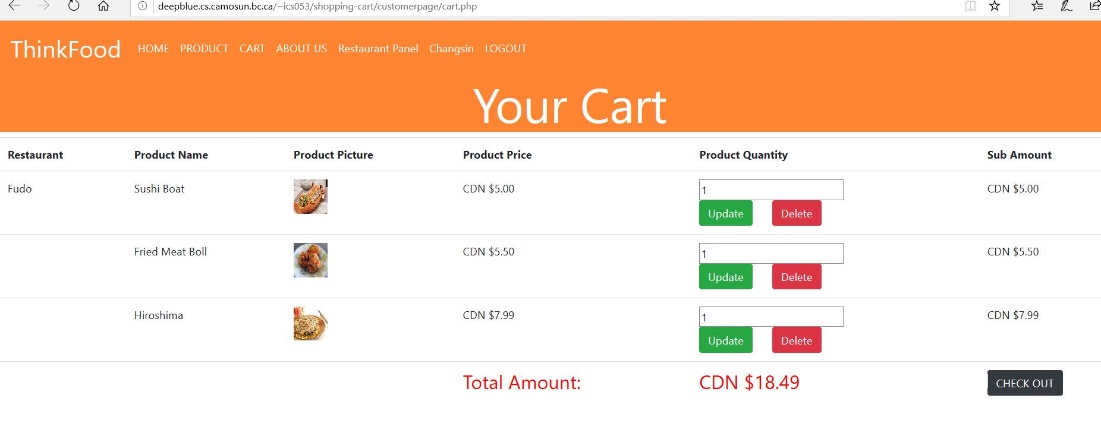
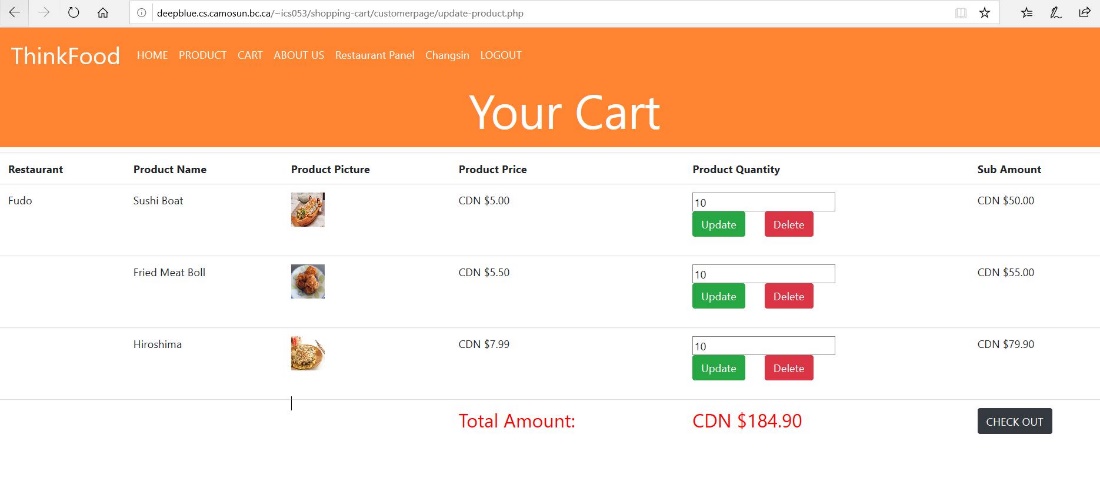
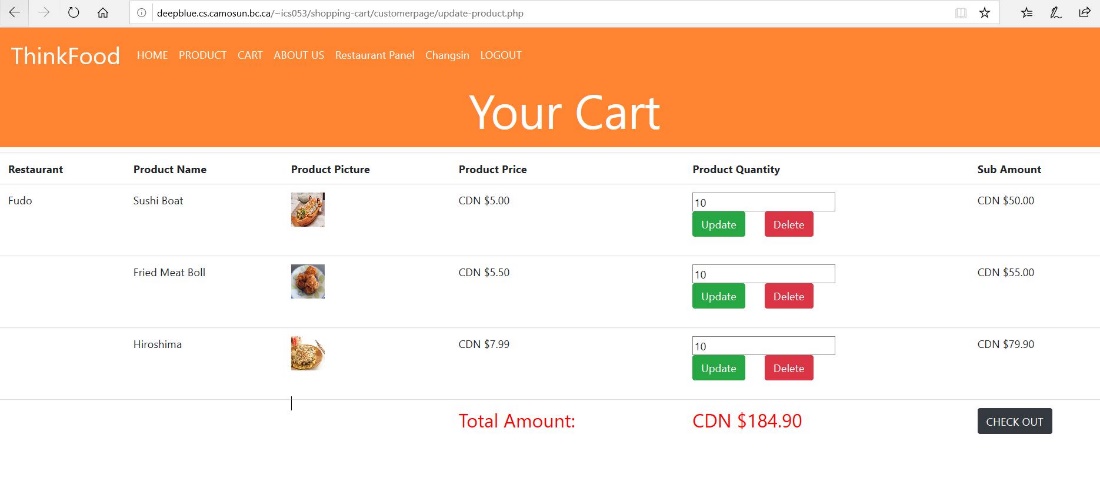
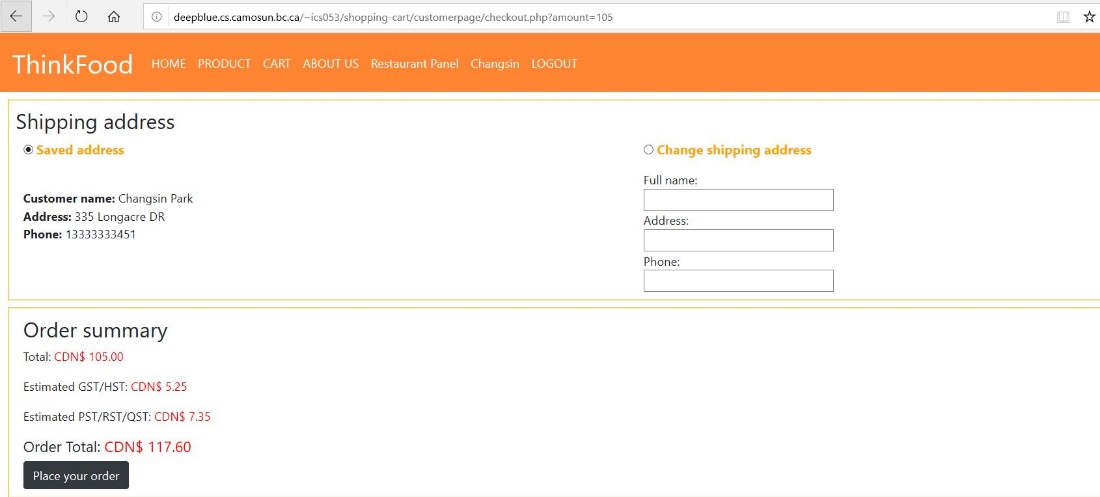
* + 1. **Cart**Step 1. go to cart to check the ordered product

Figure 10. Cart

Step 2. change each items' quantity  
  
  
Figure 11. Cart

Step 3. Update quantity and delete a product  
  
  
Figure 12. Update a cart  
  
   
Step 4. press check out  
  
  
Figure 13. Check out

* + 1. **Check out**

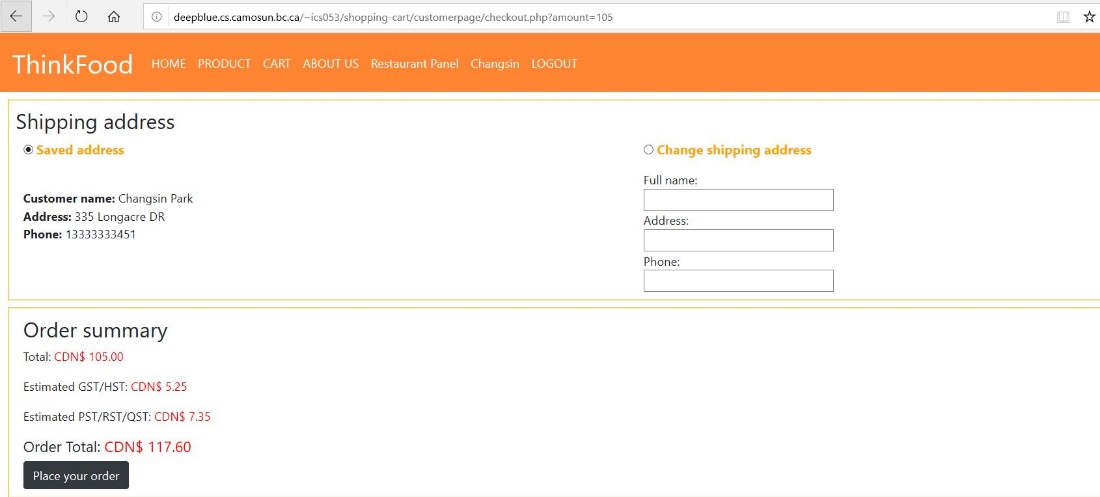
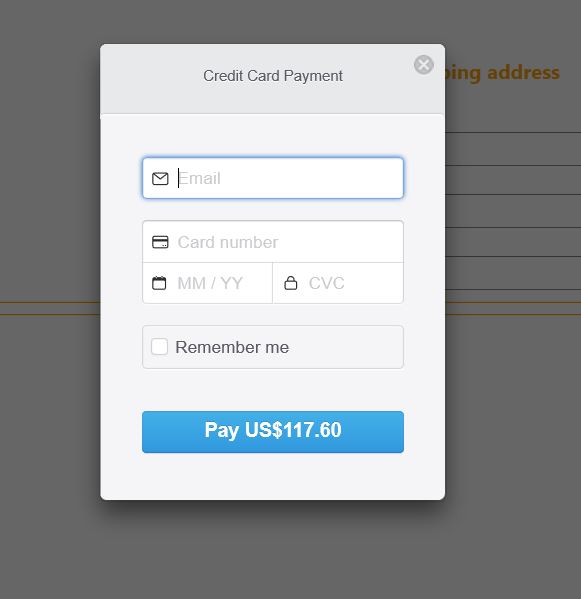
Step 1. in the checkout page users should be able to modify their shipping address and check the info **Figure 14. Order**Step 2. click 'place Your Order' button

Figure 15. Payment

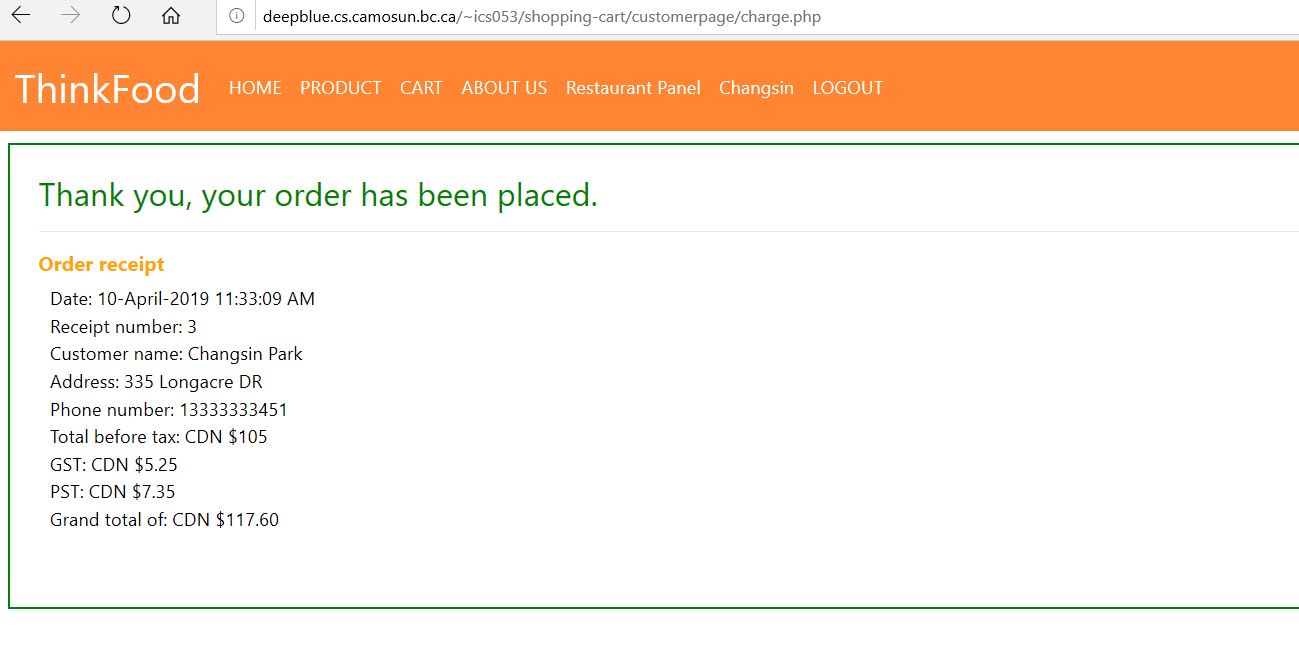
Step 3. after payment, users can go to their account view session, and see their order history  
  


Figure 16. Order Detail

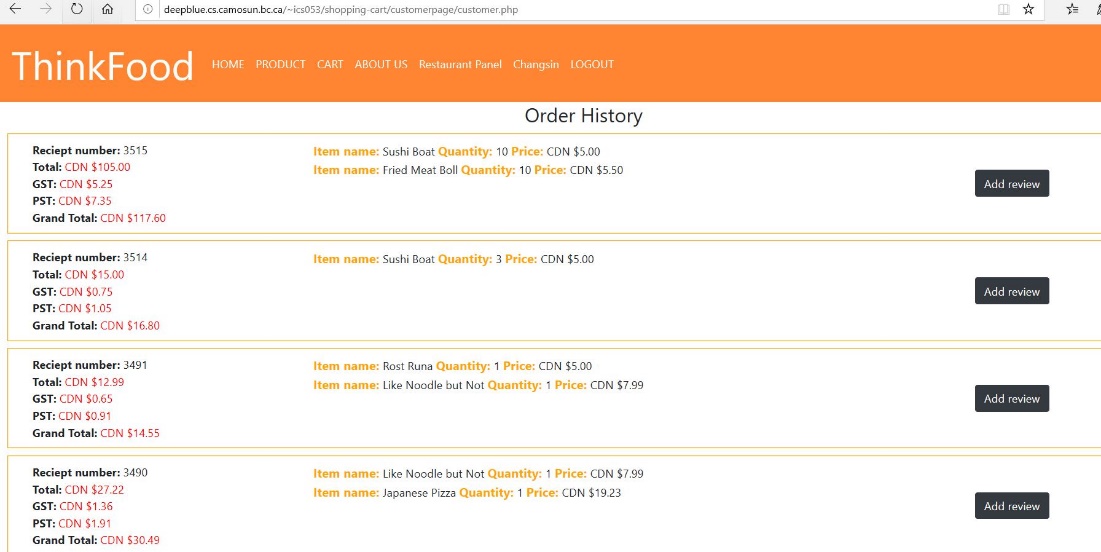
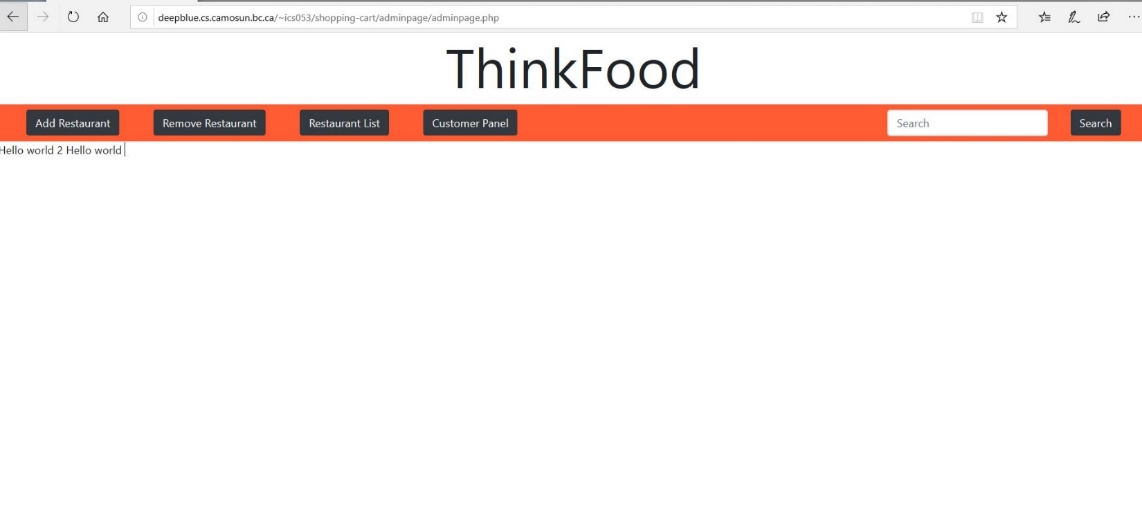


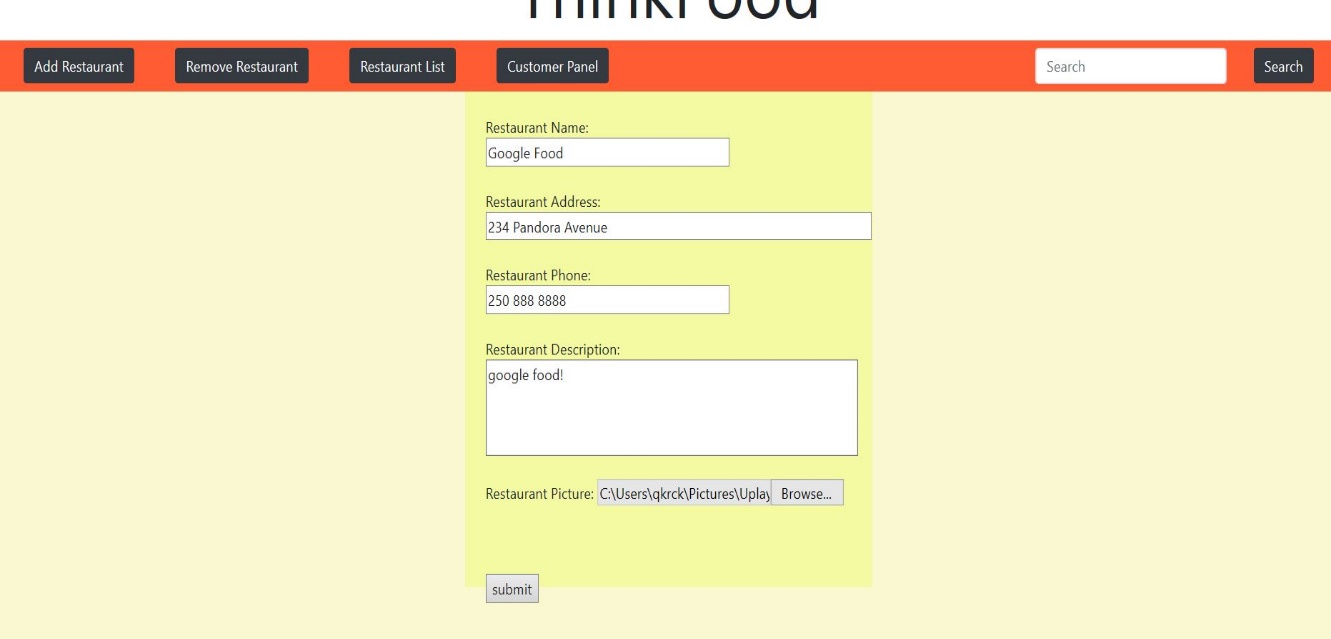
Figure 17. Order History

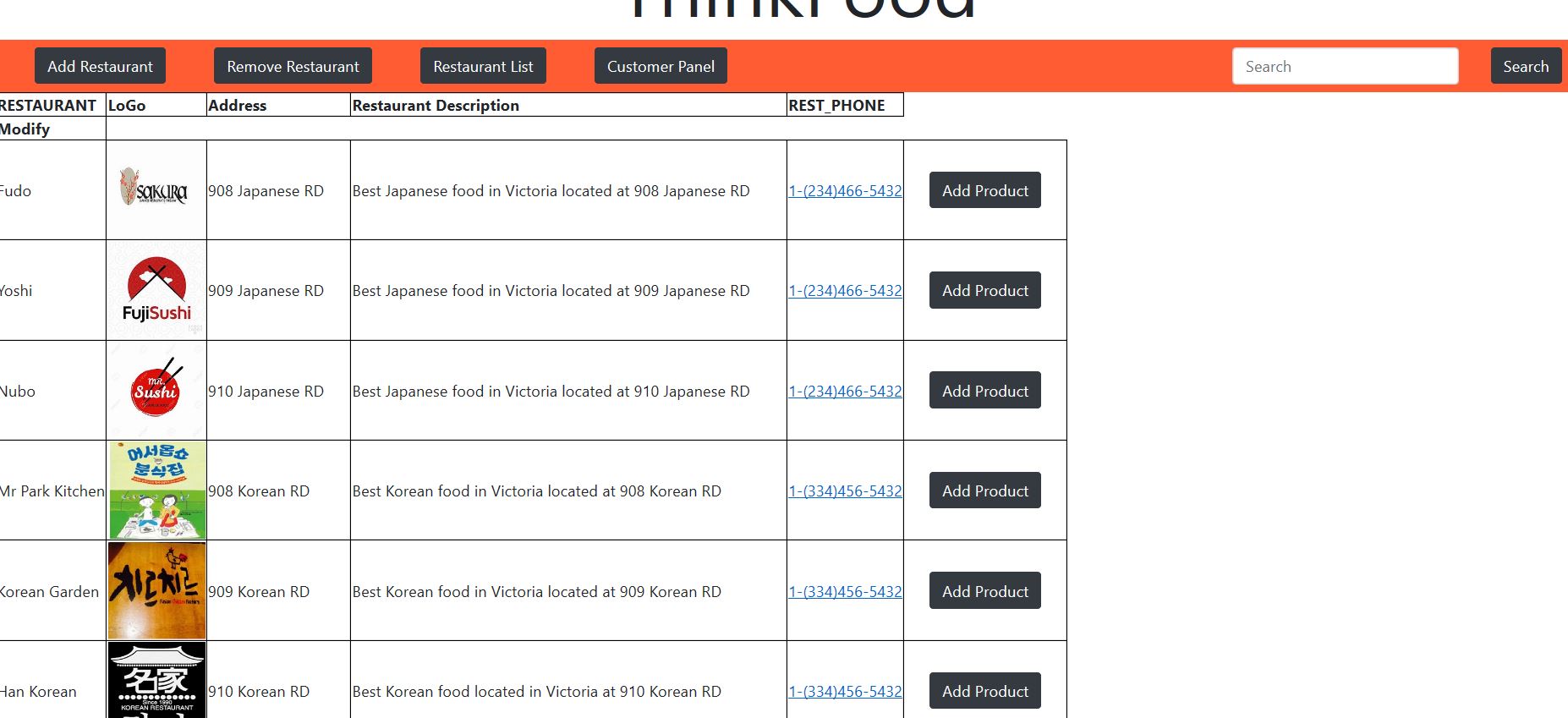
### Restaurant Step 1. once a customer orders confirmed, a client is be able to view all the orders Step 2. by monitoring food status, when the restaurant are ready to deliver their order, it changes status from 'in progress' to 'on the go' Step 3. after delivery is success, the order status changes to 'completed'Step Step 4. when the order delivery completed, restaurant owner should be able to remove the order Figure 18. Restaurant manage

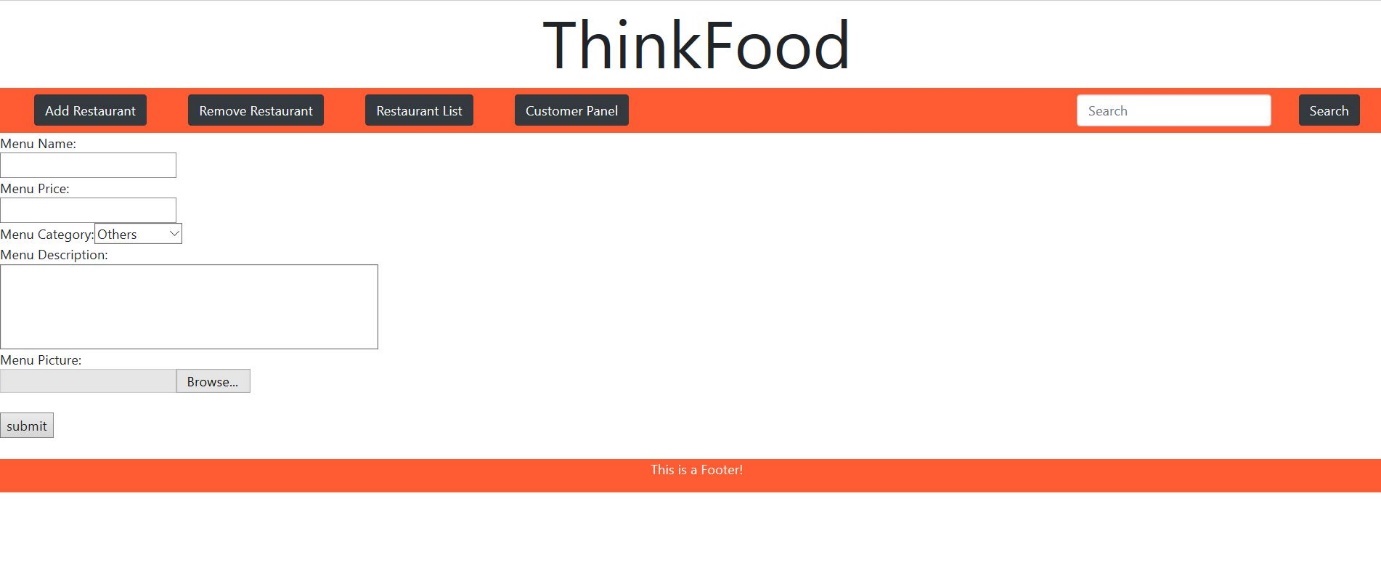
### 

### Administrator

3.3.1 Administrator page  
Step1. After login, you can go to administrator page automatically  
  
  
Figure 19. Administrator Page

3.3.2 Add a restaurant  
Step1. Fill out mandatory items  
  
  
Figure 20. Enroll a restaurant

3.3.3 Add a product  
Step1. Go to restaurant list page and add a product on the restaurant  
  
  
Figure 21. Restaurant Page

Step2. Fill out manadatory items  
  
  
Figure 22. Enroll a product

## Problems and Challenges

### 4.1 Managing tasks on schedule

We faced some problems to manage the schedule for dealyed tasks, missing tasks, new tasks by changement  
 - We handeld it by modifying the priorities of some tasks and changed the schedule.

### 4.2 Standards

We didn’t define naming rules on filename, variable, table & column, so it was hard to understand the others tasks. It was time consuming to us to figure out the sources and documents for deliverables.- We defined the base files for standards and tried to follow the names in those docuemnts.

### 4.3 GitLab

It was hard to be used at managing the resources using GitLab, we lost some code files and faced conflicted files between GitLab maste and deepblue servers.  
- We studied a GitLab and practiced some commands like pull, fetch and push command after making guideline.

### 4.4 Analying requirements

We implemented the sites based on the written requirements but didn’t figure out all of them in detail, so we   
didn’t meet some unexpected checklist on checking by Doug and John.  
- We got the check at the beginning of the class to prepare a missing requirement and immediate changement, so we could make a proper action on time.

## Skill Learned

### 5.1 Cooperation

We had regular meetings to check the tasks’ progress and cooperate. We used Slack for quick communication and set the alarm using it for new changement on our GitLab. We could practice how to work togegher in efficient ways.

### 5.2 GitLab

Using a gitlab encourged us to manage our each sources consistent. We used only one way from our GitLab   
master to deepblue server whenever it come to deploy new versions of sources for version management.

### 5.3 MySQL and Workbench

We learned Oracle database and SQL Developer during past database classes. One of the non-functional requirement of this project is using an open source software, so we used MySQL for our data. The syntax is very  
similar with Oracle, We also used MySQL Workbench for modeling and reverse , forward engineer function for extracting metadata and creating objects each.

### 5.4 PHP, HTML, CSS

We could review web technologies such HTML, CSS and PHP through this project. Bootstrap is used for   
predefined styles for UI. PHP is used for server side implementation between user interface and database.  
It was a meaningful experience to us to practice all knowledge we learned at 1st year.

## Proposed Changes

### 6.1 Time management

We will focus on planning to assign resources and tasks based on the priority with enough time in case some tasks are delayed by technical issues and requirement changement. This will be helpful to finish the projec on time.

### 6.2 Dedicated role assign

We had some difficulties when we switch to the other roles every weekend. We could learn from the others’ tasks and practice throug roles. But it was time consuming to explain my taks to team members. It will be more efficient to assign one team member on on role.

### 6.3 Avoid duplicate statements

### 6.4 Avoid duplicate statements

### 6.5 Avoid duplicate statements

## Code Snippets

|  |  |
| --- | --- |
| 7.1 Batch for recommend products  |  | | --- | | [ics19906@deepblue ~]$ crontab -l  00 \* \* \* \* sh /home/student/ics19906/script/batch01.sh  CREATE DEFINER=`ICS199Group06\_prod`@`%` PROCEDURE `SP\_RECOMMEND\_PRODUCT`()  BEGIN  UPDATE PRODUCT SET RECOMMEND\_YN = 'N';  UPDATE PRODUCT SET RECOMMEND\_YN = 'Y'  WHERE PRODUCT\_ID IN (  SELECT AA.PRODUCT\_ID  FROM (  SELECT A.PRODUCT\_ID, A.RATE\_POINT, A.PRODUCT\_CATEGORY  , (CASE @vproduct\_category WHEN A.PRODUCT\_CATEGORY  THEN @rownum:=@rownum+1 ELSE @rownum:=1 END) RNUM  , (@vproduct\_category:=A.PRODUCT\_CATEGORY) CATEGORY  FROM (  SELECT P.PRODUCT\_ID,  SUM(S.QUANTITY \* CASE S.RATE\_POINT  WHEN 1 THEN -1 ELSE 1 END) RATE\_POINT,  P.PRODUCT\_CATEGORY  FROM ORDERINFO O, SOLD\_PRODUCT S, PRODUCT P  WHERE O.ORDER\_STATUS = '03'  AND O.ORDER\_ID = S.ORDER\_ID  AND S.PRODUCT\_ID = P.PRODUCT\_ID  GROUP BY P.PRODUCT\_ID, P.PRODUCT\_CATEGORY  ) A , (SELECT @vproduct\_category:=0, @rownum:=0 FROM DUAL) B  ORDER BY A.PRODUCT\_CATEGORY ASC, A.RATE\_POINT DESC  ) AA  WHERE AA.RNUM < 4  );  INSERT INTO LOG VALUES('01',NOW());  END | |

### 7.2 Add product to cartt

|  |
| --- |
| // 03: same CUST add same product at same restaurant  // 02: same CUST add diff product at same restaurant  // 01: same CUST add diff product at diff restaruant OR first add in to cart    $query = "SELECT MAX(K.KIND) kind FROM (  SELECT '03' KIND FROM ORDERINFO O, SOLD\_PRODUCT S  WHERE O.ORDER\_ID = S.ORDER\_ID  AND O.REST\_ID = $rest\_id AND O.CUST\_ID = $cust\_id  AND O.ORDER\_STATUS = '05 AND S.PRODUCT\_ID = $product\_id  UNION ALL  SELECT '02' KIND FROM ORDERINFO O, SOLD\_PRODUCT S  WHERE O.ORDER\_ID = S.ORDER\_ID  AND O.REST\_ID = $rest\_id AND O.CUST\_ID = $cust\_id  AND O.ORDER\_STATUS = '05'  UNION ALL  SELECT '01' FROM dual  ) K;";  if ($row['kind'] == '03') {  $query\_03\_01 = "UPDATE SOLD\_PRODUCT SET QUANTITY = QUANTITY + 1  WHERE PRODUCT\_ID = $product\_id  AND ORDER\_ID = (SELECT ORDER\_ID  FROM ORDERINFO  WHERE ORDER\_STATUS='05'  AND CUST\_ID = $cust\_id AND REST\_ID = $rest\_id);";  else if ($row['kind'] == '02') {  $query\_02\_02 = "INSERT INTO SOLD\_PRODUCT (PRODUCT\_ID, ORDER\_ID, QUANTITY) VALUES  ($product\_id, (SELECT ORDER\_ID FROM ORDERINFO where ORDER\_STATUS='05'  and cust\_id = $cust\_id AND REST\_ID = $rest\_id), 1);";  $result\_02\_02 = mysqli\_query($connectData, $query\_02\_02);  else {  $query\_01\_01 = "INSERT INTO ORDERINFO (ORDER\_STATUS, CUST\_ID, REST\_ID) VALUES ('05', $cust\_id, $rest\_id);";  $result\_01\_01 = mysqli\_query($connectData, $query\_01\_01);  $query\_01\_02 = "INSERT INTO SOLD\_PRODUCT (PRODUCT\_ID, ORDER\_ID, QUANTITY) VALUES  ($product\_id, (SELECT ORDER\_ID ORDER\_ID FROM ORDERINFO where ORDER\_STATUS='05'  and cust\_id = $cust\_id AND REST\_ID = $rest\_id), 1);";  $result\_01\_02 = mysqli\_query($connectData, $query\_01\_02);  } |

### 7.3 ppppppp