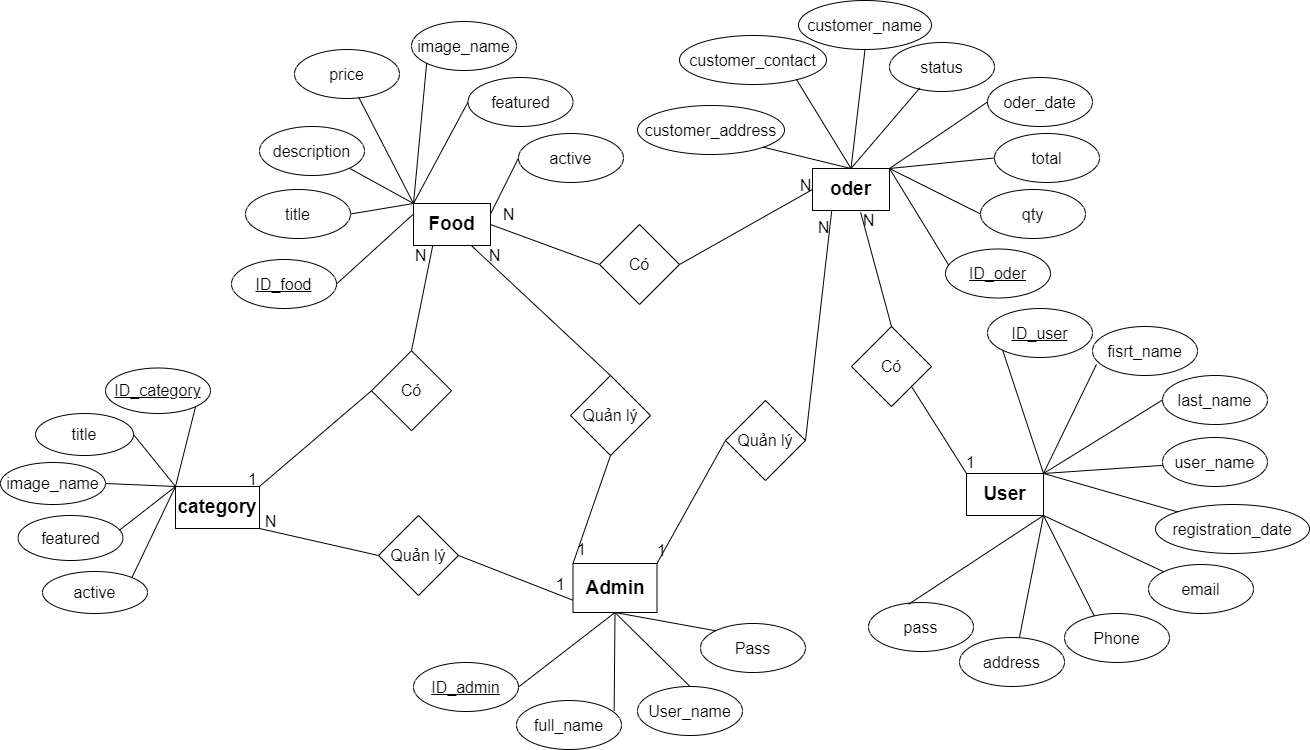
|  |
| --- |
| **TRƯỜNG ĐẠI HỌC THỦY LỢI**  **KHOA CÔNG NGHỆ THÔNG TIN**    **CÔNG NGHỆ PHẦN MỀM**  ***Đề tài:* Thiết kế hệ thống đặt đồ ăn nhanh**  Giảng viên hướng dẫn : Nguyễn Thị Ngọc Hà  Lớp : 61HT  Sinh viên thực hiện : Ngô Hoàng Thủy  Nguyễn Thị Cẩm Tú  Phạm Xuân Thọ  Lê Đình Thọ  Trần Văn Sinh |

# **Nhiệm vụ 4: Sprint 2**

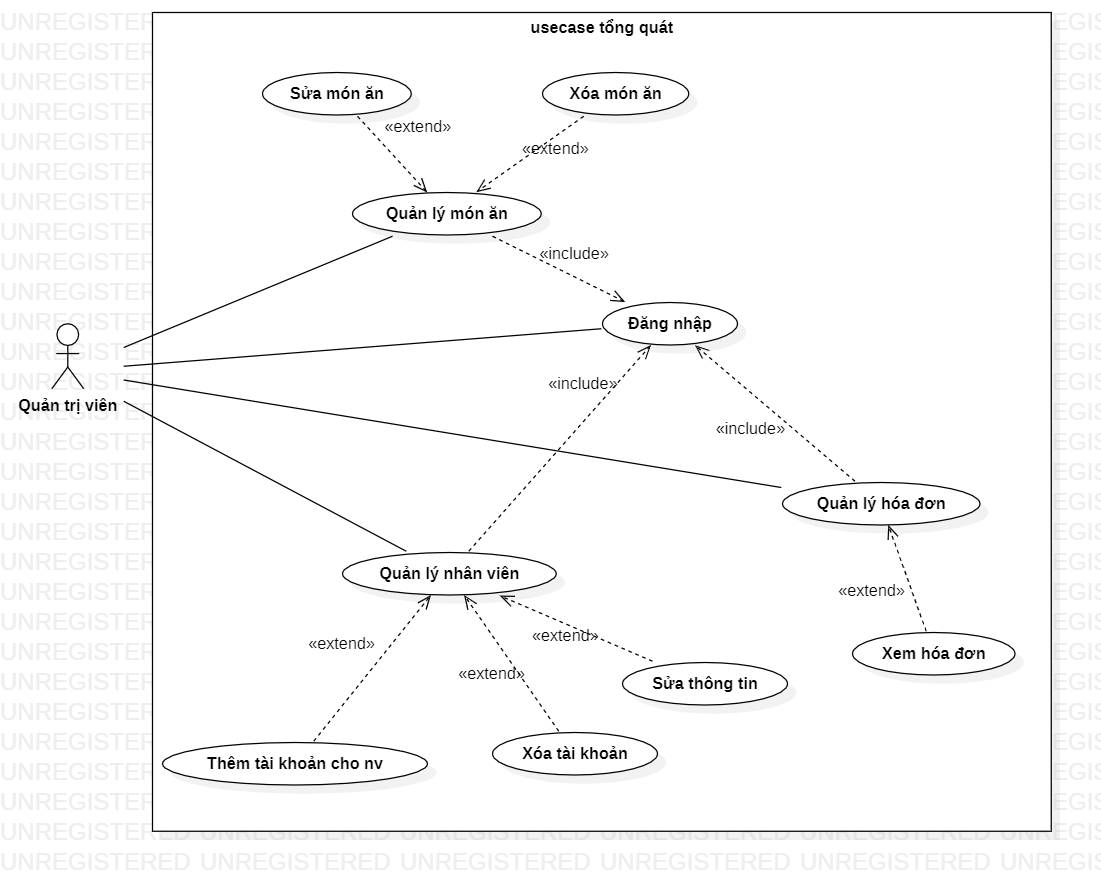
|  |  |  |
| --- | --- | --- |
| **ID** | **User Stories** | **Mức độ ưu tiên** |
| **US08** | Là **Người quản trị viên**, tôi muốn sửa thông tin sản phẩm để cập nhật thông tin phù hợp hơn. | **2** |
| **US09** | Là **Người quản trị viên**, tôi muốn xóa sản phẩm để loại bỏ những sản phẩm không còn kinh doanh nữa. | **2** |
| **US10** | Là **Người quản trị viên**, tôi muốn thêm tài khoản cho nhân viên để nhân viên có quyền làm việc trong hệ thống như quản lý. | **3** |
| **US11** | Là **Người quản trị viên**, tôi muốn sửa thông tin nhân viên để phù hợp với nhân viên. | **3** |
| **US12** | Là **Người quản trị viên**, tôi muốn xóa tài khoản của nhân viên để thu hồi lại quyền truy cập vào hệ thống dành cho quản lý. | **3** |
| **US13** | Là **Người quản trị viên**, tôi muốn xem thông tin chi tiết các hóa đơn | **4** |

## **Đặc tả yêu cầu phần mềm**

### **Biểu đồ quan hệ thực thể**



### **Sơ đồ use case tổng quát**



### **Chi tiết các use case**

#### *3.1. Use case Quản trị viên sửa thông tin món ăn*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 08 | | |
| UC Name | Sửa thông tin món ăn | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng sửa thông tin món ăn giúp quản trị viên sửa thông tin món ăn khi thông tin trước đó nhập sai hoặc có thêm cập nhật mới về món ăn |
| Trigger | Chức năng sửa thông tin món ăn được thực hiện khi quản trị viên click vào “Sửa món” |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên thực hiện sửa một món ăn thành công |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Sửa món”  3. Hệ thống hiển thị giao diện sửa món ăn  4. Quản trị viên sửa lại thông tin món ăn(Tên món, ngày sản xuất ….), và nhấn “Lưu”  5. Hệ thống hiển thị thông báo sửa thành công món ăn, lưu thông tin món ăn vừa sửa |
| Alternative Flows |  |
| Exceptions |  |

#### *3.2. Use case Quản trị viên xóa món ăn*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 09 | | |
| UC Name | Xóa món ăn | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng xóa món ăn sử dụng để loại bỏ những món ăn mà cửa hàng không còn kinh doanh nữa |
| Trigger | Chức năng sửa thông tin món ăn được thực hiện khi quản trị viên click vào “Xóa món” |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên thực hiện xóa một món ăn thành công |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Xóa món ăn”  3. Hệ thống hiển thị thông báo xóa thành công món ăn |
| Alternative Flows |  |
| Exceptions |  |

#### *3.3. Use case Quản trị viên thêm tài khoản cho nhân viên*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 10 | | |
| UC Name | Thêm mới tài khoản cho nhân viên | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng thêm mới tài khoản nhân viên giúp cho quản trị viên thêm nhân viên vào hệ thống để nhân viên có quyền quản trị hệ thống. |
| Trigger | Chức năng này được thực hiện khi quản trị viên nhấn vào “Thêm nhân viên” |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên thêm thành công tài khoản của nhân viên vào hệ thống và nhân viên đăng nhập thành công vào trang quản trị bằng tài khoản của mình. |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Thêm thành viên”  3. Hệ thống hiển thị giao diện thêm thành viên  4. Quản trị viên thêm mới thông tin nhân viên (Họ tên, Email, mật khẩu ….), và nhấn “Lưu”  5. Hệ thống hiển thị thông báo thêm thành công, lưu thông tin nhân viên. |
| Alternative Flows | Nếu thông tin email trùng khớp với dữ liệu đã lưu thì hiển thị thông báo email đã tồn tại |
| Exceptions |  |

#### *3.4. Use case Quản trị viên sửa thông tin nhân viên*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 11 | | |
| UC Name | Sửa thông tin nhân viên | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng sửa thông tin nhân viên giúp quản trị viên sửa thông tin nhân viên khi thông tin trước đó nhập sai hoặc có thêm cập nhật mới về thông tin của nhân viên. |
| Trigger | Chức năng này được thực hiện khi quản trị viên nhấn vào nút “Sửa nhân viên” |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên sửa thành công thông tin nhân viên |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Sửa thành viên”  3. Hệ thống hiển thị giao diện sửa thành viên  4. Quản trị viên sửa thông tin nhân viên (Họ tên, Email, mật khẩu ….), và nhấn “Lưu”  5. Hệ thống hiển thị thông báo sửa thành công, lưu thông tin nhân viên. |
| Alternative Flows |  |
| Exceptions |  |

#### *3.5. Use case Quản trị viên xóa nhân viên*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 12 | | |
| UC Name | Xóa nhân viên | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng xóa nhân viên dùng để xóa tài khoản của nhân viên, thu hồi quyền truy cập vào trang quản trị hệ thống. |
| Trigger | Chức năng xóa nhân viên thực hiện khi quản trị viên click vào “Xóa nhân viên” |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên xóa thành công tài khoản nhân viên |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Xóa nhân viên”  3. Hệ thống hiển thị thông báo xóa thành công |
| Alternative Flows |  |
| Exceptions |  |

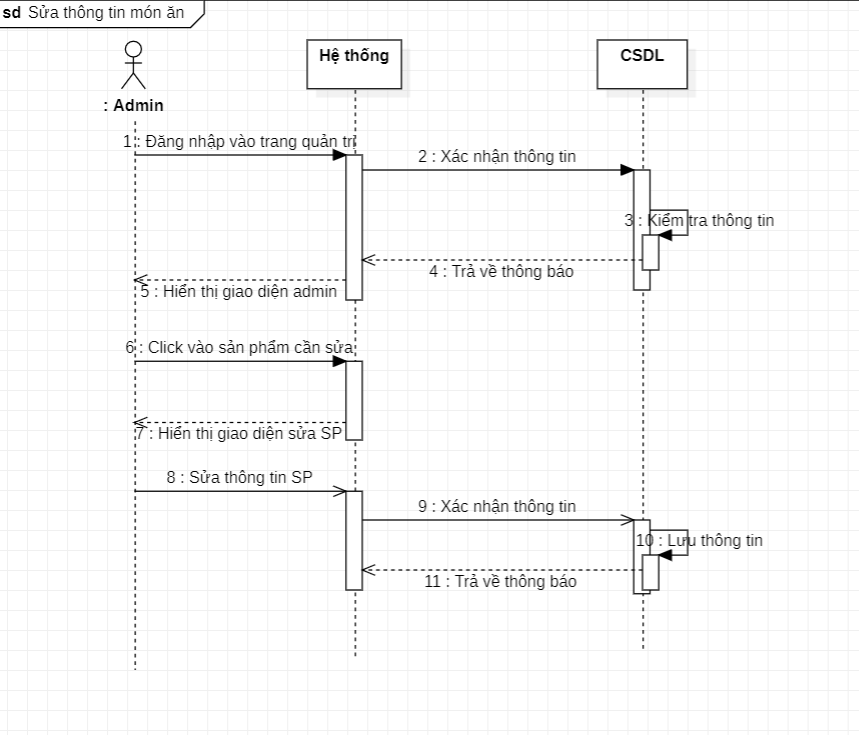
#### *3.6. Use case Quản trị viên xem thông tin hóa đơn*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID | 13 | | |
| UC Name | Xem thông tin hóa đơn | | |
| Created By |  | Last Update By |  |
| Date Created |  | Date Last Update |  |

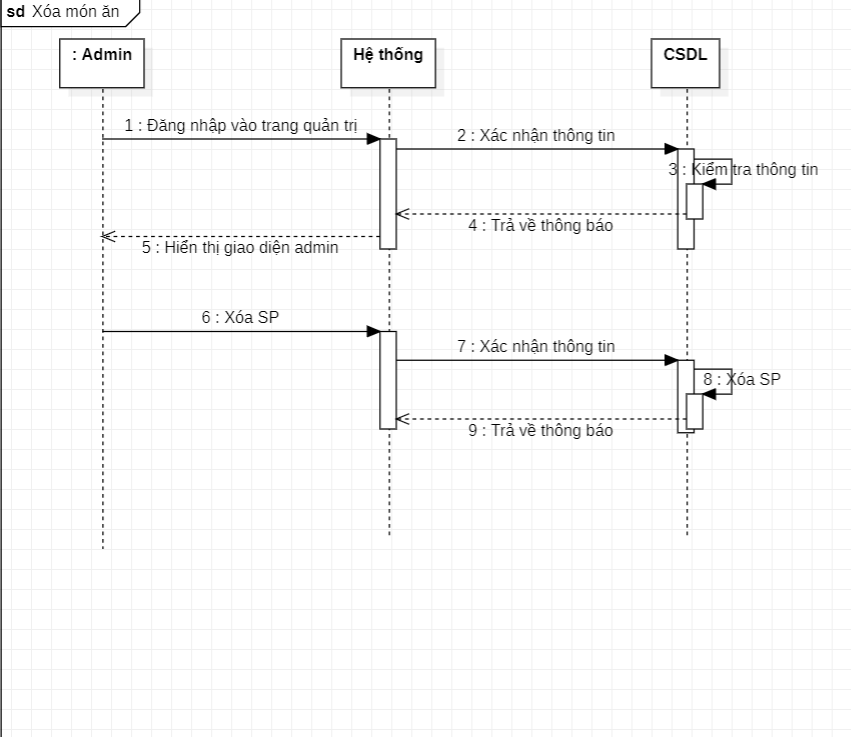
|  |  |
| --- | --- |
| Actor | Quản trị viên |
| Description | Chức năng xem thông tin hóa đơn dùng khi quản trị viên muốn xem chi tiết thông tin các món ăn mà cửa hàng đã bán. |
| Trigger | Chức năng xem thông tin hóa đơn được sử dụng khi quản trị viên chọn “Hóa đơn “ |
| Pre-conditions | Quản trị viên đăng nhập thành công vào hệ thống |
| Post conditions | Quản trị viên xem được thông tin các hóa đơn |
| Flow of Event | 1. Quản trị viên truy cập vào website, và đăng nhập  2. Quản trị viên chọn “Hóa đơn”  3. Hệ thống hiển thị giao diện các hóa đơn có trong hệ thống  4. Quản trị viên chọn hóa đơn mình muốn xem thông tin  4. Hệ thống hiển thị thong tin chi tiết của hóa đơn đó |
| Alternative Flows |  |
| Exceptions |  |

### **Biểu đồ trình tự**

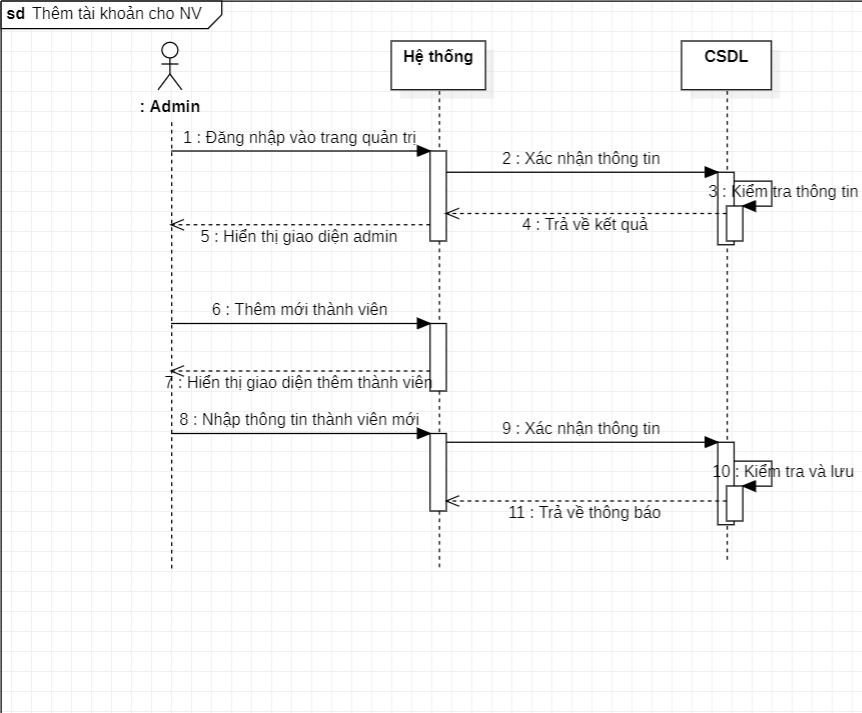
#### *4.1. Use case Sửa thông tin món ăn*



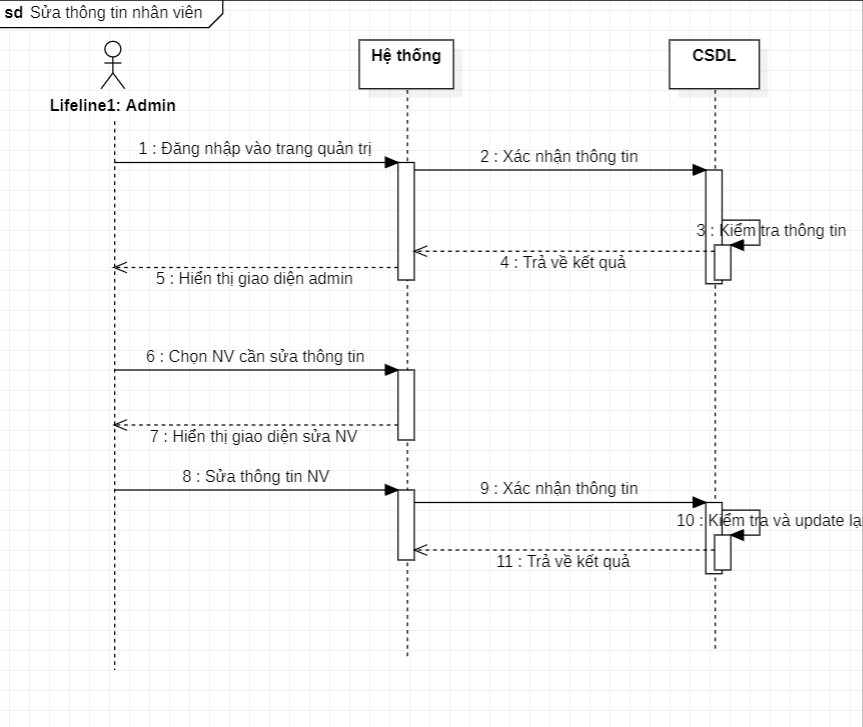
#### *4.2. Use case Xóa món ăn*



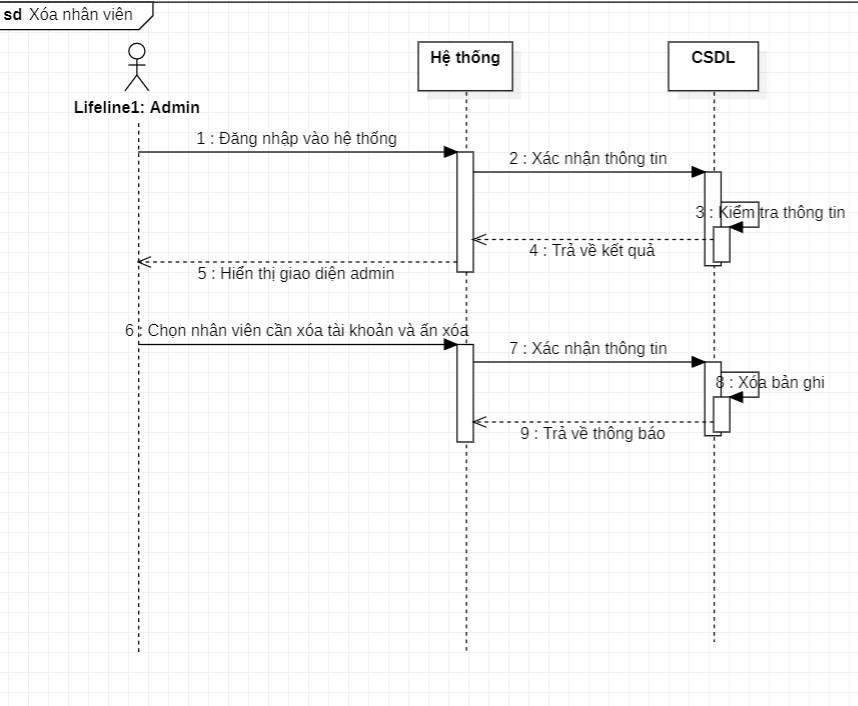
#### *4.3. Use case Thêm tài khoản cho nhân viên*



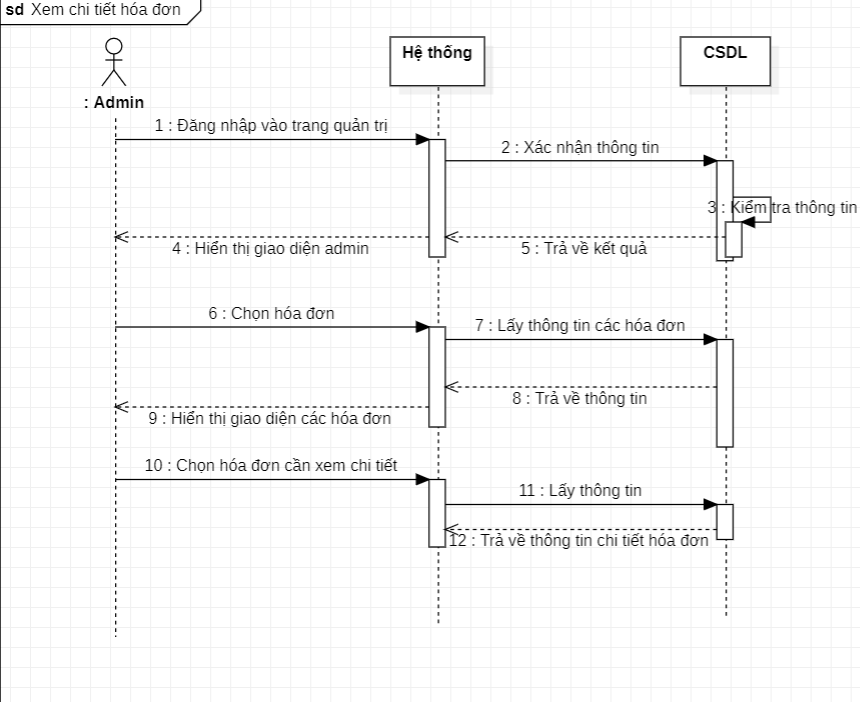
#### *4.4. Use case Sửa thông tin nhân viên*



#### *4.5. Use case Xóa nhân viên*



#### *4.6. Use case Xem thông tin chi tiết hóa đơn*



## **Thiết kế dữ liệu**

### **Thiết kế kiến trúc**

|  |
| --- |
| **Giao diện Web** |

|  |
| --- |
| Hệ thống Đăng ký, Quản lý giao diện Quản lý giao diện  Đăng nhập Người dùng Admin |

|  |
| --- |
| Các tiện ích Đặt hàng Các chức năng  (Tìm kiếm, Xem thông tin) và Thanh toán quản trị |

|  |
| --- |
| DB Admin DB User DB Food DB Order |

Mô hình phân tầng

Tầng đầu tiên là giao diện của Website

Tầng thứ 2 Quản lý các giao diện người dùng

Tầng thứ 3 là các chức năng và tiện ích của hệ thống

Tầng thứ 4 là các Database tương ứng

### **Thiết kế hướng đối tượng**

****

### **Thiết kế dữ liệu**

#### *3.1. Các thực thể và thuộc tính*

* **Admin** (Id\_admin, full\_name, user\_name, pass)
* **User** (id\_user, fisrt\_name, last\_name, user\_name, registration\_date, email, phone, address, pass)
* **Food** (id\_food, title, description, price, image\_name, featured, active)
* **Order** (id\_order, qty, total, order\_date, status, customer\_contact, customer\_address)
* **Category** (Id\_category, title, image\_name, featured, active)

#### *3.2. Mối liên kết*

* **Admin** và **Food** là liên kết 1-N
* **Admin** và **category** là liên kết 1-N
* **Admin** và **order** là liên kết 1-N
* **Order** và **Food** là liên kết N-N
* **Category** và **Food** là liên kết 1-N
* **User** và **order** là liên kết 1-N

#### *3.3. Chuyển sang quan hệ*

* Liên kết 1-N -> Khóa chính của quan hệ bên một trở thành khóa ngoại của quan hệ bên nhiều.
* Liên kết N – N -> Tạo bảng quan hệ mới, chuyển khóa chính của hai quan hệ phía N và N thành khóa ngoại của quan hệ mới. Khóa chính của quan hệ mới là sự kết hợp của hai khóa ngoại.

Kết quả

* **Admin** (Id\_admin, full\_name, user\_name, pass)
* **User** (id\_user, fisrt\_name, last\_name, user\_name, registration\_date, email, phone, address, pass)
* **Food** (id\_food, title, description, price, image\_name, featured, active, *Id\_admin,Id\_category*)
* **Order** (id\_order, qty, total, order\_date, status, customer\_contact, customer\_address, *id\_user, Id\_admin*)
* **Category** (Id\_category, title, image\_name, featured, active, *Id\_admin,*)
* **Order detail** ( id\_food, id\_order, qty, price)

#### *3.4. Chuẩn hóa dữ liệu*

##### *3.4.1. Xét các phụ thuộc hàm*

* Id\_admin xác định một người dùng duy nhất bao gồm full\_name, User\_name, pass
* Id\_food xác định một món ăn duy nhất bao gồm title, description, price, imge\_name, featured, active, id\_admin, id\_category
* Id\_order xác định một hóa đơn duy nhất bao gồm qty, total, order\_date, status, customer\_name, customer\_contact, customer\_address, id\_admin, id\_user
* Id\_category xác định một danh mục duy nhất bao gồm title, image\_name, featured, active, id\_admin
* Id\_user xác định một khách hàng duy nhất bao gồm fisrt\_name, last\_name, user\_name, registration\_date, email, phone, address, pass
* Id\_food, id\_order xác định chi tiết các sản phẩm của một hóa đơn duy nhất, bao gồm: qty, price

##### *3.4.2. Xét các dạng chuẩn*

* **Admin** (Id\_admin, full\_name, user\_name, pass)

Đặt id\_admin = A, full\_name = B, user\_name = C, pass = D

Phụ thuộc hàm: F = {A -> B, A-> C, A-> D}

=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **User** (id\_user, fisrt\_name, last\_name, user\_name, registration\_date, email, phone, address, pass)

Đặt id\_user = A, fisrt\_name = B, last\_name = C, user\_name = D, registration\_date = E, email = G, phone = H, address = I, pass = K

Phụ thuộc hàm: F = {A -> B, A -> C, A -> D, A -> E, A -> G, A -> H, A -> I, A -> K}

=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **Food** (id\_food, title, description, price, image\_name, featured, active, *Id\_admin,Id\_category*)

Đặt id\_food = A, title = B, description = C, price = D, image\_name = E, featured = G, active = H, id\_admin = I, id\_category = K

Phụ thuộc hàm: F = {A -> B, A -> C, A -> D, A -> E, A -> G, A -> H, A -> I, A -> K}

=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **Order** (id\_order, qty, total, order\_date, status, customer\_contact, customer\_address, *id\_admin, id\_user*)

Đặt id\_order = A, qty = B, total = C, order\_date = D, status = E, customer\_contact = G, customer\_address, = H, id\_admin = I, id\_user = K

Phụ thuộc hàm: F = {A -> B, A -> C, A -> D, A -> E, A -> G, A -> H, A -> I, A -> K}

=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **Category** (Id\_category, title, image\_name, featured, active, *Id\_admin,*)

Đặt id\_category = A, title = B, image\_name = C, featured = D, active = E, *Id\_admin* = G

Phụ thuộc hàm: F = {A -> B, A -> C, A -> D, A -> E, A -> G}

=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **Order detail** ( id\_food, id\_order, qty, price)

Đặt id\_food = A, id\_order = B, qty = C, price = D

Phụ thuộc hàm: F = {AB -> C, AB -> D}

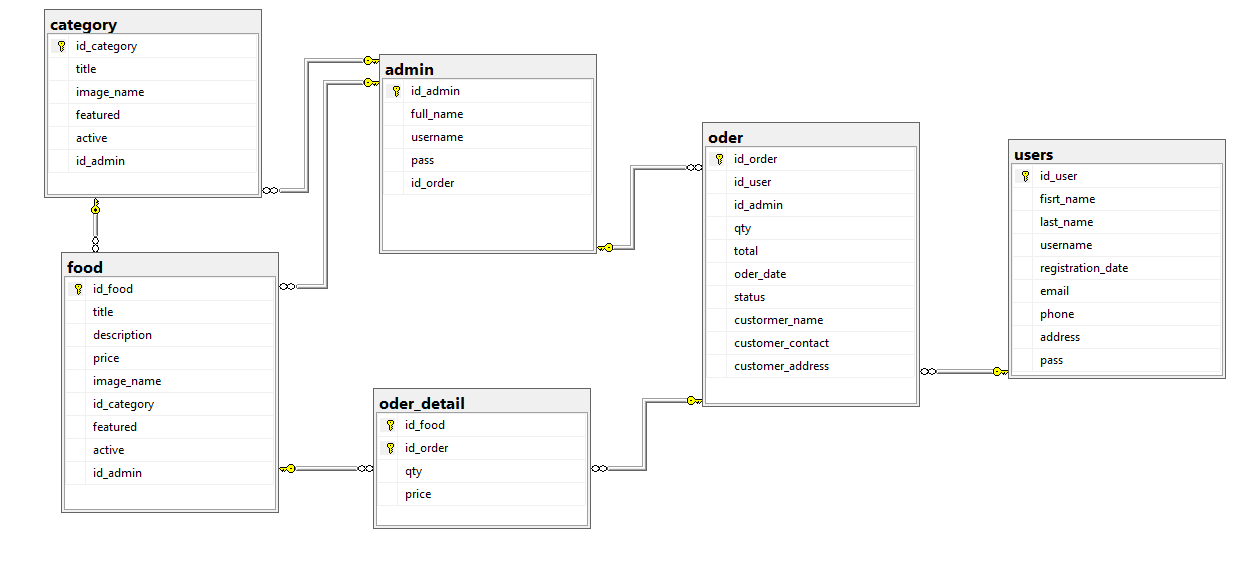
=> Đạt chuẩn 1NF vì không có thuộc tính đa trị

=> Đạt chuẩn 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính AB

=> Đạt chuẩn 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính AB

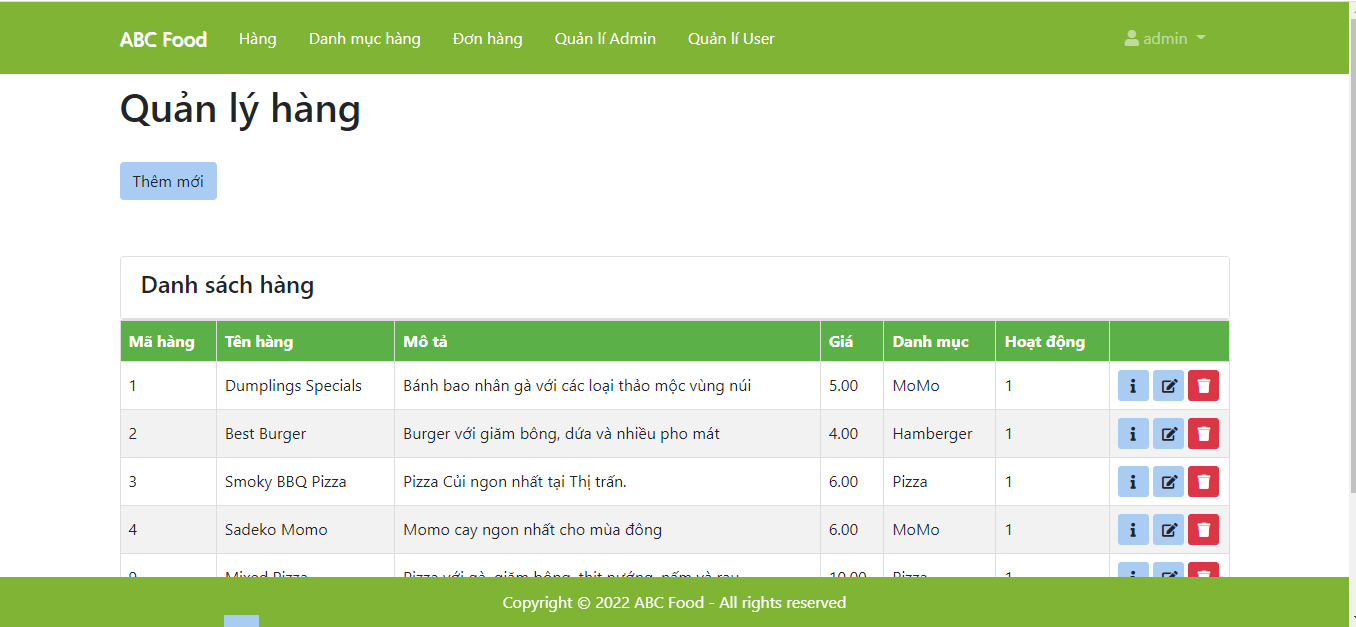
=> Đạt chuẩn BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

#### *3.5. Lược đồ quan hệ*



### **Thiết kế giao diện người dùng**

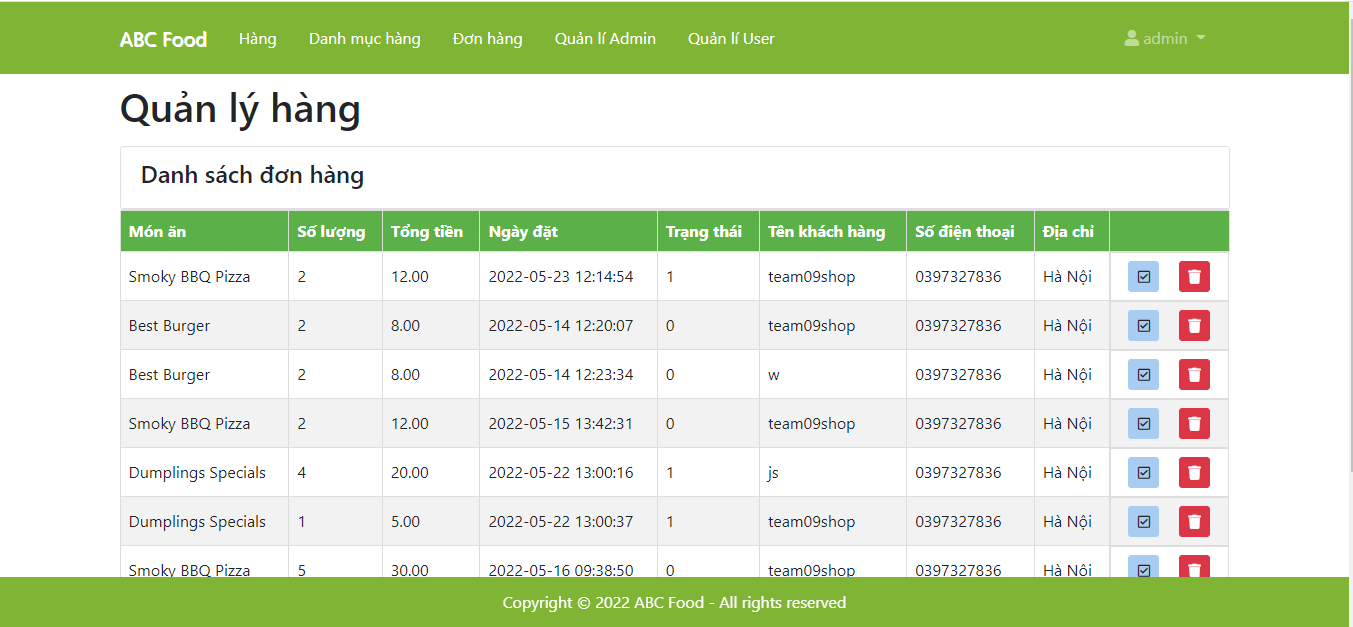
#### *4.1. Quản lý sản phẩm*



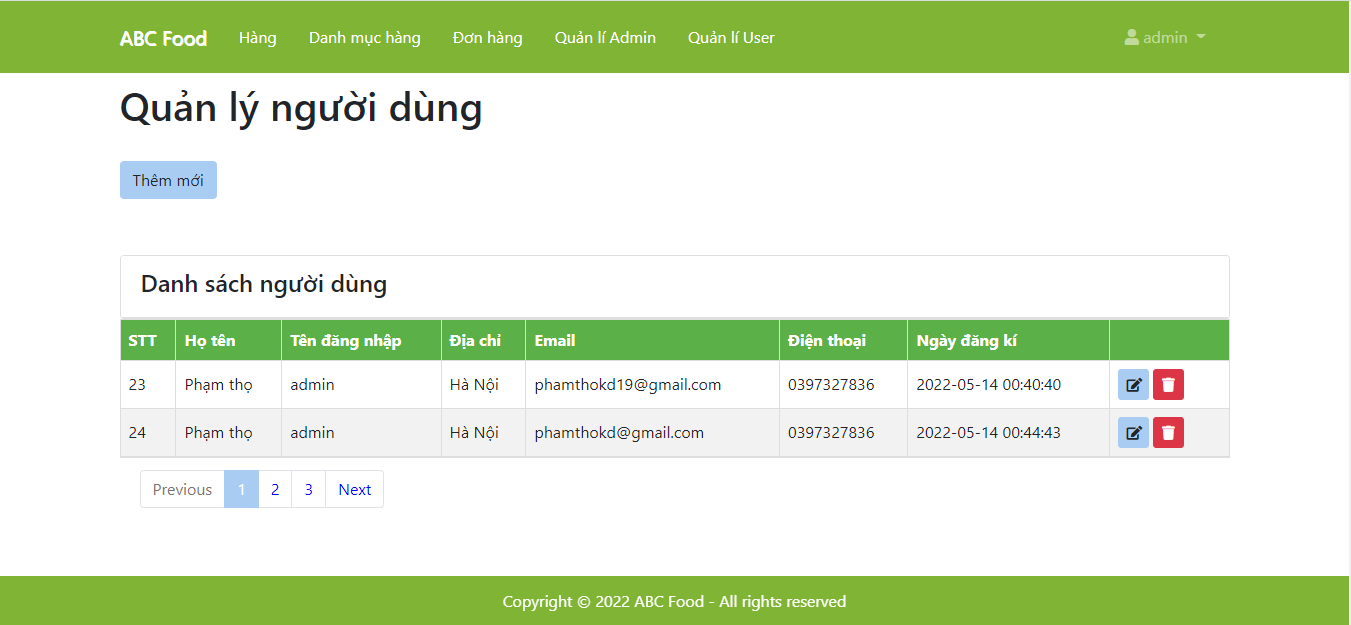
#### *4.2. Quản lý danh mục*



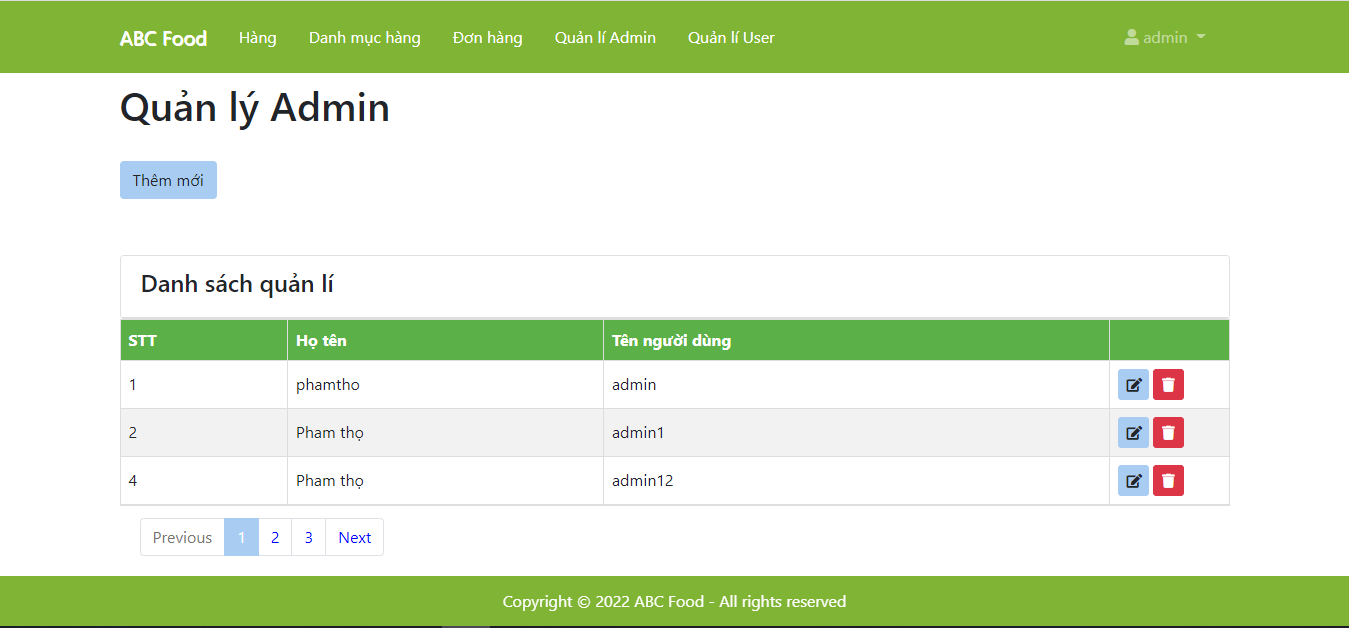
#### *4.3. Quản lý đơn hàng*



#### *4.4. Quản lý users*



#### *4.5. Quản lý admin*



## **Mã nguồn:https://github.com/Phamthokd/BTN\_QLPM.git**

## **Kiểm thử**