**FPT ACADEMY INTERNATIONAL**



**FPT – APTECH COMPUTER EDUCATION**

Organic Product Store

|  |  |  |  |
| --- | --- | --- | --- |
| **Center Name:** | FPT – Aptech – HCM2 | | |
|  | Address:590 Cach Mang Thang Tam,  District 3, Ho Chi Minh City | | |
| **Instructor:** | Mr. Hoang Duc Quang | | |
|  |  | | |
| **Class:** | T1.2002.E0 | | |
| **Group:** | 02 | | |
| **Group members** | **Name**  Nguyen Viet Minh Chi | | **Student ID**  Student1251715 | |
|  | Dao Van Ngoc Hanh | | Student1251713 | |
|  | Nguyen Thi Thanh Mai  Nguyen Tien Dat | | Student1266513  Student1225631 | |
|  |  | |  | |
| **Project Duration** | Start date | 15th June 2020 | | |
|  | End date | 17th July 2020 | | |

**This is to certify that**

#### **Mr./Ms.:**

Nguyen Viet Minh Chi

Dao Van Ngoc Hanh

Nguyen Thi Thanh Mai

Nguyen Tien Dat

**has successfully designed & developed**

eProject: Organic Product Store

**Submitted by:**

Nguyen Viet Minh Chi

**Date of issue:** June 15th 2020

**Authorized Signature:**

TABLE OF CONTENTS

[ACKNOWLEDGE 4](#_Toc45200937)

[SYNOPSIS 4](#_Toc45200939)

[PROBLEM DEFINITION 4](#_Toc45200943)

[CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS) 5](#_Toc45200944)

[SCOPE OF THE WORK (IN BRIEF) 7](#_Toc45200948)

[ARCHITECTURE AND DESIGN OF THE SYSTEM 8](#_Toc45200949)

[DIAGRAM OF THE WEBSITE 9](#_Toc45200950)

[TASK SHEET REVIEW 1 10](#_Toc45200951)

[SITE MAP 11](#_Toc45200952)

[MOCK OF THE WEBSITE 12](#_Toc45200953)

[TASK SHEET REVIEW 2 24](#_Toc45200954)

[WEBSITE DESCRIPTION 25](#_Toc45200955)

[JAVASCRIPT DESCRIPTION 43](#_Toc45200956)

[TASK SHEET REVIEW 3 48](#_Toc45200957)

**REVIEW 1**

# ACKNOWLEDGE

Our group would like to devote this chapter to express our deepest appreciation to all those who offered us valuable assistance during in carrying out this eProject. First and foremost, we owe a debt of gratitude to our dedicated instructor, Mr. Hoang Duc Quang, whose suggestions and encouragement helped us to overcome not only technological challenges but also psychological tension. Without him, our work could not have been completed successfully. Secondly, we highly appreciate the eProject Team at the Head Office which has provided us a golden 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. Besides, 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, 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

# During the period of information technology and computer development people can search and find anything at home via internet connection. The sale and purchase transaction is completed electronically and interactively in real- time. Therefore, we are developing a website named Organic Product attract more potential customers.

# Organic Product is a group of people who wants to develop an online site where the user who is too busy to go of where they can buy fresh product. It will provide its user the following features.

# For Organic Product’s customer:

# Customer can register account.

# Customer can login and manage their account.

# Customer can search any vegetables, food, canned food…

# Customer can search products by price.

# Customer can see product introduction before buying.

# Customer can buy products after add products to cart.

# Customer can view cart history.

# Customer can check out.

# PROBLEM DEFINITION

After reading the project specifications of Organic Product Store, our group has discussed about what should to be done to fulfil their 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

1. A home page will show about main the products, also design a suitable logo and include it on homepage, display the logo on all the pages in the site.
2. A menu which will include all the functions implemented in the website.
3. 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

1. A login page will grant access to Organic Product’s admin.
2. Admin can view, add, update, delete products.
3. Admin can view, add, update, delete, find, lock, unlock customer account.
4. Admin can manage news and orders.

# CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS)

## **CLIENT:** Organic Product Store

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

The project is to design an online website for selling organic product, for instant: vegetabe, fresh food, canned food, cereals, etc.

**Input to the system:**

## **Organic Product store’s administrators need to provide:**

* Information required in order to create account to access to the admin system.
* Product’s are going to be shown on the website.

## **Organic Product store’s customers need to provide:**

* Personal information to buy products.

**Output from the system:**

## **Organic Product 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) news.
* Manage (view, update) products, order.

## **Organic Product store’s guide will be able to:**

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

## **Organic Product 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, news, 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 | Internet Information Service |
| 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 framewok.
* jQuery.
* Bootstrap.
* HTML/CSS
* Bootstrap.

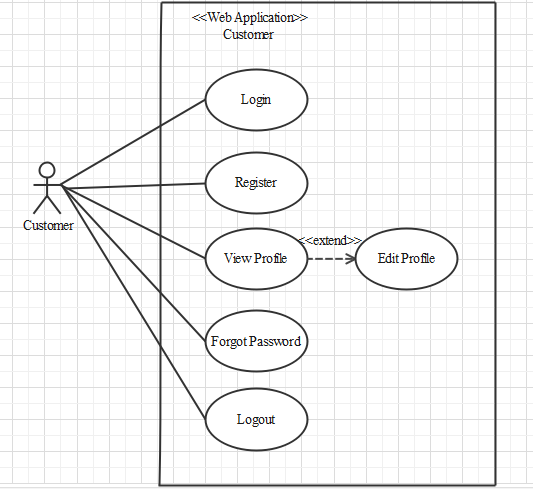
# TASK SHEET REVIEW 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.: Organic Product 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 | Organic Product Store | Chi | 15-Jun-20 | 1 | Chi | Completed |
| 2 | Introduction |  | 1 | Chi | Completed |
| 3 | Problem Definition |  | 1 | Hạnh | Completed |
| 4 | Customer Requirement Specification (CRS) |  | 2 | Mai | Completed |
| 5 | Scope of Work |  | 2 | Đạt | Completed |
| 6 | Hardware/Software Requirement |  | 2 | Chi | Completed |
| 7 | Task Sheet |  |  | Chi | Completed |

|  |  |
| --- | --- |
| **Date: 17 Jun 2020** | |
| Signature of Instructor:  **Mr. Hoang Duc Quang** | Signature of Team Leader:  **Nguyen Viet Minh Chi** |

**REVIEW 2**

**1. 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 which does not exist in database. | 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 can not 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  - “Gender” text field |

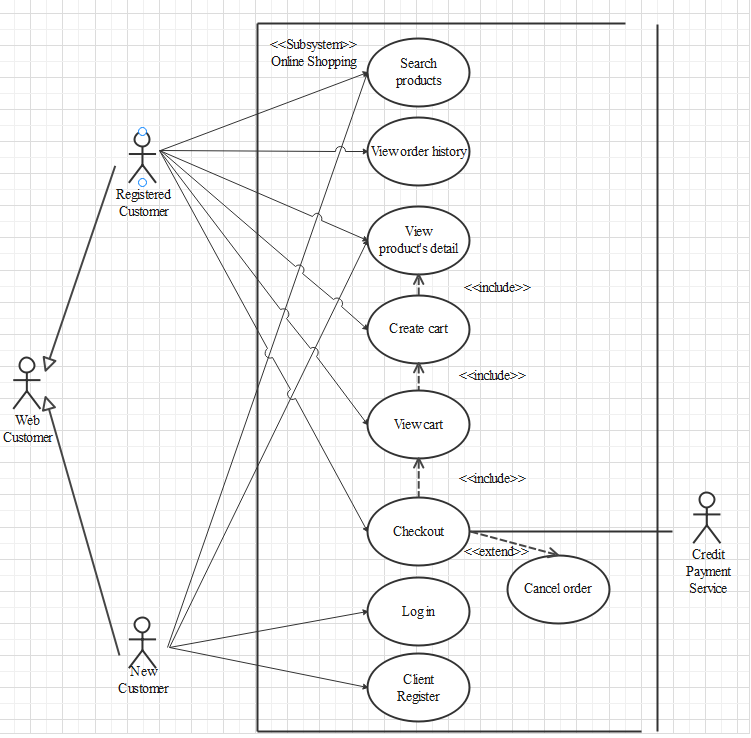
**Edit profile:**

|  |  |  |
| --- | --- | --- |
| **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  - “Gender” 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 can not 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 product\_id |
| **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 “Go back to shopping”  3. Actor clicks “X” button on each product. | 2. System redirects to product page.  4. System removes 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 chooses their cart and click “Checkout” button  3. Actor clicks “Process Order” button.  5. Actor inputs order information then click “Place Order” button.  7. Actor inputs Card owner’s name, card number and security code.  10. 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.  - “Order name” text field.  - “Address” text field.  - “Delivery date” text field.  - “Delivery place” text field.  - “Payment” checkbox.  6. Systems show Payment page if user chooses Pay with Card method.  8. System validates information.  9. System redirects to Order page and show order’s status. |
| **Alternative flow** | 1. Actor clicks “Go back to cart”  3. Actor clicks “X” button on each product. | 2. System redirects to cart page.  4. System removes product from cart. |
| **Exceptions** | **Actor’s action** | **System’s response** |
| 1. Actor input invalid voucher code.  2. Actor inputs more products than there are in stock.  3. Actor left “Billing name” text field blank.  4. Actor left “Address” text field blank.  5. Actor left “Delivery date” blank.  6. Actor left “Payment” checkboxes blank.  7. Actor inputs invalid card information. | System shows messages:  - “Invalid voucher”.  - “Product out of stock”.  - “Order name must not be blank”.  “Address must not be blank”.  “Must choose a delivery date”.  “Must choose a payment method”.  “Invalid card”. |

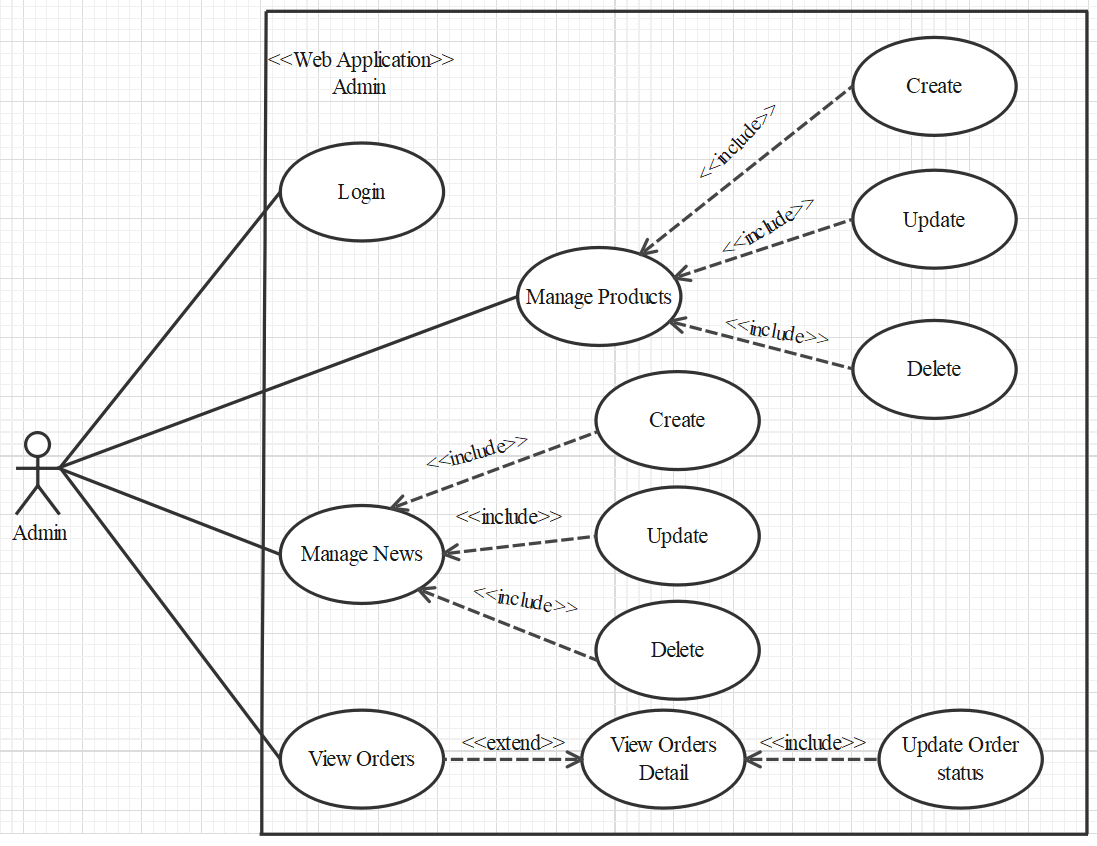
**Cancel order**

|  |  |  |
| --- | --- | --- |
| **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 Tab on Menu  3. Actor chooses Order Tab on sides bar.  5. Actor selects the order which they want to cancel in Order List and clicks “Cancel” Button. | 2. System redirects to Account page.  4. System redirects to Order List page and displays a list of existing orders.  6. System validates inputs.  7. System change order status in database.  8. System displays message: successful for error. |

**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 Tab on Menu  3. Actor chooses Order Tab on sides bar.  5. Actor selects the order that they want to view in Order List and clicks “View” Button. | 2. System redirects to Account page.  4. System redirects to Order List page and displays a list of existing orders and their status.  6. System validates inputs.  7. System displays all order’s detail and its status. |

**Admin**



1. **Login**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Login | |
| **Actor** | Admin | |
| **Description** | When admin wants to manage products, news, 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 informations. 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. |

1. **Products**
2. **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. |

1. **Update Product**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Update product’s detail | |
| **Actor** | Admin | |
| **Description** | When admin wants to update product | |
| **Requirements** | Admin provides product’s informations | |
| **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. |

1. **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. |

1. **News**
2. **Create new News**

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

1. **Update News**

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

1. **Delete News**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Delete News | |
| **Actor** | Admin | |
| **Description** | When admin wants to delete News from the webiste | |
| **Requirements** | Admin provides News 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 News Tab on menu.  3. Actor clicks ‘Delete’ button. | 2. System redirects to ‘News List’ page and display a list of News.  4. System delete News in the website  5. System displays confirmation or error. |

1. **Order**
2. **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 can not 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. |

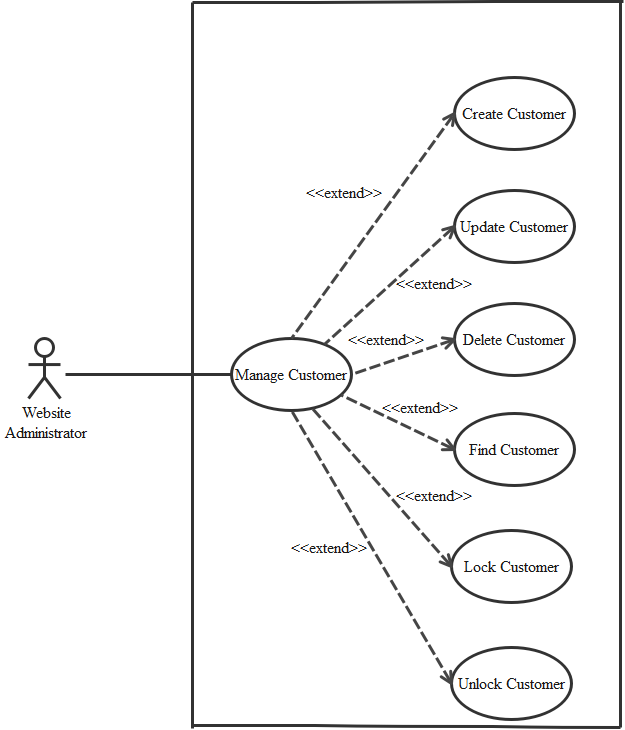
1. **View Order Detail**

|  |  |  |
| --- | --- | --- |
| **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. |

1. **Update Order Status**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Update Order Status | |
| **Actor** | Admin | |
| **Description** | When admin wants to update Order Status from the webiste | |
| **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**



1. **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  - gender  - active  6. System validates inputs.  7. System adds new customer into database.  8. System displays confirmation message or error message. |

1. **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 ‘Custumer’ page and display a list of existing Customers.  4. System displays a form with the following fields:  - fullname  - username  - password  - address  - phone  - email  - birthday  - gender  - 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. |

1. **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. |

1. **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. |

1. **Lock Customer**

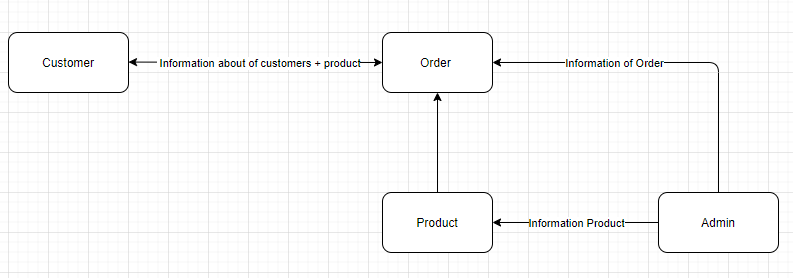
|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Lock Customer | |
| **Actors** | Admin | |
| **Description** | When admin wants to lock a customer | |
| **Requirements** | Actor provides Customer’s information | |
| **Pre-conditions** | Actor is logged in admin | |
| **Post-condition** | 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. |

1. **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. |

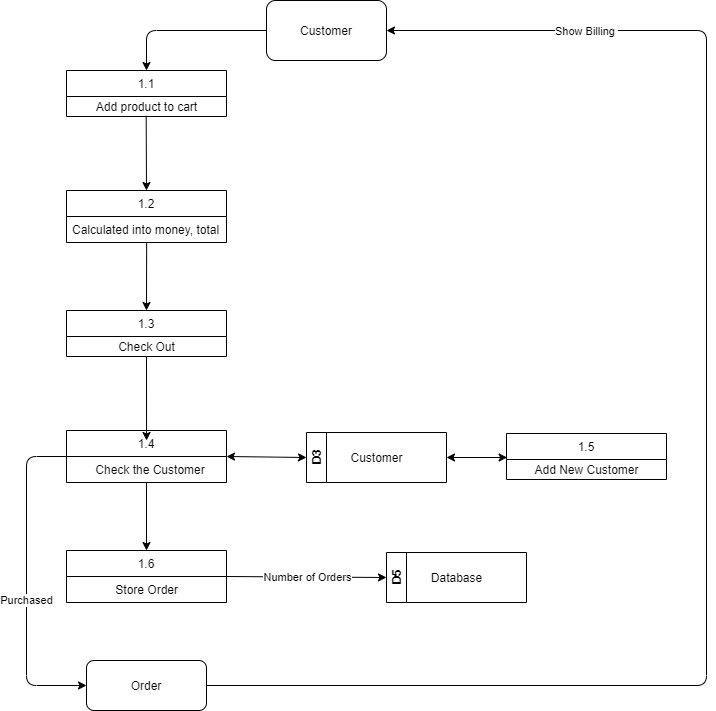
**2. Data Flow Diagram (DFD)**

**2.1 Level 0:**

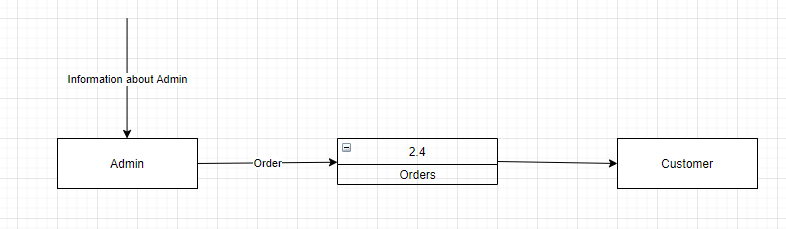
****

**2.2 Level 1:**

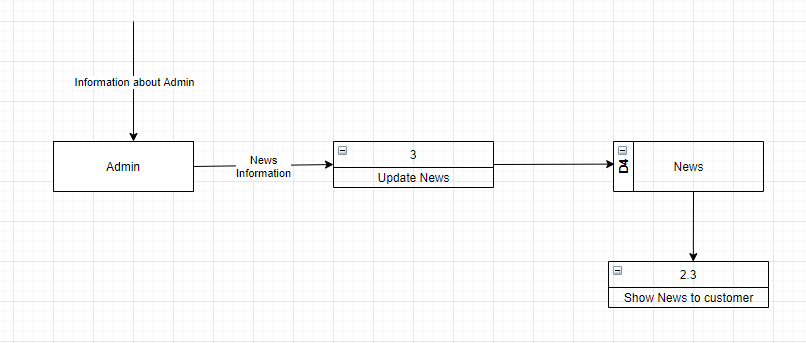
**- Function 1:**

****

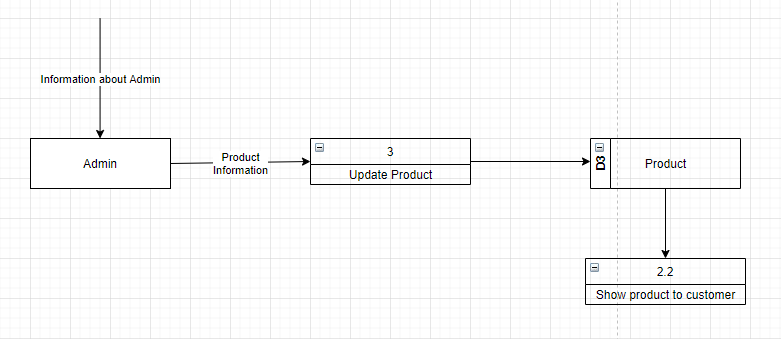
**- Function 2:**

****

**- Function 3:**

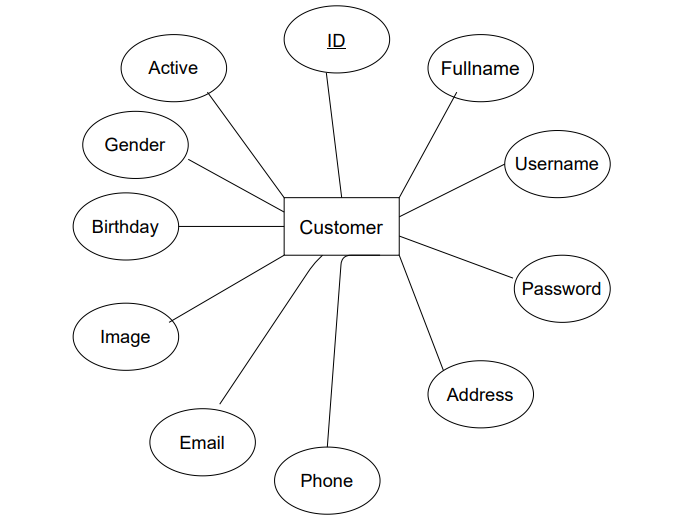
****

**- Function 4:**

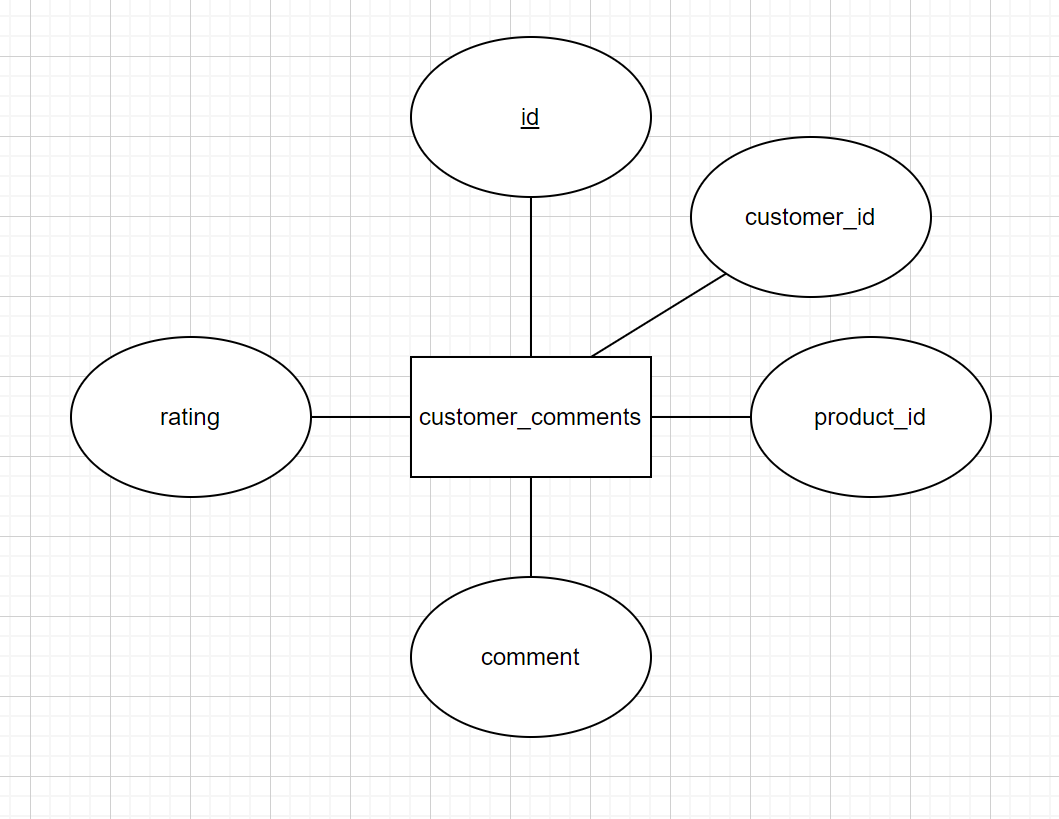
****

**3. ENTITY PROPERTIES**

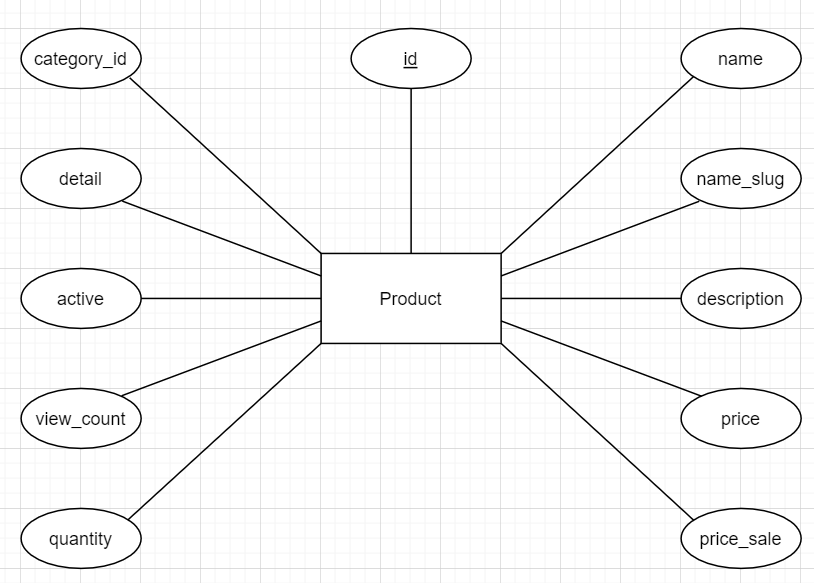
**3.1 Customer**



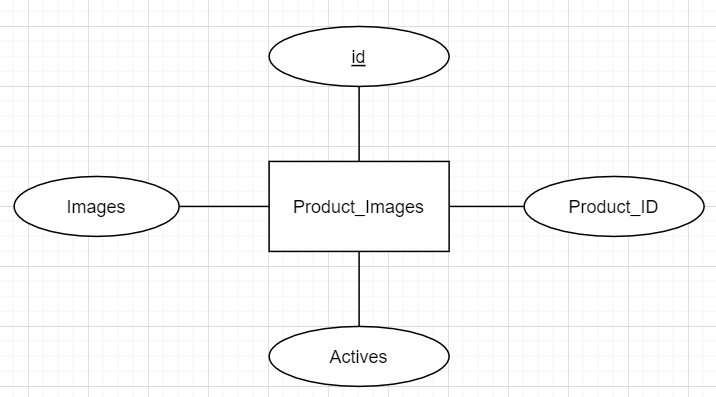
**3.2 Customer\_Comment**



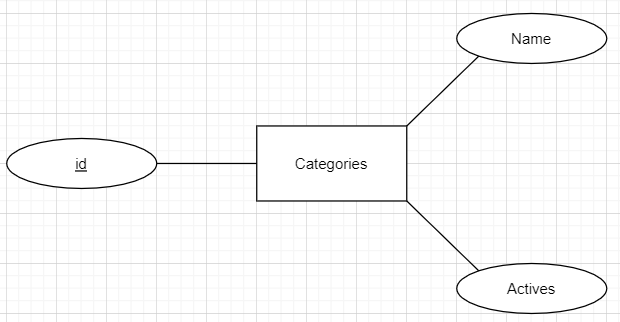
**3.3 Product**



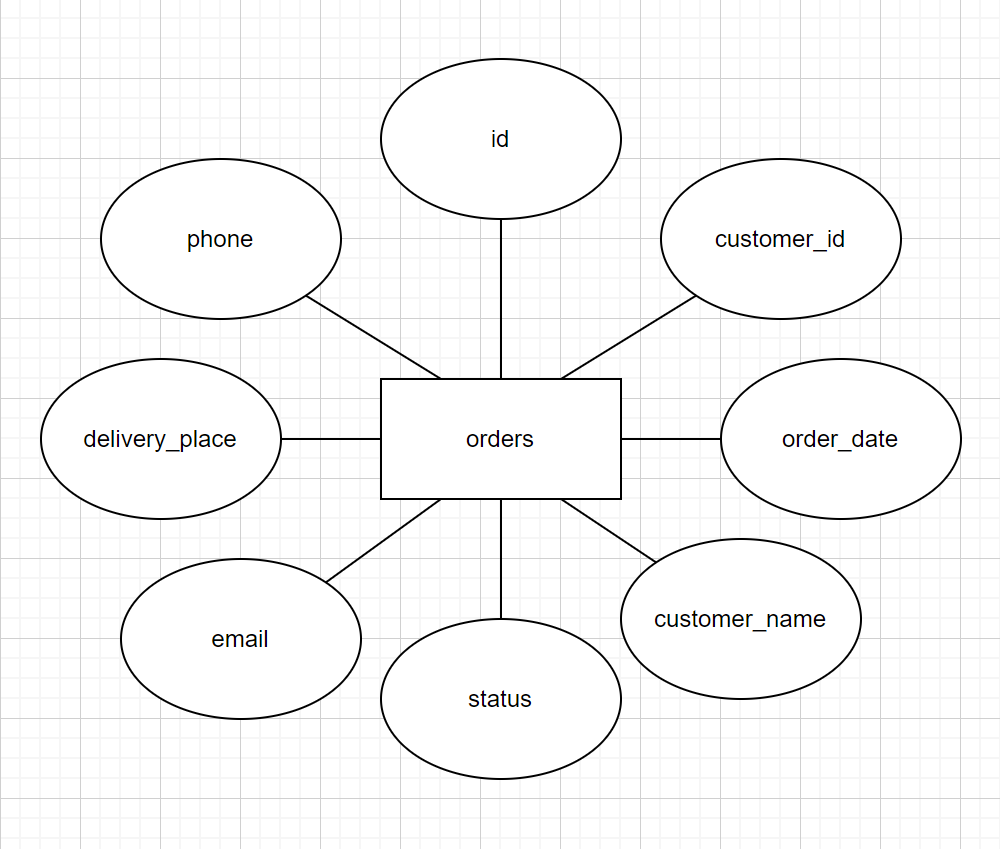
**3.4 Product\_Image**



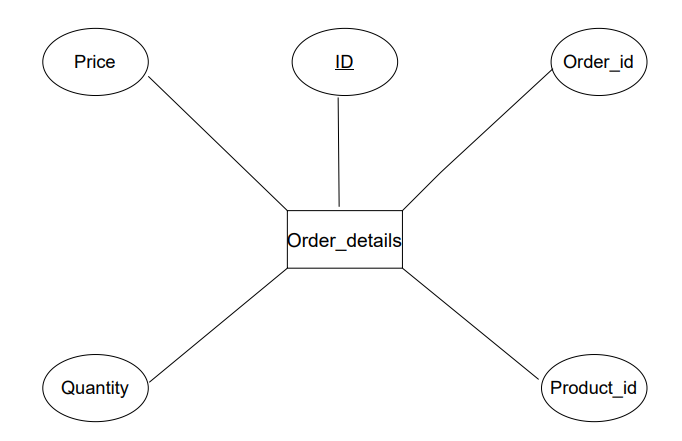
**3.5 Category**



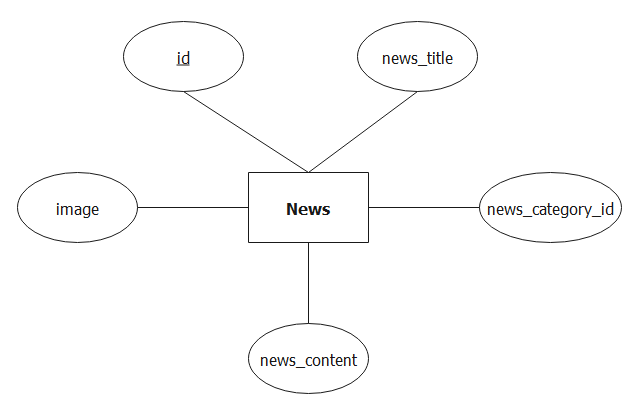
**3.6 Order**



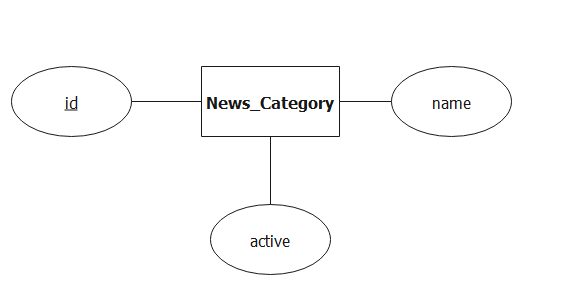
**3.7 Order\_Detail**



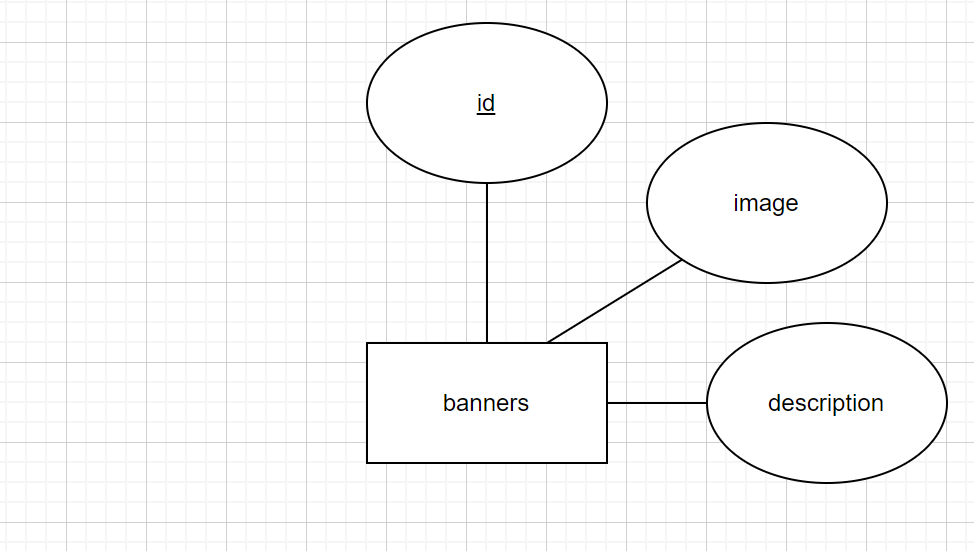
**3.8 News**

****

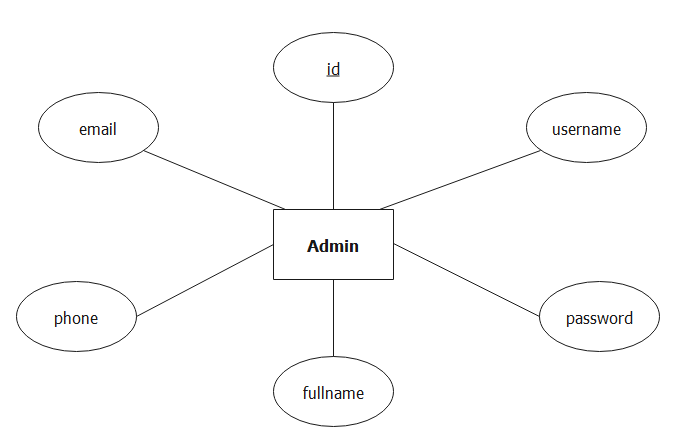
**3.9 News\_Category**

****

**3.10 Banner**

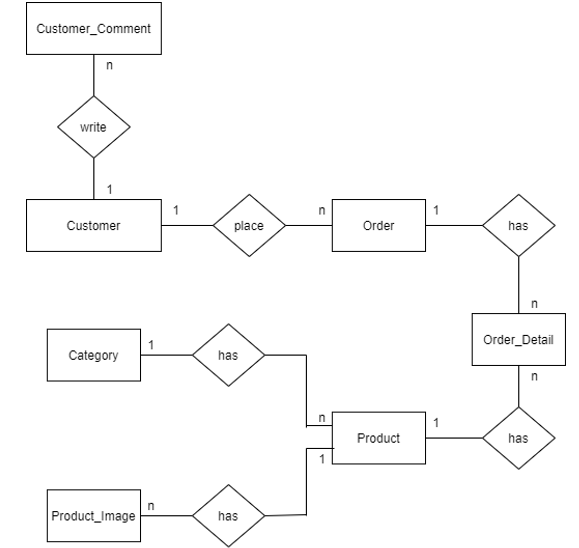


**3.11 Admin**

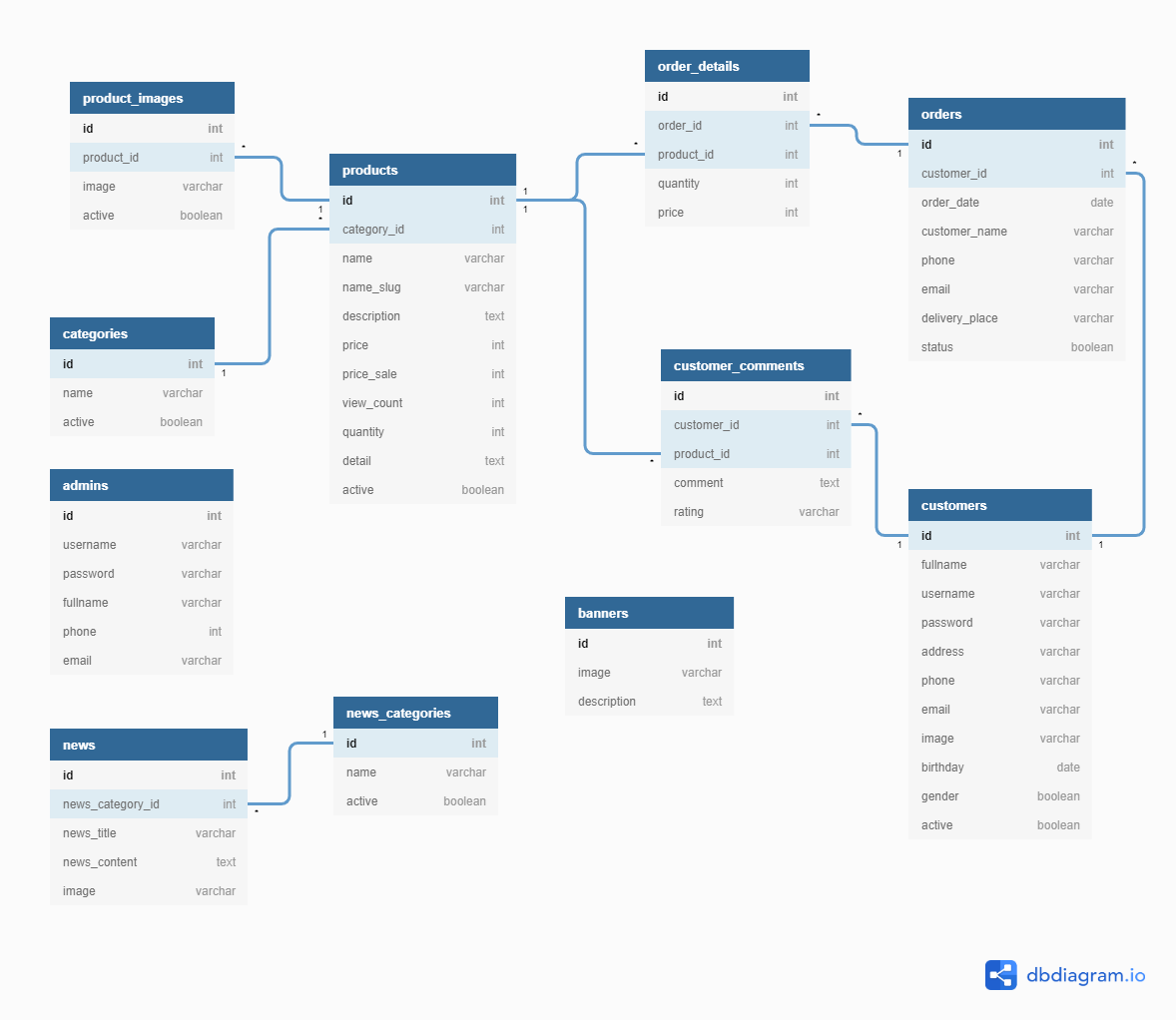
****

**4. Entity Relationship Diagram (ERD)**

**4.1 Logical ERD**

****

**4.2 Physical ERD**

****

**5. TABLE DESIGN**

**5.1 Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Customer** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 1 | Id | BIGINT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of the customer |
| 2 | fullname | VARCHAR(225) | NOT NULL | Store the full name of the customer |
| 3 | username | VARCHAR(225) | NOT NULL | Store the user name of the customer |
| 4 | password | VARCHAR(225) | NOT NULL | Store the password of the customer |
| 5 | address | VARCHAR(225) | NOT NULL | Store the Address of the customer |
| 6 | phone | VARCHAR(225) | NOT NULL | Store the Phone of the customer |
| 7 | email | VARCHAR(225) | NOT NULL | Store the Email of the customer |
| 8 | image | VARCHAR(225) | NOT NULL | Store the Image of the customer |
| 9 | birthday | DATE | NOT NULL | Store the Birthday of the customer |
| 10 | gender | BOOLEAN | NOT NULL | Store the Gender of the customer |
| 11 | active | BOOLEAN | NOT NULL | Store the Status the of the customer |

**5.2 Customer\_Comment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Customer\_comments** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 1 | id | BIGINT(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 | Store the ID of product |
| 4 | comment | DATE | NOT NULL | Store comment of customer |
| 5 | rating | INT | NOT NULL | Store rating of customer |

**5.3 Product**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: products** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 1 | Id | BIGINT(20) IDENTITY(1,1) | 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 | name | VARCHAR(225) | NOT NULL | Stores the name of product |
| 4 | name\_slug | VARCHAR(225) | NOT NULL | Stores the name\_slug of product |
| 5 | desciption | TEXT | NOT NULL | Stores the desciption of product |
| 6 | price | INT | NOT NULL | Stores the price of product |
| 7 | view\_count | INT | NOT NULL | Stores the number of purchased product |
| 8 | quantity | INT | NOT NULL | Stores the quantity of product |
| 9 | detail | TEXT | NOT NULL | Stores the name\_slug of product |
| 10 | active | BOOLEAN | NOT NULL | Stores the active of product |

**5.4 Product\_Image**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Products\_Images** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 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 | Strore the Active of the Cart |

**5.5 Category**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Categories** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 1 | id | BIGINT(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 | Strore the Active of the Cart |

**5.6 Order**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Orders** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 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\_place | VARCHAR(225) | NOT NULL | Store delivery address of receiver |
| 8 | status | BOOLEAN | NOT NULL | Store status of order |

**5.7 Order\_Detail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Order\_details** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 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 | price | INT | NOT NULL | Stores the price of product |

**5.8 News**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: news** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL PRIMARY KEY | Stores the ID of News |
| 2 | news\_category\_id | BIGINT(20) | NOT NULL FOREIGN KEY | Stores the Id of News\_Category |
| 3 | news\_title | VARCHAR(255) | NOT NULL | Stores the Title of News |
| 4 | news\_content | VARCHAR(255) | NOT NULL | Stores the Content of News |
| 5 | image | VARCHAR(255) | NOT NULL | Stores the Image of News |

**5.9 News\_Category**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: news\_categories** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL PRIMARY KEY | Stores the ID of News\_Category |
| 2 | name | VARCHAR(255) | NOT NULL | Stores the Name of News\_Category |
| 3 | active | BIT(1) | NOT NULL | Stores the Status of News\_Category |

**5.10 Banner**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Banners** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Desciption** |
| 1 | id | BIGINT(20) IDENTITY(1,1) | NOT NULL  PRIMARY KEY | Store the ID of banner |
| 2 | image | VARCHAR | NOT NULL  FOREIGN KEY | Store image of banner |
| 3 | description | VARCHAR(100) | NOT NULL | Store description of banner |

**5.11 Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: admins** | | | | |
| **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 | phone | VARCHAR(255) | NOT NULL | Stores the Phone of Admin |
| 6 | email | VARCHAR(255) | NOT NULL | Stores the Email of Admin |

# TASK SHEET REVIEW 2

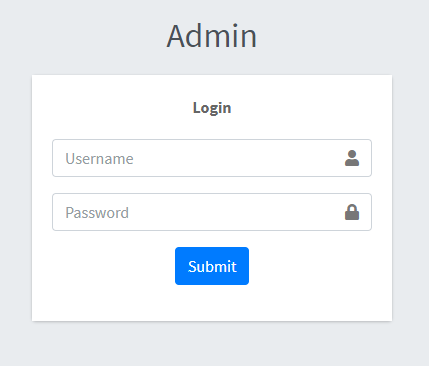
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.: Organic Product Store** | | **Project Name** | **Activity Plan Prepared By** | **Date of Preparation of Activity Plan:** | | | |
| **Sr./No.** | **Task** | **Actual Start Date** | **Actual Days** | **Team member name** | **Status** |
| 1 | Use Case | Organic Product Store | Chi | 15-Jun-20 | 1 | Chi | Completed |
| 2 | Data Flow Diagram (DFD) |  | 1 | Hạnh | Completed |
| 3 | Entity Relationship Diagram (ERD) |  | 1 | Chi | Completed |
| 4 | Entity Properties |  | 2 | Đạt | Completed |
| 5 | Table Design |  | 1 | Mai | Completed |

|  |  |
| --- | --- |
| **Date: 17 Jun 2020** | |
| Signature of Instructor:  **Mr. Hoang Duc Quang** | Signature of Team Leader:  **Nguyen Viet Minh Chi** |

**REVIEW 3**

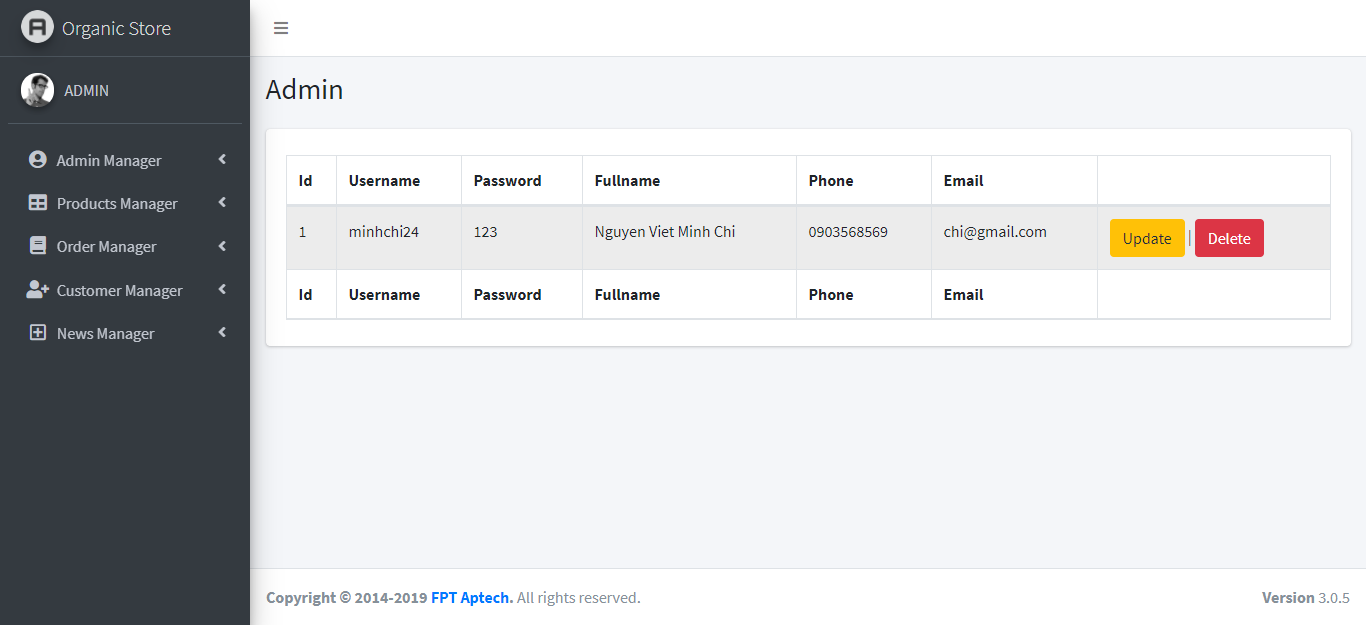
**GUI DESIGN**

**I. You must first login before entering the site**

****

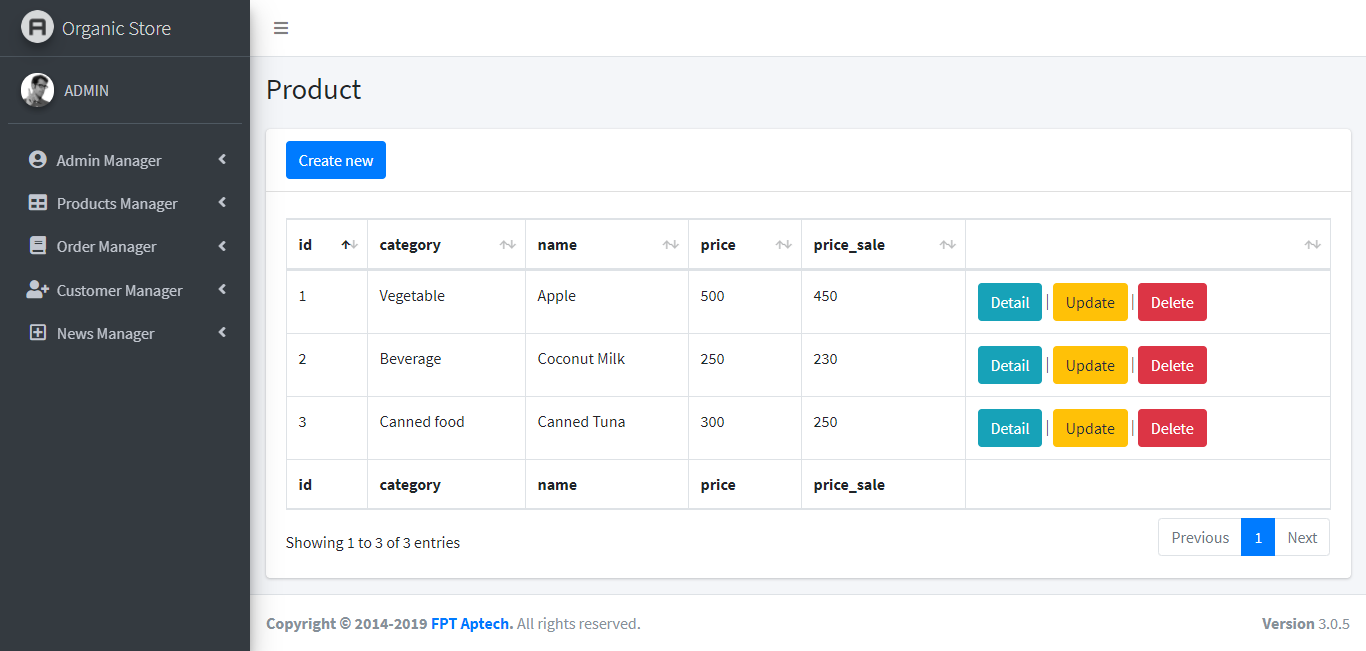
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | username | Not null, force matches in the database. |  | Admin’s account name. |
| 2 | password | Not null, force matches in the database. |  | Admin’s password. |

**II. Admin manager**

****

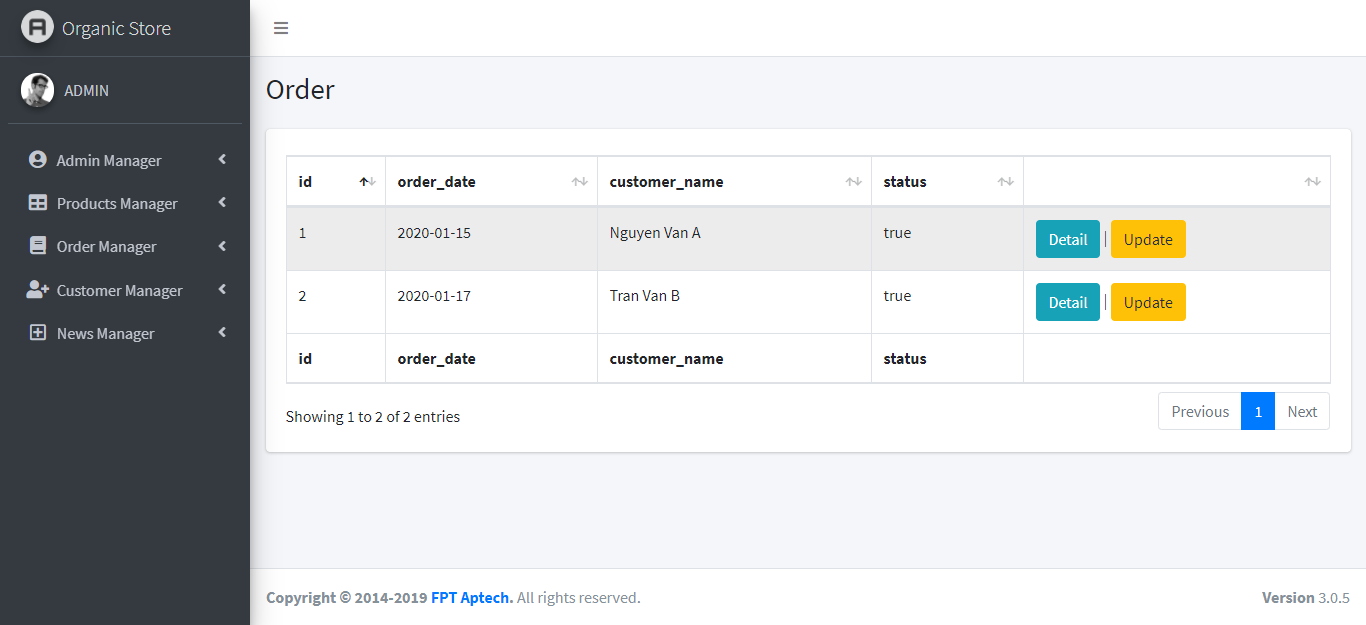
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Admin’s id. |
| 2 | username | Not null, force matches in the database. |  | Admin’s account name. |
| 3 | password | Not null, force matches 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. |

**III. Product manager**

****

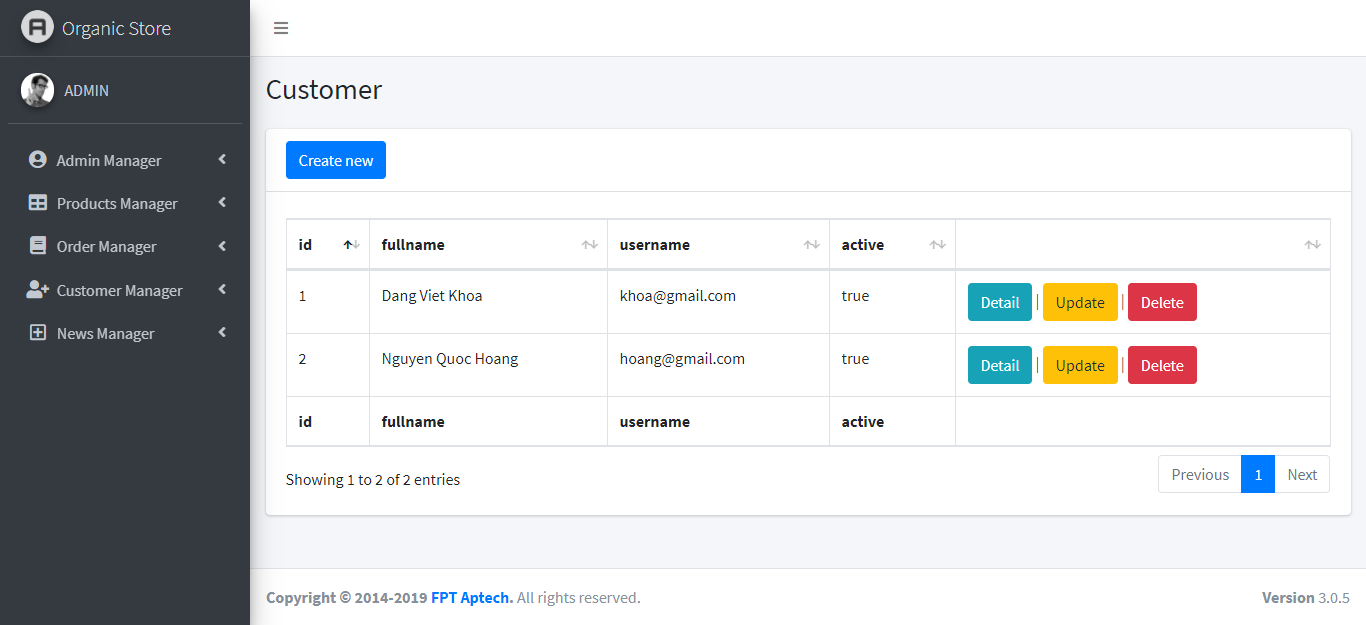
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Product’s id. |
| 2 | category | Not null, force matches in the database. |  | Product’s category. |
| 3 | name | Not null, force matches in 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. |

**IV. Order manager**

****

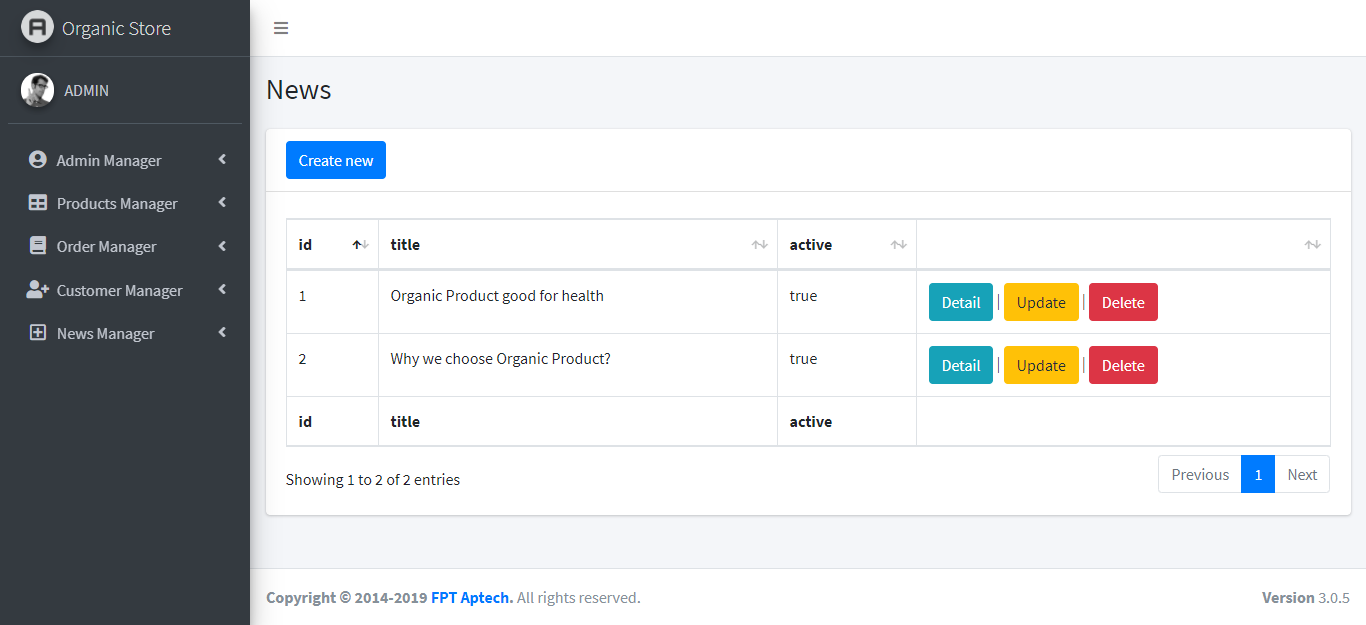
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Validation** | **Event** | **Description** |
| 1 | id | Not null, must be number. |  | Order’s id. |
| 2 | order\_date | Not null, must be date. |  | Order’s date. |
| 3 | customer\_name | Not null, not number. |  | Customer’s name. |
| 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. |

**V. 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. |

**VI. News manager**

****

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