**FPT ACADEMY INTERNATIONAL**



**FPT – APTECH COMPUTER EDUCATION**

Saigon Bakery

|  |  |  |  |
| --- | --- | --- | --- |
| **Center Name:** | FPT – Aptech – HCM2 | | |
|  | Address:590 Cach Mang Thang Tam,  District 3, Ho Chi Minh City | | |
| **Instructor:** | Mrs. Le Mong Thuy | | |
| **Class:** | T1.2008.A0 | | |
| **Group:** | 05 | | |
| **Group members** | Name  Phan Phuc Thinh | | Student ID  Student1251715 | |
|  |
|  | Nguyen Nhat Tan  Nguyen Do Hong Quang | | Student1198574  Student1272823 | |
|  |  | |  | |
| **Project Duration** | Start date | 23th June 2021 | | |
|  | End date | 5th August 2021 | | |

**This is to certify that**

**Mr./Ms.:**

Phan Phuc Thinh

Nguyen Nhat Tan

Nguyen Do Hong Quang

Has successfully designed & developed

Project: Saigon Bakery

**Submitted by:**

Phan Phuc Thinh

**Date of issue:** June 23th 2021

**Authorized Signature:**

CONTENT

[REVIEW 1 2](#_Toc61824781)

[1. ACKNOWLEDGE 2](#_Toc61824782)

[2. INTRODUCTION 2](#_Toc61824783)

[3. PROBLEM DEFINITION 2](#_Toc61824784)

[4. CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS) 2](#_Toc61824785)

[4.1 CLIENT: 2](#_Toc61824786)

[5. TASK SHEET REVIEW 1 2](#_Toc61824787)

[REVIEW 2 2](#_Toc61824788)

[1. USECASE 2](#_Toc61824789)

[2. DATA FLOW DIAGRAM (DFD) 2](#_Toc61824790)

[3. ENTITY PROPERTIES 2](#_Toc61824791)

[4. Order\_Detail 2](#_Toc61824792)

[5. ENTITY RELATIONSHIP DIAGRAM (ERD) 2](#_Toc61824793)

[6. TABLE DESIGN 2](#_Toc61824794)

[7. TASK SHEET REVIEW 2 2](#_Toc61824795)

[REVIEW 3 2](#_Toc61824796)

[1. SITE MAP 2](#_Toc61824797)

[2. SAIGON BAKERY DESIGN 2](#_Toc61824798)

[TASK SHEET REVIEW 3 Error! Bookmark not defined.](#_Toc61824799)

# REVIEW 1

## ACKNOWLEDGE

Our group would like to devote this chapter to express our sincere appreciation to all those who offered us valuable assistance during in carrying out this Project.

Firstly, we owe a debt of gratitude to our dedicated instructor, Mrs. Le Mong Thuy, whose suggestions and encouragement helped us to overcome not only technological challenges but also psychological tension. Without him, our work could not complete successfully.

Secondly, we highly appreciate the Project Team at the Head Office, which has provided us an excellent opportunity to apply what we learned into practice. We are certain that the experiences we gained from this project will facilitate our future career greatly.

Thirdly, we would also like to express my gratitude to all the staff members of my center for not only providing me with the opportunity to work with them on this project, but also for their support and encouragement throughout the process.

Finally yet importantly, we wish to say a personal thank you to all our dear classmates and team members who are always willing to support us openly and enthusiastically.

## INTRODUCTION

We are in an era where information technology makes buying and selling easier than ever. People can order an expensive phone to a daily kitchenware and everything is completed electronically and interactively in real- time. Saigon Bakery is also out of this cycle. Therefore, we are developing a website named Saigon Bakery to bring good products to more customers.

Saigon Bakery is for an organization who wants to develop an online store where their users can be a person who does not have much time to go to the Saigon Bakery store or person has shopping online habit or just simply they stay too far from standalone store. It will provide its user the following features.

For Saigon Bakery’s customer:

* Customer can register account.
* Customer can login and manage their account.
* Customer can search any products such as vegetables, food, canned foods, etc.
* Customer can search products by price.
* Customer can see products’ introduction before buying.
* Customer can buy products after adding products to cart.
* Customer can view cart history.
* Customer can check out.
* Customer can interact with website by giving question, feedback on contact us function.

## PROBLEM DEFINITION

After analyzing the project specifications of Saigon Bakery Store, to fulfil customer’s requirements properly, we came to conclusion that in order to achieve the best result, we should deal with following queries:

### From the view point of visitors

* A homepage will show the main products, suitable logo, display the logo on all the pages in the site.
* A menu that will include all the functions implemented in the website.
* Customer can buy products, add product to cart, view cart history and check out after login their account.

### From the view point of the Administrator

* A login page will grant access to Saigon Bakery’s admin.
* Admin can view, add, update, and delete products.
* Admin can view, add, update, delete, find, lock, and unlock customer account.
* Admin can manage blogs and orders.

## CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS)

### **CLIENT:** Saigon Bakery Store

### ****BUSINESS/PROJECT OBJECTIVE:****

The project is to design an online website for selling Saigon Bakerys, for instant: fast foods, fresh foods, etc.

#### ****Input to the system:****

**Saigon Bakery store’s administrators need to provide:**

* Information required in order creating account to access to the admin system.
* Products are going to be shown on the website.

**Saigon Bakery store’s customers need to provide:**

* Personal information to buy products.

#### ****Output from the system:****

**Saigon Bakery store’s administrators will be able to:**

* Login/Logout to the admin system.
* Manage (view, add, update, delete) products, price, promotion.
* Manage (view, add, update, delete, find, lock, unlock) customer account.
* Manage (view, add, update, delete) blogs.
* Manage (view, update) products, order.

**Saigon Bakery store’s guide will be able to:**

* Register customer account.
* Buy products after add information.
* View products that they have chosen.
* View order history.

**Saigon Bakery store’s customer will be able to:**

* Register account.
* Login/Logout to the account.
* Search products they want to buy by name, price.
* Add product to cart and checkout.
* View purchase history.

### SCOPE OF WORK

The application should perform the following functions:

#### Login****:****

* There are 1 types of account in the application (“admin”). Administrator will login as “admin”.
* For “admin”: can view the products id, name, add new, update, delete.
* For “admin”: can view the customer id, name, add new, update, delete, lock, unlock.
* For “admin”: can view and manage view.
* For “admin”: can view and manage order.

#### ****Buying products:****

* When customer choose a product and the selected product will be added to the cart.
* Customer can buy products at shopping cart after pressing the “Checkout” button and fill in the information.

#### ****Database:****

* The information about products, customers, blogs, orders will be stored in the database.
* The information from the contact us page.

### HARDWARE/ SOFTWARE REQUIREMENTS:

#### ****Hardware****

##### **Web Server**

|  |  |
| --- | --- |
| **Processor** | Intel Core I5 or higher. |
| **Memory** | 4 GB RAM or greater. |
| **Modem/ADSL** | Internet access is required. |

##### **Client**

|  |  |
| --- | --- |
| **Processor** | Intel Core I3 or higher. |
| **Memory** | 2 GB RAM or greater. |
| **Monitor** | Super VGA (1024x768) or higher resolution. |
| **Modem/ADSL** | Internet access is required. |

#### ****Software****

##### **Web Server**

|  |  |
| --- | --- |
| **Operation System** | Window 7 or later. |
| **Browser** | Google Chrome version 35. |
| **Database** | MySQL. |
| **Software** | XAMPP-Apache |
| PHP - Laravel framework |

##### **Client**

|  |  |
| --- | --- |
| **Operation System** | Window 7 or higher. |
| **Browser** | Google Chrome version 35. |

#### ****Development software****

* Microsoft Windows 10.
* MySQL.
* Visual Studio Code.
* Google Chrome.
* XAMPP.

#### ****Technology****

* PHP - Laravel framework.
* JQuery.
* Bootstrap.
* HTML/CSS

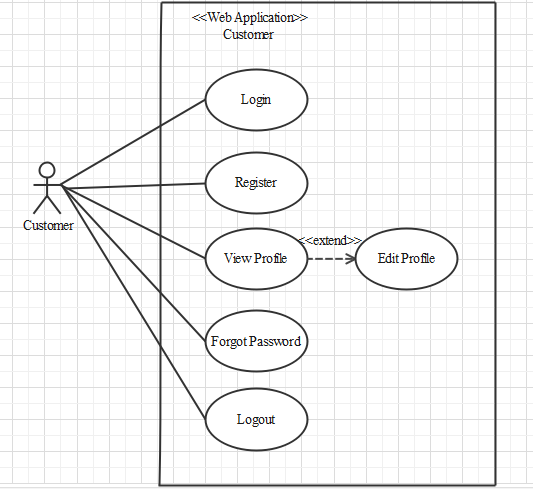
## TASK SHEET REVIEW 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Ref. No.: Saigon Bakery Store | | Project Name | Activity Plan Prepared By | Date of Preparation of Activity Plan: | | | | |
| Sr./No. | Task | Actual Start Date | Actual Days | Team member name | Status |
| 1 | Acknowledgment | Saigon Bakery | Phan Phuc Thinh | 23-June-21 | 1 | All Member | Completed |
| 2 | Introduction | 23-June-21 | 1 | All Member | Completed |
| 3 | Problem Definition | 23-June-21 | 1 | All Member | Completed |
| 4 | Customer Requirement Specification (CRS) | 23-June-21 | 3 | All Member | Completed |
| 5 | Scope of Work | 23-June-21 | 2 | All Member | Completed |
| 6 | Hardware/Software Requirement | 23-June-21 | 2 | All Member | Completed |
| 7 | Task Sheet | 23-June-21 | 1 | Thinh | Completed |

|  |  |
| --- | --- |
| Date: 23 June 2021 | |
| Signature of Instructor:  **Mrs. Le Mong Thuy** | Signature of Team Leader:  **Phan Phuc Thinh** |

# REVIEW 2

## USECASE



### CUSTOMER

#### LOGIN AS CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Login | |
| **Actors** | Customer | |
| **Description** | Customer who has registered an account can login | |
| **Requirements** | Customer provides username and password | |
| **Pre-conditions** | Customer must register an account | |
| **Post-conditions** | Success: Customer is logged in to website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Customer clicks ‘Login’ button on Home Page.  3. Actor inputs Email and Password, then click the ‘Login’ button. | 2. System redirects to Login Page with the following controls:  - “Email” text field  - “Password” text field  - “Login” button  4. System checks the information.  5. System redirects to Customer’s dashboard. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid email and password. | System redirects to Login page with the following controls:  - “Email” text field  - “Password” text field  - “Login” button  System shows message: “Invalid email and password”. |

#### REGISTER AS CUSTOMER:

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Register as Customer | |
| **Actors** | Customer | |
| **Description** | Customer who has registered an account can login | |
| **Requirements** | Customer registers an account on the website and becomes a member | |
| **Pre-conditions** | Customer provides their register information | |
| **Post-conditions** | Success: Customer creates a User account in this website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Customer clicks “Register” button on Home Page.  3. Customer inputs register’s information and click “Register” button. | 2. System redirects to Register page and displays Registration form with the following controls:  - “Username” text field  - “Password” text field  - “Confirm Password” text field  - “Email” text field  - “Phone” text field  - “Register” button  - “Reset” button  4. System validates the information.  5. System inserts the account into database  6. System show success message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| - Actor left “Username” text field blank.  - Username has already existed.  - Actor left “Password” text field blank.  - Actor left “Confirm Password” text field blank.  - Password and confirm password don’t match.  - Actor left “Email” text field blank.  - Email has already existed.  - Actor left “Phone” text field blank.  - Input phone is not numeric format or phone length is not between 8 and 14 characters.  - Actor left “Email” text field blank or input an invalid email address. | System displays message to customer:  - “Username cannot be blank”.  - “This Username has already existed”.  - “Password cannot be blank”.  - “Confirm Password cannot be blank”.  - “Confirm Password does not match password.”.  - “Email cannot be blank”.  - “This Email has already existed”.  - “Phone cannot be blank”.  - “The phone number should be numeric only and must be between 8 and 14 digits”.  - “Email is invalid”. |

#### FORGOT PASSWORD

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Forgot password | |
| **Actors** | Customer | |
| **Description** | Customer who has forgotten password can make a request to change password | |
| **Requirements** | Customer provides email address | |
| **Pre-conditions** | Customer must be logged on | |
| **Post-conditions** | Success: Actor can change their password | |
| Fail: Password can’t change | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks on “Forgot Password” button.  3. Actor inputs Email and clicks “Submit”.  5. Actor inputs reset token in Password Page and click “Submit”.  7. Actor inputs new password and click “Save”. | 2. System redirects to Password Page with the following controls:  - “Email” text field  - “Submit” button  4. System checks the information and sends a reset token to user.  6. System redirects to Password form with the following controls:  - “New Password” text field  - “Confirm Password” text field  - “Save” button  8. System validates password then updates password in database.  9. System shows message: “Your password has been changed successfully”. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs an email that does not exist in database. | 2. System shows message: “Invalid email”. |

#### VIEW PROFILE

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | View profile | |
| **Actors** | Customer | |
| **Description** | Customer who want to view their profile | |
| **Requirements** | Customer provides their personal information | |
| **Pre-conditions** | Customer is logged in | |
| **Post-conditions** | Success: Actor can view their profile | |
| Fail: Actor cannot view their profile | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks “Account” button on Home Page. | 2. System redirects to Account page and displays Account form with the following controls:  - “Fullname” text field  - “Address” text field  - “Email” text field  - “Phone” text field  - “Birthday” text field |

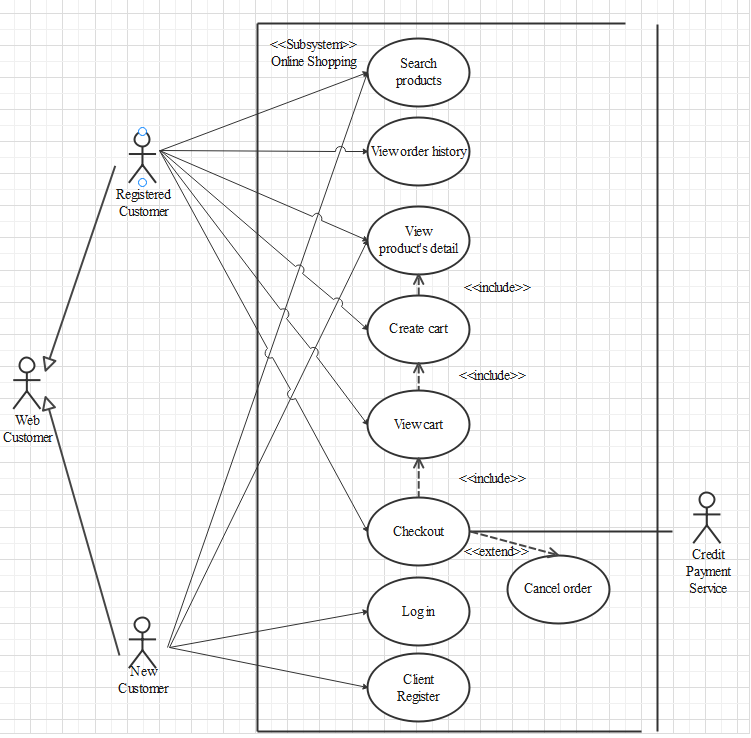
#### EDIT PROFILE:

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Edit profile | |
| **Actors** | Customer | |
| **Description** | Customer who want to update their profile | |
| **Requirements** | Customer provides their personal information | |
| **Pre-conditions** | Customer is logged in | |
| **Post-conditions** | Success: Actor can edit their profile | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks “Account” button on Home Page.  3. Actor inputs personal information and clicks “Submit”. | 2. System redirects to Account page and displays Account form with the following controls:  - “Fullname” text field  - “Address” text field  - “Email” text field  - “Phone” text field  - “Birthday” text field  - “Submit” button  4. System validates password then sends to database.  5. System shows message: “Your account has been update successfully”. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor loggin an account which does not exist in database. | System shows message: “Invalid account”. |

#### LOGOUT

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Logout | |
| **Actors** | Customer | |
| **Description** | Customer clicks on “Logout” button and their session is terminated. | |
| **Requirements** | N/A | |
| **Pre-conditions** | 1. Customer is logged in  2. Customer no longer wants to be logged in | |
| **Post-conditions** | Success: Customer can logout | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks “Log out” button on Home Page. | 2. The system logs the actor out and invalidates the session.  3. The system redirects to the default logout page. |
| **Exceptions** | Actor’s actions: | System’s responses: |
| 1. Actor cannot logout | System shows message: “Not logged in”. |

### CART



#### SEARCH PRODUCTS

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Search product | |
| **Actors** | Registered Customer/ New Customer | |
| **Description** | Customer can search product | |
| **Requirements** | Customer provides search phrase | |
| **Pre-conditions** | N/A | |
| **Post-conditions** | Success: Show product that matches search phrase | |
| Fail: No product is shown | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs search phrase.  3. Actor clicks product’s name in list or view every product that matches search phrase. | 2. System shows list of products that match the search phrase. |

#### VIEW PRODUCT DETAIL

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | View product | |
| **Actors** | Registered Customer/ New Customer | |
| **Description** | Customer can view product detail | |
| **Requirements** | Customer select product to show product’s detail | |
| **Pre-conditions** | N/A | |
| **Post-conditions** | Success: Display product detail matching the selected product | |
| Fail: No product is shown | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor select product | 2. System shows product’s detail matching selected product |

#### CREATE CART

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Create cart | |
| **Actors** | Registered customer | |
| **Description** | Customer can add product to their cart | |
| **Requirements** | Customer must choose at least one product | |
| **Pre-conditions** | Customer must log in before adding product | |
| **Post-conditions** | Success: Actor add product to cart successfully | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses a product and click “Add to cart”. | 2. System adds selected product to cart by id of this product. |
| **Exceptions** | **Actor’s action** | **System’s response** |
| 1. Actor inputs more products than there are in stock. | 2. System shows messages:  - “Product out of stock”. |

#### VIEW CART

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | View cart | |
| **Actors** | Registered customer | |
| **Description** | Customer can view their cart | |
| **Requirements** | N/A | |
| **Pre-conditions** | Customer must log in before view cart | |
| **Post-conditions** | Success: Customer can view their cart | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click on “Cart” icon. | 2. System shows cart and added products. |
| **Alternative flow** | 1. Actor clicks “x”– Close button- on table  3. Actor change quantity of a product  5. Actor clicks “x” – close button – in the right side of each product. | 2. System redirects to current page.  4. System update quantity of this product into cart.  6. System removes this product from cart. |

#### CHECKOUT

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Check out and Place order | |
| **Actors** | Registered customer | |
| **Description** | Customer can order products | |
| **Requirements** | Customer must has at least 1 product in cart | |
| **Pre-conditions** | Customer must log in before adding product | |
| **Post-conditions** | Success: Actor order successfully | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor click “Add to cart” button or “Cart” icon.  3. Actor clicks “Check out with” button.  5. Actor inputs order information then click “Place Order” button.  6. Actor inputs Card owner’s name, card number and security code if actor choose payment by credit card.  9. Actor views order’s status in Order page. | 2. System shows cart and added products.  4. System shows all products in cart with calculated total price and show following controls:  - “Customer name” text field.  - “Phone” text field.  - “Address” text field.  - “Province” select box.  - “Delivery time frame” select box.  - “Payment” table.  Systems shows Payment tab type according to type of payment actors choose.  7. System validates information.  8. System redirects to Order page and show order’s status. |
| **Alternative flow** | 1. Actor clicks “x”– Close button- on table  3. Actor clicks “X” button on each product.  5. Actors clicks “Home” button. | 2. System redirects to cart page.  4. System removes product from cart.  6. System redirect to Home pages. |
| **Exceptions** | **Actor’s action** | **System’s response** |
| 1. Actor inputs more products than there are in stock.  2. Actor left “Billing name” text field blank.  3. Actor left “Address” text field blank.  4. Actor left “Delivery date” blank.  5. Actor left “Delivery time frame” blank.  6. Actor left “Payment” dropdown blank.  7. Actor inputs invalid card information. | 8. System shows messages:  - “Product out of stock”.  - “Order name must not be blank”.  “Address must not be blank”.  “Must choose a delivery date”.  “Must choose a delivery time frame”.  “Must choose a payment method”.  “Invalid card”. |

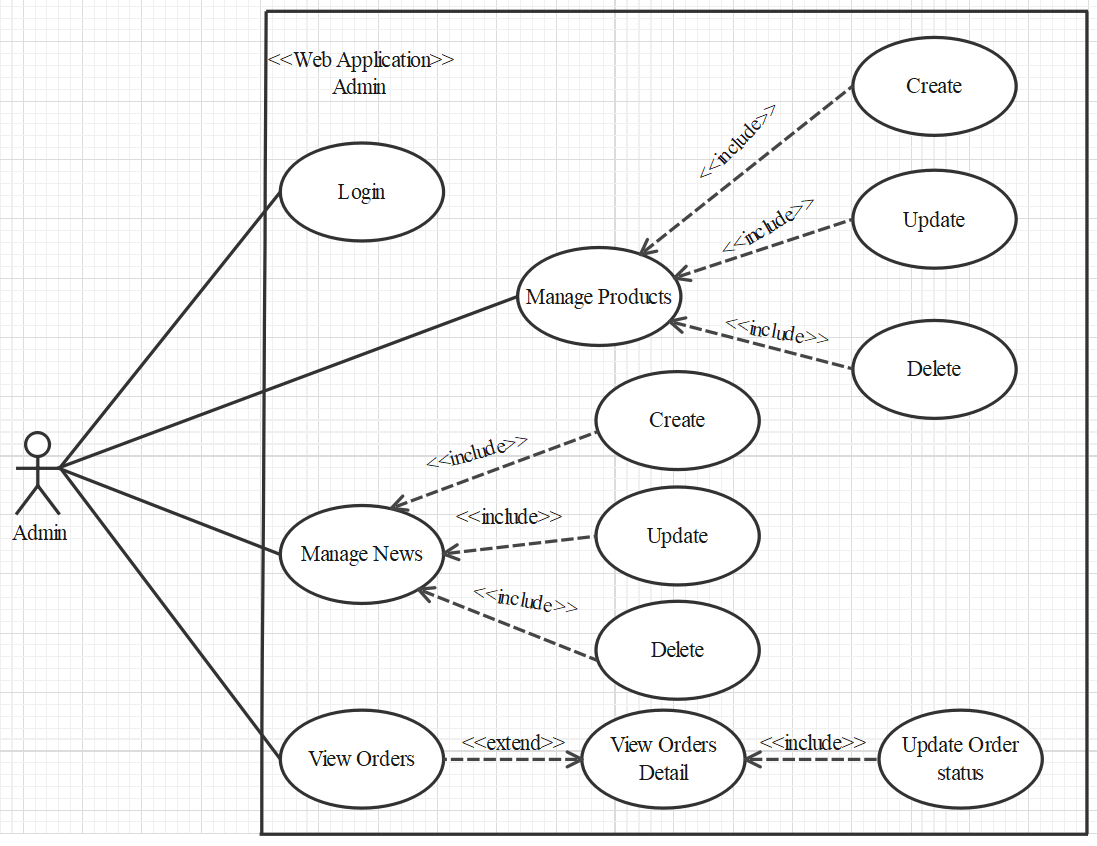
#### VIEW ORDER HISTORY

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | View order history | |
| **Actors** | Registered customer | |
| **Description** | Customer can view their order history | |
| **Requirements** | N/A | |
| **Pre-conditions** | Customer must log in to website | |
| **Post-conditions** | Success: Customer can view their order. | |
| Fail: Customer cannot see their order. | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses “Account” icon on Menu  3. Actor chooses “Purchase history” on sides bar.  5. Actor selects the order that they want to view in “Purchase history” and clicks “View” Button. | 2. System redirects to Account page.  4. System redirects to “Purchase history” page and displays a list of existing orders and their status.  6. System validates inputs by id of Order.  7. System displays all order’s detail and its status. |

#### CANCEL ORDER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Cancel order | |
| **Actors** | Registered customer | |
| **Description** | Customer can cancel their order | |
| **Requirements** | Customer provides order’s id to cancel | |
| **Pre-conditions** | Customer must log in to website and only can cancel when this order has not been confirmed by admin | |
| **Post-conditions** | Success: Cancel successfully. | |
| Fail: Order cannot cancel. | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses “Account” icon on Menu  3. Actor chooses “Purchase history” on sides bar.  5. Actor selects the order that they want to cancel in “Purchase history” and clicks “Cancel” Button. | 2. System redirects to Account page.  4. System redirects to “Purchase history” page and displays a list of existing orders.  6. System validates inputs by id of order.  7. System change order status in database.  8. System displays message: successful for error. |

### ADMIN



#### LOGIN

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Login | |
| **Actor** | Admin | |
| **Description** | When admin wants to manage products, blogs, views orders and orders detail | |
| **Requirements** | Admin provides username and password | |
| **Pre-conditions** | Admin must be exist on database | |
| **Post-conditions** | Success: Admin is logged in to website admin | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks on ‘Login’ button.  3. Actor inputs Email and Password, then click the ‘Login’ button. | 2. System redirects to Login Page with the following controls:  - 'Email' text field  - 'Password' text field  - 'Login' button  4. System validates the information.  5. System redirects to Admin website |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid email and password | System redirects to Login page with the following controls:  - ‘Email’ text field  - ‘Password’ text field  - ‘Login’ button. |

#### PRODUCTS

##### Create new Product

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Create new product | |
| **Actor** | Admin | |
| **Description** | When admin wants to create a new product | |
| **Requirements** | Actor provides product’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: new product is added into the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks ‘Add New Product’ button.  5. Actor inputs information to create new product | 2. System redirects to ‘Product’ page and display a list of existing product.  4. System displays a form with the following fields:  - id  - category\_id  - sale\_id  - name  - name\_slug  - description  - price  - view\_count  - quantity  - active  - detail  - image  6. System validates inputs.  7. System adds product into database.  8. System displays confirmation message or error message. |

##### Update Product

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Update product’s detail | |
| **Actor** | Admin | |
| **Description** | When admin wants to update product | |
| **Requirements** | Admin provides product’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: product is updated in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks on ‘View Details’ on a product.  5. Actor chooses the criteria they want to update and input information. | 2. System redirects to ‘Product Details’ page and display a list of existing product.  4. System displays a form with the following fields.  - category\_id  - sale\_id  - name  - name\_slug  - description  - price  - view\_count  - quantity  - active  - detail  - image  6. System validates inputs.  7. System update product into database.  8. System displays confirmation message or error message.  9. System redirects to product’s detail page. |

##### Delete Product

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Delete Product | |
| **Actor** | Admin | |
| **Description** | When admin wants to delete product from the database | |
| **Requirements** | Admin provides product’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: product is delete in the database | |
| Fail: product is not delete from the database | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks ‘Delete’ button. | 2. System redirects to ‘Product List’ page and display a list of existing product.  4. System delete product in the database  5. System displays confirmation or error. |

### BLOGS - BLOGS

#### Create new BLOGs

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Create BLOGs | |
| **Actor** | Admin | |
| **Description** | When admin wants to create a Blogs | |
| **Requirements** | Admin provides New content | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: Blogs content is added into the website | |
| Fail: admin can’t create blogs | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Blogs Tab on menu.  3. Actor clicks ‘Add New Content’ button.  5. Actor inputs information to create blogs | 2. System redirects to ‘Blogs’ page and display a list of Blogs.  4. System displays a form with the following fields:  - id  - name  - content  - image  6. System validates inputs.  7. System adds Blogs into website.  8. System displays confirmation message or error message. |

#### Update Blogs

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Update Blogs on the website | |
| **Actor** | Admin | |
| **Description** | When admin wants to update blogs information | |
| **Requirements** | Admin provides Blogs Content | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: Blogs is updated in the website | |
| Fail: admin can’t update Blogs | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Blogs Tab on menu.  3. Actor clicks ‘View Details’ button.  5. Actor chooses the criteria they want to update and input information. | 2. System redirects to ‘Blogs’ page and display a list of Blogs.  4. System displays a form with the following fields.  - name  - content  - image  6. System validates inputs.  7. System update Blogs Content to website.  8. System displays confirmation message or error message.  9. System redirects to Home. |

#### Delete Blogs

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Delete Blogs | |
| **Actor** | Admin | |
| **Description** | When admin wants to delete Blogs from the website | |
| **Requirements** | Admin provides Blogs Content | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: product is delete in the website | |
| Fail: product is not delete from the website | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Blogs Tab on menu.  3. Actor clicks ‘Delete’ button. | 2. System redirects to ‘Blogs List’ page and display a list of Blogs.  4. System delete Blogs in the website  5. System displays confirmation or error. |

### ORDER

#### View Order

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | View Order | |
| **Actor** | Admin | |
| **Description** | When admin wants to know the order of the product | |
| **Requirements** | Admin has at least 1 order on website. | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: product is shown in the website | |
| Fail: Page View Order cannot see from the website | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses View Order on menu.  3. Actor clicks ‘View’ button. | 2. System redirects to ‘Order List’ page and display a list of Order.  4. System show Order in the website.  5. System displays confirmation or error. |

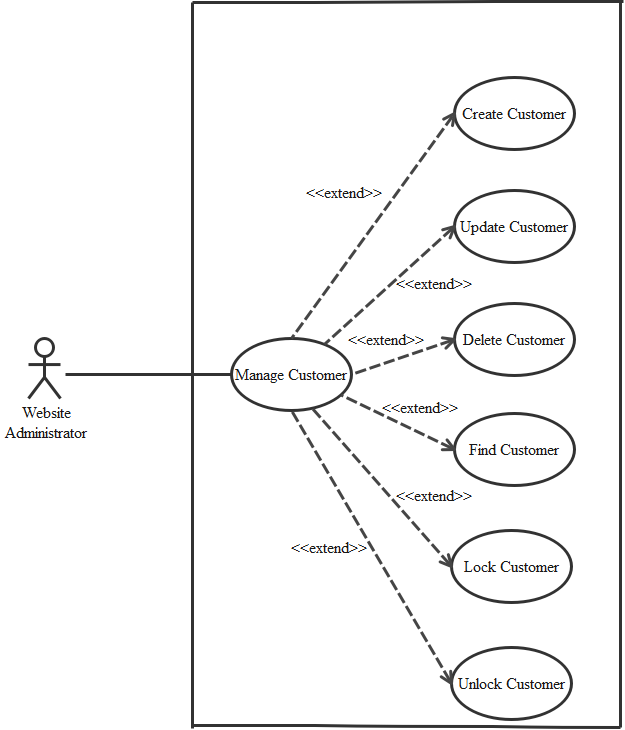
#### View Order Detail

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | View Order Detail | |
| **Actor** | Admin | |
| **Description** | When admin wants to see View Order Detail from the website | |
| **Requirements** | Admin has at least 1 order on website. | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: Admin can view Order Detail on the website | |
| Fail: Admin cant’ view the Order Detail on the website | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses View Order on menu.  3. Actor clicks ‘Detail’ button. | 2. System redirects to ‘Order List’ page and display a list of Order.  4. System show Order Details in the website  5. System displays confirmation or error. |

#### Update Order Status

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Update Order Status | |
| **Actor** | Admin | |
| **Description** | When admin wants to update Order Status from the website | |
| **Requirements** | Admin has at least 1 order on website. | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: Admin can update Order Status on the website | |
| Fail: Admin can’t update Order Status on the website | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses View Order Tab on menu.  3. Actor clicks ‘Delete’ button. | 2. System redirects to ‘Order List’ page and display a list of Order.  4. System update Order Status in the website  5. System displays confirmation or error. |

### WEBSITE ADMINISTRATOR



#### CREATE CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Create Customer | |
| **Actor** | Admin | |
| **Description** | When admin wants to create customer | |
| **Requirements** | Actor provides customer information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-conditions** | Success: new Customer is added into the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1. Actor chooses Customer Tab on menu.  3. Actor clicks ‘Add New Customer’ button.  5. Actor inputs information to create Customer. | 2. System redirects to ‘Customer’ page and display a list of existing Customers.  4. System displays a form with the following fields:  - id  - fullname  - username  - password  - address  - phone  - email  - birthday  - active  6. System validates inputs.  7. System adds new customer into database.  8. System displays confirmation message or error message. |

#### UPDATE CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Update Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to Update customer | |
| **Requirements** | Actor provides customer information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | Success: Customer is update into the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1. Actor chooses Customer Tab on menu.  3. Actor clicks on ‘View Details’ on a Customer  5. Actor chooses the criteria they want to update and input information. | 2. System redirects to ‘Customer’ page and display a list of existing Customers.  4. System displays a form with the following fields:  - fullname  - username  - password  - address  - phone  - email  - birthday  - active  6. System validates inputs.  7. System update customer into database.  8. System displays confirmation message or error message.  9. System redirects to customer’s detail page. |

#### DELETE CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Delete Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to delete customer | |
| **Requirements** | Actor provides customer information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | Success: Customer is delete from the database | |
| Fail: Customer is not deleted from the database | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1. Actor chooses Customer Tab on menu.  3. Actor clicks ‘Delete’ button. | 2. System redirects to ‘Customer’ page and display a list of existing Customers.  4. System delete customer in the database.  5. System displays confirmation message or error message. |

#### FIND CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Find Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to find customer by ID | |
| **Requirements** | Actor provides Customer’s id | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | Success: Show customer that matches ID | |
| Fail: No customer is shown | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1 Actor inputs ID.  3. Actor clicks customer’s title in list or view, which matches the ID. | 2. System show list of customer that match the ID. |

#### LOCK CUSTOMER

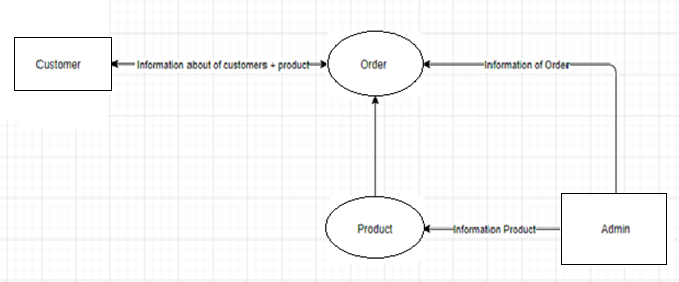
|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Lock Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to lock a customer | |
| **Requirements** | Actor provides Customer’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | Success: set customer’s status into “Lock” and updated in database | |
| Fail: No changes are made | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1. Actor chooses Customer row on User list and click “disable” button | 2. System updates Customer’s status into database.  3. System displays confirmation message or error message. |

#### UNLOCK CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Unlock Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to unlock a customer | |
| **Requirements** | Actor provides Customer’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | Success: set customer’s status into “Unlock” and updated in database | |
| Fail: No changes are made | |
| **Basic flow** | **Actor’s action** | **System’s responses:** |
| 1. Actor chooses Customer row on User list and click “enabled” button | 2. System updates Customer’s status into database.  3. System displays confirmation message or error message. |

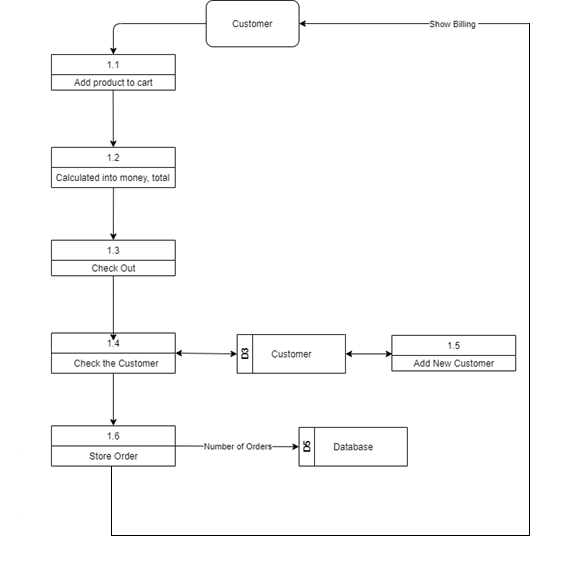
## DATA FLOW DIAGRAM (DFD)

### LEVEL 0

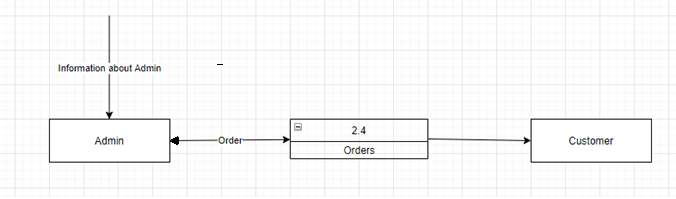


### LEVEL 1

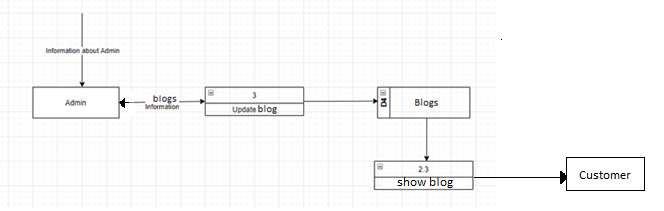
#### Function 1



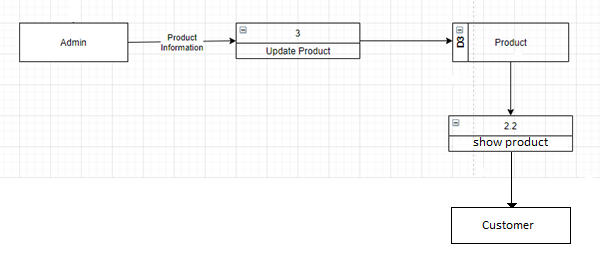
#### Function 2



#### Function 3

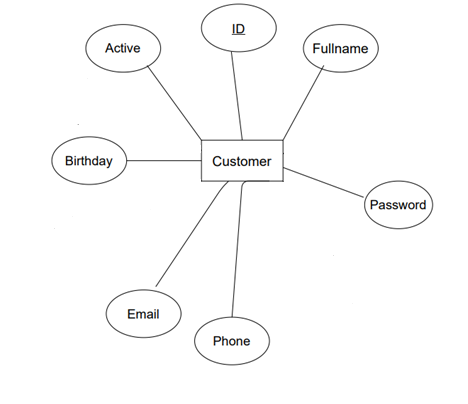


#### Function 4

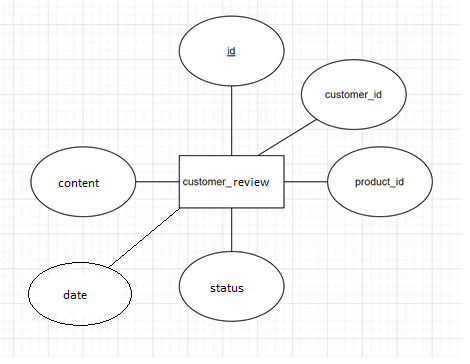


## ENTITY PROPERTIES

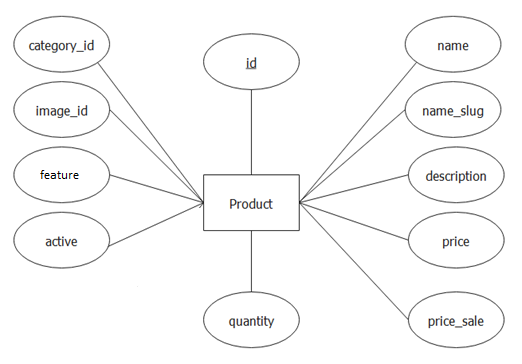
### Customer



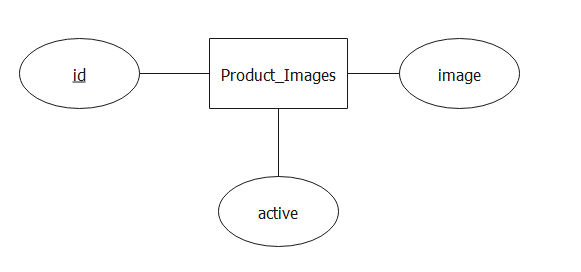
### Customer\_review



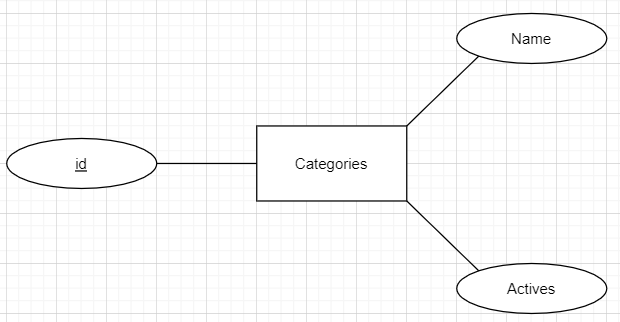
### Product



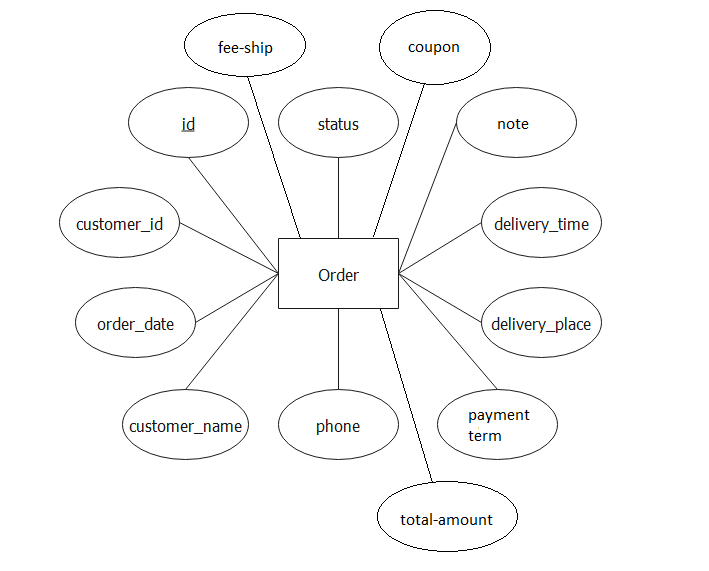
### Product\_Image

****

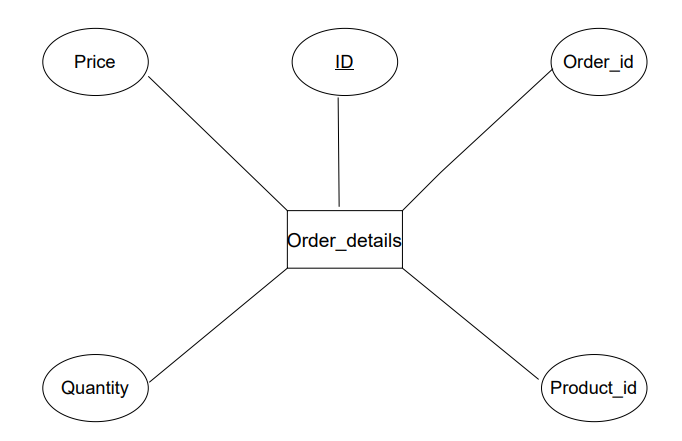
### Category



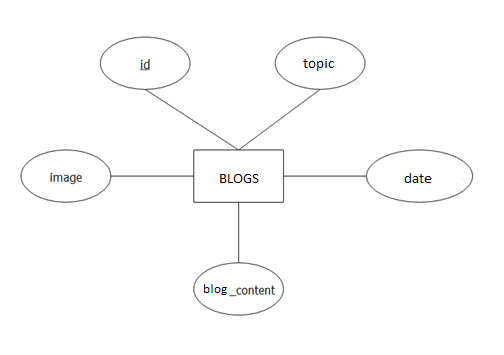
### Order



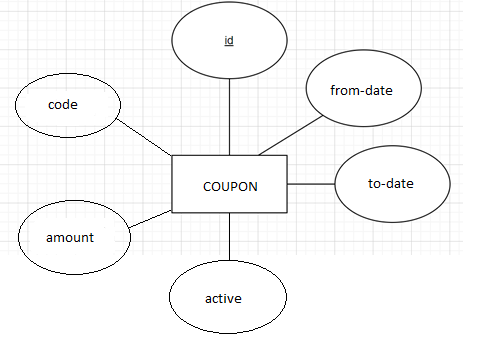
### Order\_Detail



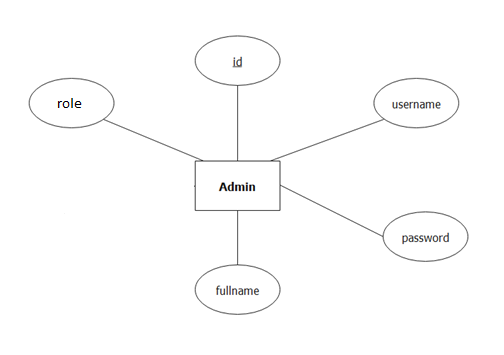
### Blogs



### Coupon

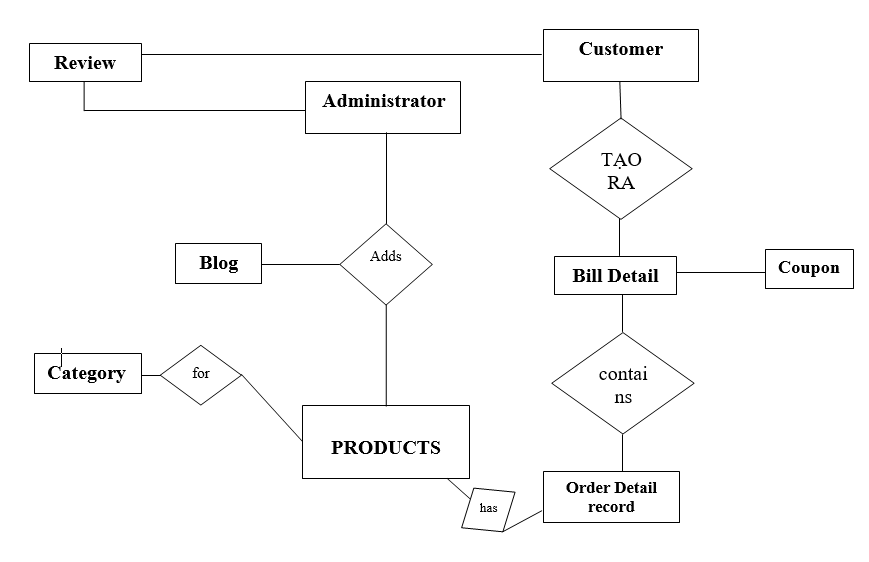


### Admin

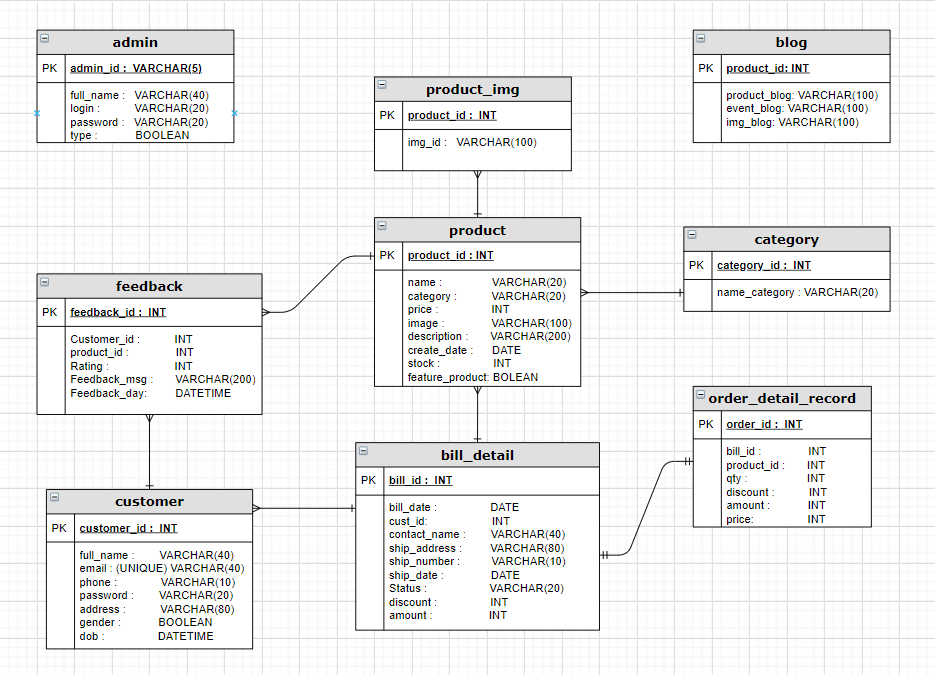


## ENTITY RELATIONSHIP DIAGRAM (ERD)

### LOGICAL ERD



### PHYSICAL ERD



## TABLE DESIGN

### Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: Customer | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | Id | INT  IDENTITY(100,1) | NOT NULL  PRIMARY KEY | Store the ID of the customer |
| 2 | fullname | VARCHAR(225) | NOT NULL | Store the full name of the customer |
| 3 | password | VARCHAR(225) | NOT NULL | Store the password of the customer |
| 4 | phone | VARCHAR(225) | NOT NULL | Store the Phone of the customer |
| 5 | email | VARCHAR(225) | NOT NULL | Store the Email of the customer |
| 6 | birthday | DATE | NOT NULL | Store the Birthday of the customer |
| 7 | active | BOOLEAN | NOT NULL | Store the Status the of the customer |

### review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: Review | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | Id | INT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of customer comment |
| 2 | customer\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of customer |
| 3 | product\_id | VARCHAR(100) | NOT NULL  FOREIGN KEY | Store the ID of product |
| 4 | status | VARCHAR(100) | NOT NULL | Store status of customer |
| 5 | content | VARCHAR(100) | NOT NULL | Store content of customer |
| 6 | date | DATE | NOT NULL | Store datetime of status |

### Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Products** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | INT(20) IDENTITY(100,1) | NOT NULL  PRIMARY KEY | Store the ID of the product. |
| 2 | category\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of the category |
| 3 | image\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of the image. |
| 4 | name | VARCHAR(225) | NOT NULL | Stores the name of product |
| 5 | name\_slug | VARCHAR(225) | NOT NULL | Stores the name\_slug of product. |
| 6 | desciption | TEXT | NOT NULL | Stores the desciption of product |
| 7 | price | INT | NOT NULL | Stores the price of product. |
| 8 | price\_sale | INT | NOT NULL | Stores the price promotion of product. |
| 9 | feature | BOLEAN | NOT NULL | Stores the feature product. |
| 10 | quantity | INT | NOT NULL | Stores the quantity of product. |
| 11 | detail | TEXT | NOT NULL | Stores the detail of product |
| 12 | active | BOOLEAN | NOT NULL | Stores the active of product |

### Product\_Image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Products\_Images** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of the product\_image |
| 2 | product\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of the product |
| 3 | image | VARCHAR(225) | NOT NULL | Store the Image of the sale |
| 4 | active | BOOLEAN | NOT NULL | Store the Active of the Cart |

### Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Categories** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | INT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of the product\_image |
| 2 | name | VARCHAR(225) | NOT NULL | Store the name of the sale |
| 3 | active | BOOLEAN | NOT NULL | Store the Active of the Cart |

### Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: Orders | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of the order. |
| 2 | Customer\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of customer. |
| 3 | customer\_name | VARCHAR(100) | NOT NULL | Store customer name. |
| 4 | order\_date | DATE | NOT NULL | Store order date. |
| 5 | phone | INT | NOT NULL | Store phone number of receiver. |
| 6 | email | VARCHAR(100) | NOT NULL | Store email of receiver |
| 7 | delivery\_address | VARCHAR(225) | NOT NULL | Store delivery address of receiver. |
| 8 | status | BOOLEAN | NOT NULL | Store status of order |
| 9 | delivery\_time | VARCHAR(100) | NOT NULL | Stores the delivery time of product. |
| 10 | Fee-ship | INT | NOT NULL | Store the fee-ship of receiver. |
| 11 | Coupon | VARCHAR(10) | NOT NULL | Store the coupon of order. |
| 12 | Total\_amount | INT | NOT NULL | Store the TOTAL of order. |
| 13 | Payment-term | VARCHAR(100) | NOT NULL | Store the payment-term of customer. |
| 14 | Note | VARCHAR(100) | NOT NULL | Store the note of customer. |

### Order\_Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Order\_details** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of the order details |
| 2 | order\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Store the ID of the order |
| 3 | product\_id | BIGINT(20) | NOT NULL  FOREIGN KEY | Stores the ID of the product |
| 4 | quantity | INT | NOT NULL | Stores the quantity of product |
| 5 | Unit price | INT | NOT NULL | Stores the unit price of product |
| 6 | Discount | INT | NOT NULL | Stores the discount of product |
| 7 | amount | INT | NOT NULL | Stores the amount of order |
| 8 | Customer\_id | INT | NOT NULL | Stores the ID of customer |

### Blogs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Blogs** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | INT(20) IDENTITY(1,1) | NOT NULL PRIMARY KEY | Stores the ID of Blogs |
| 2 | topic | VARCHAR(255) | NOT NULL | Stores the topic of Blogs |
| 3 | date | DATETIME | NOT NULL | Stores the date of Blogs |
| 4 | content | VARCHAR(255) | NOT NULL | Stores the Content of Blogs |
| 5 | image | VARCHAR(255) | NOT NULL | Stores the Image of Blogs |

### Coupon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Coupon** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL PRIMARY KEY | Stores the ID of Coupon |
| 2 | code | VARCHAR(255) | NOT NULL | Stores the code of Coupon |
| 3 | active | BIT(1) | NOT NULL | Stores the Status of Coupon |
| 4 | From-date | DATETIME | NOT NULL | Stores the start time of coupon |
| 4 | To-date | DATETIME | NOT NULL | Stores the expiration date of Coupon |

### Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: admin** | | | | |
| No. | Column Name | Data Type | Constraint | Description |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL PRIMARY KEY | Stores the ID of admin |
| 2 | username | VARCHAR(255) | NOT NULL | Stores the Username of Admin |
| 3 | password | VARCHAR(255) | NOT NULL | Stores the Password of Admin |
| 4 | fullname | VARCHAR(255) | NOT NULL | Stores the Fullname of Admin |
| 5 | role | INT | NOT NULL | Stores the ROLE of Admin |

## TASK SHEET REVIEW 2

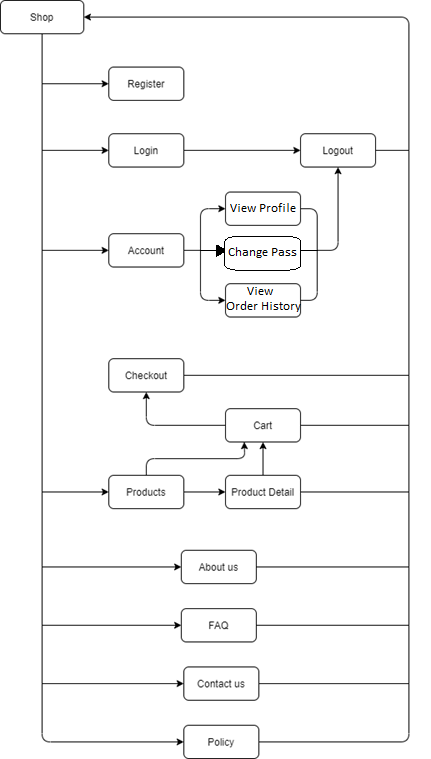
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Ref. No.: Saigon Bakery Store | | Project Name | Activity Plan Prepared By | Date of Preparation of Activity Plan: | | | |
| Sr./No. | Task | Actual Start Date | Actual Days | Team member name | Status |
| 1 | Use Case | Saigon Bakery | Phan Phuc Thinh | 23-June-21 | 3 | All Member | Completed |
| 2 | Data Flow Diagram (DFD) | 26-June-21 | 2 | All Member | Completed |
| 3 | Entity Relationship Diagram (ERD) | 28-June-21 | 3 | All Member | Completed |
| 4 | Entity Properties | 30-June-21 | 2 | All Member | Completed |
| 5 | Table Design | 1-July-21 | 1 | All Member | Completed |

|  |  |
| --- | --- |
| Date: 23 June 2021 | |
| Signature of Instructor:  Mrs. Le Mong Thuy | Signature of Team Leader:  Phan Phuc Thinh |

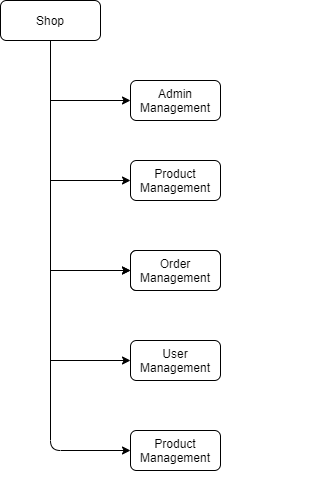
# REVIEW 3

## SITE MAP

### Customer’s Site Map

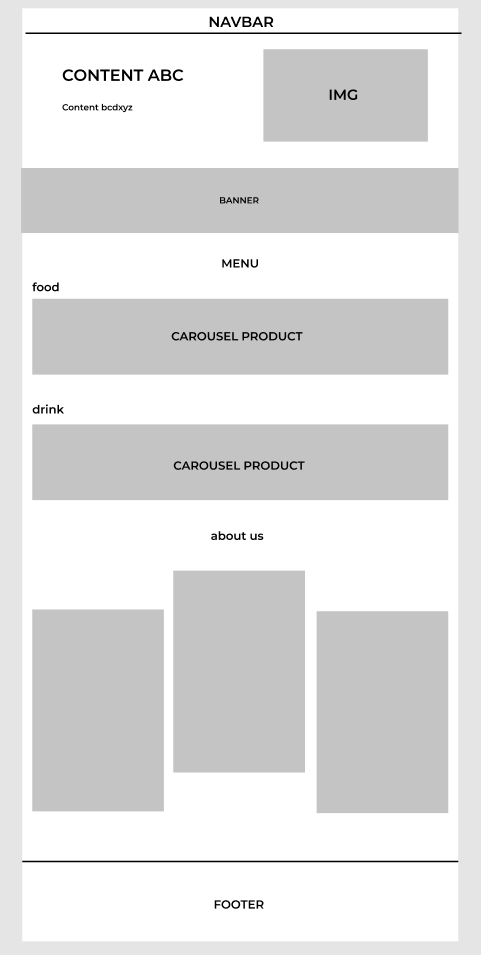


### Admin’s Site Map



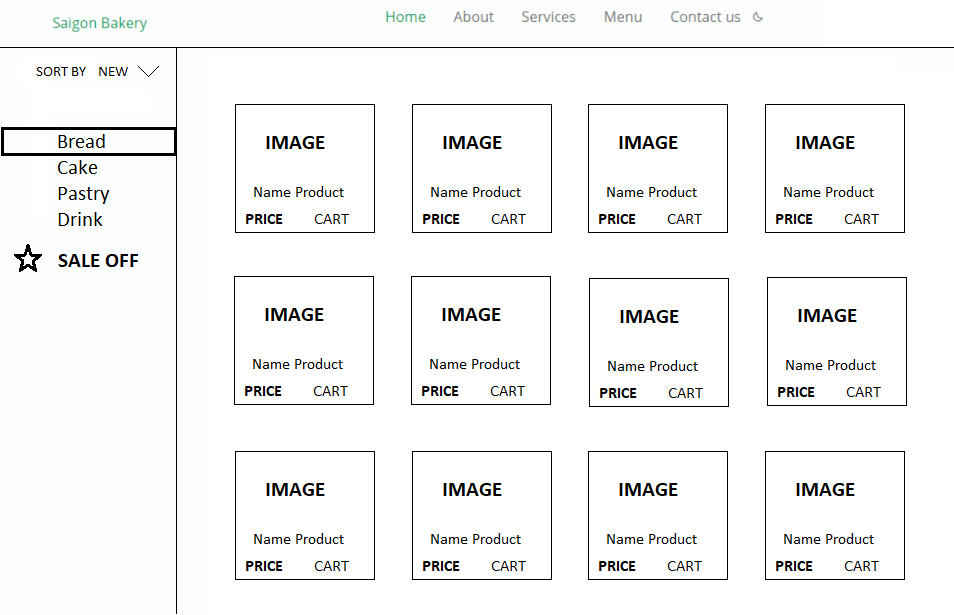
## Saigon bakery DESIGN template

### Homepage



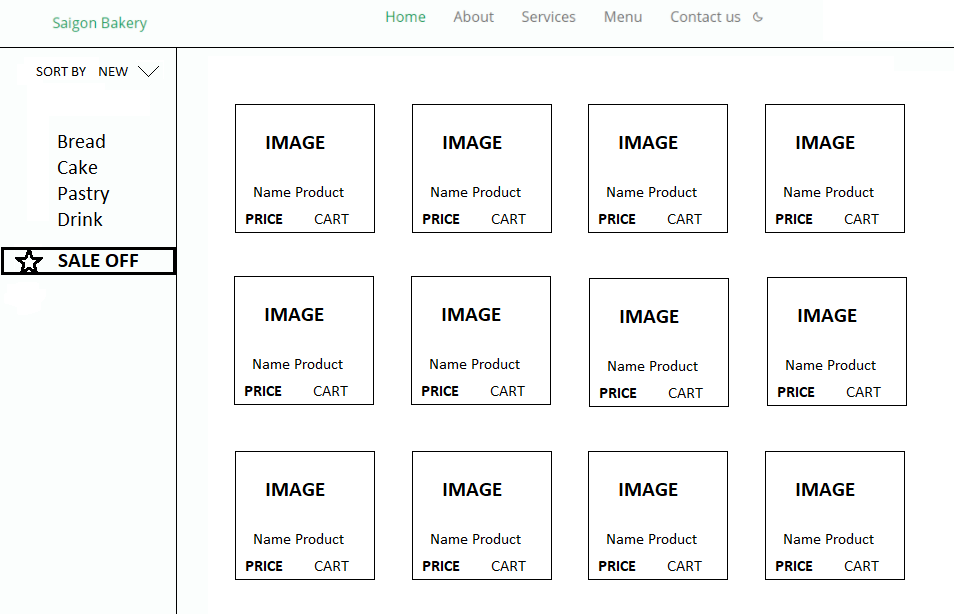
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
|  | Special offer today | Link |  | Click | Go to product sales off page. |
|  | View cart | Link |  | Click | View product in cart, its price and total price. |
|  | Login/Sign up | Button |  | Click | Log in to existing account or register new account. |
|  | Contact us | Link |  | Click | Go to contact us. |
|  | Logo | Link |  | Click | Go to homepage. |
|  | Navigation bar (Blogs) | Link |  | Click | Go to blogs page. |
|  | Navigation bar (About us) | Link |  | Click | Go to about us page. |
|  | Hotline (phone and email) | Text |  | Click | Show phone of website and go to send email. |
|  | Side bar  (category:   * Bread * Pastry * Cake * Beverage | Link |  | Click | Show category of products. |
|  | New product | Link |  | Click | Show new product and is image. |
|  | Hot offer | Link |  | Click | Show hot offer and is image. |
|  | Add to cart | Button |  | Click | Add to cart, view pop up cart. |
|  | Top product | Link |  | Click | Show top product and is image. |
|  | Subscription | Button |  | Click | Allow customer to subscript the website. |
|  | Footer (information) | Link |  | Click | Go to information for Saigon Bakery. |
|  | Footer (product information) | Link |  | Click | Go to Question and Answer page, Privacy page, Term of use page. |
|  | Footer(what in store) | Link |  | Click | Go to category in the store. |
|  | Footer(Blogs) | Link |  | Click | Show blogs of products. |

### Product List



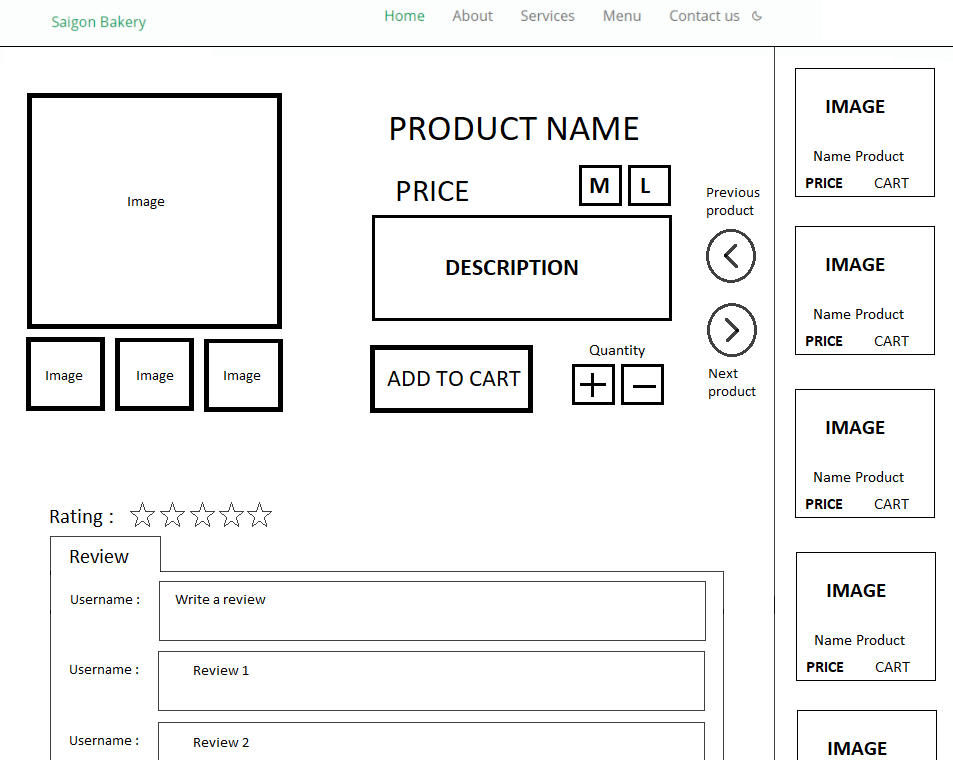
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Category name | Text |  |  | Show name of category. |
| 2 | Image | Text |  |  | Show image of products. |
| 3 | Sort list dropdown | Button |  | Click | View product by selection. |
| 4 | Cart | Button |  | Click | Link Add to Cart |

### Sale off product



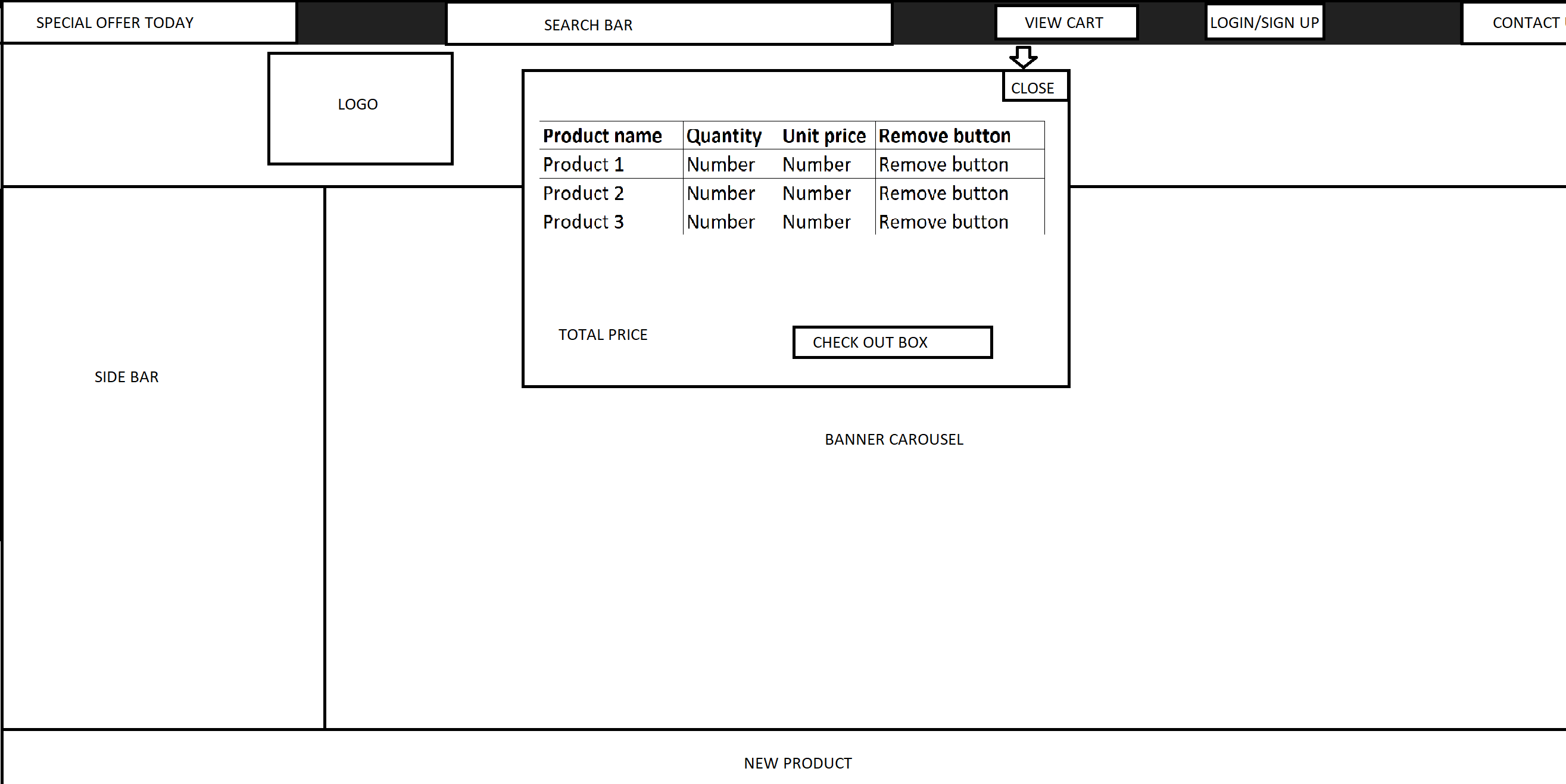
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Sales off products | Text |  |  | Show name of sales products. |
| 2 | Image | Text |  |  | Show image of products. |
| 3 | Sort list dropdown | Button |  | Click | View product by selection. |
| 4 | Cart | Button |  | Click | Link Add to Cart |

### Product detail



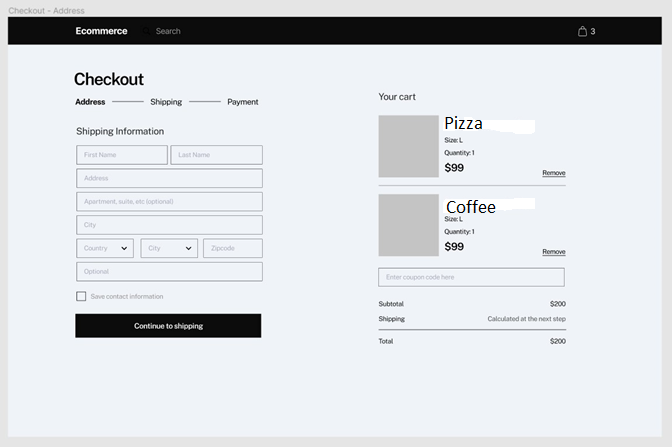
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Product name | Text |  |  | Show name of product. |
| 2 | Image | Text |  |  | Show image of named product. |
| 3 | Rating | Image |  | Click | View rate of product. |
| 4 | Description | Text |  |  | Show description of product. |
| 5 | Related product | Image |  |  | Show relative product. |
| 6 | Change page | Link |  | Click button left and right | Go to next or previous page. |

### View cart



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Close | Link |  | Click | Go back to previous page. |
| 2 | Table (Name) | Text |  |  | Show name of selected products. |
| 3 | Table (Quantity) | Number |  |  | Show quantity of selected products. |
| 4 | Table (Price) | Number |  |  | Show unit price of selected products. |
| 5 | Table (remove button) | Button |  | Click | Remove selected product. |
| 6 | Total price | Number |  |  | Show total price of selected products after plus quantity. |
| 7 | Check out | Button |  | Click | Go to check out page. |

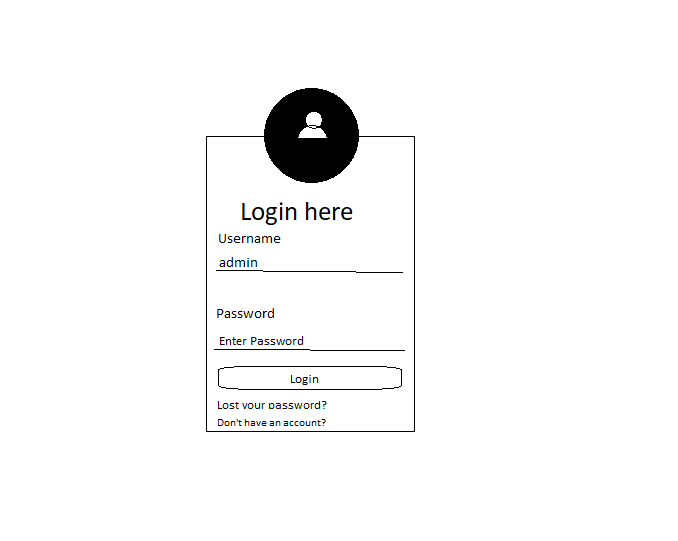
### Check out

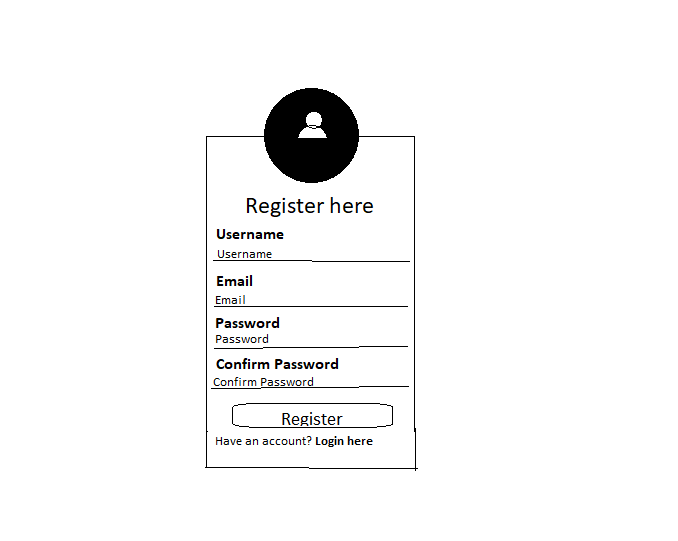


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Check out | Link |  |  | Check out process. |
| 2 | Count of number product in cart | Text |  |  | Show how many product types in cart. |
| 3 | Table (No) | Number |  |  | Numerical order of product. |
| 4 | Table (Product Image) | Image |  |  | Show product image. |
| 5 | Table (Quantity) | Number |  |  | Show quantity of product. |
| 6 | Table (Unit price) | Number |  |  | Show unit price of product. |
| 7 | Table (Remove button) | Button |  | Click | Remove selected product. |
| 8 | Summary table | Text |  |  | Summary price of product. |
| 9 | Delivery detail  (- “Customer name” text field.  - “Phone” text field.  - “Address” text field.  - “Province” select box.  - “Delivery time frame” select box.  ) | Text |  |  | Show table to fill delivery detail and contact of receiver. |
| 10 | Payment method | Text |  | Click | Show payment method. |

### Customer

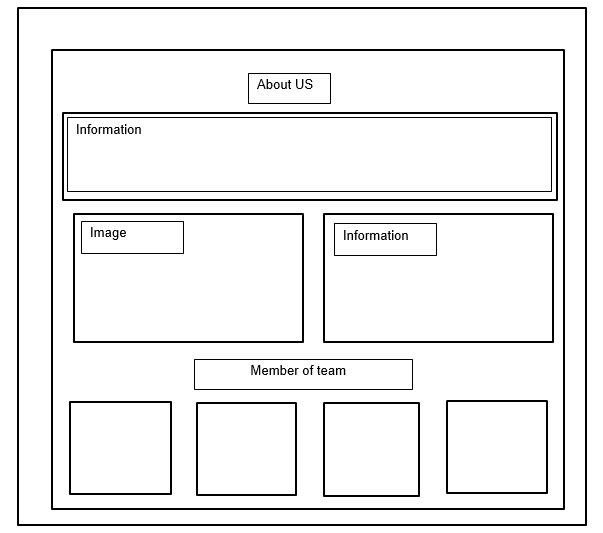
#### Customer Login & Register



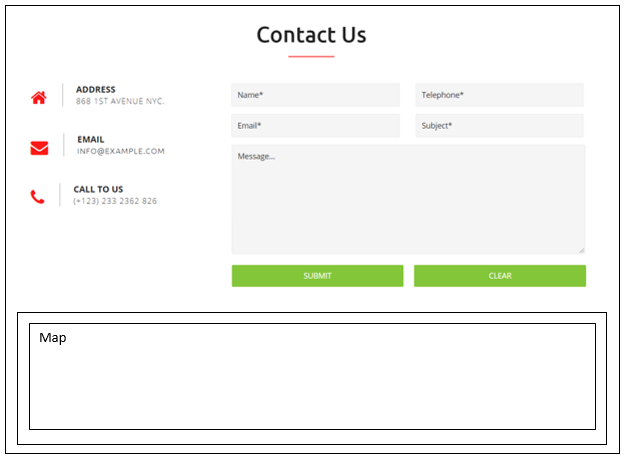


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Username | Null | Not null, not number |  | Customer/Admin’s name |
| 2 | Password | Null | Not null, not number |  | Customer/Admin’s password |
| 3 | Email Address | Null | Not null, must be email (have .com and @) |  | Customer/Admin’s email |
| 4 | Phone Number | Null | Not null, must be phone number |  | Customer’s phone |
| 5 | Fullname | Null | Not null, not number |  | Customer’s full name |
| 6 | Address | Null | Not null |  | Customer’s address |
| 7 | Birthday | Null | Not null, must be date |  | Customer’s birthday |
| 8 | Login | Button |  | Click | Link to index page |
| 9 | Register | Button |  | Click | Link to index page |
| 10 | Lost your password | Button |  | Click | Link to Contact US |
| 11 | Don’t have an account? | Button |  | Click | Link to Register |
| 12 | Have an account? | Button |  | Click | Link to Login |
|  | | | | | |

#### About US

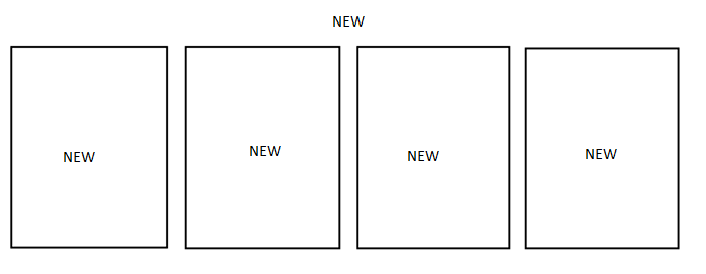


#### Contact US

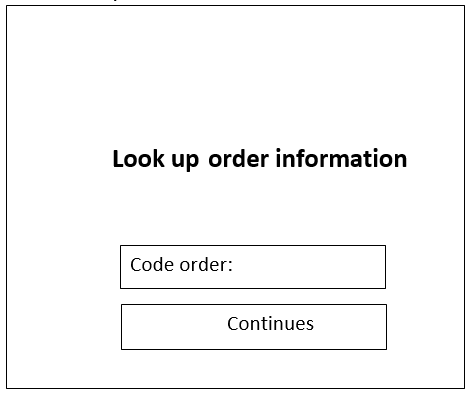


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Name | Null | Not null, not number |  | Customer’s name |
| 2 | Email Address | Null | Not null, must be email (have .com and @) |  | Customer’s email |
| 3 | Phone Number | Null | Not null, must be phone number |  | Customer’s phone |
| 4 | Message | Null | Not null, not number |  | Customer’s message |
| 5 | Submit | Button |  | Click | Send mail |
| 6 | Clear | Button |  | Click | Clear text from form |
|  | | | | | |

#### BLOGS

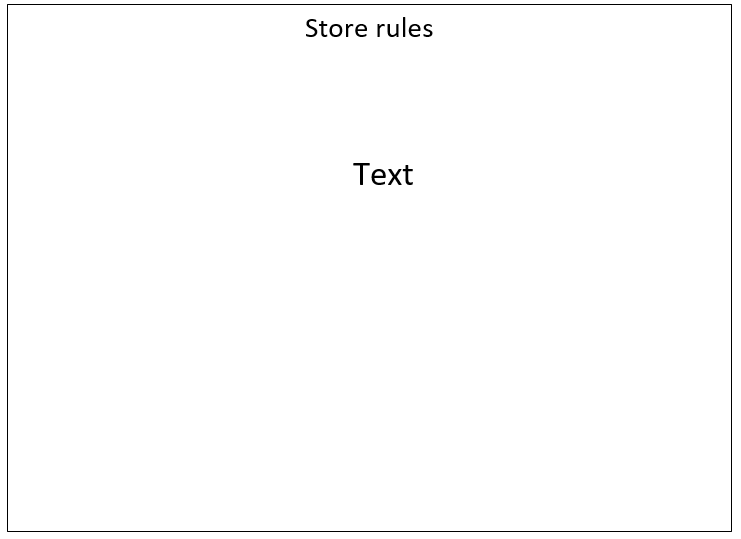


#### Purchase history

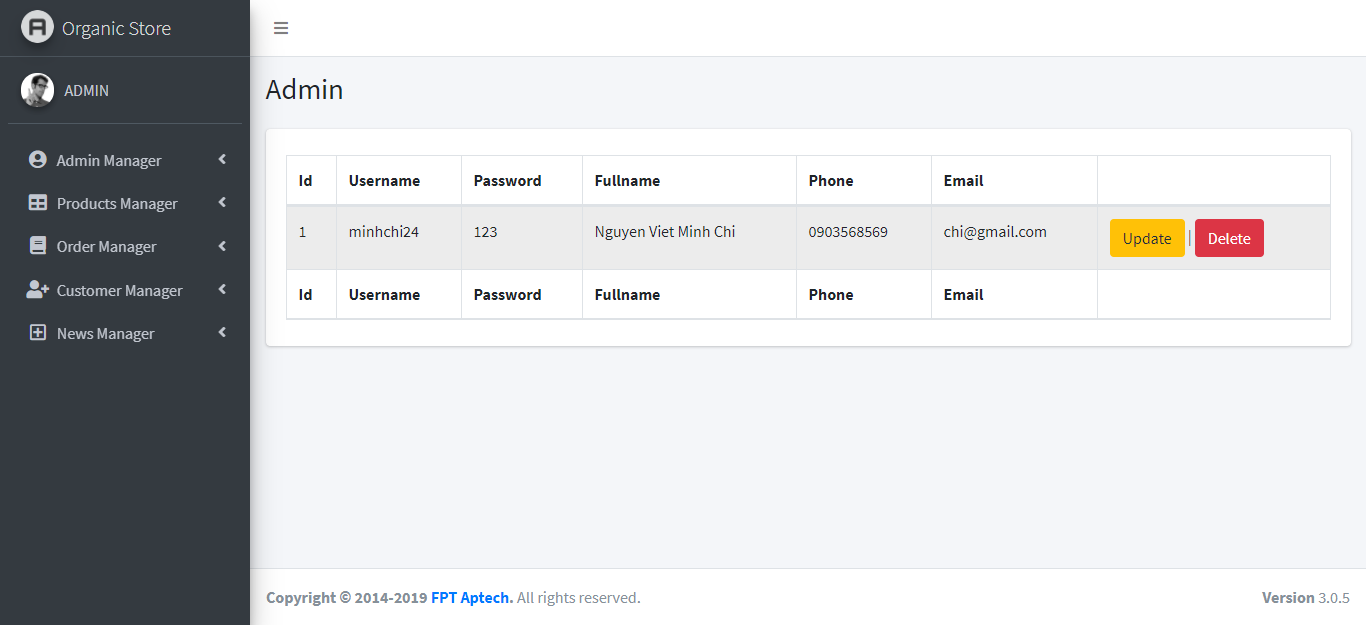


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** |
| 1 | Look up order information | NULL | Not null must be code order. |  | Code order. |
|  | | | | | |

#### Store rules

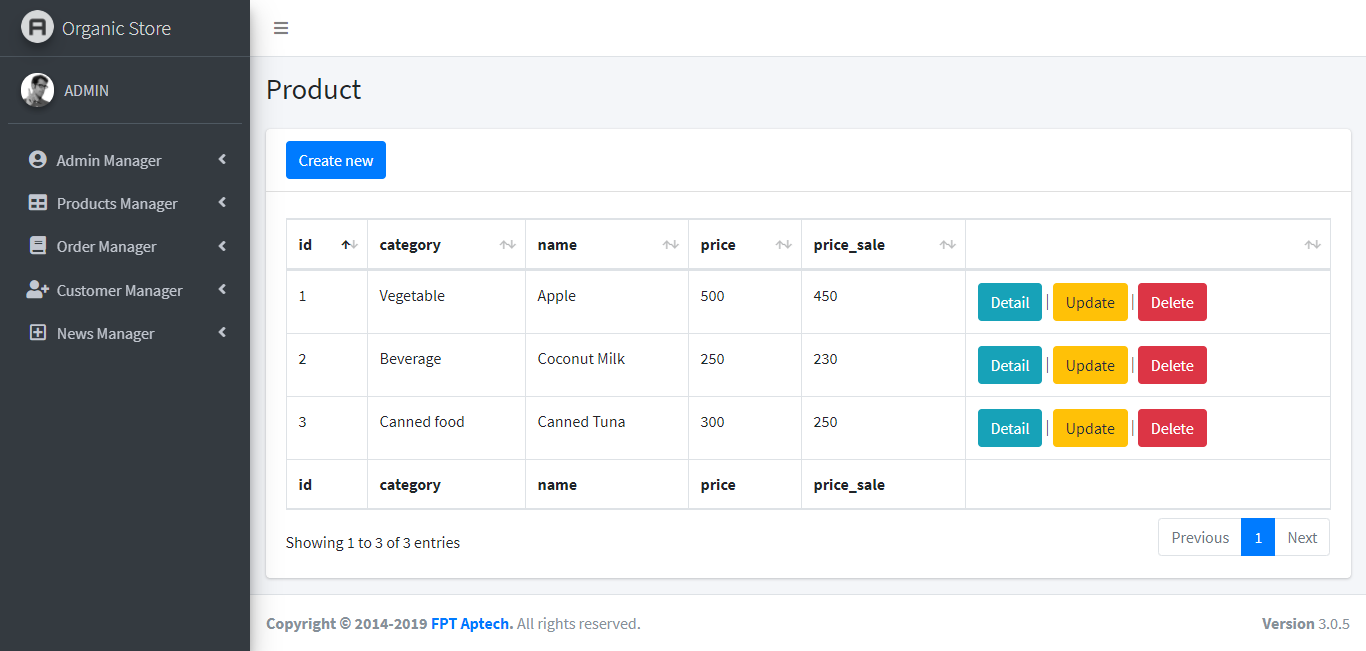


#### Admin manager

****

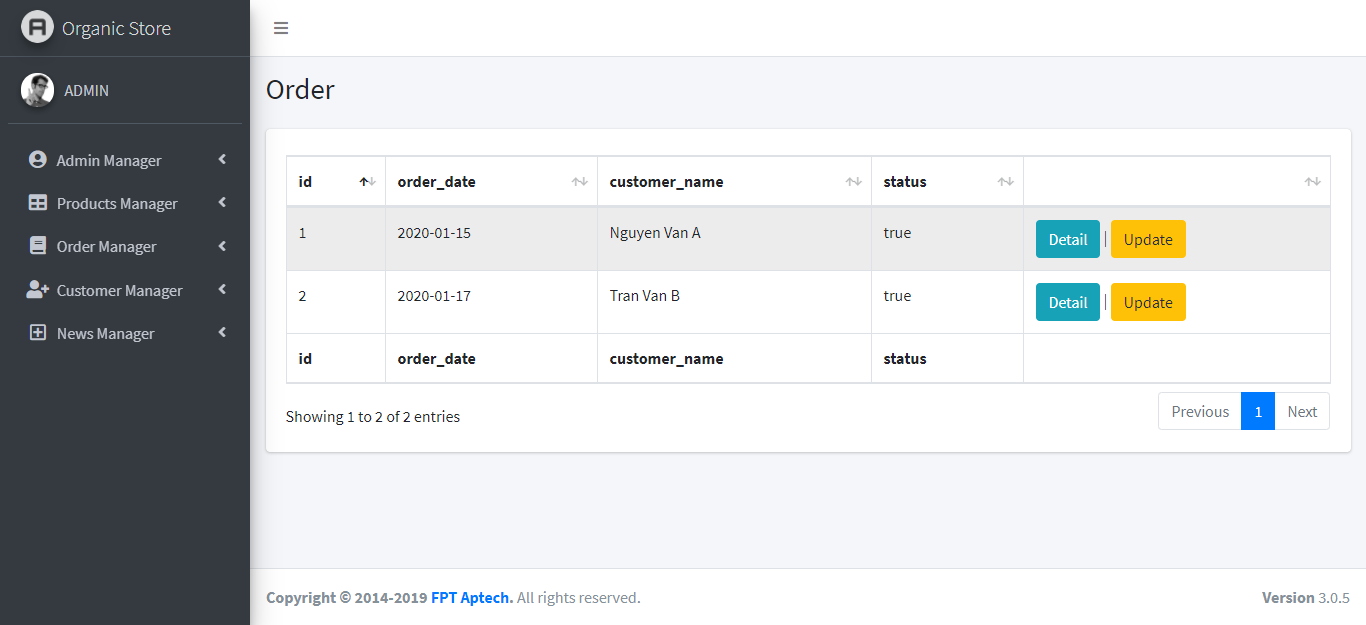
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Admin’s id. |
| 2 | username | Not null, force matches in the database. |  | Admin’s account name. |
| 3 | password | Not null, force matches in the database. |  | Admin’s password. |
| 4 | fullname | Not null, not number. |  | Admin’s name. |
| 5 | phone | Not null, must be phone number. |  | Admin’s phone. |
| 6 | email | Not null, must be email (have .com and @). |  | Admin’s email. |
| 7 | update |  | Click | Link to update admin form. |
| 8 | delete |  | Click | Link to delete admin form. |

#### Product manager

****

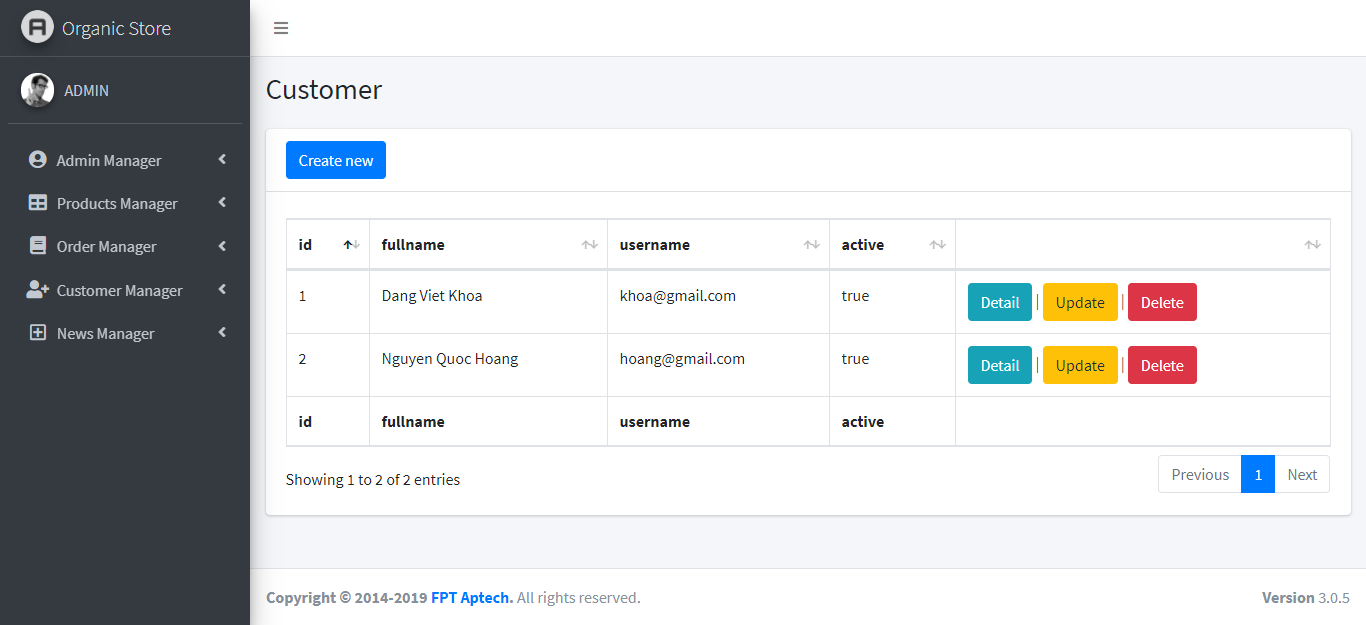
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Product’s id. |
| 2 | category | Not null, force matches in the database. |  | Product’s category. |
| 3 | name | Not null, force matches in the database. |  | Product’s name. |
| 4 | price | Not null, must be number. |  | Product’s price. |
| 5 | price\_sale | Not null, must be number. |  | Product’s price sale off. |
| 6 | Create new |  | Click | Link to create product form. |
| 7 | Detail |  | Click | Link to detail product form. |
| 8 | Update |  | Click | Link to update product form. |
| 9 | Delete |  | Click | Link to delete product form. |

#### Order manageMENT

****

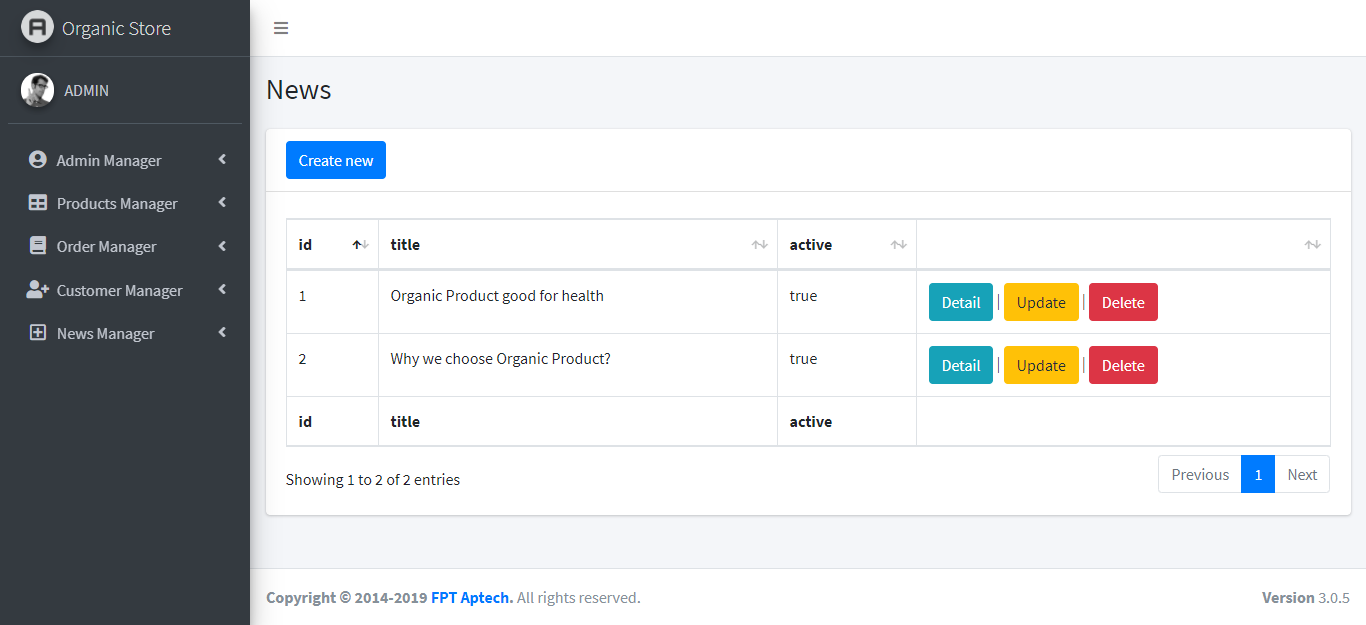
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Order’s id. |
| 2 | order\_date | Not null, must be date. |  | Order’s date. |
| 3 | customer\_name | Not null, not number. |  | Customer’s name. |
| 4 | status | Not null, force matches in the database. |  | Order’s status. |
| 5 | Detail |  | Click | Link to detail order form. |
| 6 | Update |  | Click | Link to update order form. |

#### Customer manager

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Customer’s id. |
| 2 | fullname | Not null, not number. |  | Customer’s name. |
| 3 | username | Not null, force matches in the database. |  | Customer’s account name. |
| 4 | active | Not null, force matches in the database. |  | Customer’s status. |
| 5 | Create new |  | Click | Link to create customer form. |
| 6 | Detail |  | Click | Link to detail customer form. |
| 7 | Update |  | Click | Link to update customer form. |
| 8 | Delete |  | Click | Link to delete customer form. |

#### Blogs manageMENT

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Blogs’s id. |
| 2 | title | Not null, force matches in the database. |  | Blogs’s title. |
| 3 | active | Not null, force matches in the database. |  | Blogs’s status. |
| 4 | Create new |  | Click | Link to create blogs form. |
| 5 | Detail |  | Click | Link to detail blogs form. |
| 6 | Update |  | Click | Link to update blogs form. |
| 7 | Delete |  | Click | Link to delete blogs form. |

## task sheet review 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.: Saigon Bakery Store** | | **Project Name** | **Activity Plan Prepared By** | **Date of Preparation of Activity Plan:** | | | |
| **Sr./No.** | **Task** | **Actual Start Date** | **Actual Days** | **Team member name** | **Status** |
| 1 | Admin page | SaiGon Bakery | Thinh | 30-Jun-21 | 30 | Tan | Completed |
| 2 | Home page | 30-Jun-21 | 30 | Thinh | Completed |
| 3 | Product page | 30-Jun-21 | 30 | Quang | Completed |
| 4 | Promotion page | 30-Jun-21 | 30 | Quang | Completed |
| 5 | Login page | 30-Jun-21 | 30 | Tan | Completed |
| 6 | Register page | 30-Jun-21 | 30 | Tan | Completed |
| 7 | Cart page | 30-Jun-21 | 30 | Thinh | Completed |
| 8 | Checkout page | 30-Jun-21 | 30 | Thinh | Completed |
| 9 | Blogs page | 30-Jun-21 | 30 |  |  |
| 10 |  | 30-Jun-21 | 30 |  |  |
| 11 |  | 30-Jun-21 | 30 |  |  |
| 12 |  | 30-Jun-21 | 30 |  |  |
| 14 |  | 30-Jun-21 | 30 |  |  |

|  |  |
| --- | --- |
| **Date: 25 June 2021** | |
| Signature of Instructor:  **Mrs. Le Mong Thuy** | Signature of Team Leader:  **Phan Phuc Thinh** |