|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject Code** | CS5281 | | |  |
| **Subject Title** | Internet Application Development | | |  |
| **Semester** | B, 2017/18 | | |  |
| **Assignment Title** | System Analysis and Design Report | | |  |
| **Team Name** | Unicorn | | |  |
| **Student(s) Name & ID** | PENG CHENGCHENG (5492 6938) – Team Leader | | |  |
|  | GARAJ, Martin (5494 5178) | | |  |
|  | JIANG CHENGYI (5488 5190) | | |  |
|  | JIANG NINGZI (5517 0317) | | |  |
|  | LI CHI YEUNG, Ken (5047 6460) | | |  |
|  | LIU YUNSONG (5512 0338) | | |  |
|  | WANG ZISHEN (5507 6544) | | |  |
| **Submit to** | Dr. Kenneth KC LEE | **Dept** | CS | | |
| **Date of Submission** | Apr 24th, 2018 | **Total**  **no. of pages** | 90 | | |



TABLE OF CONTENTS

TABLE OF CONTENTS 2

1. Background 3

2. System Infrastructure 3

3. Functional Flow 4

4. Data Flow 5

5. Security 5

6. Assumption 6

7. Main Functional Flow Description 6

8. Function Specification of Search Dish and Food Recommendation 18

9. Function Specification of Place Order 21

10. Function Specification of Comment 33

11. Function Specification of Login and Logout 35

12. Function Specification of Sign Up 38

13. Function Specification of User Profile 44

14. Function Specification of Notification 51

15. Function Specification of Contact Us 52

16. Function Specification of Add Food 53

17. Exception Handling 56

18. Different CSS Style for Web Page and Printing 57

19. Cookie Management 57

20. AJAX Implementation 58

21. Database Design 58

23. Browser Capability Testing 63

24. Functional Testing 65

25. System Configuration 82

26. Environment Setup 82

27. Application Deployment 86

28. Source Control 87

APPENDICES 88

Appendix I Browser Capability Testing Result 88

Appendix II Functional Testing Report and Bug Fixing Result 88

Appendix III Development Guide and Reference 88

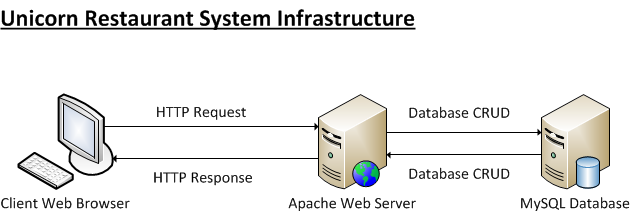
References 89

# 1. Background

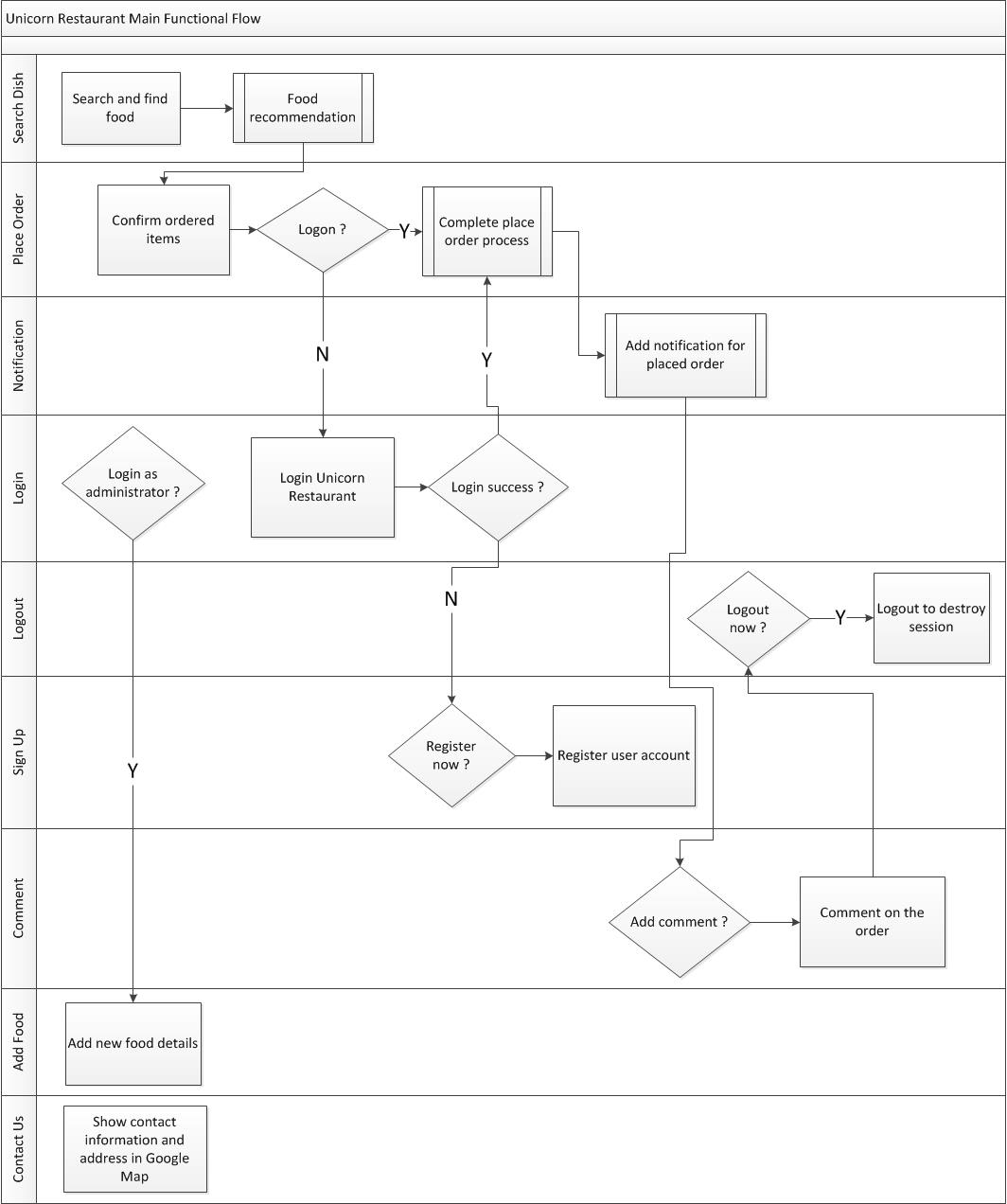
Unicorn restaurant is a food ordering web-based application system which is implemented by using HTML, CSS, PHP and JavaScript technologies. There are 8 core functions provided by the system included,

* Search Dish and Food Recommendation
* Place Order
* Comment
* Login / Logout
* Sign Up
* User Profile
* Notification
* Contact Us
* Add Food (Administrative function)

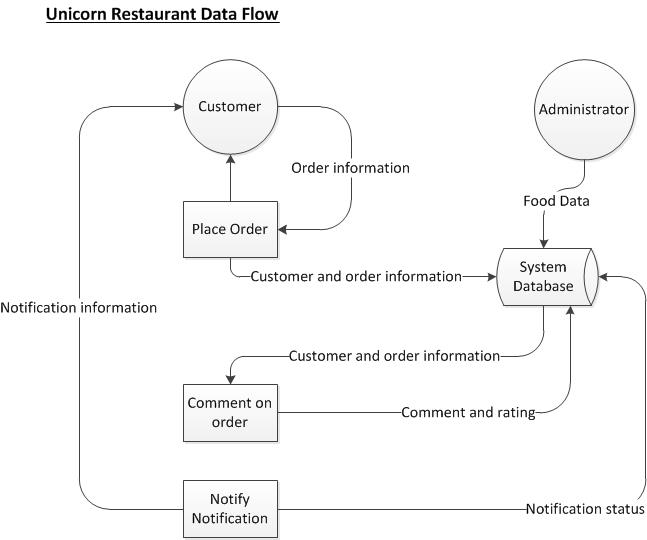
# 2. System Infrastructure



# 3. Functional Flow



# 4. Data Flow



# 5. Security

User account password will be encrypted by md5 function and store in MySQL database instead of storing plaintext password in database.

Client side and server side validation rules will be applied to all input fields in online web functions. Client side validation is implemented by JavaScript and server side validation is implemented by PHP. In case JavaScript is disabled in customer web browser, PHP server side validation can make sure the input field data format is corrected and protect the system from cross-site scripting hacking.

When JavaScript is disabled in customer web browser, an exception page will be shown and user must enable JavaScript for accessing Unicorn Restaurant web application system.

The system makes use of PHP PDO prepared statement for all MySQL database manipulation to prevent SQL injection.

Besides, login session will be automatically logout if there is no user interaction after 30 minutes.

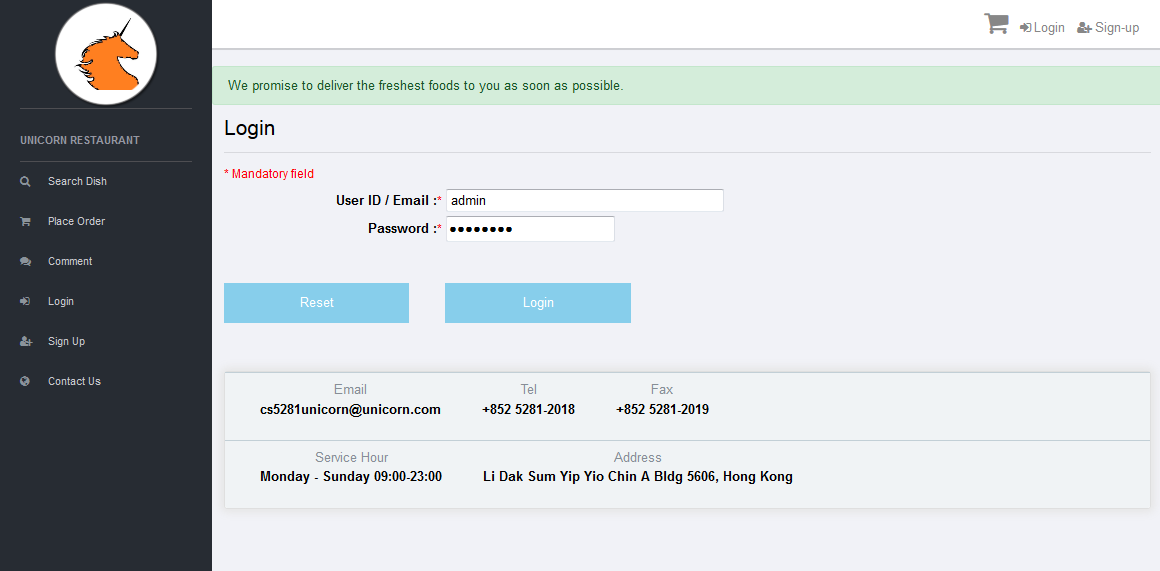
# 6. Assumption

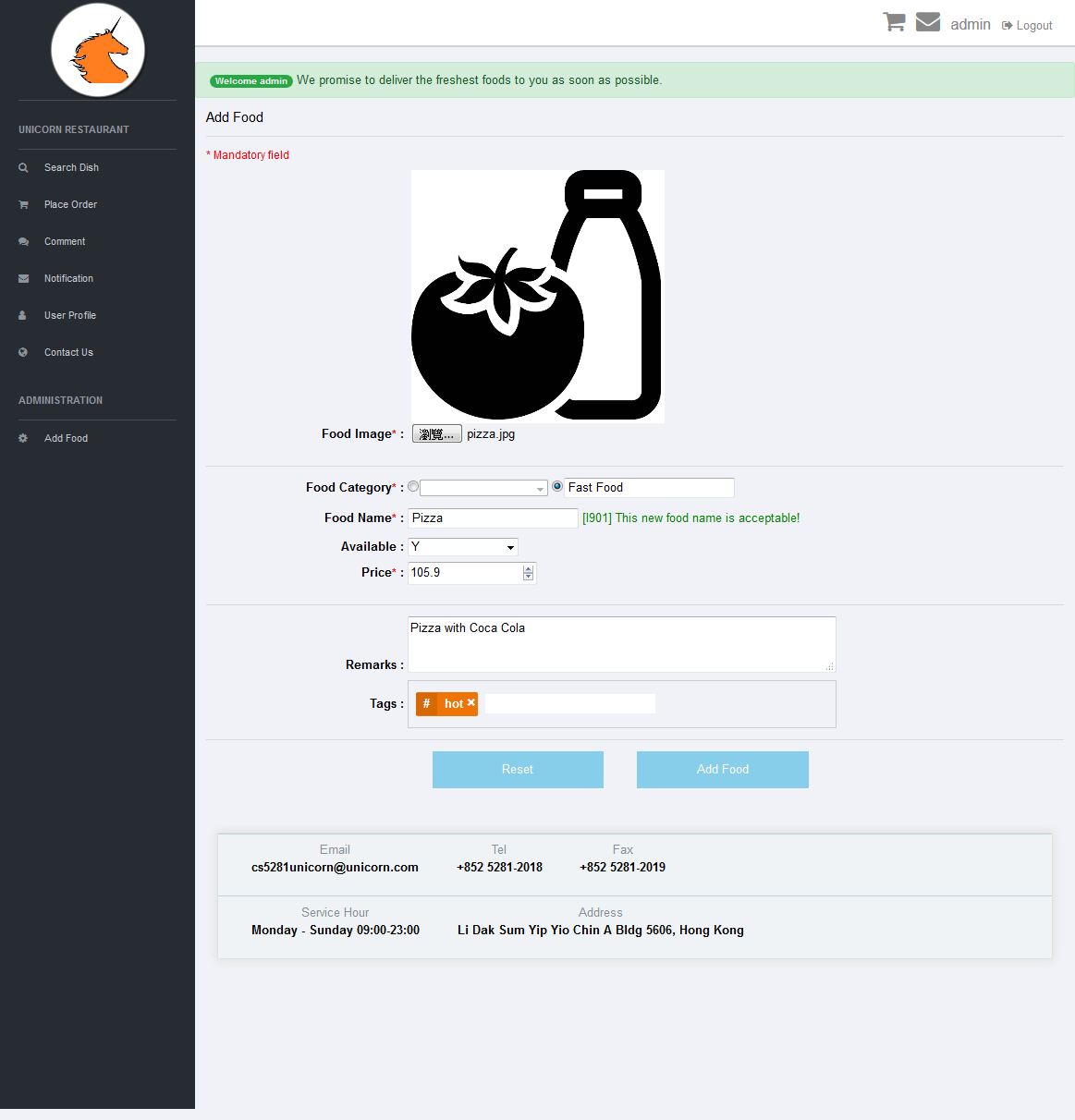
There are 7 assumptions for the system included,

* The system is deployed and running in a standalone workstation.
* Assume the concurrent user access will NOT greater than 2 people.
* Assume the http response time which is less or equal than 4 seconds is acceptable to user.
* User use Firefox, Chrome and Internet Explorer web browser to access Unicorn Restaurant web application.
* Assume customer web browser is enabled JavaScript for access full function of Unicorn Restaurant web application.
* Assume customer web browser allow pop-up message box, otherwise alert message box will be disabled.
* Assume customer web browser allow cookie storage, otherwise some function based on cookie will be disabled.

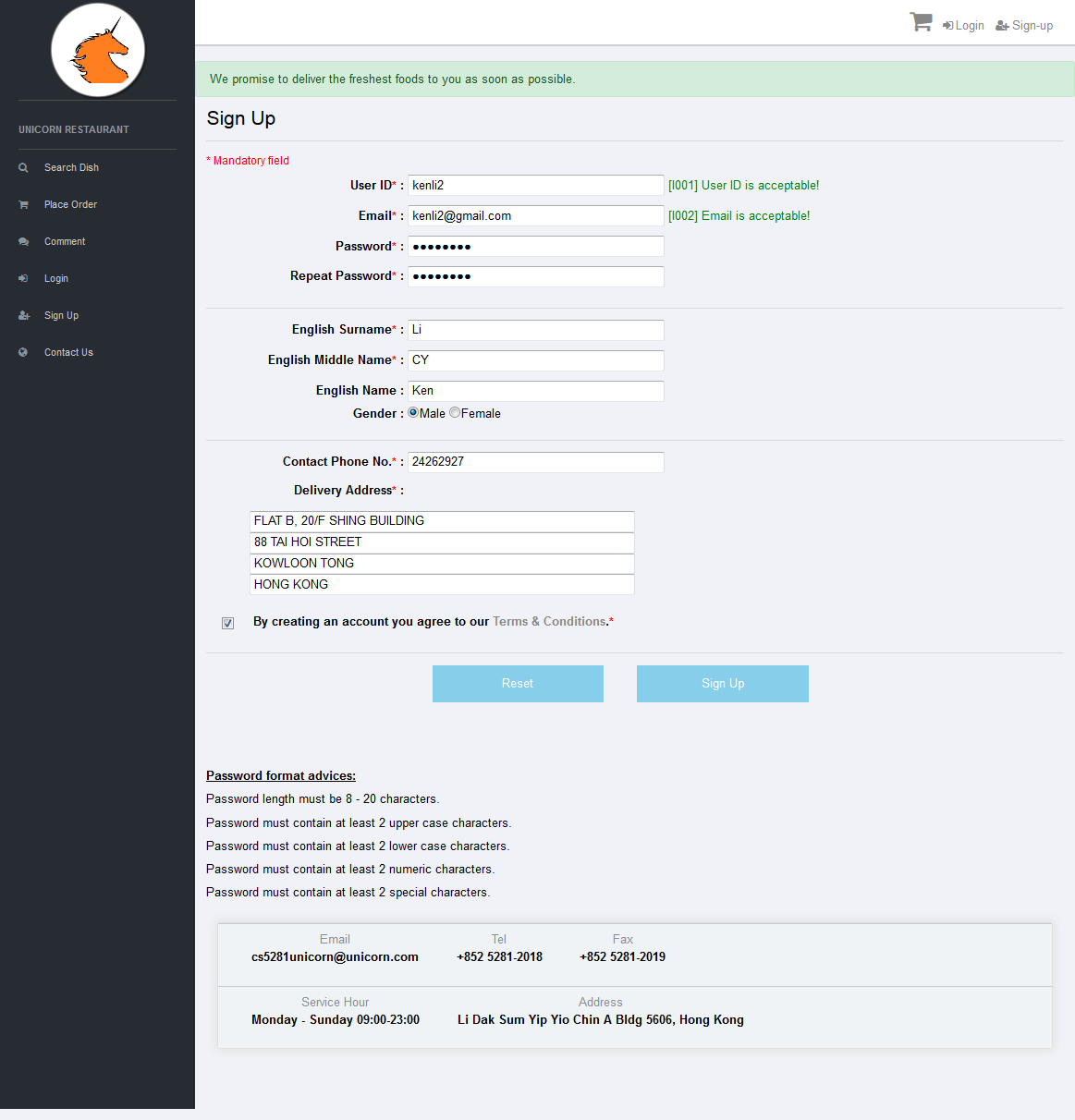
# 7. Main Functional Flow Description

7.1 Login as administrator to add food

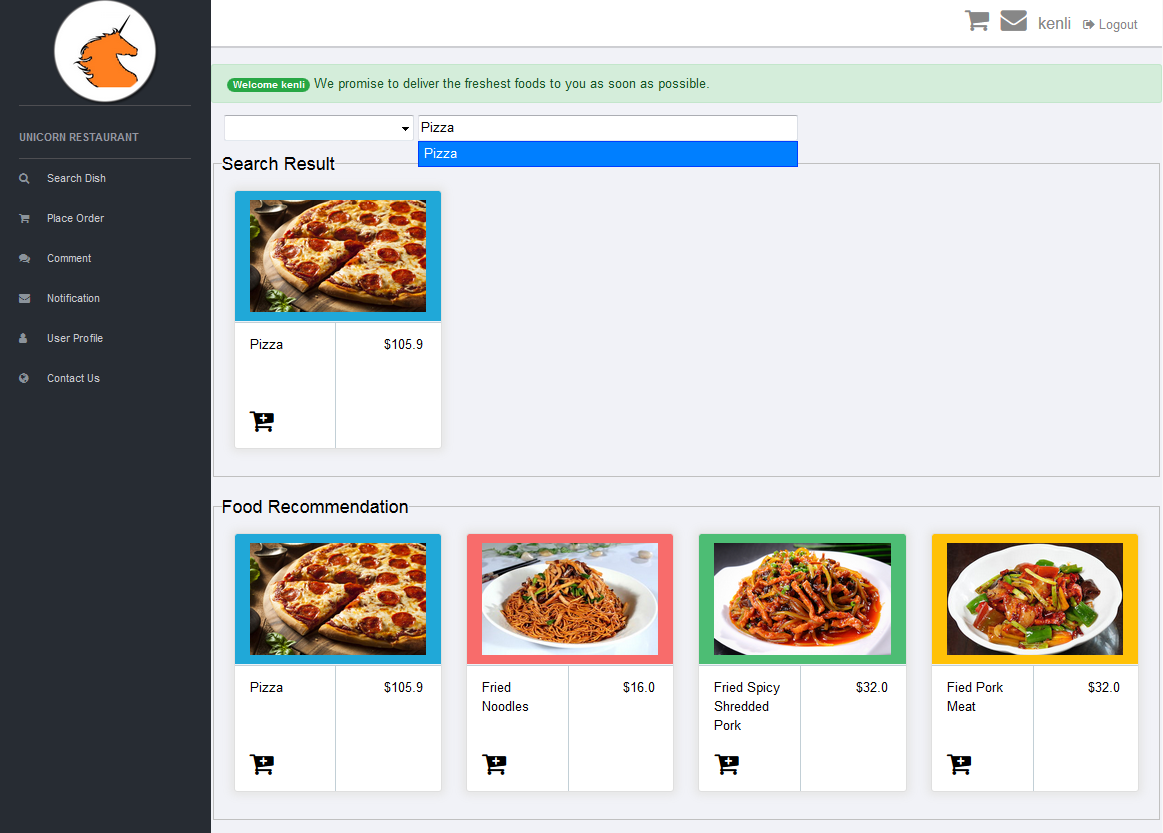


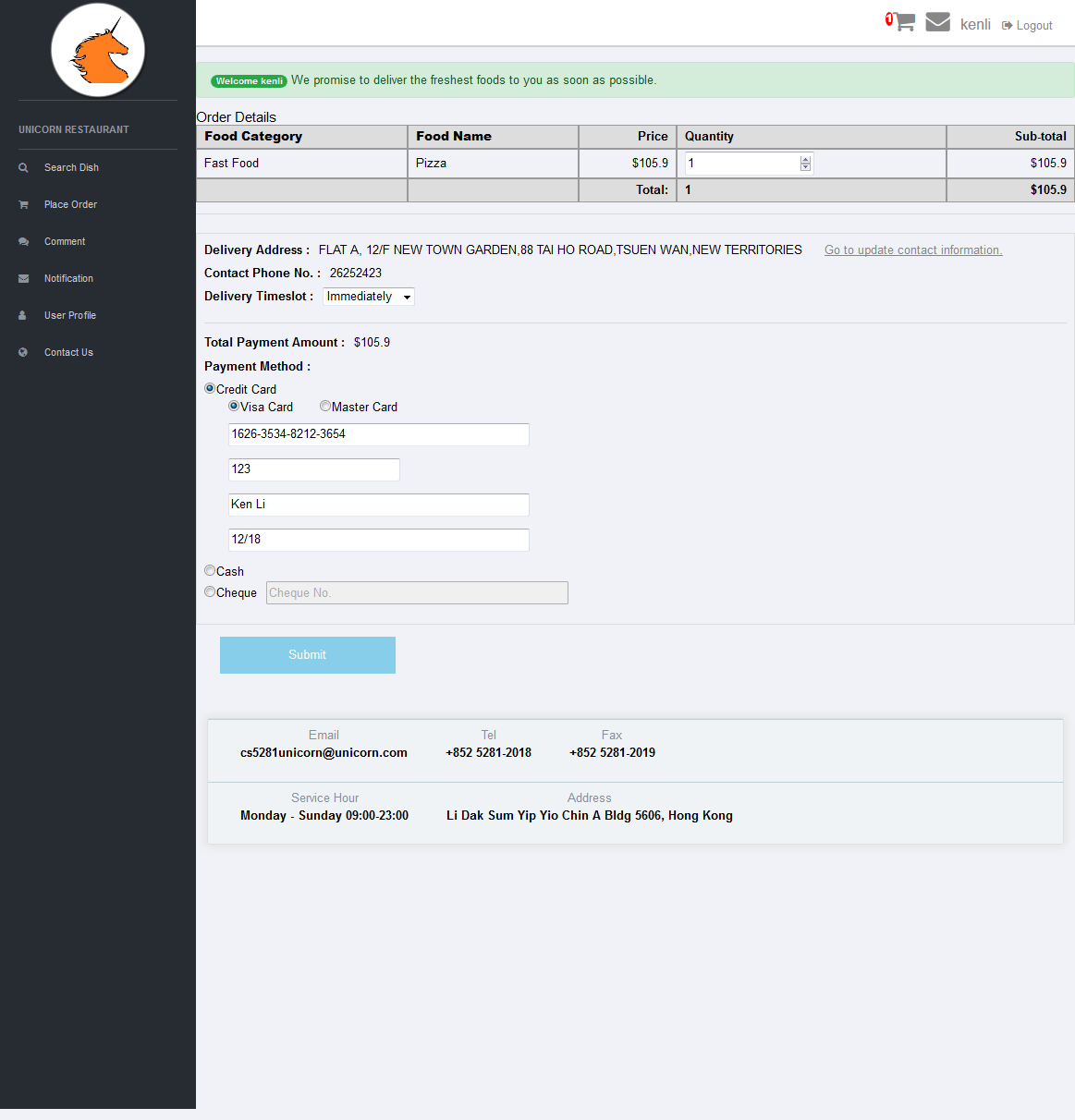


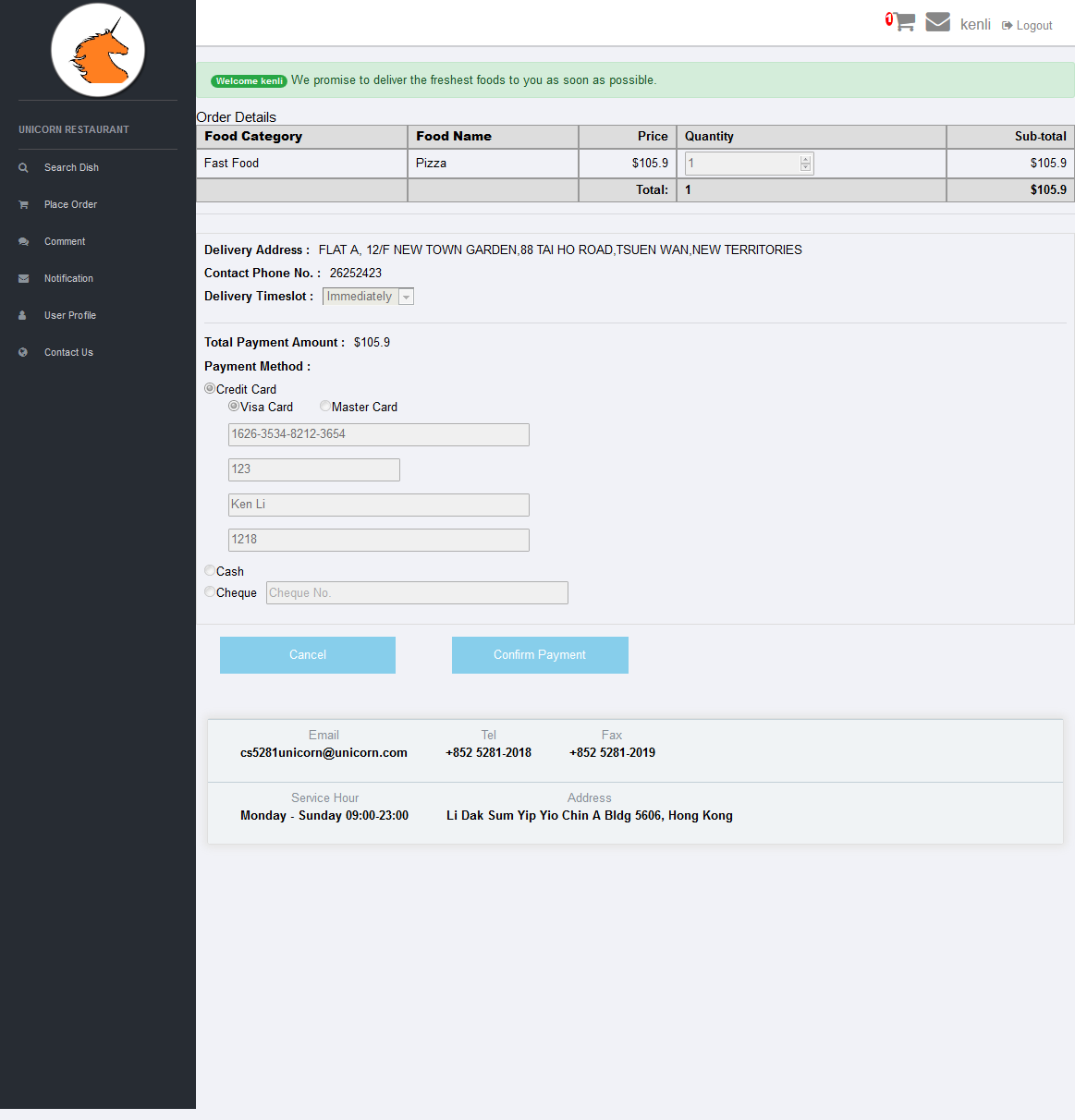
7.2 Sign up and activate user account

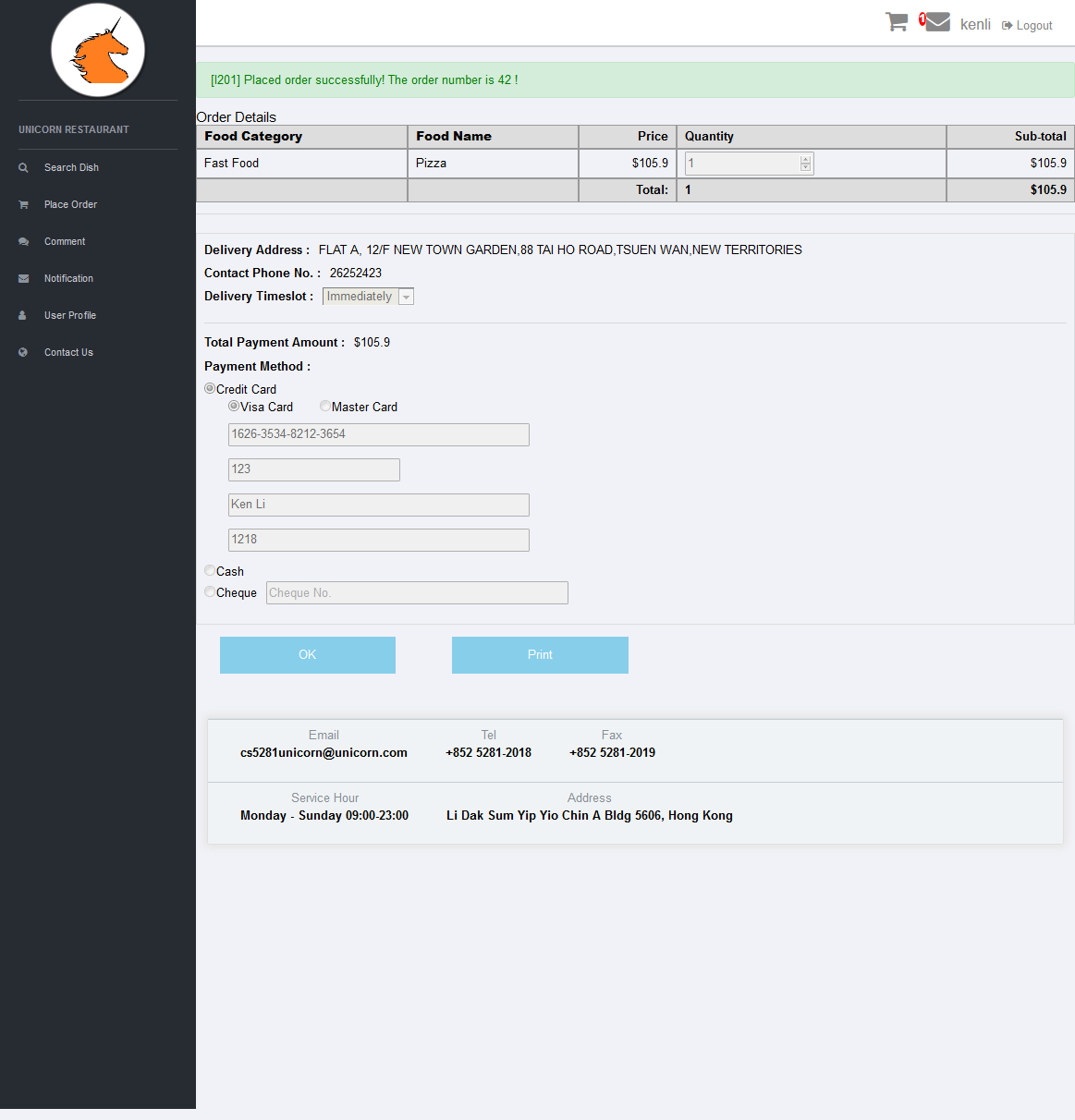


7.3 Login as customer to search food and place order

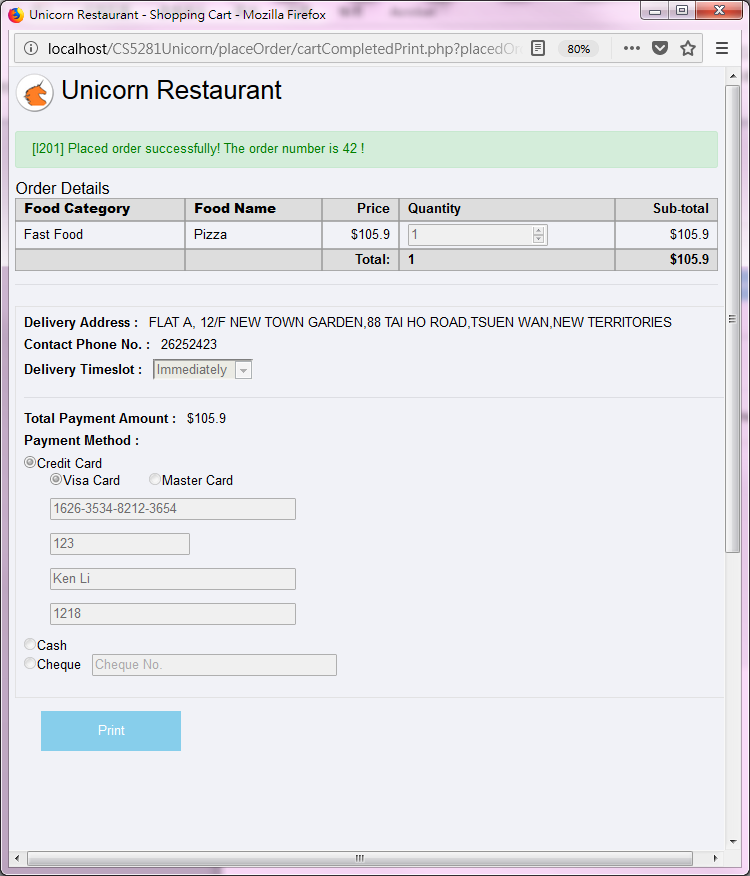


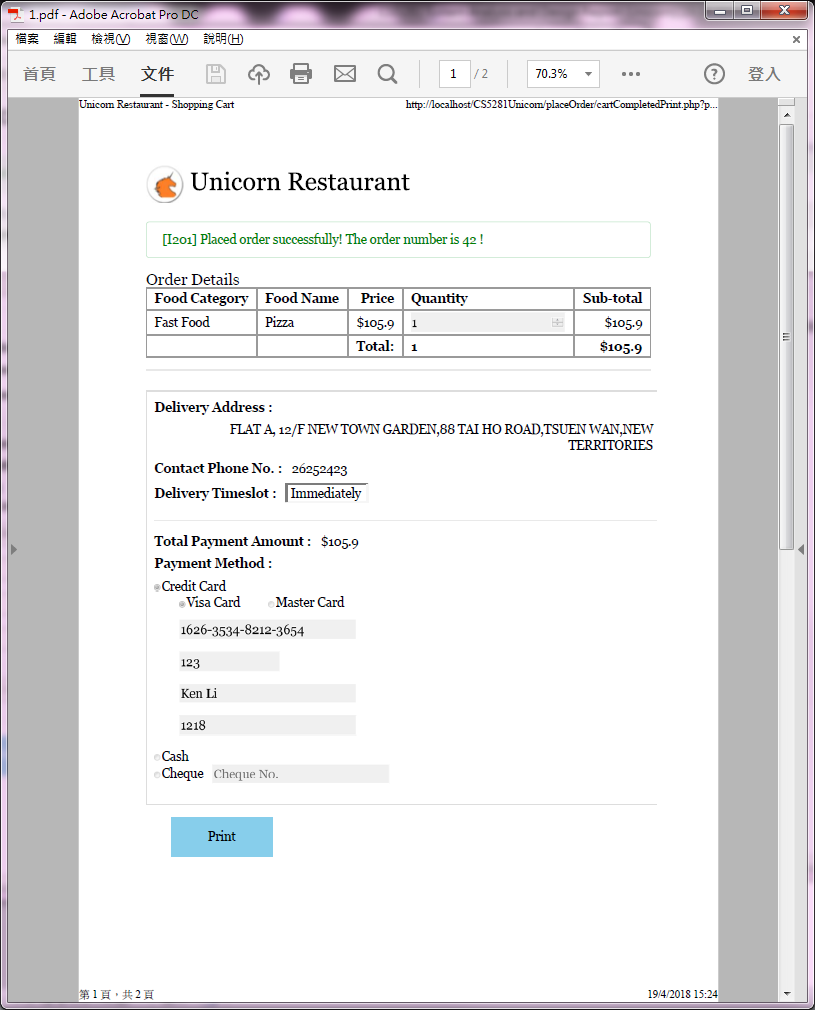




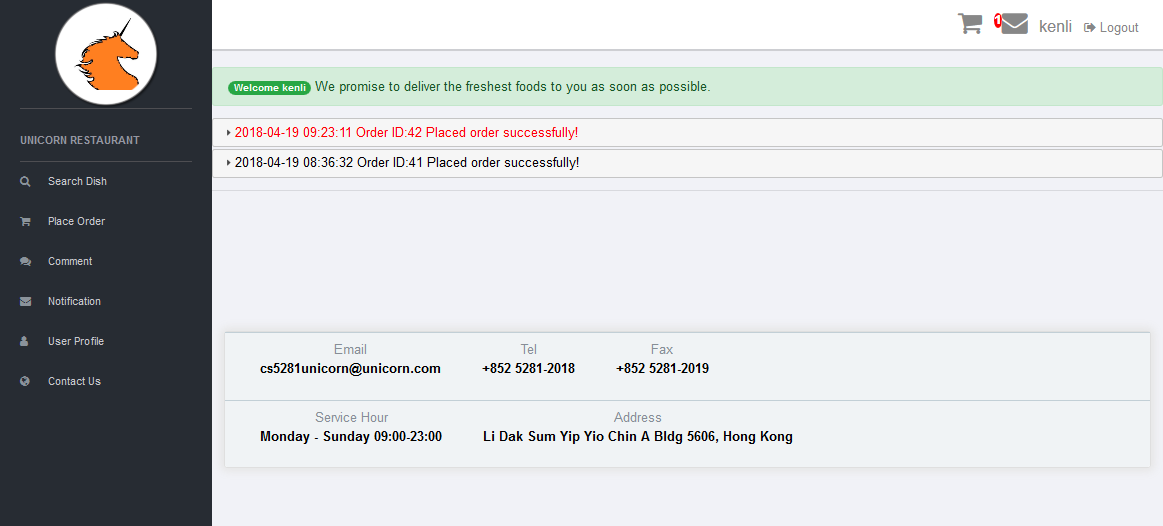


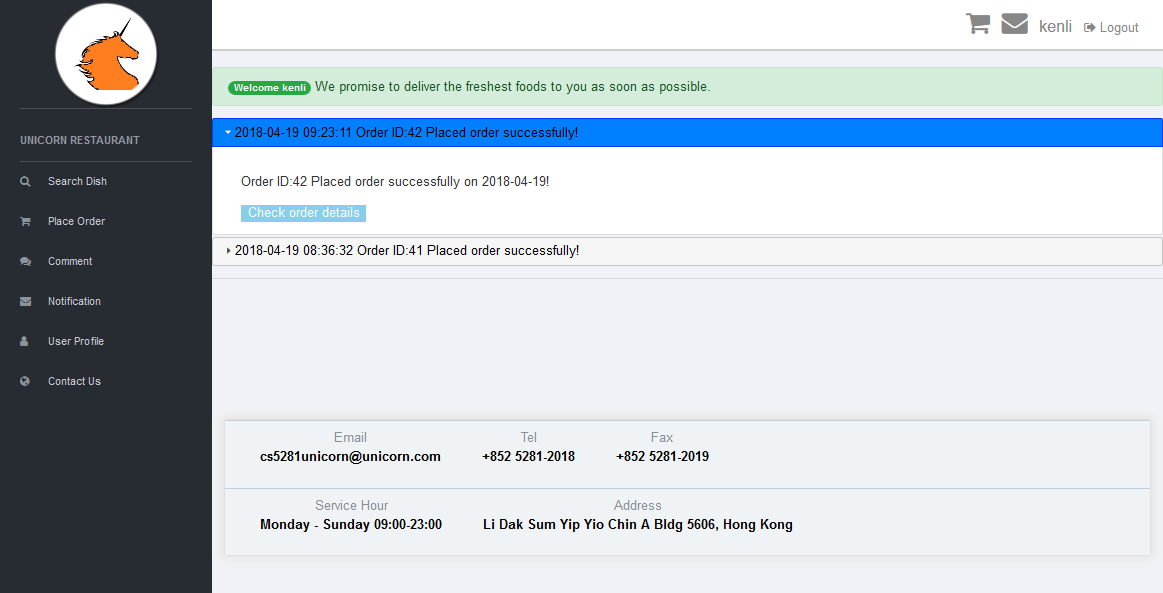
7.4 Print the order confirmation letter



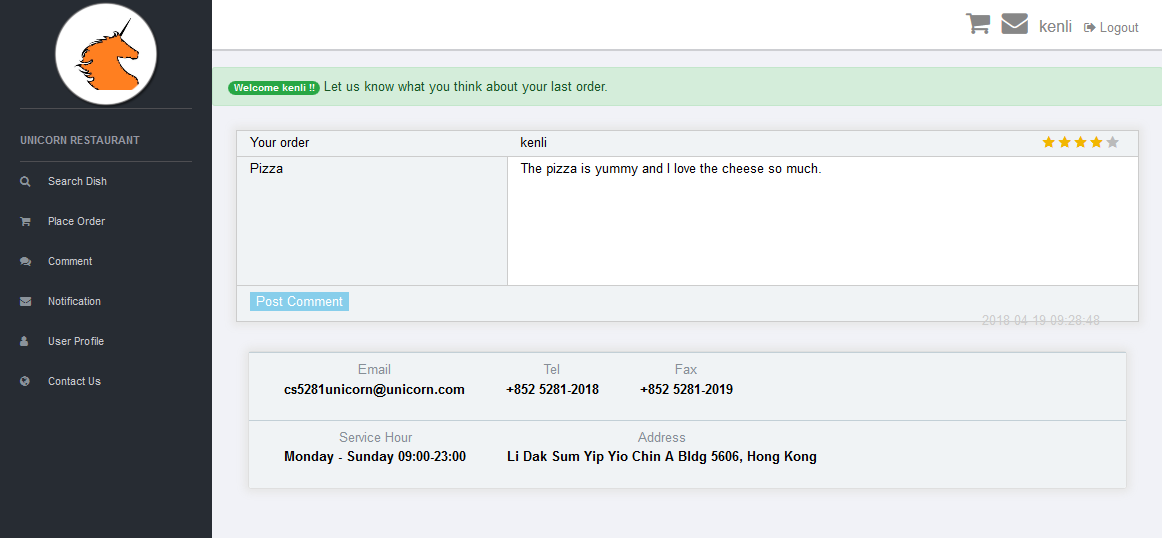


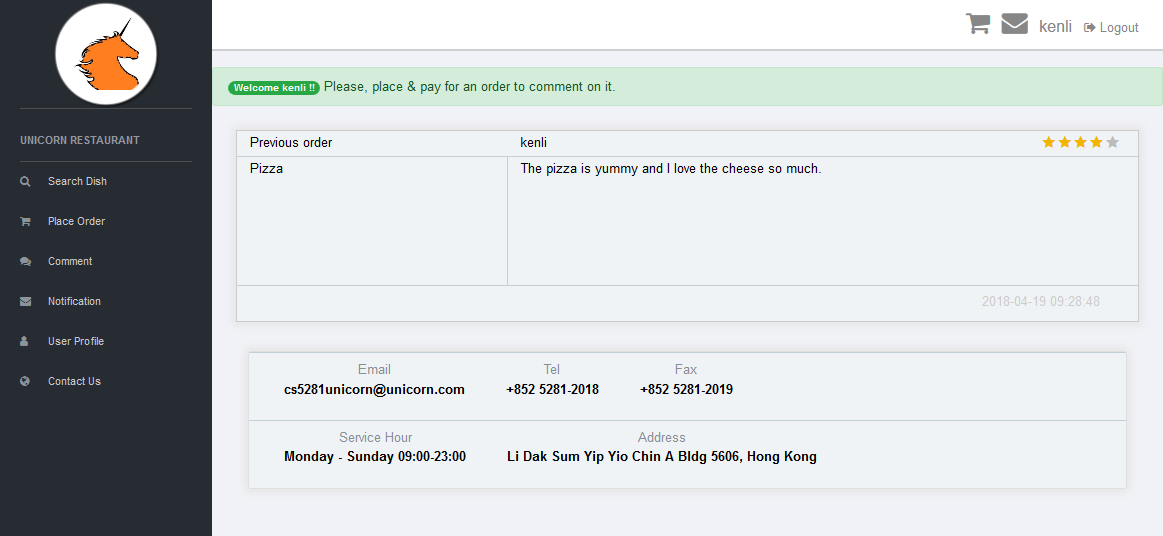
7.5 Check notification



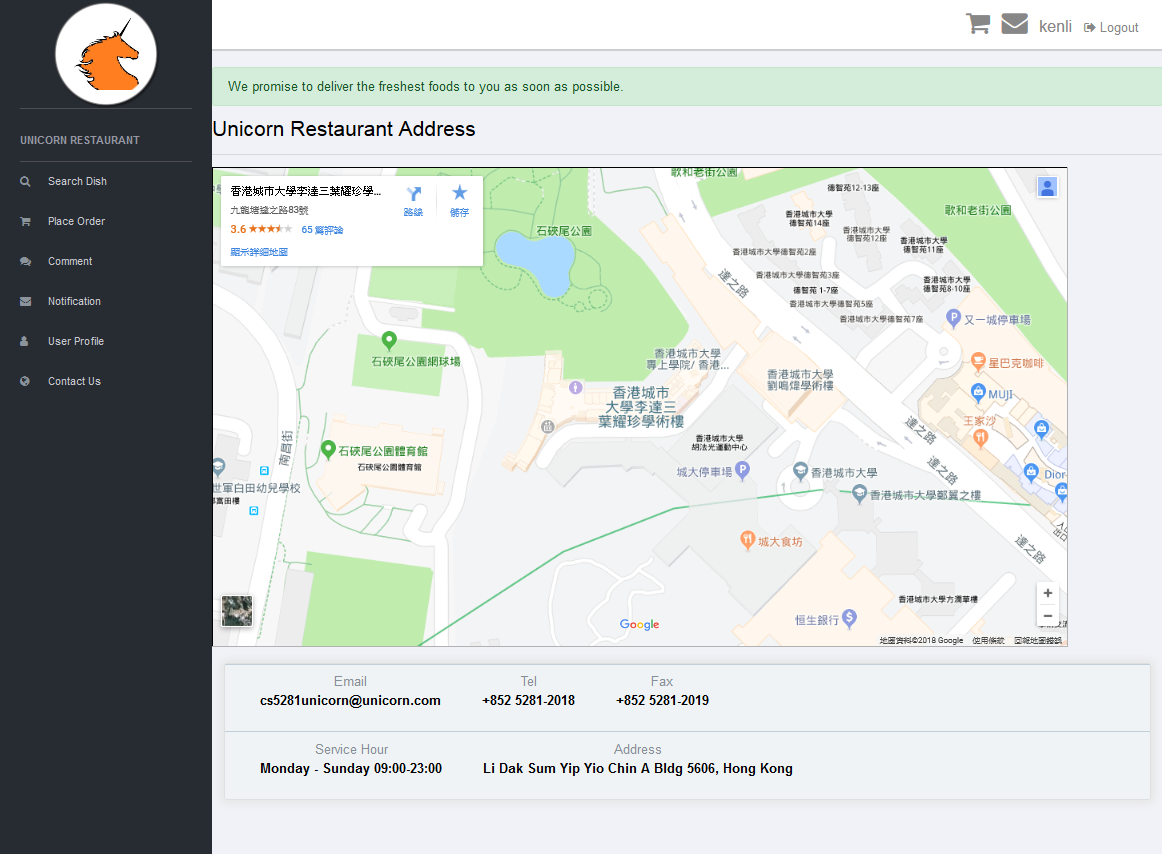


7.6 Comment on the placed order

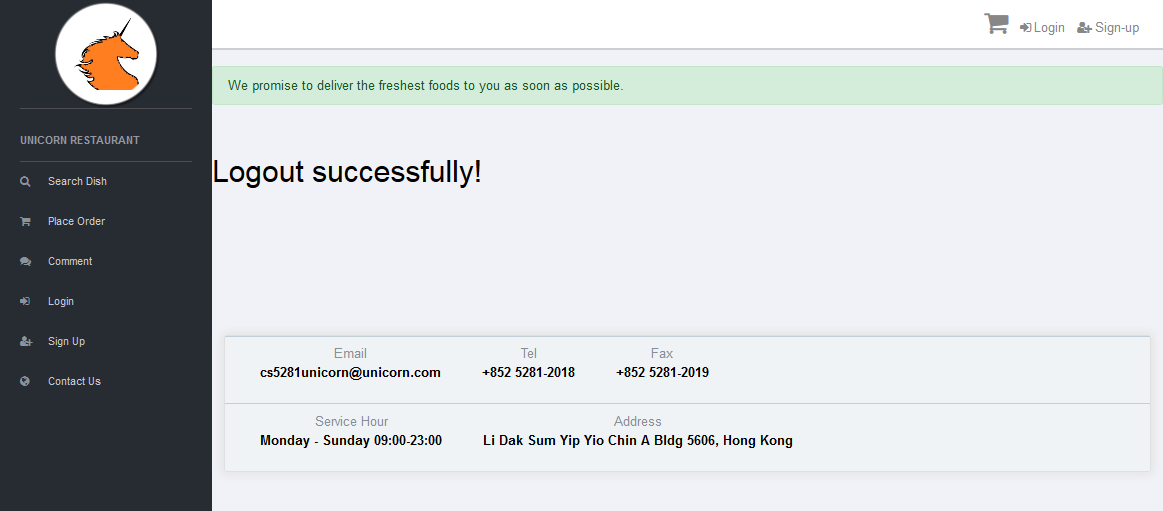




7.7 Check contact information

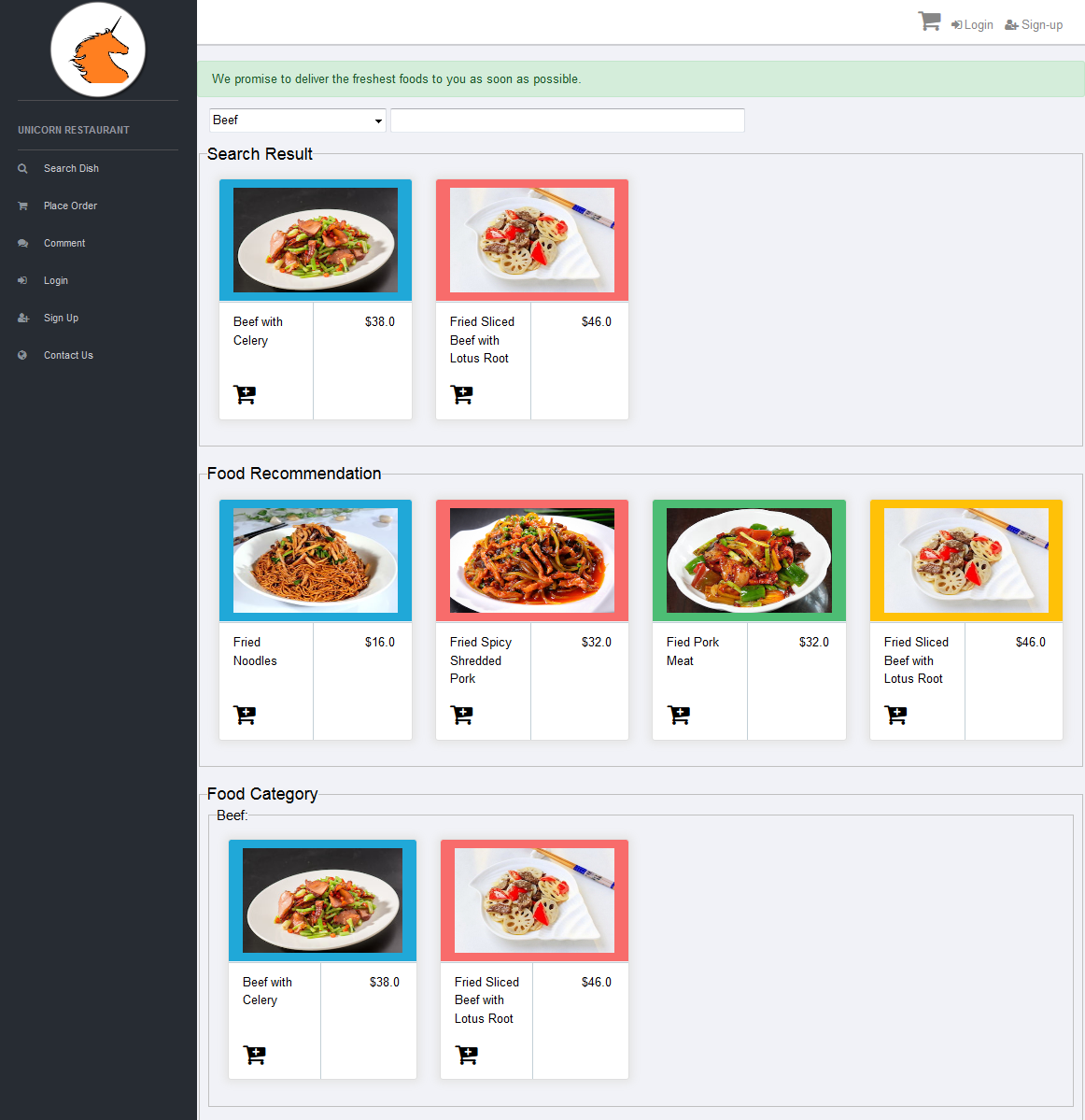


7.8 Logout



# 8. Function Specification of Search Dish and Food Recommendation

8.1 Screen Design of recom\_home.php

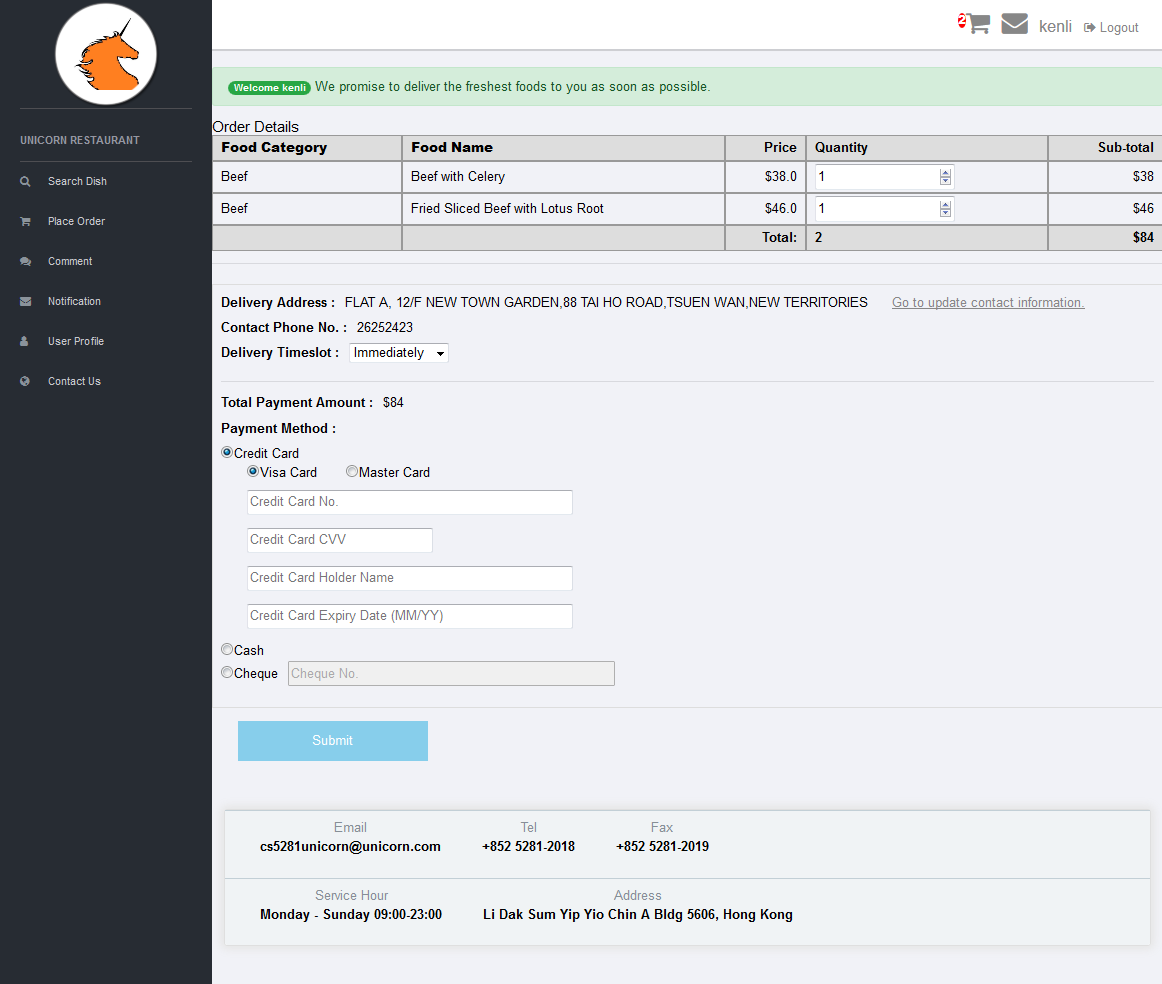


8.2 Input / output field and action definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Field Name | Input /Output | Mandatory /Optional | Format |
| **Search Result section** | | | | |
| 1 | Food Category | I | O | Input text |
| **Validation / Processing**   * This is a unique category id which can be used to search dishes in selected category. * When search button is clicked, a SQL command will be triggered to retrieve the CATEGORY table and the DISH table in the database with correspondent category. * If the selected category is "All", then display all the dishes. * If the selected category is not "All" and the food name input field is empty, all the dishes in the selected category will be displayed. * If the selected category is not "All" and the food name input field is not empty, only dishes with the same string (key words) and also under the selected category will be displayed. * The input category will be kept in cookie (cookie name is “unicornCookie”) for 30 days. This cookie will be loaded and input for searching when recom\_home.php page is being accessed. | | | |
| 2 | Search Text | I | O | Input text |
| **Validation / Processing**   * This is a string input by user which can be used to search dishes with the same or partially matched data. * When input the value in category or food name input field, a SQL command will be triggered to retrieve the detail from FOOD and FOOD\_TAG tables in the database with correspondent string. * An Ajax call will be triggered to check with database whether the input string is the same as strings in the database and display possible strings in real-time. * If the input string is found in the database, then displays all the dishes with the same string. * If the input string is not found in the database, then displays Error Message   [E302] No results! | | | |
| Food Image | O | M | Image |
| Food Name | O | M | Output text |
| Food Price | O | M | Output text |
| Add to cart | I | M | Button |
| **Validation / Processing**   * Add the selected food to shopping card in user login session. * Increase the number next to the shopping card in header bar. | | | |
| **Food Recommendation section** | | | | |
| 3 | Food Image | O | M | Image |
| Food Name | O | M | Output text |
| Food Price | O | M | Output text |
| Add to cart | I | M | Button |
| **Validation / Processing**   * Recommend 4 lowest price foods in this section. * Add the selected food to shopping card in user login session. * Increase the number next to the shopping card in header bar. | | | |
| **Food Category section** | | | | |
|  | Food Category | O | M | Output text |
| Food Image | O | M | Image |
| Food Name | O | M | Output text |
|  | Food Price | O | M | Output text |
|  | Add to cart | I | M | Button |
|  | **Validation / Processing**   * Show all foods group by food category. * Add the selected food to shopping card in user login session. * Increase the number next to the shopping card in header bar. | | | |

# 9. Function Specification of Place Order

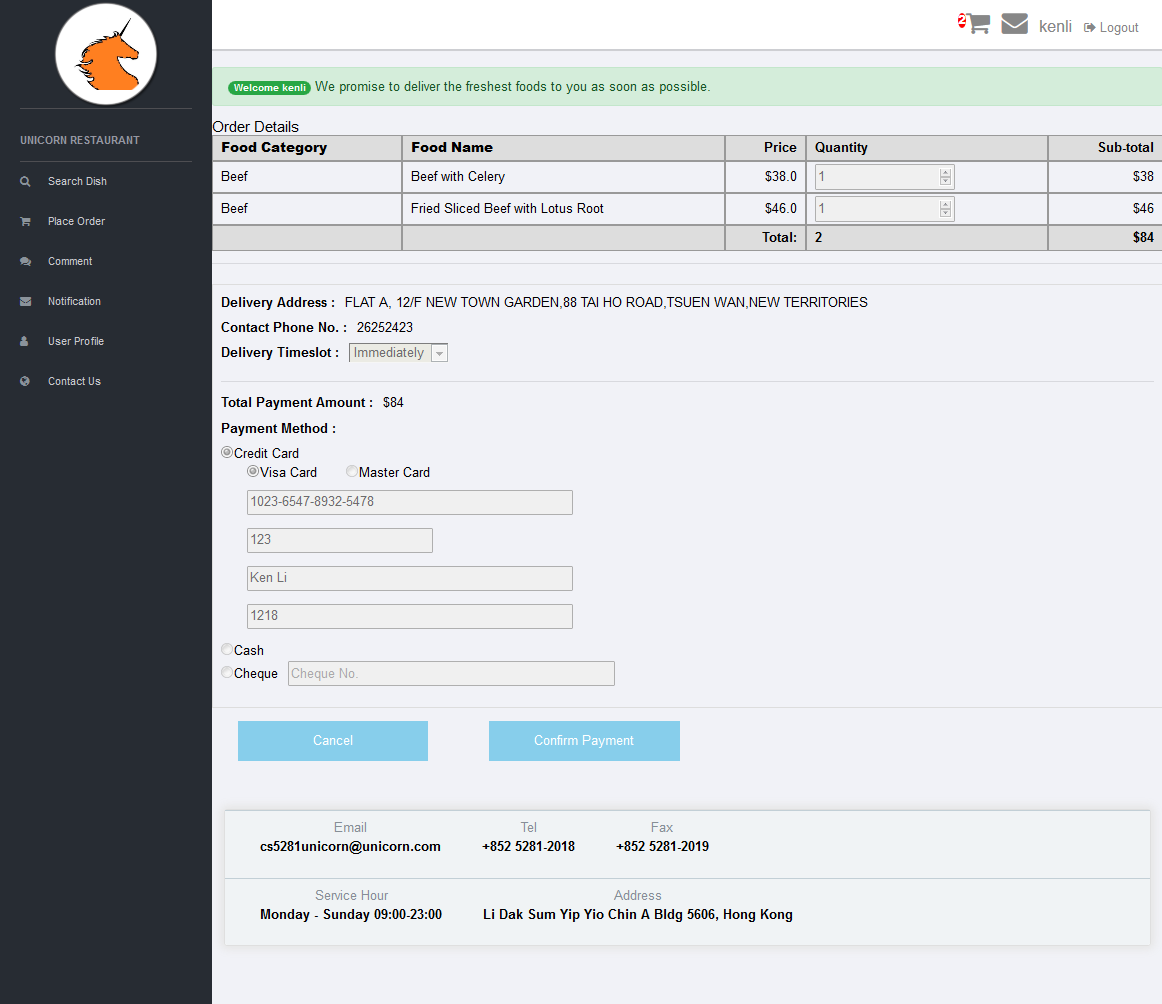
9.1 Screen Design of cart.php



9.2 Input / output field and action definition of cart.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Food Category | O | M | Output text |
| **Validation / Processing**   * Food category of the food selected by customer. * When customer selects food, the system will store the Food ID and corresponding quantity into login session. When go into cart.php page, the page will load the Food ID and quantity from login session and will get “Food Category”, “Food Name” and “Food Price” from database by using food id and display in the table for customer to complete the payment process. | | | |
|  | Food Name | O | M | Output text |
| **Validation / Processing**   * Food name of the food selected by customer. | | | |
|  | Price | O | M | Output text |
| **Validation / Processing**   * Food price of the food selected by customer. | | | |
|  | Quantity | I | M | Text box with maximum length 3.  Minimum value : 0  Maximum value : 100 |
| **Validation / Processing**   * When customer updates this quantity, corresponding Sub-total price, Total quantity and Total price will be updated immediately. * If Quantity greater than 5, show error message,   [E201] Quantity of a food must be less than or equal to 5! | | | |
|  | Sub-total | O | M | Output text |
| **Validation / Processing**   * Sub-total = Price x Quantity of the food. | | | |
|  | (Total quantity) | O | M | Output text |
| **Validation / Processing**   * (Total quantity) = Sum of all foods’ quantity. | | | |
|  | (Total payment amount) | O | M | Output text |
| **Validation / Processing**   * (Total payment amount) = Sum of all foods’ sub-total price. * If (Total payment amount) equal to 0, show error message,   [E202] Total payment amount must be greater than 0!   * If order is submitted successfully, show information message,   [I201] The order is submitted successfully! | | | |
|  | Delivery Address | O | M | Output text |
| **Validation / Processing**   * User registered address. Food will be delivered to this address. | | | |
|  | Go to update contact information | I | O | Hyperlink |
| **Validation / Processing**   * Hyperlink to User Profile which allow user to update personal and contact information. | | | |
|  | Contact Phone No. | O | M | Output text |
| **Validation / Processing**   * User registered contact phone no. | | | |
|  | Delivery Timeslot | I | M | Selection box |
| **Validation / Processing**   * Food delivery timeslot * Selection box with 3 options,   -. Immediately (Default value)  -. After 1 hour  -. After 2 hours | | | |
|  | Total Payment Amount | O | M | Output text |
| **Validation / Processing**   * Total payment amount for all ordered food. | | | |
|  | Payment Method | I | M | Radio button |
| **Validation / Processing**   * There are 3 payment methods,   -. Credit Card  -. Cash  -. Cheque | | | |
| 1. 14 | Credit Card Type | I | O | Radio button |
| **Validation / Processing**   * There are 2 credit card type,   -. Visa Card (Default)  -. Master Card | | | |
|  | Credit Card No. | I | O | Text box with maximum length 16. |
| **Validation / Processing**   * This textbox will be shown only when “Credit Card” payment method is selected. * When “Credit Card” payment method is selected and “Credit Card No.” is empty, show error message,   [E203] Credit card no. must be input!   * When “Credit Card” payment method is selected and “Credit Card No.” length is NOT numeric, show error message,   [E204] Credit card no. must be numeric!   * When “Credit Card” payment method is selected and “Credit Card No.” length is NOT 16 digits, show error message,   [E205] Credit card no. length must be 16 digits! | | | |
|  | Credit Card CVV | I | O | Text box with maximum length 3. |
| **Validation / Processing**   * This textbox will be shown only when “Credit Card” payment method is selected. * When “Credit Card CVV” is empty, show error message,   [E206] Credit card CVV no. must be input!   * When “Credit Card CVV” is NOT numeric, show error message,   [E207] Credit card CVV no. must be numeric!   * When “Credit Card CVV” is NOT 3 digits, show error message,   [E208] Credit card CVV no. length must be 3 digits! | | | |
|  | Credit Card Holder name | I | O | Text box |
| **Validation / Processing**   * This textbox will be shown only when “Credit Card” payment method is selected. * When “Credit Card Holder Name” is empty, show error message,   [E209] Credit card holder name must be input!   * When “Credit Card Holder Name” is NOT in English alphabets, show error message,   [E217] Credit card holder name must be composed of English alphabets! | | | |
|  | Credit Card Expiry Date | I | O | Text box |
| **Validation / Processing**   * This textbox will be shown only when “Credit Card” payment method is selected. * When “Credit Card Expiry Date” is empty, show error message,   [E210] Credit card expiry date must be input!   * When “Credit Card Expiry Date” is NOT numeric, show error message,   [E211] Credit card expiry date must be numeric!   * When “Credit Card Expiry Date” is NOT 4 digits, show error message,   [E212] Credit card expiry date length must be 4 digits!   * When “Credit Card Expiry Date” is NOT a valid date, show error message,   [E213] Credit card expiry date is invalid! | | | |
|  | Cheque No. | I | M | Text box with maximum length 10. |
| **Validation / Processing**   * This textbox will be shown only when “Cheque” payment method is selected. * When “Cheque” payment method is selected and “Cheque No.” is empty, show error message,   [E214] Cheque no. must be input!   * When “Cheque” payment method is selected and “Cheque No.” length is NOT numeric, show error message,   [E215] Cheque no. must be numeric!   * When “Cheque” payment method is selected and “Cheque No.” length is NOT 10 digits, show error message,   [E216] Cheque no. length must be 10 digits! | | | |
|  | Submit | N/A | N/A | Input button |
| **Validation / Processing**   * Trigger validation for input fields. * For any system error, show error message,   [E217] System error, please try again later!   * When input fields validations are passed, go to cartConfirm.php | | | |

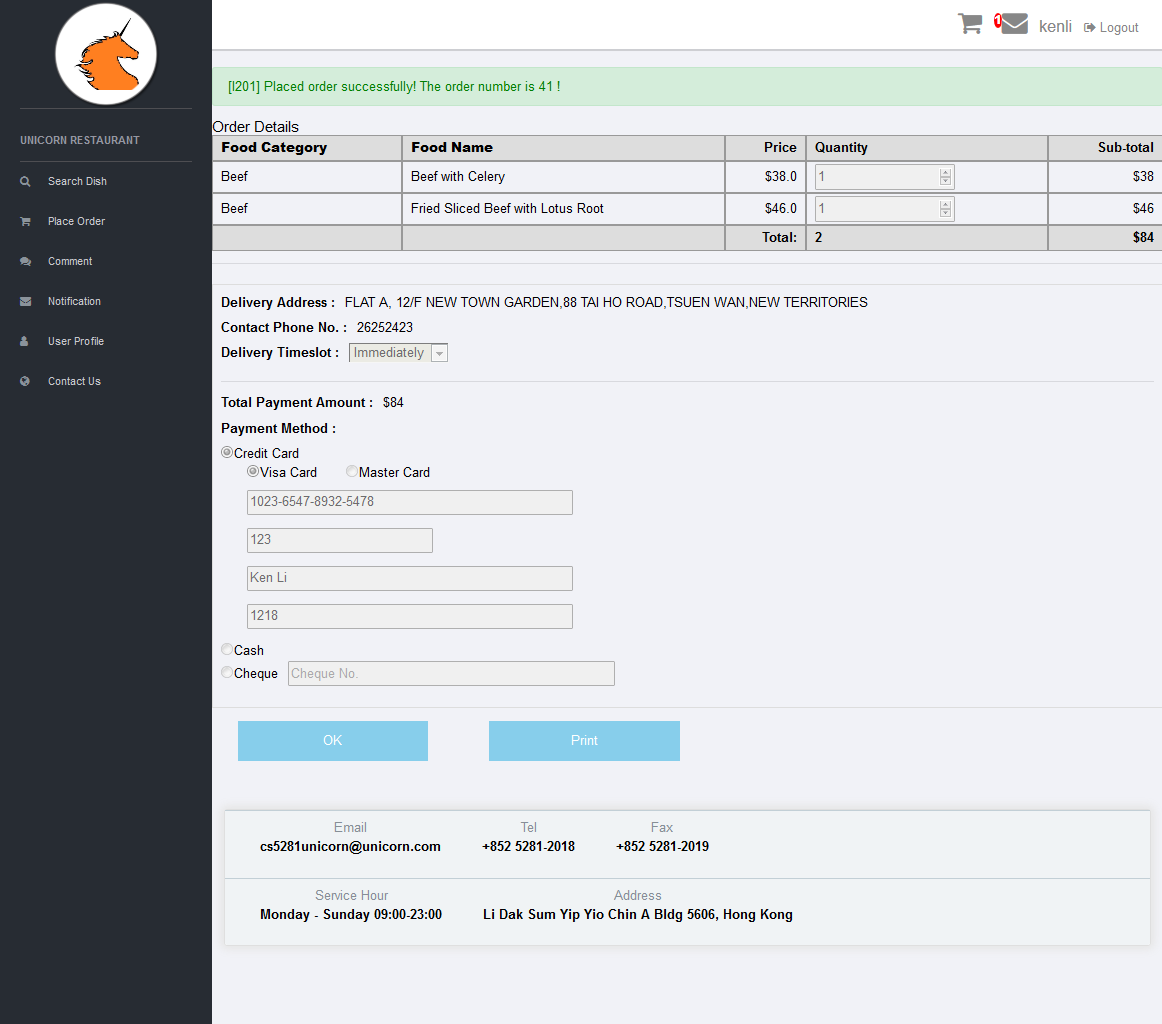
9.3 Screen Design of cartConfirm.php,



9.4 Input / output fields and action definition of cartConfirm.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Food Category | O | M | Output text |
|  | Food Name | O | M | Output text |
|  | Price | O | M | Output text |
|  | Quantity | O | M | Output text |
|  | Sub-total | O | M | Output text |
|  | (Total quantity) | O | M | Output text |
|  | (Total payment amount) | O | M | Output text |
|  | Delivery Address | O | M | Output text |
|  | Go to update contact information | I | O | Hyperlink |
|  | Contact Phone No. | O | M | Output text |
|  | Delivery Timeslot | O | M | Disabled selection box |
|  | Total Payment Amount | O | M | Output text |
|  | Payment Method | O | M | Disabled radio button |
|  | Credit Card Type | O | O | Disabled radio button |
|  | Credit Card No. | O | O | Output text |
|  | Credit Card CVV | O | O | Output text |
|  | Credit Card Holder name | O | O | Output text |
|  | Credit Card Expiry Date | O | O | Output text |
|  | Cheque No. | O | O | Output text |
|  | Cancel | N/A | N/A | Input button |
| **Validation / Processing**   * Go to back to cart.php page. | | | |
|  | Confirm Payment | N/A | N/A | Input button |
| **Validation / Processing**   * Insert order information and order detail into database. * Insert order notification. * When order details and notifications are inserted successfully, go to cartCompleted.php. | | | |

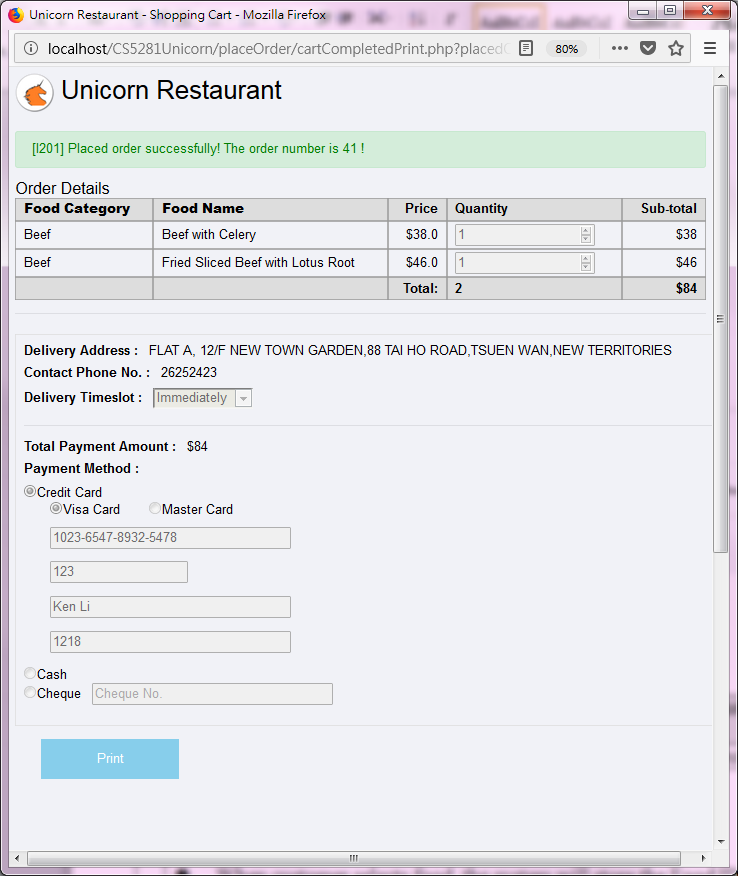
9.5 Screen Design of cartCompleted.php,



9.6 Input / output fields and action definition of cartCompleted.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Food Category | O | M | Output text |
|  | Food Name | O | M | Output text |
|  | Price | O | M | Output text |
|  | Quantity | O | M | Output text |
|  | Sub-total | O | M | Output text |
|  | (Total quantity) | O | M | Output text |
|  | (Total payment amount) | O | M | Output text |
|  | Delivery Address | O | M | Output text |
|  | Go to update contact information | I | O | Hyperlink |
|  | Contact Phone No. | O | M | Output text |
|  | Delivery Timeslot | O | M | Disabled selection box |
|  | Total Payment Amount | O | M | Output text |
|  | Payment Method | O | M | Disabled radio button |
|  | Credit Card Type | O | O | Disabled radio button |
|  | Credit Card No. | O | O | Output text |
|  | Credit Card CVV | O | O | Output text |
|  | Credit Card Holder name | O | O | Output text |
|  | Credit Card Expiry Date | O | O | Output text |
|  | Cheque No. | O | O | Output text |
|  | OK | N/A | N/A | Input button |
| **Validation / Processing**   * Go to back to home page (recom\_home.php). | | | |
|  | Print | N/A | N/A | Input button |
| **Validation / Processing**   * Pop up cartCompletedPrint.php page for printing. | | | |

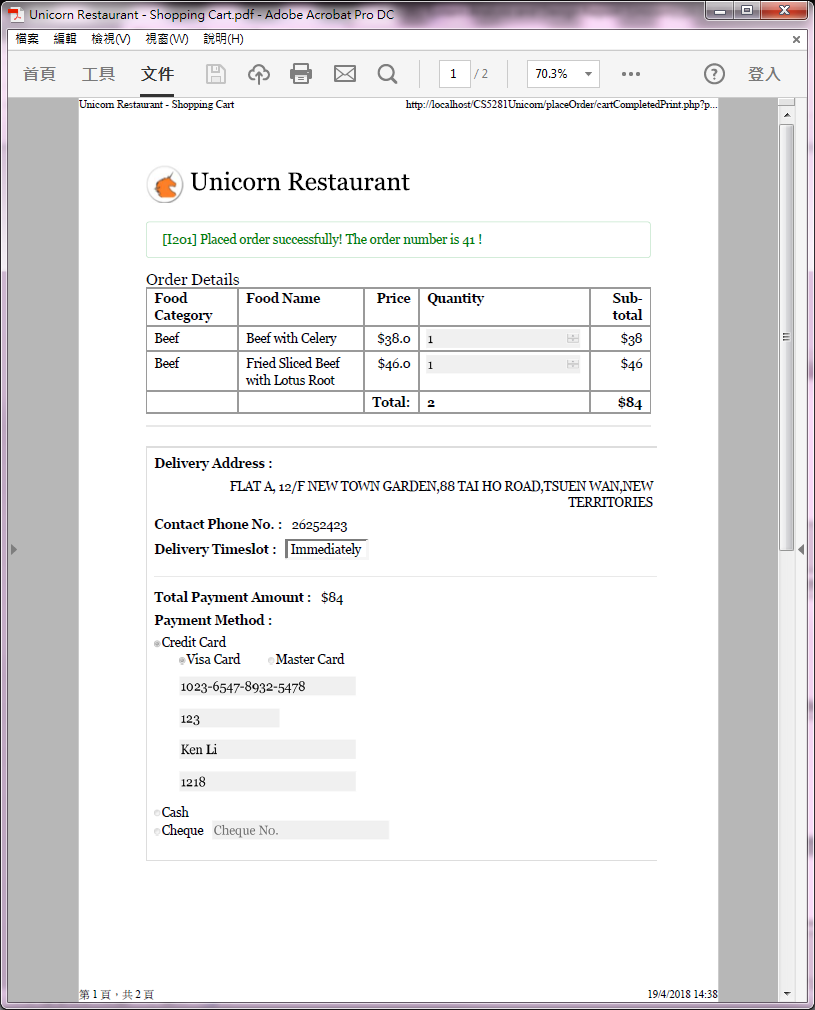
9.7 Screen Design of cartCompletedPrint.php,



9.8 Input / output fields and action definition of cartCompletedPrint.php

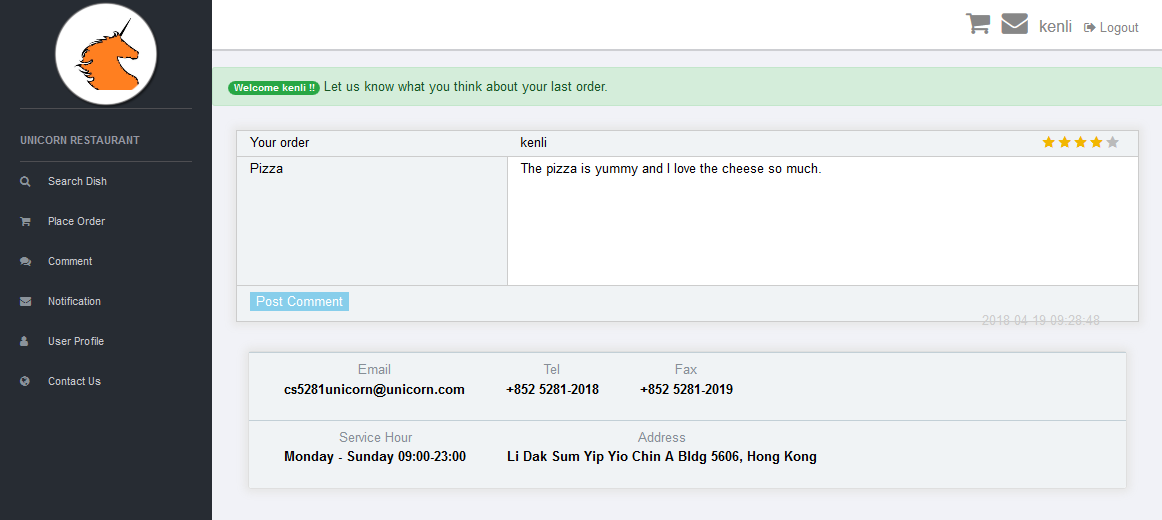
| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Food Category | O | M | Output text |
|  | Food Name | O | M | Output text |
|  | Price | O | M | Output text |
|  | Quantity | O | M | Output text |
|  | Sub-total | O | M | Output text |
|  | (Total quantity) | O | M | Output text |
|  | (Total payment amount) | O | M | Output text |
|  | Delivery Address | O | M | Output text |
|  | Contact Phone No. | O | M | Output text |
|  | Delivery Timeslot | O | M | Disabled selection box |
|  | Total Payment Amount | O | M | Output text |
|  | Payment Method | O | M | Disabled radio button |
|  | Credit Card Type | O | O | Disabled radio button |
|  | Credit Card No. | O | O | Output text |
|  | Credit Card CVV | O | O | Output text |
|  | Credit Card Holder name | O | O | Output text |
|  | Credit Card Expiry Date | O | O | Output text |
|  | Cheque No. | O | O | Output text |
|  | Print | N/A | N/A | Input button |
| **Validation / Processing**   * Trigger printing to windows default printer. | | | |

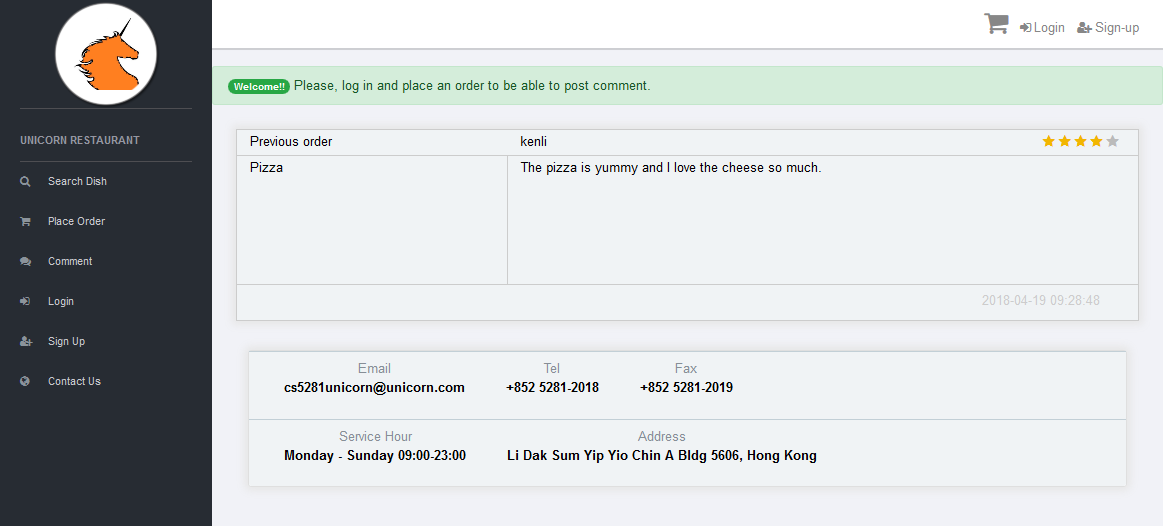
9.9 Print layout



# 10. Function Specification of Comment

10.1 Screen Design of comment.php

****

****

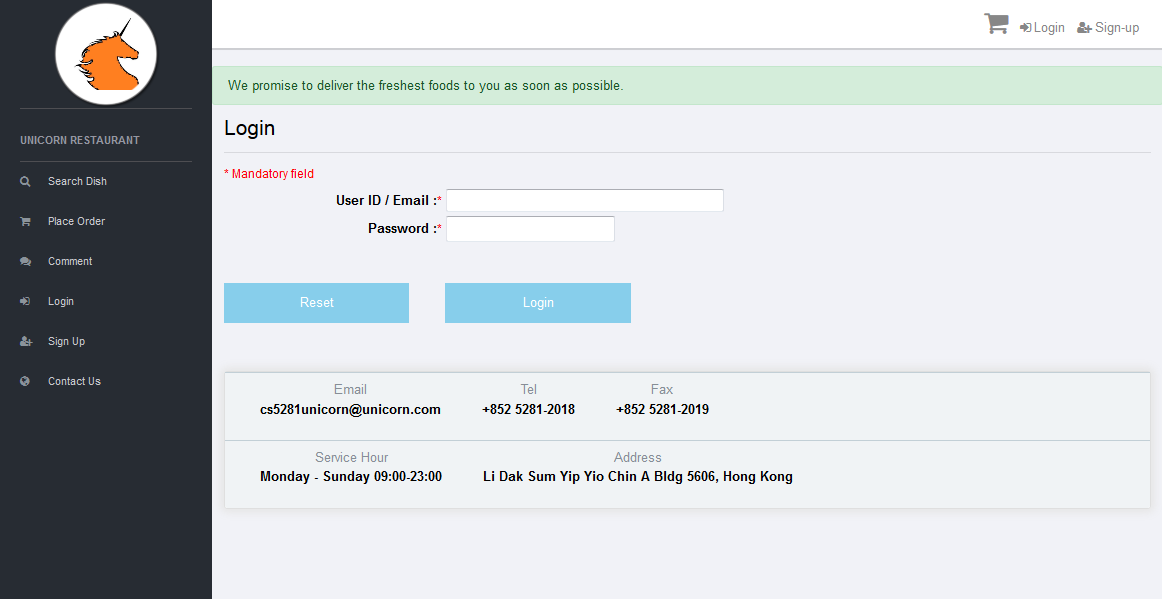
10.2 Input / output fields and action definition of comment\_section.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
| 1. 1 | Previous Order | O | M | Output text |
| **Validation / Processing**   * Previous order food name. | | | |
|  | User ID who placed the order | O | M | Output text |
|  | Comment | I | M | Textarea |
|  | Rating | I | M | Radio button |
|  | Comment date | O | M | Output text |
|  | Post comment | N/A | N/A | Input button |
|  | **Validation / Processing**   * Insert comment into database and stay in current page. Disabled allow input field and hide “Post Comment” button. Show information message, | | | |

# 

# 11. Function Specification of Login and Logout

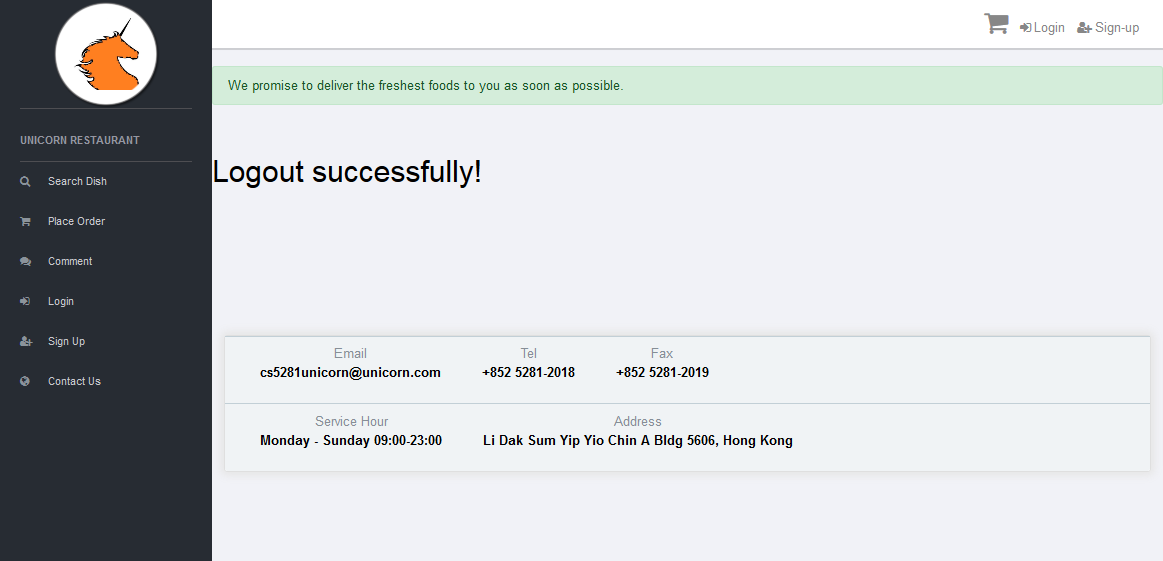
11.1 Screen Design of login.php



11.2 Input / output fields and action definition of login.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | User ID / Email | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * If User ID / Email is empty, show error message,   [E601] User ID / Email must be input! | | | |
|  | Password | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * If Password is empty, show error message,   [E602] Password must be input!   * Password length must be 8 – 20 characters, otherwise show error message,   [E603] Password length must be 8 - 20 characters!   * Password must contain at least 2 upper case characters, otherwise show error message,   [E604] Password must contain at least 2 upper case characters!   * Password must contain at least 2lower case characters, otherwise show error message,   [E605] Password must contain at least 2 lower case characters!   * Password must contain at least 2 numeric characters, otherwise show error message,   [E606] Password must contain at least 2 numeric characters!   * Password must contain at least 2 special characters, otherwise show error message,   [E607] Password must contain at least 2 special characters!  Special characters included  !@#$%^&\*()\_- | | | |
|  | Login | N/A | N/A | Input Button |
| **Validation / Processing**   * Execute client side JavaScript field validations. * Execute server side PHP field validations. * If 1 of field validation rules is failed, stay in login.php page. * If field validation rules are passed,   -. Execute md5 hash function to encrypt user password.  -. Check the user table in database.  -. Compare the input User ID and Password with the one in database.  -. Go to recom\_home.php page if success. Then set session variables and start session.  -. Go to login.php page if failure.   * New session and keep user ID, email, user account privilege, login date time and last request time into session variable. * Set the timeout limit to 30 minutes. When last request time was more than 30 minutes ago, clear all session variables and destroy all sessions. | | | |

11.3 Screen Design of logout.php

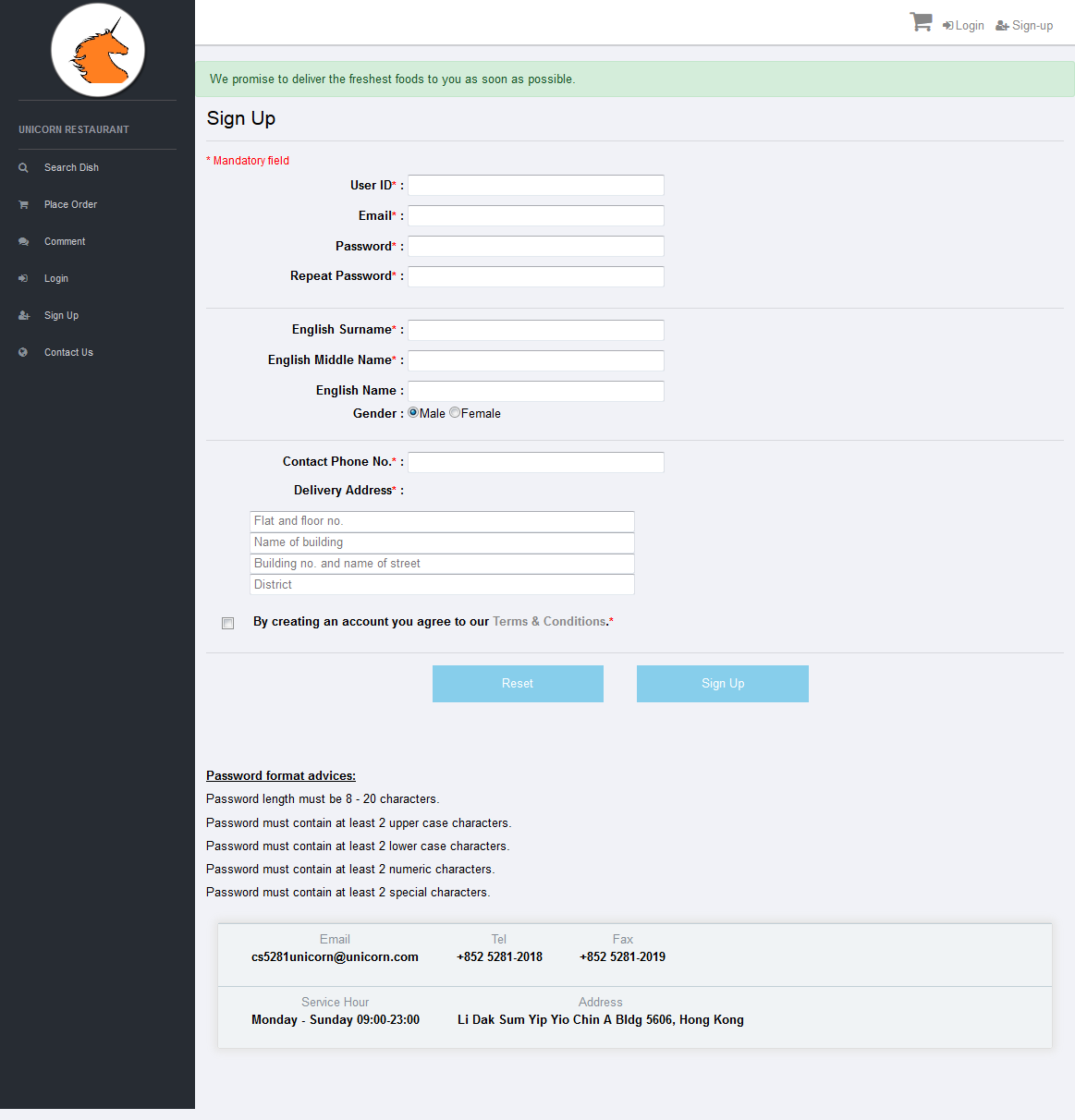


11.4 Input / output fields and action definition of logout.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Logout | N/A | N/A | Input button |
| **Validation / Processing**   * If user click logout button, go to logout.php page. * Clear all session variables and destroy all sessions. | | | |

# 12. Function Specification of Sign Up

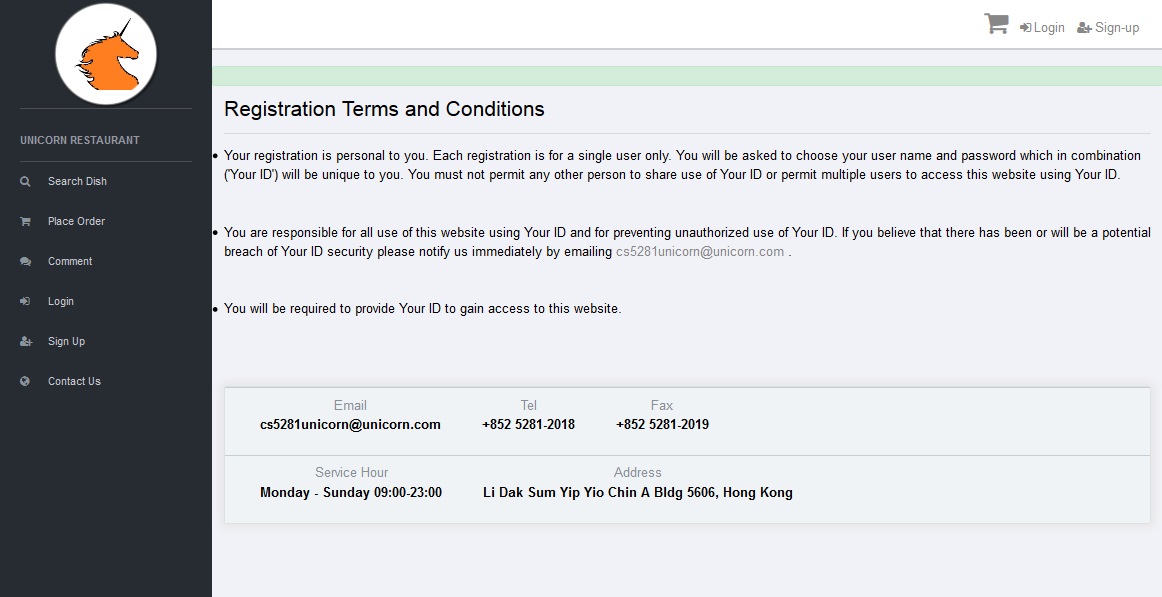
12.1 Screen Design of registerForm.php,



12.2 Input / output fields and action definition of registerForm.php

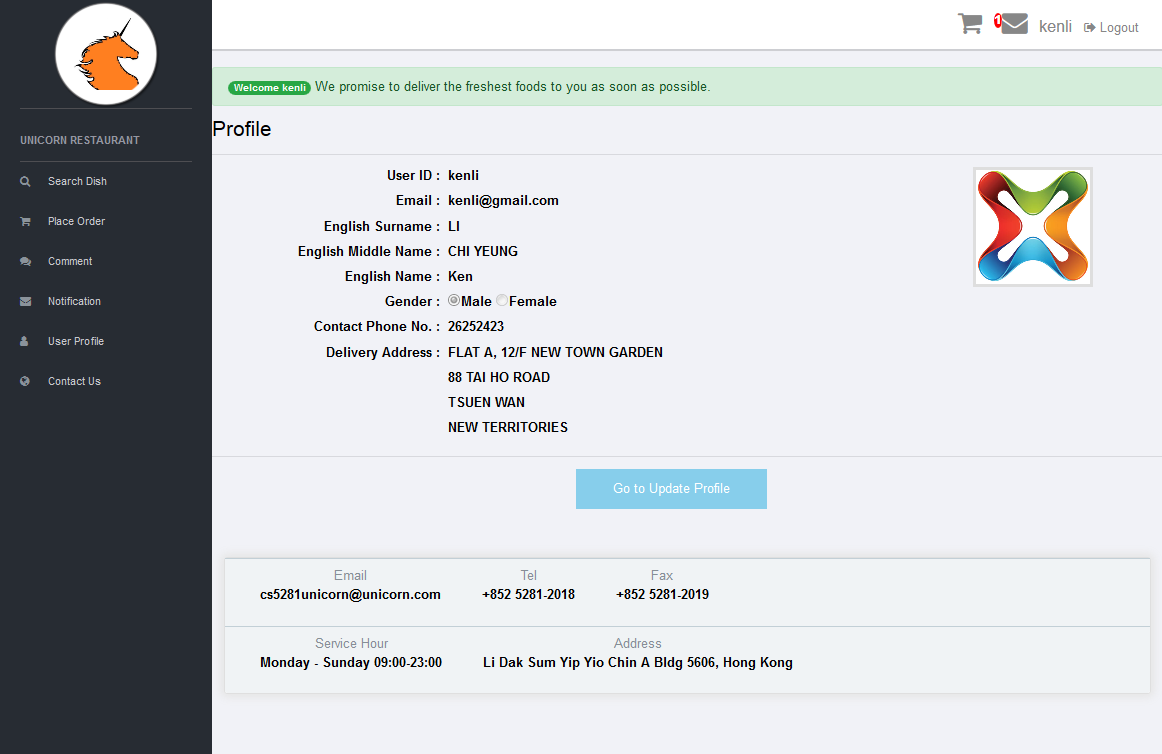
| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | User ID | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * This is a unique user id which can be used to login Unicorn Restaurant system. * An Ajax call will be triggered to check with database if the inputted User ID is already in used. * If User ID is empty, show error message,   [E001] User ID must be input!   * User ID must be unique. If User ID is not found in database USER table, show information message,   [I001] User ID is acceptable!   * If User ID is found in database USER table, show error message,   [E002] The User ID is already in used; please try a different User ID!   * User ID length must be 5 – 50 characters, otherwise show error message,   [E003] User ID length must be 5 - 50 characters!   * User ID must start with English alphabet, otherwise show error message,   [E004] The User ID must start with English alphabet!   * User ID must be composed of English alphabets and numbers, otherwise show error message,   [E005] The User ID must be composed of English alphabets or English alphabets and numbers! | | | |
|  | Email | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is a unique email address which can be used to login Unicorn Restaurant system. * If Email is empty, show error message,   [E006] Email must be input!   * Email must be unique. If Email is not found in database USER table, show information message,   [I002] Email is acceptable!   * If Email is found in database USER table, show error message,   [E007] Email is already registered, please register by other email!   * Validate email address with the following conditions,  1. Must has 1 @ character in the email address 2. Not contain the following special characters before @   ^<>()[\]\\.,;:@\"   1. Not contain space before @ 2. Only allow the following characters after @   0-9, a-z, A-Z and .  Invalid email address, show error message,  [E008] Invalid Email! | | | |
|  | Password | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is user password used to login Unicorn Restaurant system. The password will be encrypted by md5 hash function and store in database USER table. * If password is empty, show error message,   [E009] Password must be input!   * Password length must be 8 – 20 characters, otherwise show error message,   [E010] Password length must be 8 - 20 characters!   * Password must contain at least 2 upper case characters, otherwise show error message,   [E011] Password must contain at least 2 upper case characters!   * Password must contain at least 2lower case characters, otherwise show error message,   [E012] Password must contain at least 2 lower case characters!   * Password must contain at least 2 numeric characters, otherwise show error message,   [E013] Password must contain at least 2 numeric characters!   * Password must contain at least 2 special characters, otherwise show error message,   [E014] Password must contain at least 2 special characters!  Special characters included  !@#$%^&\*()\_- | | | |
|  | Repeat Password | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This repeat password is used to make sure user type in their expected password value; it will be used to cross check with password. * Repeat Password is empty, show error message,   [E015] Repeat Password must be input!   * Repeat Password must be equal to Password, otherwise show error message,   [E016] Repeat Password must be equal to Password! | | | |
|  | English Surname | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * User English surname. * If empty, show error message,   [E017] English Surname must be input! | | | |
|  | English Middle Name | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * User English middle name. * If empty, show error message,   [E018] English Middle Name must be input! | | | |
|  | English Name | I | O | Text box with maximum length 50. |
| **Validation / Processing**   * User English name. | | | |
|  | Gender | I | M | Radio button.  Default value is “Male” |
| **Validation / Processing**   * User gender. | | | |
|  | Contact Phone No. | I | M | Text box with maximum length 50.  Only allow numeric. |
| **Validation / Processing**   * This is user contact phone number. * If empty, show error message,   [E019] Contact Phone No. must be input!   * Contact Phone No. must be numeric, otherwise show error message,   [E020] Contact Phone No. must be numeric! | | | |
|  | Address (Flat and floor no.) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 1 of user address which is expected to input flat and floor no. * If Address (Flat and floor no.) empty, show error message,   [E021] Flat and floor no. must be input! | | | |
|  | Address (Name of building) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 2 of user address which is expected to input name of building. * If Address (Name of building) empty, show error message,   [E022] Name of building must be input! | | | |
|  | Address (Building no. and name of street) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 3 of user address which is expected to input building no. and name of street. * If Address (Building no. and name of street) empty, show error message.   [E023] Building no. and name of street must be input! | | | |
|  | Address (District) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 4 of user address which is expected to input district. * If Address (District) empty, show error message,   [E024] District must be input! | | | |
|  | By creating an account you agree to our Terms & Conditions | I | M | Checkbox |
| **Validation / Processing**   * This is terms and conditions for creating user account in Unicorn Restaurant system. * If this checkbox is NOT checked, show error message,   [E025] Terms & Conditions must be accepted! | | | |
|  | Terms & Conditions | N/A | N/A | Hyperlink |
| **Validation / Processing**   * Hyperlink to Terms and Conditions page. | | | |
|  | Reset | N/A | N/A | Input Button |
| **Validation / Processing**   * Reset / clear registration form and stay in registerForm.php page. | | | |
|  | Sign Up | N/A | N/A | Input Button |
| **Validation / Processing**   * Execute client side JavaScript field validations. * Execute server side PHP field validations. * If 1 of field validation rules is failed, stay in registerForm.php page. * If field validation rules are passed,   -. Execute md5 hash function to encrypt user password.  -. Generate an activation token by concatenating current date and hash (sha256) value of User ID.  -. Create user account in database UNICORN\_USER table with encrypted user password and the activation token.  -. Send activation email to user email with an activation URL which is appended the activation token.  -. Go to registerSuccess.php page if success.  -. Go to registerFailure.php page if failure. | | | |
|  | /CS5281Unicorn/registration/activateService.php |  |  | Activation URL in activation email.  The URL composed by activateService.php page + current date concating hash value of USER\_ID + EMAIL |
| **Validation / Processing**   * When user click on the activation URL in the activation email, * The activationService.php will validate the regToken with the REG\_TOKEN in USER table. * Remove REG\_TOKEN from UNICORN\_USER table. * Login Unicorn web application immediately by updating the LAST\_LOGIN\_TIME in USER table. * Prepare session with the following session variables,   -. login\_user\_id  -. login\_user\_email  -. login\_date\_time  -. last\_request\_time   * Go to recom\_home.php if success. * Go to activateFailure.php if failure. | | | |

12.3 Screen Design of registerTerms&Conditions.php,



# 13. Function Specification of User Profile

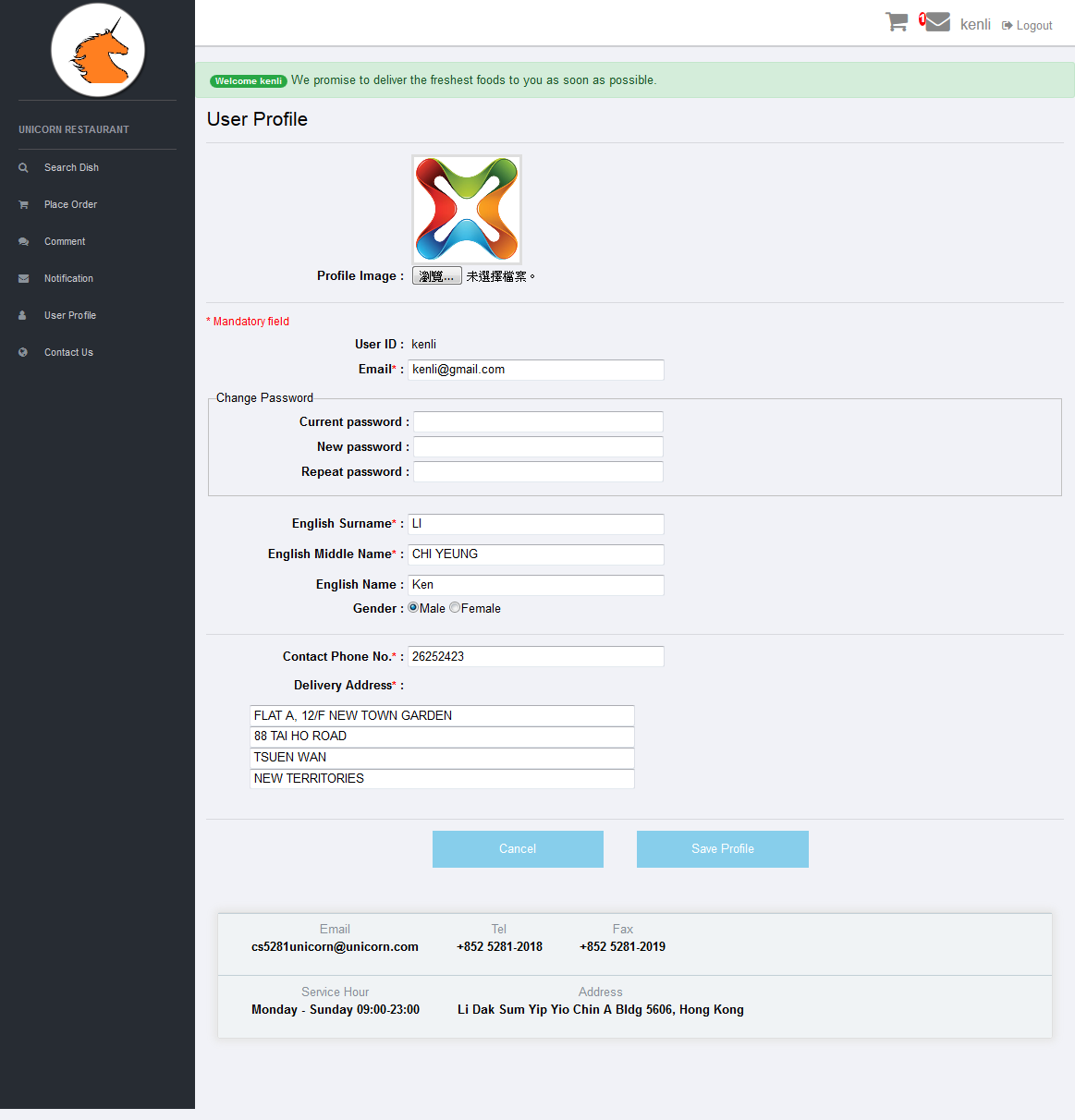
13.1 Screen Design of userProfile.php



13.2 Input / output fields and action definition of userProfile.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | User ID | O | M | Output text |
|  | Email | O | M | Output text |
|  | English Surname | O | M | Output text |
|  | English Middle Name | O | M | Output text |
|  | English Name | O | O | Output text |
|  | Gender | O | M | Disabled radio button. |
|  | Contact Phone No. | O | M | Output text |
|  | Address (Flat and floor no.) | O | M | Output text |
|  | Address (Name of building) | O | M | Output text |
|  | Address (Building no. and name of street) | O | M | Output text |
|  | Address (District) | O | M | Output text |

13.3 Screen Design of updateUserProfile.php

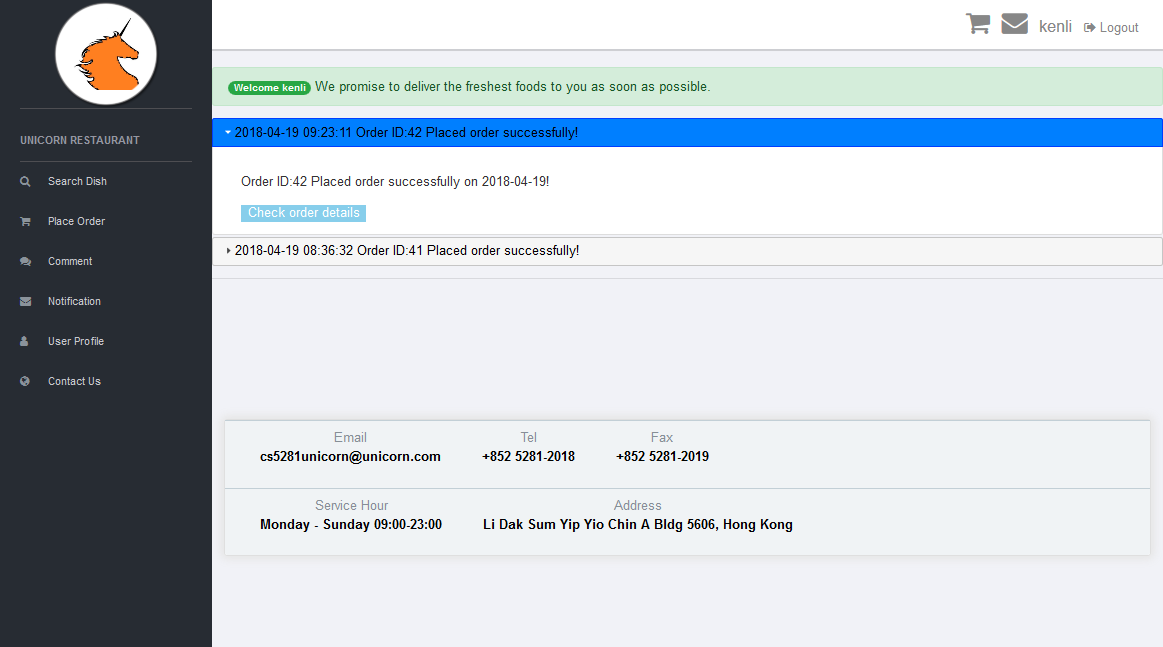


13.4 Input / output fields and action definition of updateUserProfile.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Profile Image | I | M | Profile image file path |
| **Validation / Processing**   * Image file will be kept in resources folder and file name will be same to the User ID. * If file path is empty, show error message,   [E712] Invalid file!   * If file size is greater than 1MB, show error message,   [E713] File size must be equal to or less than 1MB!   * If file type is NOT jpg, png, jpeg or gif, show error message,   [E714] Only JPG, JPEG, PNG and GIF files are allowed!   * If there is any system error during file uploading, show error message,   [E715] File upload error, please try again later!   * If file upload successfully, show information message,   [I703] The file {file name} has been uploaded! | | | |
|  | User ID | O | M | Output text |
|  | Email | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is a unique email address which can be used to login Unicorn Restaurant system. * If Email is empty, show error message,   [E701] Email must be input!   * Email must be unique. If Email is not found in database USER table, show information message,   [I701] Email is acceptable!   * If Email is found in database USER table, show error message,   [E711] Email is already registered, please register by other email!   * Validate email address with the following conditions,  1. Must has 1 @ character in the email address 2. Not contain the following special characters before @   ^<>()[\]\\.,;:@\"   1. Not contain space before @ 2. Only allow the following characters after @   0-9, a-z, A-Z and .  Invalid email address, show error message,  [E702] Invalid Email! | | | |
|  | Current password | I | O | Text box with maximum length 100. |
| **Validation / Processing**   * If “New password” or “Repeat password” is inputted and “Current password” is empty, show error message,   [E716] Current password must be input!   * If “New password” or “Repeat password” is inputted and “Current password” is NOT matched to database record, show error message,   [E725] Current password is wrong! | | | |
|  | New password | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is user password used to login Unicorn Restaurant system. The password will be encrypted by md5 hash function and store in database USER table. * If “Current password” is inputted and “New password” is empty, show error message,   [E717] New password must be input!   * Password length must be 8 – 20 characters, otherwise show error message,   [E719] New password length must be 8 - 20 characters!   * Password must contain at least 2 upper case characters, otherwise show error message,   [E720] New password must contain at least 2 upper case characters!   * Password must contain at least 2lower case characters, otherwise show error message,   [E721] New password must contain at least 2 lower case characters!   * Password must contain at least 2 numeric characters, otherwise show error message,   [E722] New password must contain at least 2 numeric characters!   * Password must contain at least 2 special characters, otherwise show error message,   [E723] New password must contain at least 2 special characters!  Special characters included  !@#$%^&\*()\_- | | | |
|  | Repeat password | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This repeat password is used to make sure user type in their expected password value; it will be used to cross check with password. * If “New password” is inputted and “Repeat password” is empty, show error message,   [E718] Repeat Password must be input!   * “Repeat password” must be equal to “New password”, otherwise show error message,   [E724] Repeat Password must be equal to New password! | | | |
|  | English Surname | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * User English surname. * If empty, show error message,   [E703] English Surname must be input! | | | |
|  | English Middle Name | I | M | Text box with maximum length 50. |
| **Validation / Processing**   * User English middle name. * If empty, show error message,   [E704] English Middle Name must be input! | | | |
|  | English Name | I | O | Text box with maximum length 50. |
| **Validation / Processing**   * User English name. | | | |
|  | Gender | I | M | Radio button.  Default value is “Male” |
| **Validation / Processing**   * User gender. | | | |
|  | Contact Phone No. | I | M | Text box with maximum length 50.  Only allow numeric. |
| **Validation / Processing**   * This is user contact phone number.   **JavaScript client side validation rules and PHP server slide validation rules:**   * If empty, show error message,   [E705] Contact Phone No. must be input!   * Contact Phone No. must be numeric, otherwise show error message,   [E706] Contact Phone No. must be numeric! | | | |
|  | Address (Flat and floor no.) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 1 of user address which is expected to input flat and floor no.   **JavaScript client side validation rules and PHP server slide validation rules:**   * If Address (Flat and floor no.) empty, show error message,   [E707] Flat and floor no. must be input! | | | |
|  | Address (Name of building) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 2 of user address which is expected to input name of building.   **JavaScript client side validation rules and PHP server slide validation rules:**   * If Address (Name of building) empty, show error message,   [E708] Name of building must be input! | | | |
|  | Address (Building no. and name of street) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 3 of user address which is expected to input building no. and name of street.   **JavaScript client side validation rules and PHP server slide validation rules:**   * If Address (Building no. and name of street) empty, show error message.   [E709] Building no. and name of street must be input! | | | |
|  | Address (District) | I | M | Text box with maximum length 100. |
| **Validation / Processing**   * This is part 4 of user address which is expected to input district.   **JavaScript client side validation rules and PHP server slide validation rules:**   * If Address (District) empty, show error message,   [E710] District must be input! | | | |
|  | Cancel | N/A | N/A | Input Button |
| **Validation / Processing**   * Reset / clear form and go to userProfile.php page. | | | |
|  | Save Profile | N/A | N/A | Input Button |
| **Validation / Processing**   * Execute client side JavaScript field validations. * Execute server side PHP field validations. * If 1 of field validation rules is failed, stay in updateUserProfile.php page. * If field validation rules are passed,   -. Execute md5 hash function to encrypt new password.  -. Update user account in database UNICORN\_USER table with encrypted new password.  -. Go to userProfile.php page if success and show information message,  [I702] Update user profile successfully!  -. Otherwise, stay in current page and show error message,  [E712] Update user profile failed, please try again later! | | | |

# 14. Function Specification of Notification

14.1 Screen Design of notification.php

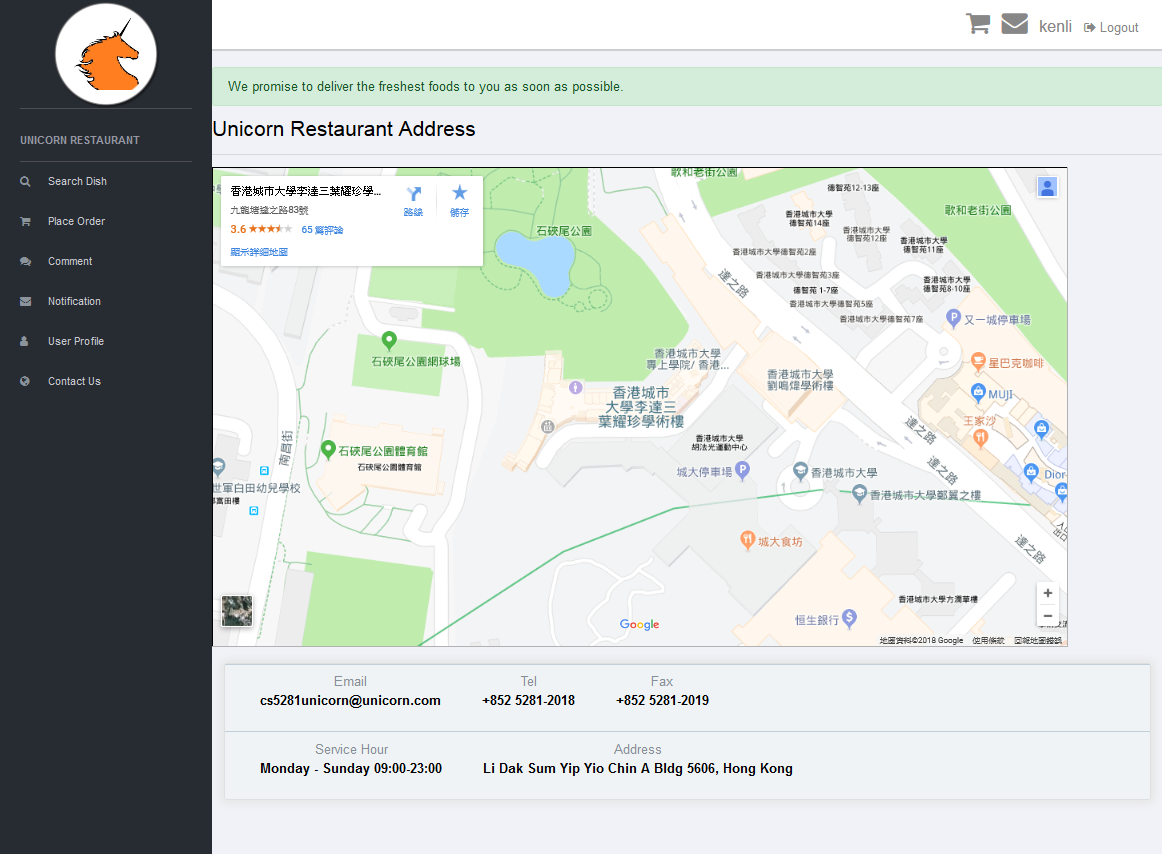


14.2 Input / output fields and action definition of notification.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
| 1. sdfsdf | Notification Subject | O | M | Output text |
| **Validation / Processing**   * Update the notification status to “Read”. | | | |
|  | Notification content | O | M | Output text |
|  | Check order details | O | M | Output text |
| **Validation / Processing**   * Go to cardCompleted.php to view order details. | | | |

# 15. Function Specification of Contact Us

15.1 Screen Design of contactUs.php



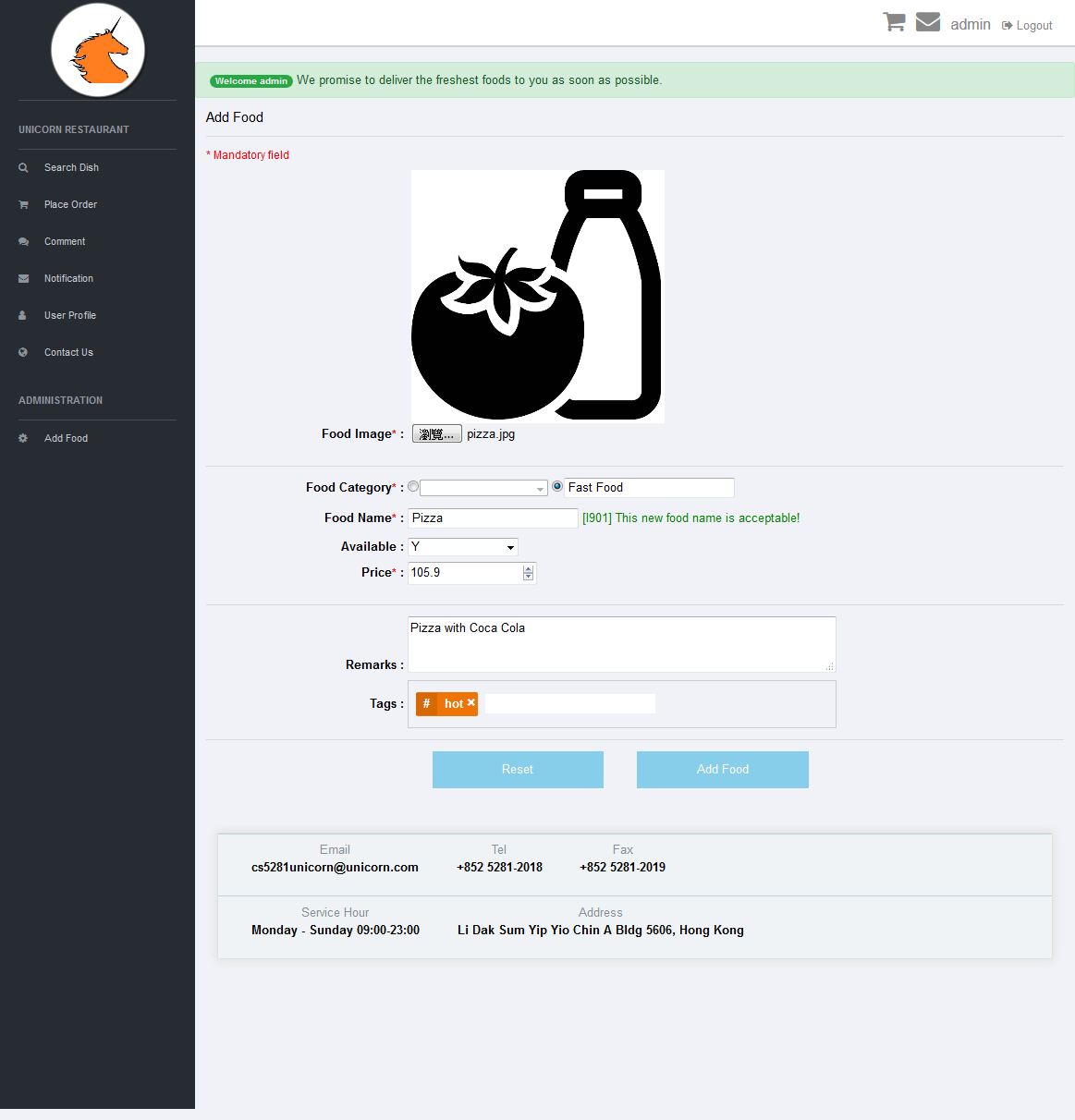
15.2 Input / output fields and action definition of contactUs.php

| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
| 1. sdfsdf | Google map | O | M | Output google map canvas |
| **Validation / Processing**   * Show the Unicorn Restaurant address in Google Map. | | | |

# 16. Function Specification of Add Food

16.1 Screen Design of addFood.php

This is an administrative function and only user id with administration privilege can access to the page.



16.2 Input / output fields and action definition of addFood.php

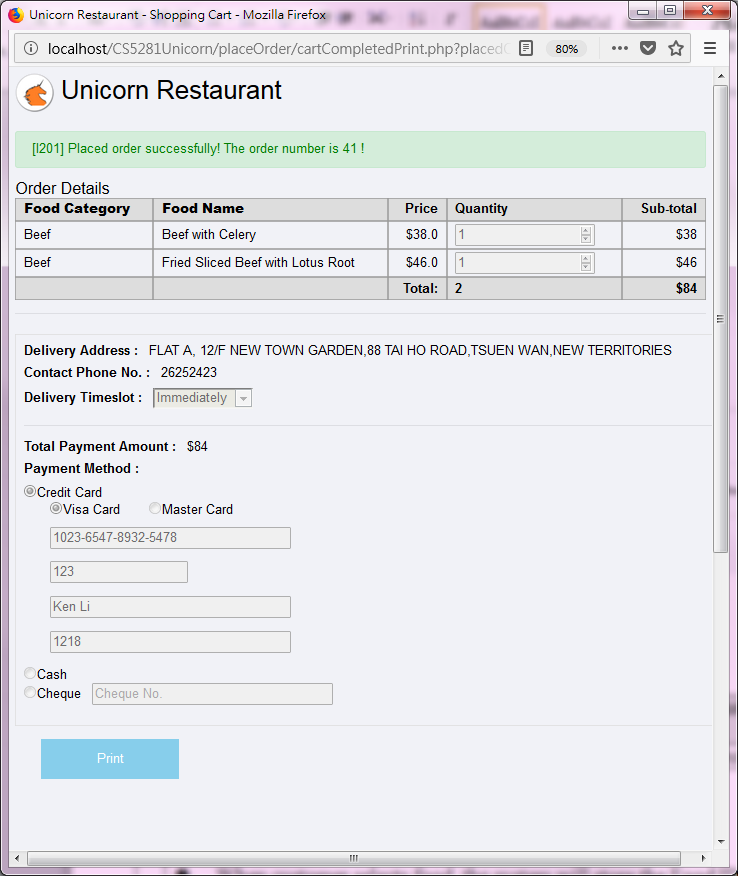
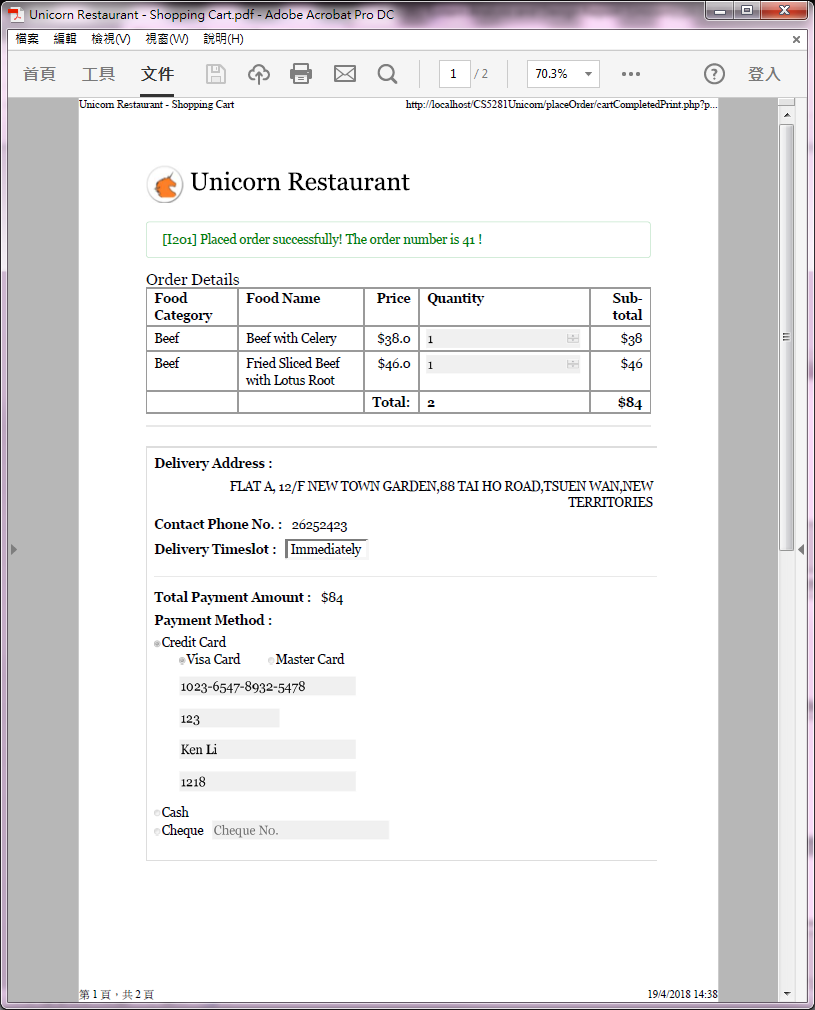
| # | Field Name | Input /  Output | Mandatory /Optional | Format |
| --- | --- | --- | --- | --- |
|  | Food Image | I | M | Food image file path |
| **Validation / Processing**   * Image file will be kept in resources folder and file name will be composed by using food category and food name. * If file path is empty, show error message,   [E908] Invalid file!   * If file size is greater than 1MB, show error message,   [E909] File size must be equal to or less than 1MB!   * If file type is NOT jpg, png, jpeg or gif, show error message,   [E910] Only JPG, JPEG, PNG and GIF files are allowed!   * If there is any system error during file uploading, show error message,   [E911] File upload error, please try again later!   * If selected file for uploading and no food category and food name input value, show error message,   [E907] Food category and food name must be input for uploading food image!   * If file upload successfully, show information message,   [I902] The file {file name} has been uploaded! | | | |
|  | Food category type | I | M | Radio button |
| **Validation / Processing**   * User can select from existing food category or input a new food category. | | | |
|  | Existing food category | I | O | Selection box |
| **Validation / Processing**   * If existing food category radio button is checked and not select food category, show error message,   [E901] Food category must be selected! | | | |
|  | New food category | I | O | Text box |
| **Validation / Processing**   * If new food category radio button is checked and new food category text box is empty, show error message,   [E902] Food category must be input! | | | |
|  | Food name | I | M | Text box |
| **Validation / Processing**   * If Food name is empty, show error message,   [E903] Food name must be input!   * If Food category and food name combination is NOT found in database, show information message,   [I901] This new food name is acceptable!   * If Food name is already existed in the food category, show error message,   [E917] The food name is existing in the food category! | | | |
|  | Price | I | M | Text box |
| **Validation / Processing**   * Food price * If empty, show error message,   [E904] Price must be input!   * If price is NOT numeric, show error message,   [E905] Price must be numeric!   * If price must be greater than 0 and less than 5000, otherwise show error message,   [E906] Price must be greater than 0 and less than 5000! | | | |
|  | Remarks | I | O | Text box |
| **Validation / Processing**   * Remarks for the food. | | | |
|  | Tag | I | O | Tag div section |
| **Validation / Processing**   * Food tag * If there is duplicated tag, show alert message,   Duplicate tags! <tag value>   * If there is any special character, show alert message,   Forbidden character: <tag value>   * Special characters included,   ["~", "!", "@", "#", "$", "%", "^", "&", "\*", "(", ")", "\_", "+", "`", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-", "=", "[", "]", "\\", "|", ";", ":", "'", "\"", ",", "<", ".", ">", "\/", "?"] | | | |
|  | Reset | N/A | N/A | Input Button |
| **Validation / Processing**   * Reset / clear form and go to addFood.php page. | | | |
|  | Add Food | N/A | N/A | Input Button |
| **Validation / Processing**   * Execute client side JavaScript field validations. * Execute server side PHP field validations. * If 1 of field validation rules is failed, stay in addFood.php page and display error message. * If field validation rules are passed,   -. Insert food detail to FOOD table and FOOD\_TAG table if user input any tag description.  -. Stag on addFood.php page if success and show information message,  [I903] Add food successfully!  -. Otherwise, stay in current page and show error message,  [E912] Update user profile failed, please try again later! | | | |

# 17. Exception Handling

|  |  |  |
| --- | --- | --- |
| **#** | **Exception** | **Exception Handling** |
| 1 | Exception.php | All exception occur in php page function will redirect to this Exception.php page and show the exception message. For example, if database cannot be connected, catch exception and go to exception.php. |
| 2 | NoJavaScriptException.php | If user disable Browser’s JavaScript validation function, catch exception and go to NoJavaScriptException.php. |

# 18. Different CSS Style for Web Page and Printing

The following 2 screen layouts show the difference between online web page and printing page. We change to user small icon and mainly use black and white color in printing page includes the color of background and button.

# 19. Cookie Management

If user searches a keyword such as “Beef”, “Beef” will be kept in Cookie and stored for 30 days. Once the user accesses the search page again, the Food Recommendation region will automatically demonstrate the result of searching “Beef” which is getting from Cookie. The cookie name is “UnicornCookie”.

# 20. AJAX Implementation

In Unicorn Restaurant web application, we implement a lot of functions by using AJAX included,

|  |  |  |
| --- | --- | --- |
| **#** | **Function** | **Details** |
| 1 | Search and food recommendation | 1. Call Ajax service to get all food category for preparing the Food Category selection box. 2. Call Ajax service to get food name and food tag for preparing the Food Name autocomplete searching item list. 3. Call Ajax service when user select a category or input a keyword, the search result including all dishes of a category or dish name with keyword will be displayed automatically. 4. Call Ajax service to get recommended dish and display in recommendation region. 5. Call Ajax service to get food detail and group by food category. Display these food category grouping in the home page. |
| 2 | Sign Up | 1. Call Ajax service to check if user inputted User ID or email existed in database. |
| 3 | User Profile | 1. Call Ajax service to check if user inputted User ID or email existed in database. |
| 4 | Notification | 1. Call Ajax service to load all notification from database. 2. Call Ajax service to update notification status when user click on the notification subject. |
| 5 | Add Food | 1. Call Ajax service to check if user inputted Food Category and Food Name combination existed in database. |

# 21. Database Design

21.1 Table Definition

**USER**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| USER\_ID | Unique USER\_ID.  Primary key and index field. | X(50) | Y |
| SEX | 'F' – Female; 'M' – Male | X(1) | N |
| PRIVILEGE | 'A' – Admin; 'U' – Normal User | X(1) | Y |
| ENG\_SURNAME | Surname in English | X(50) | Y |
| ENG\_MIDDLE\_NAME | Middle name in English | X(50) | Y |
| ENG\_NAME | English Christian Name | X(50) | N |
| EMAIL | Lotus notes account | X(100) | Y |
| TEL | Telephone number | X(50) | Y |
| ADDRESS\_1 | Resident address part 1 | X(100) | Y |
| ADDRESS\_2 | Resident address part 2 | X(100) | Y |
| ADDRESS\_3 | Resident address part 3 | X(100) | Y |
| ADDRESS\_4 | Resident address part 4 | X(100) | Y |
| LAST\_LOGIN\_DATE | Last login system date time | DATETIME | N |
| RESET | Force user to change new password (Y/N) | X(1) | N |
| LOCKED | A/C being locked after 3 attempts of invalid login (Y/N) | X(1) | N |
| PASSWORD | Encrypted current password | X(100) | Y |
| REG\_TOKEN | Registration token used to verify user registration | X(100) | N |
| EFFECT\_DATE | Effective date of defined user account | DATETIME | Y |
| EXPIRY\_DATE | Expiry date of defined user account. If this date is passed date, then it means that the user account is logically deleted. | DATETIME | N |
| REMARK | Additional information of the user account. | X(1000) | N |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**FOOD**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| FOOD\_ID | Unique FOOD\_ID.  Primary key and index field. | INT(11) | Y |
| FOOD\_CATEGORY | 'LU' – Shandong; 'SU' – Jiangsu; 'HUI' – Anhui; 'ZHE' – Zhejiang; 'MIN' – Fujian; 'CHUAN' – Sichuan; 'XIANG' – Hunan; 'YUE' - Canton | X(20) | Y |
| FOOD\_NAME | Food name in English | X(100) | Y |
| AVAILABLE | 'Y' – Available for sale, 'N' – Unavailable for sale | X(1) | Y |
| PRICE | Food price in Hong Kong dollar | DECIMAL(6,1) | Y |
| DISCOUNT | Discount percentage of food price | INT(3) | N |
| DISCOUNT\_EFFECT\_DATE | Discount effective date | DATETIME | N |
| DISCOUNT\_EXPIRY\_DATE | Expiry date of defined food discount. If this date is passed date, then it means that the food discount is expired and stopped. | DATETIME | N |
| IMG\_PATH | Food image relative path | X(100) | N |
| REMARK | Additional information of the user account. | X(1000) | N |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**FOOD\_TAG**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| TAG\_ID | Unique TAG\_ID.  Primary key and index field. | INT(11) | Y |
| FOOD\_ID | Unique FOOD\_ID.  Primary key and index field. | INT(11) | Y |
| TAG\_DES | Tag description | X(100) | Y |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**ORDER**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| ORDER\_ID | Unique ORDER\_ID.  Primary key and index field | INT(11) | Y |
| USER\_ID | User ID who raise the order | X(50) | Y |
| STATUS | 'OS01' - Outstanding; 'OS11' - Pending Settlement; 'OS21' - Settled; 'OS31' – Delivered; 'OS41' - Expired | X(4) | Y |
| DELIVERY\_TIMESLOT | ‘DT01’ – Deliver Immediately; ‘DT11’ – Deliver 1 hour after; ‘DT21’ – Deliver 2 hour after; | X(4) | Y |
| ORDER\_EFFECT\_DATE | Order effective date | DATETIME | N |
| ORDER\_EXPIRY\_DATE | Expiry date of order. If this date is passed date, then it means that the order is expired and invalid | DATETIME | N |
| TOTAL\_PAYMENT\_AMT | Total payment amount of the order | DECIMAL(7,1) | N |
| TOTAL\_DISCOUNT\_AMT | Total discount amount of the order | DECIMAL(7,1) | N |
| PAYMENT\_DATE | Payment settlement date | DATETIME | N |
| PAYMENT\_CHANNEL | 'CA' – Cash; 'CR' – Credit Card; 'CH' - Cheque | X(2) | N |
| CREDIT\_CARD\_TYPE | 'MA' – Master Card; 'VS' – Visa Card | X(2) | N |
| CREDIT\_CARD\_NO | Credit card number | X(16) | N |
| CREDIT\_CARD\_SECURITY\_CODE | Credit card security code | X(3) | N |
| CREDIT\_CARD\_HOLDER\_NAME | Credit card holder name | X(50) | N |
| CREDIT\_CARD\_EXPIRY\_DATE | Credit expiry date in format MM/YY | X(4) | N |
| CHEQUE\_NO | Cheque number | X(12) | N |
| REMARK | Additional information of the user account | X(1000) | N |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**ORDER\_DETAIL**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| ORDER\_ID | Primary key of ORDER table | INT(11) | Y |
| FOOD\_ID | Primary key of FOOD table | INT(11) | Y |
| QTY | Quantity | INT(100) | Y |
| PAYMENT\_AMT | Payment amount for the quantity of food | DECIMAL(6,1) | N |
| DISCOUNT\_AMT | Discount amount for the quantity of food | DECIMAL(6,1) | N |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**SYS\_CONFIG**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| CONFIG\_ID | Unique CONFIG\_ID.  Primary key and index field | INT(11) | Y |
| CONFIG\_CATEGORY | Configuration category | X(50) | Y |
| CONFIG\_NAME | Configuration name | X(50) | Y |
| CONFIG\_VALUE | Configuration value | X(1000) | Y |
| CONFIG\_EFFECT\_DATE | Configuration effective date | DATETIME | Y |
| CONFIG\_EXPIRY\_DATE | Expiry date of configuration. If this date is passed date, then it means that the configuration is expired and invalid | DATETIME | N |

**NOTIFICATION**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| NOTIFICATION\_ID | Unique NOTIFICATION\_ID,  Primary key and index field | INT(11) | Y |
| TYPE | ‘NT01’ – Account Activation Success; ‘NT11’ – Order Confirmation; ‘NT21’ - Advertisement | X(4) | Y |
| SUBJECT | Notification subject | X(50) | Y |
| CONTENT | Notification content | X(100) | Y |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

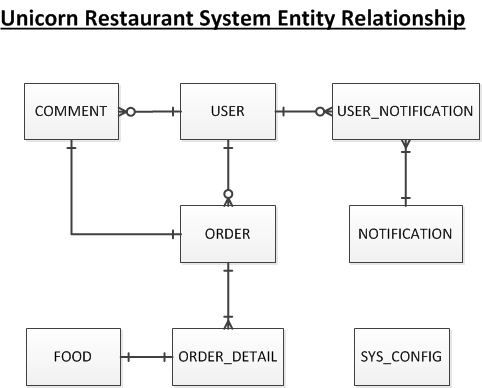
**USER\_NOTIFICATION**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| NOTIFICATION\_ID | Primary key of NOTIFICATION table | INT(11) | Y |
| USER\_ID | Primary key of USER table | X(50) | Y |
| STATUS | ‘NS01’ – Unread; ‘NS11’ – Read; ‘NS21’ – Housekeep | X(4) | Y |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

**COMMENT**

| Data Item | Description | Format | Mandatory |
| --- | --- | --- | --- |
| COMMENT\_ID | Unique COMMENT\_ID.  Primary key and index field | INT(11) | Y |
| USER\_ID | User ID who raise the order | X(50) | Y |
| ORDER\_ID | Primary key of ORDER table | INT(11) | Y |
| RATING | Customer rating for the order | INT(11) | Y |
| CONTENT | Customer comment | X(1000) | Y |
| CREATE\_DATE | Record creation date | DATETIME | Y |
| UPDATE\_DATE | Record update date | DATETIME | Y |

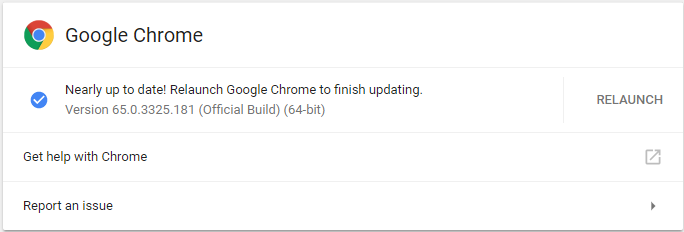
22.2 ER Diagram



# 23. Browser Capability Testing

Unicorn Restaurant web application has been fully tested in 3 main stream web browsers included Google Chrome, Firefox and Internet Explorer. The web browsers versions are list as below,

Google Chrome version: 65.0.3325.181 (Official Build) (64-bit)



Firefox version: Firefox Quantum 59.0.2 (64-bit)

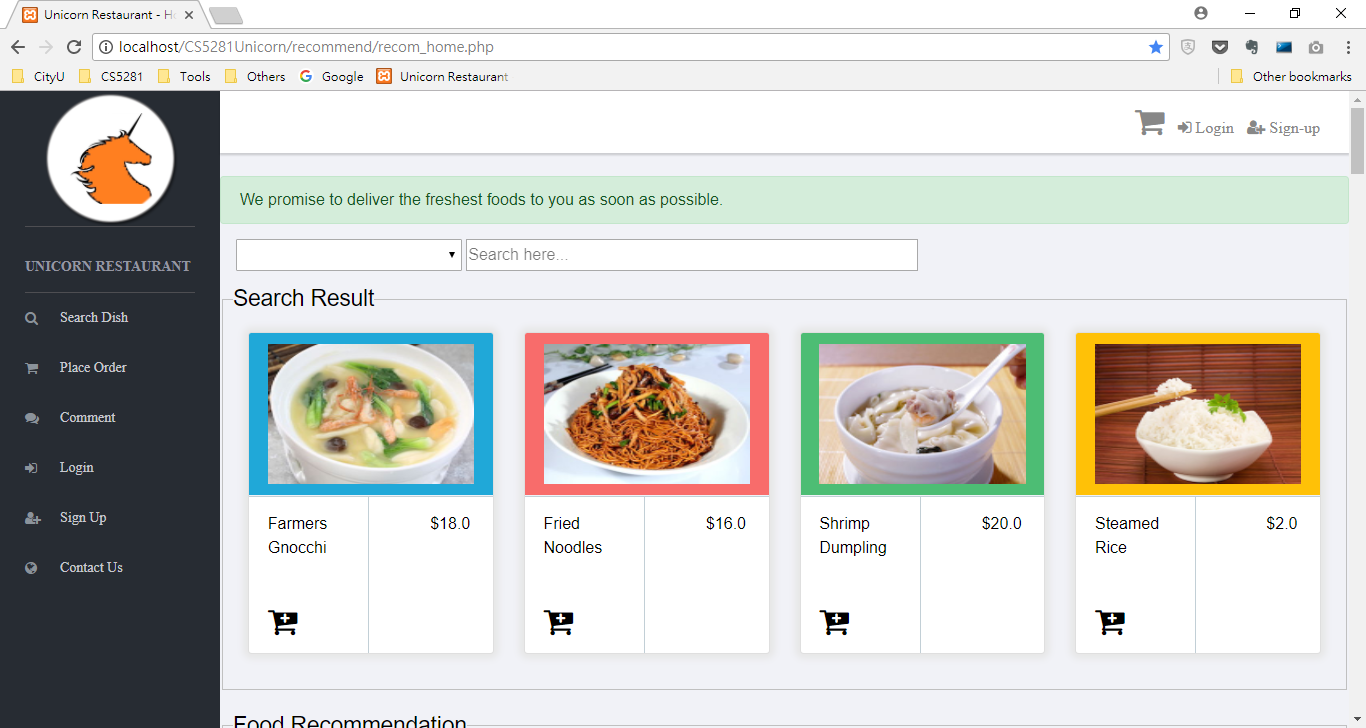


Internet Explorer version: 11.371.16299.0

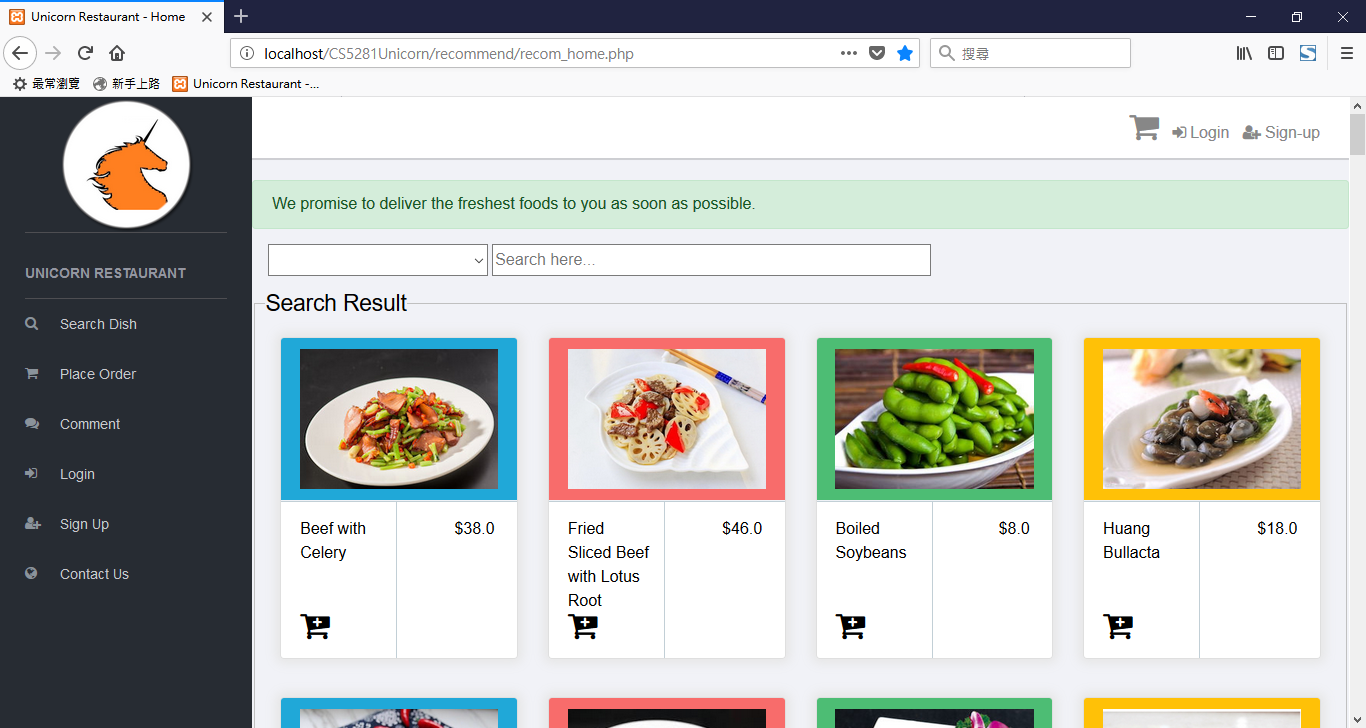


All screen layouts are aligned in these 3 web browser. For details, please refer to Appendix I Browser Capability Testing Result. For example, the Search and Food Recommendation page shown below,

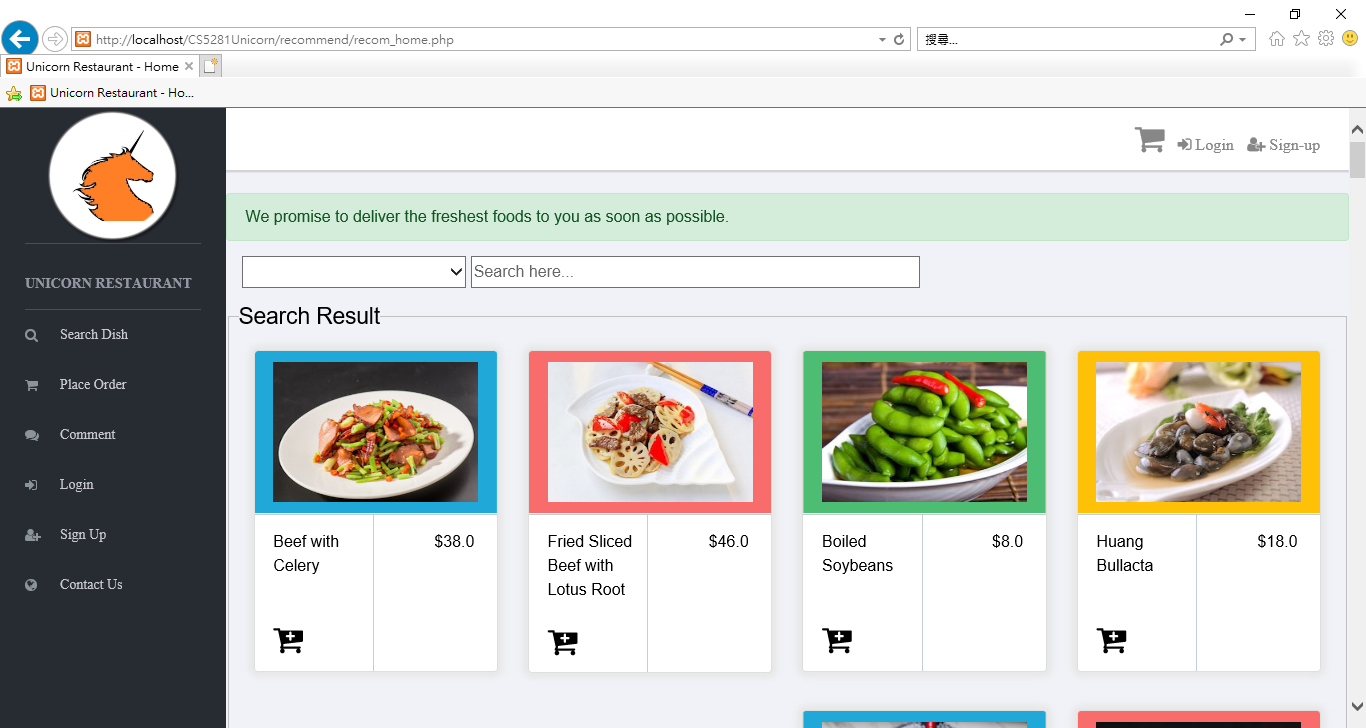
Google Chrome



Firefox



Internet Explorer

****

# 24. Functional Testing

All functions in Unicorn Restaurant web application are fully tested. The testing plan, test result and bug fixing report are attached in Appendix II Functional Testing Report and Bug Fixing Result.

The following section is the summary of the testing plan and testing result.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1-level Module** | **2-level Module** | **Pre-condition** | **Test case (steps)** | **Expected results** | **Actual results** | **Execution results** | **Developer Response** |
| Search\_Category | null value check | 1. select "All" | click searchbox press "Enter" | display all dishes | display nothing | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Search\_Category | proper value check | 1. select "Beef" | select "Beef" category | display all dishes under "Beef" category | display all dishes under "Beef" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search\_Category | proper value check | 1. select "Pork" | select "Pork" category | display all dishes under "Pork" category | display all dishes under "Pork" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search\_Category | proper value check | 1. select "Staple Des" | select "Staple Des" category | display all dishes under "Staple Des" category | display all dishes under "Staple Des" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search\_Category | proper value check | 1. select "Soup" | select "Soup" category | display all dishes under "Soup" category | display all dishes under "Soup" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "DimSum" | select "DimSum" category | display all dishes under "DimSum" category | display all dishes under "DimSum" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Seafood" | select "Seafood" category | display all dishes under "Seafood" category | display all dishes under "Seafood" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Drinks" | select "Drinks" category | display all dishes under "Drinks" category | display all dishes under "Drinks" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Vegetable" | select "Vegetable" category | display all dishes under "Vegetable" category | display all dishes under "Vegetable" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Cold Dish" | select "Cold Dish" category | display all dishes under "Cold Dish" category | display all dishes under "Cold Dish" category | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "All" | select "All" click searchbox input "a" | display food name with letter "a" in real time display food block with food name with "a" in real time | display food name with letter "a" in real time display food block with food name with "a" in real time | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "All" | select "All" click searchbox input "z" | display food name with letter "z" in real time display food block with food name with "z" in real time  or  display [E302] No results! | display [E303] No results! | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Beef" | select "Beef" click searchbox input "d" | display food name with letter "d" in real time display food block with food name with "d" in real time | display food name with letter "d" in real time display food block with food name with "d" in real time | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | proper value check | 1. select "Beef" | select "Beef" click searchbox input "z" | display food name with letter "z" in real time display food block with food name with "z" in real time  or  display [E303] No results! | display [E304] No results! | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | improper value check | 1. select "All" | select "All" click searchbox input "!" | display [E302] No results! | display [E302] No results! | PASS |  |
|  |  |
|  |  |
|  |  |
| Search | improper value check | 1. select "All" | select "All" click searchbox input "1" | display [E302] No results! | display [E302] No results! | PASS |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1-level Module** | **2-level Module** | **Pre-condition** | **Test case (steps)** | **Expected results** | **Actual results** | **Execution results** | **Developer Response** |
| Add dish | positive logic check | not sign in add a dish to the cart | click the "cart" button within any of the food block | display a movement from the food block to the cart on the top increase the number of the red circle beside the cart | display a not smooth movement from the food block to the cart on the top increase the number of the red circle beside the cart | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Add dish | positive logic check | not sign in add a dish to the cart | click the "cart" button | enter the "Order Details" page | enter the "Order Details" page | PASS |  |
|  |  |
|  |  |
|  |  |
| Add dish | positive logic check | signed in add a dish to the cart | click the "cart" button within any of the food block | display a movement from the food block to the cart on the top increase the number of the red circle beside the cart | display a not smooth movement from the food block to the cart on the top increase the number of the red circle beside the cart | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Add dish | positive logic check | signed in add a dish to the cart | click the "cart" button | enter the "Order Details" page | enter the "Order Details" page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | 1. input Arabic number(or other wrong format like symbols) in creditcard name field | click "submit" botton | display error information beside name field | no error occured and enter to payment page | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Place order | usabality check | 1.input several wrong message in some different text fields | click "submit" botton | display all the error information at a time to facilitate user operation | error information will occur line-by-line everytime user click "submit" button | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | 1.input negative number in dish quantity fields | click "submit" botton | display error information beside quantity field | not allow the submission and display error information | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | 1.input 0 in both quantity fields | click "submit" botton | remind users that total quantity should not be 0 | not allow the submission and remind the users | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | click "go to update contact information" while logging out | click "submit" botton | not allow the change of personal information and remind users to login | display "please login first" to remind the users | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input letters in the "credit card No" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input punctuation in the "credit card No" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input character in the "credit card No" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input less or more than 16 digits number in the "credit card No" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input 16 digits number in the "credit card No" text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input letters in the "credit card CVV" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input punctuation in the "credit card CVV" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input character in the "credit card CVV" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input more/less than 3 numbers in the "credit card CVV" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input 3 numbers in the "credit card CVV" text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input letters in the "credit card Holder Name" text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input punctuation in the "credit card Holder Name" text field | click "submit" botton | payment failed and remind users to input "correct message" | go to the order confirmation page without any information | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input character in the "credit card Holder Name" text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input letters in the "credit card Expiry Data" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input punctuation in the "credit card Expiry Data" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input character in the "credit card Expiry Data" text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input right format and reasonable number in the "credit card Expiry Data" text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input letters in the "Cheque No." text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input punctuation in the "Cheque No." text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | improper value check | input character in the "Cheque No." text field | click "submit" botton | payment failed and remind users to input "correct message" | display wrong message format information and block the connection | PASS |  |
|  |  |
|  |  |
|  |  |
| Place order | proper value check | input right format and reasonable number in the "Cheque No." text field | click "submit" botton | connection succeed and go to next page | connection succeed go to order confirmation page | PASS |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1-level Module** | **2-level Module** | **Pre-condition** | **Test case (steps)** | **Expected results** | **Actual results** | **Execution results** | **Developer Response** |
| Sign up | null value check | 1. leave all required fields blank | click "sign up" botton | display error beside each required field | display error beside each required field | PASS |  |
|  |  |
|  |  |
|  |  |
| Sign up | improper value check | 1. input less than 5 characters in Username | click "sign up" botton | error of username field changes to "invalid username", other errors remain. | 2 conflict messages occor, one is accepted the other is error | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Sign up | improper value check | 1. input more than 30 characters in Username | click "sign up" botton | error of username field changes to "invalid username", other errors remain. | not allow to input more than 30 char for userID | PASS |  |
|  |  |
|  |  |
|  |  |
| Sign up | improper value check | 1. input only number characters in Username | click "sign up" botton | error of username field changes to "UserID must be in English letters", other errors remain. | 2 conflict messages occor, one is accepted the other is error | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Sign up | proper value check | 1. input letter-and-number characters in Username, length in [5,30] | click "sign up" botton | error of username field hidden, other errors remain. | error of username field hidden, other errors remain. | PASS |  |
|  |  |
|  |  |
|  |  |
|  | improper value check | 1. input invalid value for Password & Repeat Password(e.g. applkie001@) 2. input valid value for other fields | click "sign up" botton | invalid password error occurs | invalid password error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | improper value check | 1. input valid value for Password(e.g.!@QW12qw), but miss one char in Repeat Password(e.g.!@QW12q) 2. input valid value for other fields | click "sign up" botton | password mismatch error occurs | password mismatch error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. leave Surname field blank 2. fill other fields with valid values | click "sign up" botton | missing surname error occurs | missing surname error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. leave Middlename field blank 2. fill other fields with valid values | click "sign up" botton | missing middlename error occurs | missing middlename error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. leave Contact Phone field blank 2. fill other fields with valid values | click "sign up" botton | missing contact phone error occurs | missing contact phone error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. fill Contact Phone field with invalid value(e.g. 123455) 2. fill other fields with valid values | click "sign up" botton | contact phone invalid error occurs | invalid phone No. can be accepted, no error occurs | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. leave Terms and conditions field unchecked 2. fill other fields with valid values | click "sign up" botton | unacception error occurs | unacception error occurs | PASS |  |
|  |  |
|  |  |
|  |  |
|  | required filed check | 1. leave any of address delivery field blank 2. fill other fields with valid values | click "sign up" botton | address required error occurs | address required error occurs | PASS |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1-level Module** | **2-level Module** | **Pre-condition(pre-steps)** | **Test case(Execution steps)** | **Expected results** | **Actual results** | **Execution results** | **Developer Response** |
| Log in | null value check | 1. leave both Username & Password fields blank | click "Log in" botton | display error beside each required field | [E601] User ID / Email must be input! [E602] Password must be input! | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | null value check | 1. only input any length characters in Username | click "Log in" botton | only display error beside Password field | display " [E602] Password must be input!" | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | null value check | 1. only input a correct value in Password field and leave Username filed blank | click "Log in" botton | only display error beside Username field | display " [E601] User ID / Email must be input!" | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input correct Username 2. input only number characters in Password | click "Log in" botton | display error "invalid password" | display "[E602] Password must be input!" | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input correct Username 2. input only letter characters in Password | click "Log in" botton | display error "invalid password" | display "[E603] Password length must be 8 - 20" or " [E604] Password must contain at least 2 upper case "characters! | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input correct Usernam 2. input incorrect Password that all the characters are special symbol | click "Log in" botton | display error "invalid password" | display "[E603] Password length must be 8 - 20" or " [E604] Password must contain at least 2 upper case "characters! | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input correct Usernam 2. input correct Password but incorrect case | click "Log in" botton | display error "invalid password" | display"[E608] Login failed, please try again!" | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input incorrect Usernam 2. input correct Password | click "Log in" botton | display error "username is unregistered" | display"[E608] Login failed, please try again!" | PASS |  |
|  |  |
|  |  |
|  |  |
| Log in | improper value check | 1. input correct Usernam but incorrect case 2. input correct Password | click "Log in" botton | display error "username is not case-insensitive" | log in success | NOT PASS | FIXED |
|  |  |
|  |  |
|  |  |
| Log in | abnormal situation check | 1. input correct Usernam 2. input correct Password | click "refresh" botton | Username and Password return blank | Username and Password return blank | PASS |  |
|  |  |
|  |  |
|  |  |

# 25. System Configuration

|  |  |
| --- | --- |
| **#** | **Item** |
| 1 | Php version 5.6.8 |
| 2 | MySQL version 5.6 |
| 3 | Windows 7 or 10 |
| 4 | RAM 8.0GB  CPU 3.6GHz |
| 5 | Development IDE:   1. Notepad++ 2. Eclipse Oxygen.1a Release (4.7.1a) (Optional) |
| 6 | Testing email server simulator,  FakeSMTP |

# 

# 26. Environment Setup

|  |  |
| --- | --- |
| **Step#** | **Item** |
| 1 | Download XAMPP 5.6.8 (xampp-win32-5.6.8-0-VC11-installer.exe) from https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/5.6.8/  XAMPP is an open source package which contains Apache Web Server, MySQL Database, Tomcat Application and Php container. |
| 2 | Download HeidiSQL 9.5 () from https://www.heidisql.com/download.php?download=installer  HeidiSQL is used to connect to MySQL database server for manipulating application data. |
| 3 | Download Notepad++ from  https://notepad-plus-plus.org/download/v7.5.4.html |
| 4 | Install XAMPP by double click on the XAMPP installer.    Apache Web Server will listen on port 80. |
| 5 | Install HeidiSQL by double click on the HeidiSQL installer.  MySQL Database listens on port 3306 by default.  New a MySQL connection with IP:127.0.0.1 (local host) and port number is 3306. |
| 6 | Install FakeSMTP to simulate SMTP email server in localhost.  a. Download FakeSMTP from the following link,  http://nilhcem.com/FakeSMTP/download.html  b. Unzip the FakeSMTP-latest.zip   1. Double click on the fakeSMTP-2.0.jar      1. Start up the FakeSMTP to listen on port 25 to receive email,      1. Allow the firewall connection in port 25 if you are Windows OS 7 or 10,      1. Backup php.ini in php folder of XAMPP,   C:\xampp\php\php.ini   1. Update the following items in php.ini to enable email sending,   -. SMTP = 127.0.0.1  -. smtp\_port = 25  -. sendmail\_from = cs5281unicorn\_admin@unicorn.com  -. sendmail\_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"  -. mail.log = "C:\xampp\php\logs\php\_mail.log"   1. Backup sendmail.ini in sendmail folder of XAMPP,   C:\xampp\sendmail\sendmail.ini   1. Update the following items in sendmail.ini to enable email sending,   -. smtp\_server=127.0.0.1  -. smtp\_port=25  -. error\_logfile=error.log  -. debug\_logfile=debug.log |
| 7 | Server log file path,  C:\xampp\sendmail\debug.log  C:\xampp\php\logs\php\_mail.log  C:\xampp\apache\logs\error.log  C:\xampp\apache\logs\access.log  C:\xampp\mysql\data\mysql\_error.log  C:\xampp\sendmail\error.log  C:\xampp\xampp-control.log |

# 27. Application Deployment

|  |  |
| --- | --- |
| Step# | Item |
| 1 | Login by using root user id.    Create database and tables.    Run SQL to create database, tables and static table records. |
| 2 | Deploy application to Apache Web Server.  Prepare HTML, CSS and PHP and place in Apache root folder such as  D:\xampp\htdocs\CS5281Unicorn\index.php    And try to access the page in browser, |
|  |  |

# 28. Source Control

Application source files will be kept in GitHub repositories in below path,

https://github.com/KenLiHK/Unicorn

# APPENDICES

# Appendix I Browser Capability Testing Result



# Appendix II Functional Testing Report and Bug Fixing Result

************************

# Appendix III Development Guide and Reference

# References

|  |  |
| --- | --- |
| waimai.meituan | Retrieved from  *http://waimai.meituan.com* |
| PHP.NET. | Retrieved from  *http://php.net/* |
| XAMPP. | Retrieved from  *https://www.apachefriends.org/index.html* |
| HeidiSQL. | Retrieved from  *https://www.heidisql.com/download.php?download=installer* |
| SOURCEFORGE. | Retrieved from  *https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/5.6.8/* |
| Notepad++ | Retrieved from  https://notepad-plus-plus.org/download/v7.5.4.html |
| Eclipse Oxygen.1a Release (4.7.1a) | Retrieved from  http://www.eclipse.org/downloads/ |
| GibHub | Retrieved from  https://github.com/KenLiHK/Unicorn |
| Chinese food picture | Retrieved from  https://food.ndtv.com/lists/10-most-popular-chinese-dishes-740725  http://www.delish.com/cooking/recipe-ideas/recipes/a51085/mongolian-beef-ramen-recipe/  https://eat24hours.com/Chinese  http://yummyyummyaz.com/  https://www.blogto.com  http://www.kowloonnc.com/  https://www.geocaching.com/geocache/GC57B2P\_chinese-food?guid=3da57501-1d8c-4e44-bce0-bde814e077fa  http://money.cnn.com/2014/12/24/smallbusiness/chinese-food-christmas/index.html |
| icoconvert | Retrieved from  http://icoconvert.com/ |
| Themewagon | Retrieved from  https://themewagon.com |
| CSS Icons | Retrieved from  https://www.w3schools.com/w3css/w3css\_icons.asp |
| Chinese food region | Retrieved from  https://en.wikipedia.org/wiki/Chinese\_cuisine  https://upload.wikimedia.org/wikipedia/commons/d/d8/Cuisines\_of\_China.png |
| Document management | Retrieved from  https://www.dropbox.com |
| FakeSMTP | Retrieve from  http://nilhcem.com/FakeSMTP/download.html |
| Ajax | Retrieve from  https://jquery.com/download/ |
| PHP form validation | Retrieved from  https://www.w3schools.com/php/php\_form\_validation.asp  https://www.w3schools.com/php/php\_form\_required.asp |
| taggingJS | Retrieved from  <https://github.com/sniperwolf/taggingJS> |
| jQuery Cookie | Retrieved from  https://plugins.jquery.com/cookie/ |
| jQuery UI | Retrieved from  https://jqueryui.com/autocomplete/ |