**Web-based Ordering & Ingredient Estimating for Bakery Manufacturer**

Software Design Document

By

**Miss. Nontra Mahachanont 542115026**

**Mr. Parinya Panyanak 542115034**

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr.Phudinan Singkhamfu**

|  |  |  |
| --- | --- | --- |
| **Software Requirement Specification** | | |
| Cross Ref. VSE-29110 | **Coverage Level:** | **Version:** |
|  | V.1.0 | V.1.0 |

|  |  |
| --- | --- |
| **Process Ownership** | **Approving Authority** |
| Web-based Ordering & Ingredient Estimating for Bakery Manufacturer | Aj.Phudinan Singkhamfu |
| **Scope** | **Approved Date** |
| Senior Project | 23 July 2014 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version Number** | **Record Date** | **Prepared/**  **Modified By** | **Reviewed By** | **Change Details** |
| **Documents History** | | | | |
| V.0.1 | 06-05-2014 | NM, PP | NM, PP | - Add Chapter 1  - Add Chapter 2 |
| V.0.2 | 07-05-2014 | NM, PP | NM, PP | - Add Chapter 3  - Fix inappropriate words |
| V.0.4 | 09-05-2014 | NM, PP | NM, PP | - Add Chapter 4  - Fix inappropriate words |
| V.0.6 | 12-05-2014 | NM, PP | NM, PP | - Adjust Chapter 2  - Fix inappropriate words |
| V.1.0 | 24-06-2014 | NM, PP | NM, PP | - Adjust Chapter 2  - Adjust Chapter 3  - Adjust Chapter 4  - Fix inappropriate words |

**\*NM = Nontra Mahachanont**

**\*PP = Parinya Panyanak**

**\*PS = Phudinan Singkhamfu**

**Table of contents**

[**Chapter One | System Architecture** 4](#_Toc392499246)

[**Chapter Two | Detailed Design** 7](#_Toc392499247)

[2.1 Class Diagram 7](#_Toc392499248)

[2.2 Class Diagram Description 9](#_Toc392499249)

[2.2.1 Controller Class 9](#_Toc392499250)

[2.2.2 Model Class 13](#_Toc392499251)

[2.2.3 Service Class 18](#_Toc392499252)

[2.3 Sequence Diagram 19](#_Toc392499253)

[2.3.1 Admin can activate member 19](#_Toc392499254)

[2.3.2 User can login 20](#_Toc392499255)

[2.3.3 User can logoff 21](#_Toc392499256)

[2.3.4 Admin can edit user information 21](#_Toc392499257)

[2.3.5 Admin can delete user 22](#_Toc392499258)

[2.3.6 Admin can view user information 23](#_Toc392499259)

[2.3.7 Admin can create bakery product 24](#_Toc392499260)

[2.3.8 Admin can edit bakery product 25](#_Toc392499261)

[2.3.9 Admin can delete bakery product 26](#_Toc392499262)

[2.3.10 User can view bakery product 27](#_Toc392499263)

[**Chapter Three | Data Architecture** 28](#_Toc392499264)

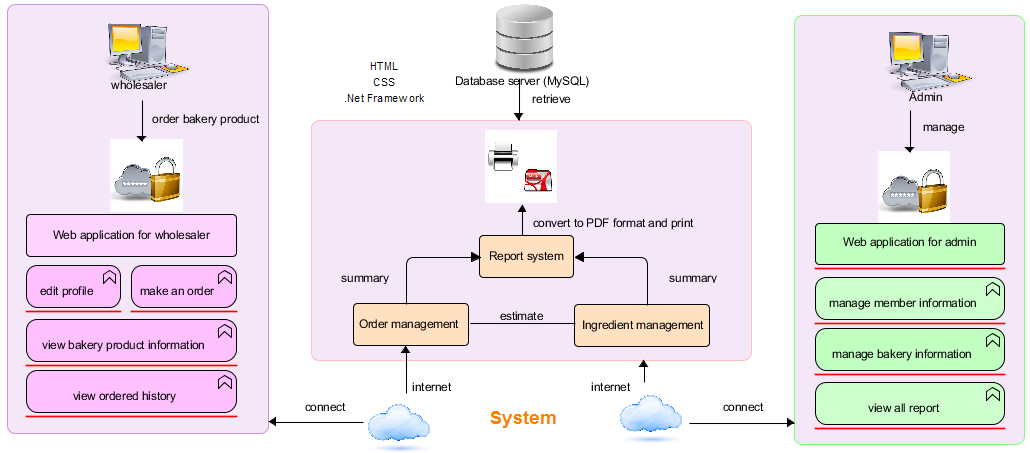
[3.1 ER Diagram 28](#_Toc392499265)

[3.2 ER Diagram Description 29](#_Toc392499266)

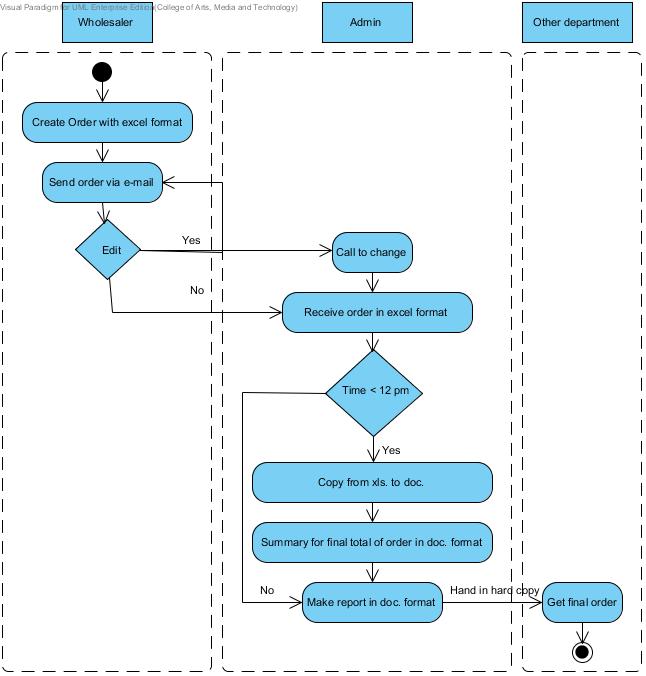
[**Chapter Four | User Interface** 31](#_Toc392499267)

# 

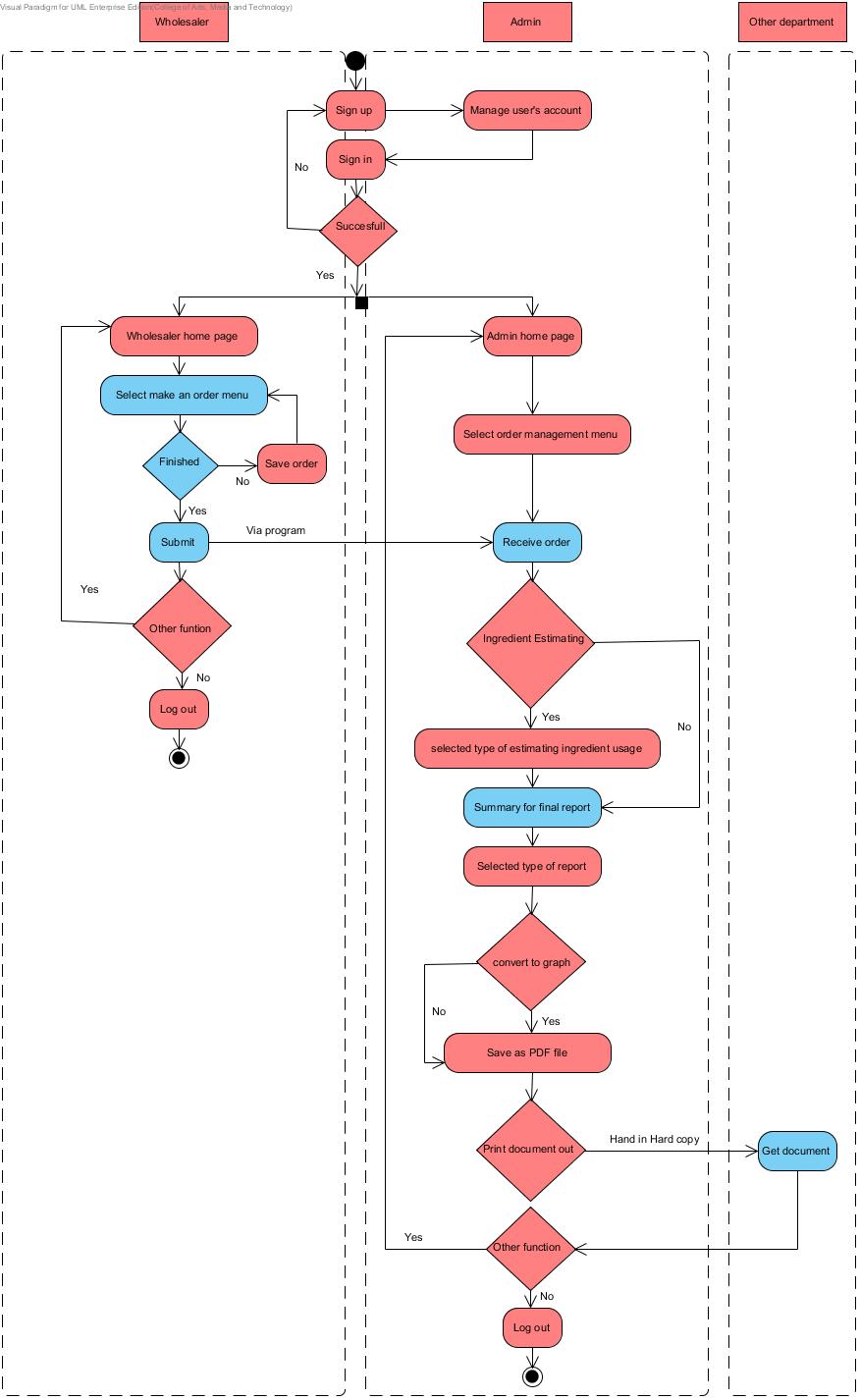
# **Chapter One |** **System Architecture**



**Figure 1: Web-based Ordering & Ingredient Estimating for Bakery Manufacturer**



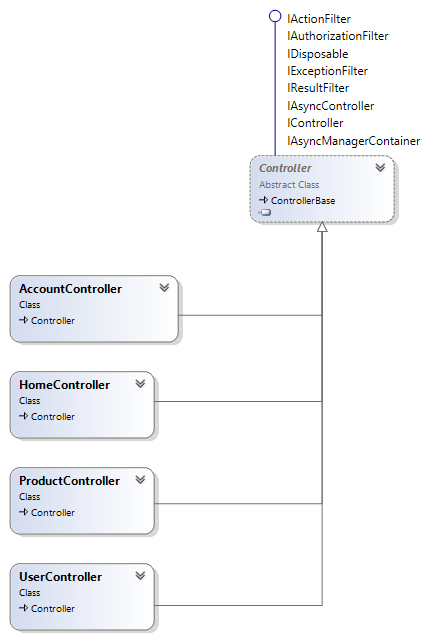
**Figure 2: Flow chart of original workflow**



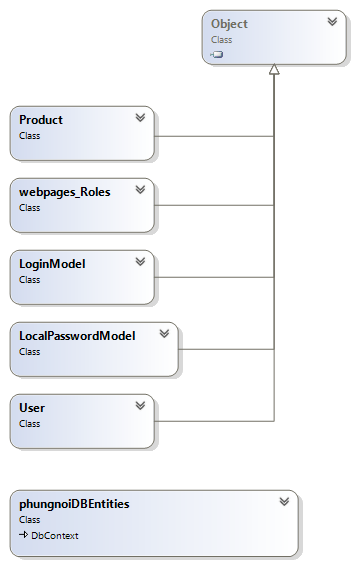
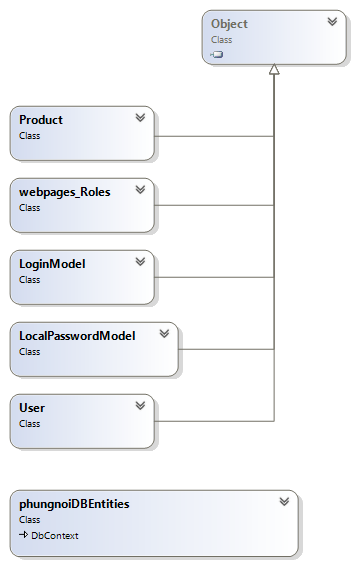
**Figure 3: Flow chart of Web-based Ordering & Ingredient Estimating for Bakery Manufacturer**

# **Chapter Two | Detailed Design**

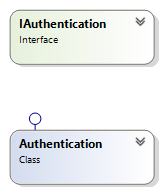
## Class Diagram



**Figure 4: Controller Class Diagram**

****

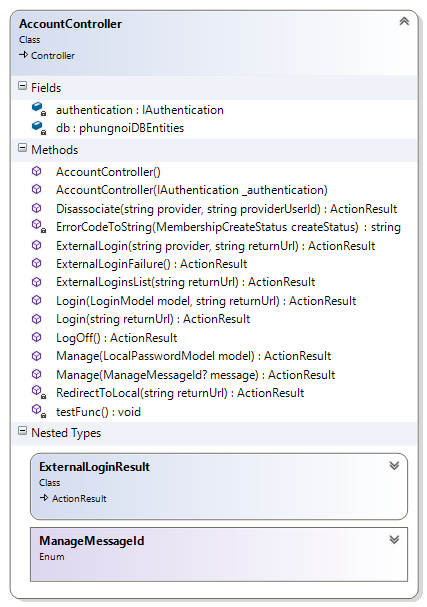
**Figure 5: Model Class Diagram**



**Figure 6: Service Class Diagram**

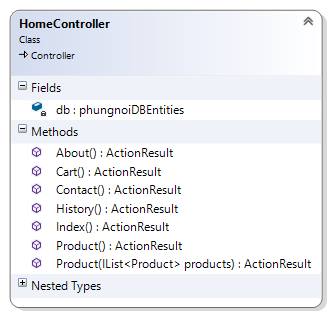
## Class Diagram Description

### 2.2.1 Controller Class



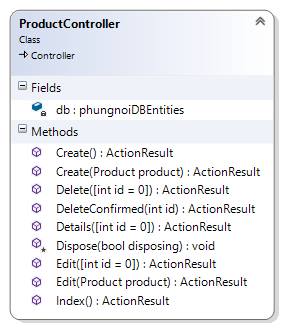
**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Login() | This method uses for logging in to the system. | **Return Type**: ActionResult  **Parameters:** model; LoginModel, returnUrl;String |
| 2 | Logoff() | This method uses for logging out from the system. | **Return Type**: ActionResult |



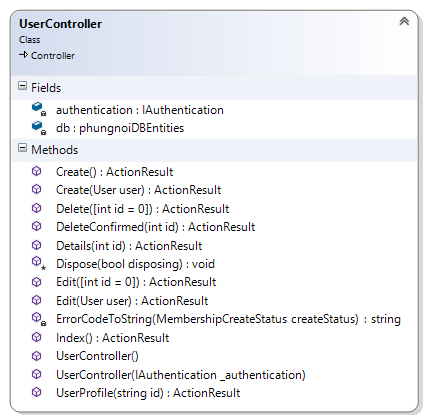
**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | About() | This method uses to call the About page represent. | **Return Type**: ActionResult |
| 2 | Cart() | This method uses to call the Cart page represent. | **Return Type**: ActionResult |
| 3 | Contact() | This method uses to call the Contact page represent. | **Return Type**: ActionResult |
| 4 | History() | This method uses to call the History page represent. | **Return Type**: ActionResult |
| 5 | Index() | This method uses to call the Index page represent. | **Return Type**: ActionResult |
| 6 | Product() | This method uses to call the Product page represent that will be shown list of product. | **Return Type**: ActionResult  **Parameters:** products; List<Product> |



**Method:**

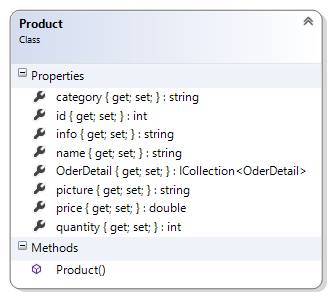
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Create() | This method uses to call the Create page represent. | **Return Type**: ActionResult |
| 2 | Create() | This method used to add the product in to database. | **Return Type**: ActionResult  **Parameters:** product; Product |
| 3 | Delete() | This method uses to call the Delete page represent. | **Return Type**: ActionResult  **Parameters:** id = 0; int |
| 4 | DeleteConfirmed() | This method uses to delete the product into database**.** | **Return Type**: ActionResult  **Parameters:** id; int |
| 5 | Edit() | This method uses to call the Edit page represent. | **Return Type**: ActionResult  **Parameters:** id=0;int |
| 6 | Edit() | This method used to update the product information into database. | **Return Type**: ActionResult  **Parameters:** producr;Product |
| 7 | Details() | This method uses to call the details of the product. | **Return Type**: ActionResult  **Parameters:** int id = 0 |
| 8 | Index() | This method uses to call the index page represent. | **Return Type**: ActionResult |



**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Create() | This method uses to call the Create page represent. | **Return Type**: ActionResult |
| 2 | Create() | This method used to add the product in to database. | **Return Type**: ActionResult  **Parameters:** user; User |
| 3 | Delete() | This method uses to call the Delete page represent. | **Return Type**: ActionResult  **Parameters:** id = 0; int |
| 4 | DeleteConfirmed() | This method uses to delete the product into database**.** | **Return Type**: ActionResult  **Parameters:** id; int |
| 5 | Edit() | This method uses to call the Edit page represent. | **Return Type**: ActionResult  **Parameters:** id=0;int |
| 6 | Edit() | This method used to update the product information into database. | **Return Type**: ActionResult  **Parameters:** user;User |
| 7 | Details() | This method uses to call the details of the product. | **Return Type**: ActionResult  **Parameters:** int id = 0 |
| 8 | Index() | This method uses to call the index page represent. | **Return Type**: ActionResult |
| 9 | UserProfile() | This method uses to call the User Profile page represent. | **Return Type**: ActionResult  **Parameters:** int id = 0 |

### 2.2.2 Model Class



**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Category | This properties uses for storing the Product category into the database. | **Type**: String |
| 2 | Id | This properties uses for storing the product id into the database. | **Type**: Int |
| 3 | info | This properties uses for storing the product information into the database. | **Type**: String |
| 4 | name | This properties uses for storing the product name into the database. | **Type**: String |
| 5 | picture | This properties uses for storing the product picture into the database. | **Type**: String |
| 6 | price | This properties uses for storing the product price into the database. | **Type**: String |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Product() | This method is a constructor of Product. | **Return Type**: ActionResult |

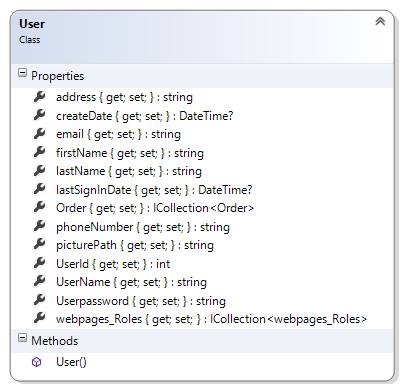


**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | RoleId | This property uses for separating the role of user from the database. | **Type**: Int |
| 2 | RoleName | This property uses for separating the name of user into the database. | **Type**: String |
| 3 | UserProfile | This property uses for storing the product information into the database. | **Type**: Collection <User> |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Webpage\_Roles() | This method is a constructor of Webpage\_Roles. | **Return Type**: ActionResult |



**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | address | This property uses for storing the address of user into the database. | **Type**: String |
| 2 | CreateDate | This property uses for storing the date of activated user into the database. | **Type**: DateTime |
| 3 | firstName | This property uses for storing the first name of user into the database. | **Type**: String |
| 4 | lastName | This property uses for storing the last name of user into the database. | **Type**: String |
| 5 | lastSignInDate | This property uses for storing the last sign in date of user into the database. | **Type**: DateTime |
| 6 | phoneNumber | This property uses for storing the phone number of user into the database. | **Type**: String |
| 7 | picturePath | This property uses for storing the picture of user into the database. | **Type**: String |
| 8 | userId | This property uses for storing the id of user into the database. | **Type**: Int |
| 9 | userName | This property uses for storing the email of user into the database. | **Return Type**: String |
| 10 | userPassword | This property uses for storing the password of user into the database. | **Return Type**: String |
| 11 | Webpage\_roles | This property uses for storing the role of user into the database. | **Return Type**: Collection <Webpage\_roles> |

**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | User() | This method is a constructor of User. | **Return Type**: ActionResult |

## https://fbcdn-sphotos-h-a.akamaihd.net/hphotos-ak-xpf1/v/t34.0-12/10515023_744250222284506_639487683_n.jpg?oh=291a48768e11855d8807d3bd8503ecd5&oe=53B9D1D4&__gda__=1404665161_f1793ba3783c5d251e2cfb1afa7015b9

**Property:**

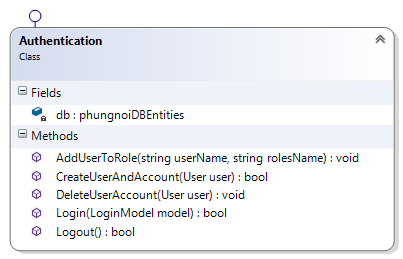
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Password | This property uses for storing the password of user into the database. | **Return Type**: String |
| 2 | UserName | This property uses for storing the username into the database. | **Return Type**: String |
| 3 | RememberMe | This property uses for storing the login into the database. | **Return Type**: Boolean |

## https://fbcdn-sphotos-h-a.akamaihd.net/hphotos-ak-xpf1/v/t34.0-12/10515023_744250222284506_639487683_n.jpg?oh=291a48768e11855d8807d3bd8503ecd5&oe=53B9D1D4&__gda__=1404665161_f1793ba3783c5d251e2cfb1afa7015b9

**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | ConfirmPassword | This property uses for storing the confirm password of user into the database. | **Return Type**: String |
| 2 | NewPassword | This property uses for storing the new password into the database. | **Return Type**: String |
| 3 | OldPassword | This property uses for storing the old password into the database. | **Return Type**: String |

### 2.2.3 Service Class



**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | AddUserToRole() | This method uses to adding a role of user. | **Return Type**: void  **Parameters:** username ;string, rolesName; string |
| 2 | CreateUserAndAccount() | This method uses to creating a new user. | **Return Type**: Boolean  **Parameters:** user; User |
| 3 | DeleteUserAccount() | This method uses to deleting a user from database. | **Return Type**: void  **Parameters:** user; User |
| 4 | Login() | This method uses to logging in before using all functional website. | **Return Type**: Boolean  **Parameters:** model; LoginModel |
| 5 | Logout() | This method uses to logging a user out from the website. | **Return Type**: Boolean |

## 

## 2.3 Sequence Diagram

### 2.3.1 Admin can activate member



**Figure 7: SD-01 Admin can activate user**

User call Create action to UserController.UserController call createUserAndAccount from Authentication to create new user and then select this user, if model is valid add role to user and return to details page.

### 2.3.2 User can login



**Figure 8: SD-02 User can login**

Admin and wholesaler must have to login before using the system. First, both of user type select menu list on the top right of homepage. Then select login to the system, AccountController will be check the model and return Login page UI-01. Then user inputs username and password. When user confirms to logging in, the system will pass data to Authentication. Then AccoutController check validation. If authentication valid will be call action RedirectTolocal in UI-02 or UI-03, separate the type of user from the database. Else the user no permission, the controller will redirect to login page.

### 2.3.3 User can logout

****

**Figure 9: SD-03 User can logout**

Admin and wholesaler can logout from the system. First, both of user type select menu list on the top right of homepage. Then select logout to the system in UI-14. When user confirms logging out, the system will be call action Logout from AccountController to Authentication. Then the system returns logout to AccoutController in UI-01.

### 2.3.4 Admin can edit user information



**Figure 10: SD-04 Admin can edit user information**

User call action edit from UserController.UserController callFirstOrDefault from Dbset<user>to found username of user if id of this username is not equal this id

### 2.3.5 Admin can delete user



**Figure 11: SD-05 Admin can delete user**

### 2.3.6 Admin can view user information



**Figure 12: SD-06 Admin can view user information**

### 

### 2.3.7 Admin can create bakery product



**Figure 13: SD-07 Admin can create bakery product**

Admin can create new bakery product information. First, select product management button on the main page of admin. Then select create new bakery product and input bakery product information in UI-09. When admin confirms addingUser call create action to ProductController and ProductController will check model if valid ProductController call add action to Dbset<Product> to add product and add finish will return to INDEX page.

### 2.3.8 Admin can edit bakery product information



**Figure 14: SD-08 Admin can edit bakery product information**

User call edit action to ProductController and ProductController will check model if model is valid ProductController will call Entry<product> action to update new model and then save change the model and call list of all product to return to index page.

### 2.3.9 Admin can delete bakery product

****

**Figure 15: SD-09 Admin can delete bakery product**

User call DeleteConfired action to Productcontroller and ProductController will call Find action to to find product with product id and return this product back to ProductController then call remove action to remove the product then finish remove ProducteController call list of product and return it.

### 

### 2.3.10 User can view bakery product information



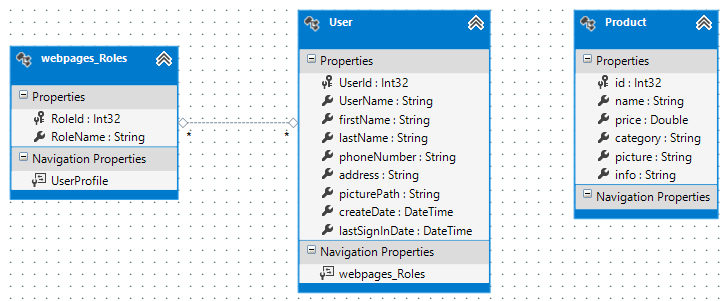
**Figure 16: SD-10 User can view bakery product information**

First, main menu user can view bakery product information by select product bar UI-02 or UI-03, separate by type of users. The system will be call product action to HomeController then check value. If have keyword from userHomeController will call LINQ action to found list of products from Dbset<product> and return list of product UI-12. Else, HomeController will call list of all products form Dbset<product>.

# 

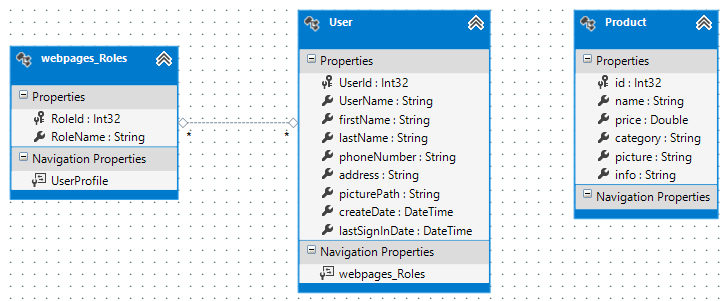
# **Chapter Three | Data Architecture**

## 3.1 ER Diagram



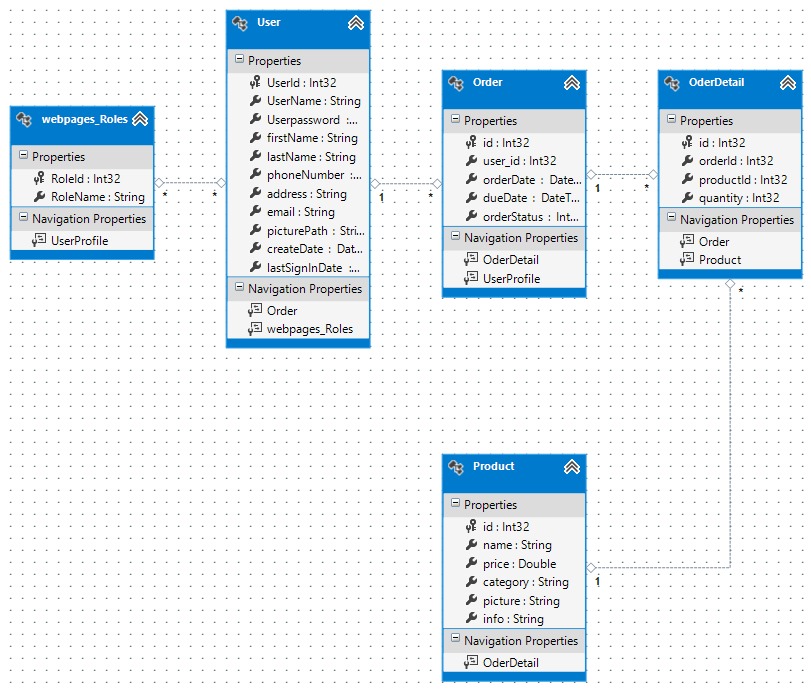
**Figure 13: ER Diagram**

## 3.2 ER Diagram Description

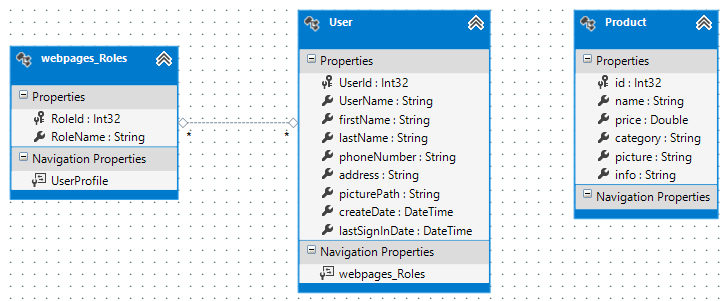


**DB-01: Product**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Extra** |
| id | Int(32) | - Primary Key  - Not Null  - Auto Incremental |
| name | String | - Not Null |
| price | Double | - Not Null |
| category | String | - Not Null |
| picture | String | - Not Null |
| info | String | - Not Null |

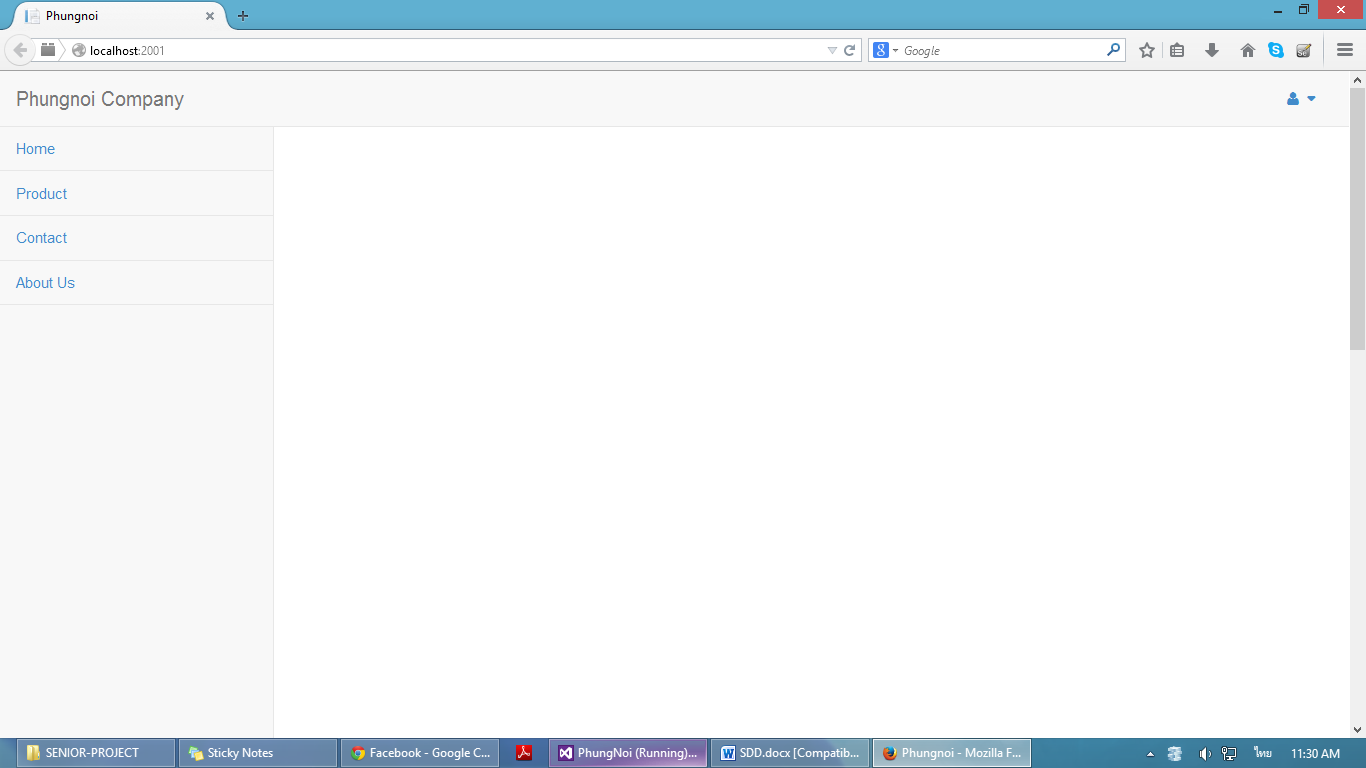
**DB-02: Webpage\_Roles**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Extra** |
| RoleId | Int(32) | - Primary Key  - Not Null |
| RoleName | String | - Not Null |

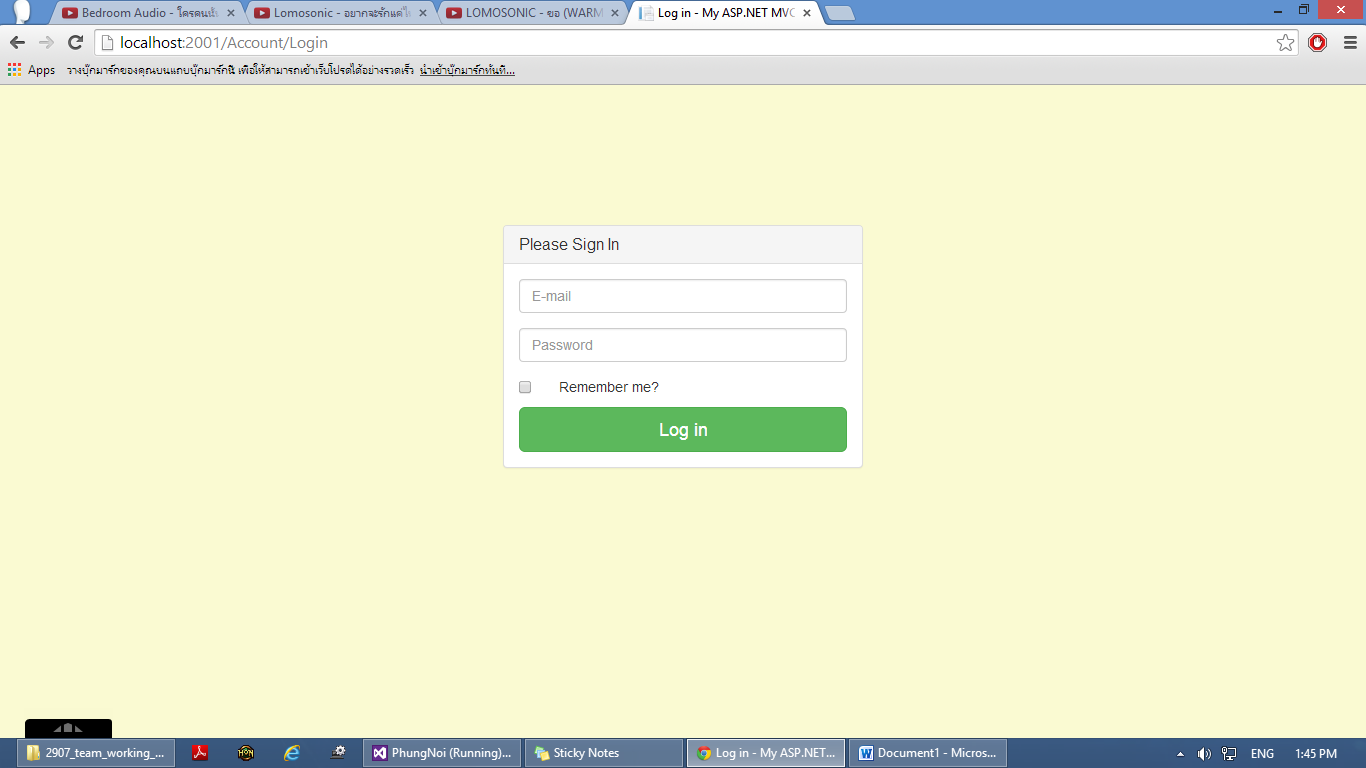
**DB-03: User**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Extra** |
| userId | Int(32) | - Primary Key  - Not Null  - Auto Incremental |
| userName | String | - Not Null |
| firstName | String | - Not Null |
| lastName | String | - Not Null |
| phoneNumber | String | - Not Null |
| address | String | - Not Null |
| picturePath | String | - Not Null |
| createDate | DateTime | - Not Null |
| lastSigninDate | DateTime | - Not Null |

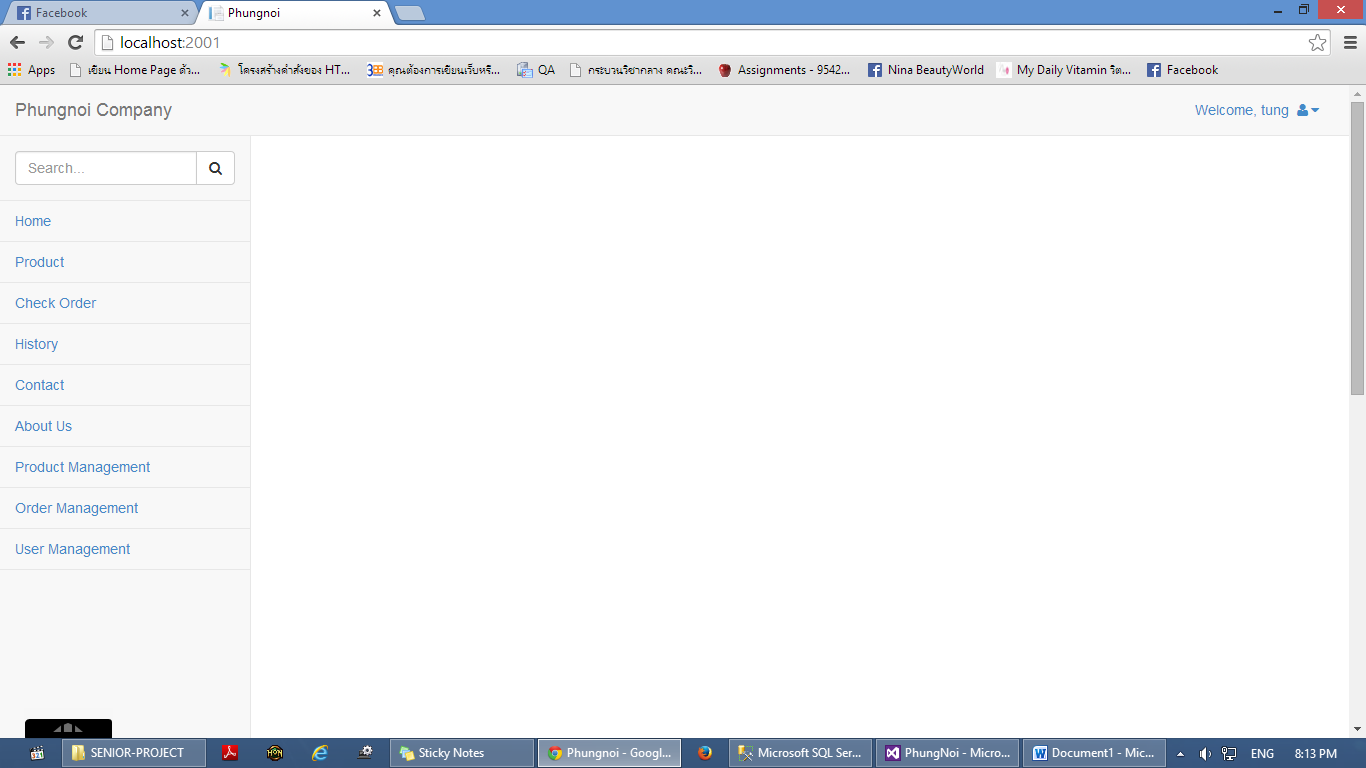
# **Chapter Four | User Interface**



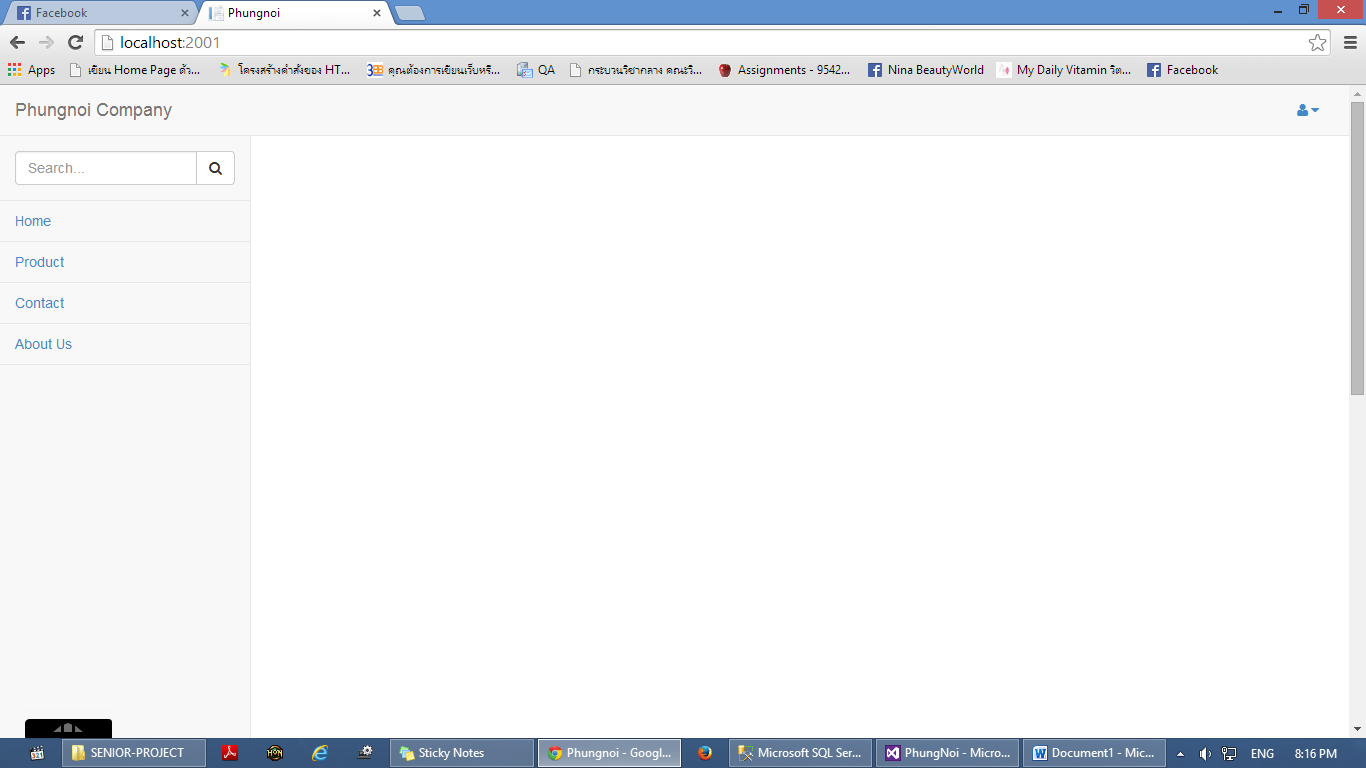
Home



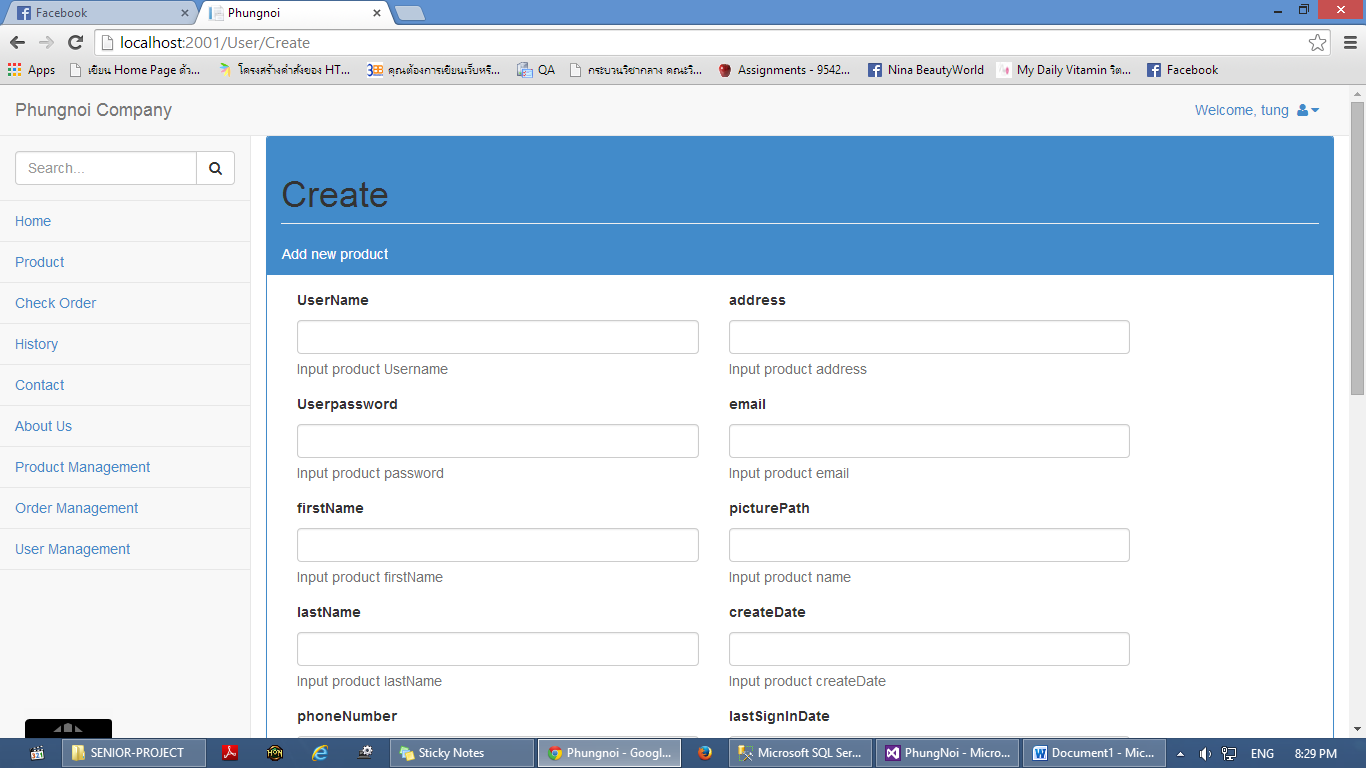
**Figure 14:** UI-01 Login page

****

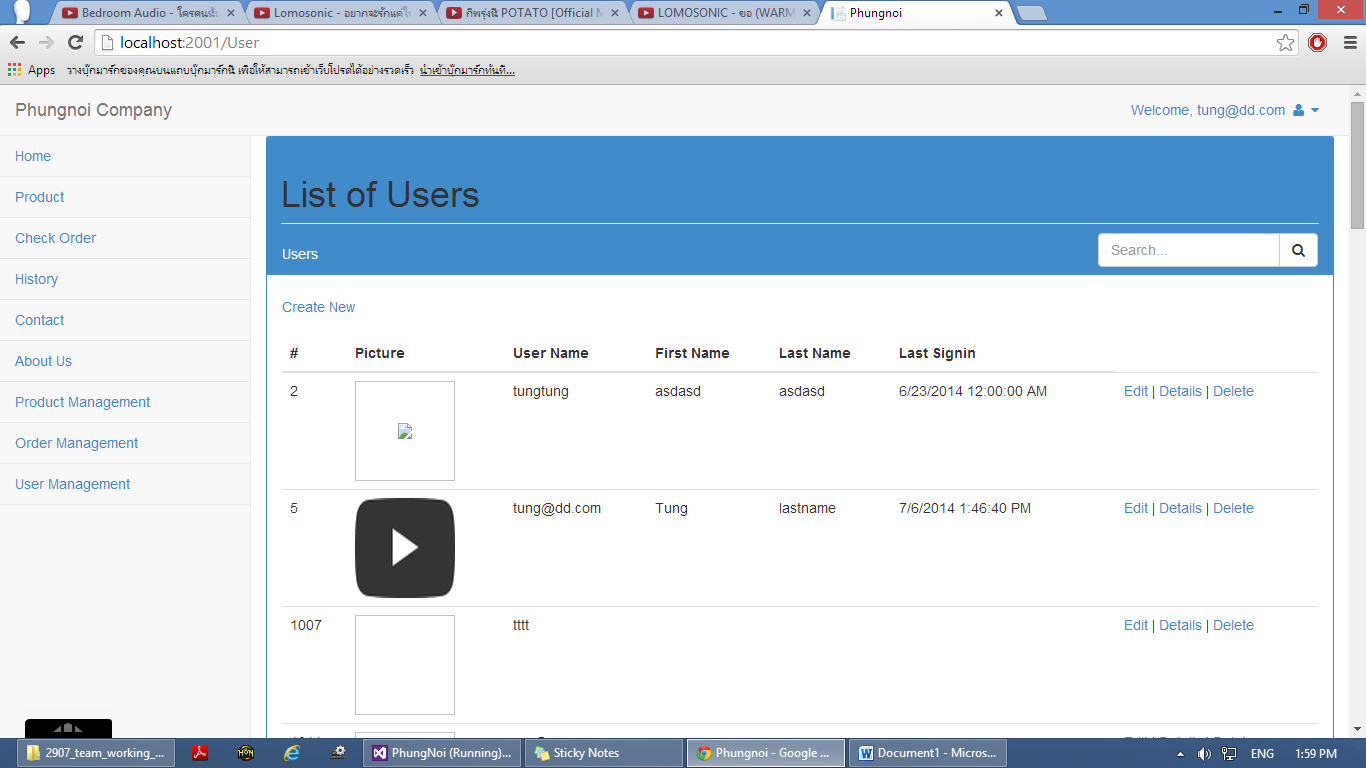
**Figure 15:** UI-02 Admin main layout page



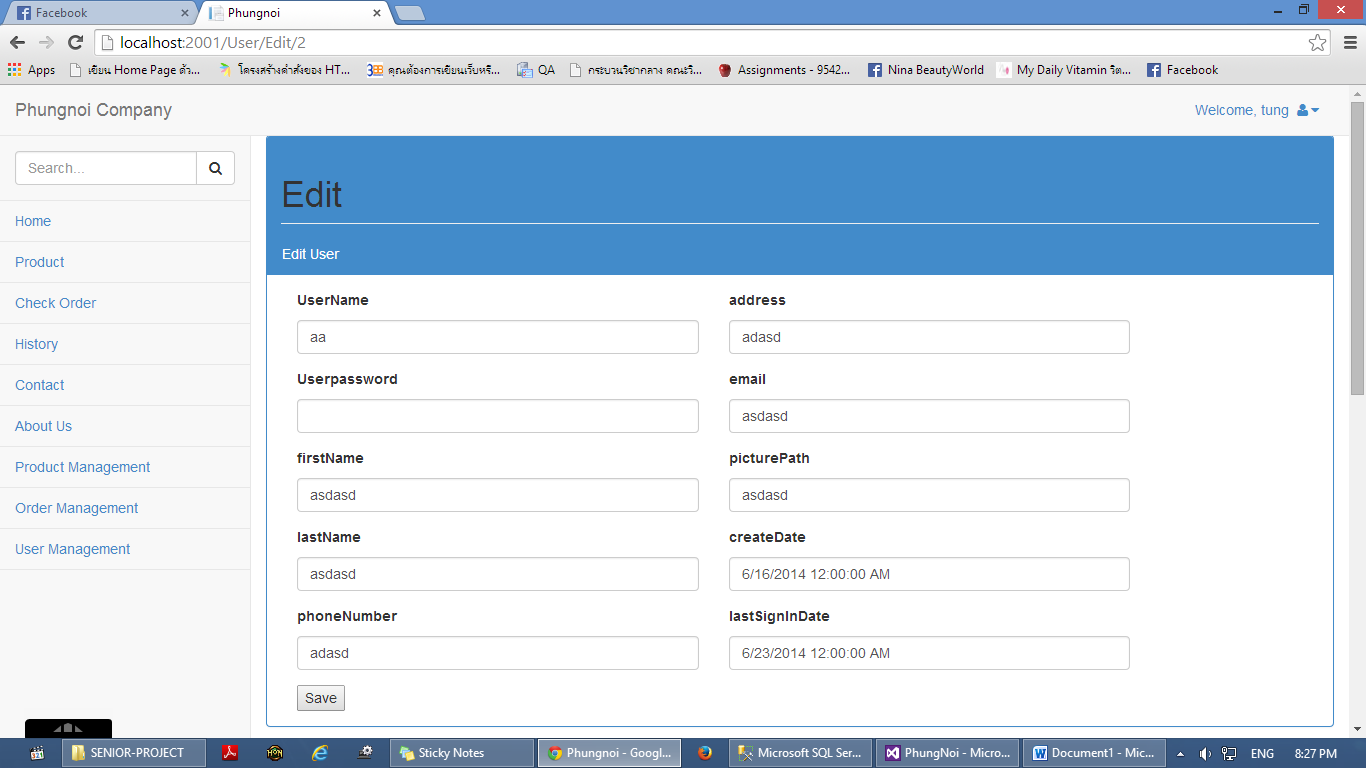
**Figure 16:** UI-03Wholesaler main layout page



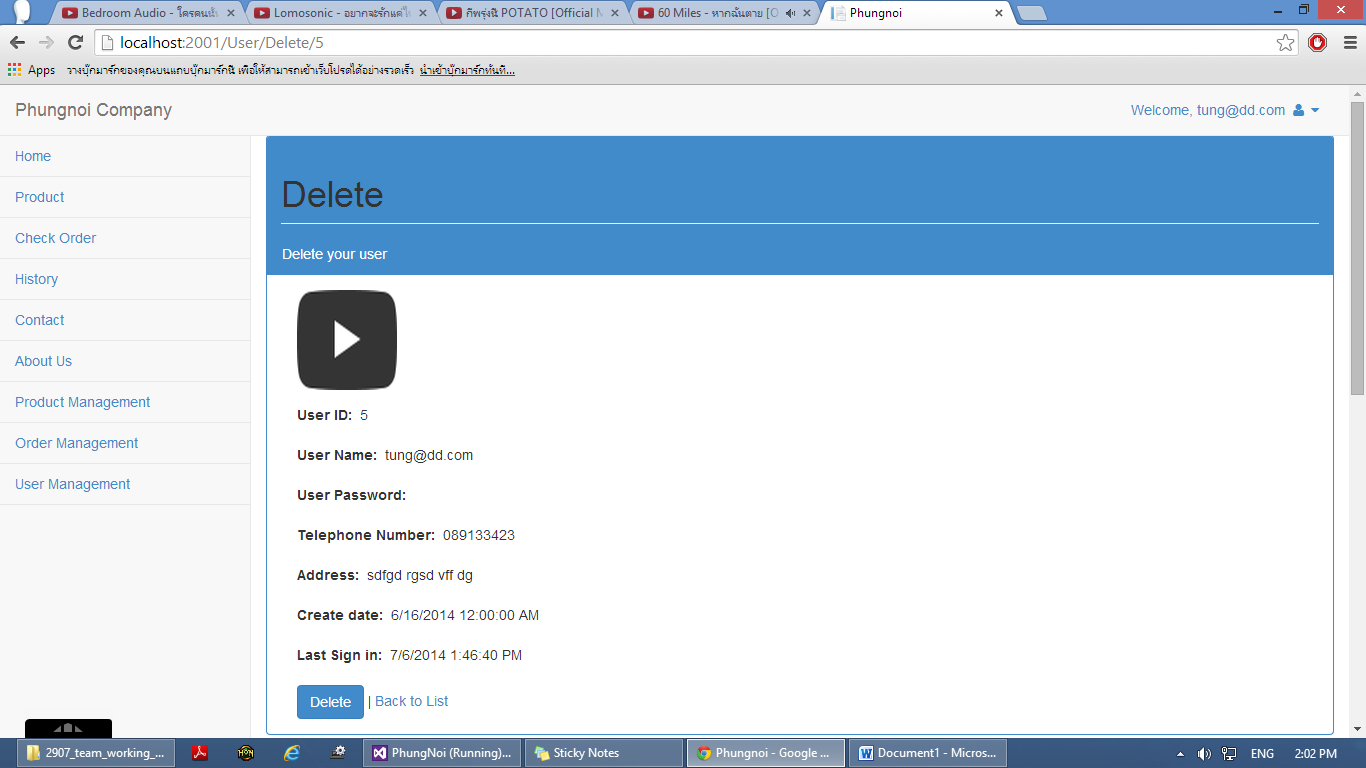
**Figure 17:** UI-04 Create user page

****

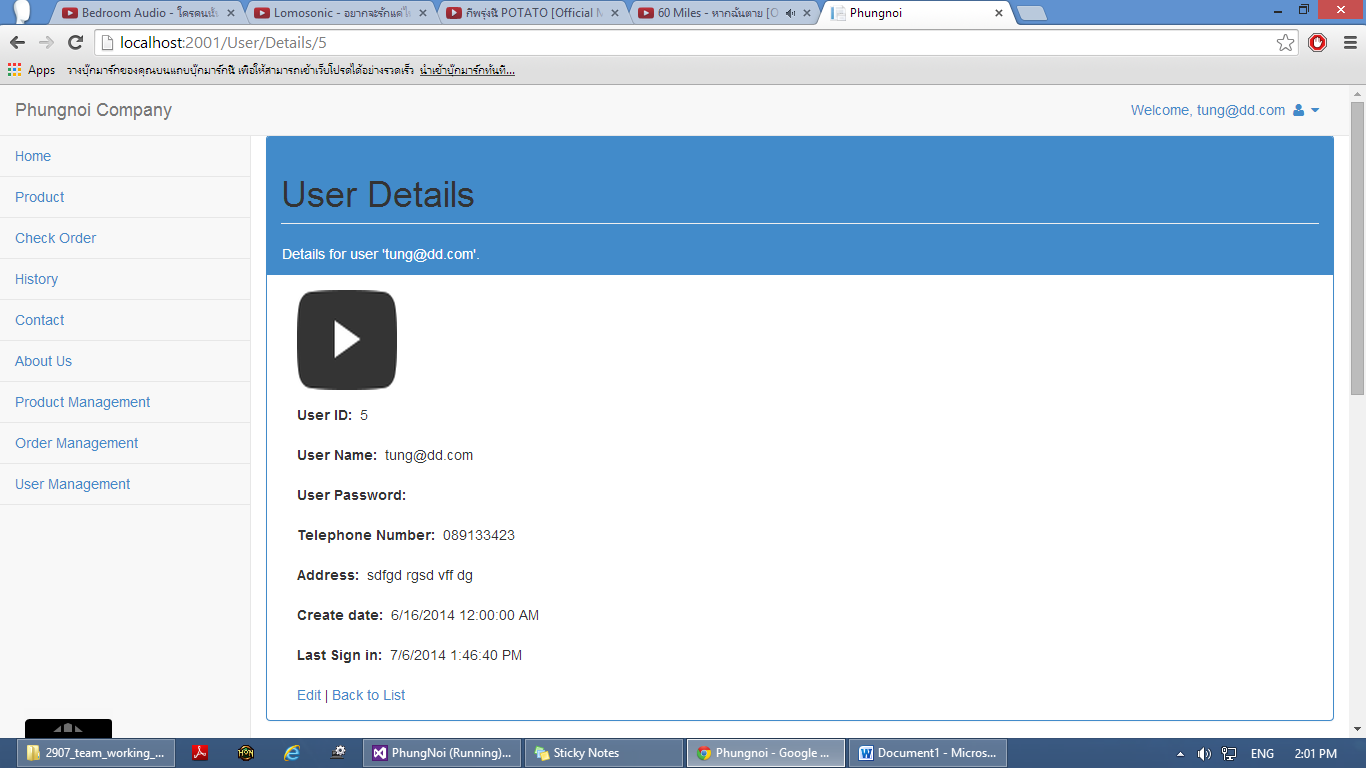
**Figure 18:** UI-05 User management page



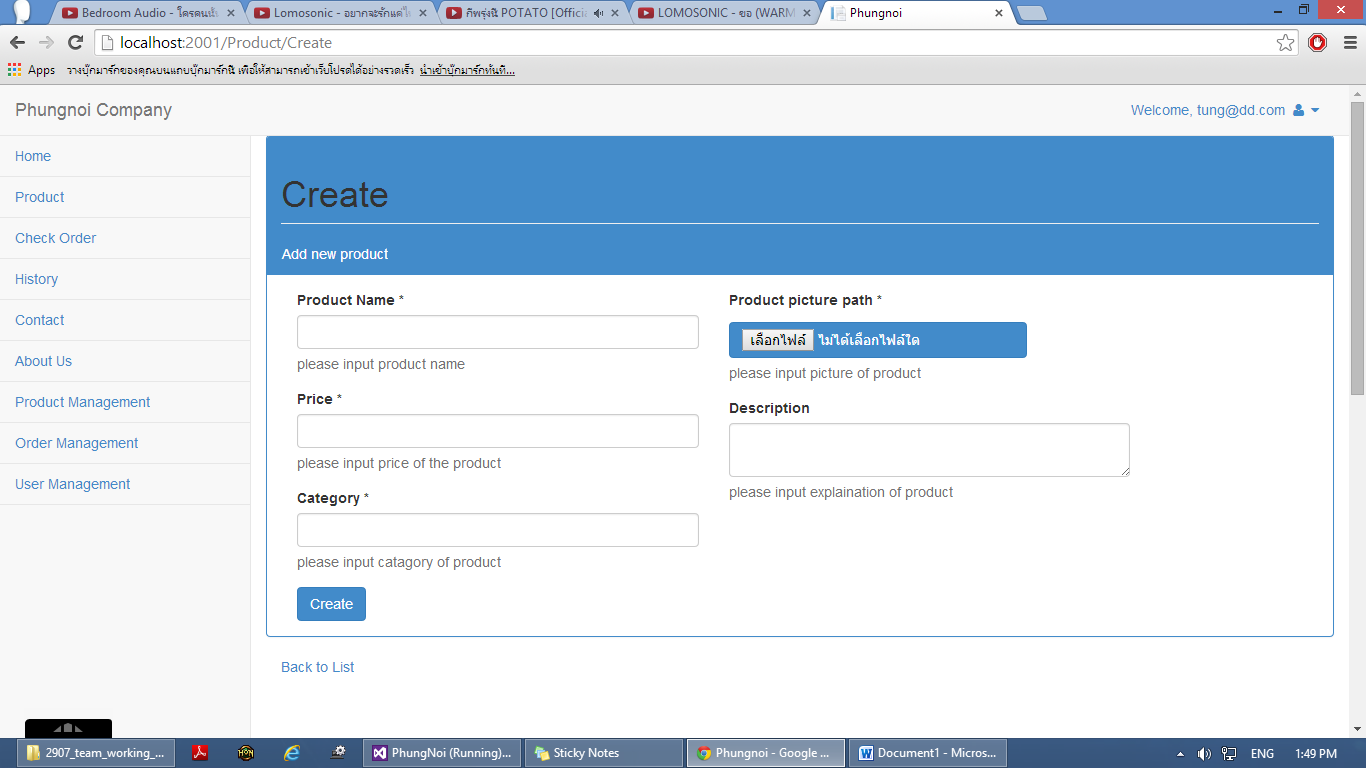
**Figure 19:** UI-06 Edit user information page

****

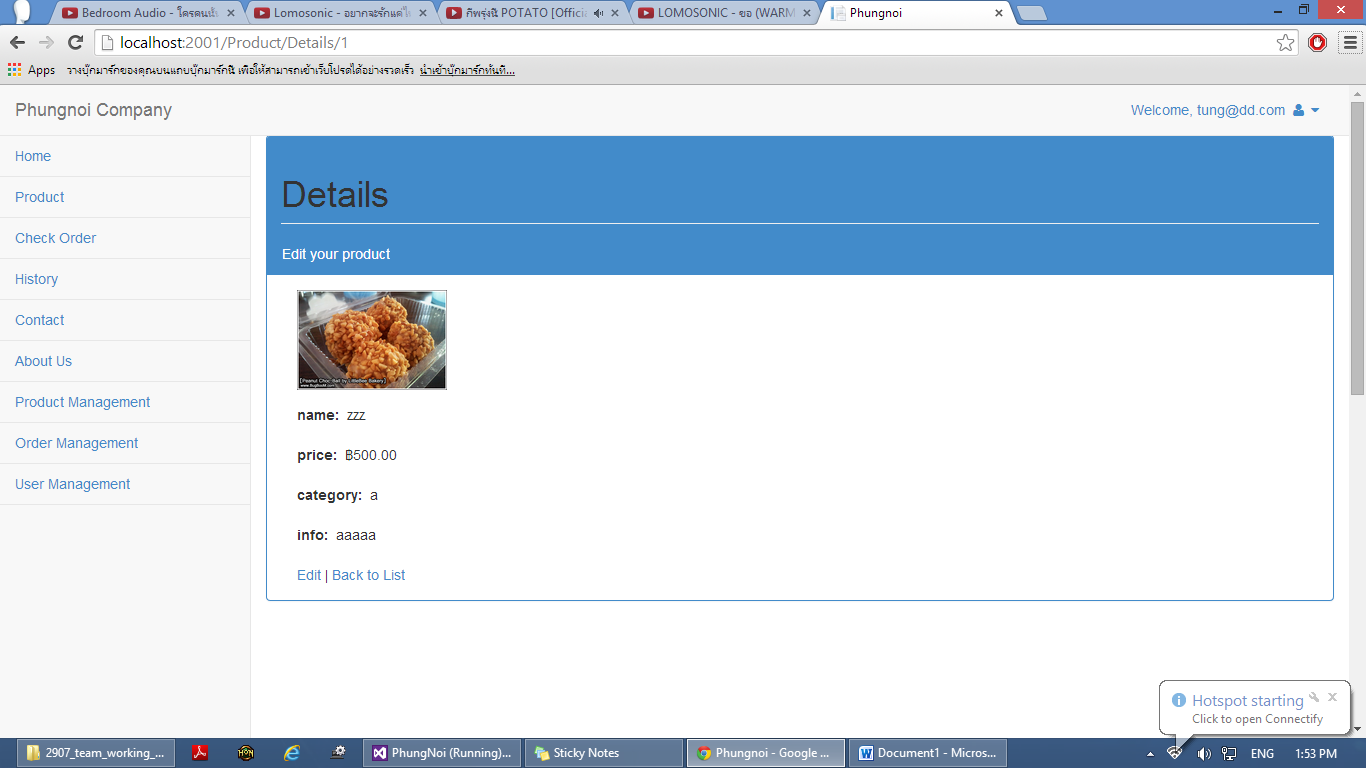
**Figure 20:** UI-07 Delete user page



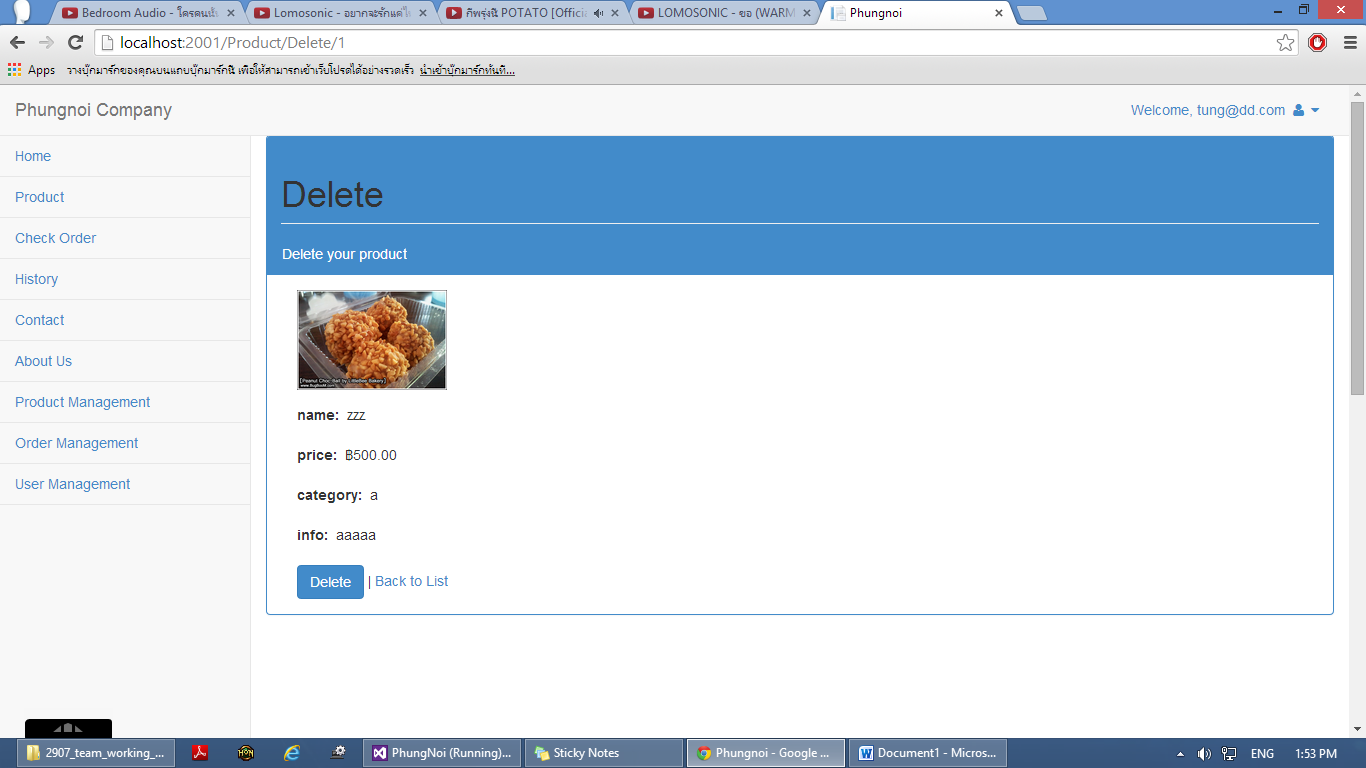
**Figure 21:** UI-08Edit own detail page



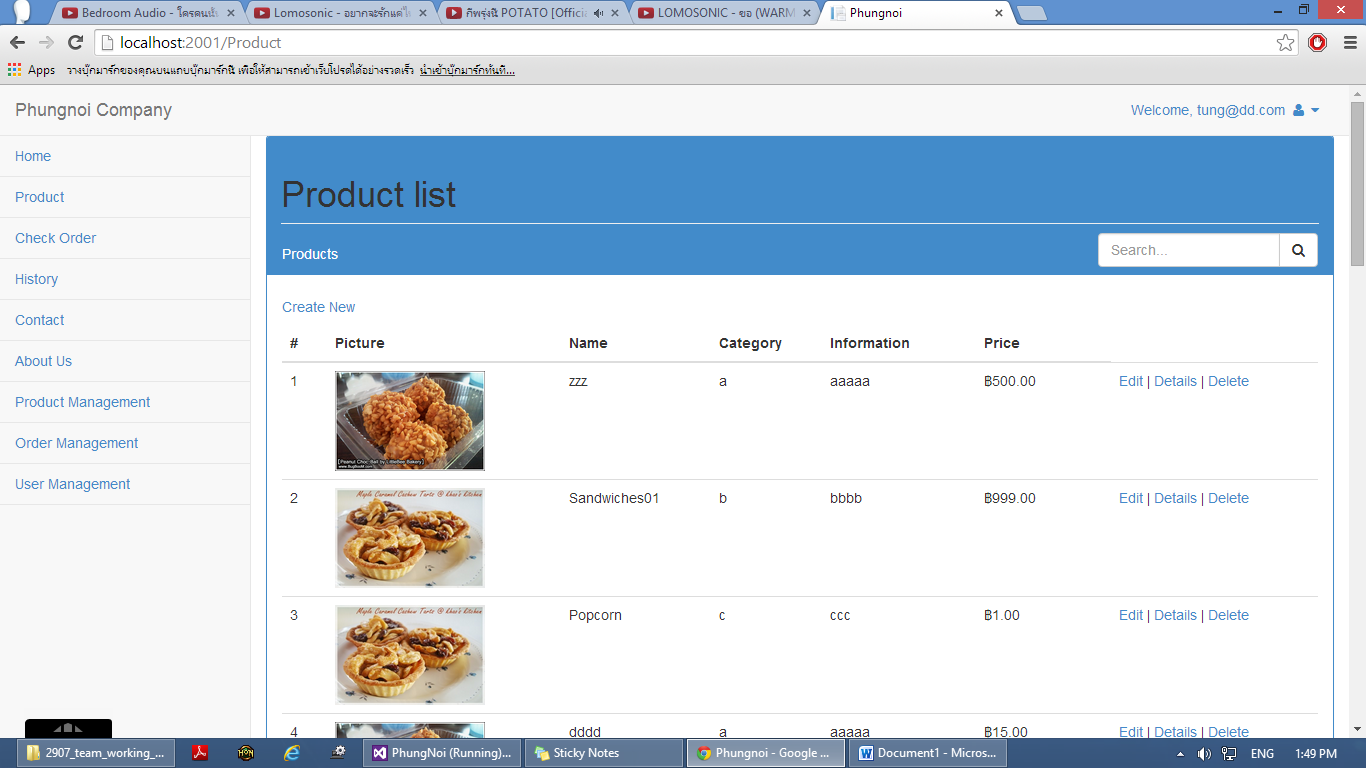
**Figure 22:** UI-09 Create bakery product page



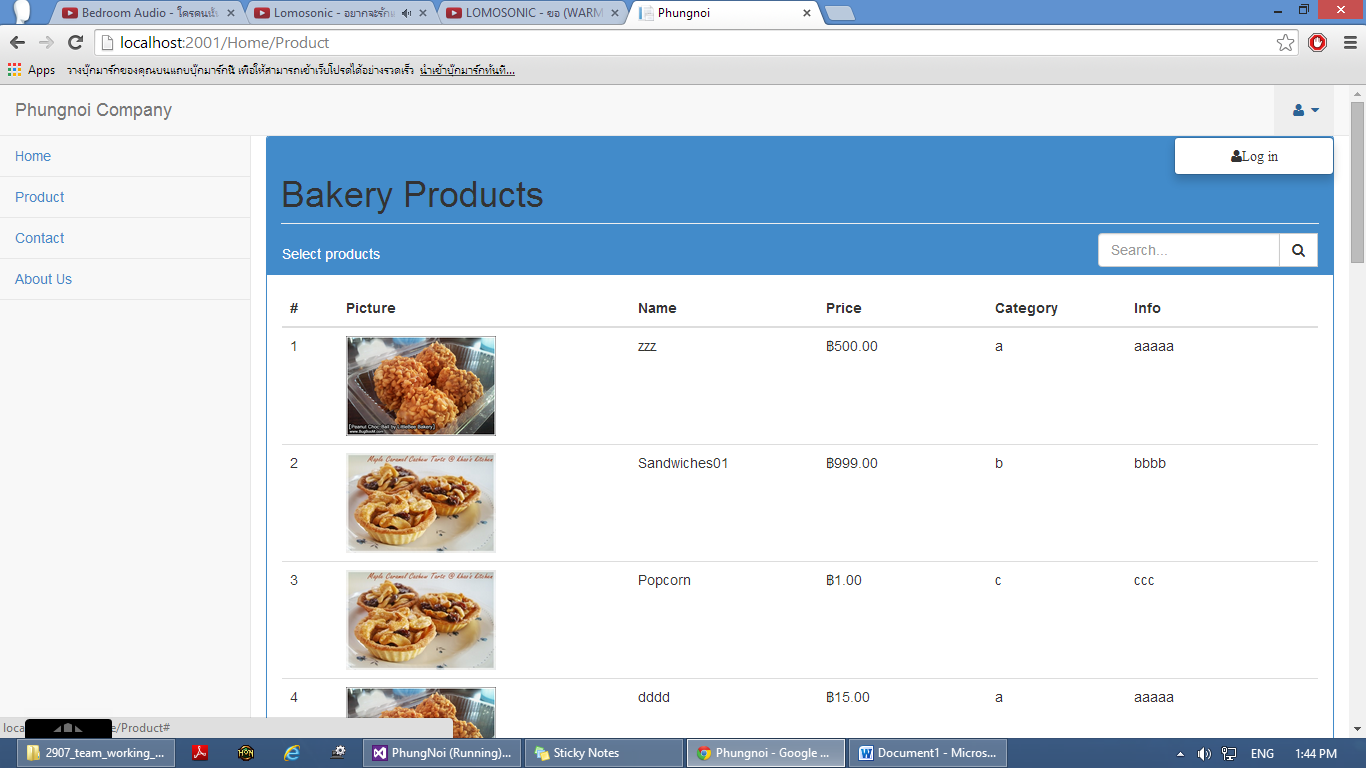
**Figure 23:** UI-10 Edit bakery product page



**Figure 24:** UI-11 Delete bakery product page

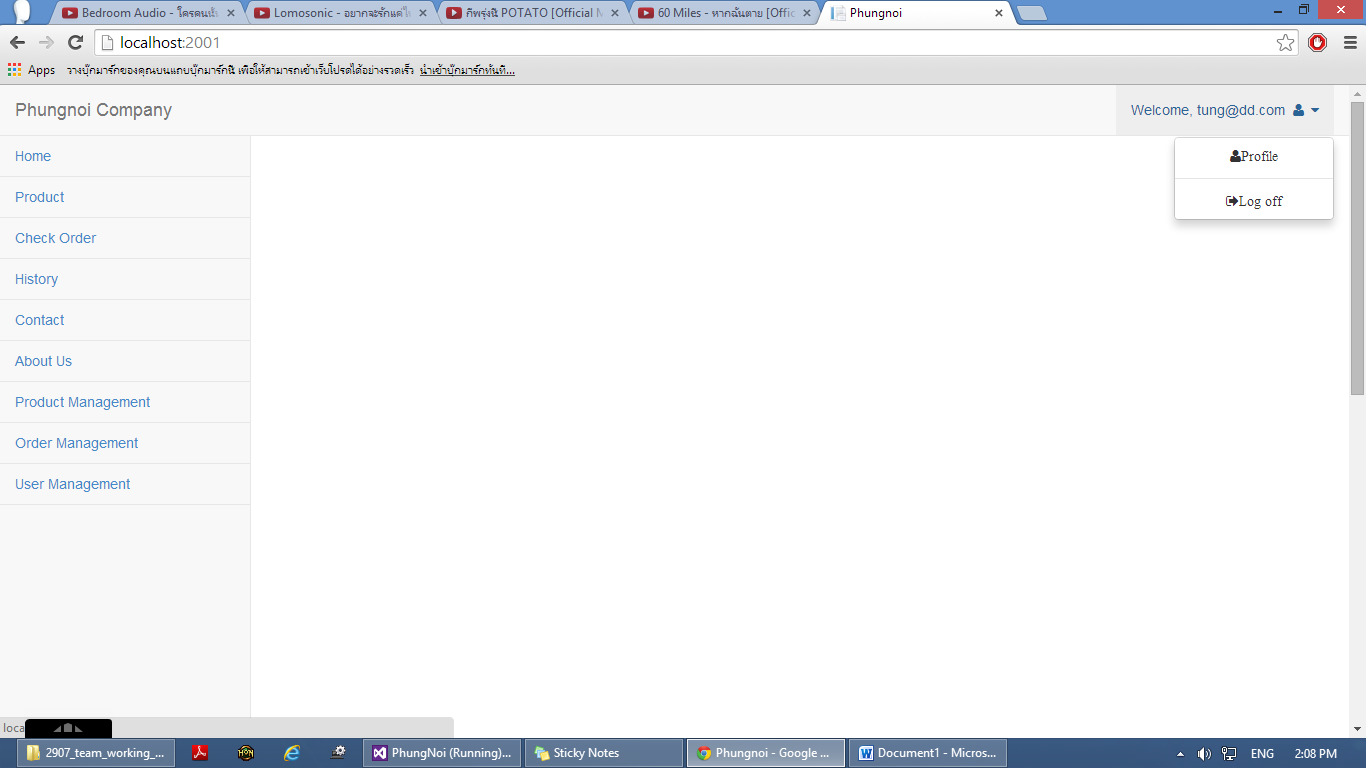


**Figure 25:** UI-12 Bakery product management page

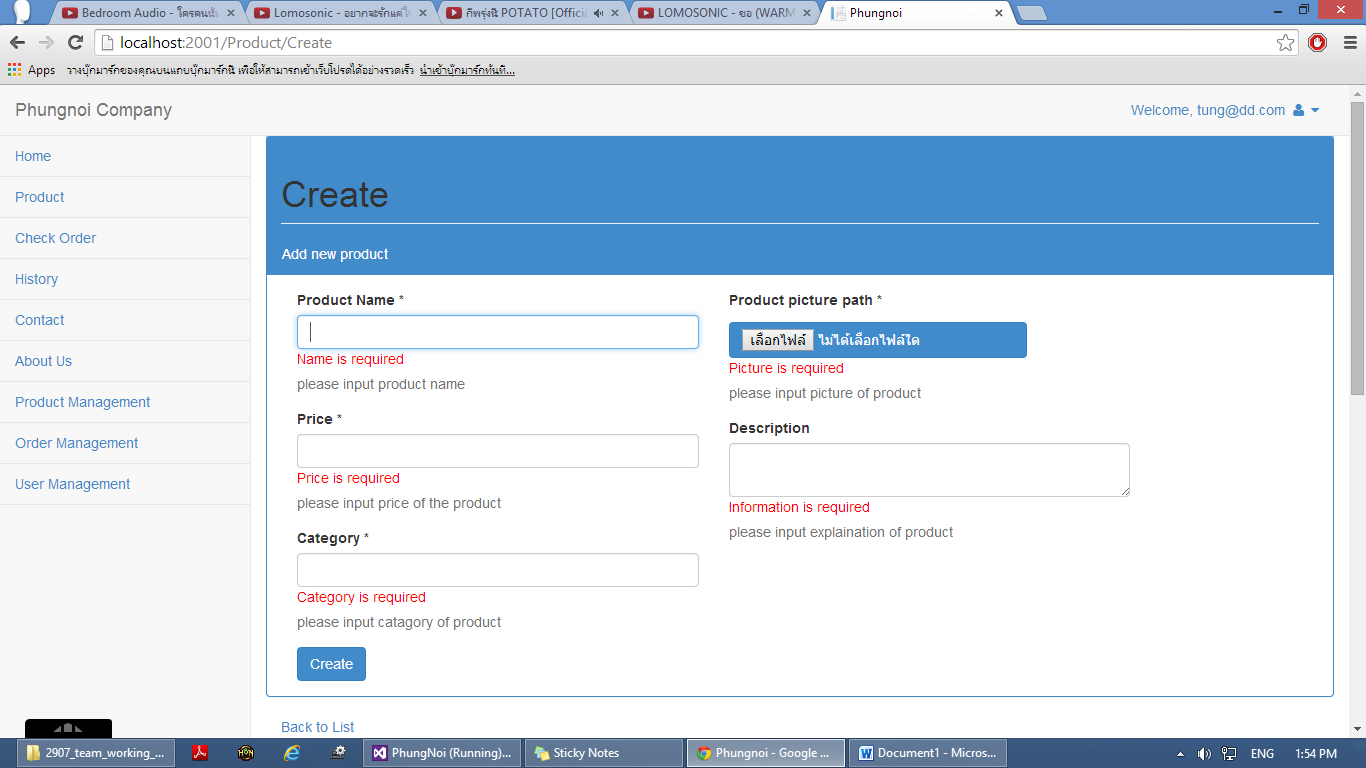


**Figure 25:** UI-16Wholesaler bakery product page

**Figure 26:** UI-13 Bakery product detail page



**Figure 27:** UI-14 Logoff page



**Figure 28:** UI-15 Error form page