

**PROJECT REPORT**

# VTC Store website

|  |  |
| --- | --- |
| **Semmester:** | Semester II |
| **Class:** | WD03 |
| **Group:** | G03 |
| **Faculty Tutorial :** | Vũ Trần Lâm |
| **Members:** | NDE18011 – Ngô Đức Nam( Leader )  NDE18074 – Nguyễn Trung Thăng( Secretary )  NDE18091 – Lý Bảo Thắng( Member )  NDE18037 – Ngô Anh Tuấn( Member )  NDE18082 – Nguyễn Quốc Quảng( Member ) |

# Addendum

**VTC Store website 1**

**Addendum 2**

**I.** **Project introduction 3**

**1.** **Proposed System 3**

**2.** **Purpose and objectives of the project 3**

**3.** **Application range 3**

**4.** **System name 3**

**5.** **Deployment environment (hardware and software) 3**

**6.** **Tools used for development 3**

**7.** **Identify customer requirements 3**

**II.** **Analyze system requirements 4**

**1.** **Use Case 4**

**2.** **Activity Diagram: 44**

**3.** **Sequence Diagram: 85**

**III.** **Detailed design 106**

**1.** **UI design 106**

**2.** **Class Diagram 129**

**3.** **Database design 131**

**IV.** **Test 136**

**V.** **Assign work to each team member 136**

**VI.** **Installation Instructions 154**

**1.** **Deployment Diagram 154**

**2.** **Installation steps 154**

**Contents 154**

# Project introduction

## Proposed System

* Building a website

## Purpose and objectives of the project

* Purpose: Building an online website
* Objectives:
* Meet the standards of a website::
* Scientific structure, tight, clear, easy to navigate, exploit and edit
* User friendly
* Stable operation, security
* Meet the functions required by customers (online sales)

## Application range

* Devices with network connection

## System name

* VTC Store

## Deployment environment (hardware and software)

* The system is installed in the appropriate environment or on the appropriate server and on any client station requires using it if there is an internet connection.
* Hardware : Desktop, PC and Mobile...

## Tools used for development

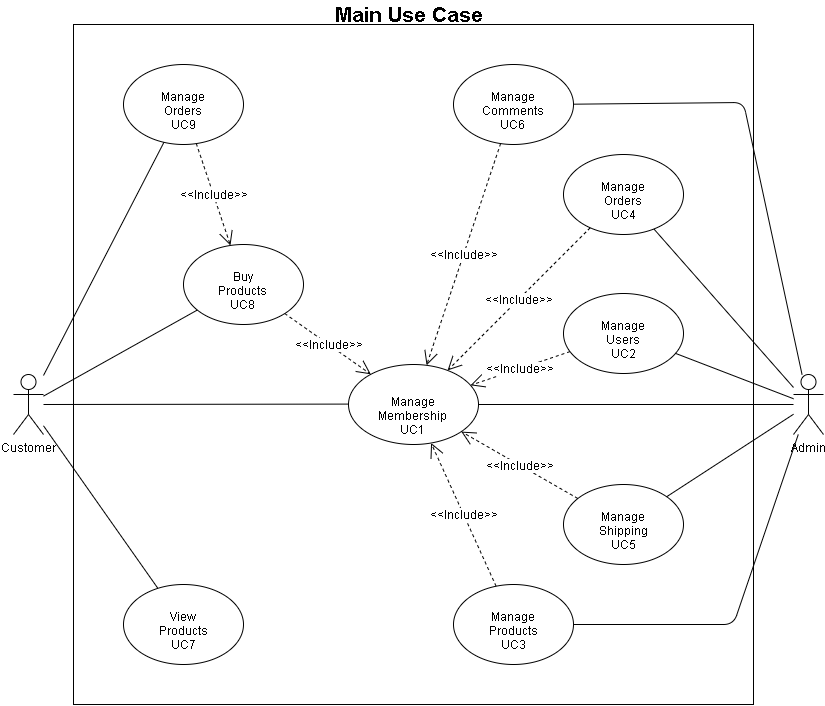
* Programming language : PHP
* Database management system : MySQL server
* Reporting tools: Microsoft Word
* Tools used for development: Visual Studio Code, Violet, Mysql, Workbench, draw.io

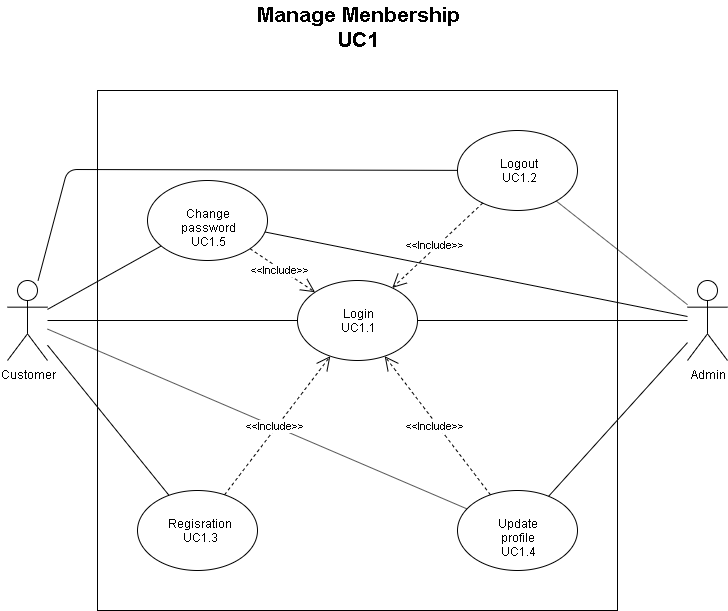
## Identify customer requirements

* Admins:
* Admins want to manage membership
* Admins want to manage users
* Admins want to manage products
* Admins wants to manage orders
* Admins want to manage comments
* Customers:
* Customers want to see products.
* Customers want to see product details.
* Customers want to buy products.
* Customers want to order

# Analyze system requirements

## Use Case





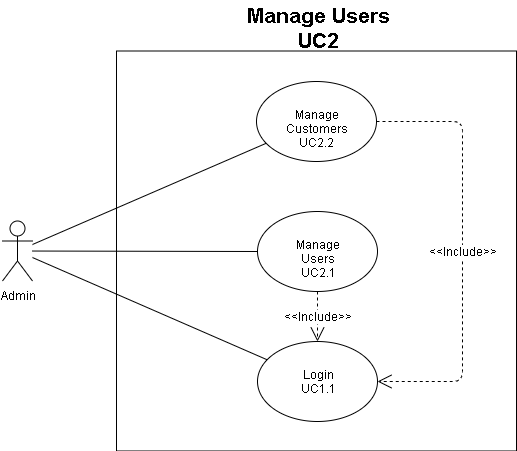
|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Use Case ID** | UC1.1 |
| **Description** | Users want to login to perform the program's functions |
| **Actor** | Admin and customer |
| **Triggers** | Avoid for outsiders to use member tasks  User -> enter account, password -> login |
| **Preconditions** | The browser is running on the computer and the user must have an account |
| **Postconditions** | The user has logged in on the browser |
| **Main Course** | 1. Enter the website address  2. Select the passenger icon  3. Enter the information in the form           3.1 Enter the account           3.2 Enter the password  4. Display the homepage. |
| **Alternate Courses** | AC1: Invalid Email / password   1. The message "Email / password is invalid."   AC2: Leave email / password blank   1. The message "Email / password must not be blank" |
| **Exceptions** | Ex1: Lost connection to the server   1. The message "Lost connection to server."   Ex 2: The user exits the browser |

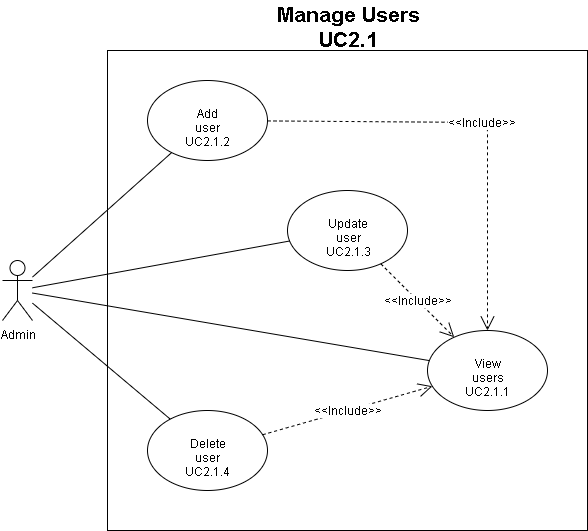
|  |  |
| --- | --- |
| **Use Case Name** | Logout |
| **Use Case ID** | UC1.2 |
| **Description** | User wants to logout browser |
| **Actor** | Admin and customer |
| **Triggers** | Avoid members' information still stored in the browser  User -> click the logout button -> logout |
| **Preconditions** | The browser is running on the computer and the user is logged out |
| **Postconditions** | The user has been logged out of the browser |
| **Main Course** | 1. Select the logout button  2. Display the home page |
| **Alternate Courses** |  |
| **Exceptions** | Ex1: Lost connection to the server   1. The message "Lost connection to server."   Ex 2: The user exits the browser |

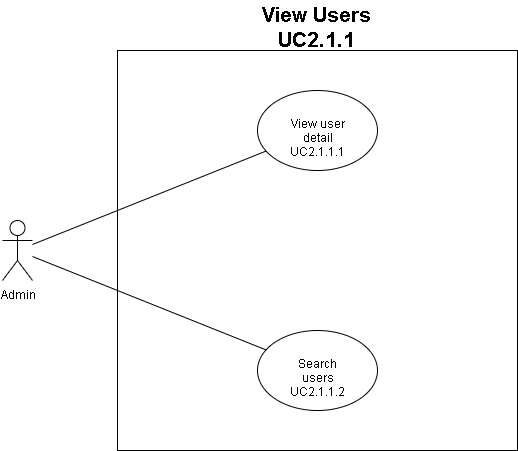
|  |  |
| --- | --- |
| **Use Case Name** | Registration |
| **Use Case ID** | UC1.3 |
| **Description** | Customers want to register to perform the member's functions |
| **Actor** | Customer |
| **Triggers** | Become a member of the program  Customers -> enter information -> register |
| **Preconditions** | The browser is running on the computer |
| **Postconditions** | Customer has successfully registered |
| **Main Course** | 1. Enter the website address  2. Select the passenger icon  3. Click the register button  4. Enter the information           4.1 Enter full name - Filter: Enter the characters as unsigned letters -> the message "The input data must be unsigned letters."           4.2 Select the date           4.3 Entering Address - Filter: Enter special characters -> the message "The input data is not a special character."           4.4 Enter email address - Filter: Type in the format -> the message "The input must be an email address."           4.5 Enter Password - Filter: Enter the number of characters under 6 -> the message "The input data must be greater than or equal to 6."           4.6 Retype Password - Filter: Enter the same password entered -> the message "Password does not match."    4. Display the home page |
| **Alternate Courses** | AC4.4: Invalid Email / password   1. The message "Email / password is invalid."   AC4.5: Leave email / password blank   1. The message "Email / password may not be blank."   AC4.4: Enter Email already exists   1. The message "Email already exists."   AC4.5 và 4.6: Re-enter Password and Re-enter Password do not match   1. The message "Password does not match." |
| **Exceptions** | Ex1: Lost connection to the server   1. The message "Lost connection to server."   Ex 2: The user exits the browser |

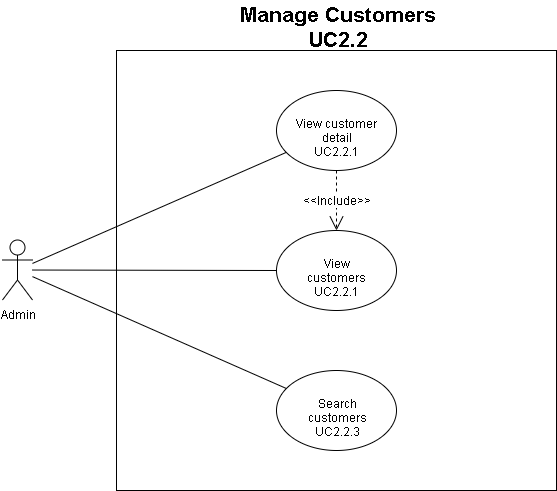
|  |  |
| --- | --- |
| **Use Case Name** | Update profile |
| **Use Case ID** | UC1.4 |
| **Description** | Users want to change personal information |
| **Actor** | Admin and customer |
| **Triggers** | Change existing personal information  Customers -> enter information -> update |
| **Preconditions** | The browser is running on the computer, the customer is logged in, and is on the profile page |
| **Postconditions** | Customer successfully updated information |
| **Main Course** | 1. Select the passenger icon  2. Enter the information           2.1 Enter customer name - Filter: Enter the characters that are unsigned letters -> the message "The input data must be unsigned letters."           2.2 Select the date and year           2.3 Enter address - Filter: Enter special characters -> the message "The input data is not a special character."  3. Display the profile page |
| **Alternate Courses** |  |
| **Exceptions** | Ex1: Lost connection to the server   1. The message "Lost connection to server."   Ex 2: The user exits the browser |

|  |  |
| --- | --- |
| **Use Case Name** | Change password |
| **Use Case ID** | UC1.5 |
| **Description** | Users want to change password |
| **Actor** | Admin and customer |
| **Triggers** | Change the current password  Customers -> enter new password -> change |
| **Preconditions** | The browser is running on the computer, the customer is logged in, and is on the profile page |
| **Postconditions** | The customer has successfully changed |
| **Main Course** | 1. Select the passenger icon  2. Enter the password           2.1 Enter new password - Filter: Enter the number of characters under 6 -> the message "The input data must be greater than or equal to 6."           2.2 Retype new password - Filter: Enter the same password entered -> the message "The password does not match."    3. Display the password page |
| **Alternate Courses** | AC1: Re-enter Password and Re-enter Password do not match   1. The message "Password does not match." |
| **Exceptions** | Ex1: Lost connection to the server   1. The message "Lost connection to server."   Ex 2: The user exits the browser |









|  |  |
| --- | --- |
| **Use Case Name** | View user |
| **Use Case ID** | UC2.1.1 |
| **Description** | Admin description function to viewuser account information |
| **Actor** | Admin. |
| **Triggers** | Admin wants to see user account information |
| **Preconditions** | 1. Admin is logged in. 2. Admins want to see user account information |
| **Postconditions** | Admins can view user account information including: full name, phone number, address, gmail of users |
| **Main Course** | Function starts when Admin wants to see delivery status. Admins select items in order: Manage users-> users-> view users. |

|  |  |
| --- | --- |
| **Use Case Name** | View user detail |
| **Use Case ID** | UC2.1.1.1 |
| **Description** | Admin description function to viewuser account details |
| **Actor** | Admin. |
| **Triggers** | Admin wants to see user account details |
| **Preconditions** | 1. Admin is logged in. 2. user |
| **Postconditions** | can view user account details including: full name, phone number, address, gmail of the user |
| **Main Course** | Function captures first when the Admin wants to see delivery status. Admins select items in order: Manage users-> users-> view users-> user details. |

|  |  |
| --- | --- |
| **Use Case Name** | Search user |
| **Use Case ID** | UC2.1.1.2 |
| **Description** | Admin description function to find any user. |
| **Actor** | Admin. |
| **Triggers** | Admin to search for users. |
| **Preconditions** | 1. Admin has logged in. 2. Admin want to search users on the system. |
| **Postconditions** | Success: Admin found user on the system.  Fail: User was not found because there is no information on the system or incorrect search information. |
| **Main Course** | This function starts when an admin wants to find customers on the system.  1. Admin choose the following items in order:                    Manage users -> user -> search users.   * 2. Admin enter the user name to search. |
| **Exceptions** | When no search is found the customer will give an example such as "User does not exist". |

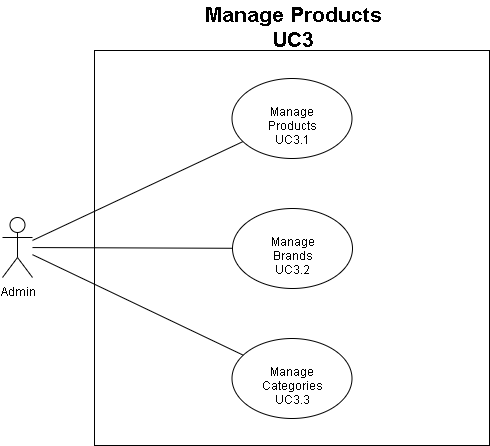
|  |  |
| --- | --- |
| **Use Case Name** | Add user |
| **Use Case ID** | UC2.1.2 |
| **Description** | Function for admin to add new users. |
| **Actor** | Admin. |
| **Triggers** | When the admin wants to add a new user. |
| **Preconditions** | 1) Admin is logged in.          2) Admin wants to add new users. |
| **Postconditions** | Success: Admin added a new user to the system.  Failure: Unable to add because the user is already in the system. |
| **Main Course** | * This function starts when the admin wants to add a new user. * 1. Admin select items in order: * Manage user-> user-> view user-> add user. * 2. Admin fill in the information of new user: full name, gmail, address, phone number. * 3. Admin confirms whether you want to add a new user. |
| **Alternate Courses** | When the Admin does not enter information in one or the required fields, the system will display a request to enter information in those items. |
| **Exceptions** | Admin has filled in information but added failure   1. The system will display the message "Add user failed" with the same reason: for example, the user account already exists.          2) Admin cancels further confirmation. |

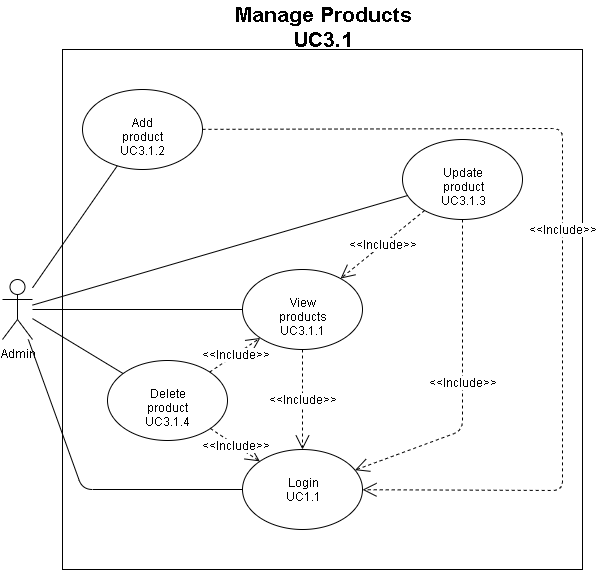
|  |  |
| --- | --- |
| **Use Case Name** | Update user |
| **Use Case ID** | UC2.1.3 |
| **Description** | Function for admin to update user information. |
| **Actor** | Admin. |
| **Triggers** | When the admin needs to update the user's information. |
| **Preconditions** | 1. Admin is logged in. 2. Admin needs to update user information. |
| **Postconditions** | Admin updates new information of users to the system. |
| **Main Course** | * This function starts when the admin wants to see the partners. * 1. Admin select items in order:               Manage user-> user-> view user-> update user.   * 2. Admin fill in the user information that you want to change eg username, email, address, phone number. * 3. Admin whether or not to update the user's new information. |
| **Exceptions** | Admin cancels to confirm more users. |

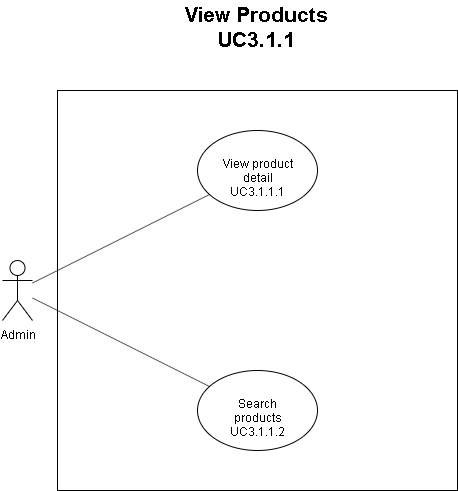
|  |  |
| --- | --- |
| **Use Case Name** | Delete user |
| **Use Case ID** | UC2.1.4 |
| **Description** | Function for admin to delete users. |
| **Actor** | Admin. |
| **Triggers** | When the admin wants to delete users on the system. |
| **Preconditions** | 1. Admin is logged in. 2. Admin wants to delete users on the system. |
| **Postconditions** | Success: Admin deleted the user on the system. |
| **Main Course** | * This function starts when the admin wants to delete users on the system. * 1. Admin select items in order: * Manage user-> user-> view user -> delete user. * 2. Admin confirms that it wants to delete the user. |
| **Exceptions** | Admin canceled confirmation of deleting the user. |

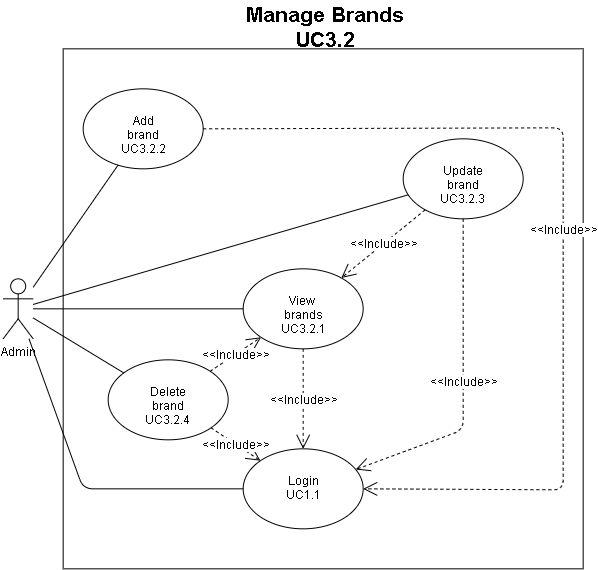
|  |  |
| --- | --- |
| **Use Case Name** | View customers |
| **Use Case ID** | UC2.2.1 |
| **Description** | The admin description function views the client. |
| **Actor** | Admin. |
| **Triggers** | 1. Admin is logged in. 2. Admin wants to see the customers. |
| **Preconditions** | Admin wants to see the customers. |
| **Postconditions** | Admin can view the list of customers. |
| **Main Course** | Function starts when the admin wants to see the customers. Admin selects items in order: Manage user -> customers -> view customers. |

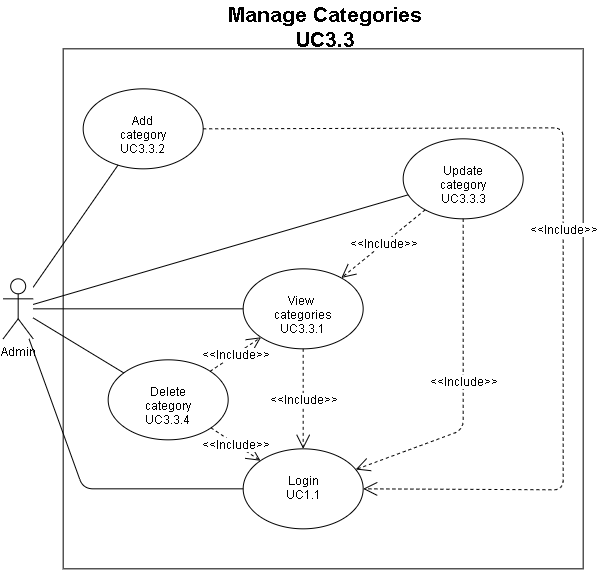
|  |  |
| --- | --- |
| **Use Case Name** | Search customer |
| **Use Case ID** | UC2.2.3 |
| **Description** | Admin description function to find any customer. |
| **Actor** | Admin. |
| **Triggers** | Admin want to find customers. |
| **Preconditions** | 1. Admin has logged in. 2. Admin want to find customer on the system. |
| **Postconditions** | Success: Admin found customers on the system.  Fail : customer not found because there is no information on the system or incorrect search information. |
| **Main Course** | This function starts when an Admin wants to find customers on the system.  1. Admins choose the items in order:                    Manage users -> customers -> search customers.   * 2. Admin enter the customer name to search. |
| **Exceptions** | When not search for information, customers will give an example such as "customer does not exist". |











|  |  |
| --- | --- |
| **Use Case Name** | View product  View product Detail |
| **Use Case ID** | UC3.1.1  UC3.1.1.1 |
| **Description** | Product list view function |
| **Actor** | Admin. |
| **Trigger** | Admin wants to see the list of products information. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to see the list of products. |
| **Post-condition** | Admin view is all items included: Product code, Product name, price, photo, category, brand, total amount, size and quantity of each size. |
| **The main Event stream** | This function starts when the admin wants to see the product list   1. Admin selection item in order: Manage Products -> Products List The list will display detailed information of all products |

|  |  |
| --- | --- |
| **Use Case Name** | Search product |
| **Use Case ID** | UC3.1.1.2 |
| **Description** | The admin function searches for a product in the list of products. |
| **Actor** | Admin. |
| **Trigger** | Admin wants to search for a product in the product list. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to search for a product in the product list. |
| **Post-condition** | Admin Search success. |
| **The main Event stream** | This function starts when the admin wants to search for a product in the product list   1. Admin selection item in order: Manage Products -> Products List->Search Product |

|  |  |
| --- | --- |
| **Use Case Name** | Add product |
| **Use Case ID** | UC3.1.2 |
| **Description** | Function Description  admin add new product. |
| **Actor** | Admin. |
| **Trigger** | 1. Admin signed in. 2. Admin want to add new products. |
| **Prerequisites** | Admin want to add new products. |
| **Post-condition** | Admin adds a new product success |
| **The main Event stream** | This function starts when the admin wants to add new product to the System.   1. 1. Admin Options item in order:   Manage Product -> Products List-> Add   1. 2. Admin fill in information of new products: Product name, photo, price, category, brand 2. 3. Admin confirm if you want to add new product. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin has filled out the information but added failed   1. The system displays a "Add product failure" message with the same cause for example: the product already exists.   Admin Cancel to confirm additional products. |

|  |  |
| --- | --- |
| **Use Case Name** | Update product |
| **Use Case ID** | UC3.1.3 |
| **Description** | Admin description Function Update product information. |
| **Actor** | Admin. |
| **Trigger** | 1. Admin signed in. 2. Admin wants to update product information. |
| **Prerequisites** | Admin wants to update product information |
| **Post-condition** | Admin updated successful product information. |
| **The main Event stream** | This function starts when the admin wants to see the partners.  1. Admin Options item in order:              Manage Product -> Products List-> Update.  1. Admin Options item in order:              Admin fill in the product's information: Product name, price, photo, category, brand, quantity.  1. Admin Options item in order:              Admin confirm if you want to update the product information. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin has filled out information but updated product information failed   1. The system will display the message "Update product information failed" in the same cause for example, the product information already exists.   Admin cancels confirmation of updated product information. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete product |
| **Use Case ID** | UC3.1.4 |
| **Description** | Function for admin Delete a product. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to delete a product from the system. |
| **Prerequisites** | 1)Admin signed in.        2)Admin want to delete a product into the system. |
| **Post-condition** | Success: Admin successfully deletes  a product from the system. |
| **The main Event stream** | This function starts when the admin wants to delete a product from the system.  1. Admin Options item in order:                   Manage Product -> Products List->  Delete. |
| **Exception** | Admin cancels confirm product deletion. |

|  |  |
| --- | --- |
| **Use Case Name** | View brand |
| **Use Case ID** | UC3.2.1 |
| **Description** | Brand  list view function |
| **Actor** | Admin. |
| **Trigger** | Admin wants to see the list of product brands. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to see the list of product brand. |
| **Post-condition** | Admin view is all items included: Brand code, Brand name. |
| **The main Event stream** | This function starts when the admin wants to see the brand  list   1. Admin selection item in order:   Manage Products -> Product brand list. |

|  |  |
| --- | --- |
| **Use Case Name** | Add brand |
| **Use Case ID** | UC3.2.2 |
| **Description** | Function for admin Add new brand. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to add new brand into the system. |
| **Prerequisites** | 1)Admin signed in.        2)Admin wants to add new brand into the system. |
| **Post-condition** | Success: Admin added new brand into the system.  Failure: Could not be added because the brand was already in the system. |
| **The main Event stream** | This function starts when the admin wants to add new brand into the system.  1. Admin Options item in order:                   Manage Product -> Product brand list -> Add.  2. Admin fill in new brand’s information: brand name.  3. Admin confirm if you want to add new brand. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin has filled out the information but added failed  1)The system will display a "Add new brand failed" message .  Admin Cancel to confirm additional brand. |

|  |  |
| --- | --- |
| **Use Case Name** | Update brand |
| **Use Case ID** | UC3.2.3 |
| **Description** | The admin function update the brand. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to update brand information. |
| **Prerequisites** | 1)Admin signed in.  Admin wants to update brand information. |
| **Post-condition** | Success: Admin updates the brand's success information. |
| **The main Event stream** | This function starts when the admin wants to update the brand’s information  1. Admin Options item in order:                   Manage Product -> Product brand list -> Update.  2. Admin fill in new brand's information: Brand name  3. Admin confirm if you want to update the brand type. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin cancels further confirmation of brand. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete brand |
| **Use Case ID** | UC3.2.4 |
| **Description** | Function for admin Delete a brand. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to delete a brand from the system. |
| **Prerequisites** | 1)Admin signed in.        2)Admin want to delete a brand into the system. |
| **Post-condition** | Success: Admin successfully deletes  a brand from the system. |
| **The main Event stream** | This function starts when the admin wants to delete a brand from the system.  1. Admin Options item in order:                   Manage brand -> Brands List->  Delete. |
| **Exception** | Admin cancels confirm brand deletion. |

|  |  |
| --- | --- |
| **Use Case Name** | Search brand |
| **Use Case ID** | UC3.2.5 |
| **Description** | The admin function searches for a brand in the list of brands. |
| **Actor** | Admin. |
| **Trigger** | Admin wants to search for a brand in the brand list. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to search for a brand in the brand list. |
| **Post-condition** | Admin Search success. |
| **The main Event stream** | This function starts when the admin wants to search for a brand in the brand list   1. Admin selection item in order:   Manage Products -> Product Brand List -> Search Brand |

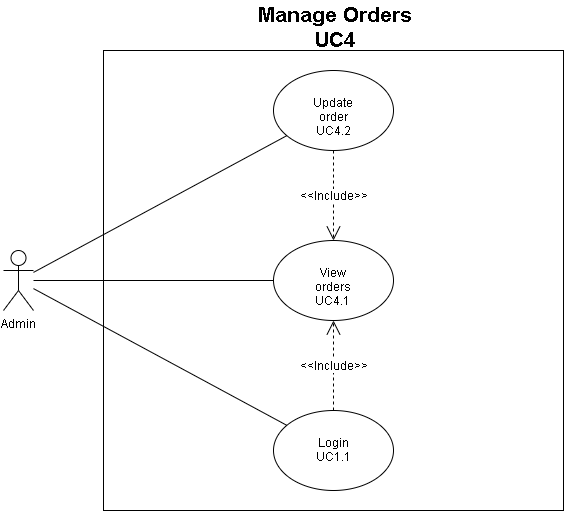
|  |  |
| --- | --- |
| **Use Case Name** | View category |
| **Use Case ID** | UC3.3.1 |
| **Description** | Category list view function |
| **Actor** | Admin. |
| **Trigger** | Admin wants to see the list of category. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to see the list of category. |
| **Post-condition** | Admin view is all items included: Category code, Category name. |
| **The main Event stream** | This function starts when the admin wants to see the category list   1. Admin selection item in order:   Manage Products -> Product category list. |

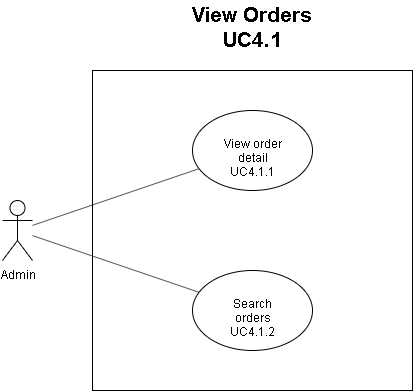
|  |  |
| --- | --- |
| **Use Case Name** | Add category |
| **Use Case ID** | UC3.3.2 |
| **Description** | Function for admin Add new category type. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to add new category type to the system. |
| **Prerequisites** | 1)Admin signed in.        2)Admin wants to add new category types to the system. |
| **Post-condition** | Success: Admin added new category type to the system.  Failure: Could not be added because the category type was already in the system. |
| **The main Event stream** | This function starts when the admin wants to add new category types to the system.  1. Admin Options item in order:                   Manage Product -> Product category list -> Add.  2. Admin fill in new category information: Category name.  3. Admin confirm if you want to add new category. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin has filled out the information but added failed  1)The system will display a "Add new category Type failed" message .  Admin Cancel to confirm additional category. |

|  |  |
| --- | --- |
| **Use Case Name** | Update category |
| **Use Case ID** | UC3.3.3 |
| **Description** | The admin function updates the category. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to update category information. |
| **Prerequisites** | 1)Admin signed in.  2)Admin wants to update category information. |
| **Post-condition** | Success: Admin updates the category's success information. |
| **The main Event stream** | This function starts when the admin wants to update the category information  1. Admin Options item in order:                   Manage Product -> Product category list -> Update.  2. Admin fill in new category information: Category name  3. Admin confirm if you want to update the category. |
| **Secondary event Flow** | When Admin does not enter information in one or the requested items, the system will display a request to enter information in those items. |
| **Exception** | Admin cancels further confirmation of category. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete category |
| **Use Case ID** | UC3.3.4 |
| **Description** | Function for admin Delete a category. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to delete a category from the system. |
| **Prerequisites** | 1)Admin signed in.        2)Admin want to delete a category into the system. |
| **Post-condition** | Success: Admin successfully deletes  a category from the system. |
| **The main Event stream** | This function starts when the admin wants to delete a category from the system.  1. Admin Options item in order:                   Manage category -> Categories List->  Delete. |
| **Exception** | Admin cancels confirm category deletion. |

|  |  |
| --- | --- |
| **Use Case Name** | Search category |
| **Use Case ID** | UC3.3.5 |
| **Description** | The admin function searches for a category in the list of categories. |
| **Actor** | Admin. |
| **Trigger** | Admin wants to search for a category in the category list. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to search for a category in the category list. |
| **Post-condition** | Admin Search success. |
| **The main Event stream** | This function starts when the admin wants to search for a category in the category list   1. Admin selection item in order:   Manage Products -> Category List -> Search Category |



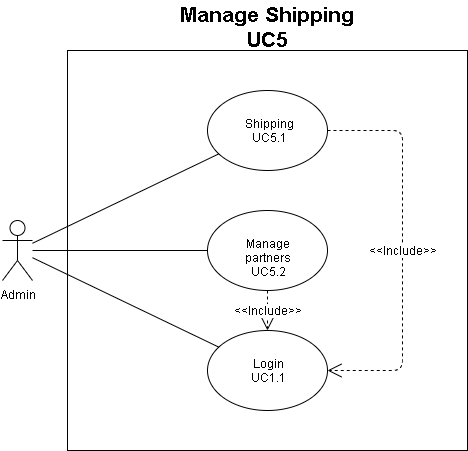


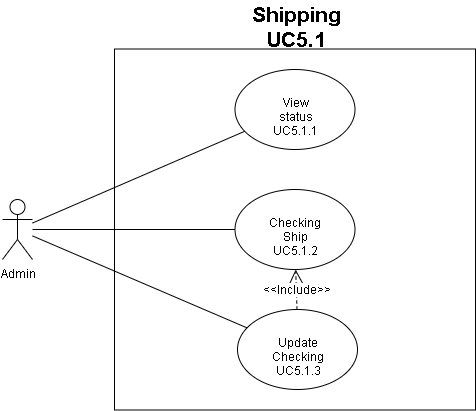
|  |  |
| --- | --- |
| **Use Case Name** | View order |
| **Use Case ID** | UC4.1 |
| **Description** | Function for admin View list order. |
| **Actor** | Admin. |
| **Trigger** | When the admin wants to see the order list. |
| **Prerequisites** | 1. Admin signed in. 2. Admin wants to see the order list. |
| **Post-condition** | Admin xem được orders. |
| **The main Event stream** | This function starts when the admin wants to see the list of orders in the system   1. 1. Admin Options item in order:                    Manage Order ->View List Order. |
| **Secondary event Flow** | No order Notification "No recent orders" |

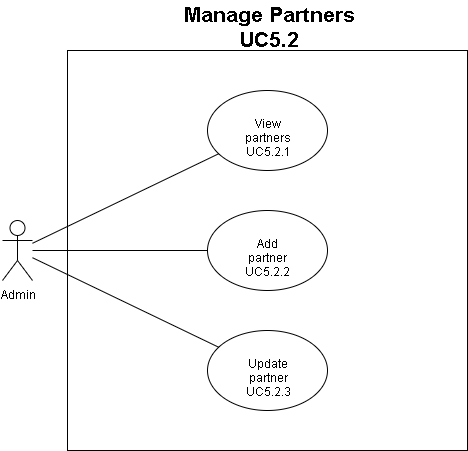
|  |  |
| --- | --- |
| **Use Case Name** | View order details |
| **Use Case ID** | UC4.1.1 |
| **Description** | The admin function view order details. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to view order details. |
| **Prerequisites** | 1)Admin signed in.   2)Admin wants to view order details. |
| **Post-condition** | Success: Admin viewed order details. |
| **The main Event stream** | This function starts when the admin wants to update order status  1. Admin Options item in order:                   Manage Order ->See more.  2. Admin view order details |
| **Exception** | Disconnect to database |

|  |  |
| --- | --- |
| **Use Case Name** | Search order |
| **Use Case ID** | UC4.1.2 |
| **Description** | Function for admin looking for a order in list order. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to search for a order in list order |
| **Prerequisites** | 1)Admin signed in.  2)Admin wants to search for a order in list order. |
| **Post-condition** | Admin finds a order on the order list. |
| **The main Event stream** | This function starts when the admin wants to search for a order in the order list.   * 1. Admin Selection item in order:                    Manage order -> search order. |
| **Secondary event Flow** | When there is no order like the input admin information, the message: “No Heart sees order” |

|  |  |
| --- | --- |
| **Use Case Name** | Update order |
| **Use Case ID** | UC4.2 |
| **Description** | The admin function update order. |
| **Actor** | Admin. |
| **Trigger** | When admin wants to update order status. |
| **Prerequisites** | 1)Admin signed in.        2)Admin wants to update order status. |
| **Post-condition** | Success: Admin updates order status success. |
| **The main Event stream** | This function starts when the admin wants to update order status  1. Admin Options item in order:                   Manage Order -> Update.  2. Admin change status of order  3. Admin confirm to change status. |
| **Exception** | Admin cancels update. |







|  |  |
| --- | --- |
| **Use Case Name** | View status |
| **Use Case ID** | UC5.1.1 |
| **Description** | Admin description function to view delivery status. |
| **Actor** | Admin. |
| **Triggers** | Admin wants to see delivery status. |
| **Preconditions** | 1. Admin is logged in. 2. Admin wants to see delivery status. |
| **Postconditions** | Admin can view the delivery status, including order receiving partner, shipper, delivery time. |
| **Main Course** | This function starts when the admin wants to see the delivery status.   * Admin selects items in order:               Manage shipping -> ship -> delivery status.              + After selecting the order in the above order will display a list of orders and delivery status of that order in the form of a table of columns: order code, partner receiving orders, area.   * If no orders are being delivered the system will display "There are currently no orders". |

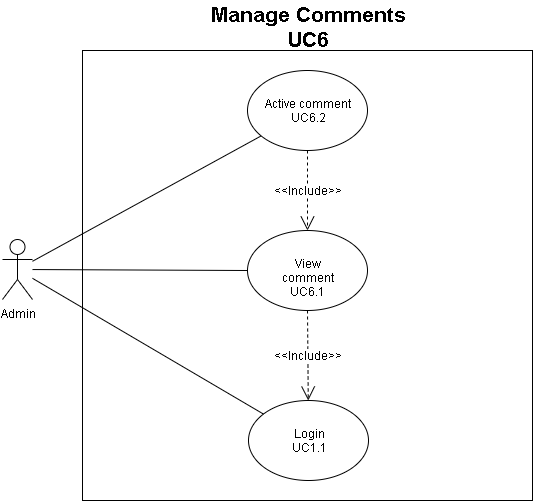
|  |  |
| --- | --- |
| **Use Case Name** | Checking ship |
| **Use Case ID** | UC5.1.2 |
| **Description** | Admin description function see checking checking ship function. |
| **Actor** | Admin. |
| **Triggers** | 1. Admin is logged in. 2. Admin wants to see checking ship function. |
| **Preconditions** | Admin want checking ship. |
| **Postconditions** | Admin can checking ship. |
| **Main Course** | * This function starts when the admin wants to checking ship. * Admin selects items in order:   Manage shipping -> ship -> checking ship.   * After selecting in the above order, the list of order codes will be displayed in the form of a table consisting of columns: Order code, order status. * Admin can choose to change the status of the order and then click the confirm button to change the status of the order. |

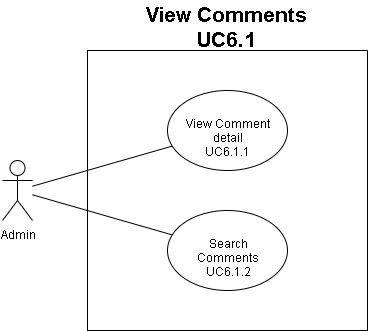
|  |  |
| --- | --- |
| **Use Case Name** | Update-checking |
| **Use Case ID** | UC\_5.1.3 |
| **Description** | Function for admin to update delivery status. |
| **Actor** | Admin. |
| **Triggers** | When the admin needs to update delivery status. |
| **Preconditions** | 1. Admin is logged in. 2. Admin needs to update delivery status. |
| **Postconditions** | Admin update delivery status. |
| **Main Course** | This function starts when the admin needs to update delivery status.   * 1. Admin selects items in order:                   Manage shipping -> checking ship -> update.   * 2. Admin change delivery status. * 3. Admin confirms to change delivery status or not. |
| **Alternate Courses** | 1. Admin cancel confirmation of updating partner information. |

|  |  |
| --- | --- |
| **Use Case Name** | View partners |
| **Use Case ID** | UC5.2.1 |
| **Description** | Admin description function to view partners. |
| **Actor** | Admin. |
| **Triggers** | 1. Admin is logged in. 2. Admin wants to see the partners. |
| **Preconditions** | Admin wants to see the partners. |
| **Postconditions** | Admin can view the list of partners. |
| **Main Course** | This function start when the admin want to see the partners.  Admin selects items in order:              Manage shipping -> delivery partner.              + after selecting in the above order, a list of partners will be displayed in the form of tables including columns: Partner name, email, phone number, address, shipping cost.  If there is no partner, the system will display "Currently there are no partners". |

|  |  |
| --- | --- |
| **Use Case Name** | Add partner |
| **Use Case ID** | UC5.2.2 |
| **Description** | Function for admin to add a new partner. |
| **Actor** | Admin. |
| **Triggers** | When the admin wants to add a new partner to the system. |
| **Preconditions** | 1. Admin is logged in. 2. Admin wants to add a new partner to the system. |
| **Postconditions** | Success: Admin added a new partner to the system.  Failure: Unable to add because the partner already exists in the system. |
| **Main Course** | This function start when the admin want to add partners to the system.   * 1. Admin selects items in order:                   Manage shipping -> delivery partner -> add.   * 2. Admin fill in information of new partner: partner's name, email, address, phone number. * 3. Admin confirms you want to add a new partner or not. |
| **Alternate Courses** | 1. When the Admin does not enter information in one or the required fields, the system will display a request to enter information in those items. 2. Admin has filled in information but added failure  * The system will display the message "Add partner failed" with the same reason for example: partner already exists.  1. Admin cancels further confirmation. |

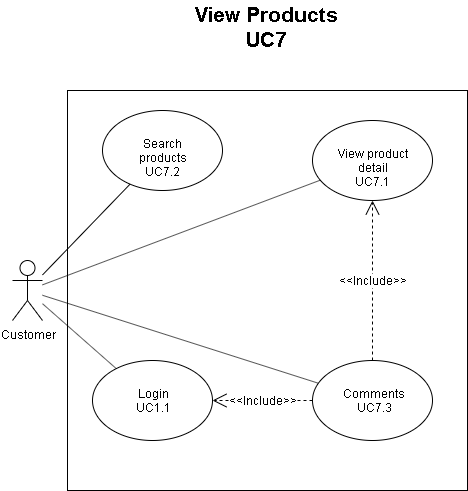
|  |  |
| --- | --- |
| **Use Case Name** | Update partner |
| **Use Case ID** | UC5.2.3 |
| **Description** | Function for admin to update information of a partner at the request of that partner. |
| **Actor** | Admin. |
| **Triggers** | When the admin needs to update the information of the partner at the request of the partner. |
| **Preconditions** | 1. Admin is logged in. 2. Admin needs to update partner information at the request of partner. |
| **Postconditions** | Admin updates new information of partners into the system. |
| **Main Course** | This function start when the admin need to update the partner's new information to the system.   * 1. Admin selects items in order:                   Manage shipping -> delivery partner -> update.   * 2. Admin fill in the information partners want to change eg partner name, email, address, phone number. * 3. Admin confirms to update new information of partners or not. |
| **Alternate Courses** | 1. When the Admin leaves blank information for one or more items, the system will display a request to enter information in those items. 2. Admin cancel confirmation of updating partner information. |





|  |  |
| --- | --- |
| **Use case name** | View Comment |
| **Use case ID** | UC6.1 |
| **Description** | Admin view comment of product |
| **Actor** | Admin |
| **Organizational**  **Benefits** | Admin can view comment of products |
| **Triggers** | Manager login -> see dashboard page -> view comment |
| **Preconditions** | Admin logged |
| **Postconditions** | Admin accesses dashboard page |
| **Main Course** | 1. Login Dashboard page  2. Choose “view comment”  3. Display list product have comment  4.Choosing a product to view comment of this product |

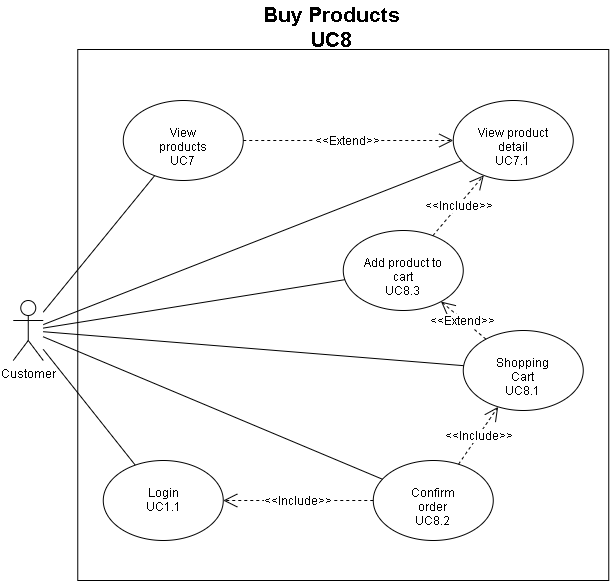
|  |  |
| --- | --- |
| **Use case name** | Active Comment |
| **Use case ID** | UC6.2 |
| **Description** | Admin active comment of customer |
| **Actor** | Admin |
| **Organizational**  **Benefits** | Admin active comment of customer |
| **Triggers** | view comment -> active comment |
| **Preconditions** | Admin logged |
| **Postconditions** | Admin active comment to comment of customer display |
| **Main Course** | 1.View comment of product  2. Display list comment of this product  3.Active comment |

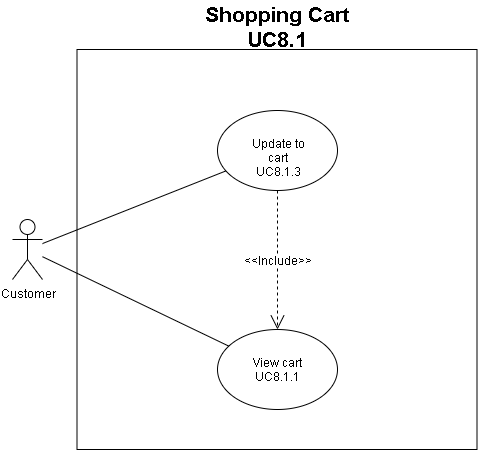


|  |  |
| --- | --- |
| **Use case name** | View product detail |
| **Use case ID** | UC7.1 |
| **Description** | Customers access the Website to view products |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Helping customers can  view the products according to customers' wishes |
| **Triggers** | Customers access Website-> view the products you want to see |
| **Preconditions** | The customer has accessed on the Website of Shop |
| **Postconditions** | Customers have viewed the products they want |
| **Main Course** | 1.View a product details  1.1 Customers access the website of VTCA Clothing Store  1.2 Select a product you want to view to see details |

|  |  |
| --- | --- |
| **Use case name** | Search product |
| **Use case ID** | UC7.2 |
| **Description** | Customers access the Website to search products |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Helping customers can search the products according to customers' wishes |
| **Triggers** | Customers access Website->Choose search bar/ a category -> search the products you want to see |
| **Preconditions** | The customer has accessed on the Website of Shop |
| **Postconditions** | Customers have viewed the products they want |
| **Main Course** | 1. Customers access the website of VTCA Clothing Store  2.Select the search bar on the VTCA Clothing Store website  3. Enter the product name you want to search  4. Display the number of results |

|  |  |
| --- | --- |
| **Use case name** | Comments |
| **Use case ID** | UC7.3 |
| **Description** | Customer comment of product |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer can comment a products |
| **Triggers** | See product detail -> view comment of a product -> fill content |
| **Preconditions** | Customer logged |
| **Postconditions** | Customer commented |
| **Main Course** | 1. See product detail first  2. Choose “comment bar”  3. Fill content |





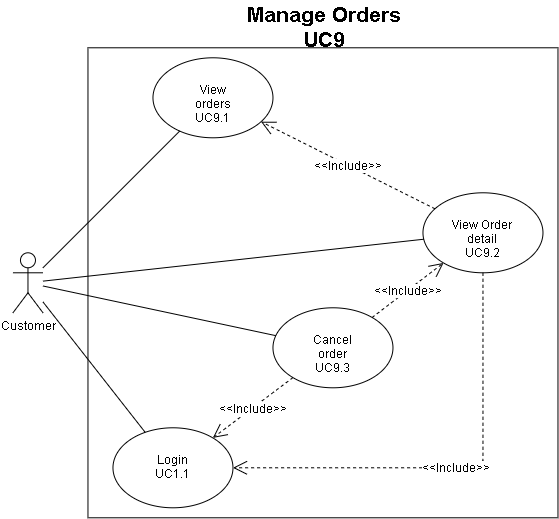
|  |  |
| --- | --- |
| **Use case name** | Add to cart |
| **Use case ID** | UC8.3 |
| **Description** | Customer add product to cart |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer can add product to cart |
| **Triggers** | Customer access website -> logged in website ->search product ->choose a product -> add to cart |
| **Preconditions** | Customer logged in website of shop |
| **Postconditions** | Customer add product to cart |
| **Main Course** | 1. Customer access website of shop  2. Customer view products and product detail(UC07)  3.Choose size and quantity  4.Choose add to cart to add product to cart |

|  |  |
| --- | --- |
| **Use case name** | View Cart |
| **Use case ID** | UC8.1.1 |
| **Description** | Customer want to view the shopping cart |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer click on the shopping cart icon. |
| **Triggers** | 1. Customer has logged in. 2. Customer added product to cart |
| **Preconditions** | Customer logged in website of shop |
| **Postconditions** | Customer add product to cart |
| **Main Course** | This function start when customer click on the shopping cart icon.   * Customer will see the product that they have added to the cart and the order details of those product. For example: quantity, size, price… * If the customer enters the shopping cart when there is no product in the cart, there will be a notification that the cart is empty and there will be a button to suggest a continue shopping. |

|  |  |
| --- | --- |
| **Use case name** | Update to cart |
| **Use case ID** | UC8.1.3 |
| **Description** | Customer update quantity of this product |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer can update quantity of product |
| **Triggers** | Customer access website -> view cart->input quantity->choose update |
| **Preconditions** | Customer added product in cart |
| **Postconditions** | Customer updated quantity of product |
| **Main Course** | 1. Customer access website of shop  2. Choose cart  3. Input quantity of product want to update  4.Choose update to update quantity of product in cart |

|  |  |
| --- | --- |
| **Use case name** | Delete a product in cart |
| **Use case ID** | UC8.1.4 |
| **Description** | Customer delete a product in cart |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer can delete a  product in cart |
| **Triggers** | Customer access website -> view cart->choose delete a product/all product |
| **Preconditions** | Customer added product in cart |
| **Postconditions** | Customer deleted a product in their cart |
| **Main Course** | 1. Customer access website of shop  2. Choose cart  3.Choose “X”/ “Xóa tất cả” to delete a  product in cart |

|  |  |
| --- | --- |
| **Use case name** | Confirm order |
| **Use case ID** | UC8.2 |
| **Description** | Customer confirm their order |
| **Actor** | Customer |
| **Organizational**  **Benefits** | Customer can confirm order |
| **Triggers** | Customer view cart -> choose “Process to payment” -> Choose “confirm order” |
| **Preconditions** | Customer accessed website of shop  Cart has product  Customer logged |
| **Postconditions** | Customer confirm order |
| **Main Course** | 1.View cart  2. Make a payment (if the user selects "Proceed to checkout")  3.Display information of customer and order  4.Choose “Confirm order” to order |

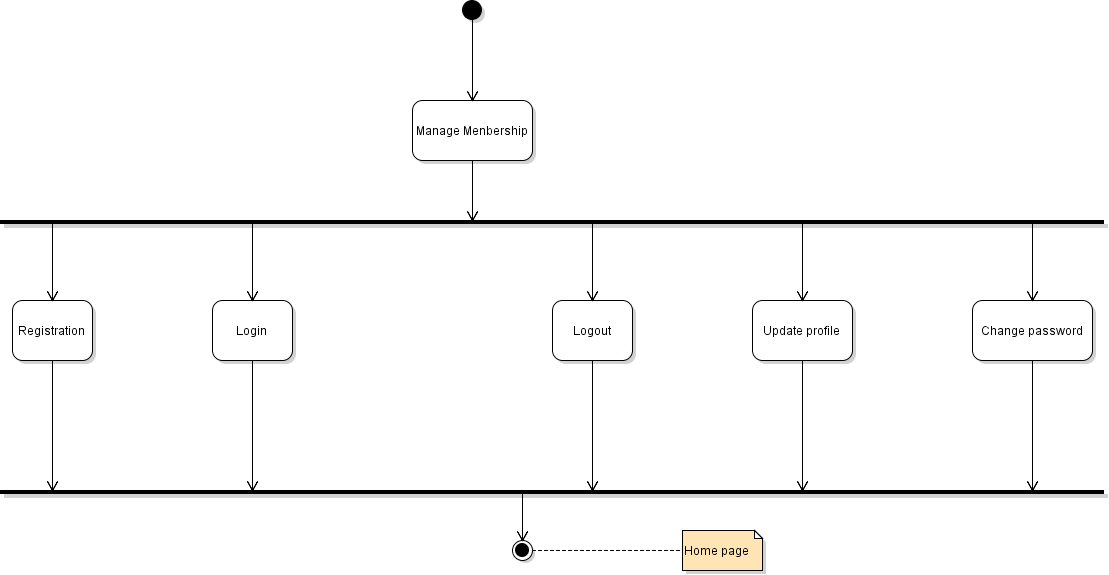


|  |  |
| --- | --- |
| **Use Case Name** | View Orders |
| **Use Case ID** | UC\_9.1 |
| **Description** | The function allows users to view their own transaction history. |
| **Actor** | Customer. |
| **Triggers** | When customers choose to see the transaction history. |
| **Preconditions** | The customer has logged in. |
| **Postconditions** | Customers can view transaction history. |
| **Main Course** | This function starts when the customer wants to see the transaction history.   * Customers choose to account themselves -> Transaction history.   + after selecting the order in the order above will display a list of transaction history.  If the customer has not purchased any items then the system will display "No transaction history". |

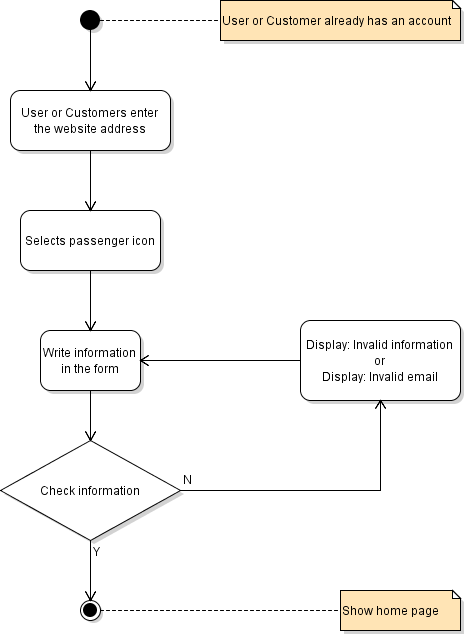
|  |  |
| --- | --- |
| **Use Case Name** | View Order Detail |
| **Use Case ID** | UC\_9.2 |
| **Description** | The function allows users to view transaction details of themselves. |
| **Actor** | Customer. |
| **Triggers** | When customers choose to see transaction details. |
| **Preconditions** | 1. The customer has logged in. 2. The customer has made a purchase. |
| **Postconditions** | Customers can view transaction details. |
| **Main Course** | This function starts when customers want to see transaction details.   * Customers choose to account themselves -> Transaction history -> Select any transaction history.   + after selecting the order in the order above will display the details of that transaction history. |

## Activity Diagram:

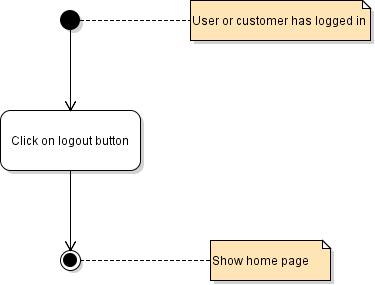
* Activity diagram: Manage Membership



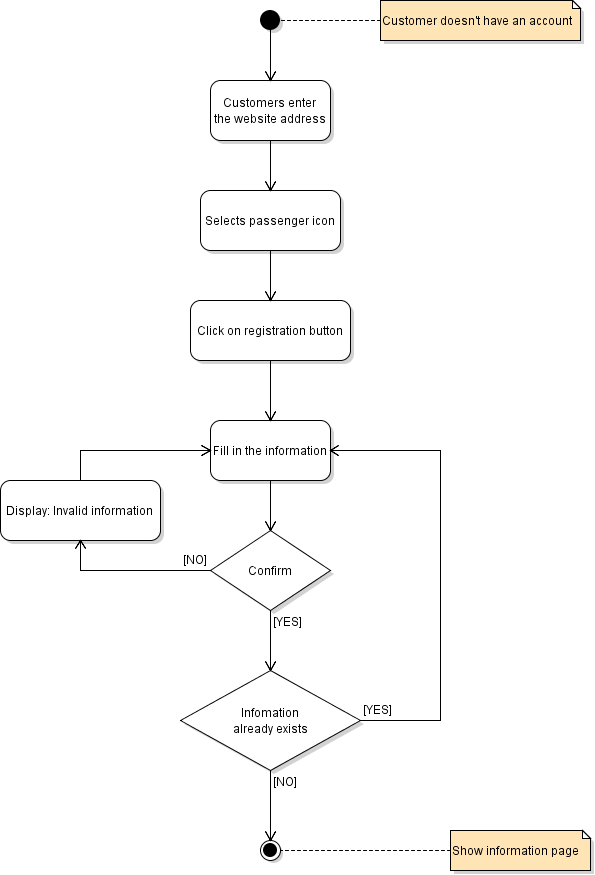
* Activity diagram: Login



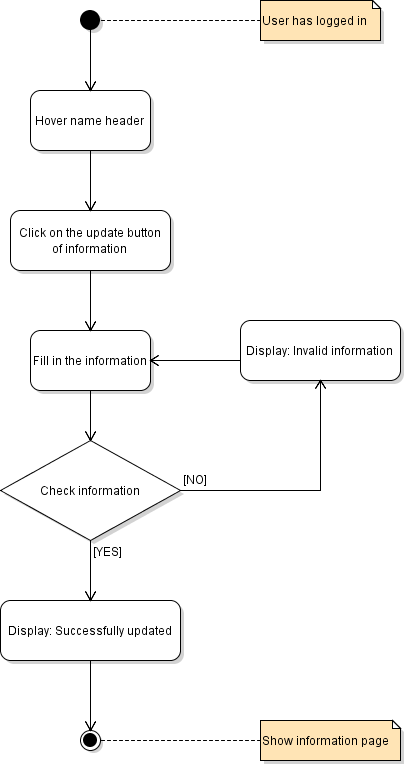
* Activity diagram: Logout



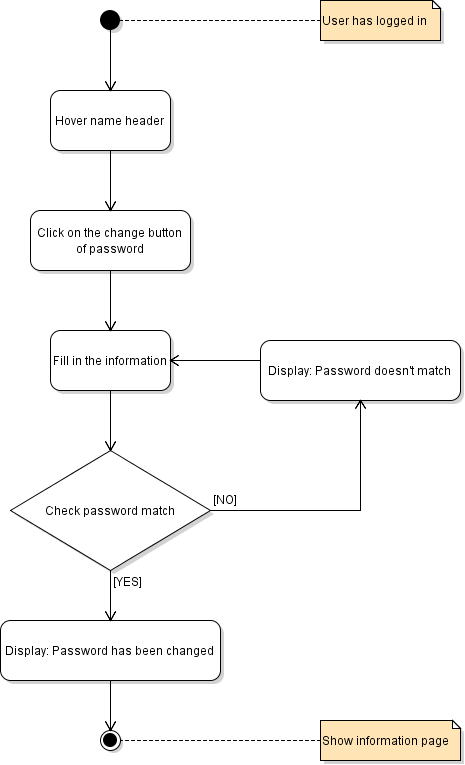
* Activity diagram: Registration



* Activity diagram: Update profile



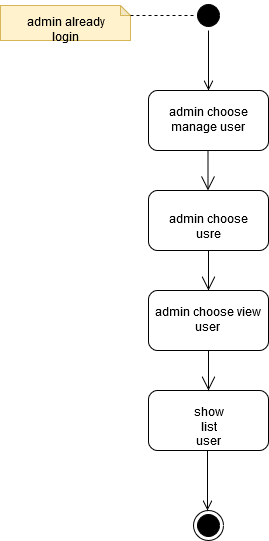
* Activity diagram: Change password



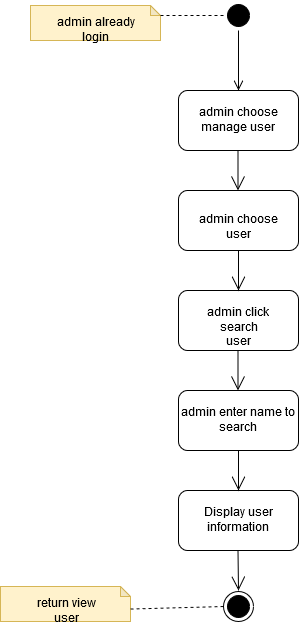
* Activity diagram: Manager user



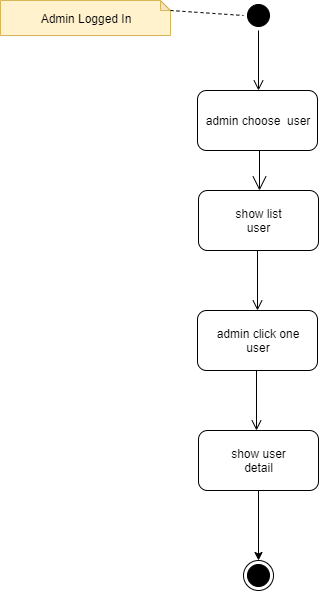
* Activity diagram: View user



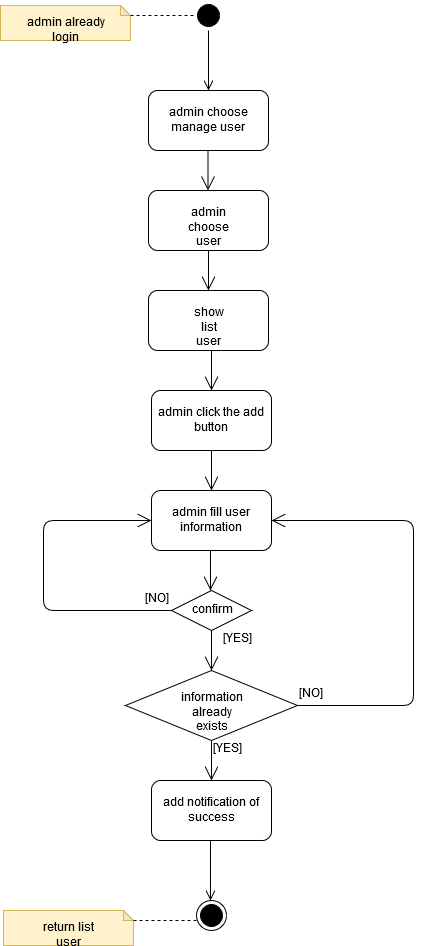
* Activity diagram: Search user



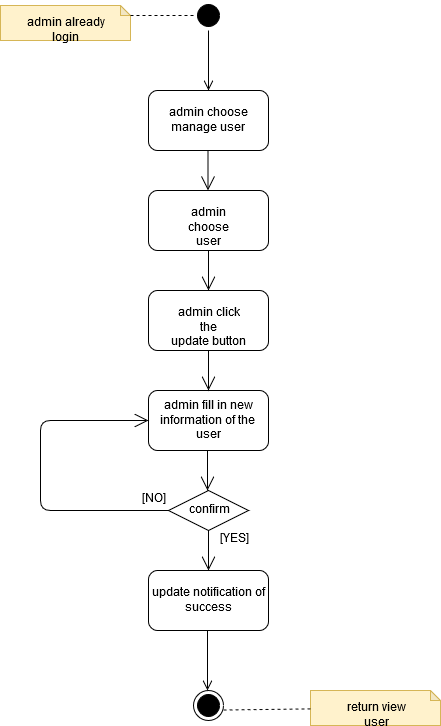
* Activity diagram: Show user detail



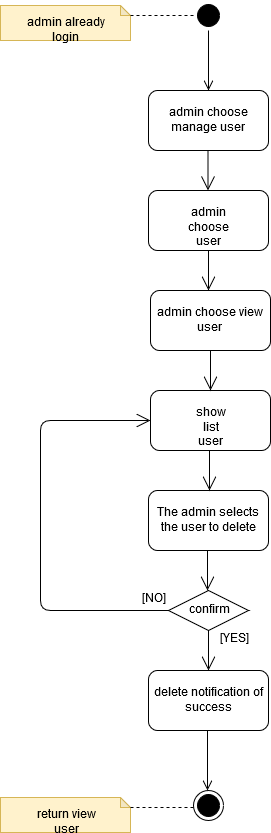
* Activity diagram: Add user



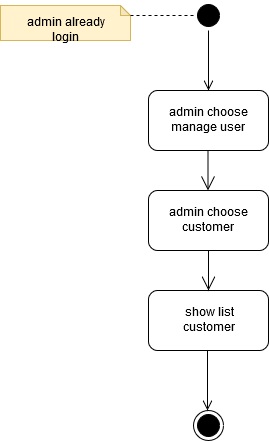
* Activity diagram: Update user



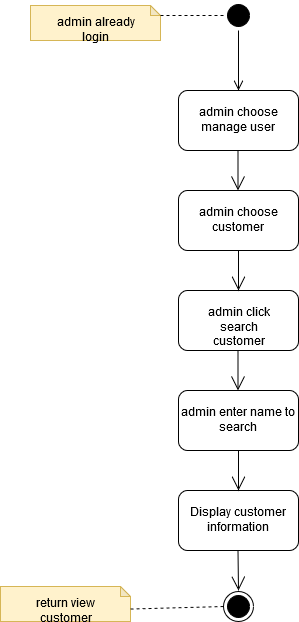
* Activity diagram: Delete user



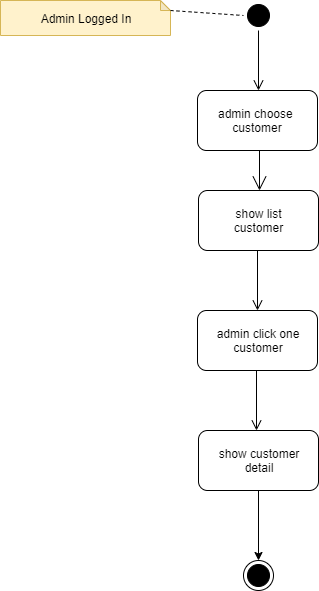
* Activity diagram: View customer



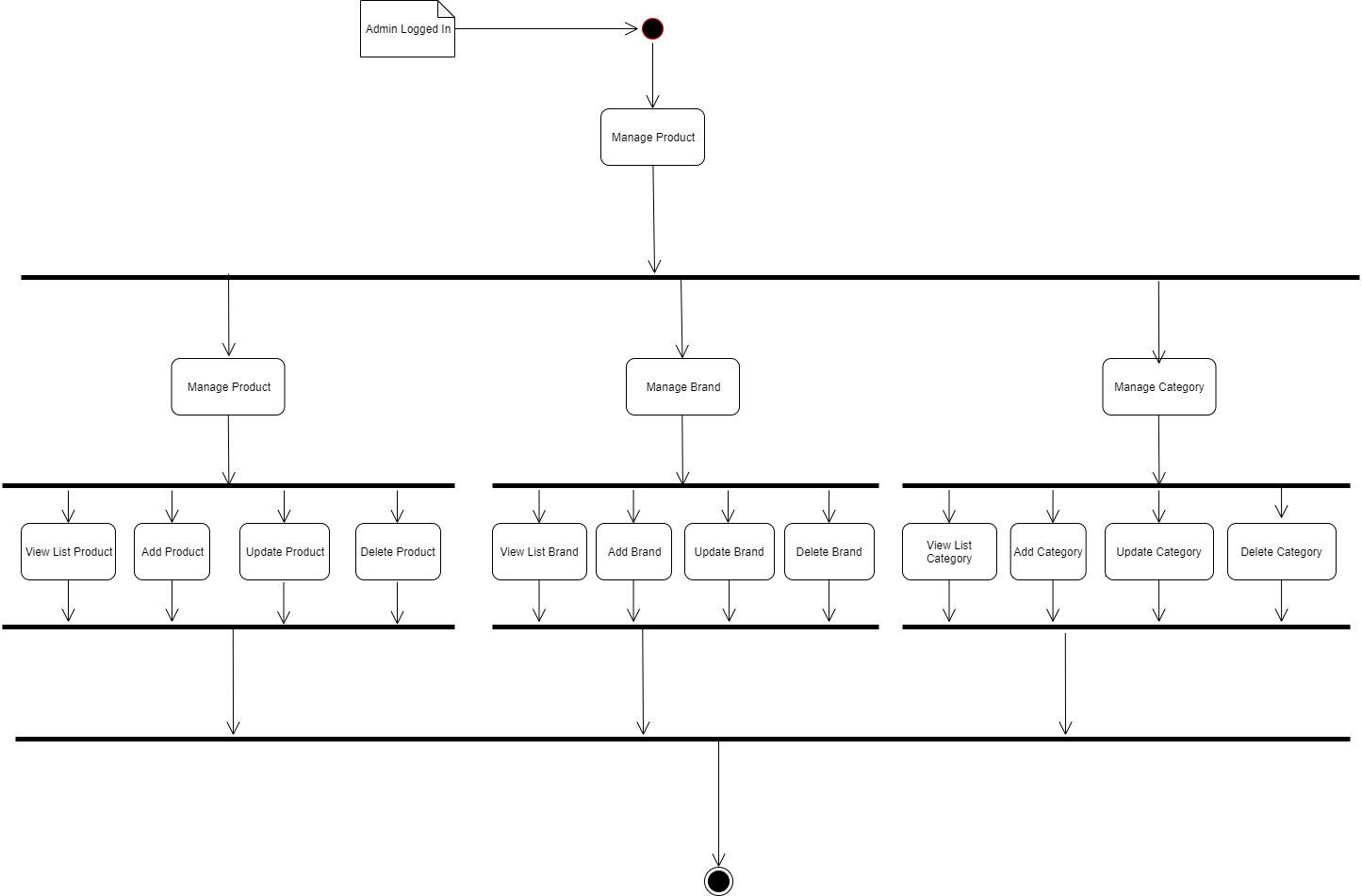
* Activity diagram: Search customer



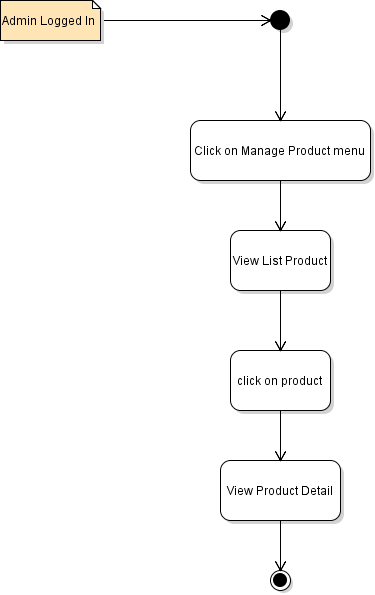
* Activity diagram: Show customer detail



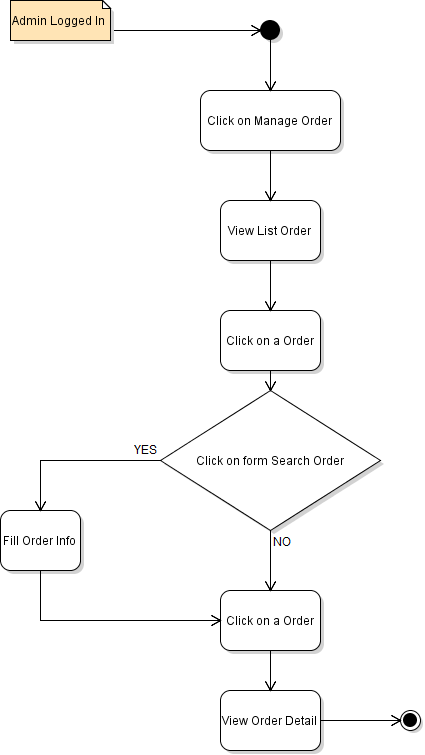
* Activity diagram: Manage product



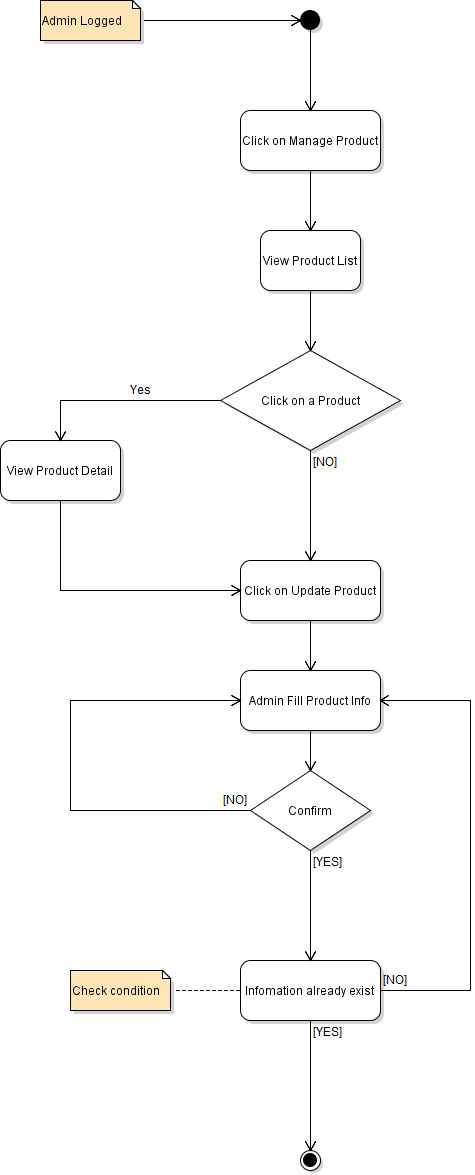
* Activity diagram: View product detail



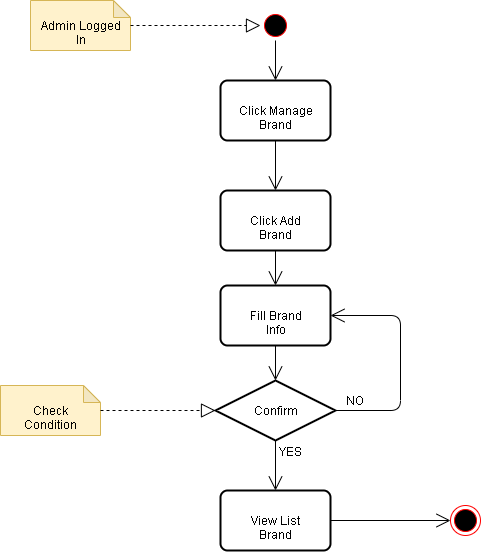
* Activity diagram: Search products



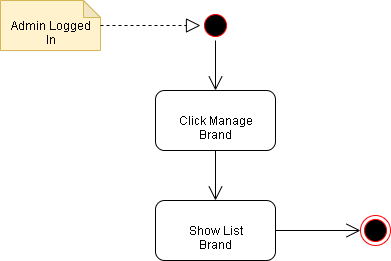
* Activity diagram: Update a product



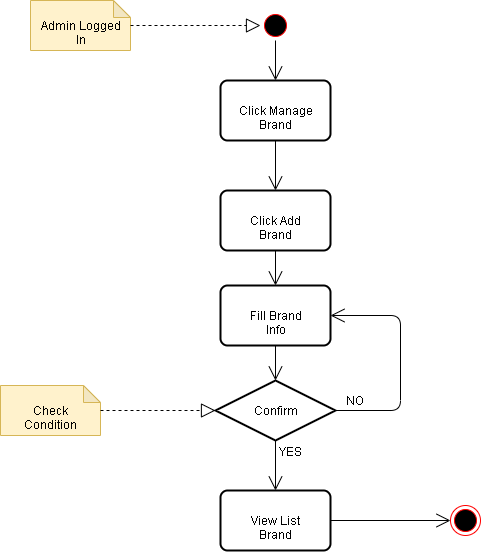
* Activity diagram: Add brand



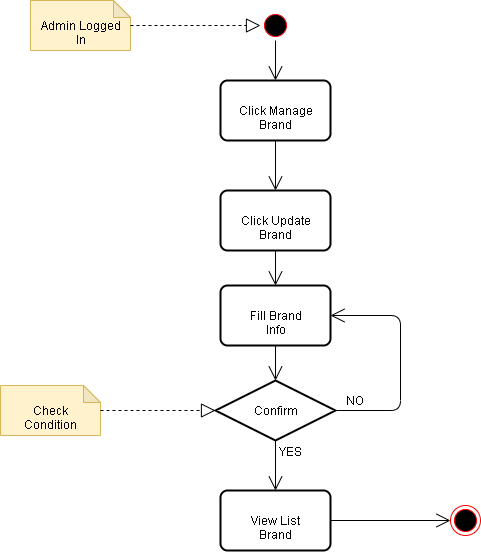
* Activity diagram: View brand



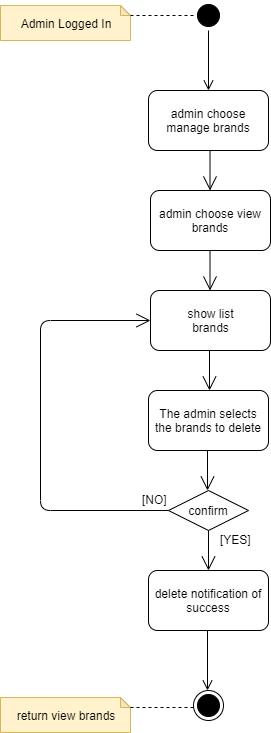
* Activity diagram: Add brand



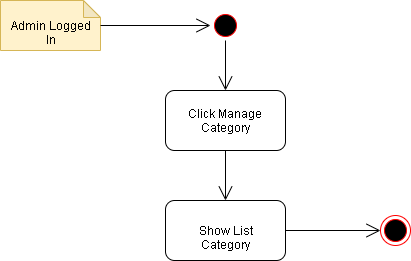
* Activity diagram: Update brand



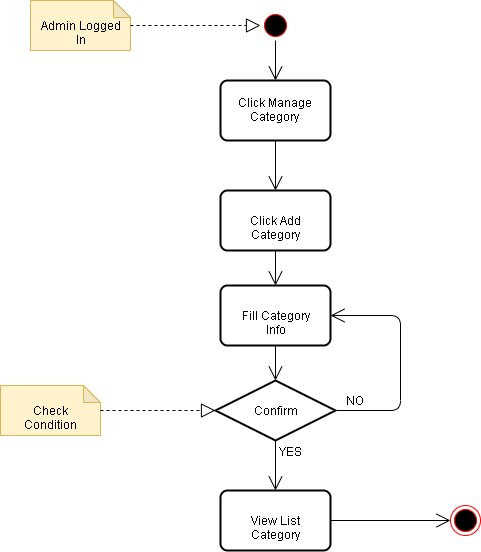
* Activity diagram: View categories



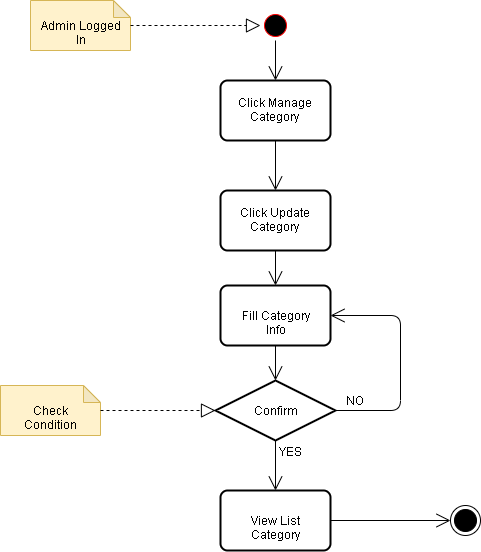
* Activity diagram: View categories



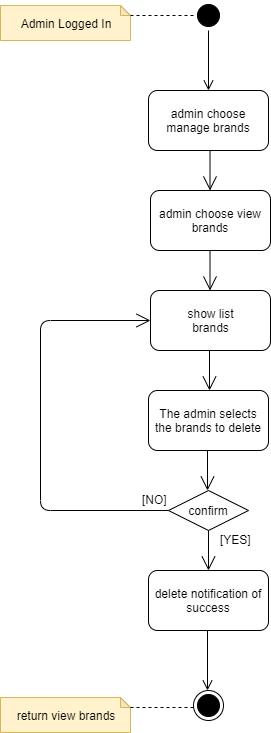
* Activity diagram: Add category



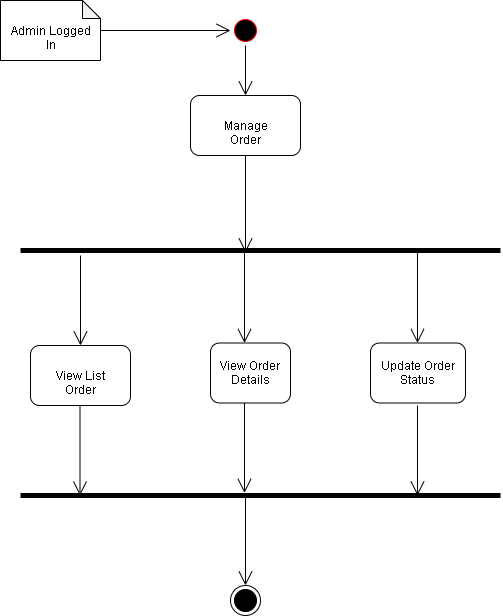
* Activity diagram: Update category



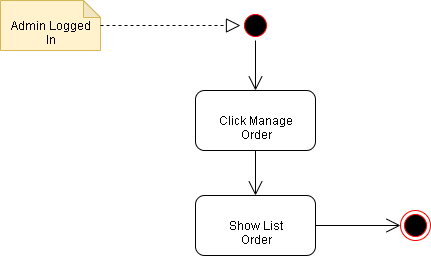
* Activity diagram: Delete category



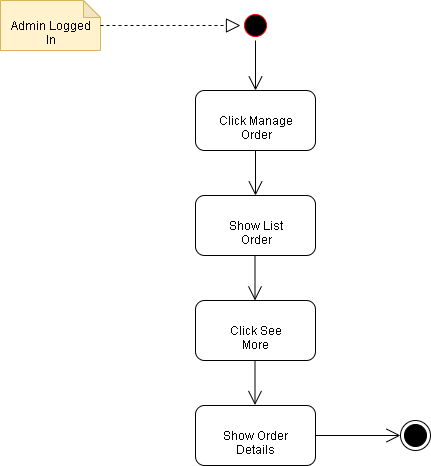
* Activity diagram: Manage order



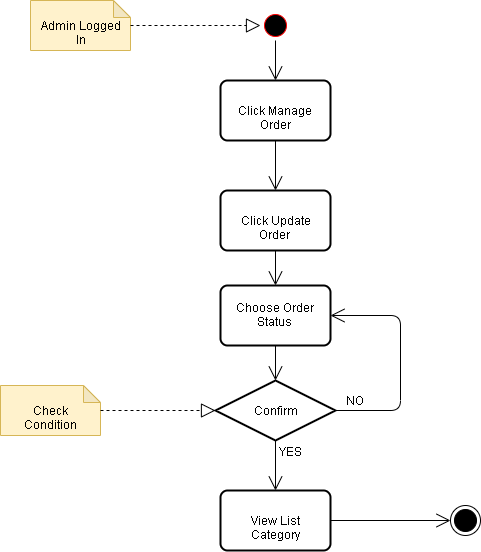
* Activity diagram: View order



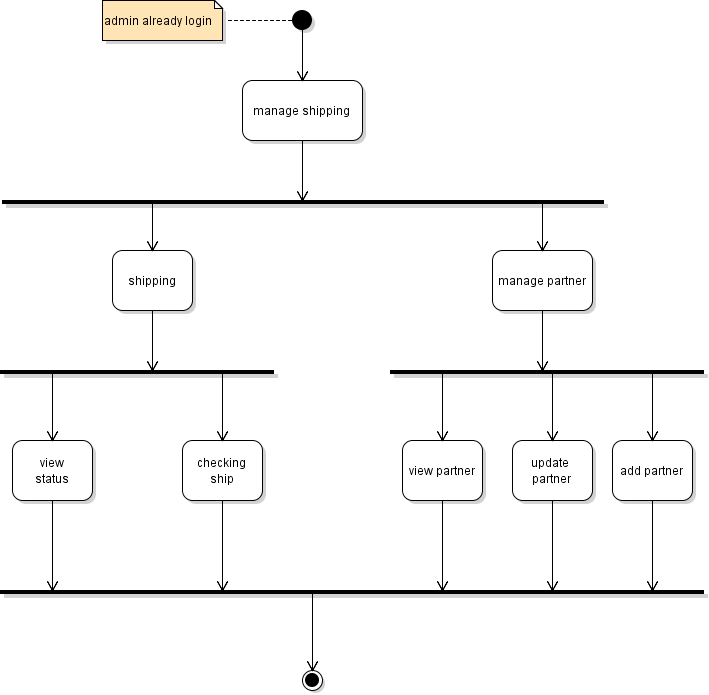
* Activity diagram: View order details



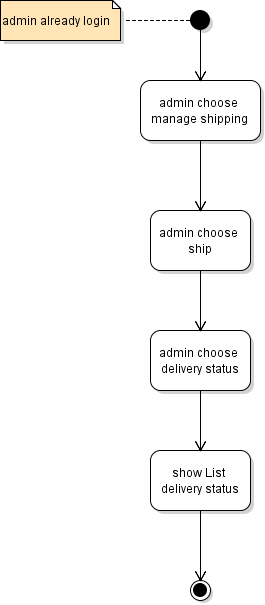
* Activity diagram: Update order status



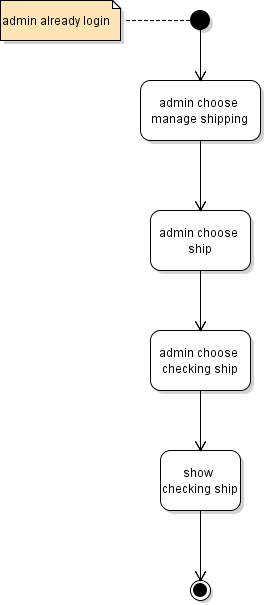
* Activity diagram: Manage Shipping



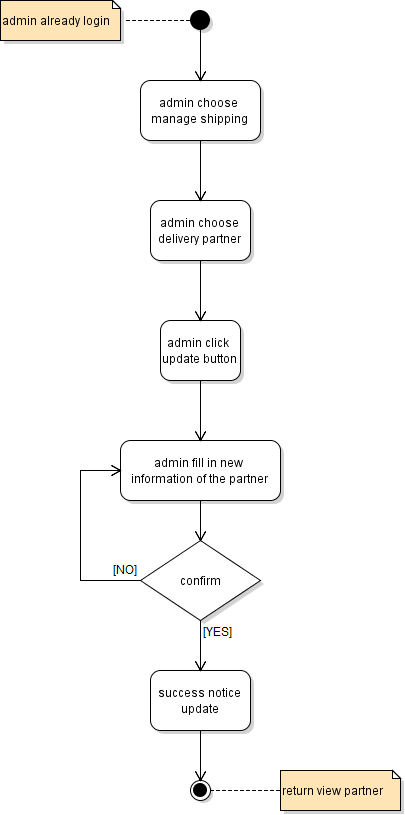
* Activity diagram: View status



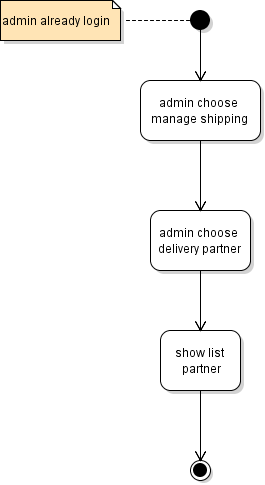
* Activity diagram: Checking ship



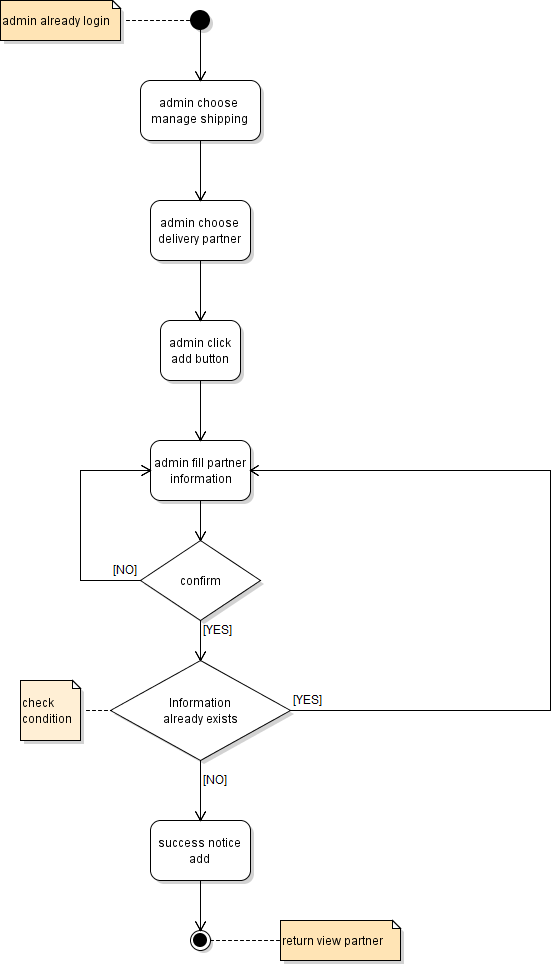
* Activity diagram: View partners



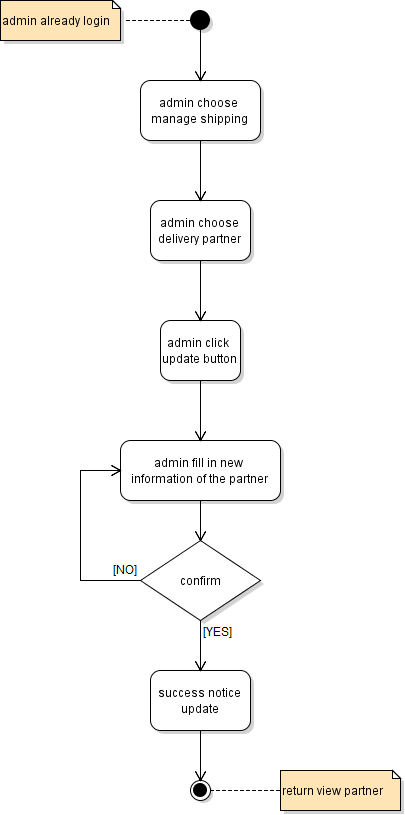
* Activity diagram: View partners



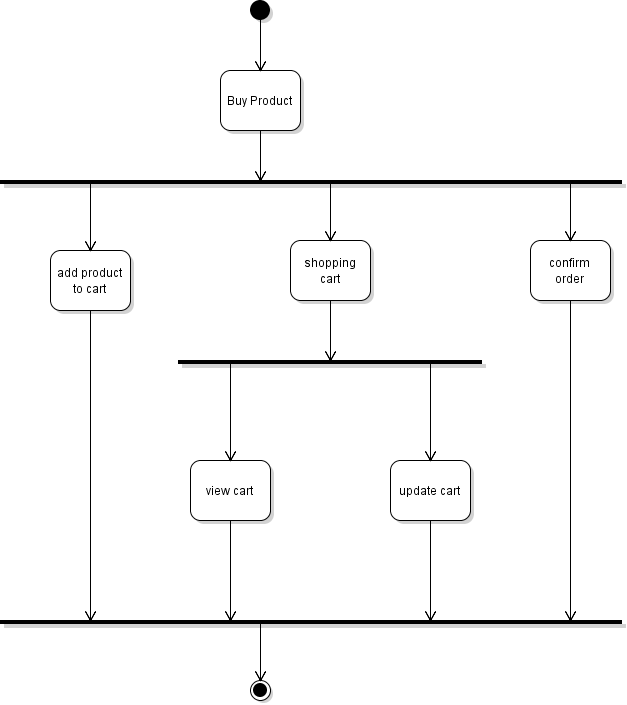
* Activity diagram: Add partner



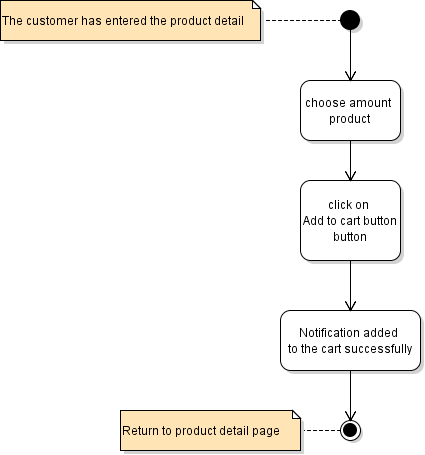
* Activity diagram: Update partner



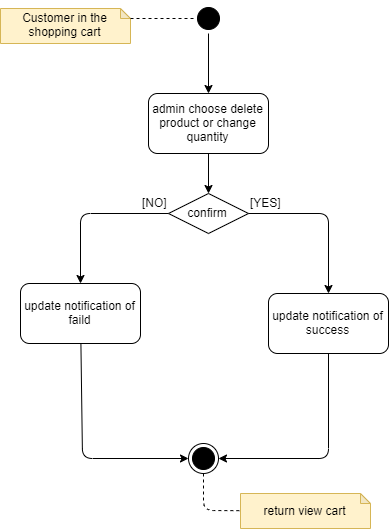
* Activity diagram: Buy product



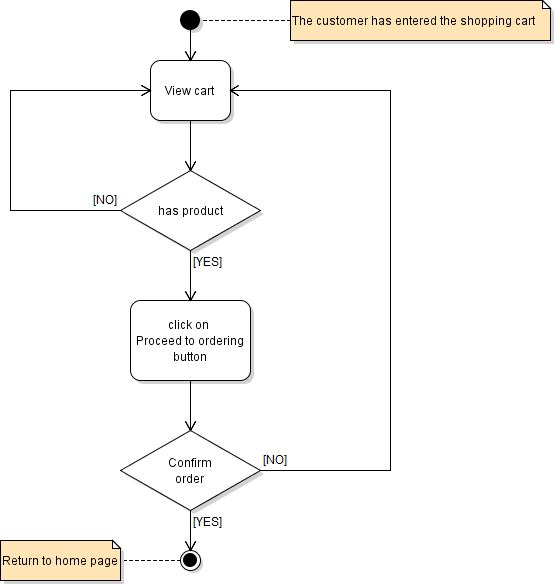
* Activity diagram: View cart
* Activity diagram: Add product to cart



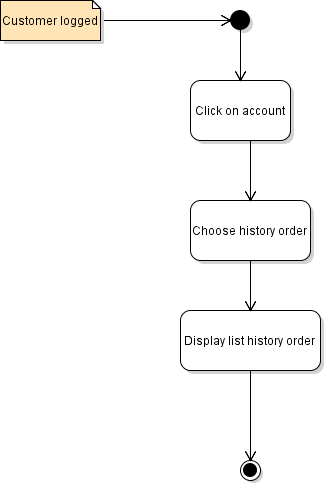
* Activity diagram: Update to cart



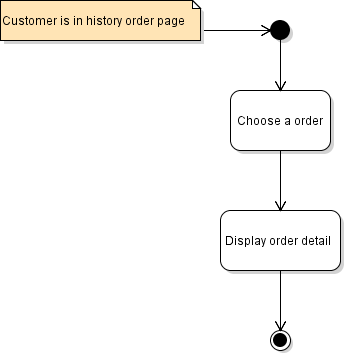
* Activity diagram: Confirm order



* Activity diagram: View orders

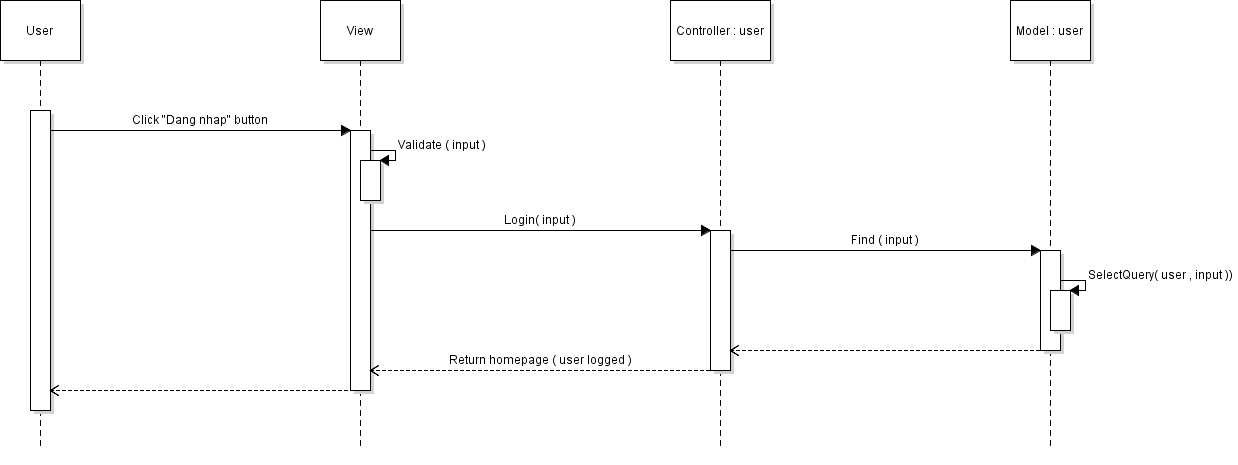


* Activity diagram: View order details

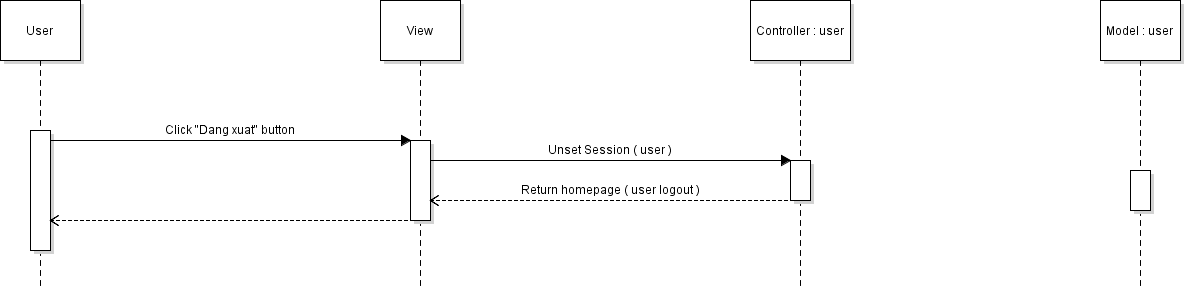


## Sequence Diagram:

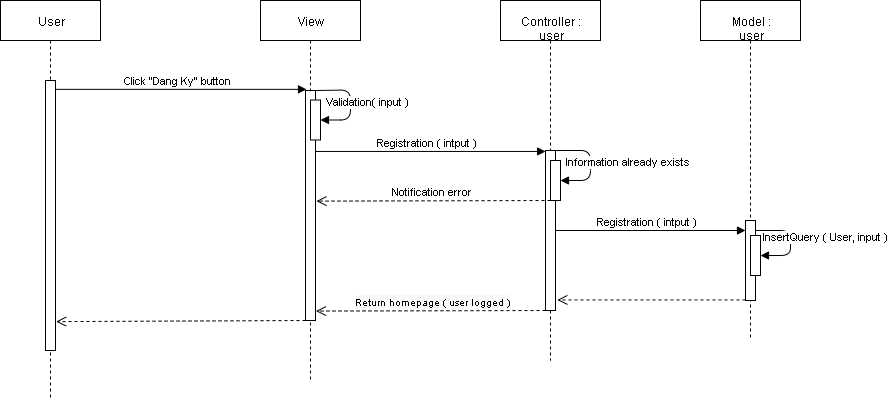
* Login:



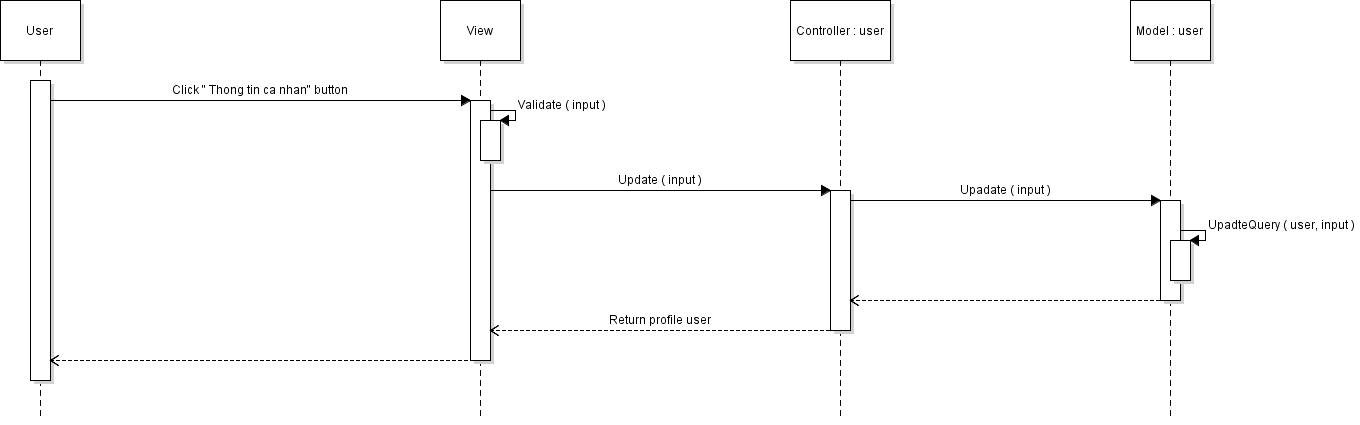
* Logout:



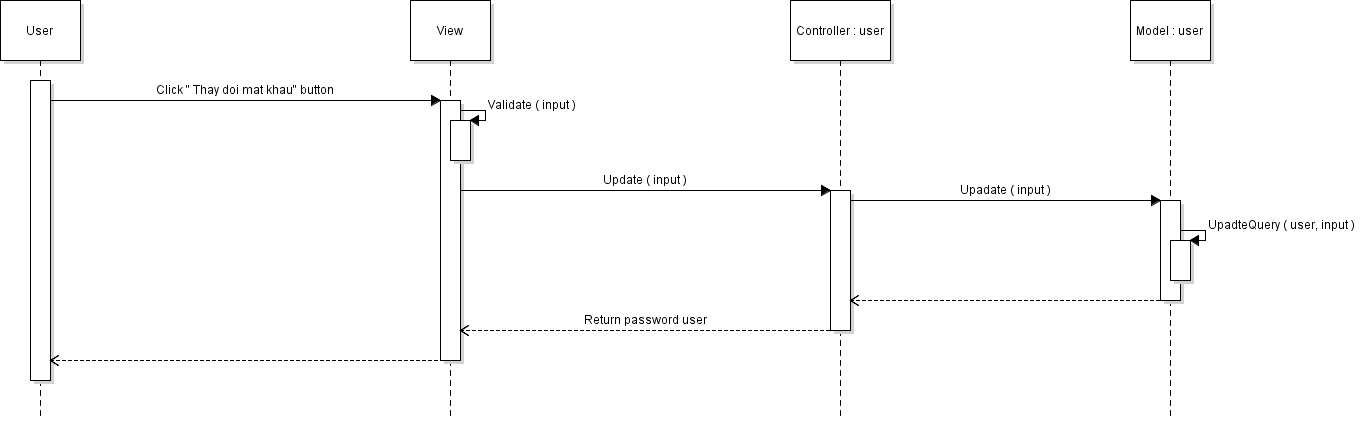
* Registrasion:



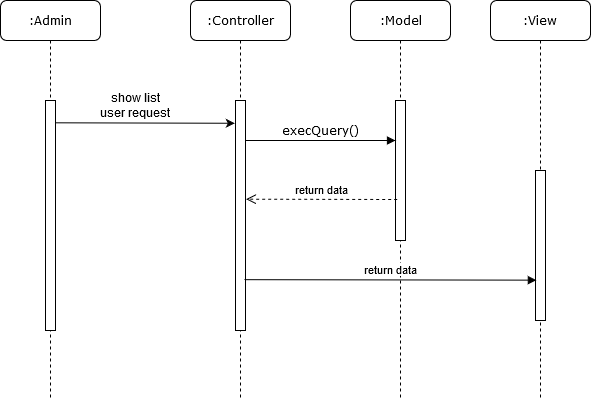
* Update profile:



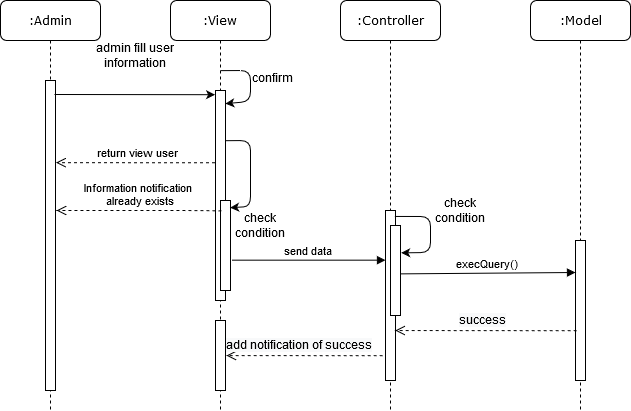
* Change password:



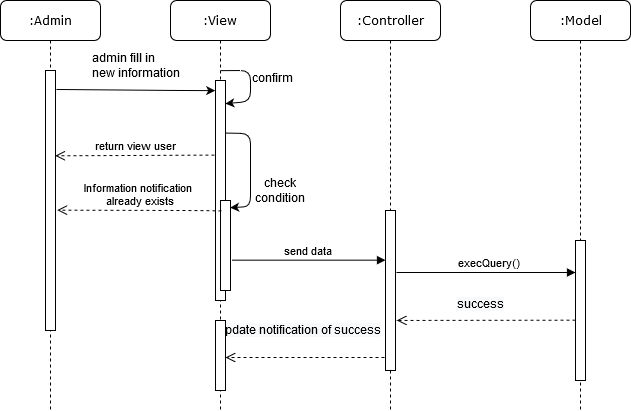
* View user:



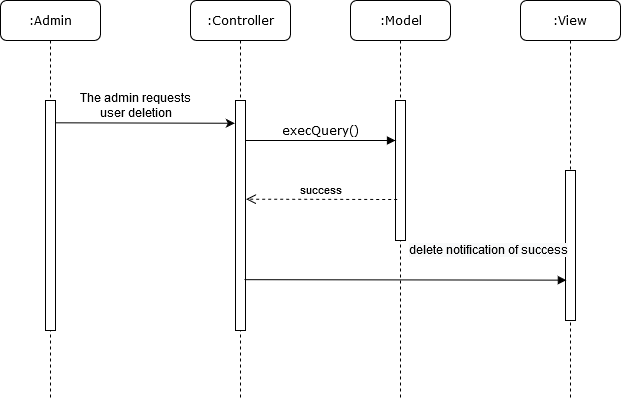
* Add user:



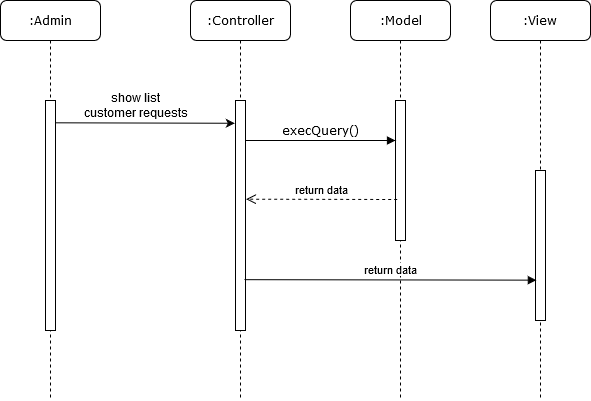
* Update user:



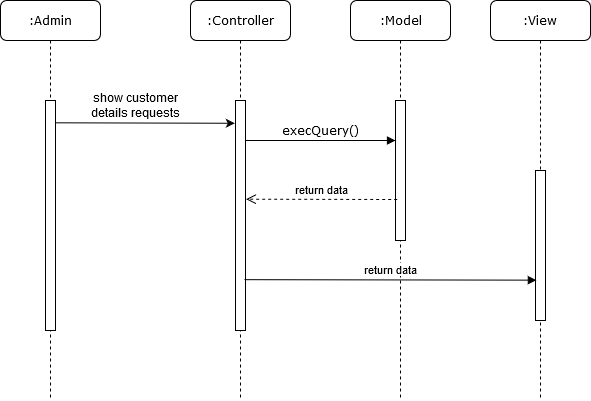
* Delete user:



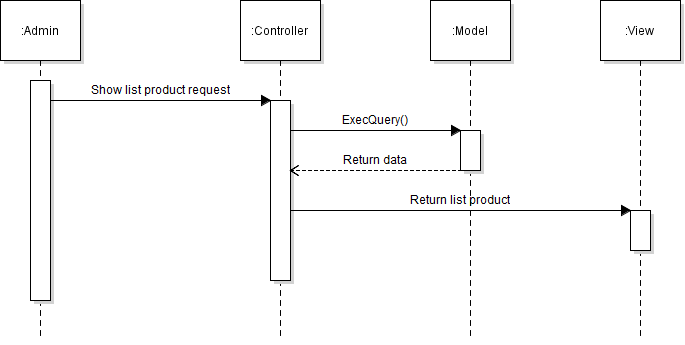
* Show customer:



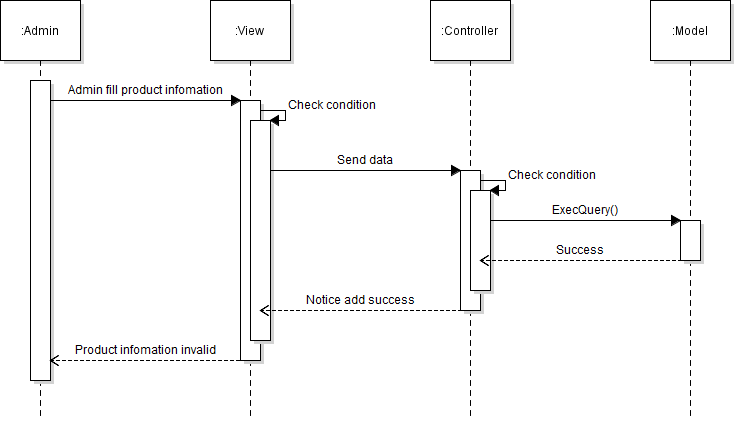
* Show customer detail:



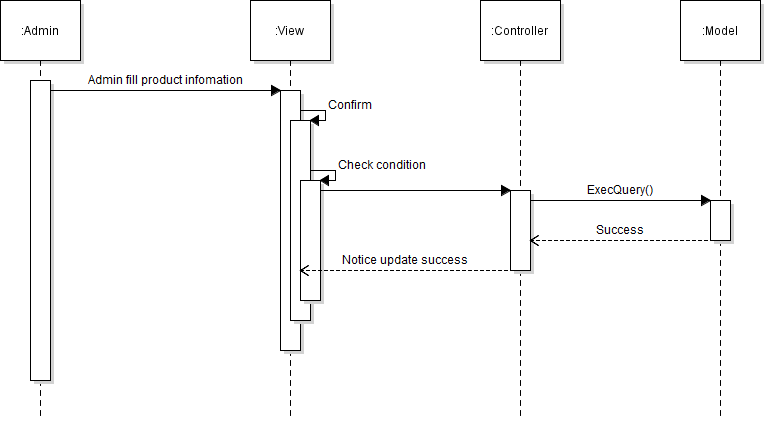
* View products :



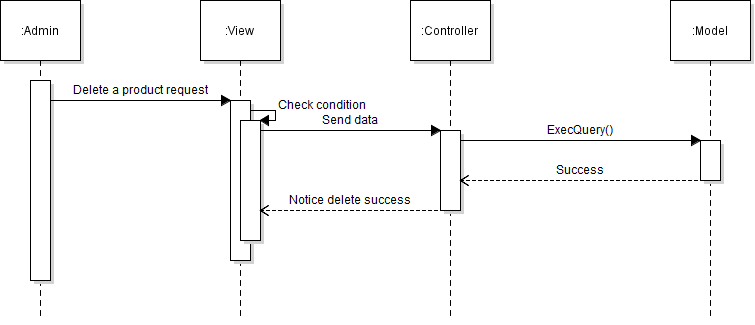
* Add product:



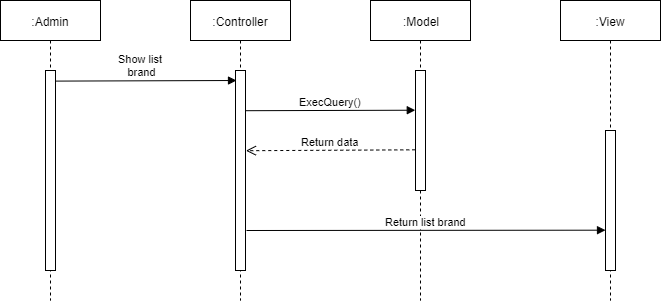
* Update product:



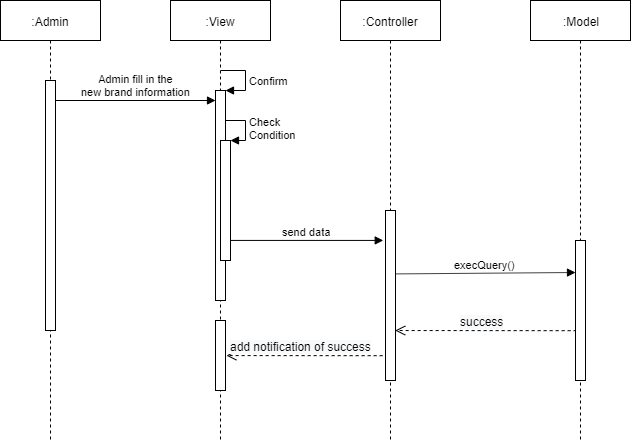
* Delete product:



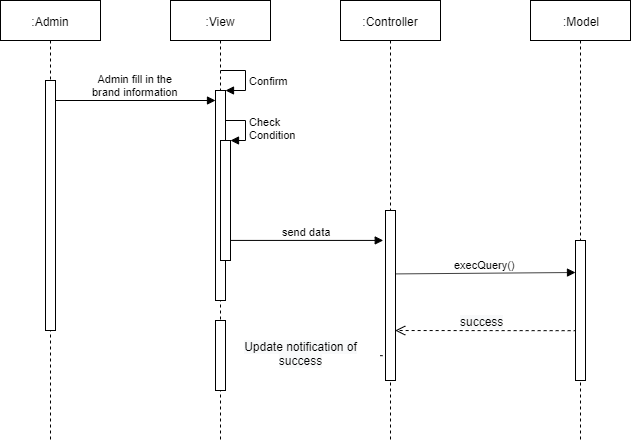
* View brand:



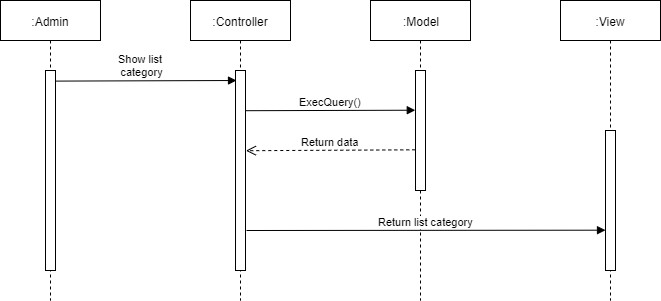
* Add brand:



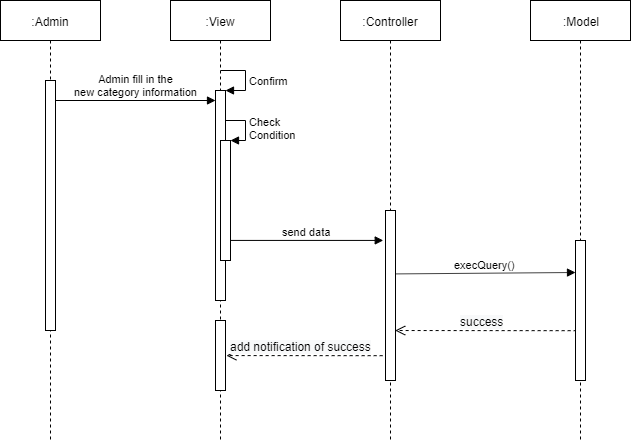
* Update brand:



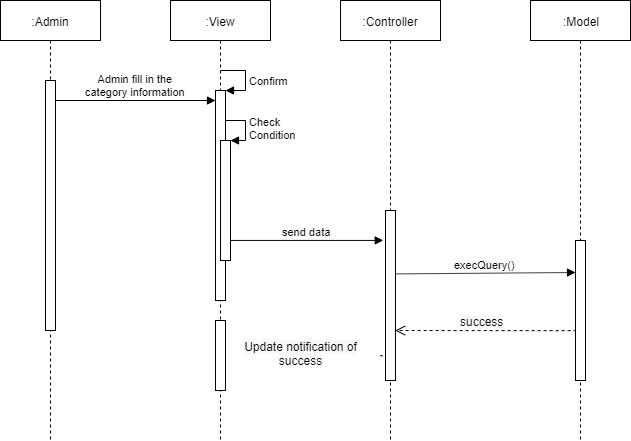
* View categories:



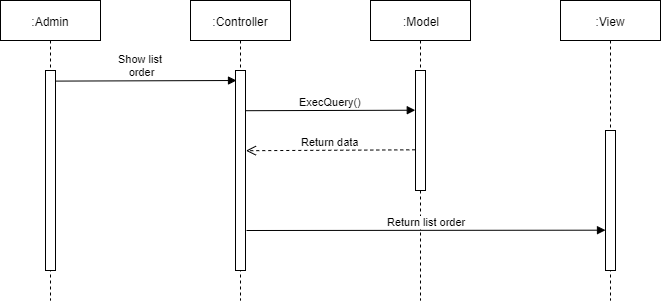
* Add category:



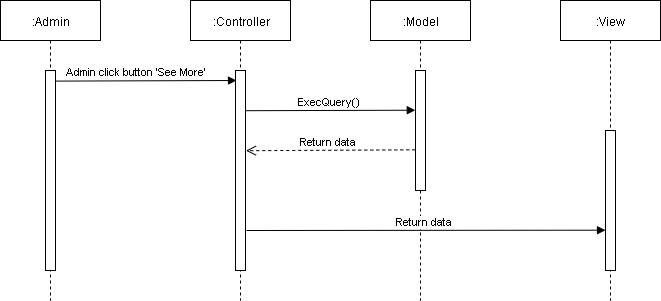
* Update category:



* View orders:



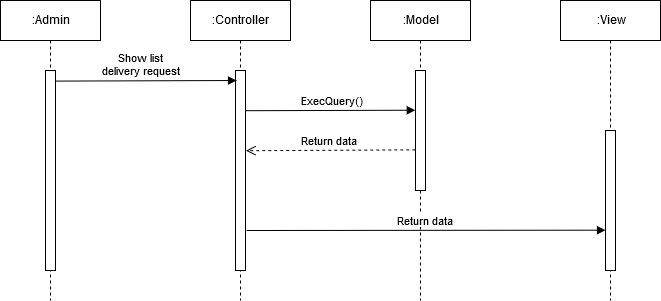
* View order details:



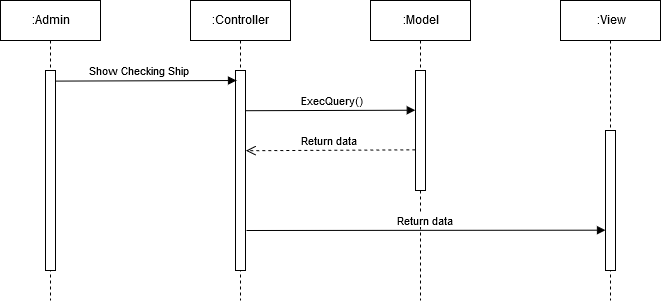
* Update order:



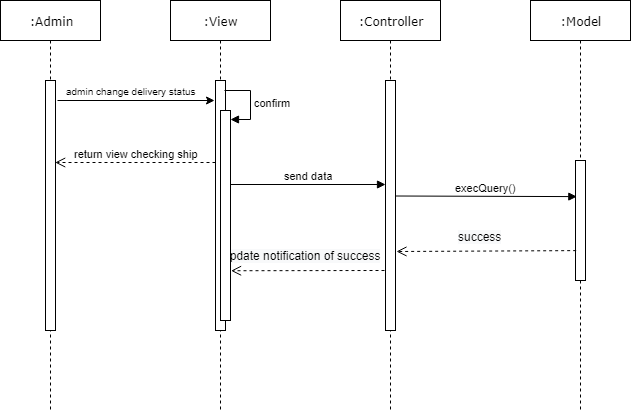
* View status:



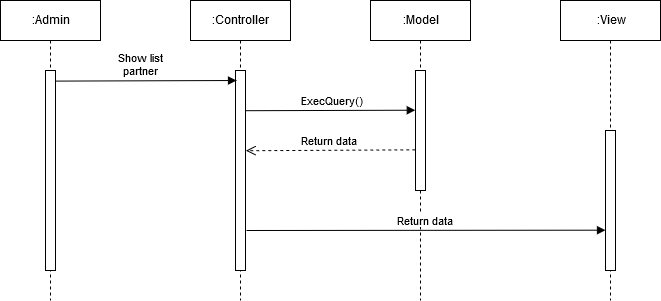
* Checking ship:



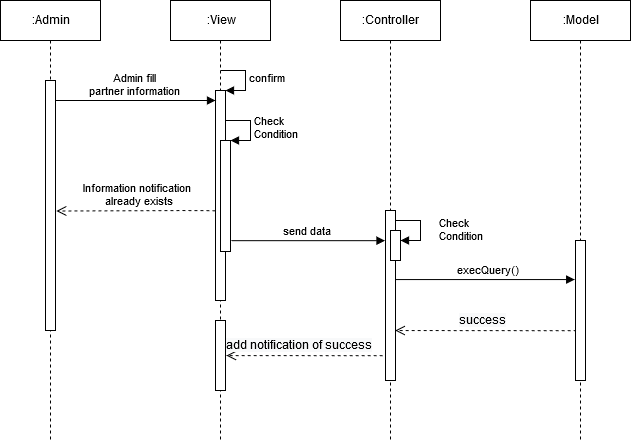
* Update checking:



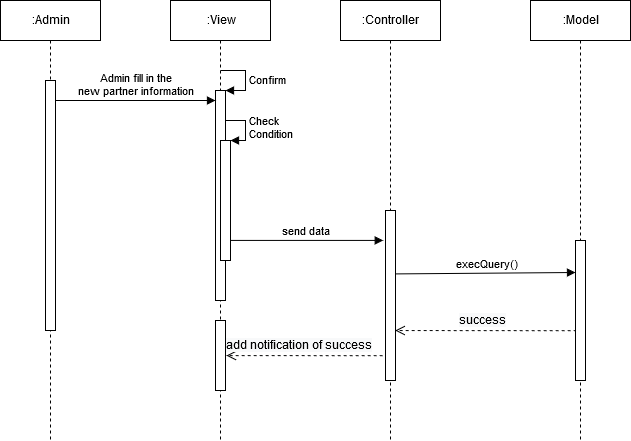
* View partners:



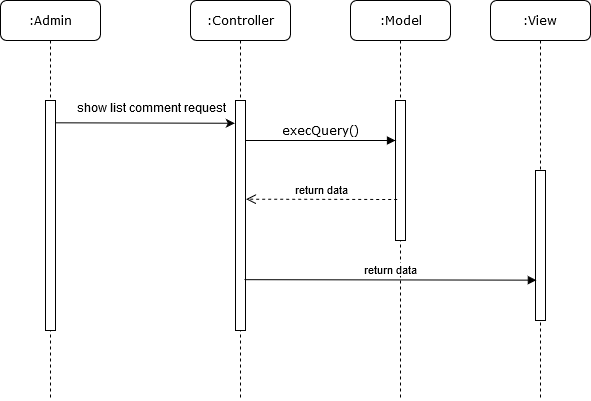
* Add partner:



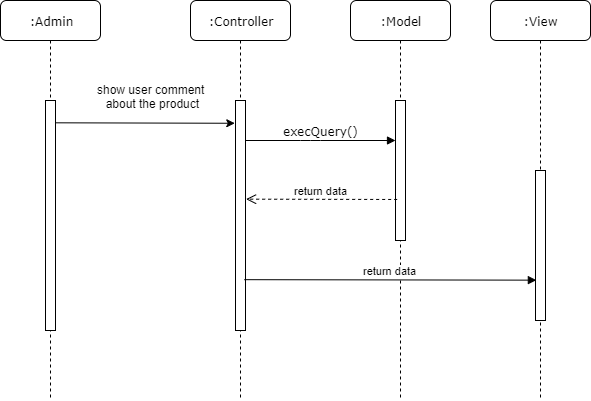
* Update partner:



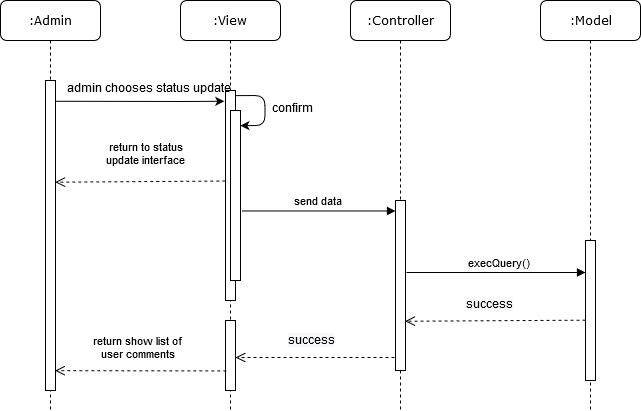
* View comments:



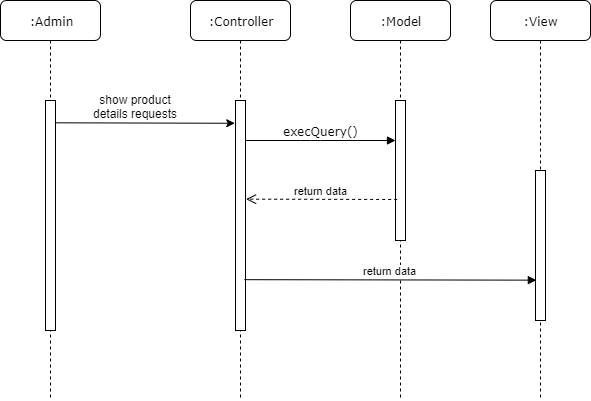
* View comment details:



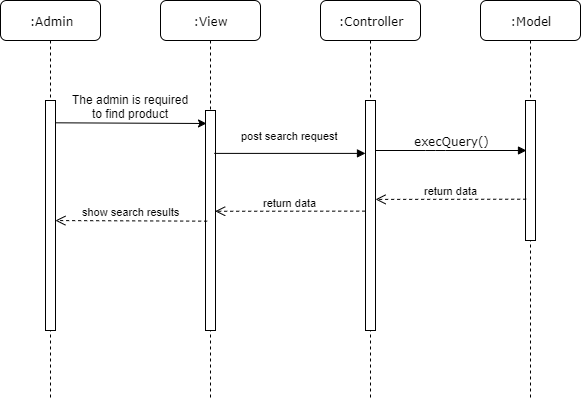
* Active comment:



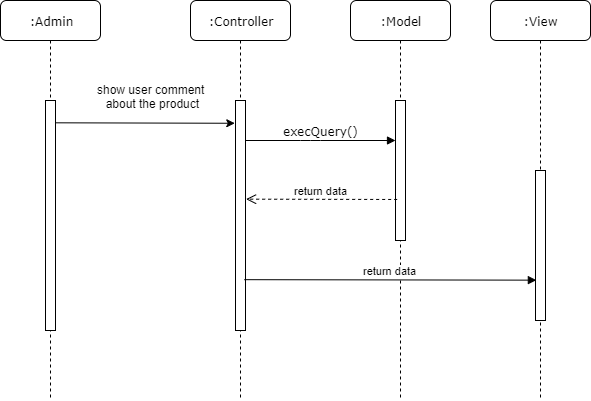
* View product detail:



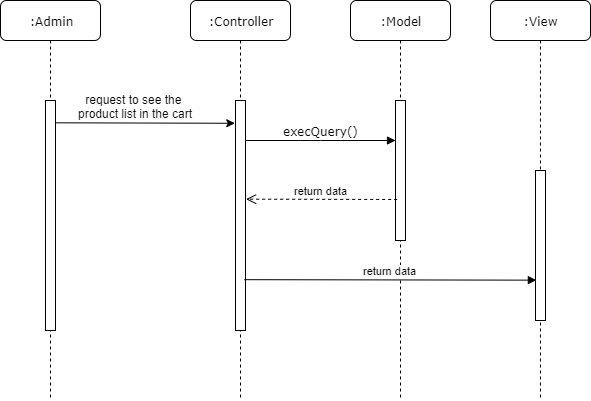
* Search products:



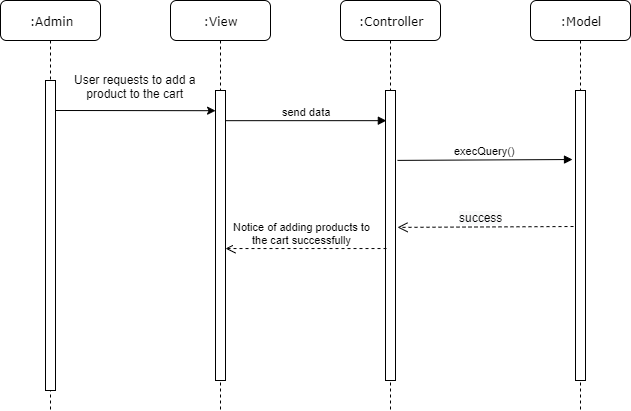
* View comment of product:



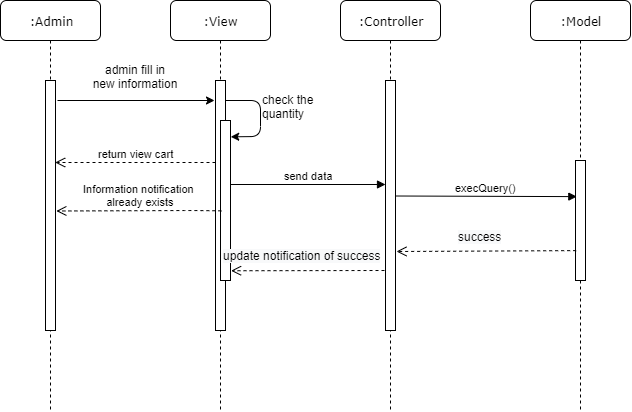
* View cart:



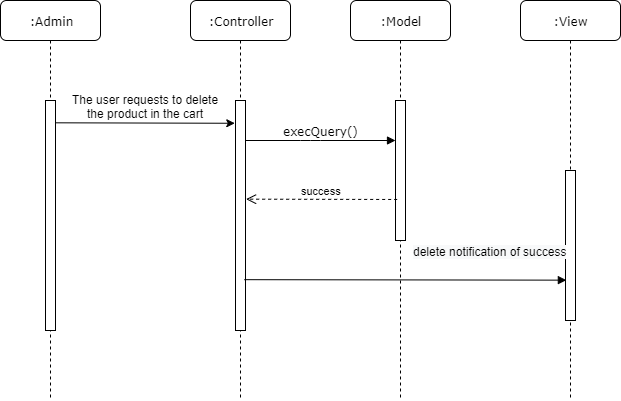
* Add products to cart:



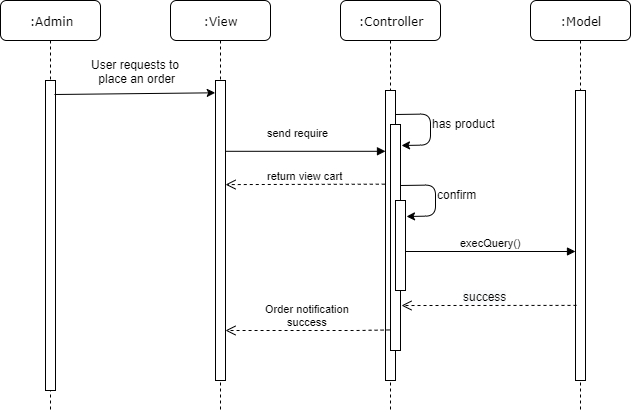
* Change quantity product from cart:



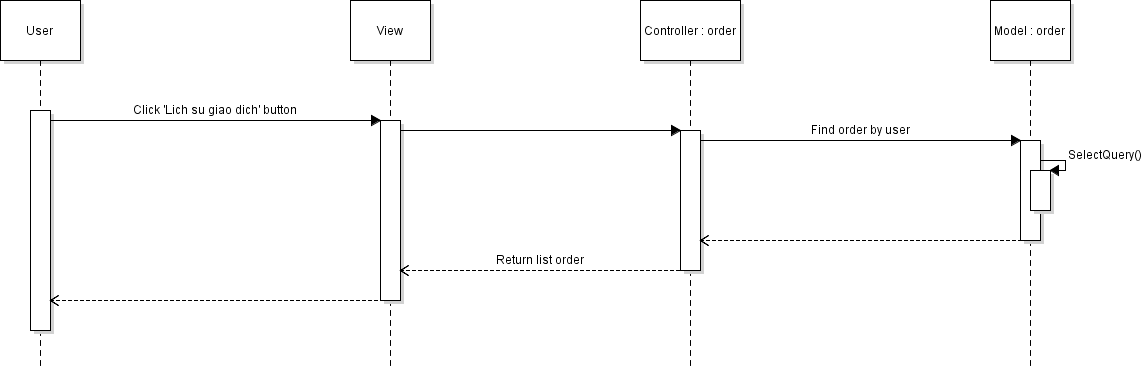
* Delete product from cart:



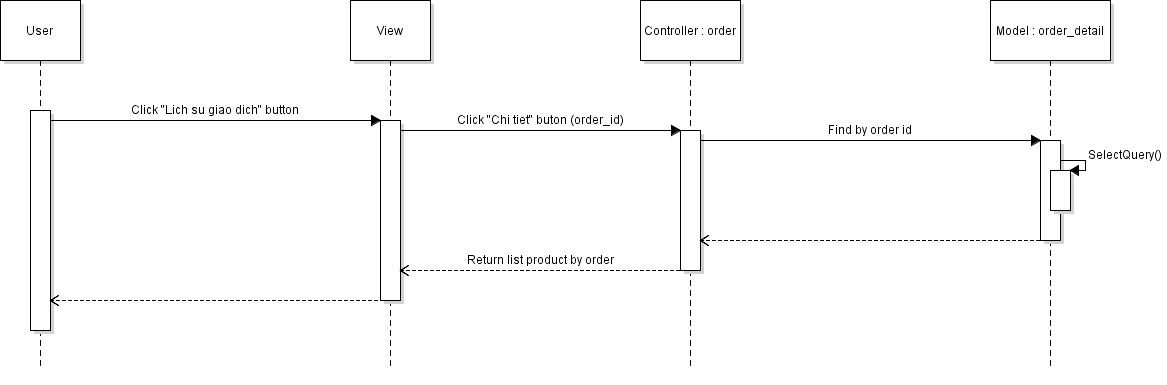
* Confirm order:



* View orders:



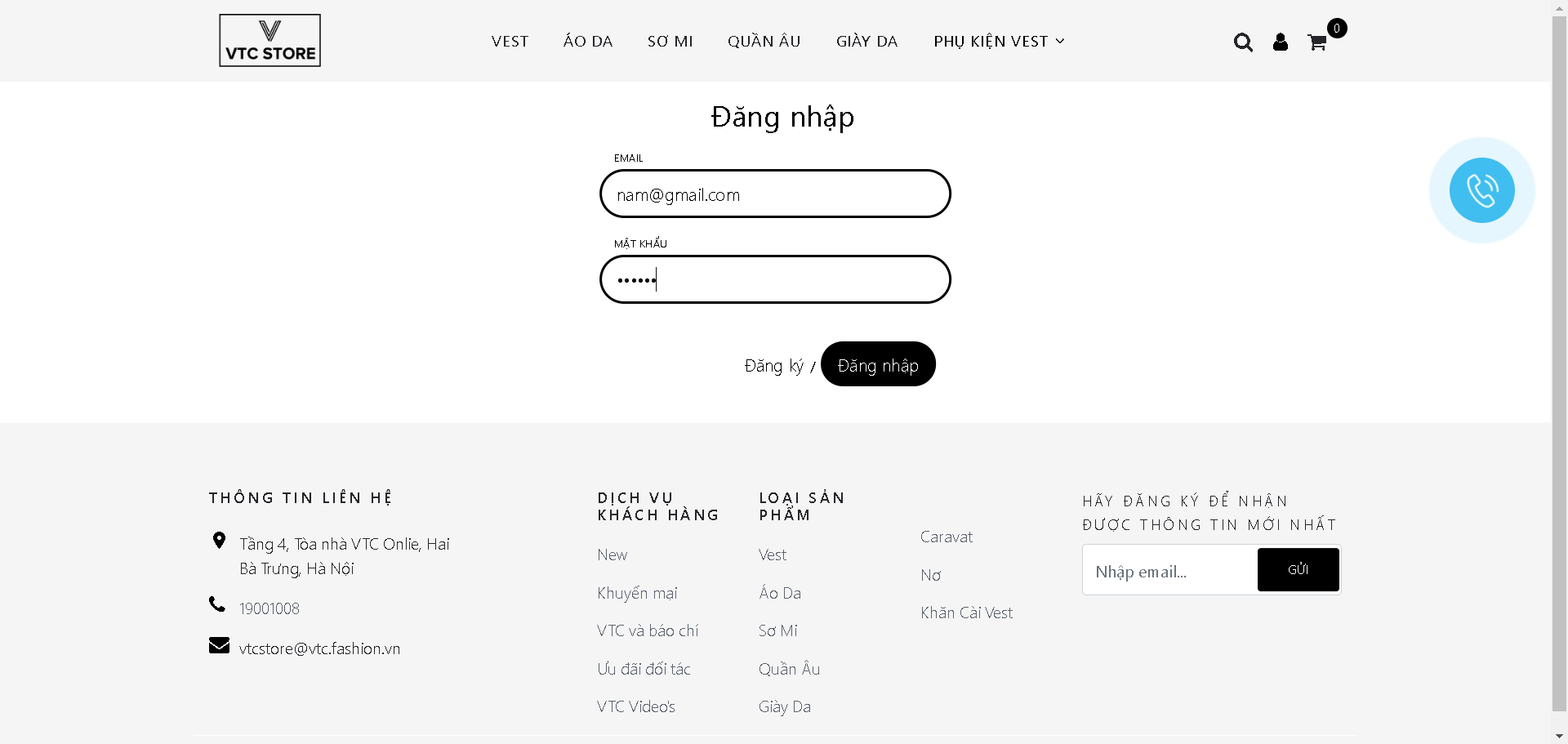
* View order details:



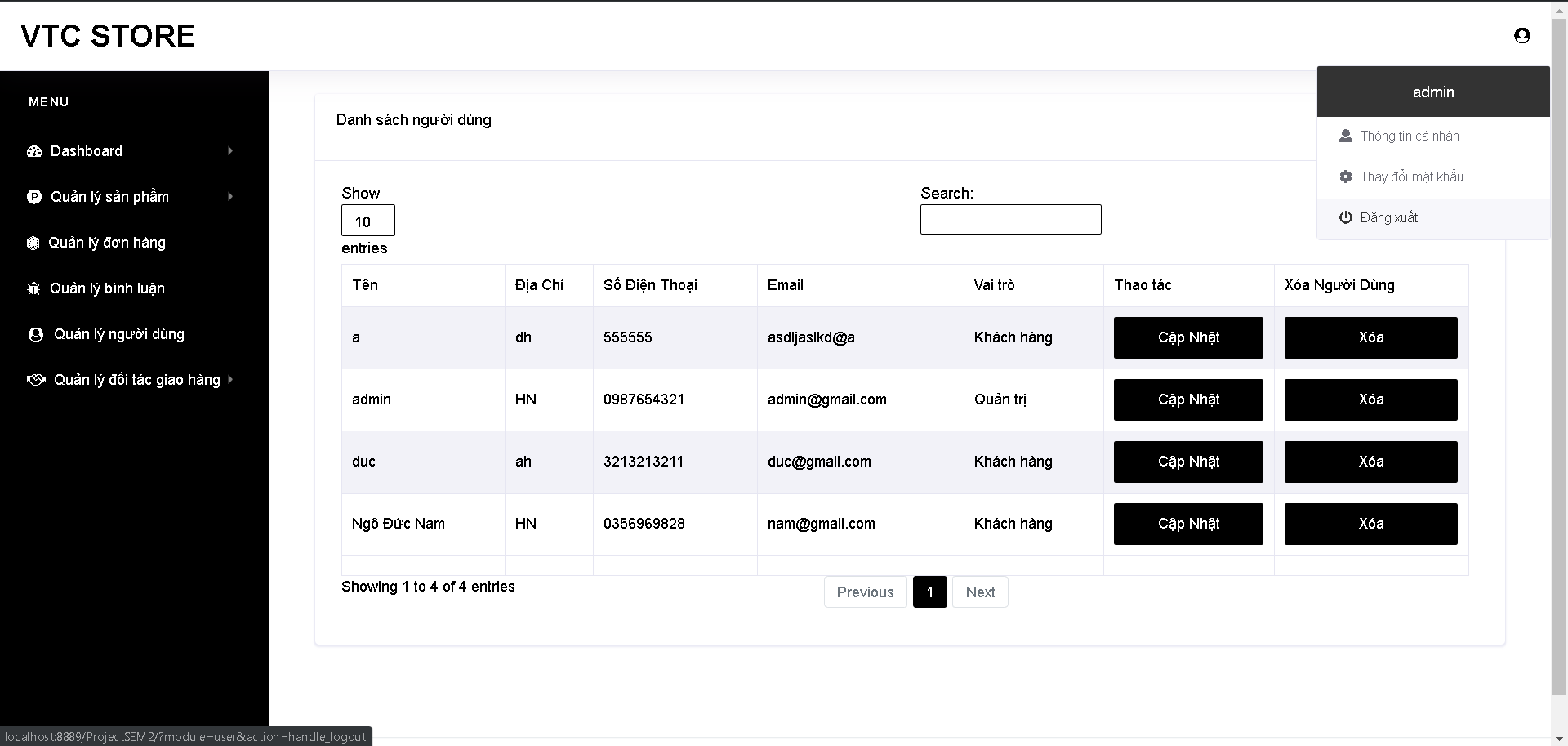
# Detailed design

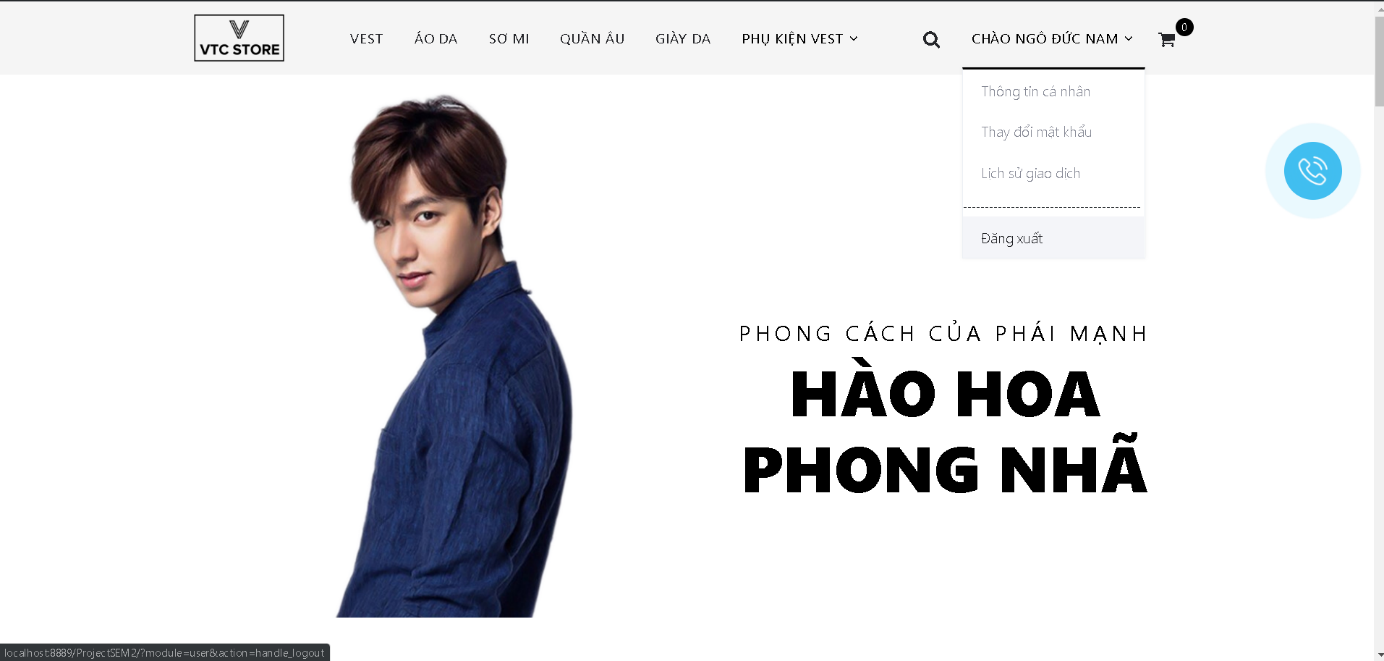
## UI design

* Login (admin and customer):

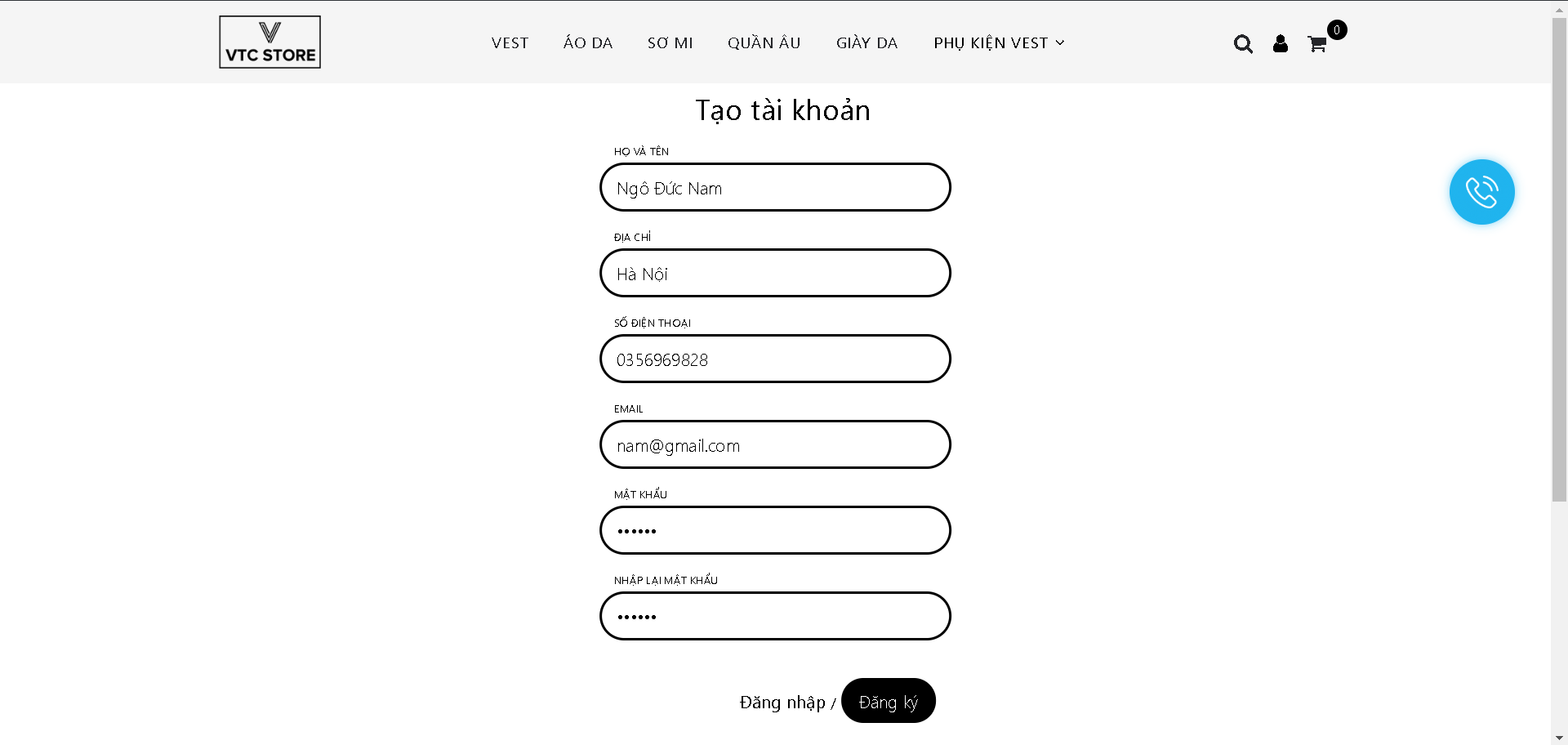


* Logout (first admin and second customer):

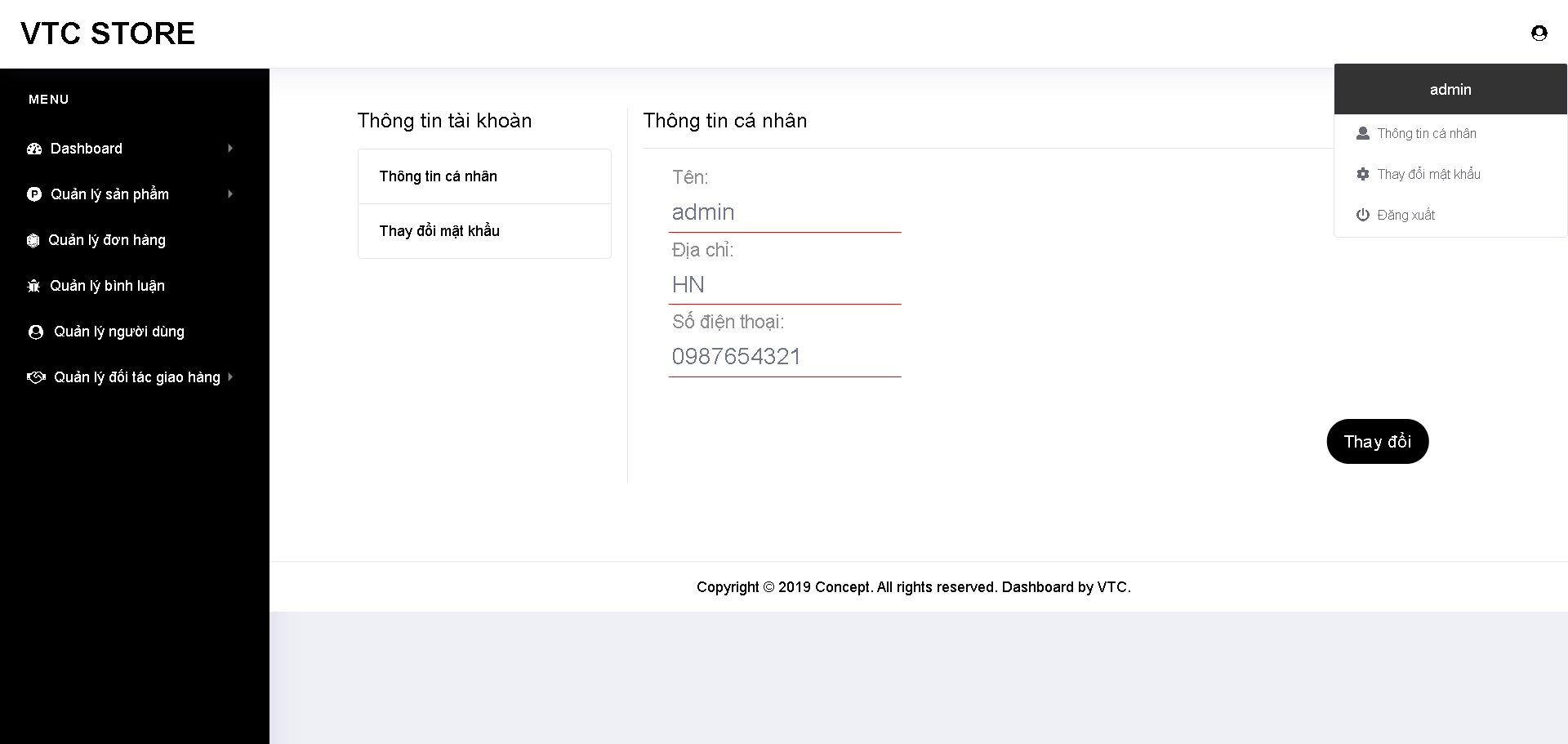


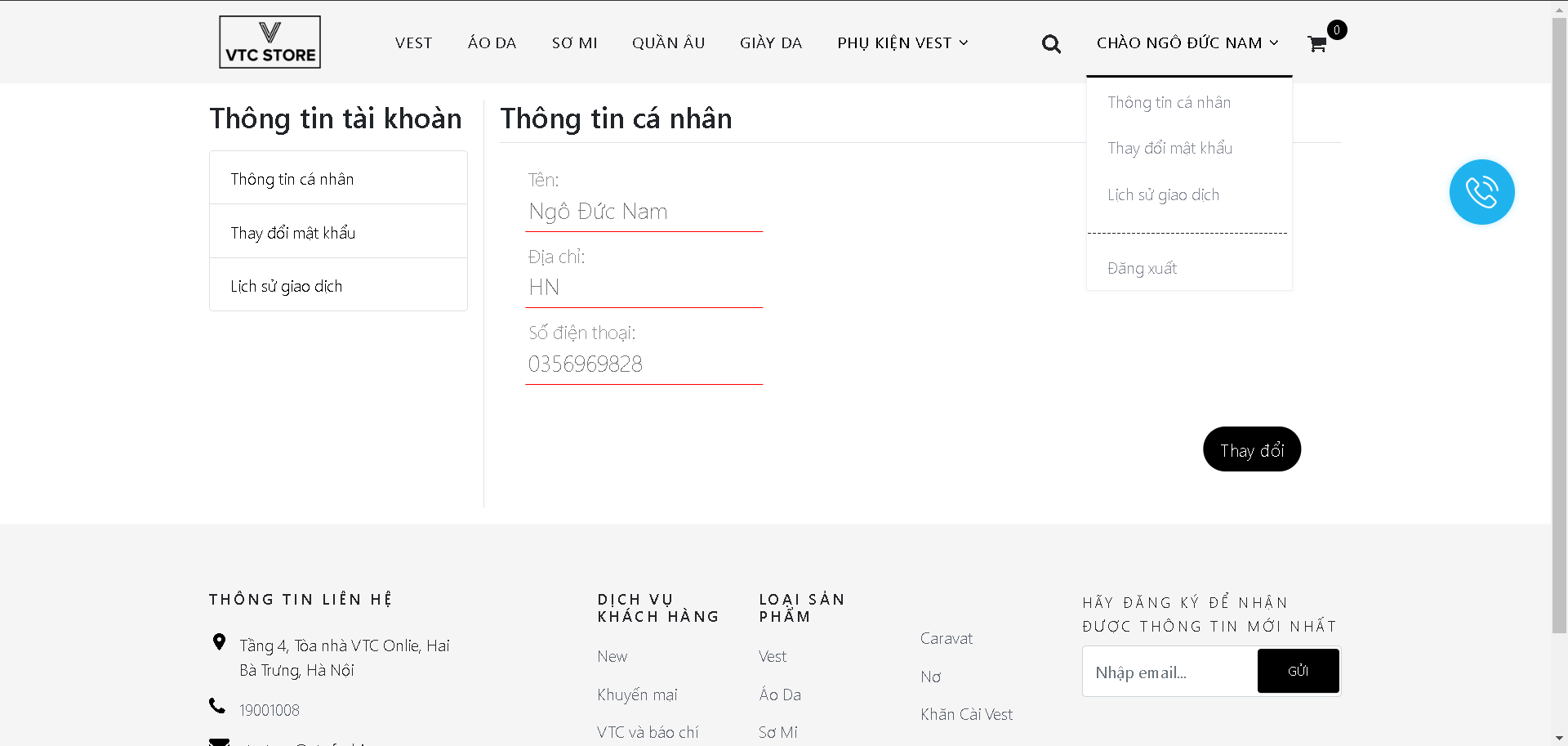


* Registration (customer):

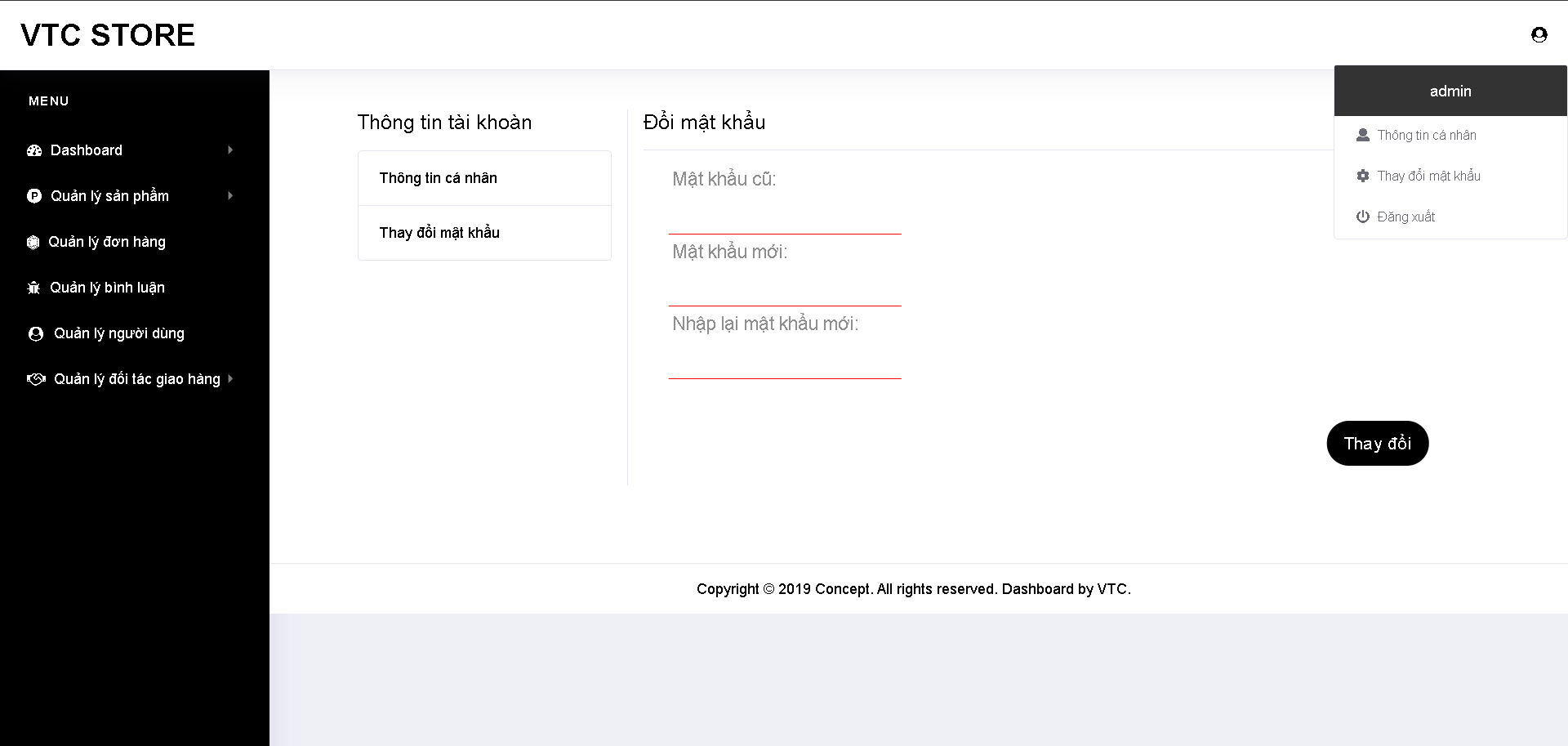


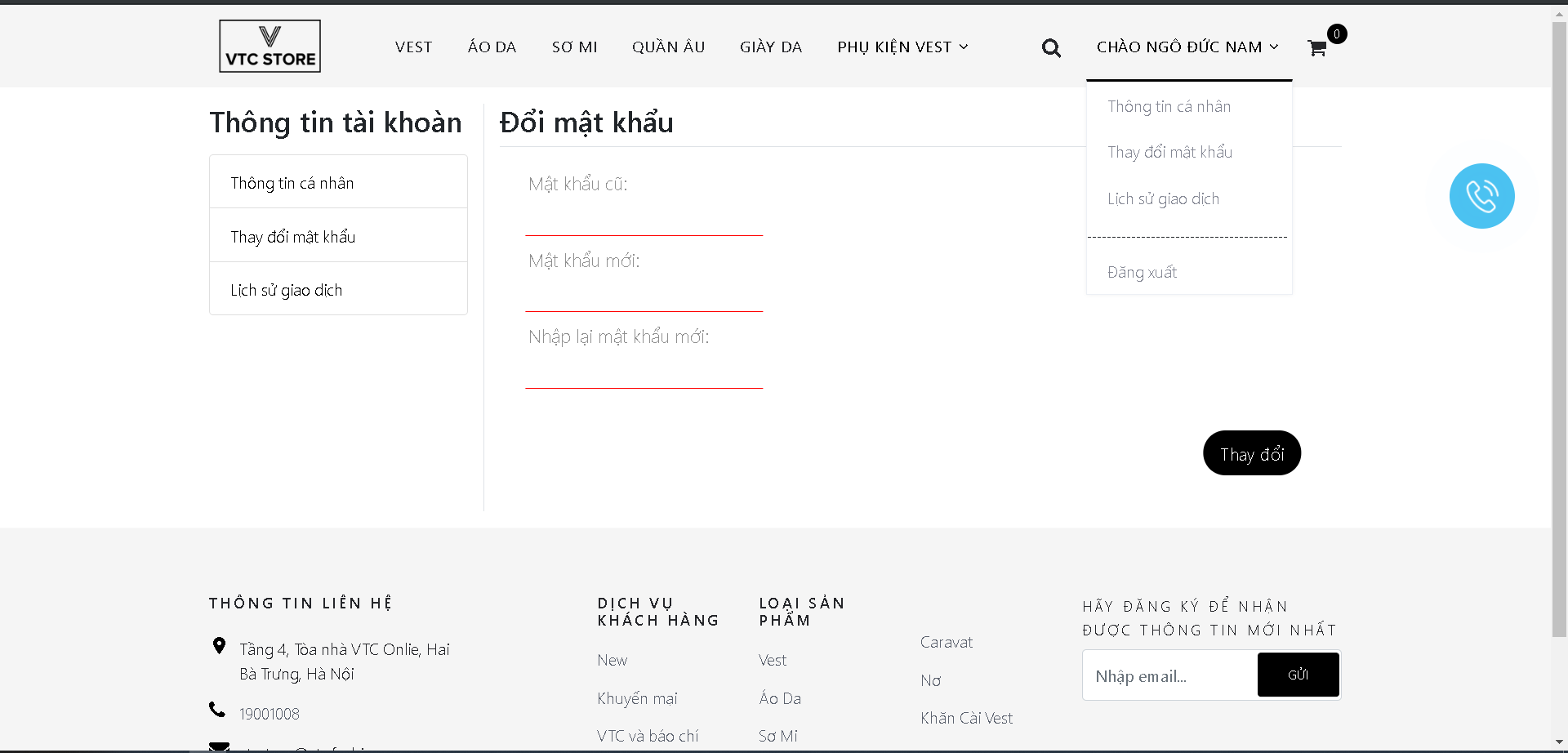
* Update profile (first admin and second customer):



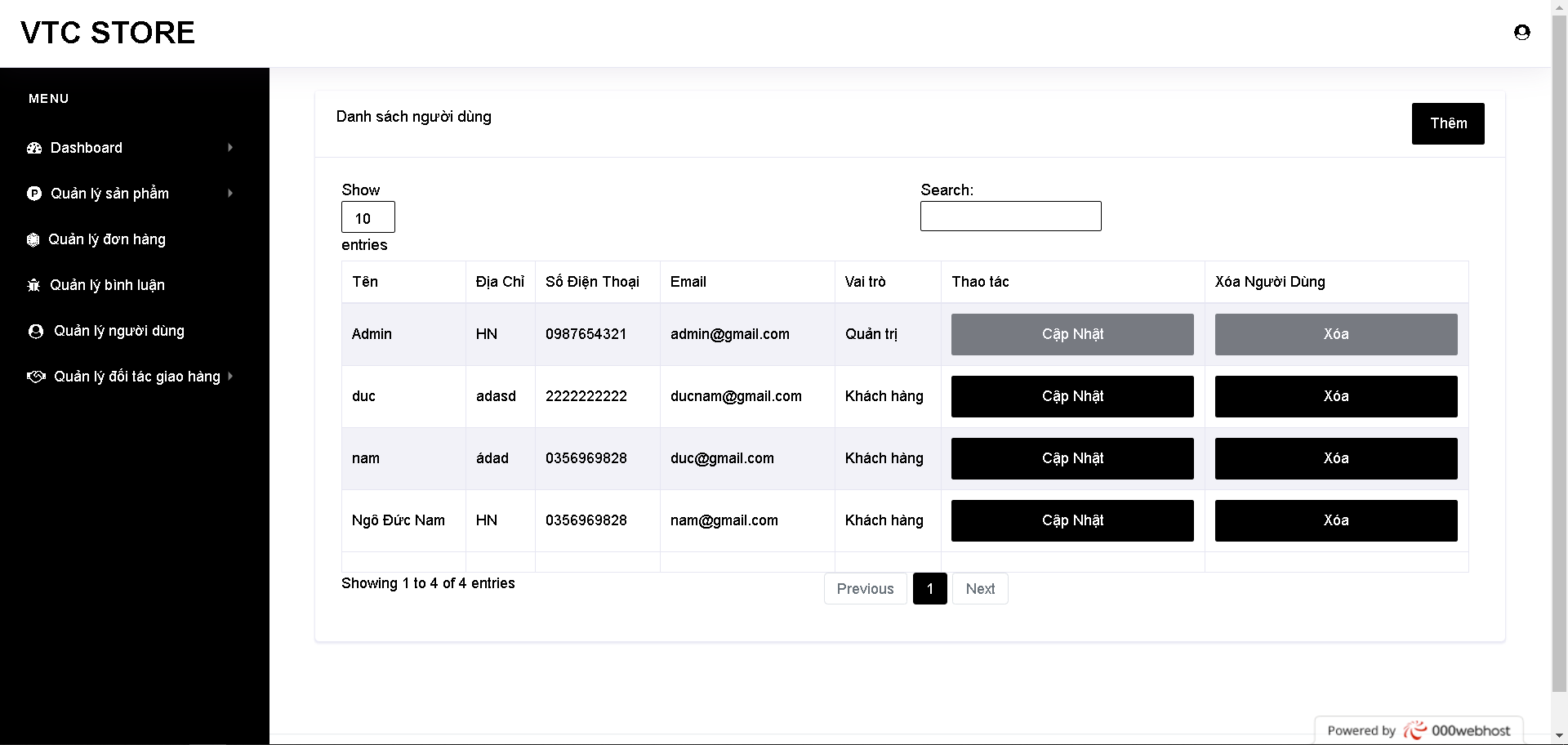


* Change password (first admin and second customer):

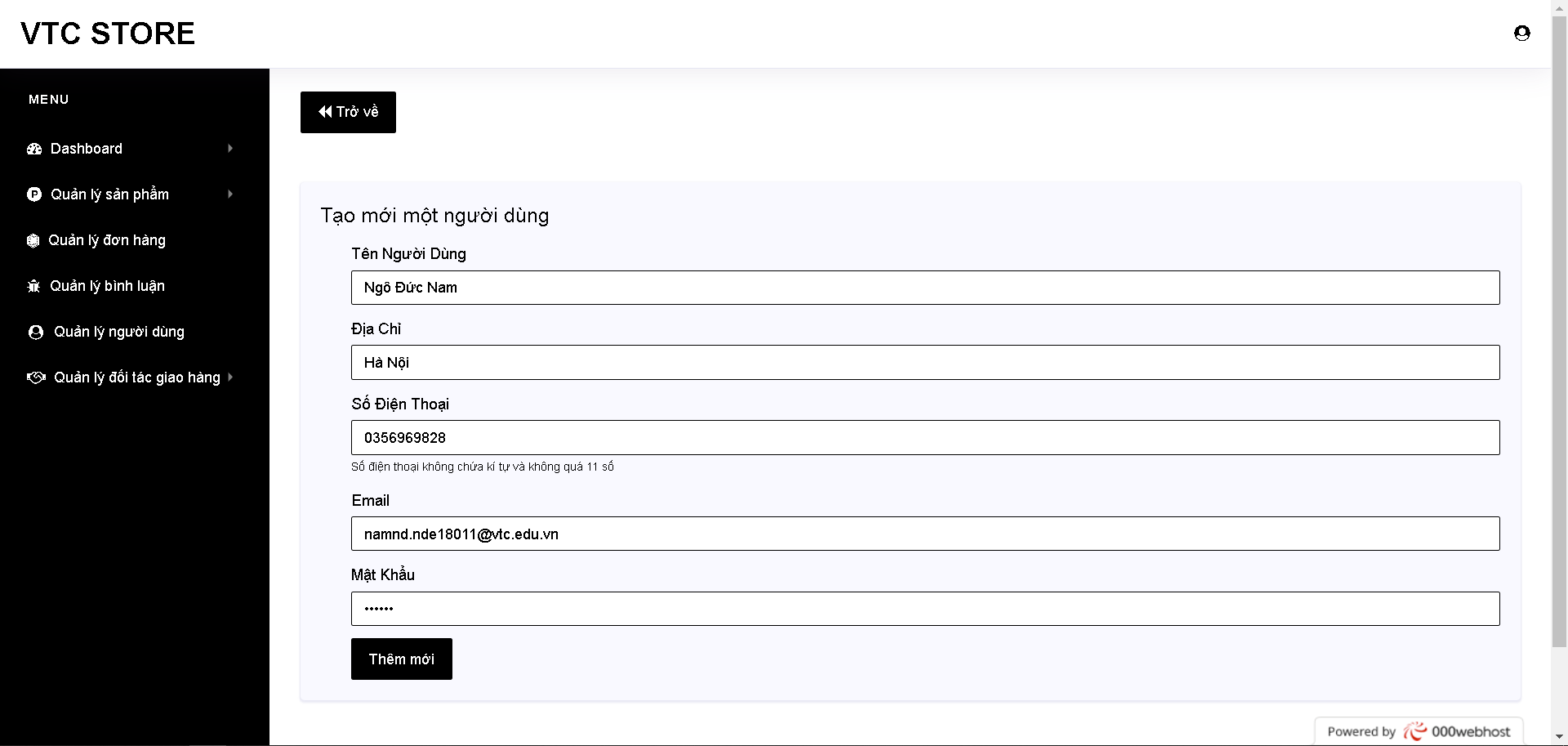




* View users:



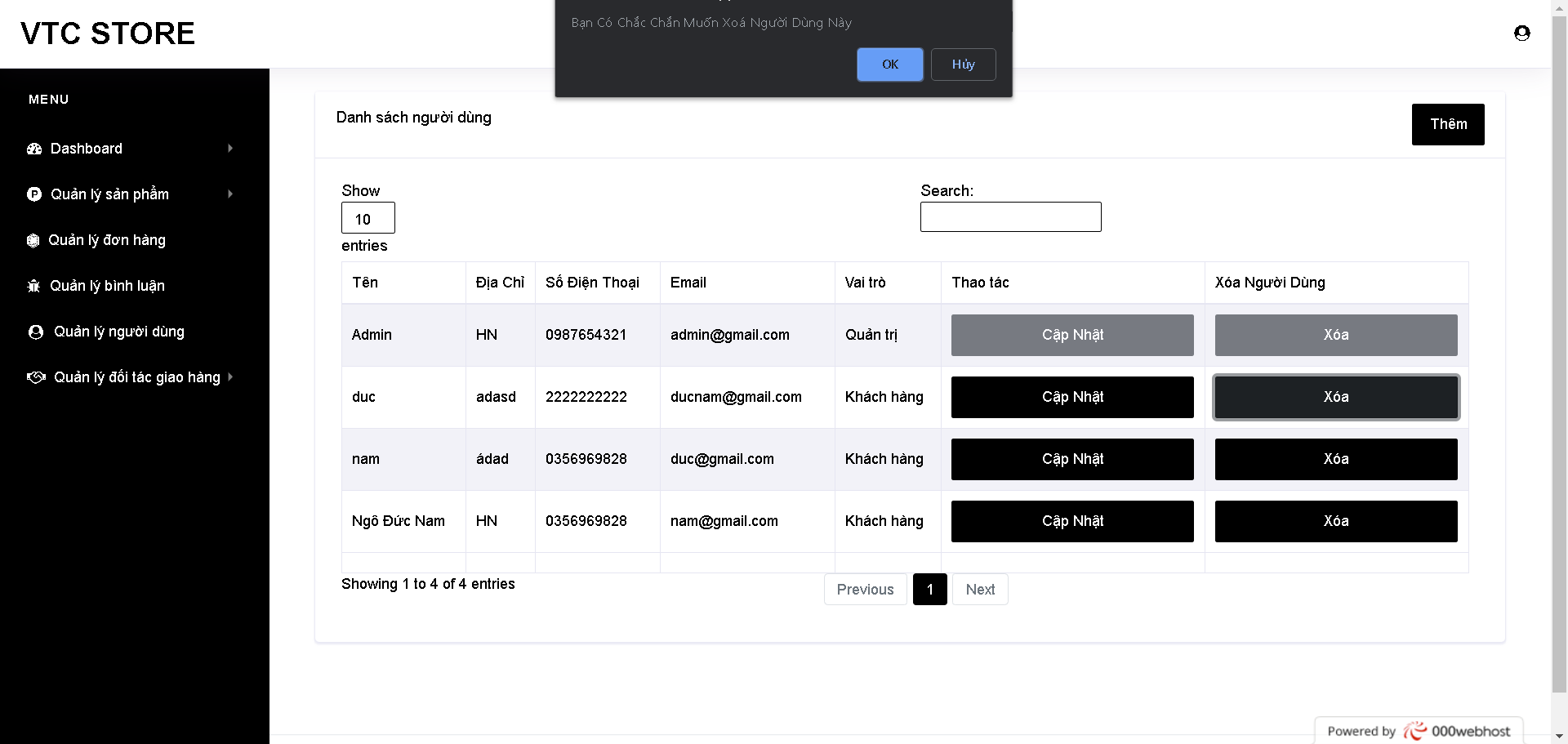
* Add user:



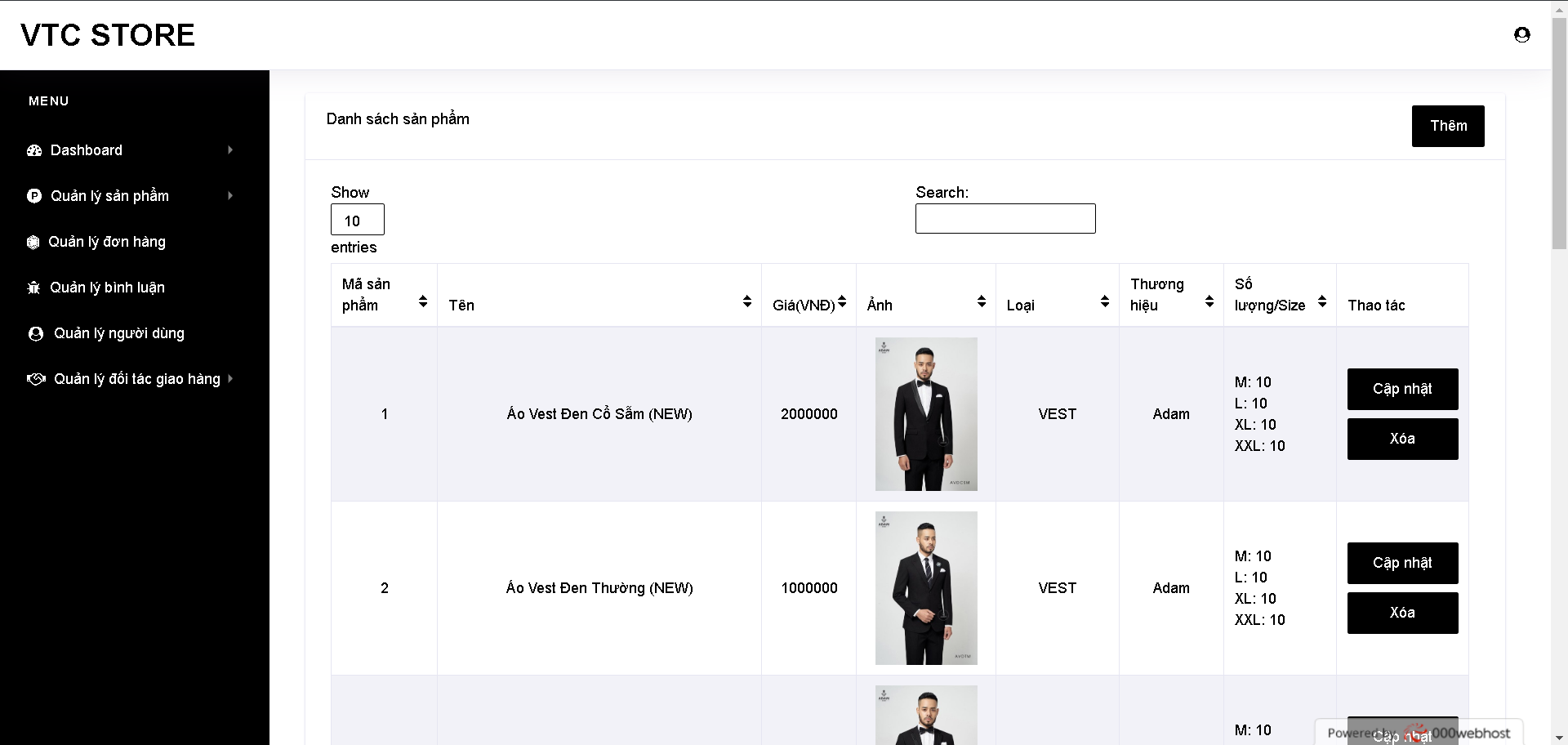
* Update user:



* Delete user:



* View products:



* Add product:



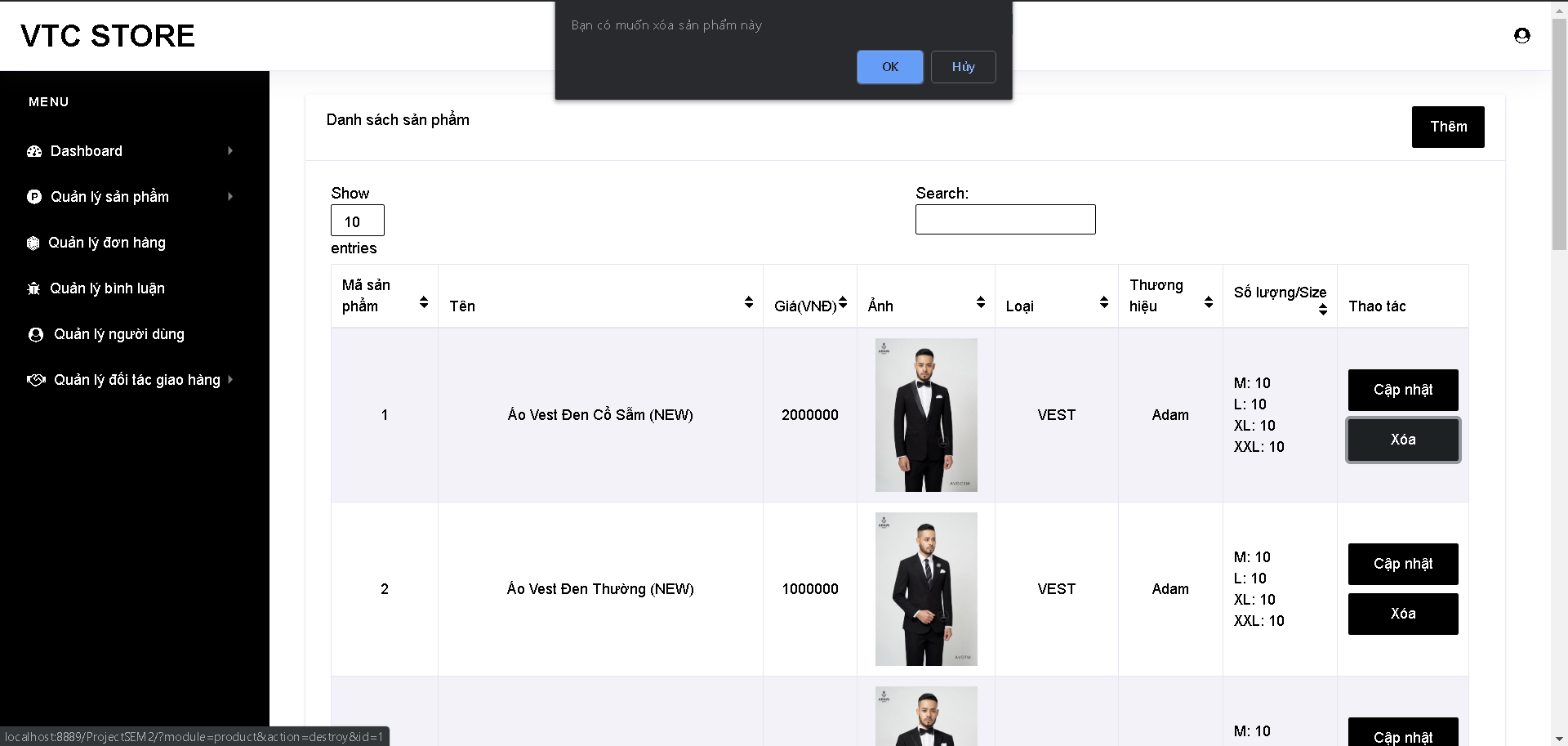
* Update product:



* Add/Update size:



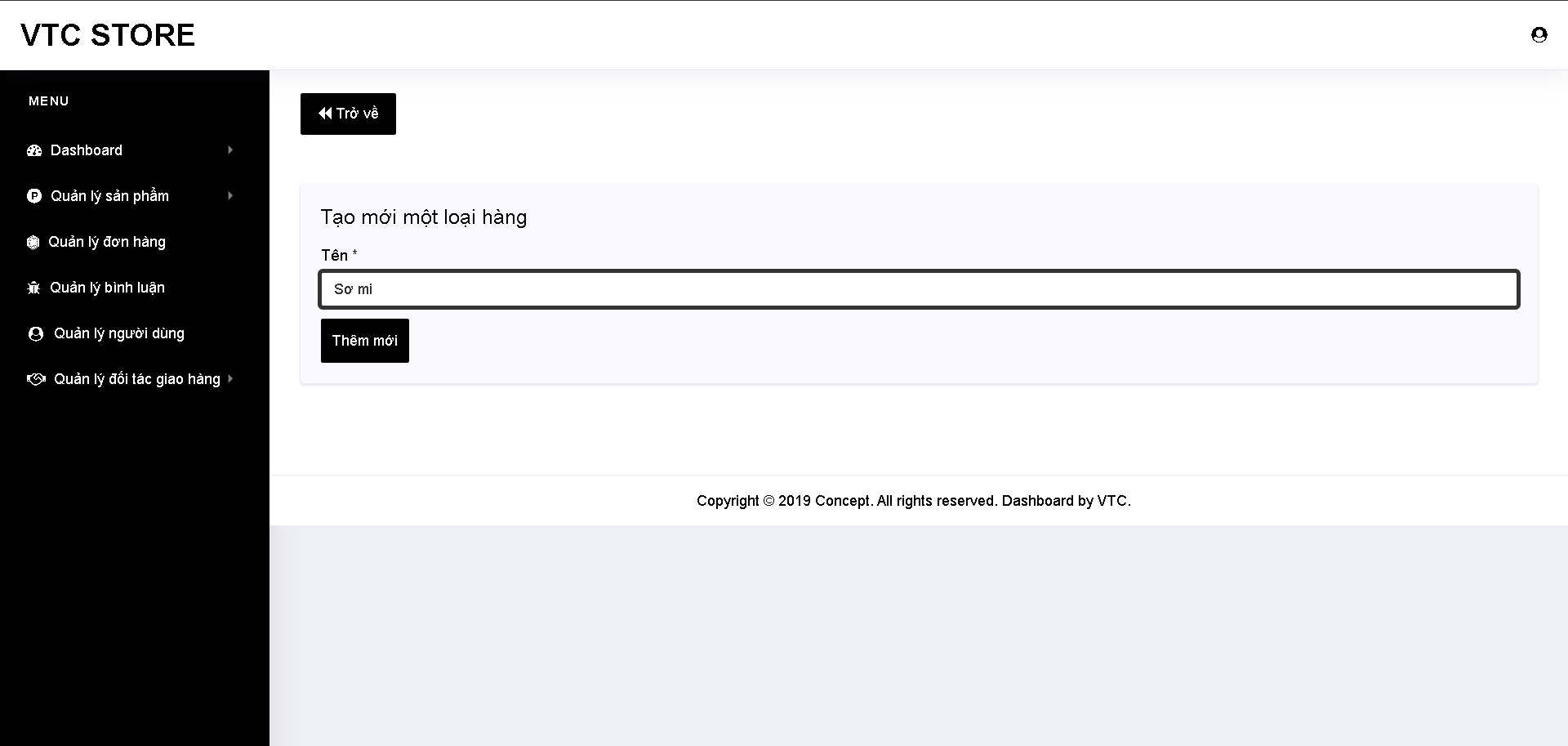
* Delete product:



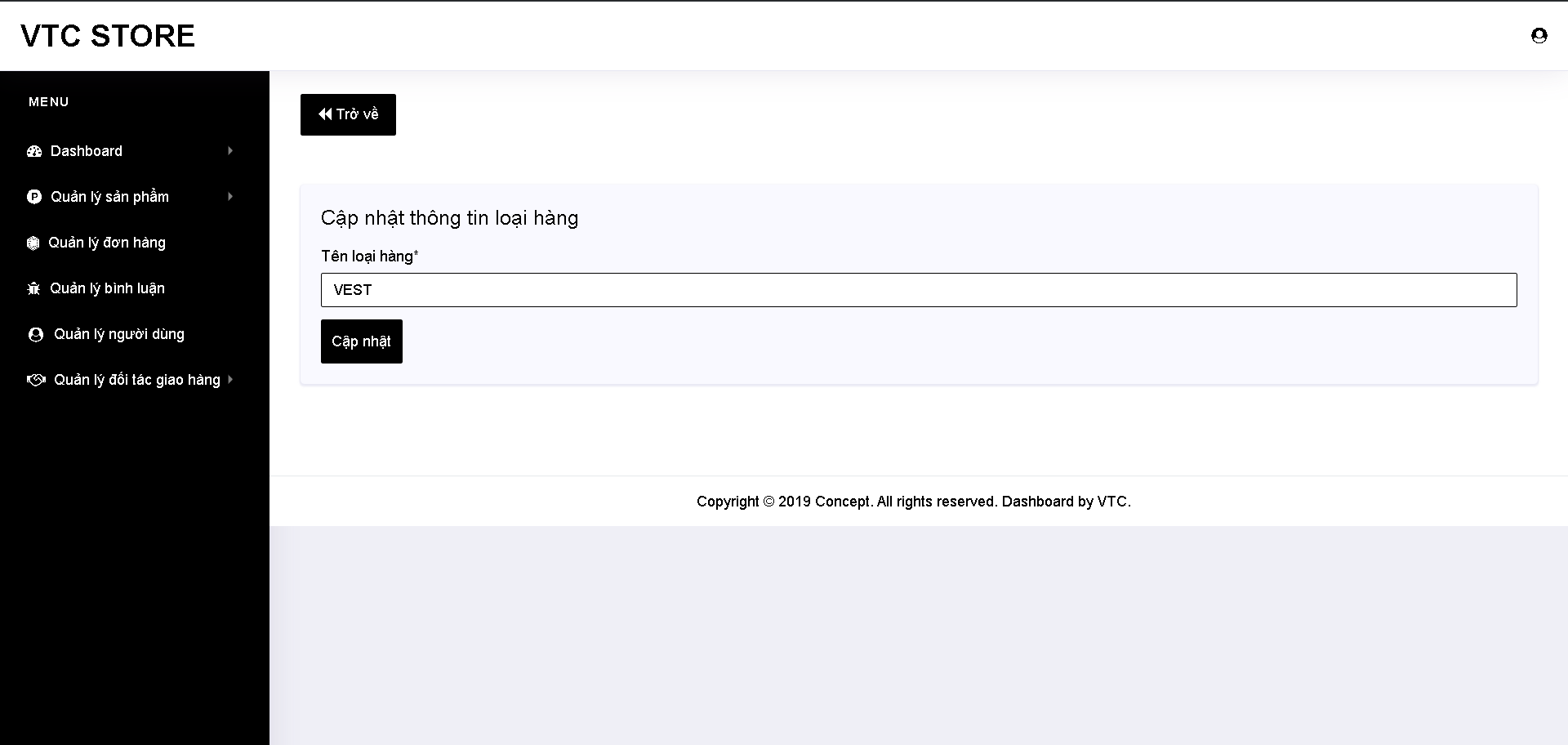
* View categories:



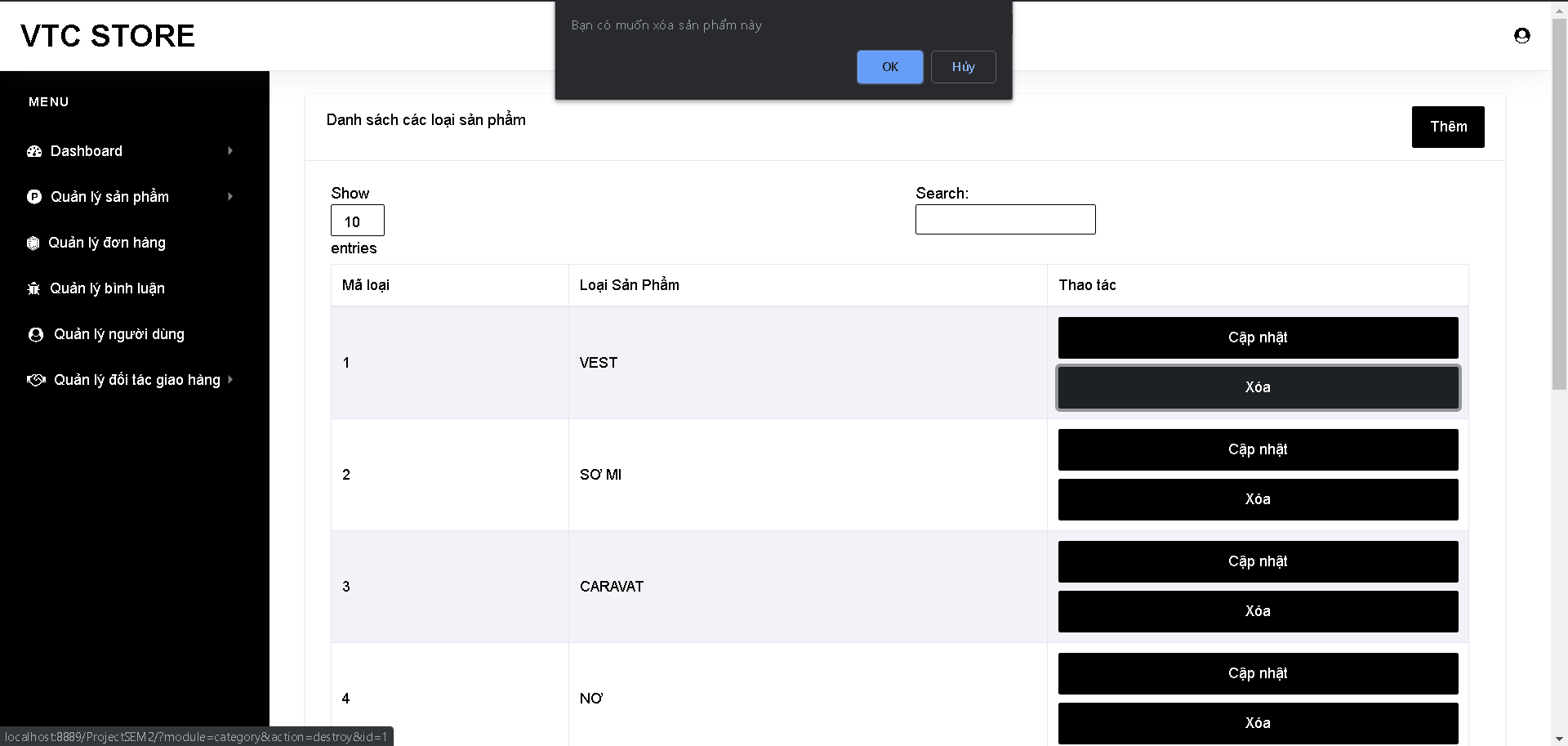
* Add categroy:



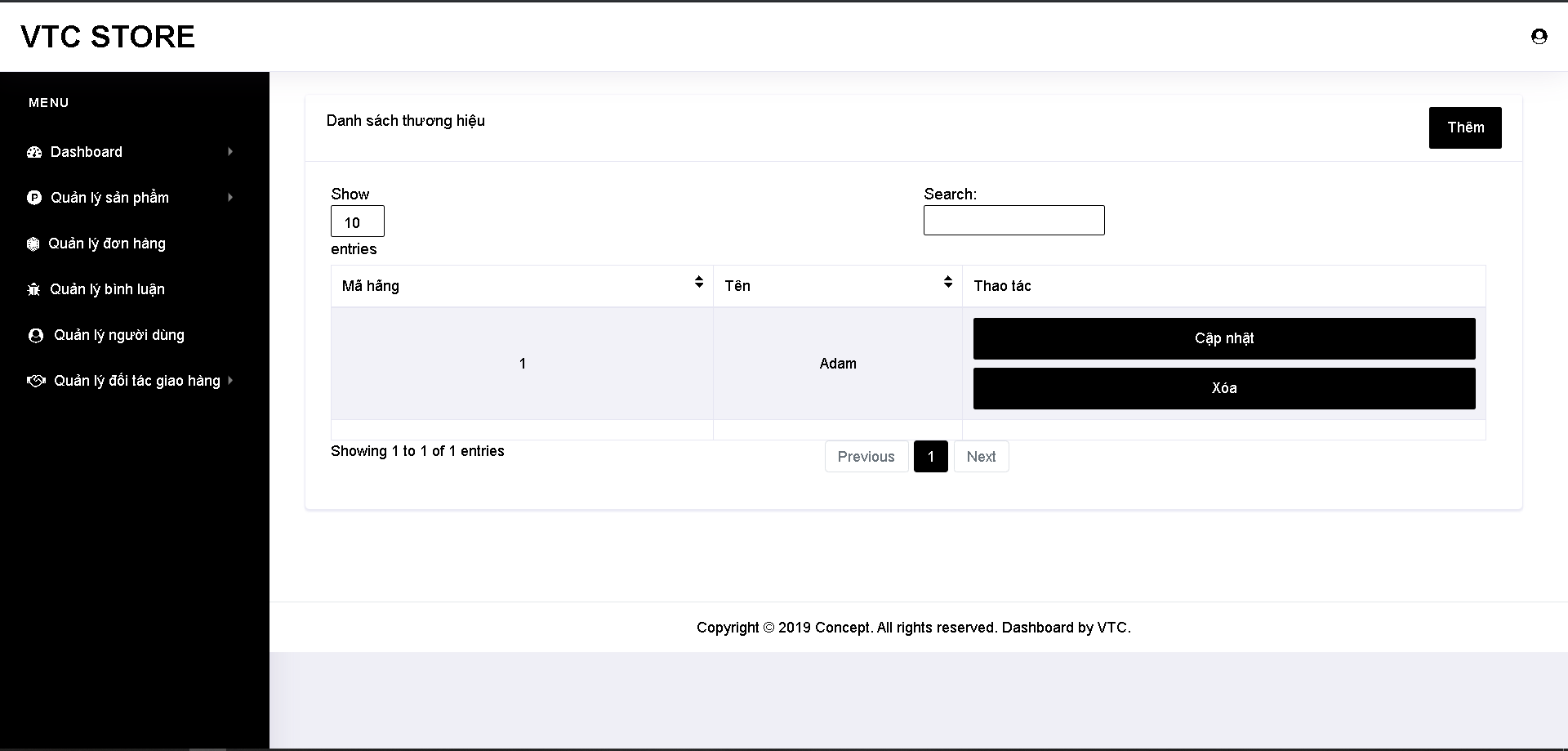
* Update category:



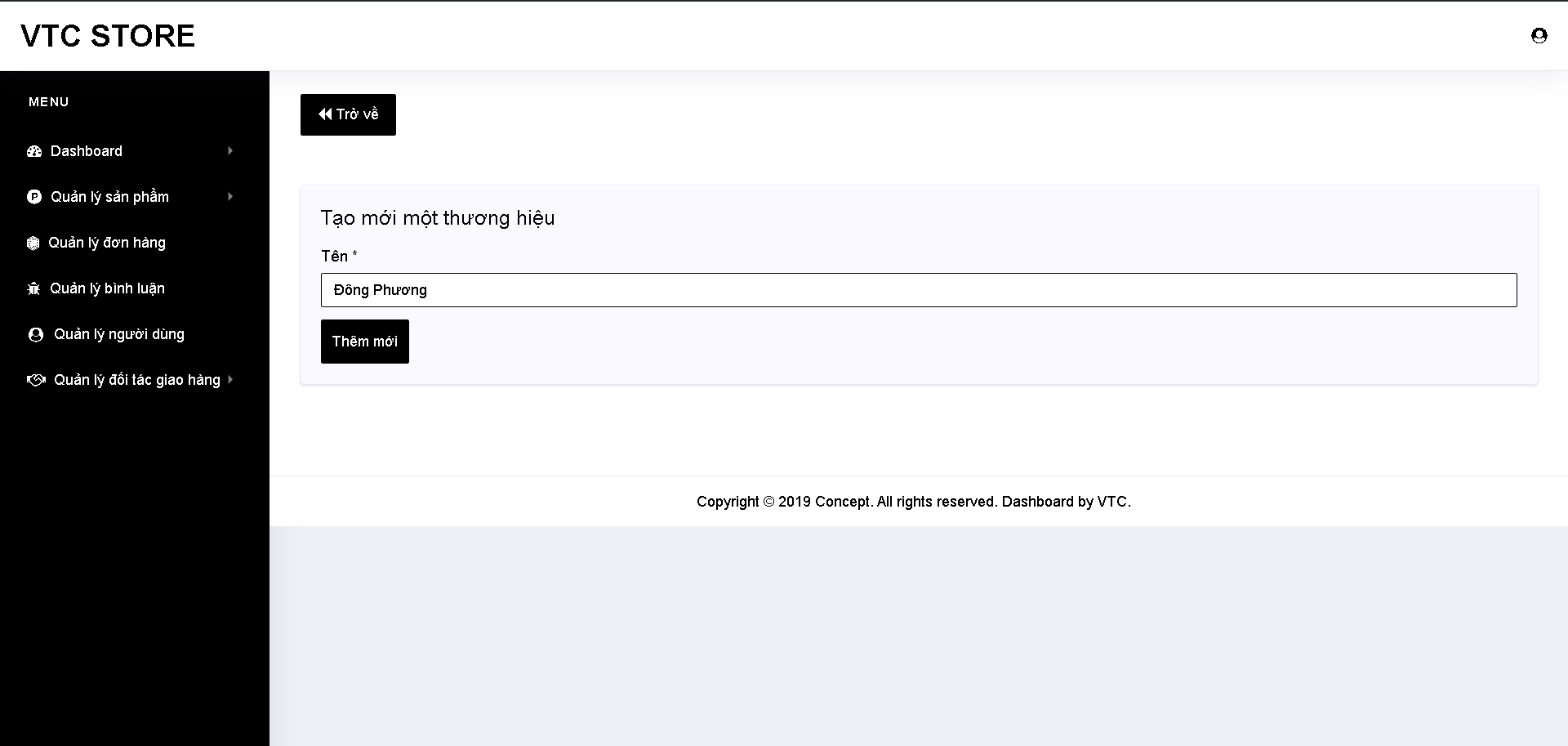
* Delete category:



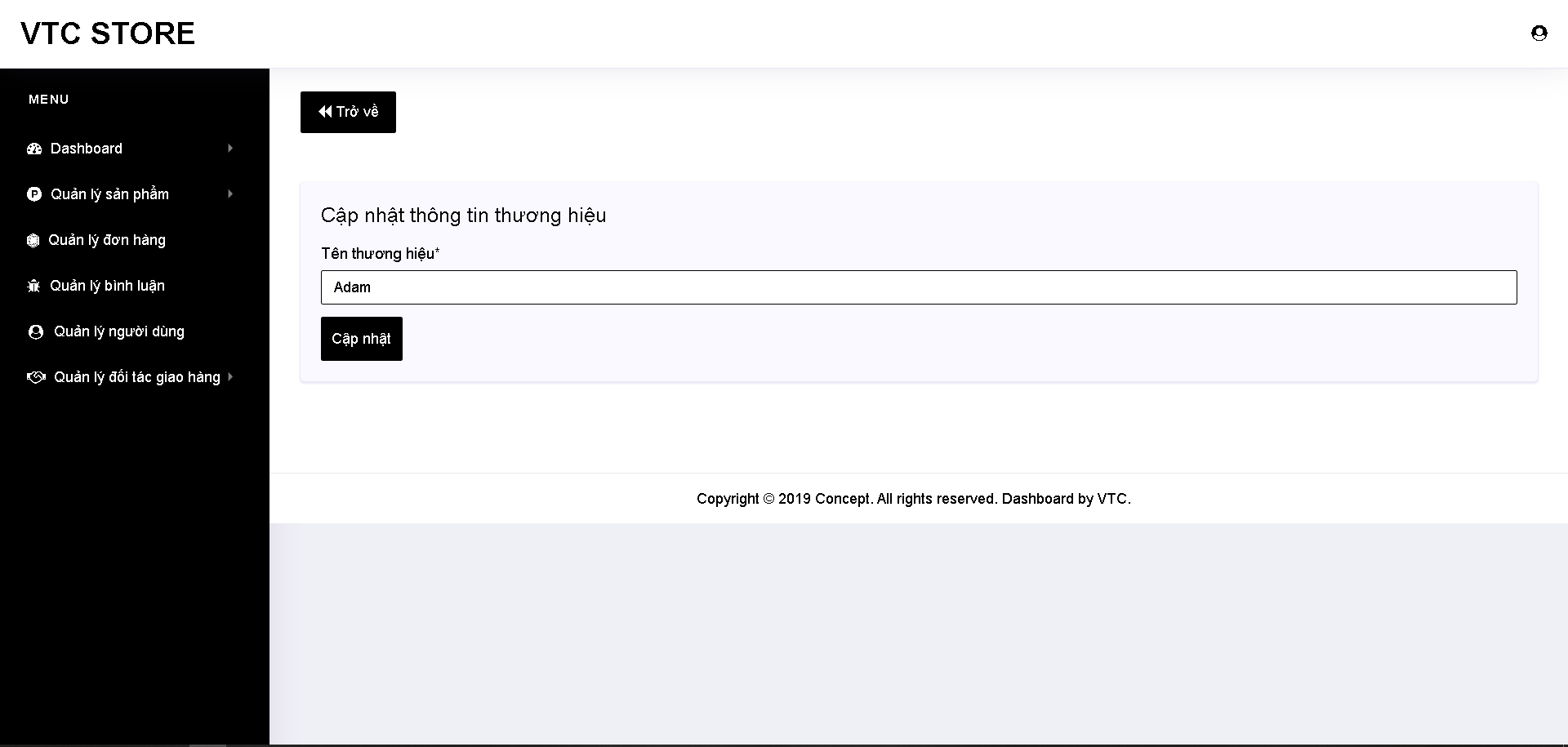
* View brands:



* Add brand:



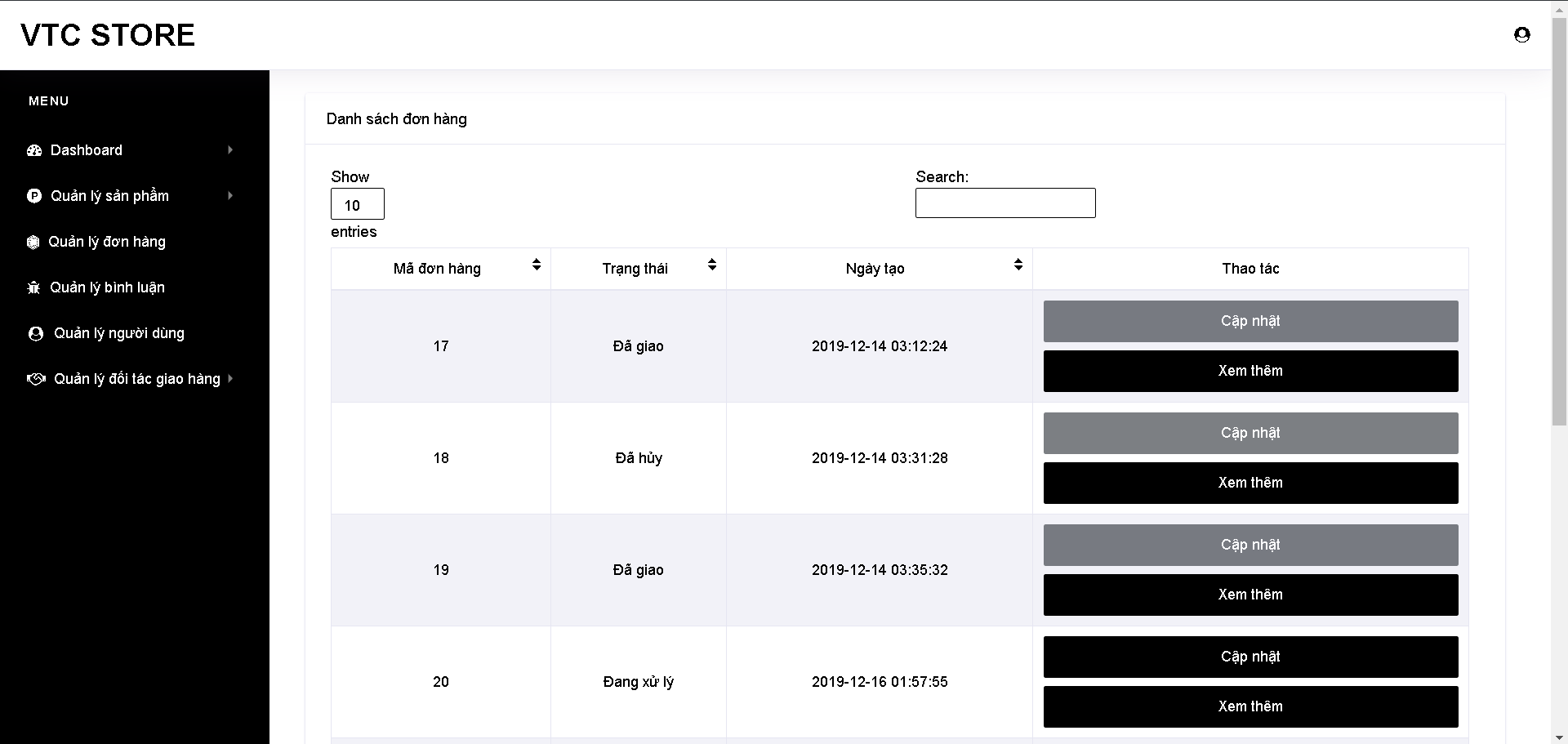
* Update brand:



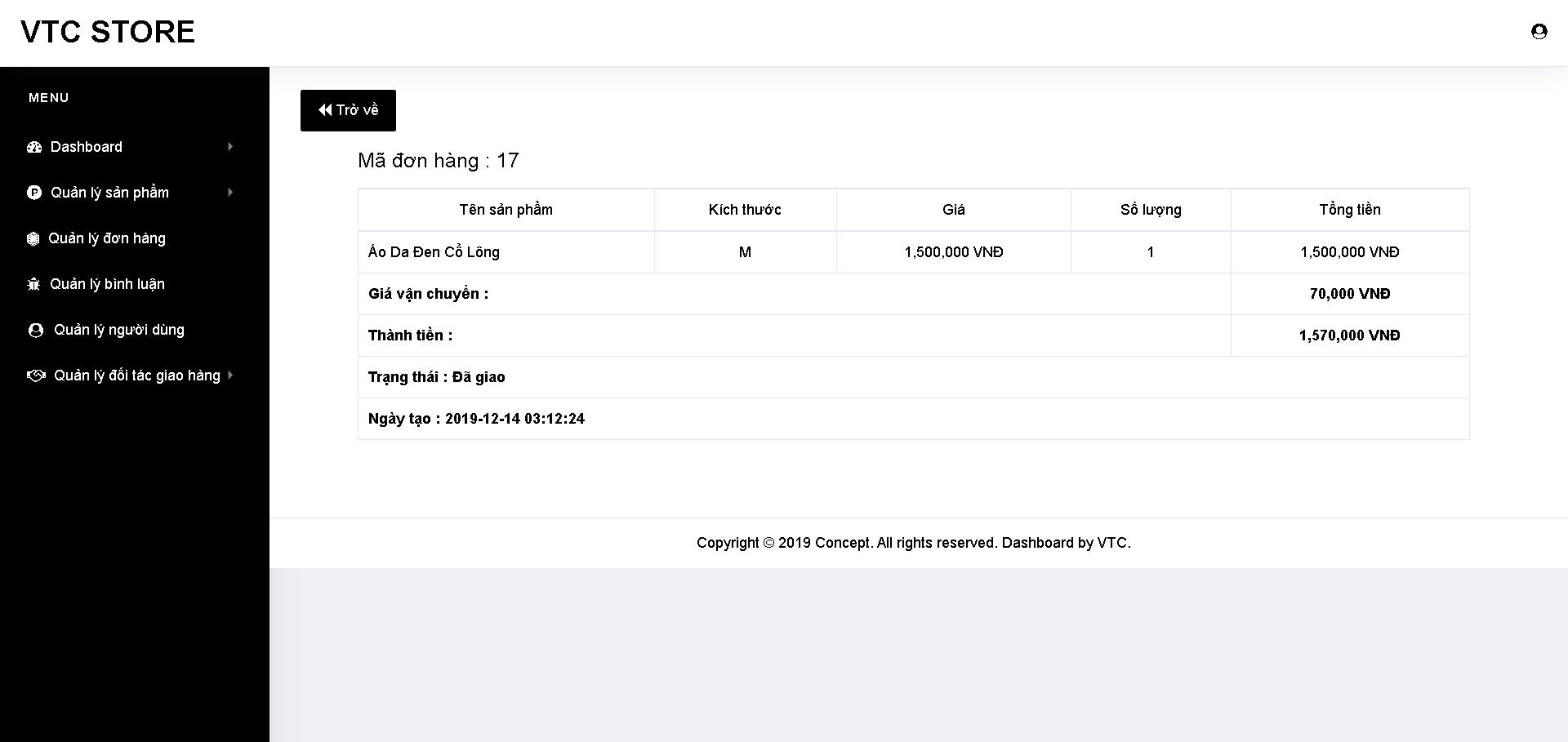
* Delete brand:



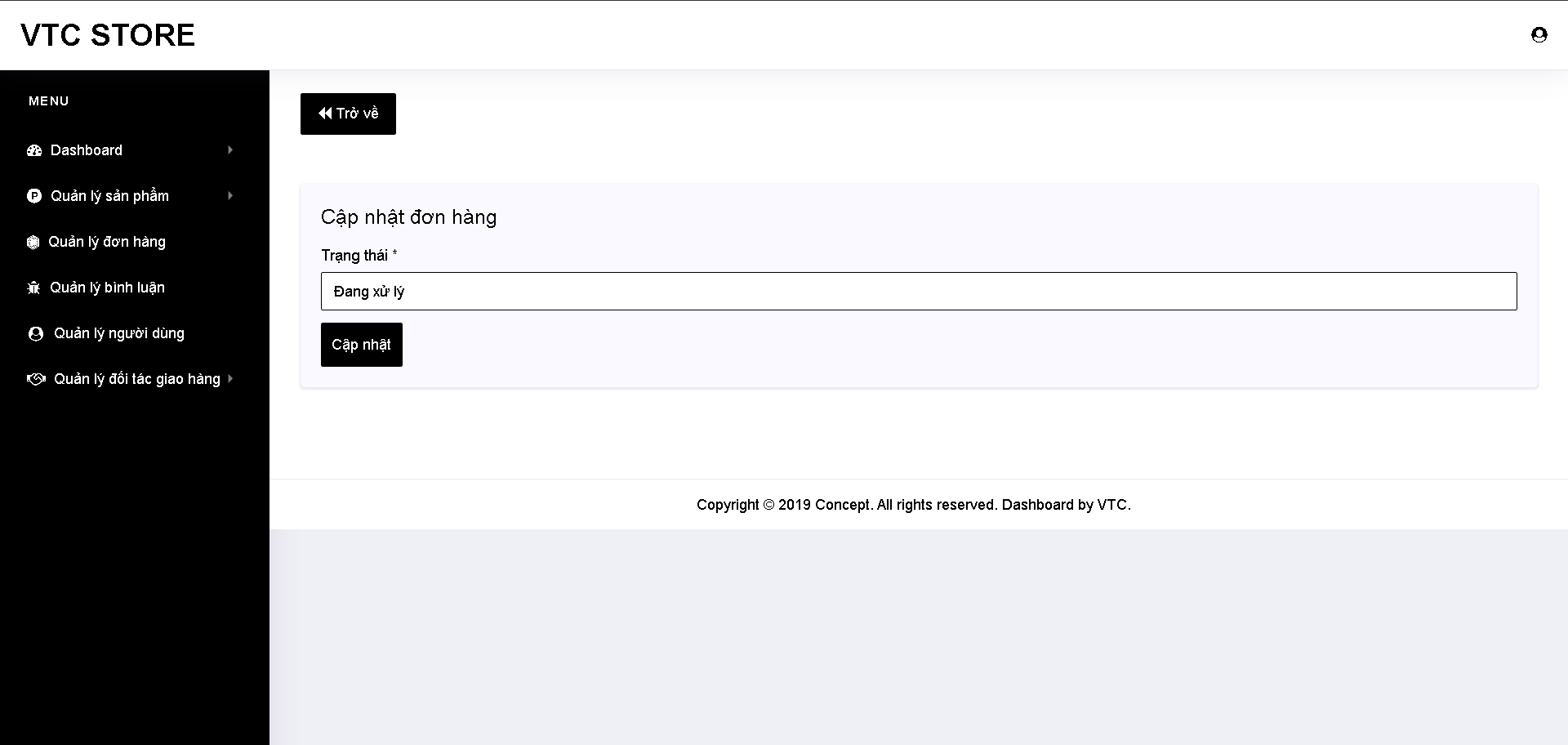
* View orders:



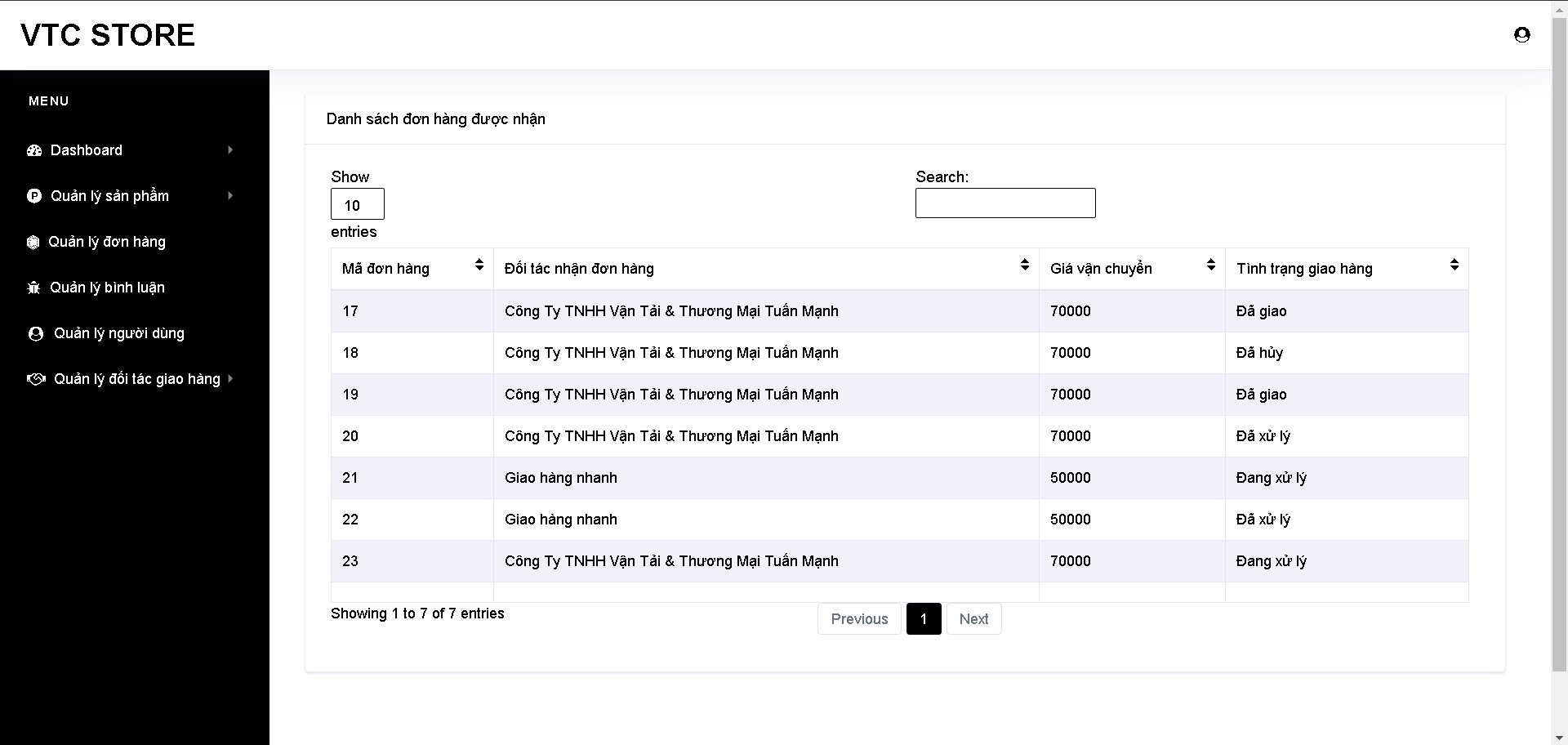
* View order details(Admin):



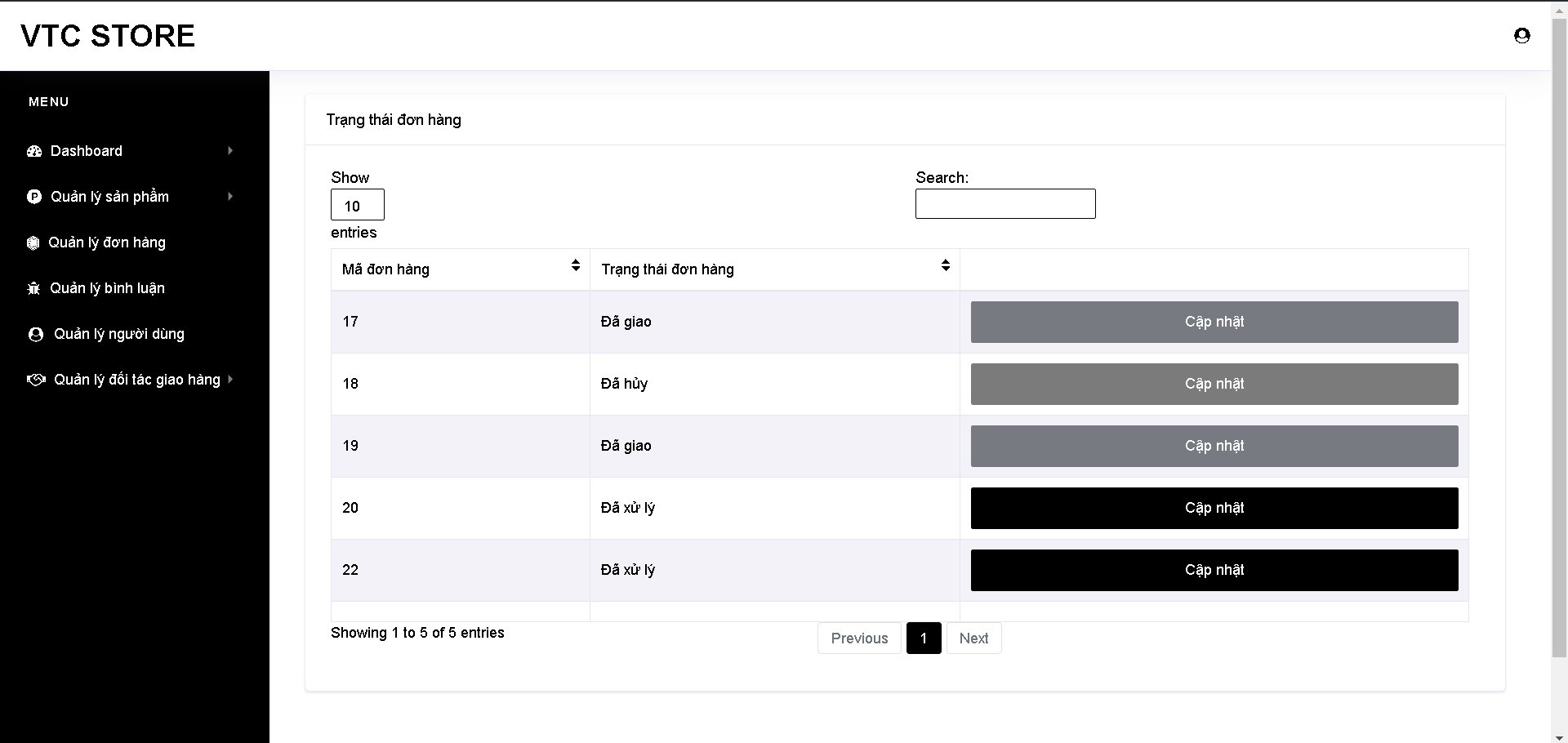
* Update order:



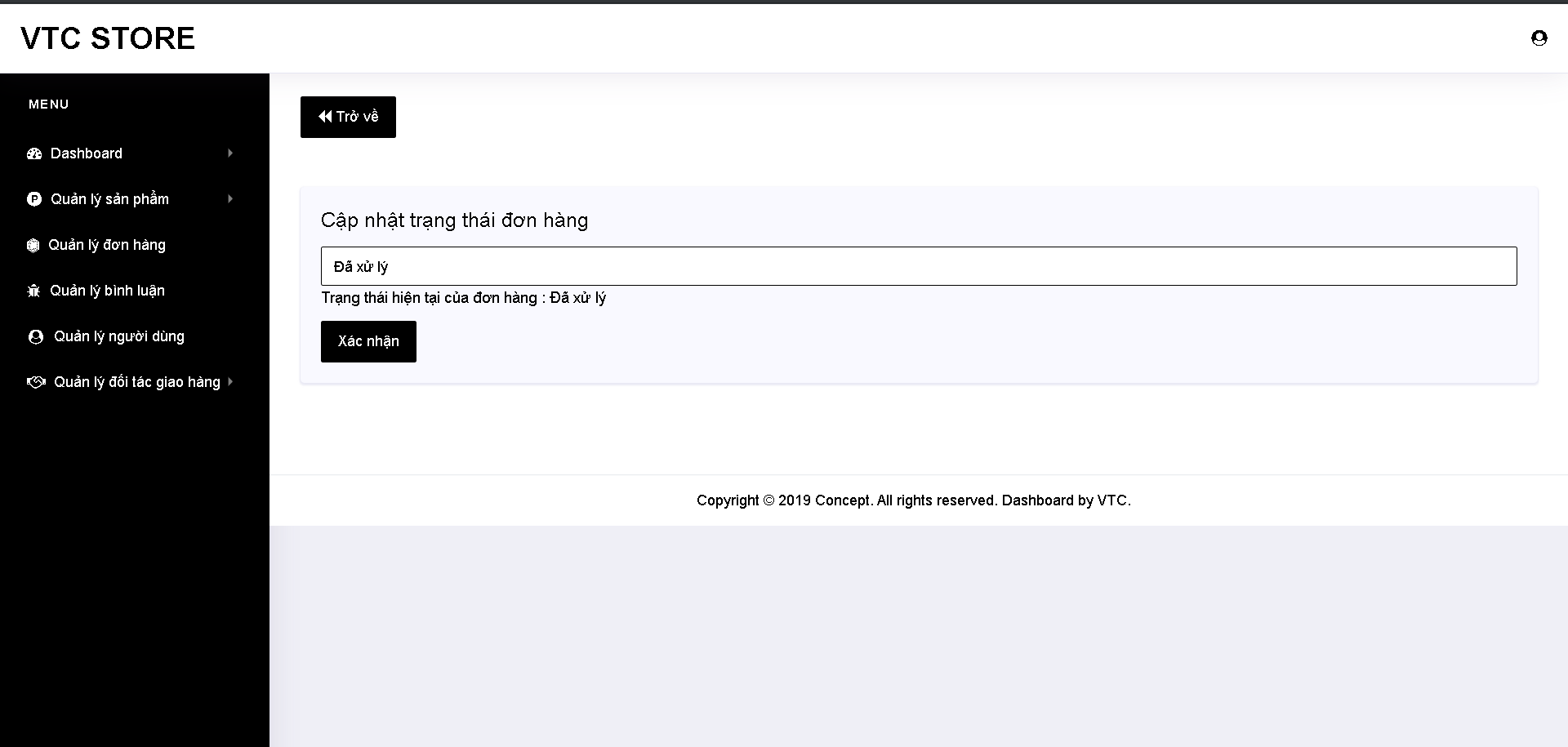
* Delivery status:



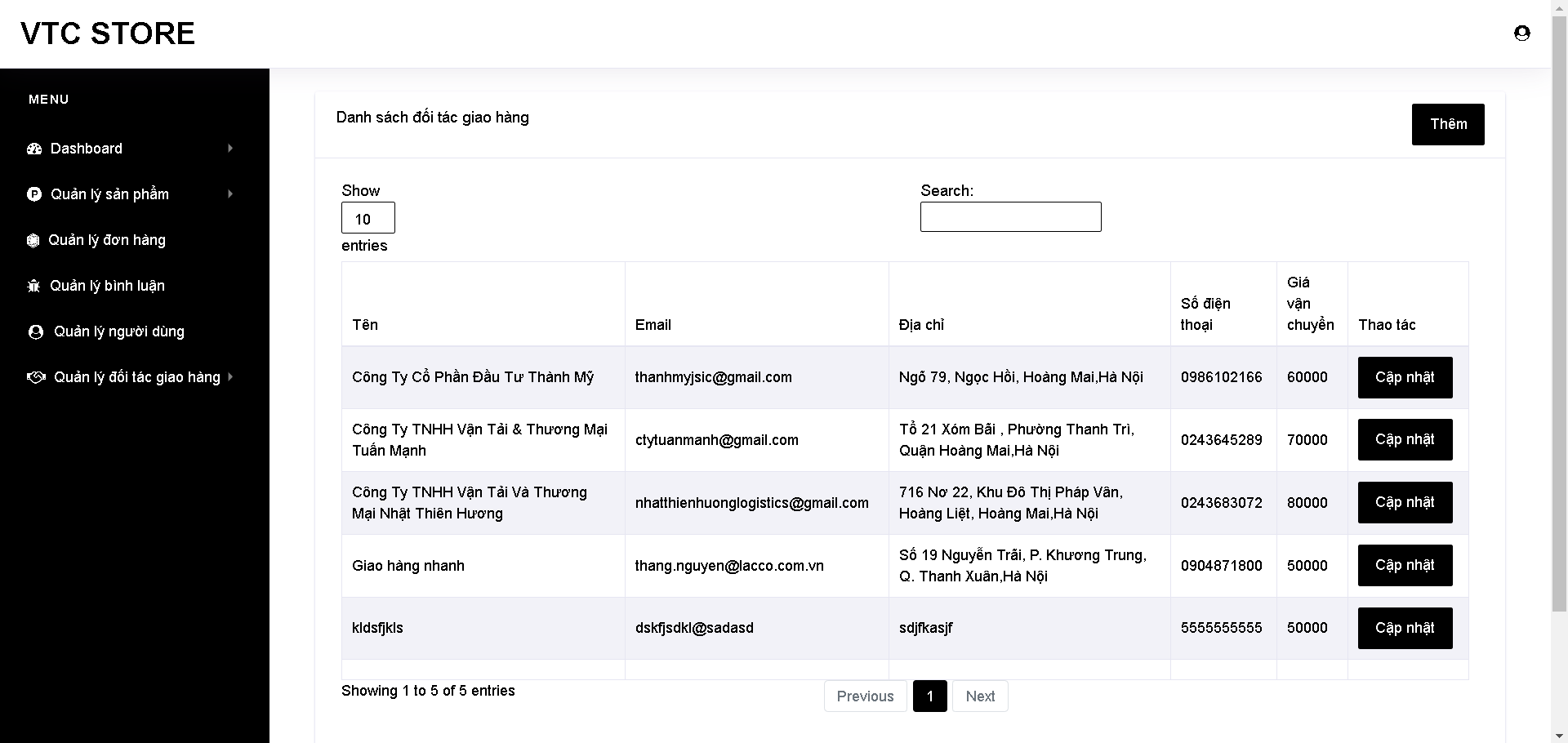
* Checking:



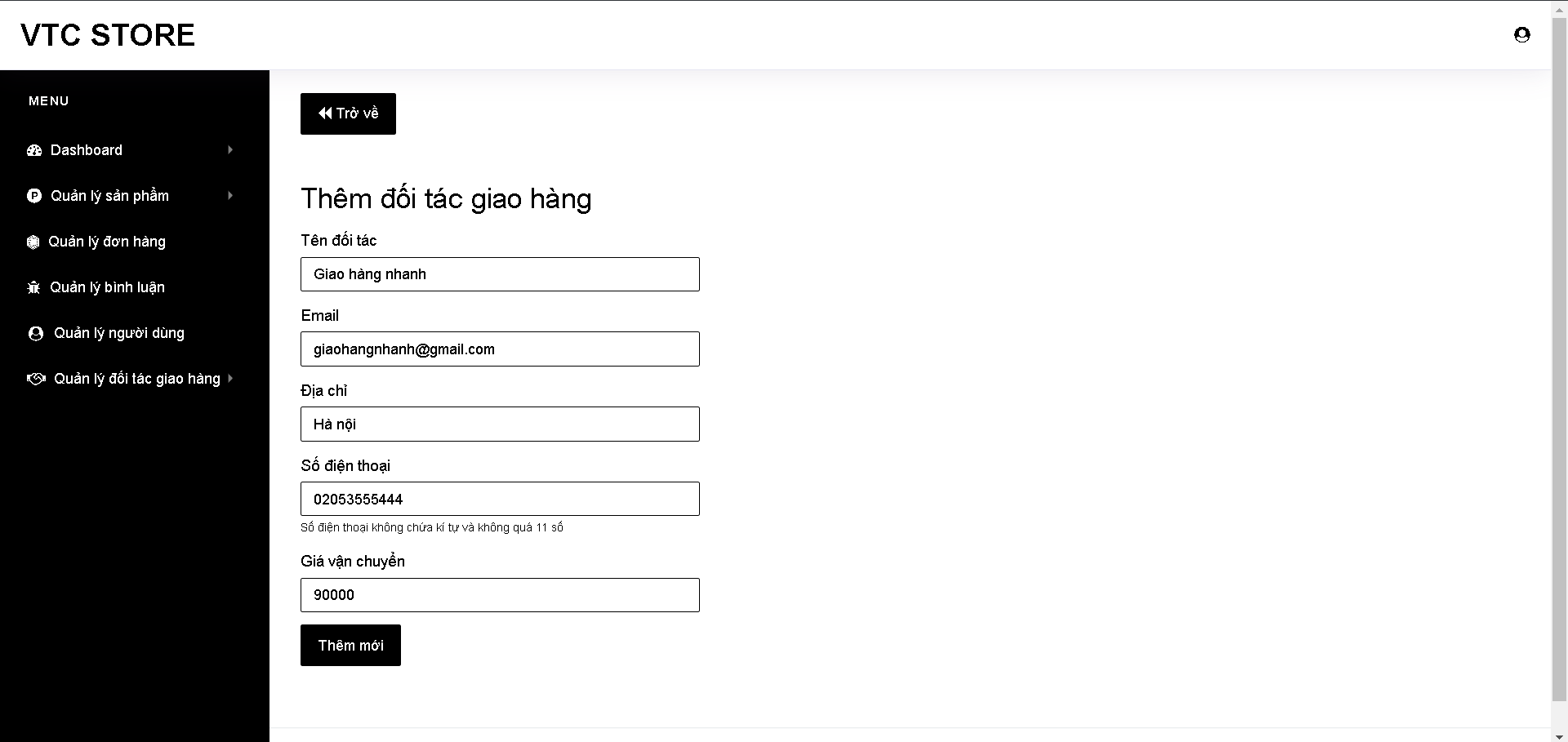
* Update checking:



* View partners:



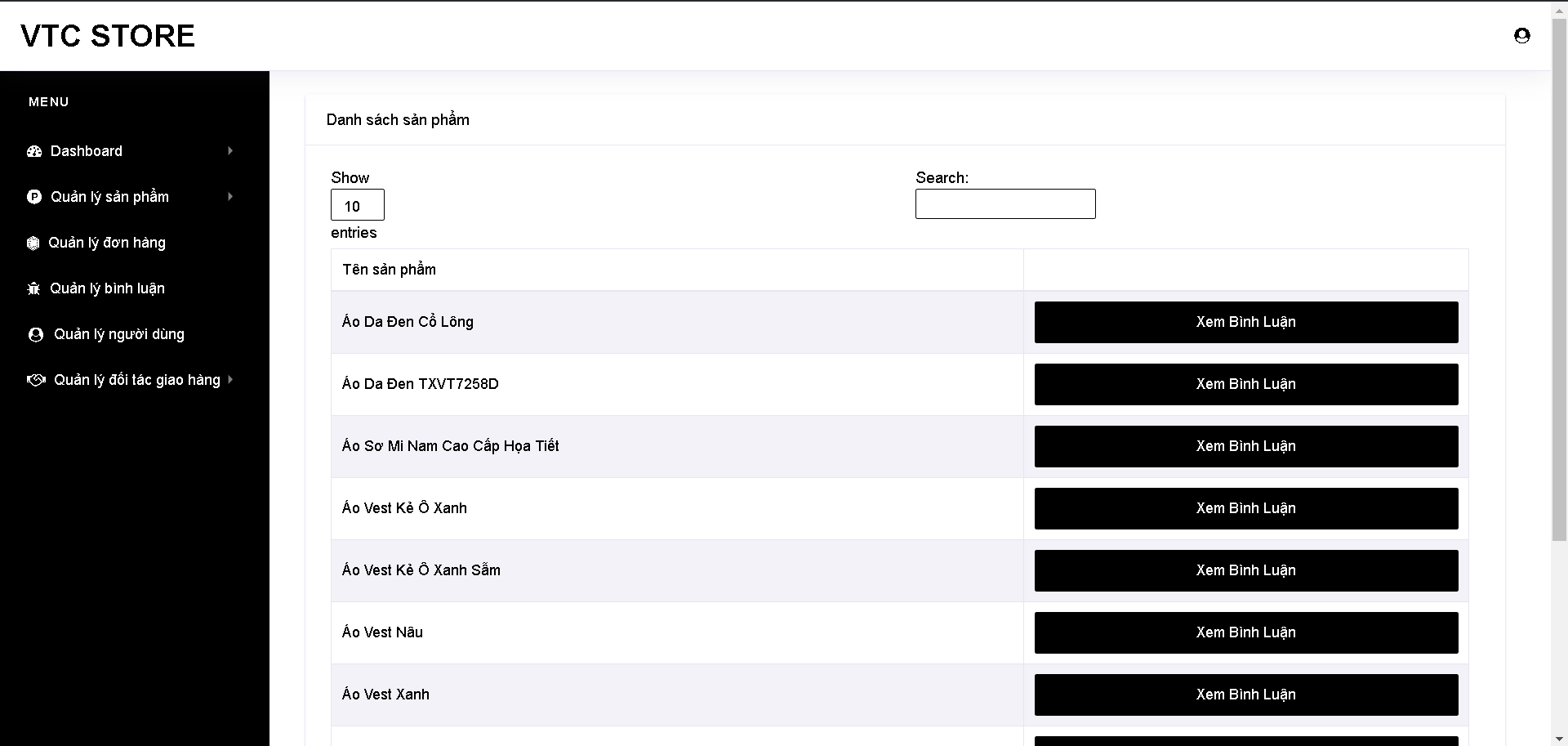
* Add partner:



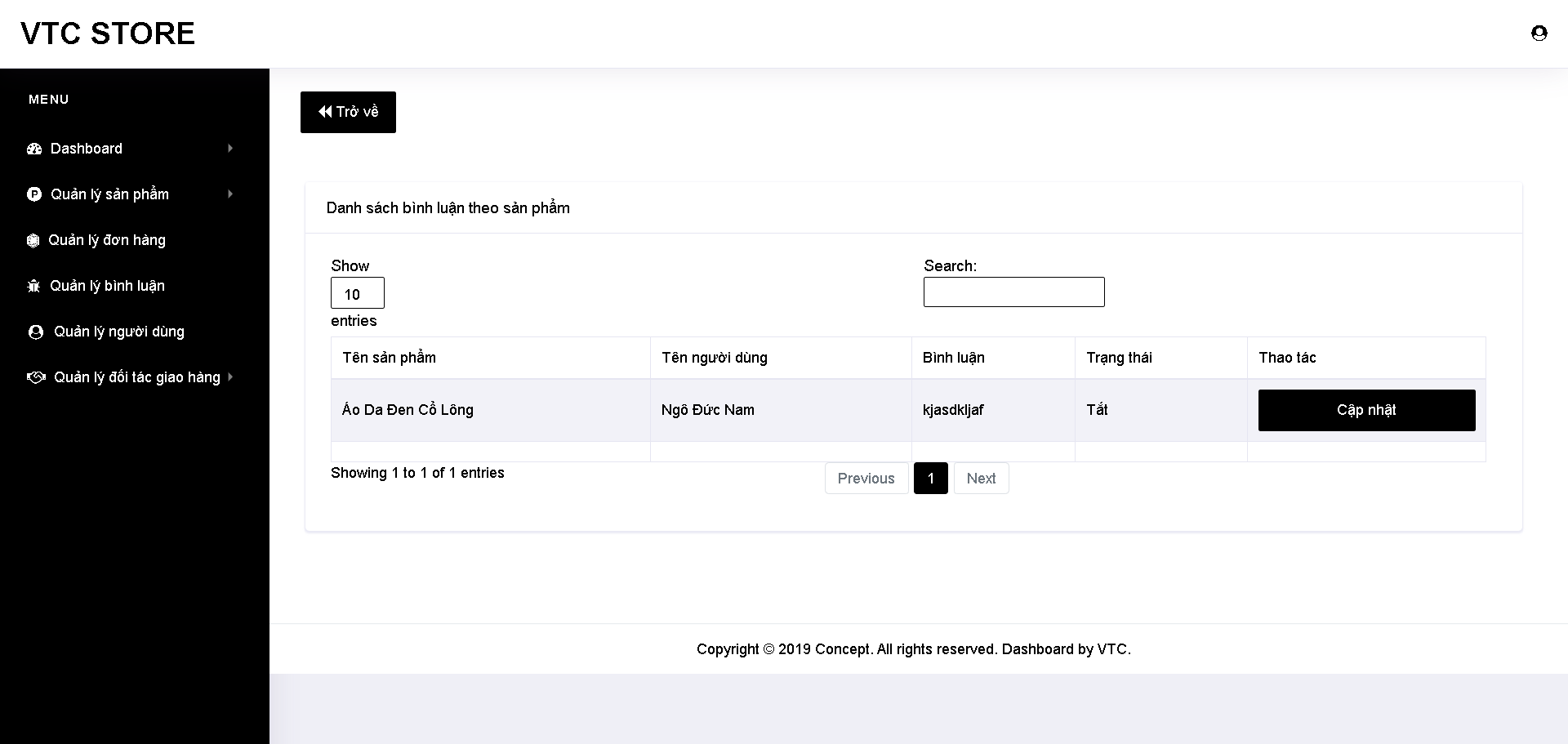
* Update partner:



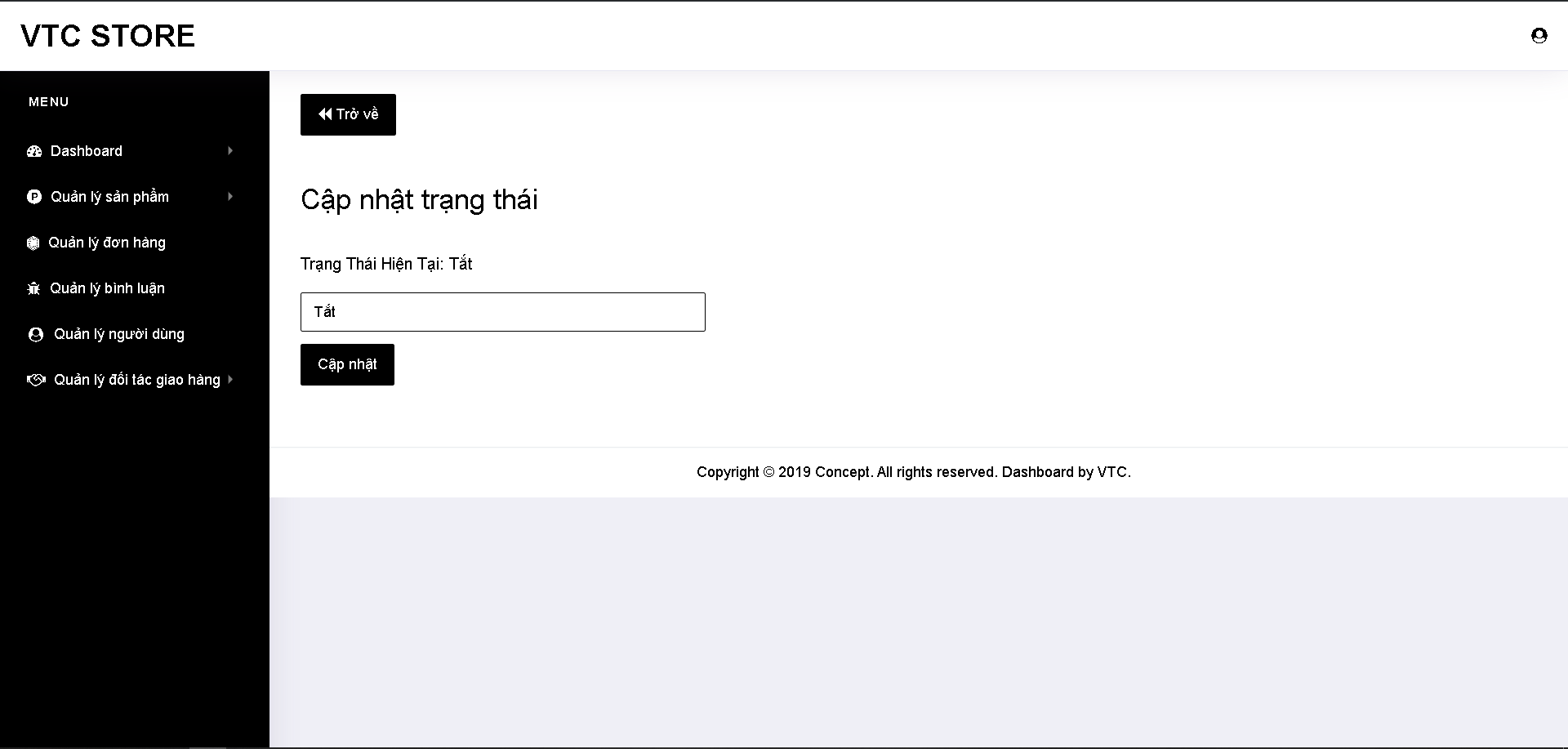
* View comments:



* View comment details:



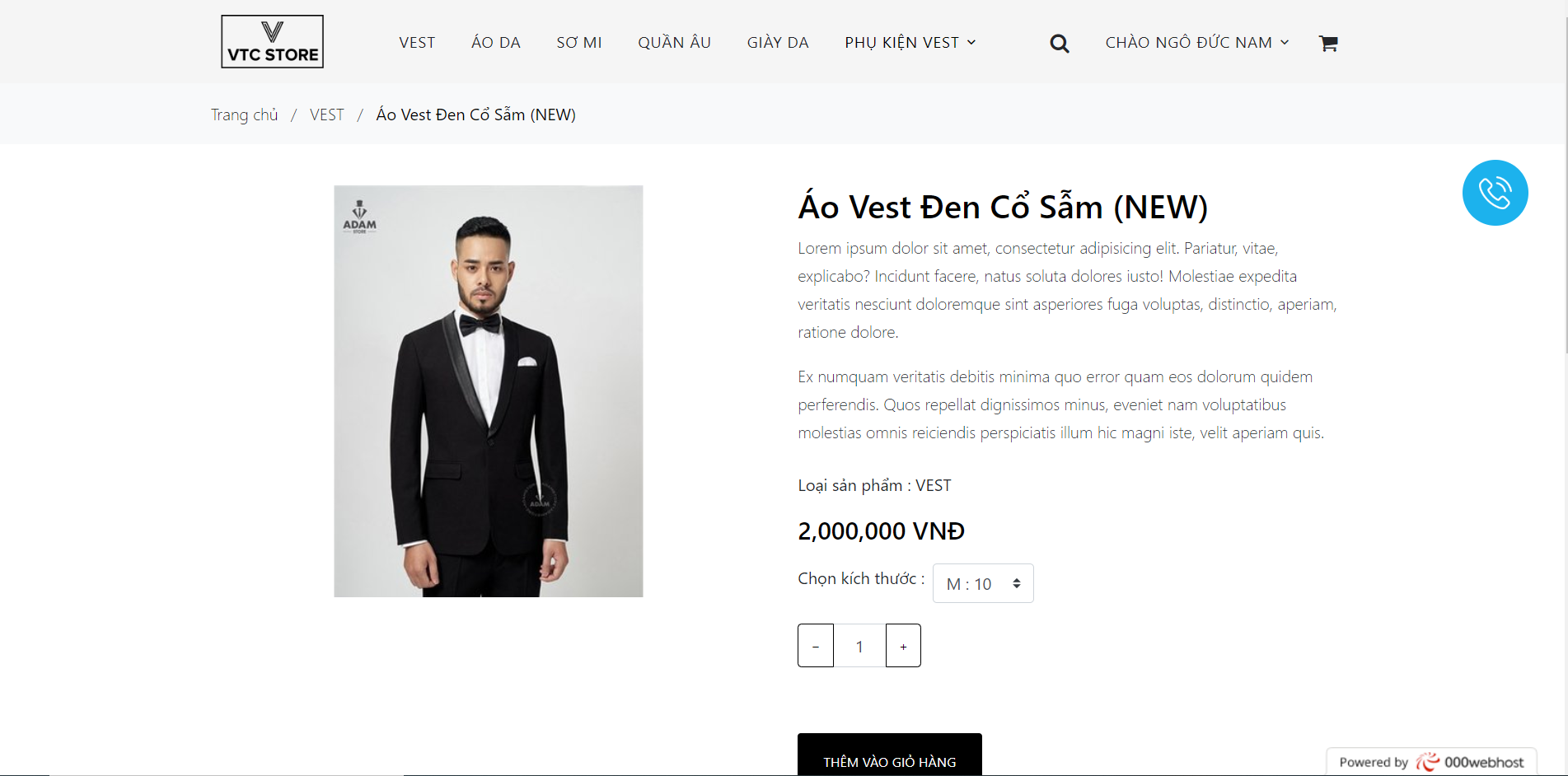
* Active comment:



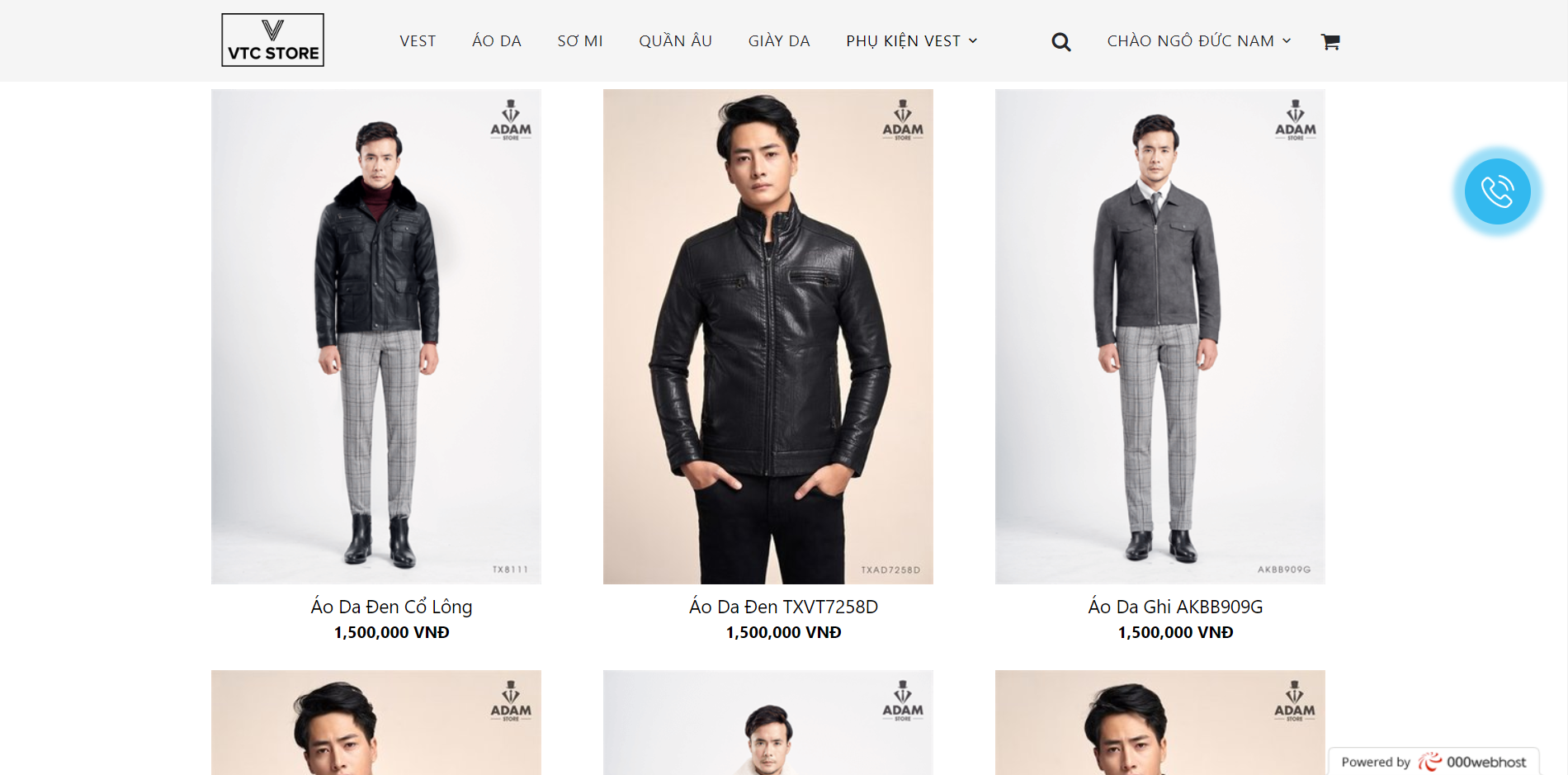
* Search products:



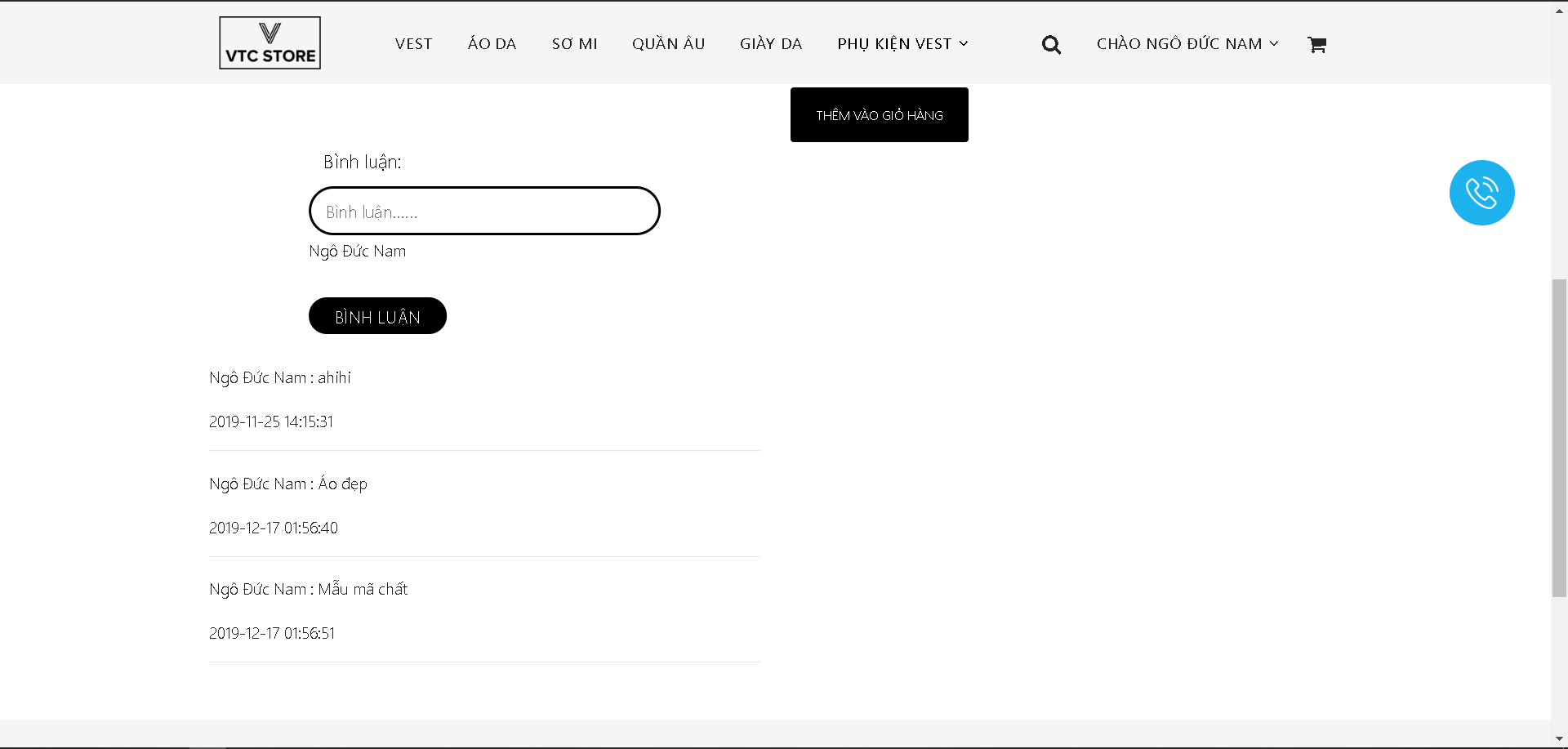
* View product details:



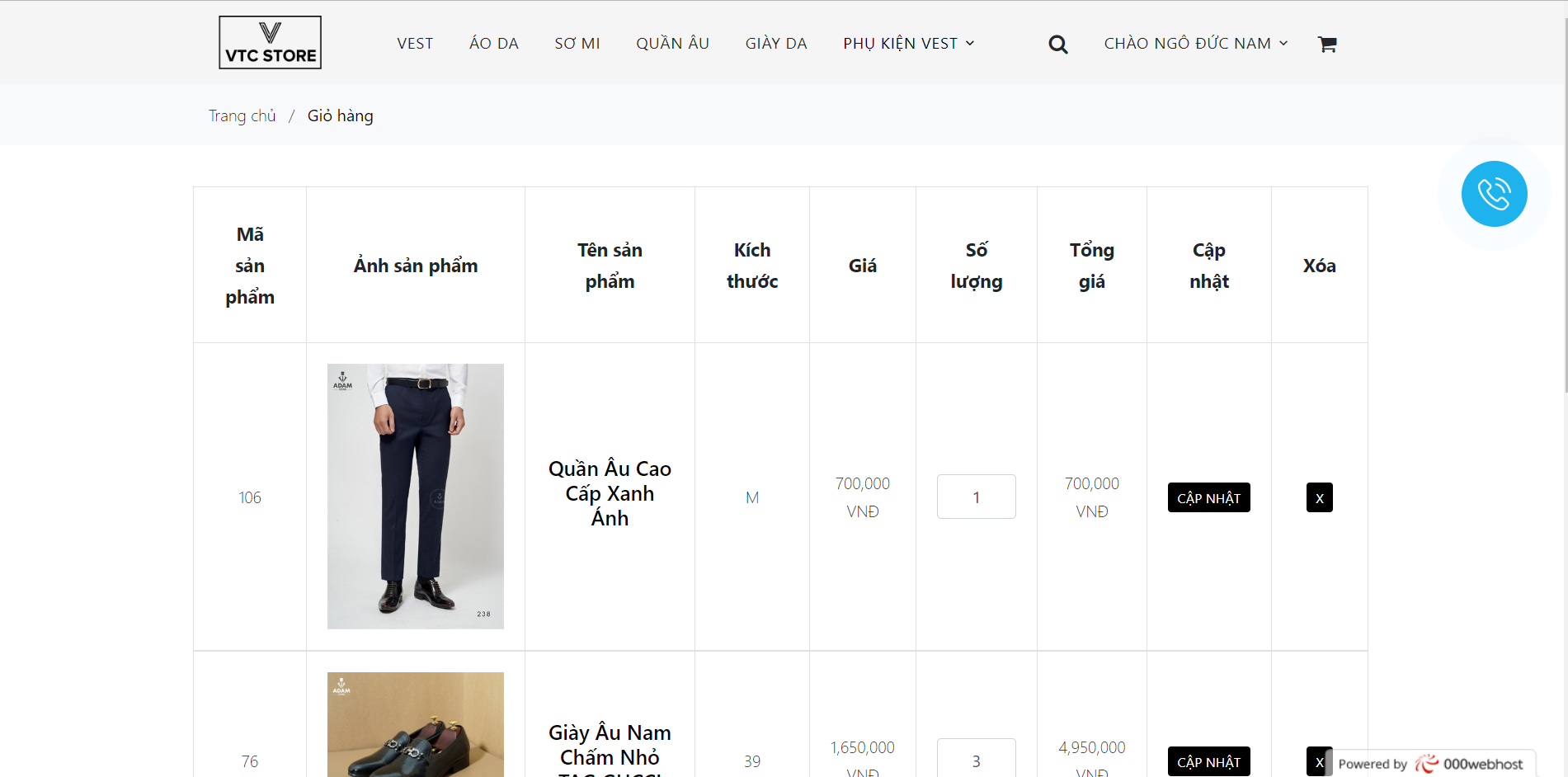
* View product by categories:



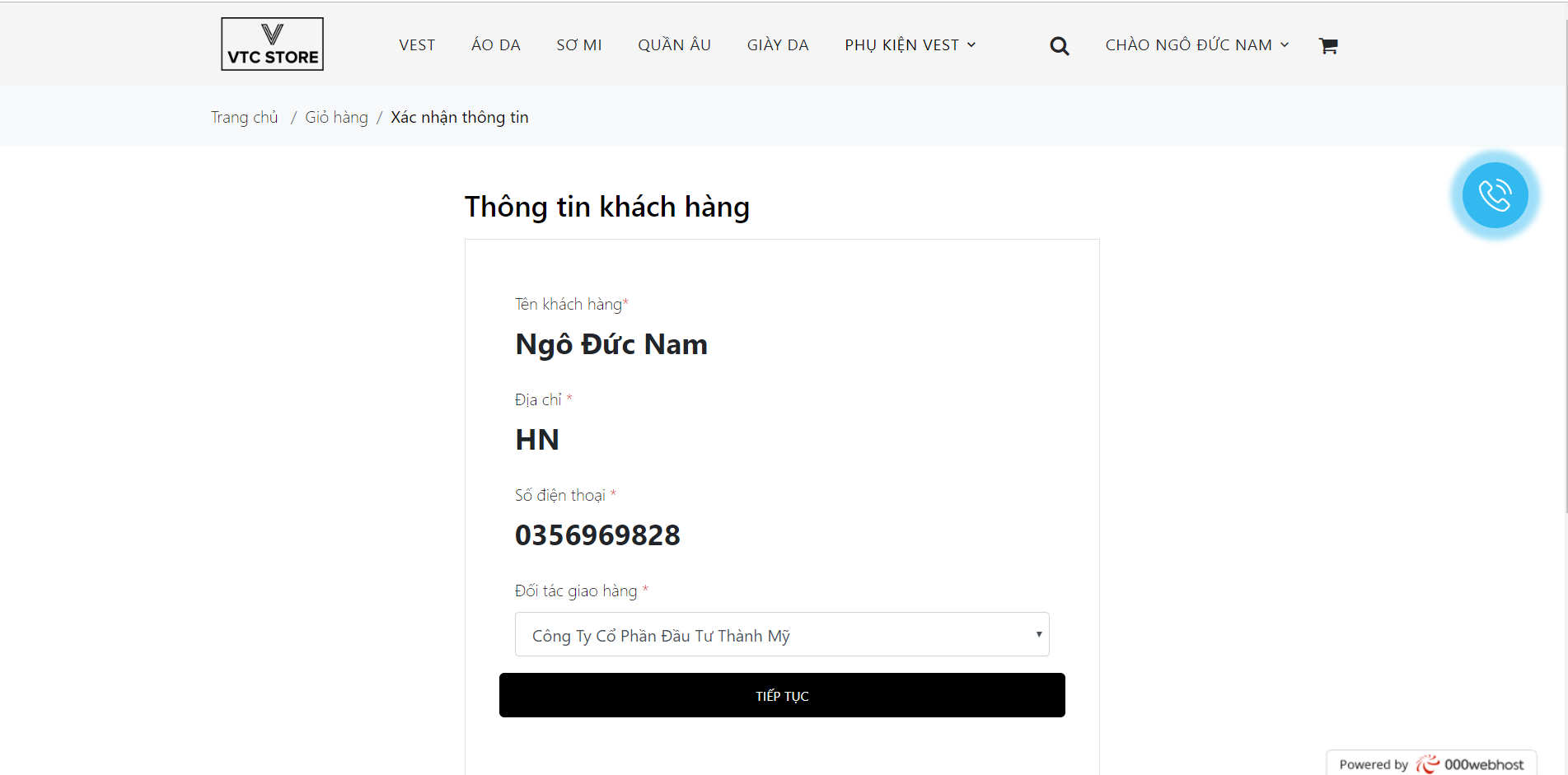
* Comment:



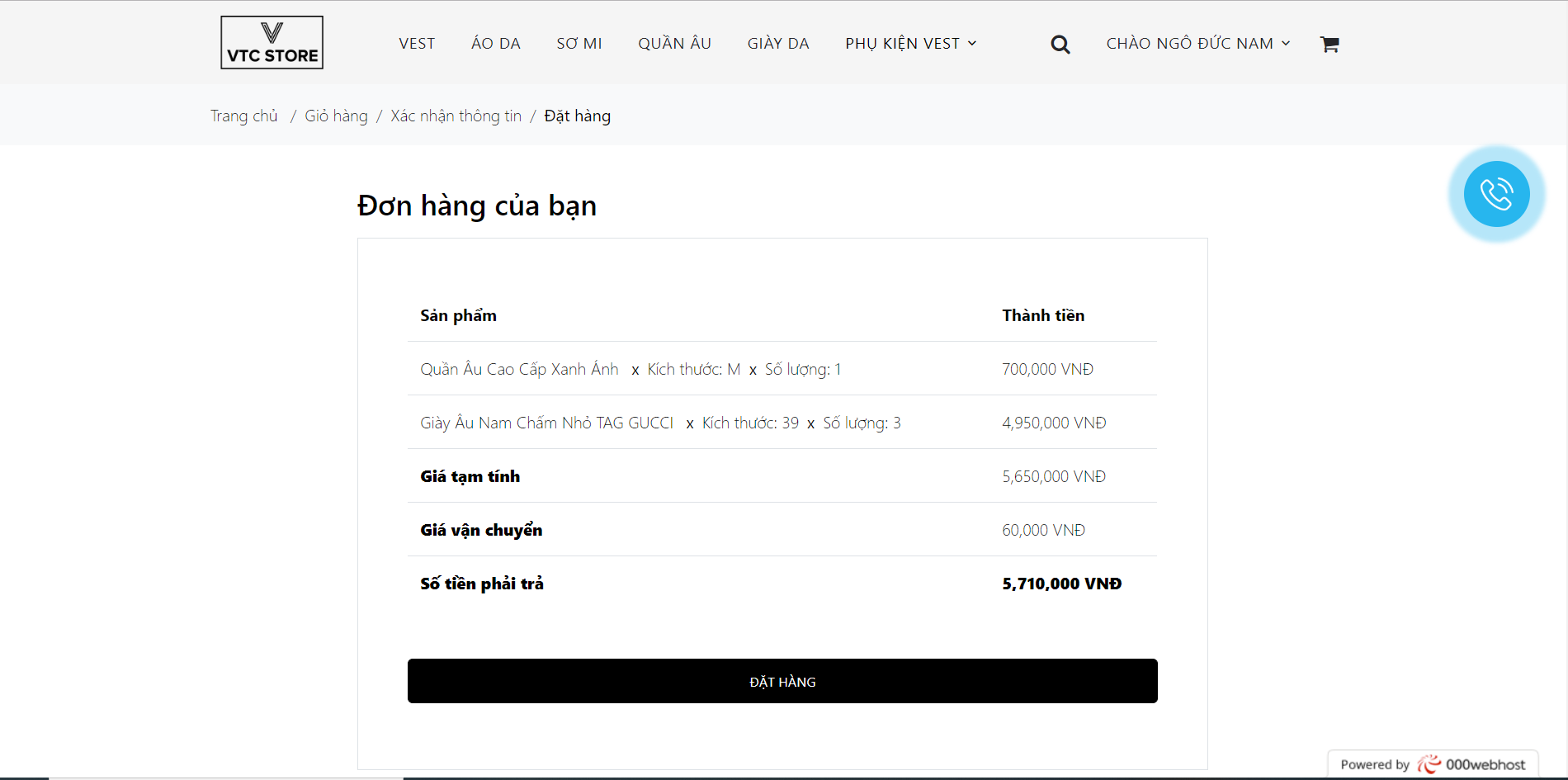
* View cart:



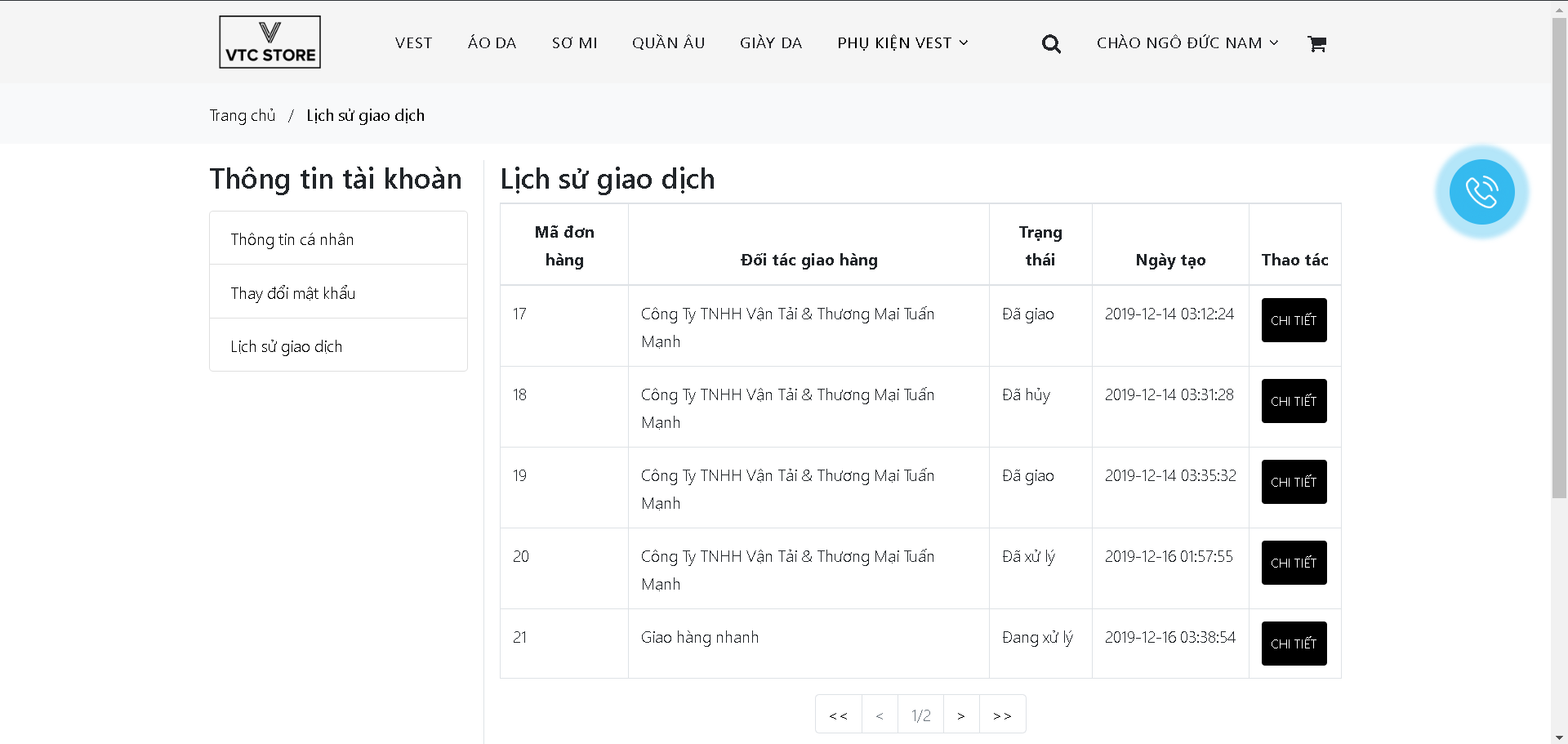
* Checkout:



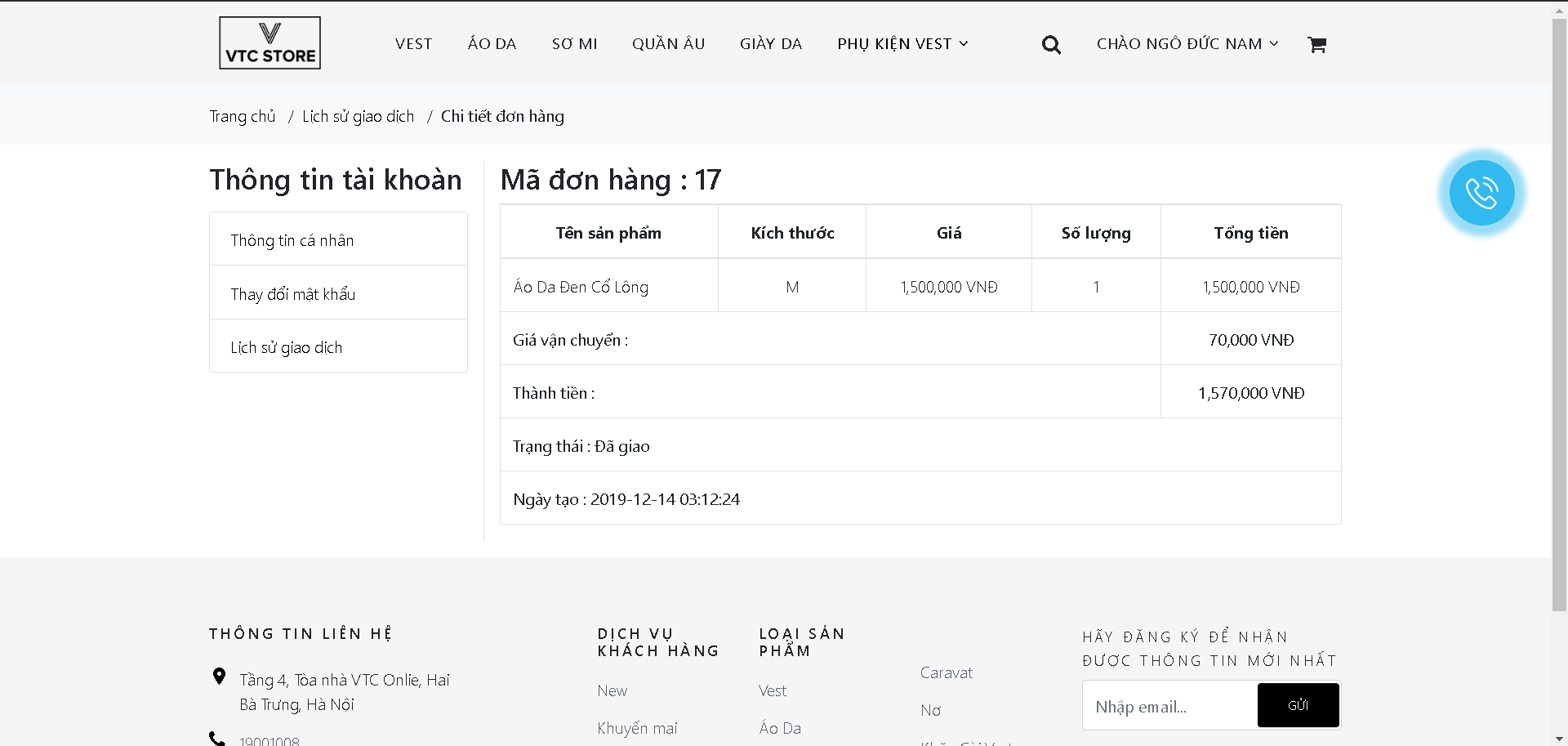
* Confirm:



* History transaction:

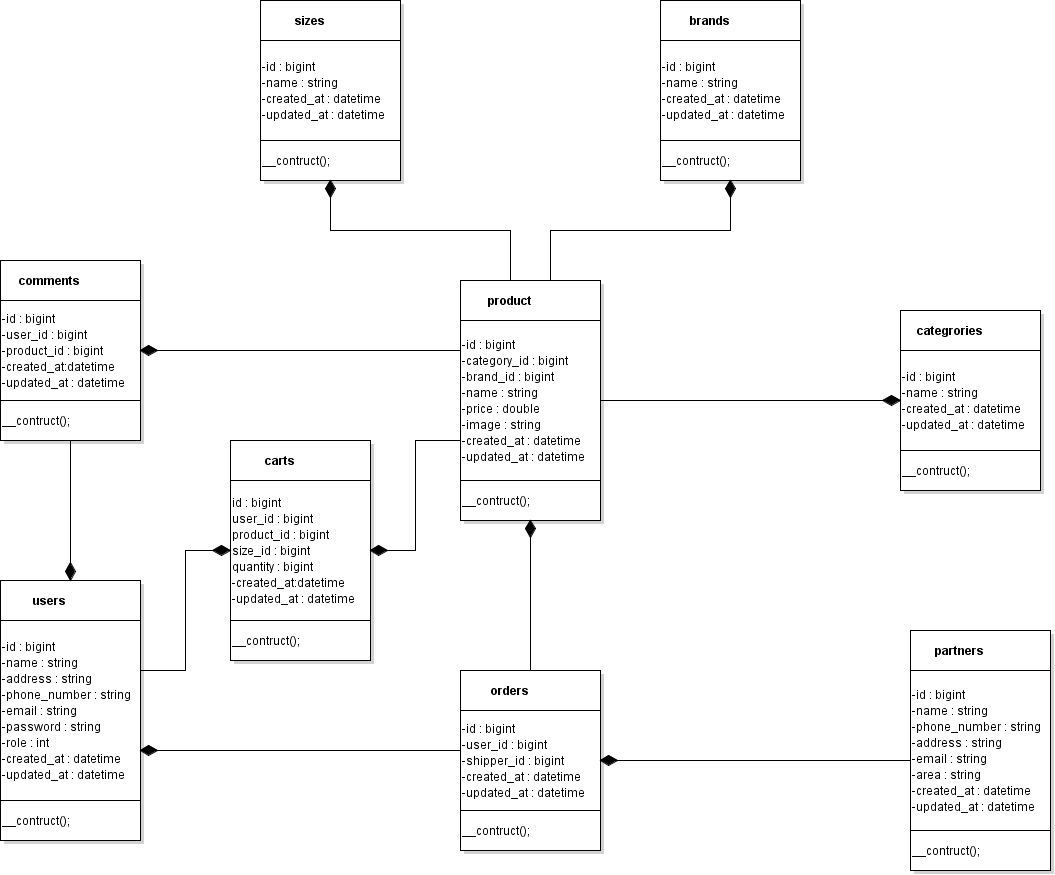


* View order details:

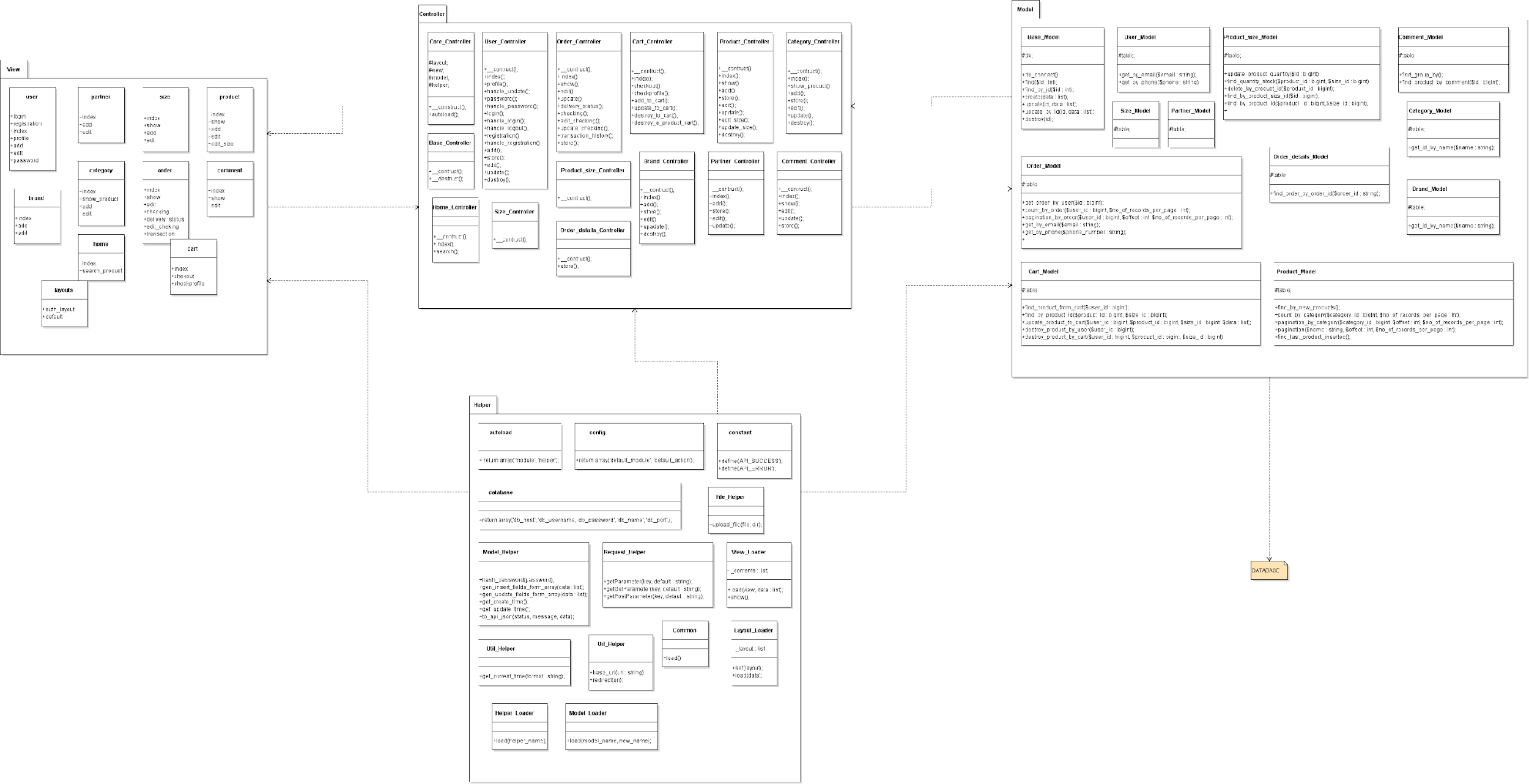


## Class Diagram

* Class Diagram:

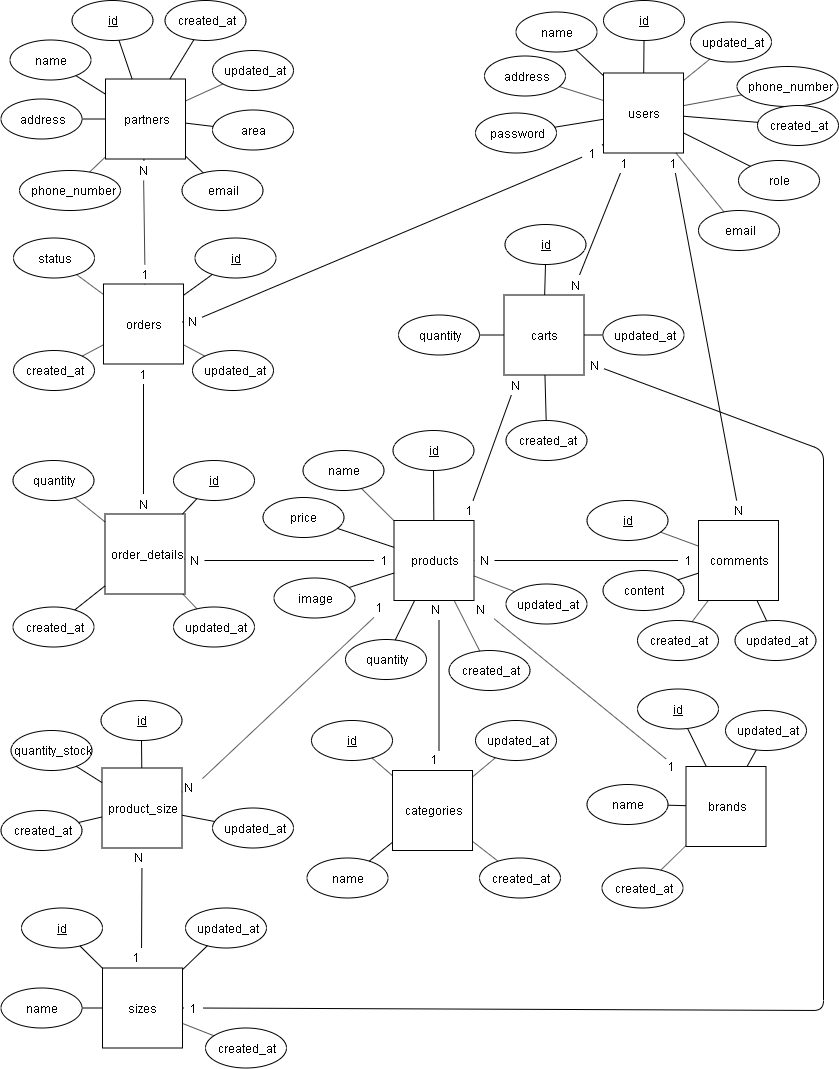


* Mô hình MVC:

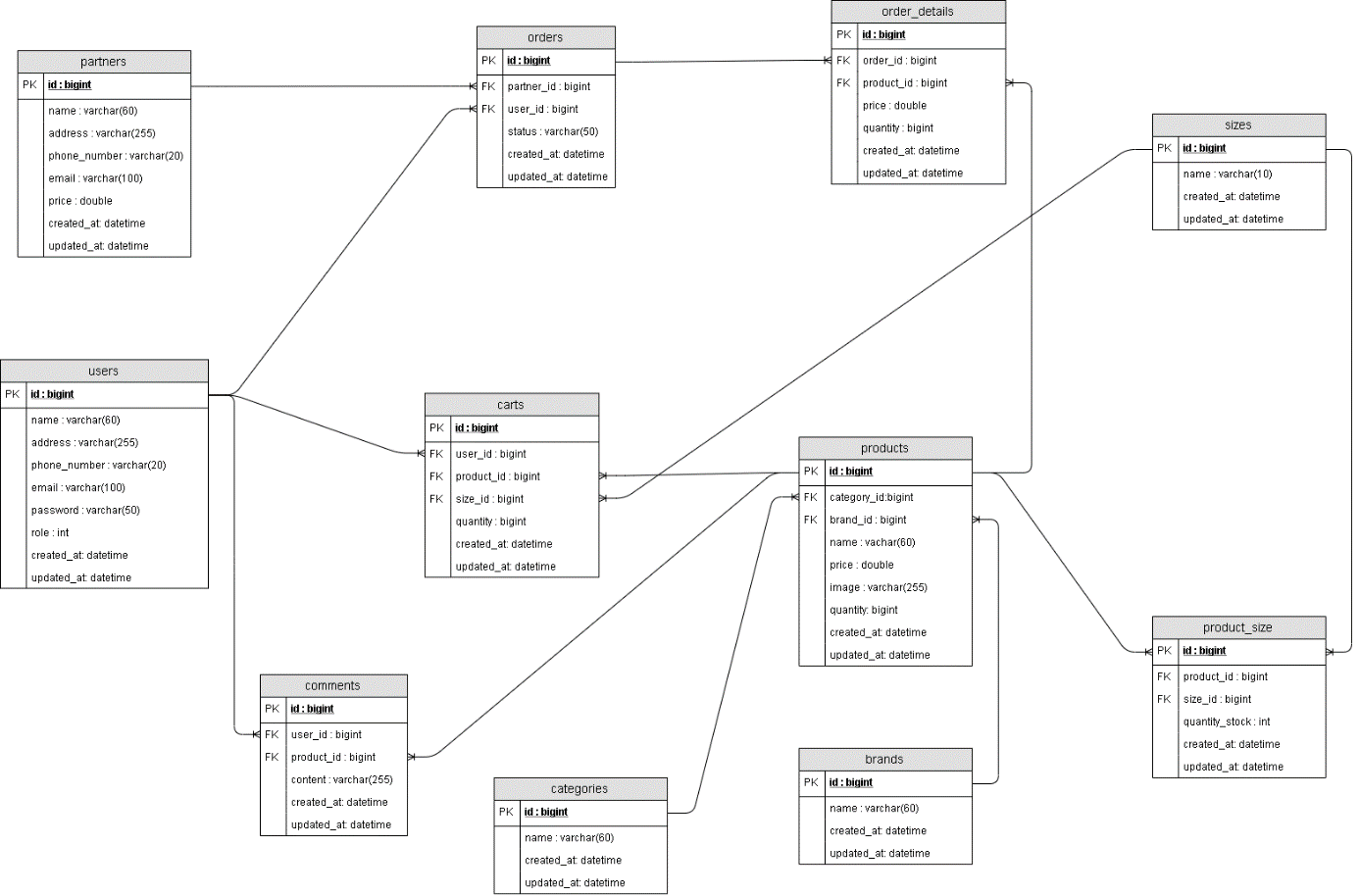


## Database design

* 1. Entity Relationship Diagram



Or:



* 1. Detailed database design

|  |  |  |  |
| --- | --- | --- | --- |
| **Users** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | User identify |
| Name | Varchar(60) | Not null | User name |
| Password | Varchar(50) | Not null | User password |
| Phone\_number | Varchar(20) | Unique, not null | User phone number |
| Address | Varchar(255) | Not null | User address |
| Email | Varchar(100) | Unique, not null | User email |
| Role | Int | Not null | User role |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Products** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Product identify |
| Brand\_id | Bigint | Foreign key, not null | Brand identify |
| Category\_id | Bigint | Foreign key, not null | Category identify |
| Name | Varchar(60) | Not null | Product name |
| Price | Double | Not null | Product price |
| Image | Varchar(255) | Not null | Product image |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Partners** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Partner identify |
| Name | Varchar(60) | Not null | Partner name |
| Address | Varchar(255) | Not null | Partner address |
| Phome\_number | Varchar(20) | Unique, not null | Partner phone number |
| Email | Varchar(255) | Unique, not null | Partner email |
| Price | Double | Not null | Price shipping |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Orders** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Order identify |
| Partner\_id | Bigint | Foreign key, not null | Partner identify |
| User\_id | Bigint | Foreign key, not null | User idetify |
| Status | Varchar(50) | Not null | Status order |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Order details** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key | Order details identify |
| Order\_id | Bigint | Foreign key | Order identify |
| Product\_id | Bigint | Foreign key | Product identify |
| Price | Double | Not null | Total price products |
| Quantity | Bigint | Not null | Total quantity a product |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Product\_Size** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, foreign key | Product identify |
| Product\_id | Bigint | Primary Key, foreign key | Product identify |
| Size\_id | Bigint | Primary Key, foreign key | Size identify |
| Quantity\_stock | Bigint | Not null | Total quantity size a product |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sizes** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Size identify |
| Name | Varchar(10) | Not null | Size name |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Category identify |
| Name | Varchar(60) | Not null | Category name |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Brands** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Brand identify |
| Name | Varchar(60) | Not null | Brand name |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Comments** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Feedback identify |
| User\_id | Bigint | Foreign key, not null | Customer identify |
| Product\_id | Bigint | Foreign key, not null | Product identify |
| Comment | Varchar(255) |  | Comment on the product |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

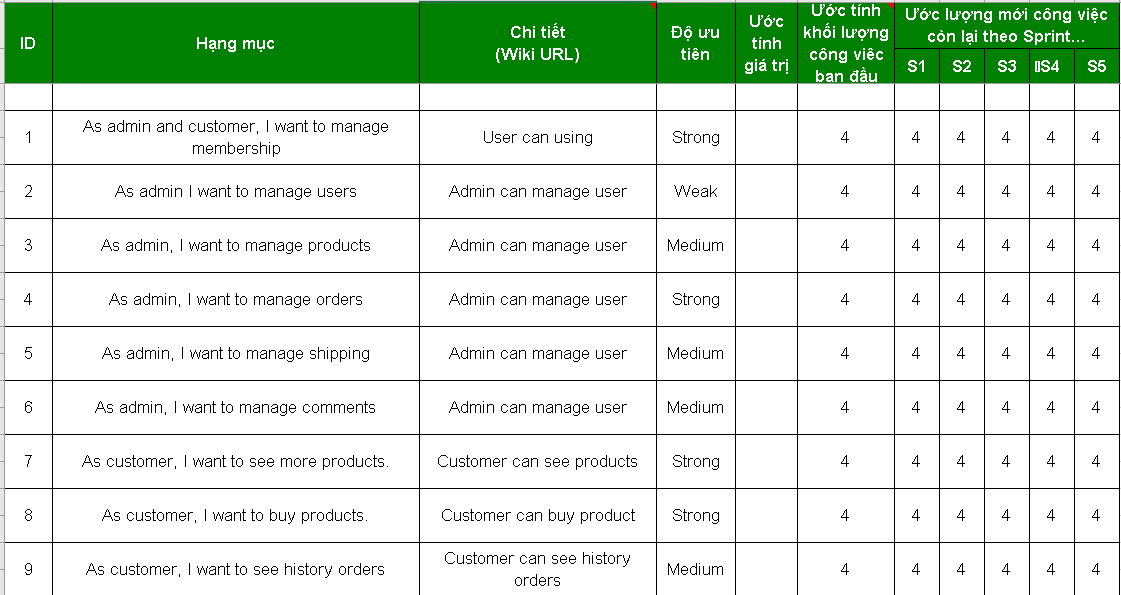
|  |  |  |  |
| --- | --- | --- | --- |
| **Carts** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | Bigint | Primary Key, auto\_increment | Feedback identify |
| User\_id | Bigint | Foreign key, not null | Customer identify |
| Product\_id | Bigint | Foreign key, not null | Product identify |
| Size\_id | Bigint | Foreign key, not null | Size identify |
| Quantity | Bigint | Not null | Total quantity a product |
| Created\_at | Datetime | Not null | Date created |
| Updated\_at | Datetime | Not null | Date updated |

# Test

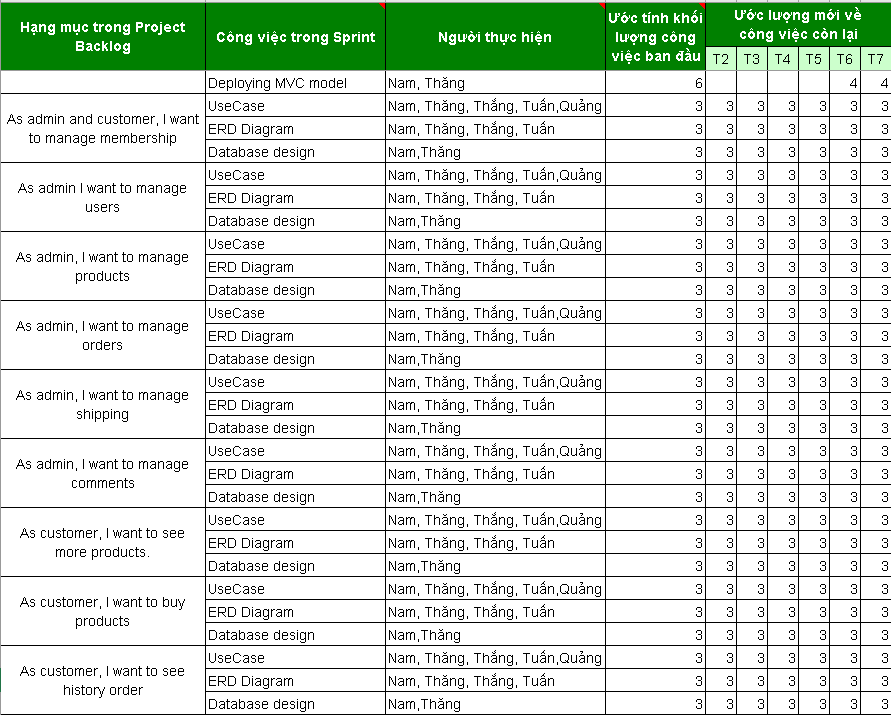
|  |  |
| --- | --- |
| **Test Case Number** |  |
| **Test Case Name** |  |
| **Test Case Description** |  |
| **Preconditions** |  |
| **Test Case Input** |  |
| **Test Case Expected Output** |  |
| **Default Value Preverving** |  |

# Assign work to each team member

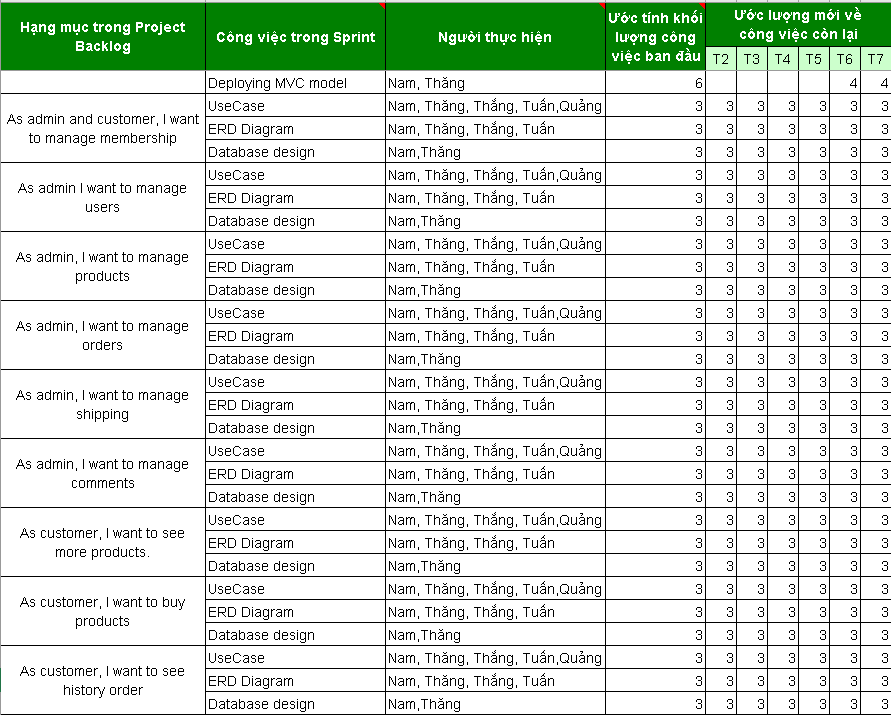
* Product Backlog:



* Sprint1:



* Sprint2:



* Sprint3:



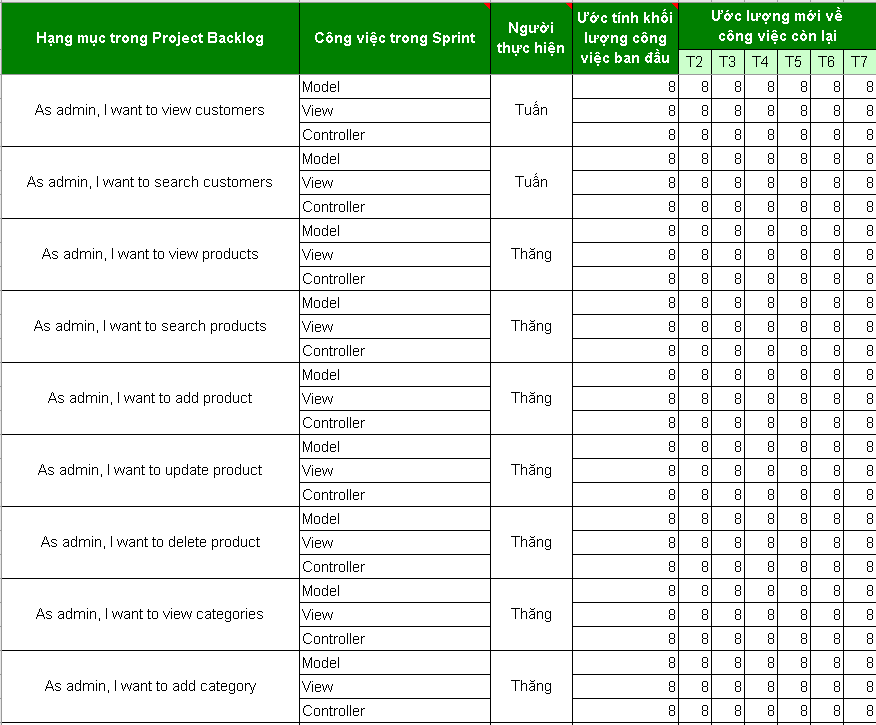


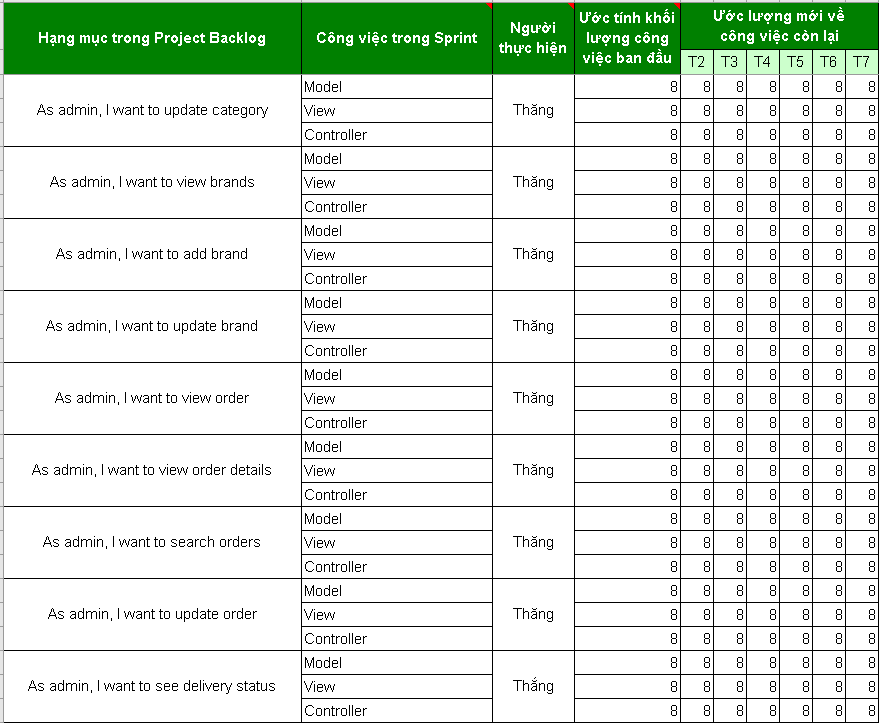


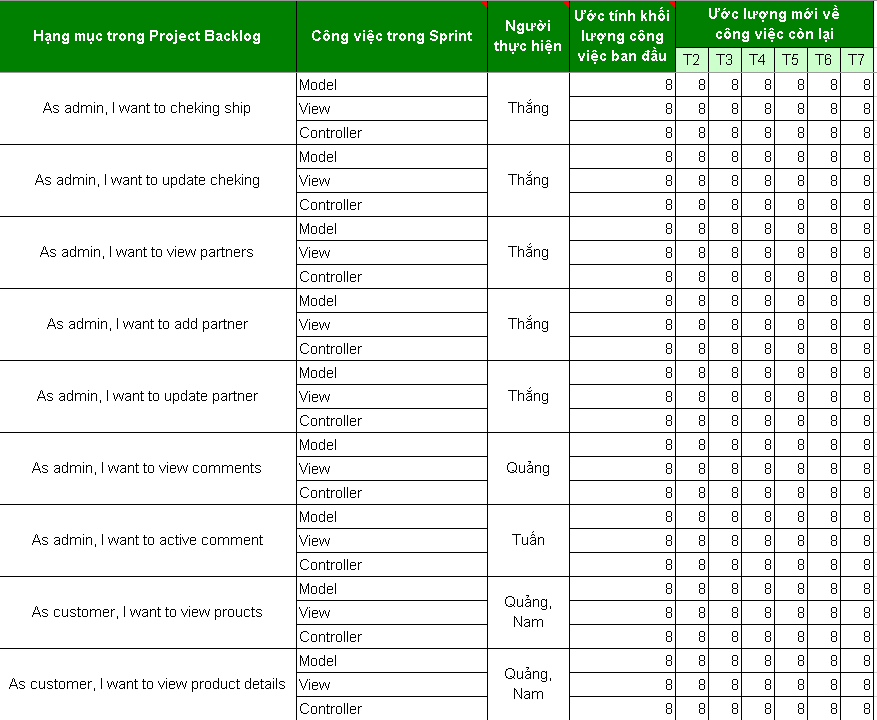


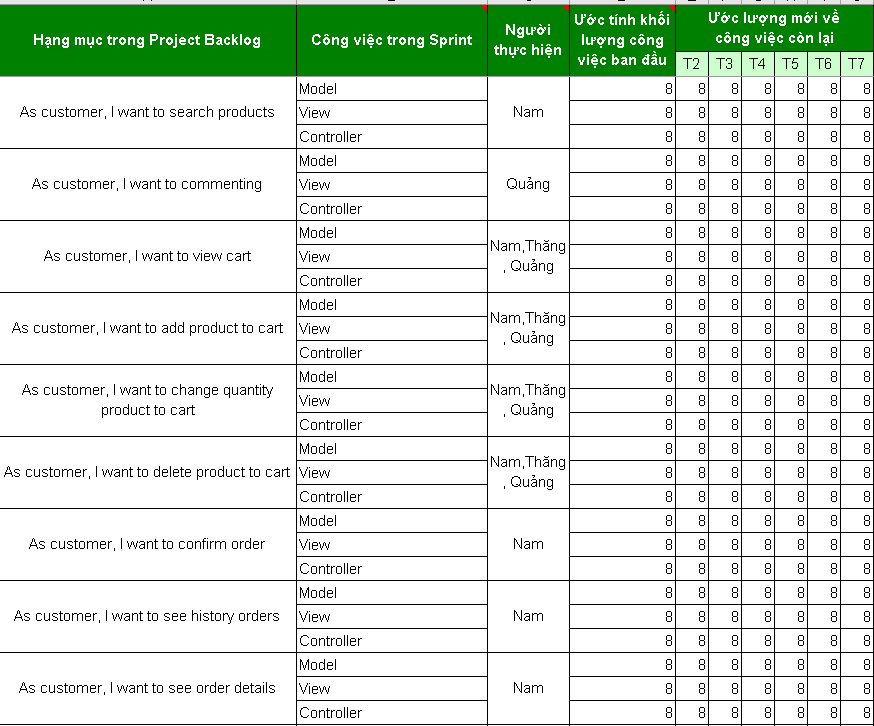
* Sprint4:





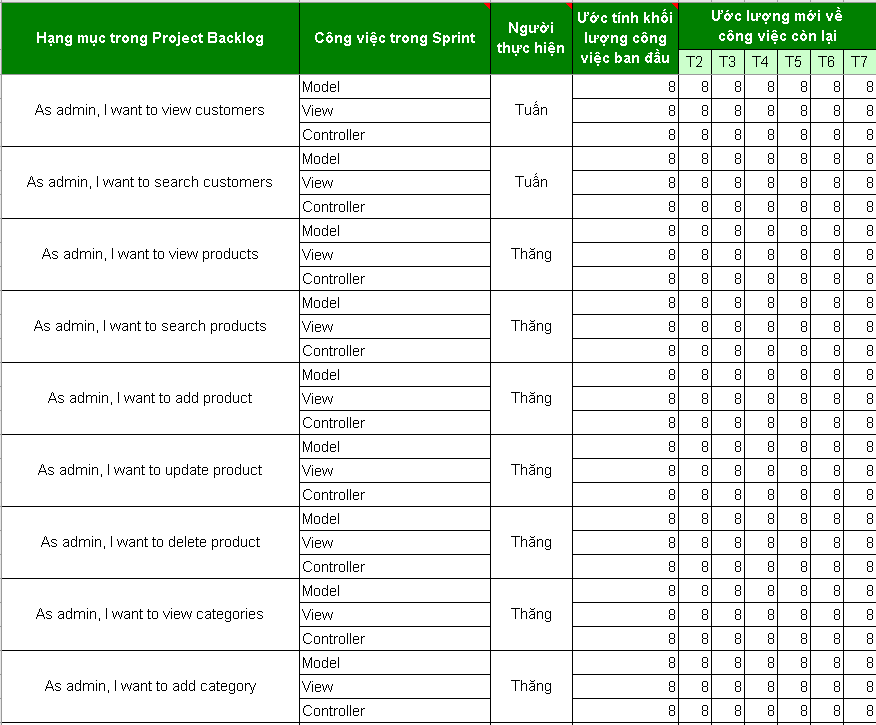


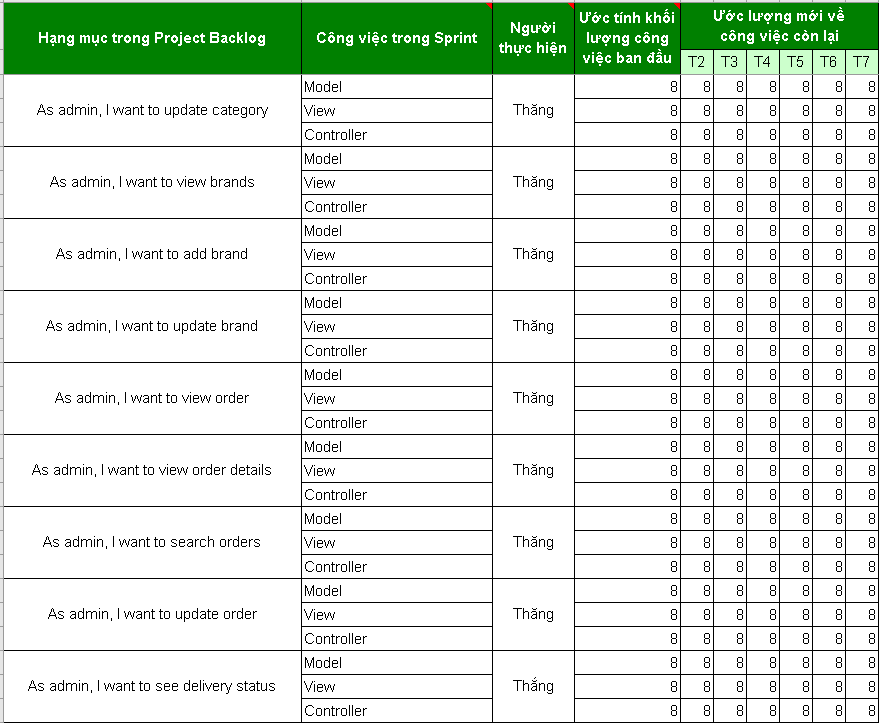


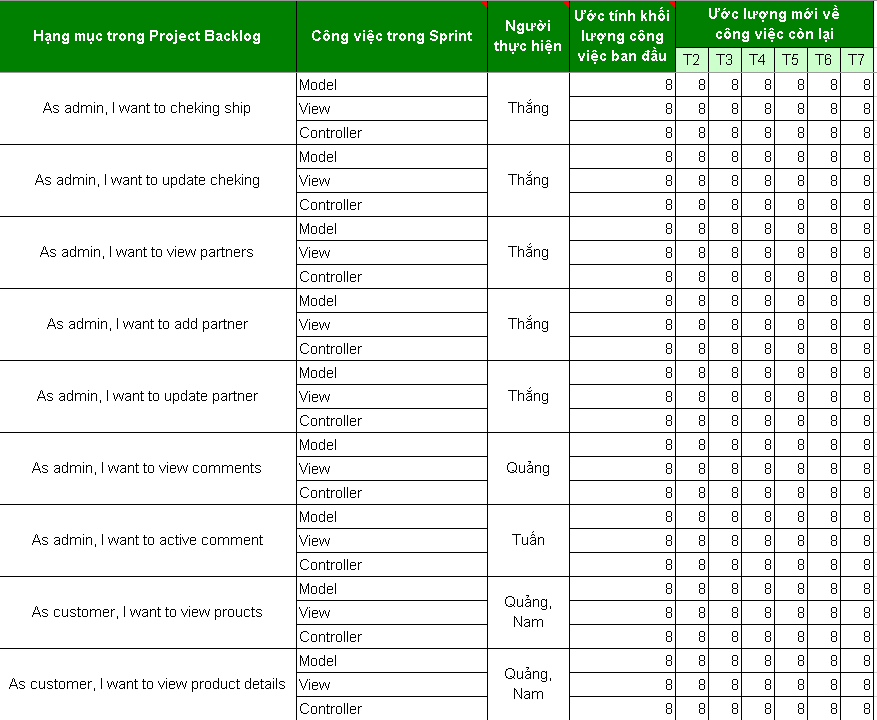


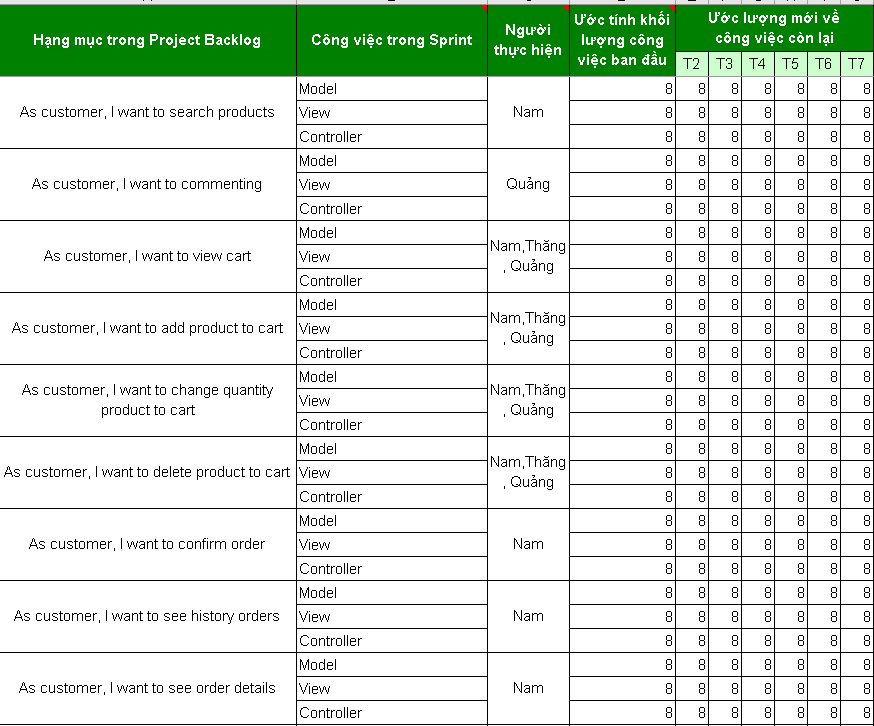
* Sprint5:



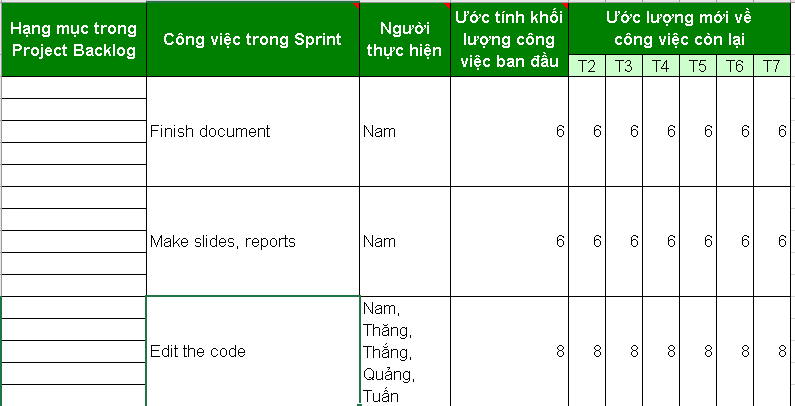








* Sprint6:



* Log:



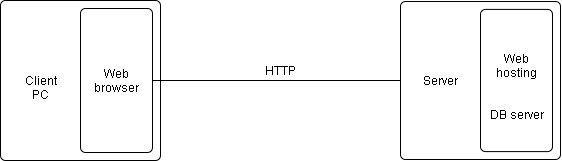


* Project-info:



# Installation Instructions

## Deployment Diagram



## Installation steps

* Environment settings:
* Register the domain name, then go to the file manager to upload the source code to web hosting, register the server database
* Database settings:
* Check the configuration of the database, then adjust the properties of the database, go to the database management section to import the sql file into the server.

# Contents

***References***

* [*https://stackoverflow.com/*](https://stackoverflow.com/)
* [*https://tiki.vn/*](https://tiki.vn/)
* *https://shopee.vn/*

***Another problem***

* ***Results***
* *Learn the working process*
* *Relational database design*
* *Draw Activity Diagram diagram, Sequence Diagram, Class diagram*
* *Analyze and solve problems logically*
* ***Limit***
* *Not solved the whole problem*
* *Also limited in knowledge*
* *Teamwork is not good*