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

Software Design Document

By

**Miss. NontraMahachanont 542115026**

**Mr. ParinyaPanyanak 542115034**

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

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

**Mr.PhudinanSingkhamfu**

|  |  |  |
| --- | --- | --- |
| **Software Requirement Specification** | | |
| Cross Ref. VSE-29110 | **Coverage Level:** | **Version:** |
|  | V.2.0 | V.2.0 |

|  |  |
| --- | --- |
| **Process Ownership** | **Approving Authority** |
| Web-based Ordering & Ingredient Estimating for Bakery Manufacturer | Aj.Phudinan Singkhamfu |
| **Scope** | **Approved Date** |
| Senior Project | 22 October 2014 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version Number** | **Record Date** | **Prepared/**  **Modified By** | **Reviewed By** | **Change Details** |
| **Documents History** | | | | |
| V.1.1 | 16-09-2014 | NM, PP | NM, PP | - Add Chapter 1  - Add Chapter 2  - Add Chapter 4 |
| V.1.2 | 17-09-2014 | NM, PP | NM, PP | - Add Chapter 3  - Fix inappropriate words |
| V.1.4 | 19-09-2014 | NM, PP | NM, PP | - Fix inappropriate words |
| V.1.6 | 20-10-2014 | NM, PP | NM, PP | - Adjust Chapter 2  - Fix inappropriate words |
| V.2.0 | 22-10-2014 | NM, PP | NM, PP | - Adjust Chapter 2  - Adjust Chapter 3  - Adjust Chapter 4  - Fix inappropriate words |

**\*NM = NontraMahachanont**

**\*PP = ParinyaPanyanak**

**\*PS = PhudinanSingkhamfu**

**Table of contents**

[**Chapter One | System Architecture** 4](#_Toc403561757)

[**Chapter Two |Detailed Design** 7](#_Toc403561758)

[2.1 Class Diagram 7](#_Toc403561759)

[2.2 Class Diagram Description 8](#_Toc403561760)

[2.2.1 Controller Class 8](#_Toc403561761)

[2.2.2 Model Class 12](#_Toc403561762)

[2.3 Sequence Diagram 15](#_Toc403561763)

[2.3.1 Wholesaler order bakery product 15](#_Toc403561764)

[2.3.2 Wholesaler send bakery product to own cart 16](#_Toc403561765)

[2.3.3 Wholesaler view their order information 17](#_Toc403561766)

[2.3.4 Admin view all order information 18](#_Toc403561767)

[2.3.5 Admin create ingredient information 19](#_Toc403561768)

[2.3.6 Admin edit ingredient information 20](#_Toc403561769)

[2.3.7 Admin delete ingredient 21](#_Toc403561770)

[2.3.8 Admin view ingredient list 22](#_Toc403561771)

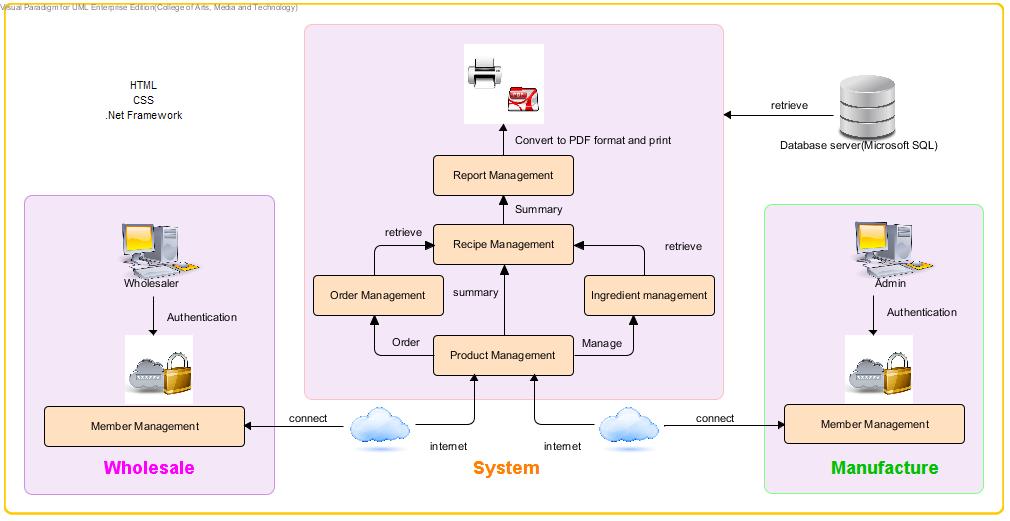
[2.3.9 Admin view ingredient information 23](#_Toc403561772)

[**Chapter Three | Data Architecture** 24](#_Toc403561773)

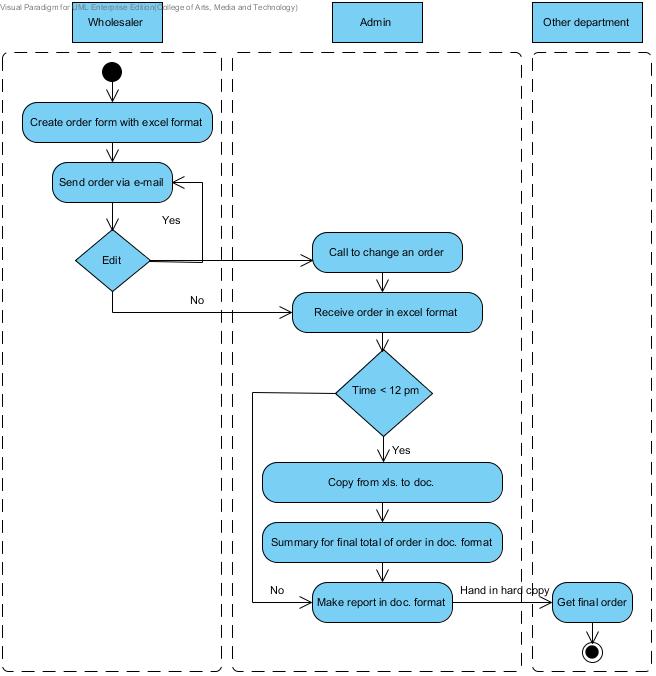
[3.1 ER Diagram 24](#_Toc403561774)

[**Chapter Four | User Interface** 25](#_Toc403561775)

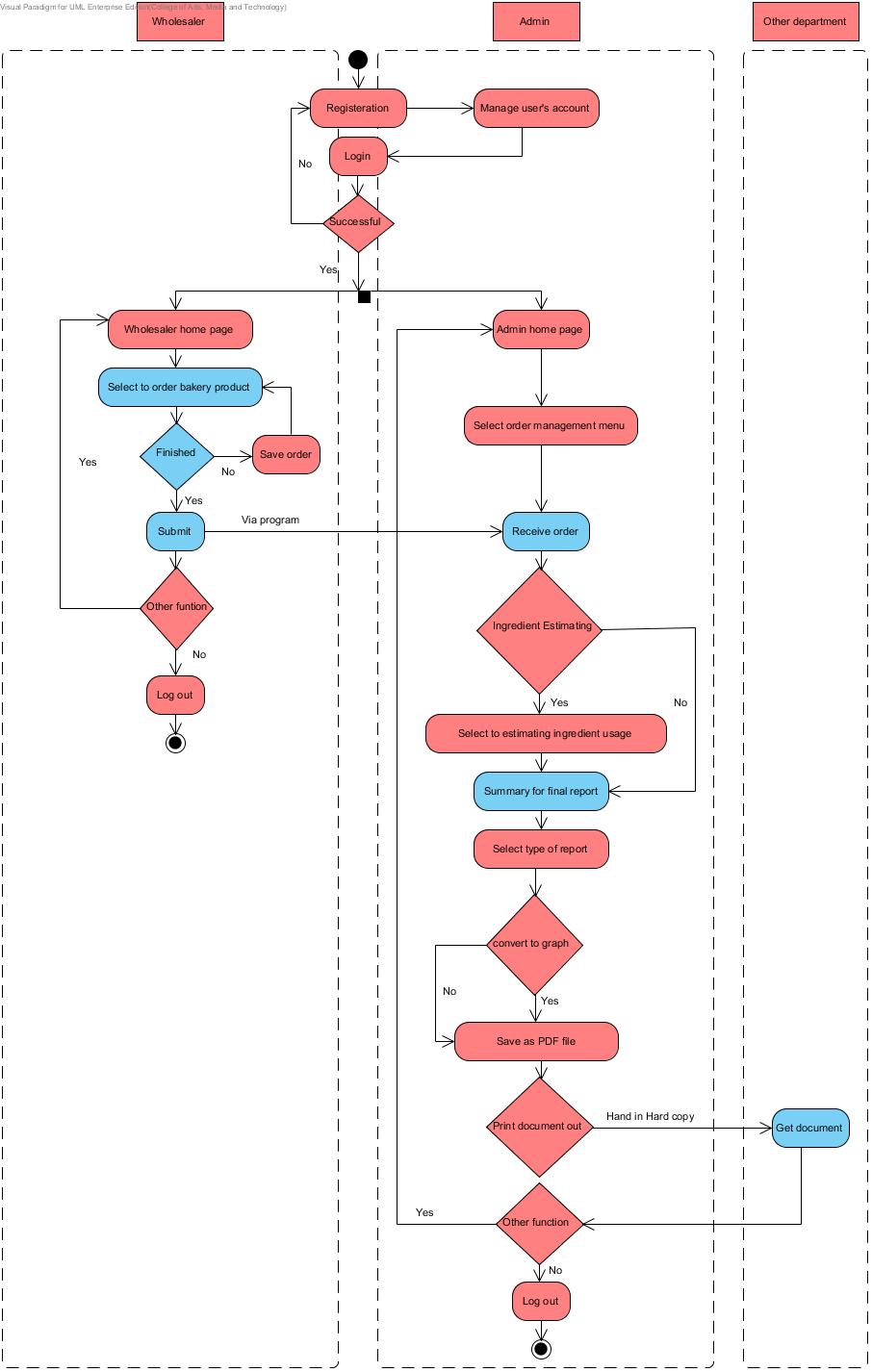
# **Chapter One |** **System Architecture**



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



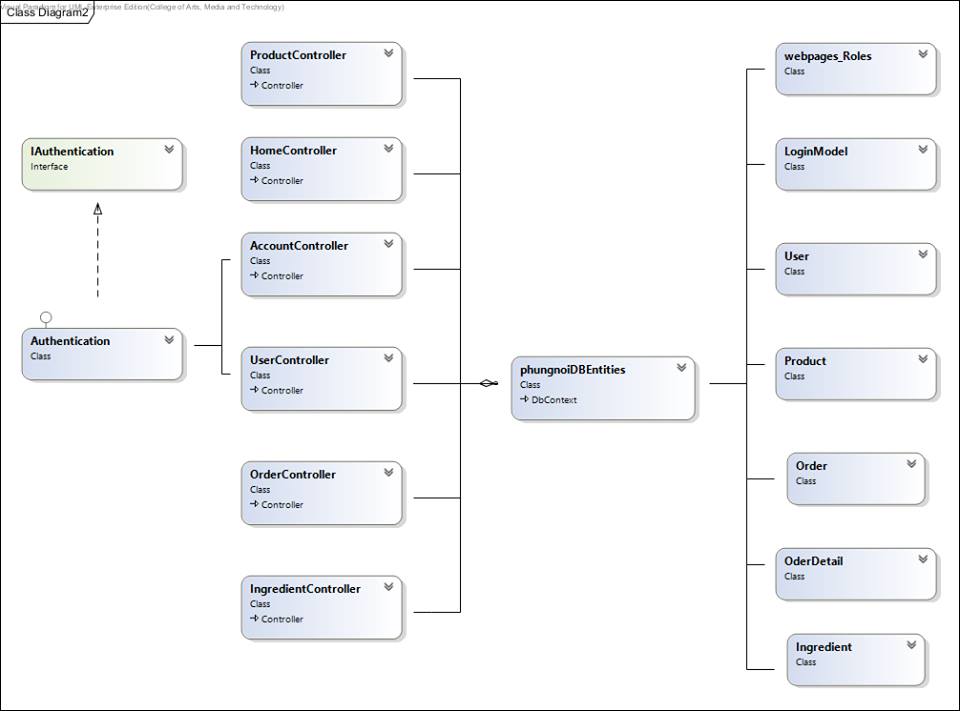
**Figure 2: Flow chart of original workflow**



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

# **Chapter Two |Detailed Design**

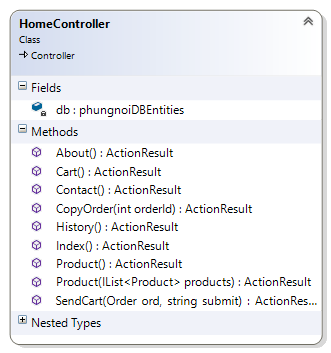
## Class Diagram



**Figure 4: Class Diagram**

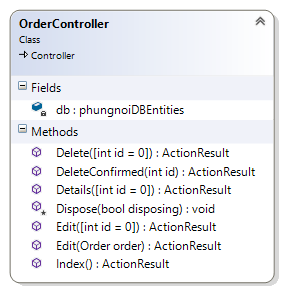
## Class Diagram Description

### 2.2.1 Controller Class



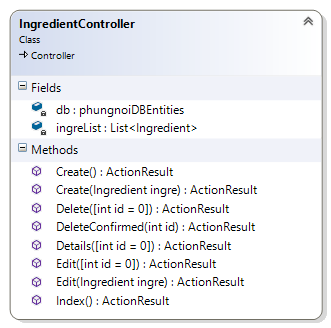
**Method:**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | CopyOrder() | This method uses to load old order from database**.** | **Return Type**: ActionResult  **Parameters:**ordrID; int |
| 5 | History() | This method uses to call the History page represent. | **Return Type**: ActionResult |
| 6 | Product() | This method uses to call the Product page represent. | **Return Type**: ActionResult |
| 7 | Product() | This method uses to add set of products to cart. | **Return Type**: ActionResult  **Parameters:**products;IList<produc> |
| 8 | SendCart() | This method uses to save or send cart to Database. | **Return Type**: ActionResult  **Parameters:**ord; int , submit;string |



**Method:**

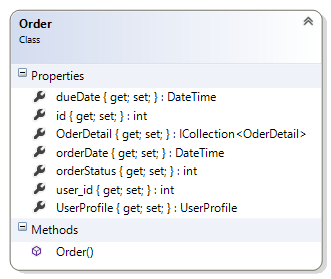
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Delete() | This method uses to call the Delete page represent. | **Return Type**: ActionResult  **Parameters:**id = 0; int |
| 2 | DeleteConfirmed() | This method uses to delete the order from database**.**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



**Method:**

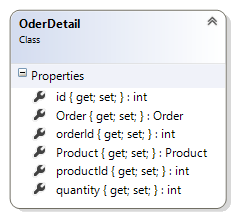
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Create() | This method uses to call the Create page. | **Return Type**: ActionResult |
| 2 | Create() | This method uses to add the product into the database. | **Return Type**: ActionResult  **Parameters:**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

### 2.2.2 Model Class



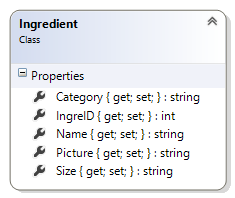
**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | dueDate | This properties uses for storing the due date into the database. | **Type**: DateTime |
| 2 | Id | This properties uses for storing the id into the database. | **Type**: Int |
| 3 | OrderDetail | This properties uses for storing the list of order detail into the database. | **Type**: ICollection<OrderDetail> |
| 4 | orderDate | This properties uses for storing the date to order into the database. | **Type**: DateTime |
| 5 | orderStatus | This properties uses for storing the status of order into the database. | **Type**: int |
| 6 | User\_id | This properties uses for storing the id of user into the database. | **Type**: int |
| 7 | UserProfile | This properties uses for storing the profile into the database. | **Type**: UserProfile |



**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Id | This properties uses for storing the order detail id into the database. | **Type**: int |
| 2 | Order | This properties uses for storing the order into the database. | **Type**: Order |
| 3 | orderId | This properties uses for storing the order Id into the database. | **Type**: int |
| 4 | Product | This properties uses for storing the Product into the database. | **Type**: Product |
| 5 | ProductId | This properties uses for storing the Product id into the database. | **Type**: int |
| 6 | quantity | This properties uses for storing the quantity of product into the database. | **Type**: int |



**Property:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Category | This property uses for store the ingredient category into the database. | **Type**: String |
| 2 | IngreId | This property uses for store the ingredient id into the database. | **Type**: Int |
| 3 | Name | This property uses for store the ingredient name into the database. | **Type**: String |
| 4 | picture | This property uses for store the ingredient picture into the database. | **Type**: String |
| 5 | Size | This property uses for store the ingredient size into the database. | **Type**: String |

## 2.3 Sequence Diagram

### 2.3.1 Wholesaler order bakery product

**Figure 5: SD - 11 Wholesaler order bakery product**

Wholesaler has to access the HomeController before creating new order information in the system. First, the wholesaler must inputs new bakery product quantity into the ingredient form. Then HomeController will be checking the bakery product quantity form. If the input information is more than zero, it will be return product to HomeController.

### 2.3.2 Wholesaler send bakery product to own cart



**Figure 6: SD - 12 Wholesaler send bakery product to own cart**

Wholesaler has to access the HomeController before sending new order information in the system. First, the wholesaler must selects submit button to add bakery product into own cart. Then HomeController will be adding the bakery product into the cart. If the input information is valid, it will be return cart page to the wholesaler.

### 2.3.3 Wholesaler view their order information



**Figure 7: SD - 13 Wholesaler view their order information**

Wholesaler has to access the HomeController before viewing own order information in the system. First, the wholesaler must login to the system and selects history bar to view current order. Then HomeController will be checking the username and retrieve their order information to display at HomeController.

### 2.3.4 Admin view all order information



**Figure 8: SD - 14 Admin view all order information**

Admin has to access the OrderController before viewing all order information in the system. First, the admin must login to the system and selects order management bar to access order management page. Then OrderController will be checking the order information and retrieve all order information to display at OrrderController.

### 2.3.5 Admin create ingredient information

****

**Figure 9: SD - 15 Admin create ingredient information**

Admin has to access the IngredientController before creating new ingredient information in the system. First, the admin must inputs new ingredient information include ingredient name, category and ingredient picture to the ingredient form. Then select save button, IngredientController will be checking the ingredient form. If the input information is valid, it will be save new ingredient into the database.

### 2.3.6 Admin edit ingredient information

****

**Figure 10: SD - 16 Admin edit ingredient information**

Admin has to access the IngredientController before editing ingredient information in the system. First, the admin must inputs updated ingredient information include ingredient name, ingredient description, category and ingredient picture to the ingredient form. Then select save button, IngredientController will be checking the ingredient form. Then the system will be updated information into the database. Then the system returns updated ingredient to IngredientController.

### 2.3.7 Admin delete ingredient

****

**Figure 11: SD - 17 Admin delete ingredient**

Admin has to access the IngredientController before deleting ingredient information in the system. First, Admin delete ingredient by select to delete and confirm to delete. Then the system calls ingredient information from the database. If the admin save this action, the system shall clear previous ingredient information from the database

### 2.3.8 Admin view ingredient list

****

**Figure 12: SD-18 Admin view ingredient list**

First, Admin has to access the index page. Then the admin search the ingredient information by input the keyword and submits to the controller. The controller will validate the existing keywords and display the related results. All ingredient information are retrieved from the phungnoi database, ingredient table. Then, it will return the required data to the admin in information page.

### 2.3.9 Admin view ingredient information

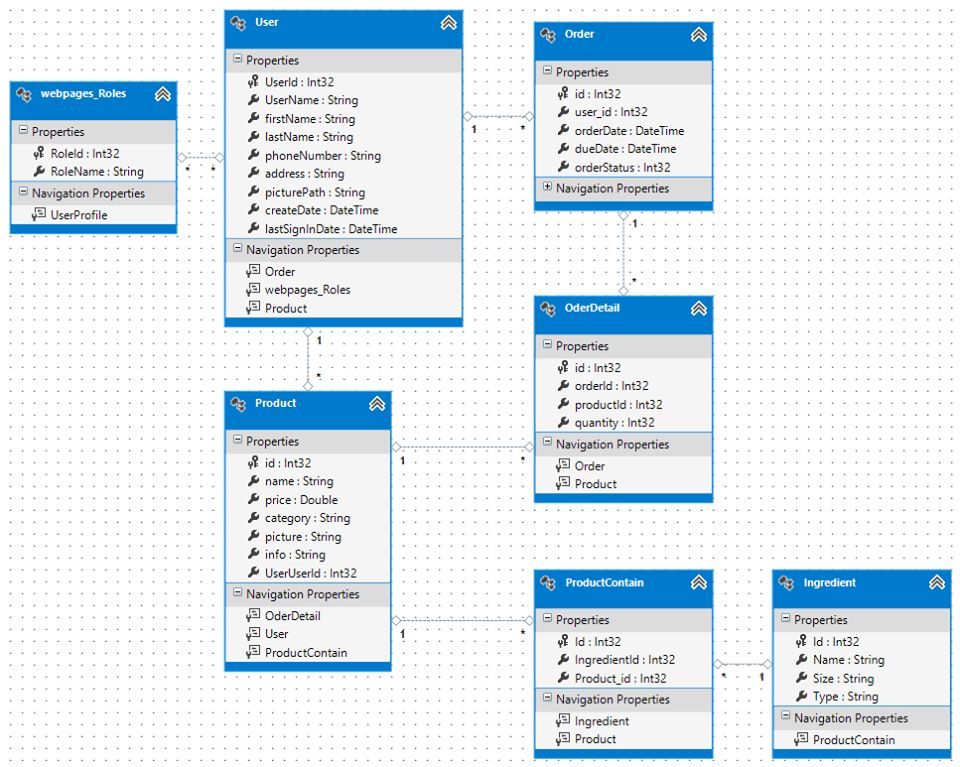
****

**Figure 13: SD-19 Admin view ingredient information**

First, Admin has to access the ingredient details page and request the require data to the IngredientController. Then, the controller will find id are matched and return those information to the admin. The admin will see the ingredient information from ingredient details page.

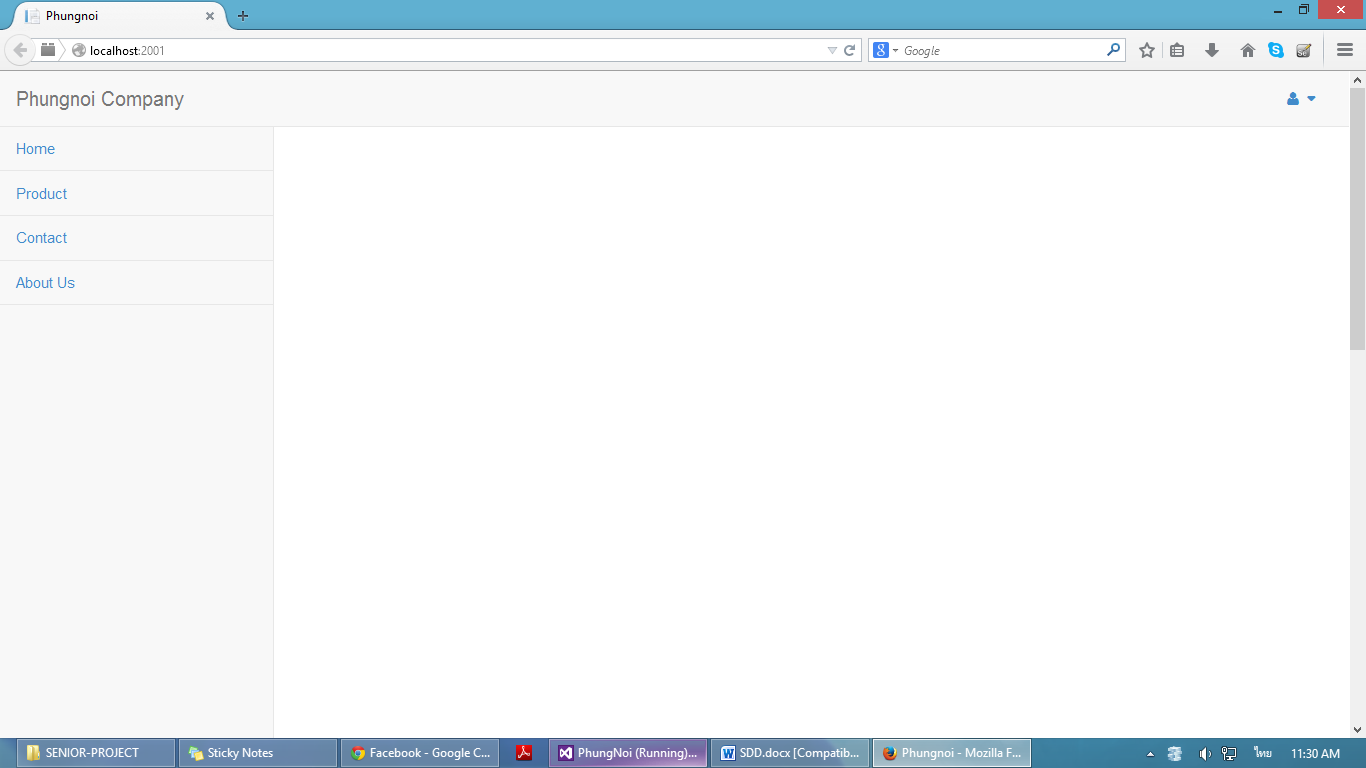
# **Chapter Three | Data Architecture**

## 3.1 ER Diagram

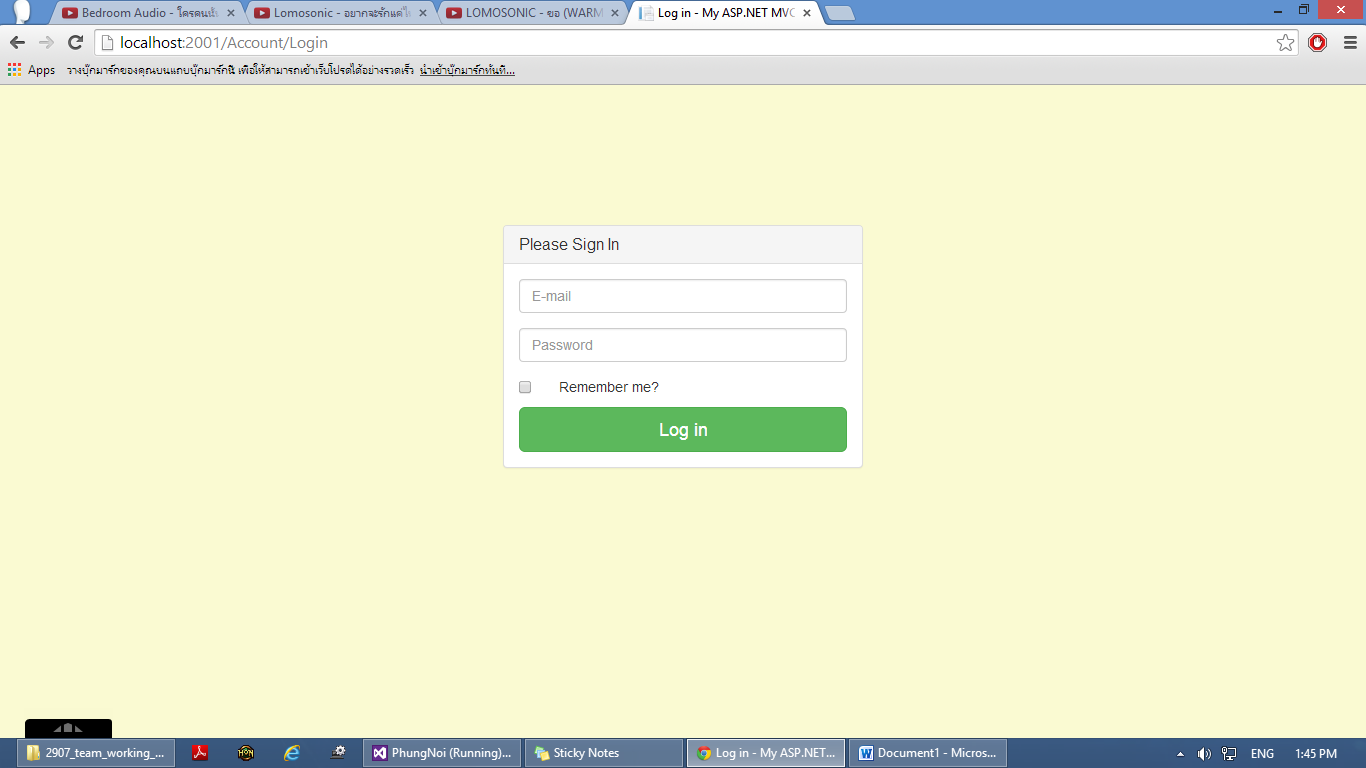


**Figure 14: ER Diagram**

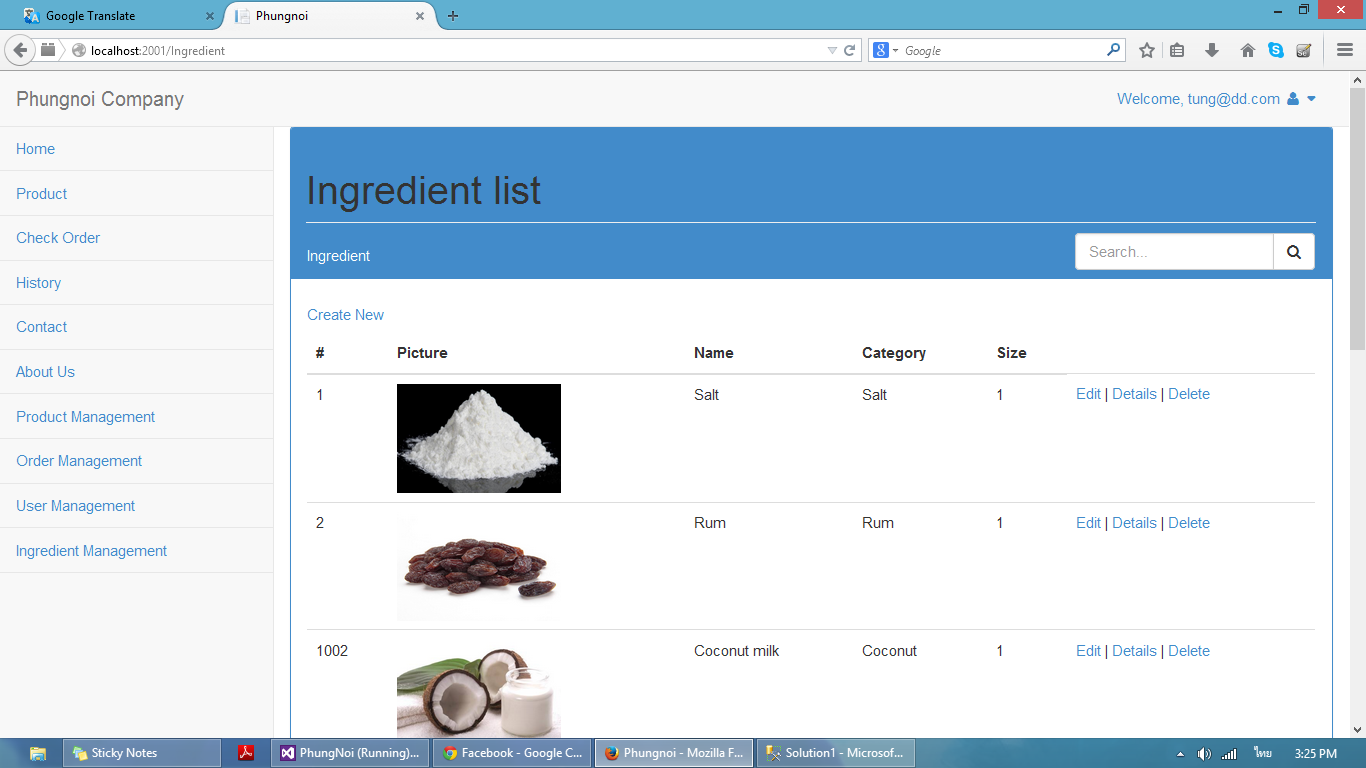
# **Chapter Four | User Interface**



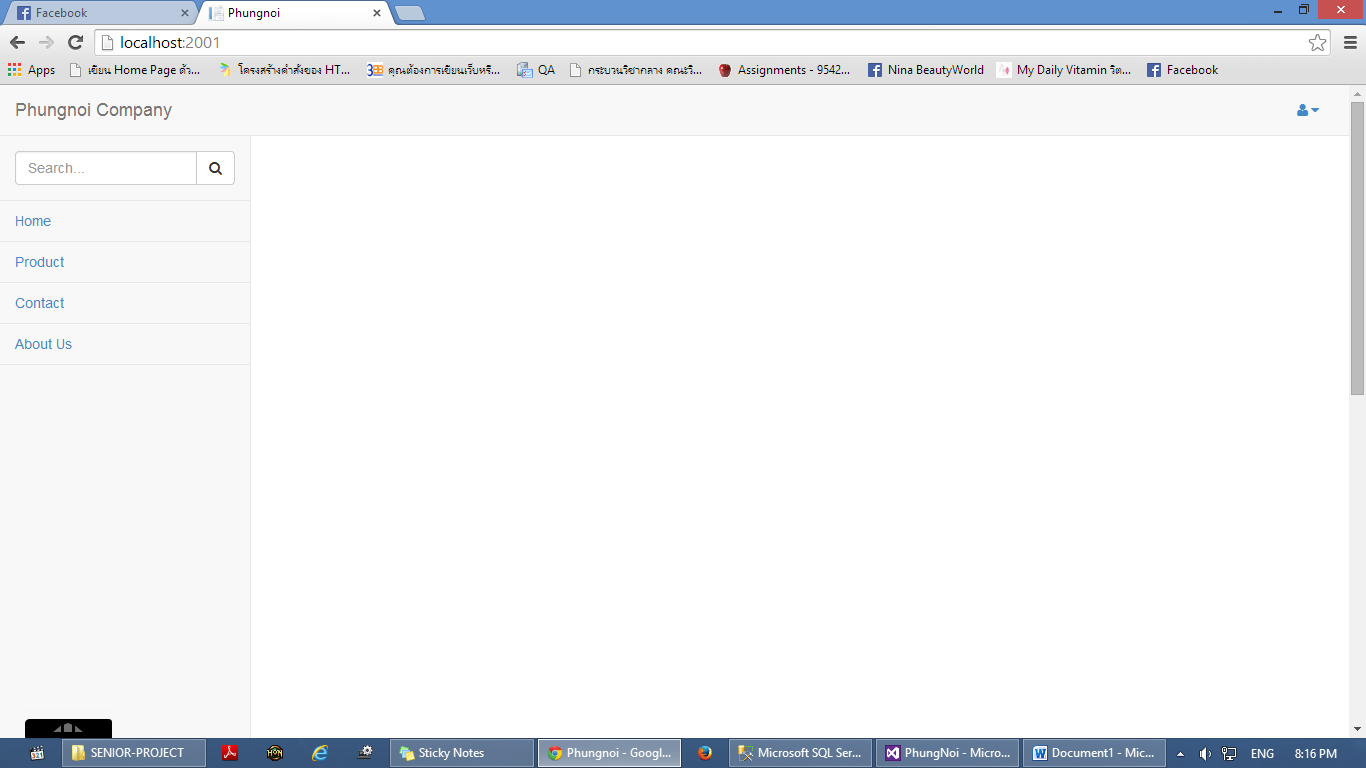
**Figure 15:** UI – 01 Home page



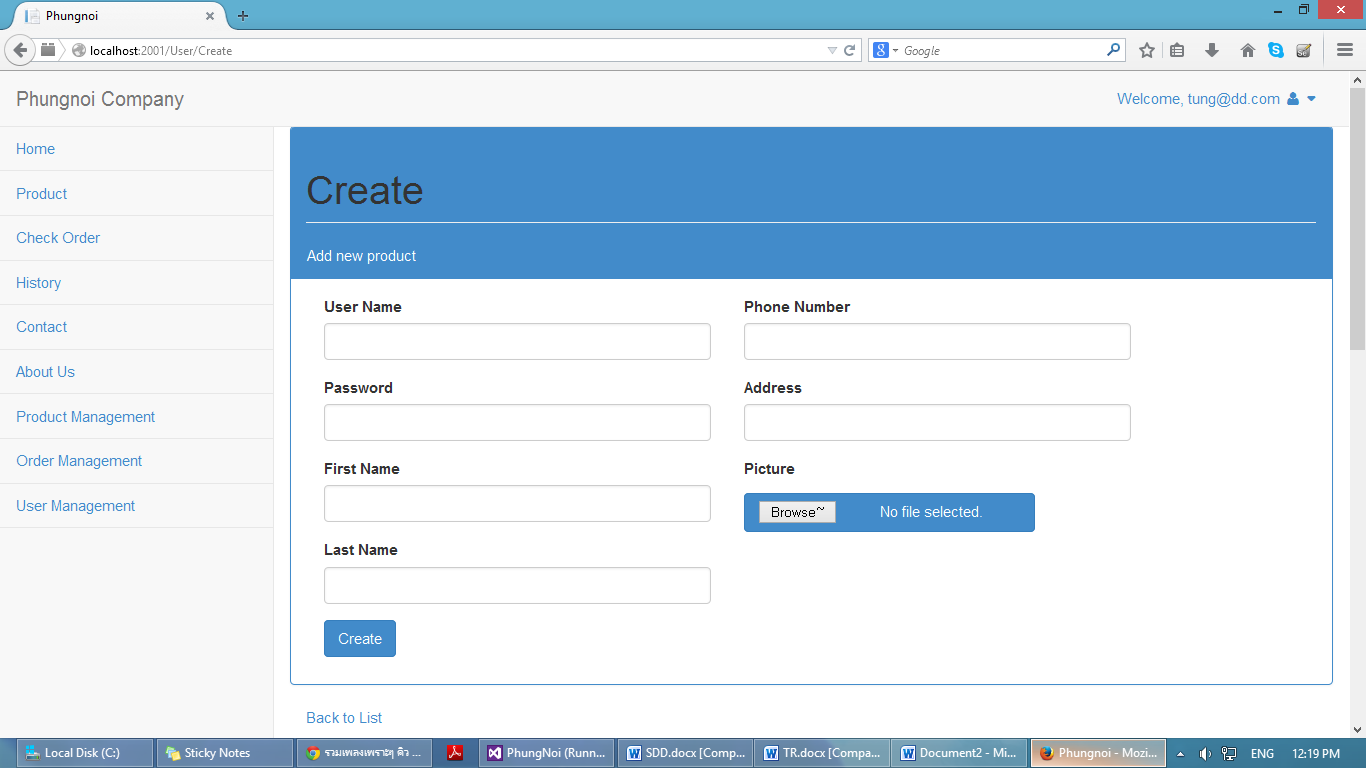
**Figure 16:** UI – 02 Login page



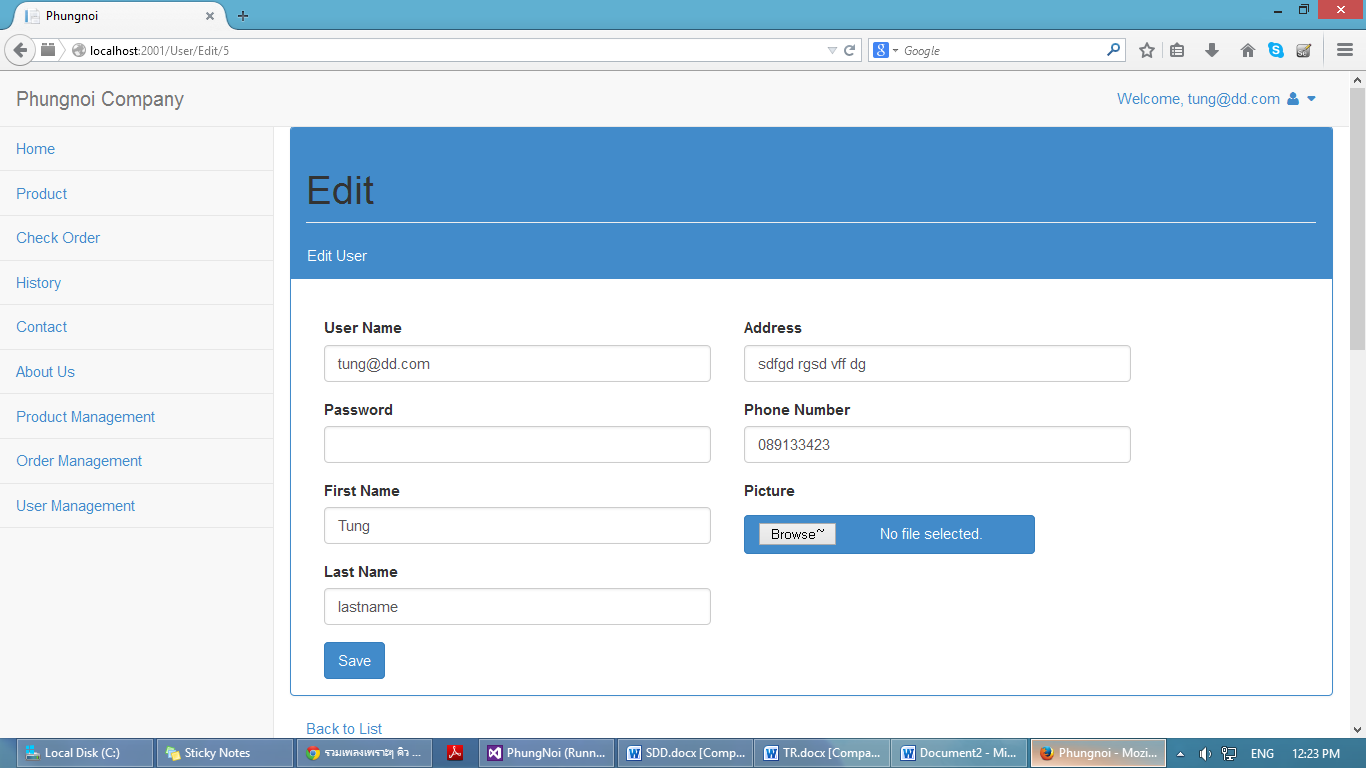
**Figure 17:** UI - 03 Admin main layout page



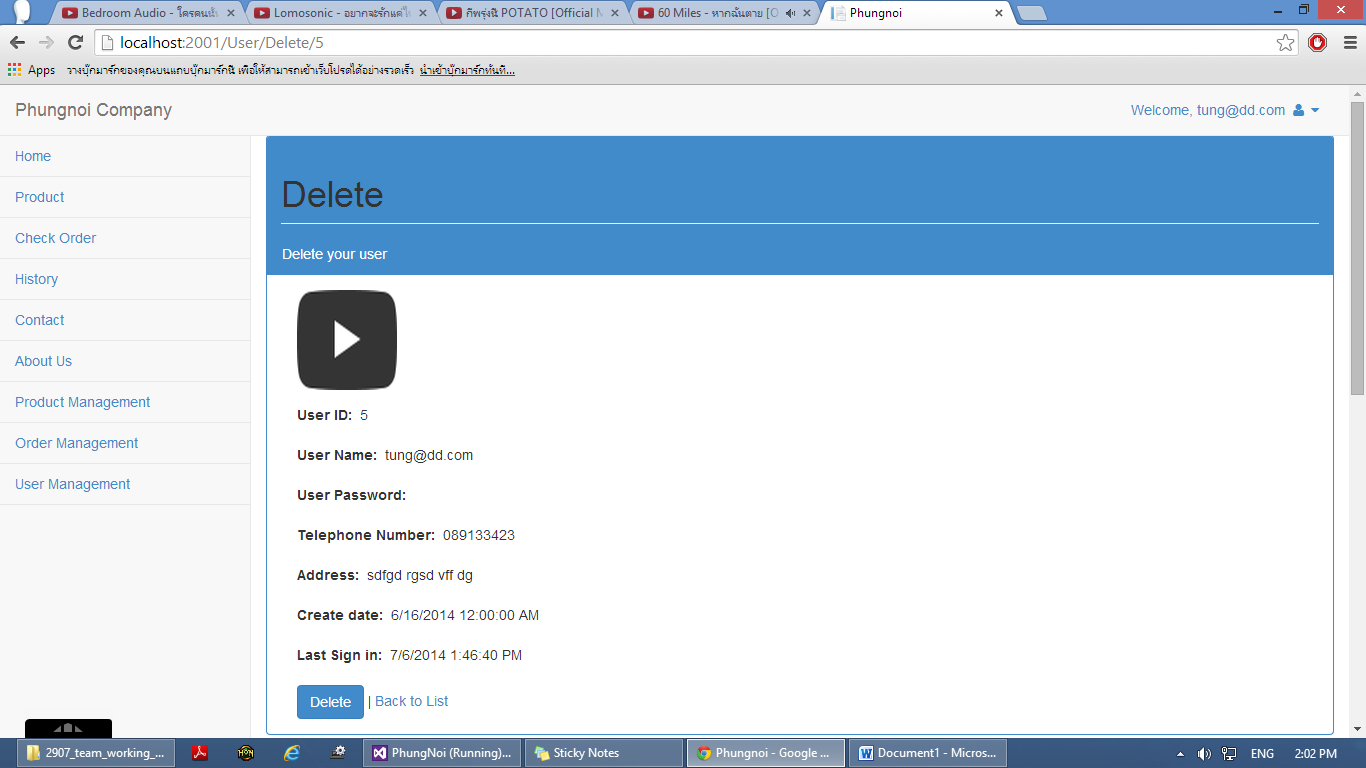
**Figure 18:** UI – 04 Wholesaler main layout page

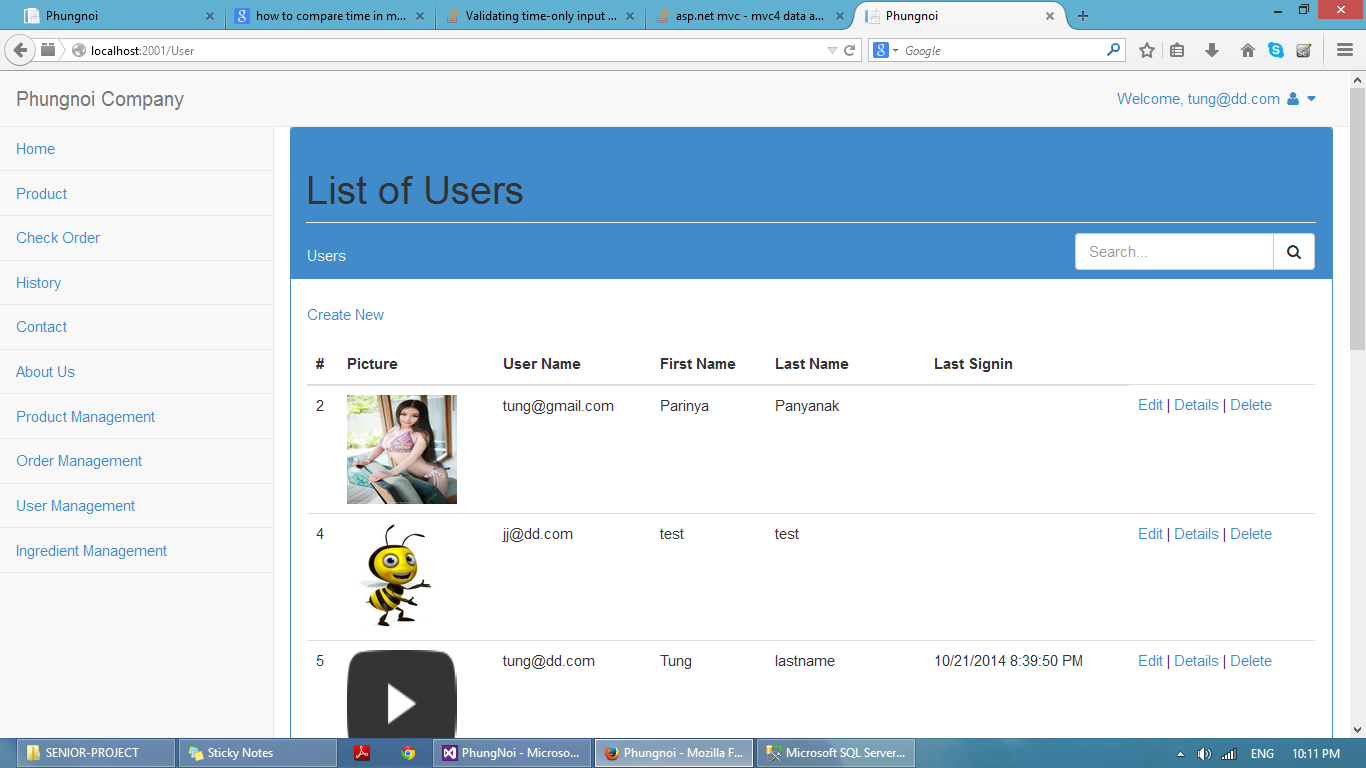


**Figure 19:** UI – 05 Create user information page

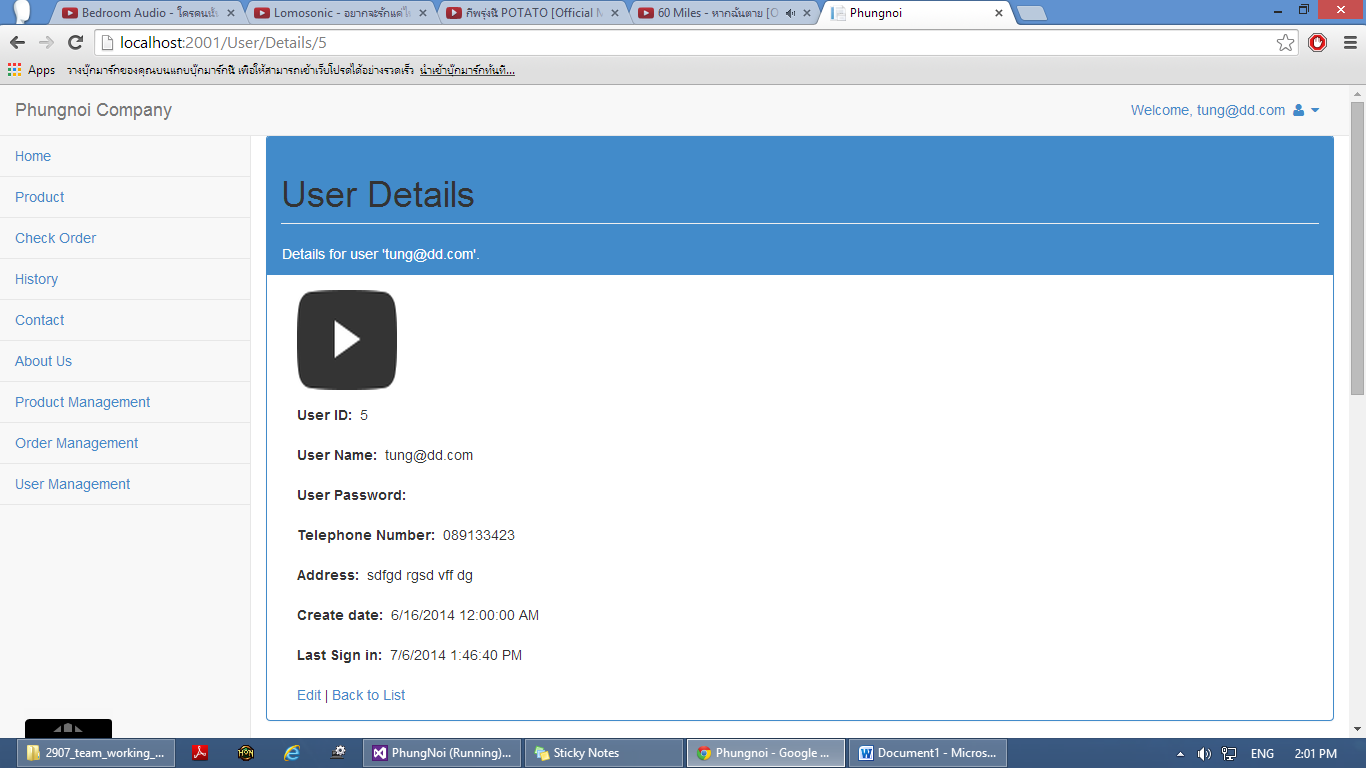


**Figure 20:** UI – 06 Edit user information page

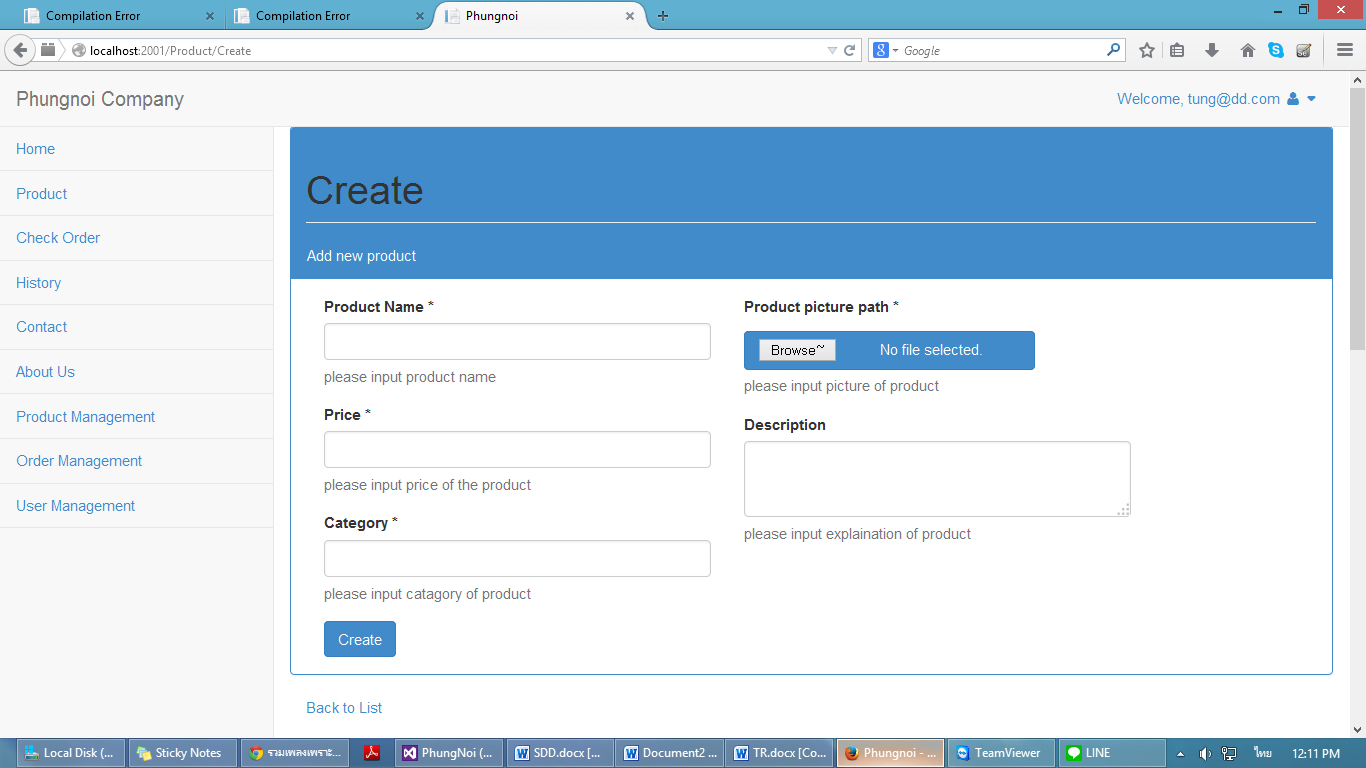


**Figure 21:** UI – 07 Delete user information page

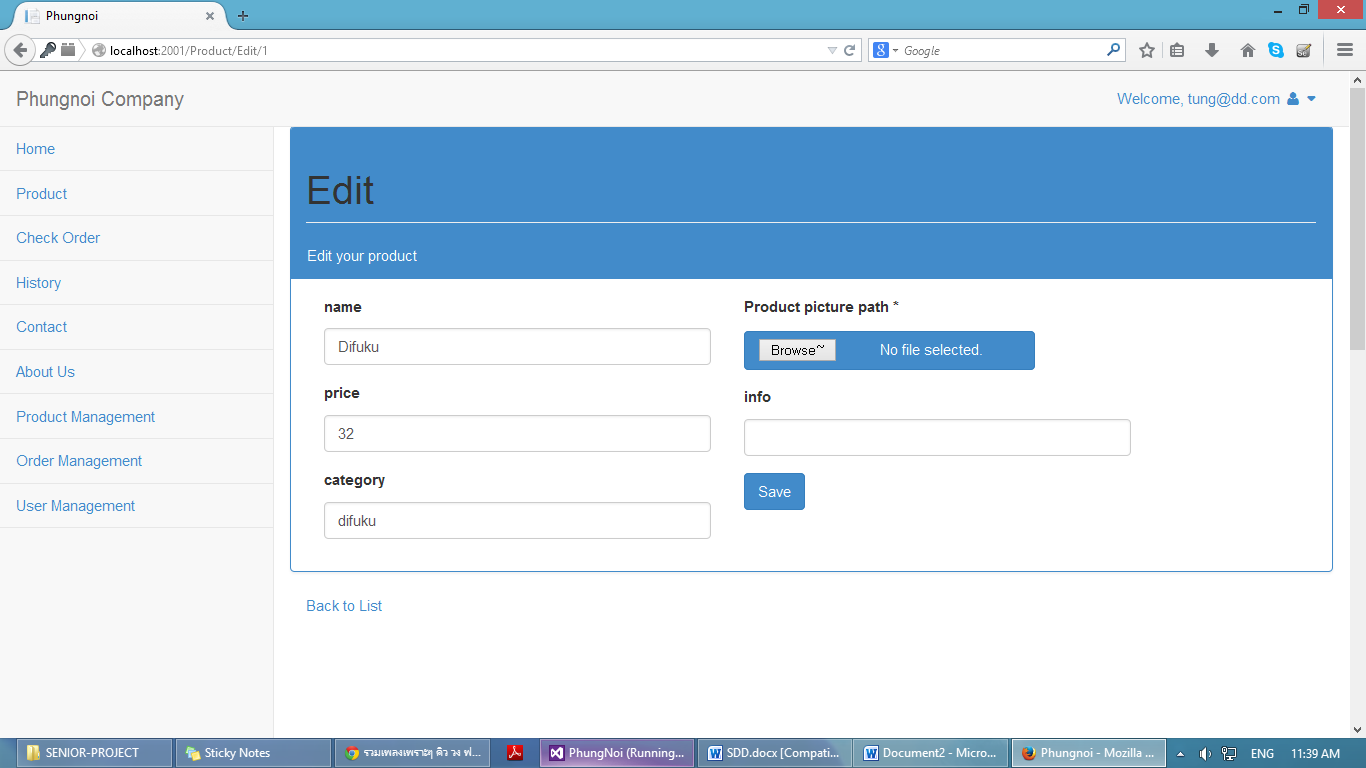
**Figure 22:** UI – 08 User list page

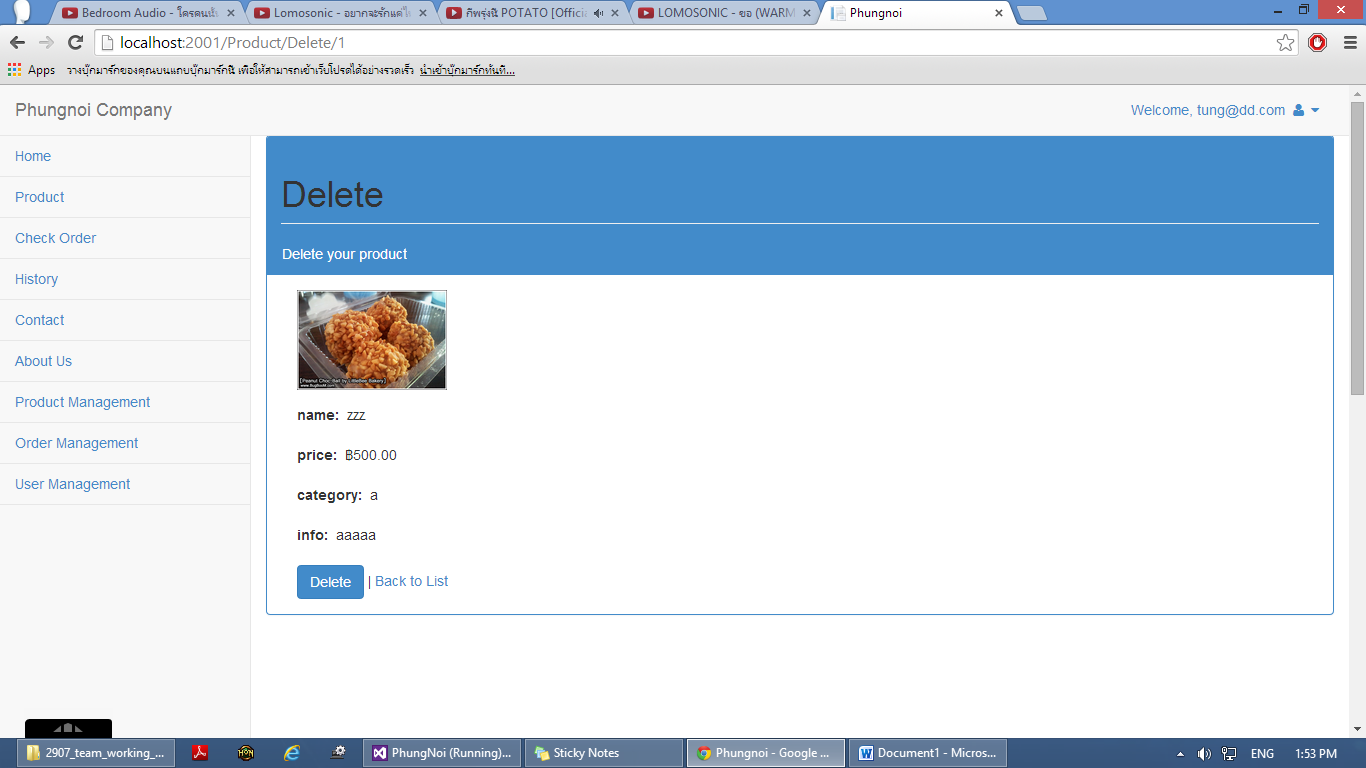


**Figure 23:** UI – 09 User details page

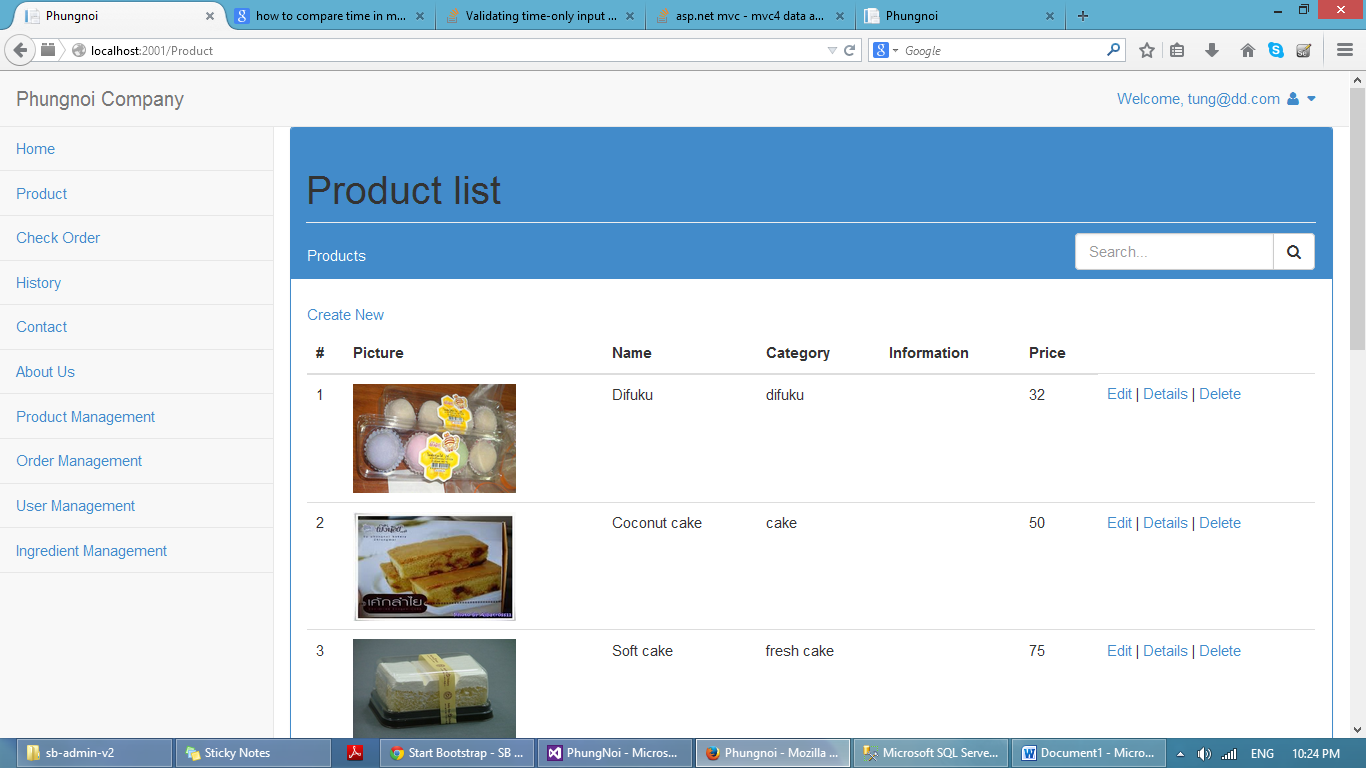


**Figure 24:** UI – 10 Create bakery product page

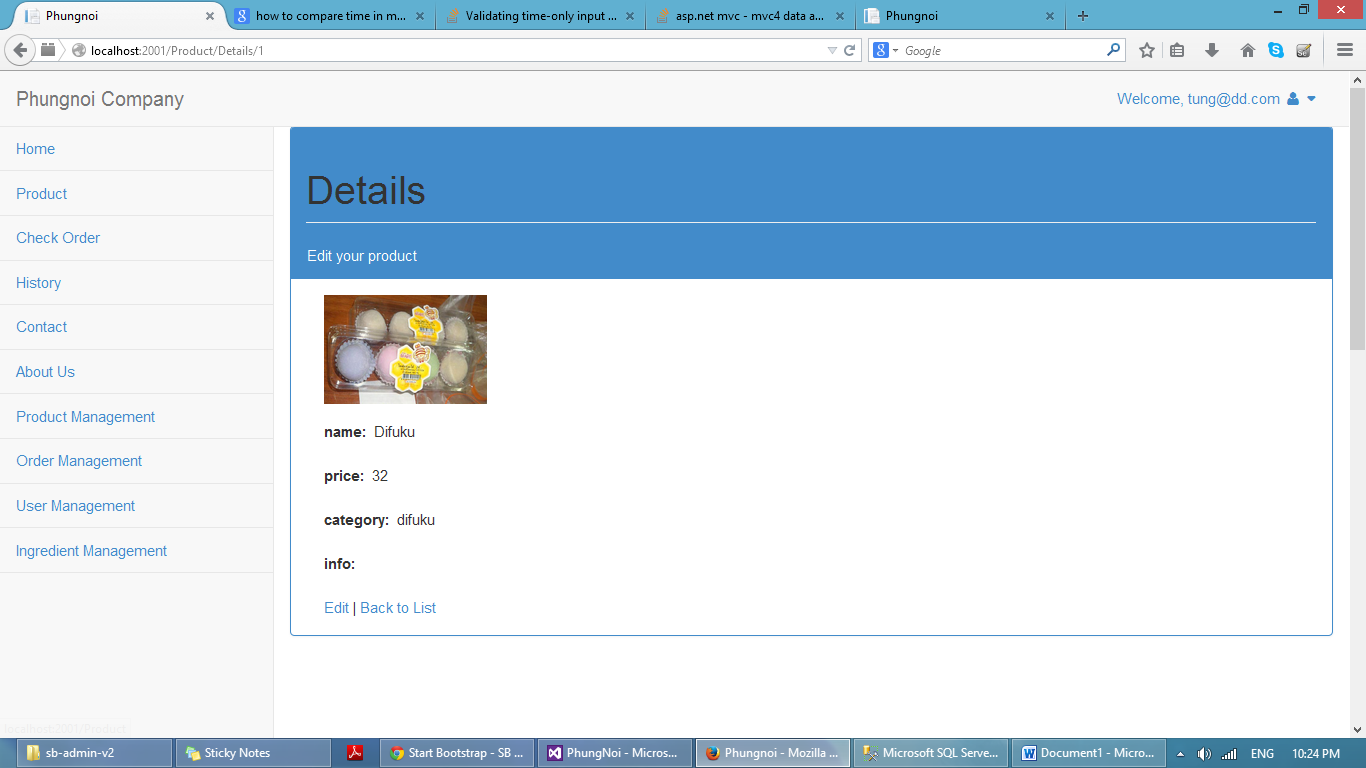
****

**Figure 25:** UI – 11 Edit bakery product page

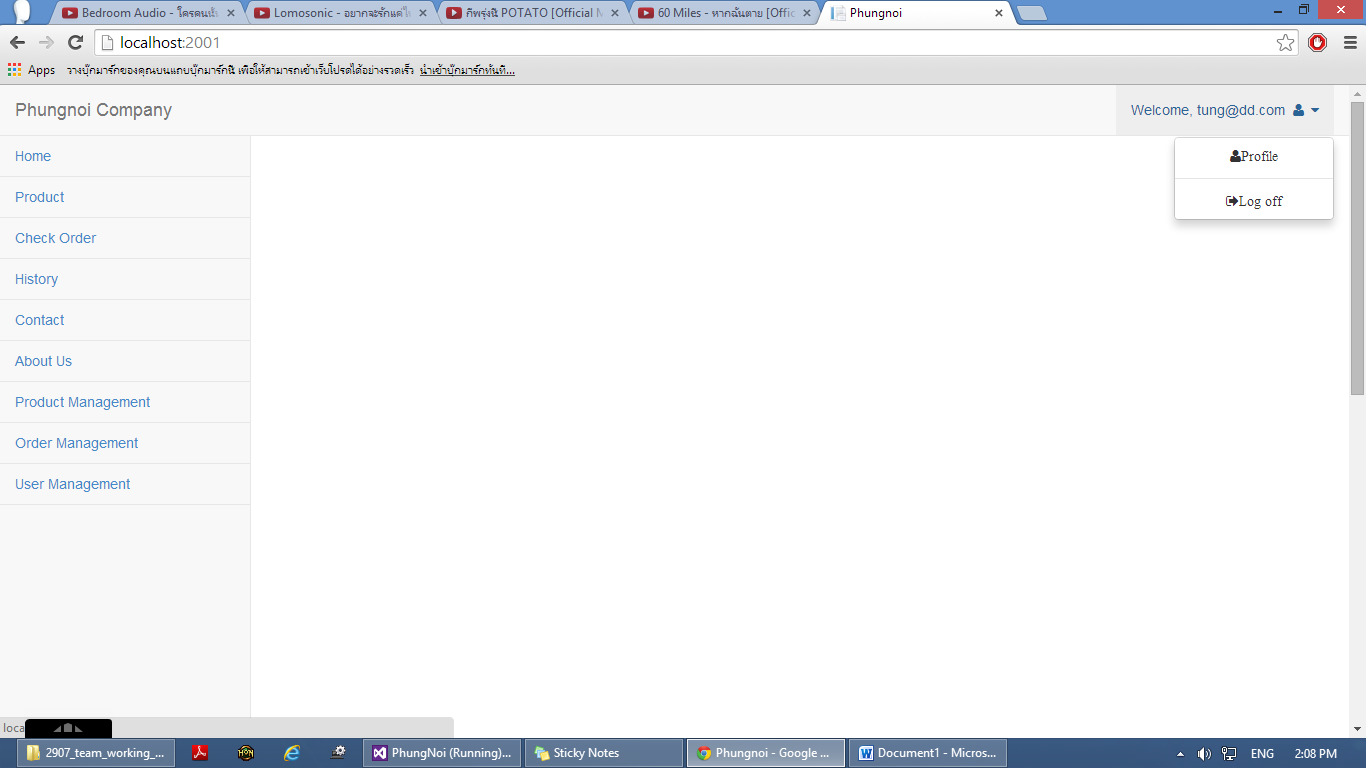
**Figure 26:** UI – 12 Delete bakery product page



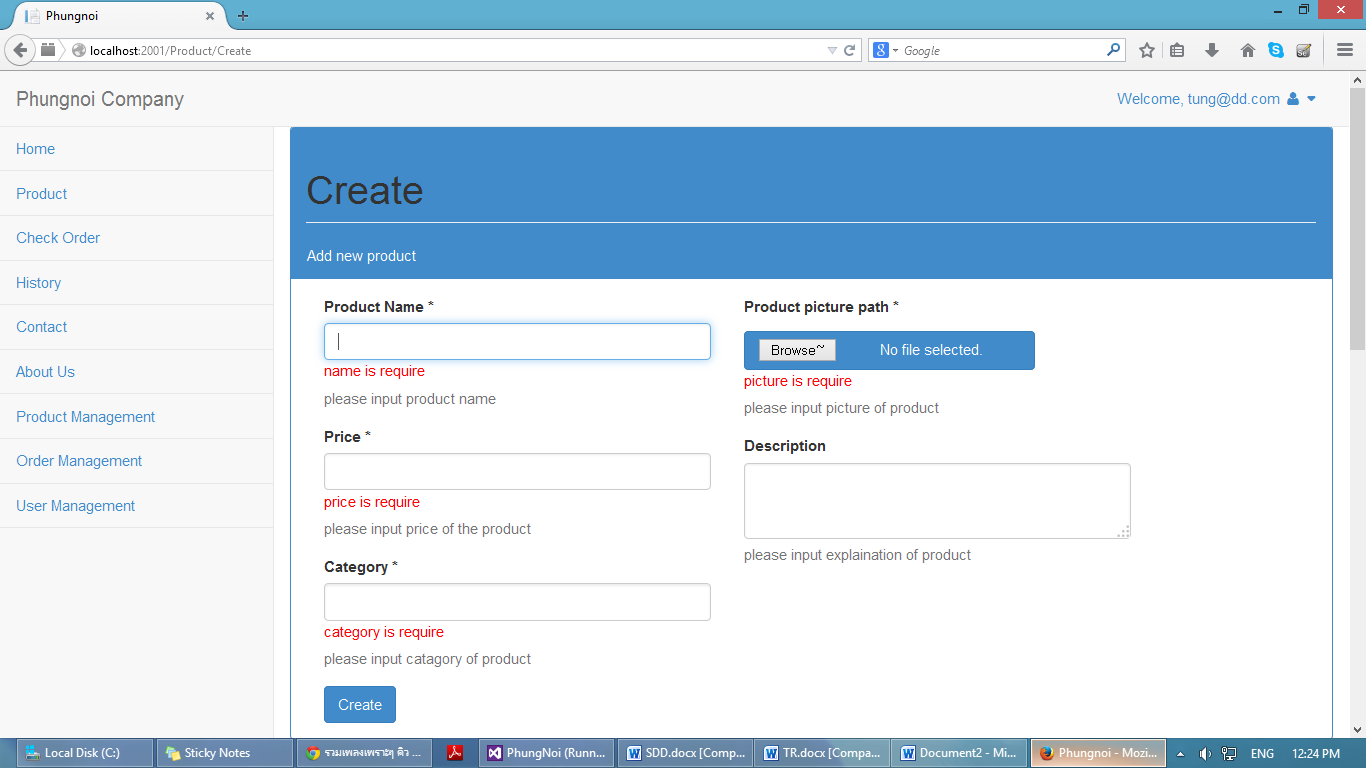
**Figure 27:** UI – 13 Bakery product list page



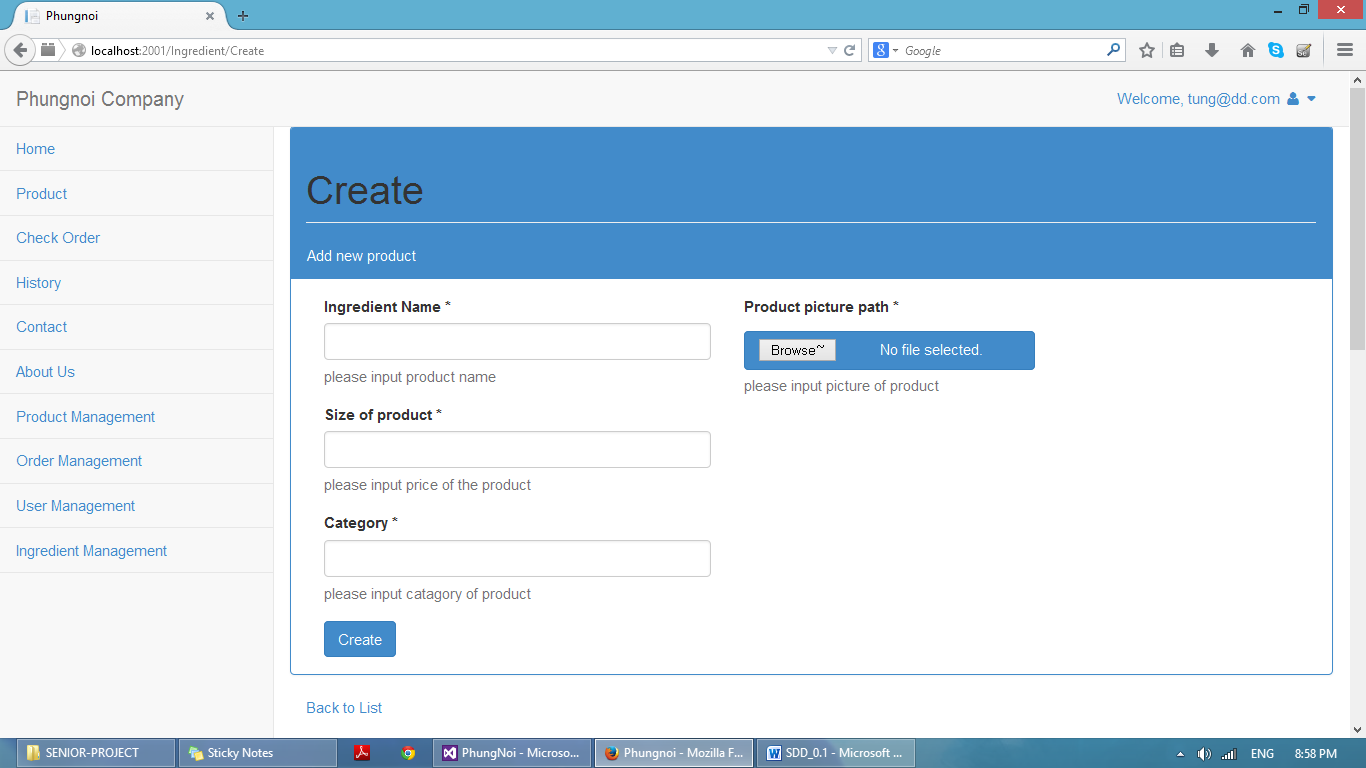
**Figure 28:** UI – 14 Bakery product details page



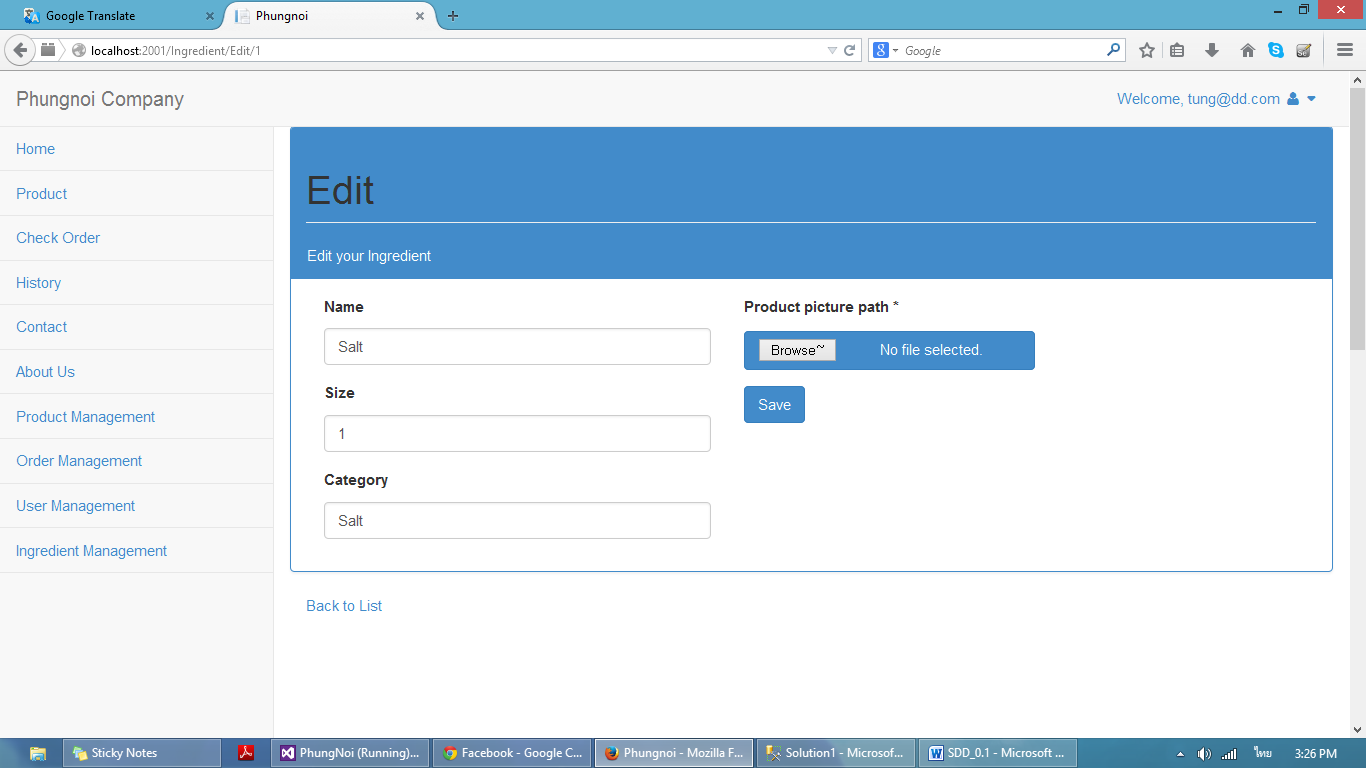
**Figure 29:** UI – 15 Logout page



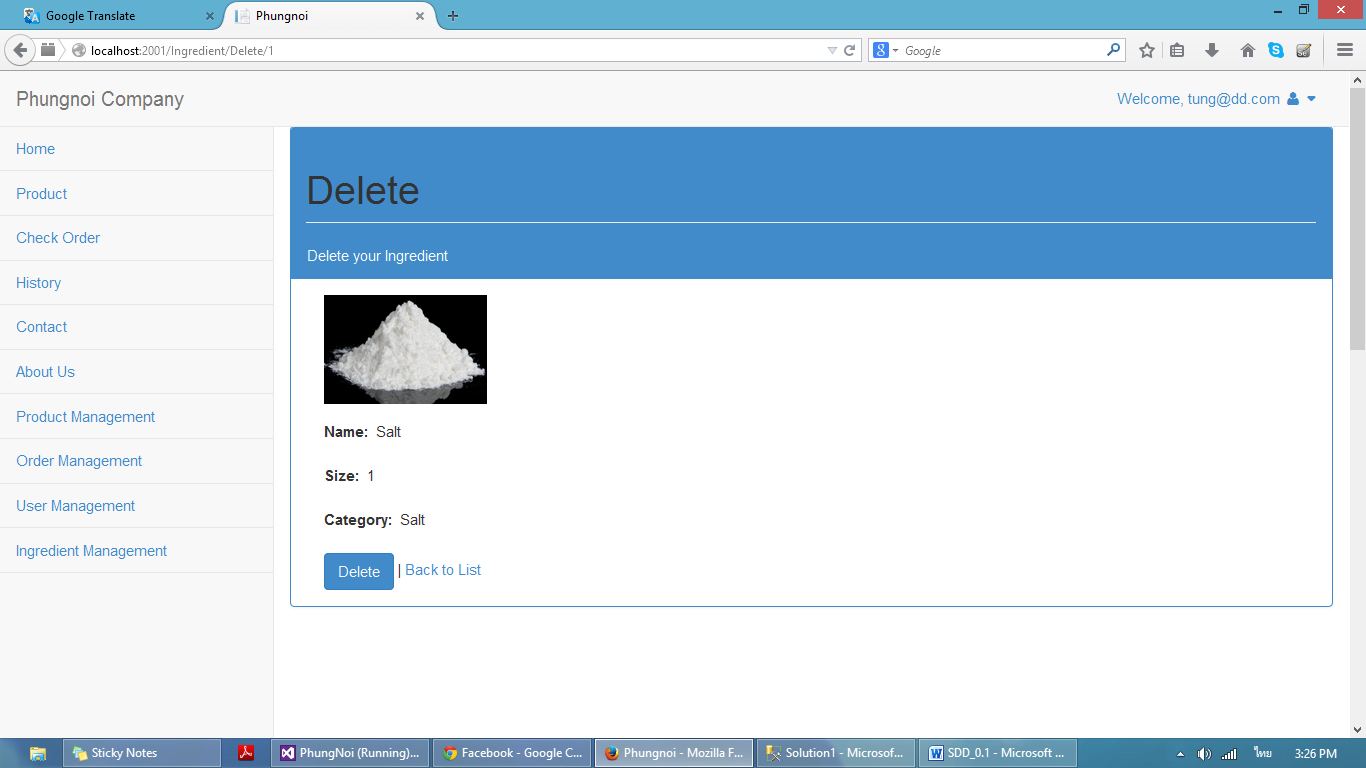
**Figure 30:** UI – 16 Validation form page



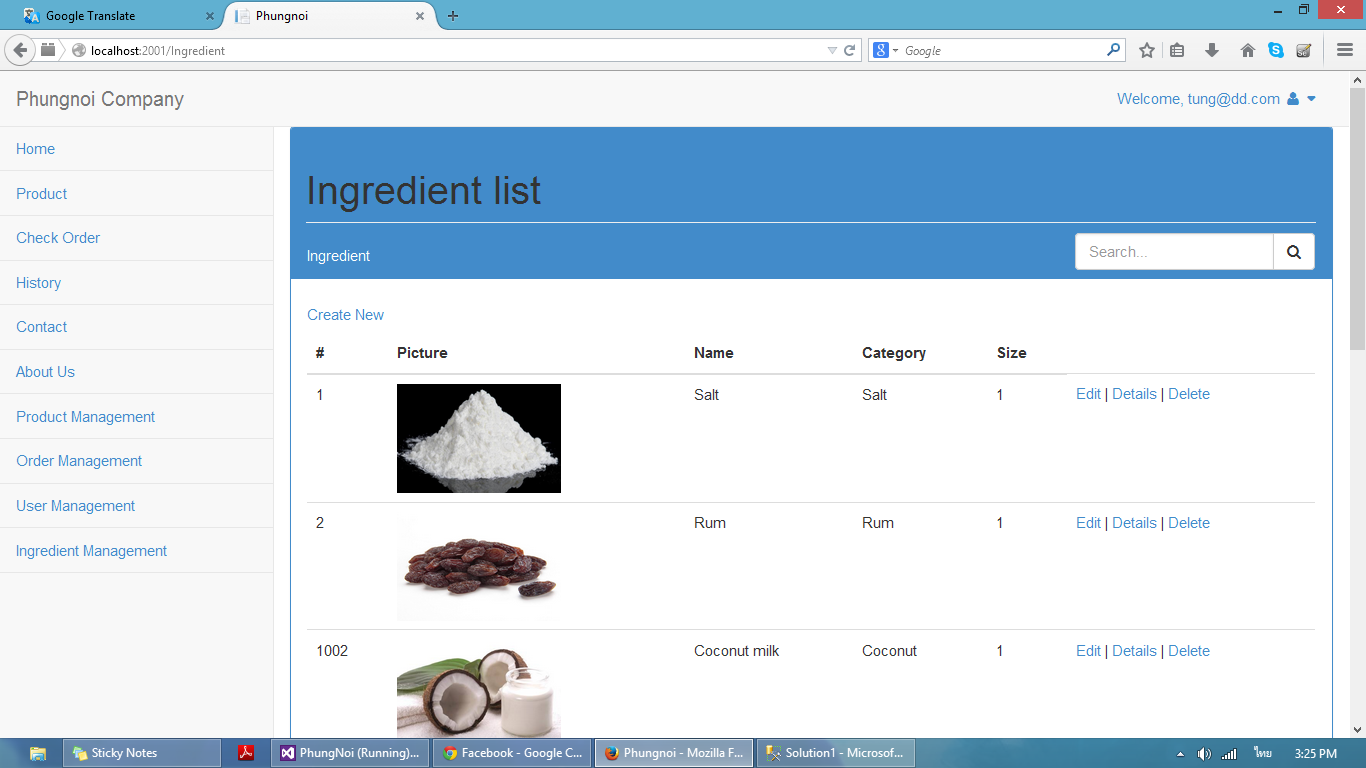
**Figure 31:** UI – 17 Create ingredient information page



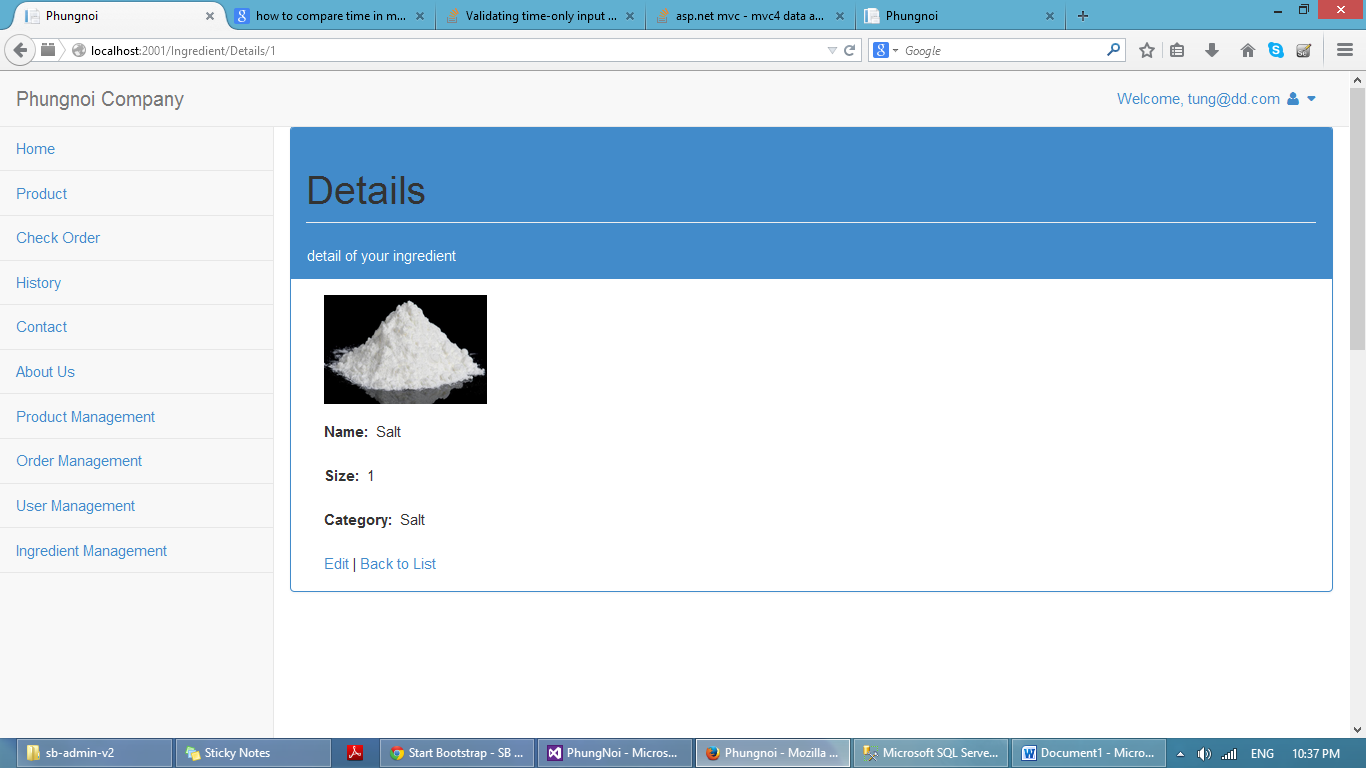
**Figure 32:** UI – 18 Edit ingredient information page



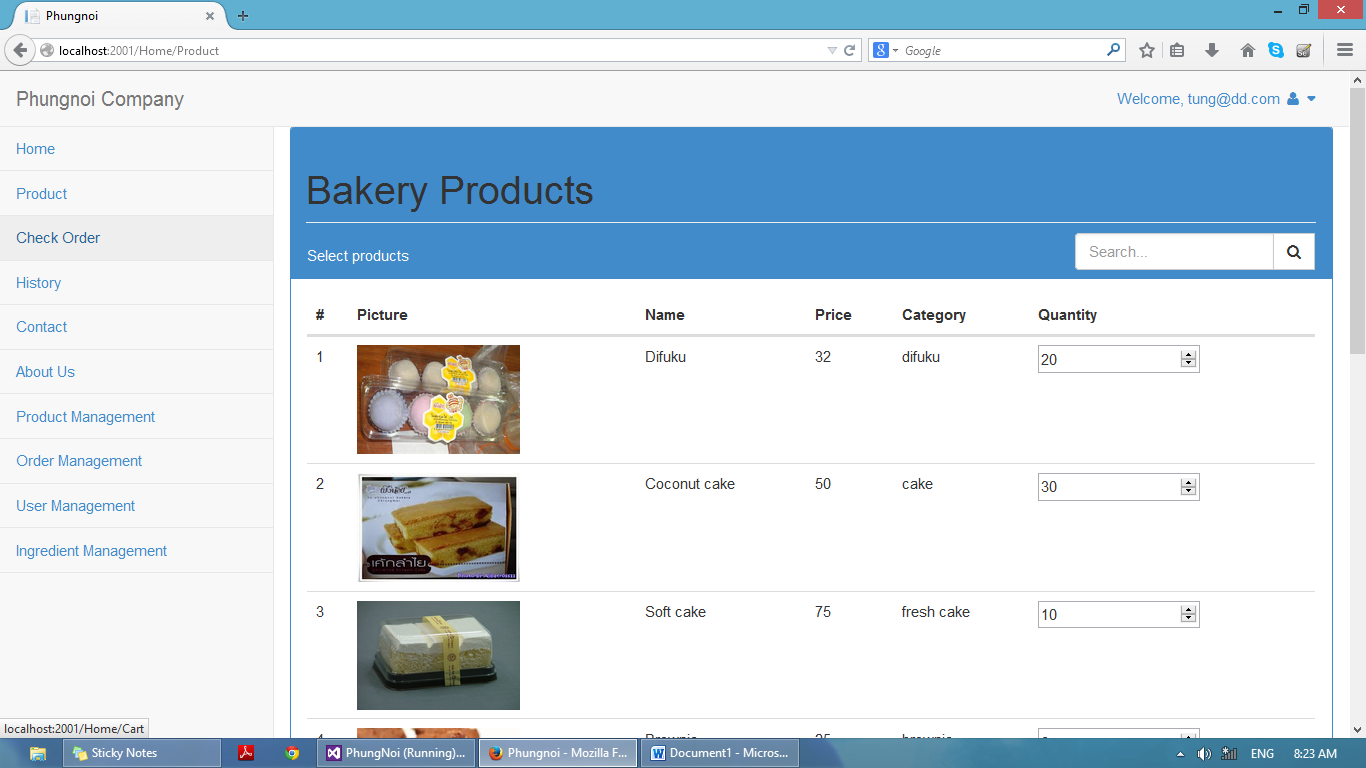
**Figure 33:** UI – 19 Delete ingredient page



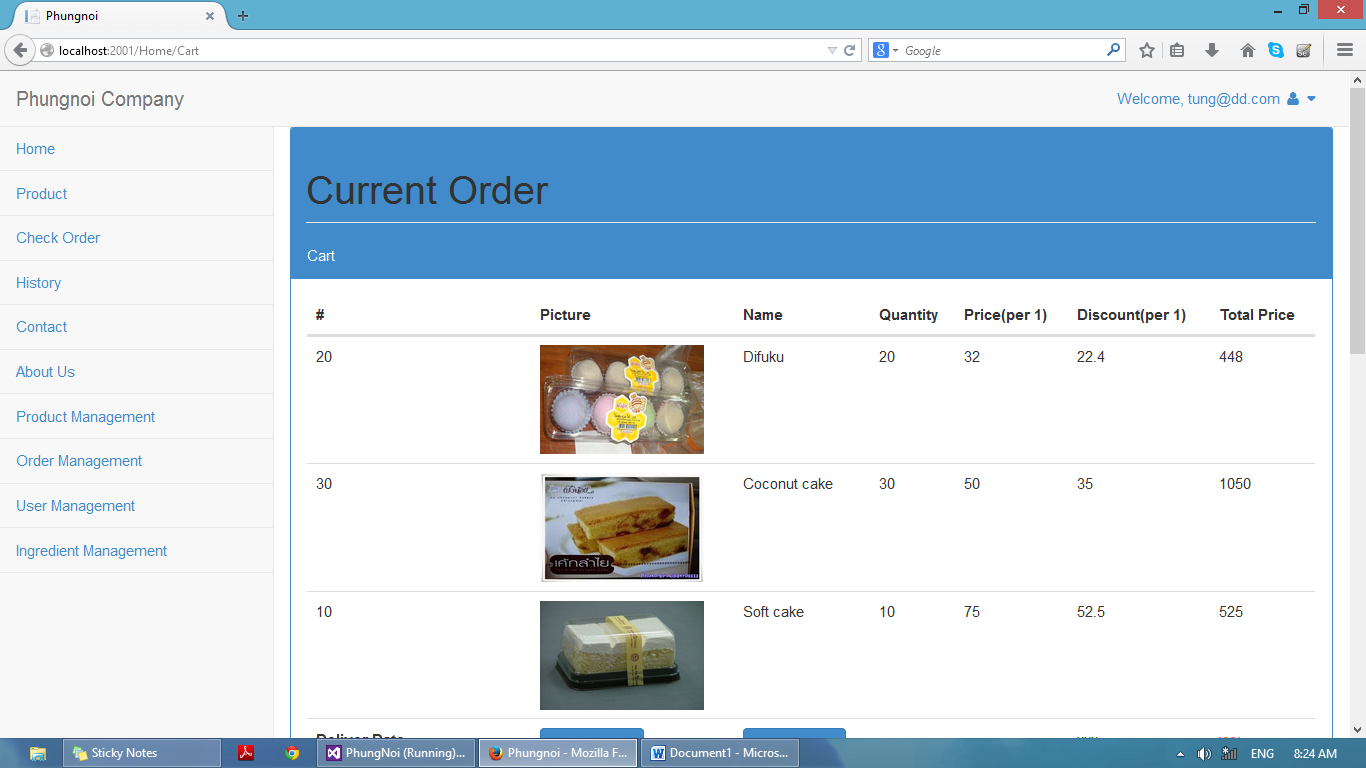
**Figure 34:** UI – 20 Ingredient list page



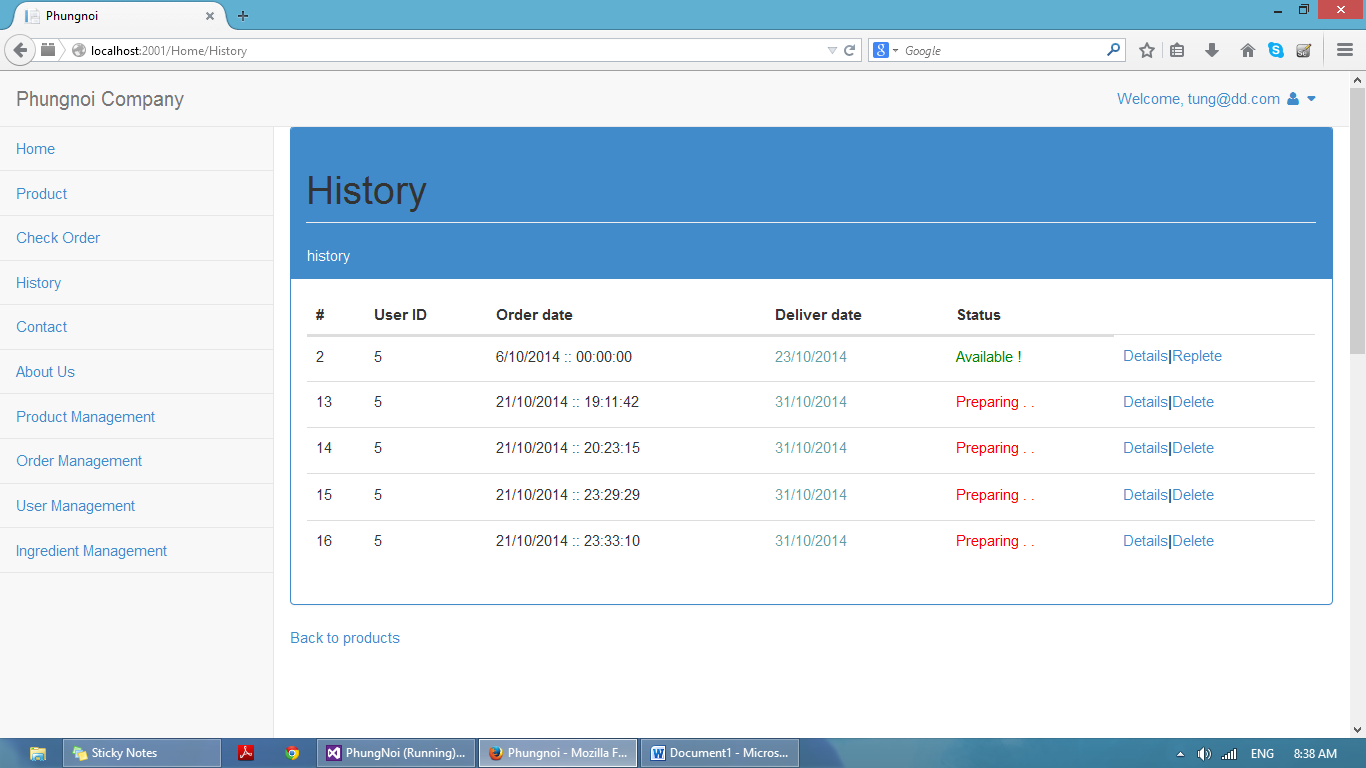
**Figure 35:** UI – 21 Ingredient Details page



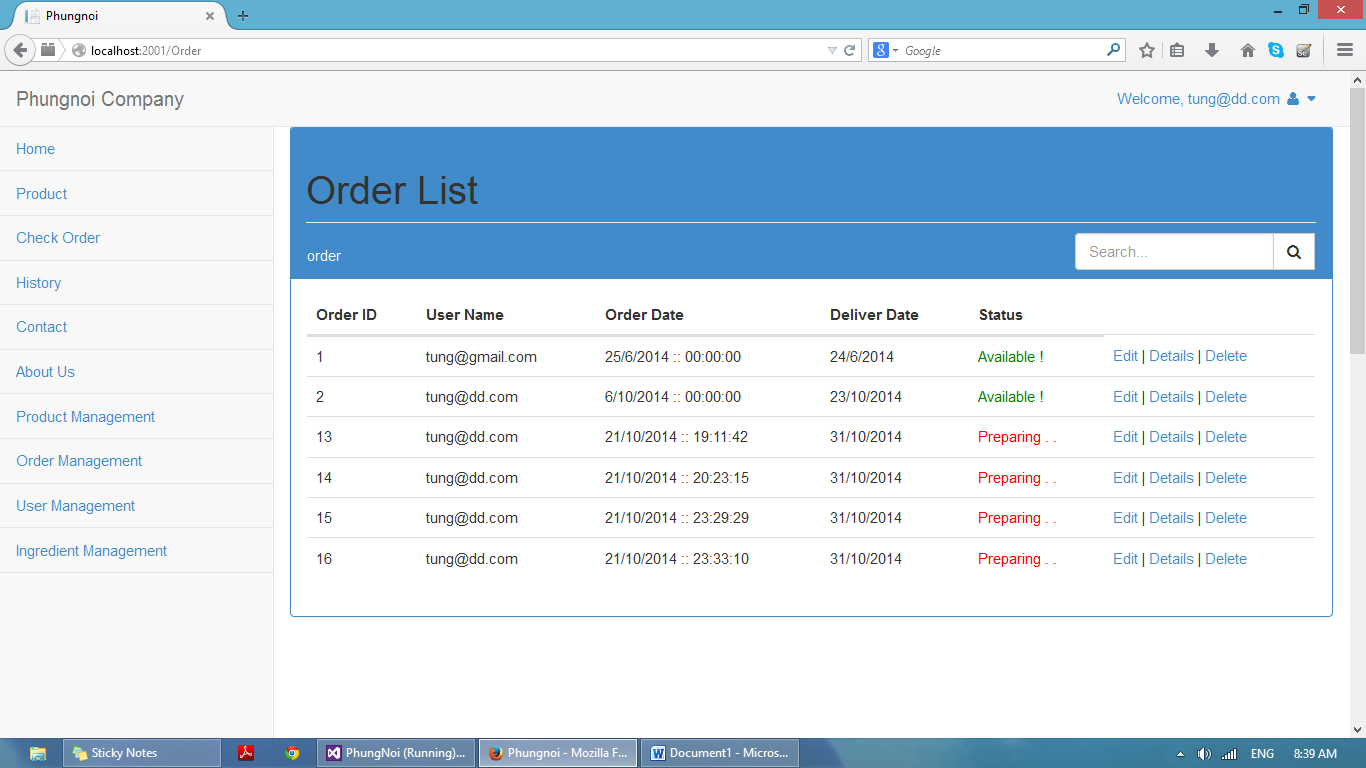
**Figure 36:** UI – 22 Order product page



**Figure 37:** UI – 22 Bakery product cart page



**Figure 38:** UI – 23 History page



**Figure 39:** UI – 24 Order list page