

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC**

*Đề tài:*

**XÂY DỰNG HỆ THỐNG THÔNG TIN TRA CỨU QUÁ  
TRÌNH CHĂM SÓC VƯỜN RAU TRÊN NỀN WEB.**

**Người hướng dẫn : ThS. HUỲNH TRỌNG THƯA  
Sinh viên thực hiện : HỒ NGỌC KHÁNH TRÌNH  
Mã số sinh viên : N12DCCN095  
Lớp : D12CQPM01-N  
Khoa : 2012  
Hệ : CHÍNH QUY**

**TP.HCM, Tháng 12 /2016**

## LỜI CẢM ƠN

Trong quá trình thực hiện đề tài *“Xây dựng hệ thống thông tin tra cứu quá trình chăm sóc vườn rau trên nền Web”*, em đã nhận được rất nhiều sự giúp đỡ, tạo điều kiện của tập thể Ban Giám Hiệu, Khoa Công Nghệ Thông Tin, giảng viên, cán bộ các phòng ban chức năng Học Viện Công Nghệ Bưu Chính Viễn Thông tại TP.HCM.

Trước hết em xin bày tỏ sự biết ơn sâu sắc tới ThS. Huỳnh Trọng Thừa – Người đã trực tiếp hướng dẫn em trong suốt quá trình nghiên cứu và thực hiện đề tài.

Em xin chân thành cảm ơn các thầy giáo, cô giáo trong khoa Công Nghệ Thông Tin 2 trường Học Viện Công Nghệ Bưu Chính Viễn Thông thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi giúp đỡ trong thời gian học tập tại trường. Mặc dù trong quá trình nghiên cứu đề tài, bản thân em đã có những cố gắng, song do trình độ và thời gian hạn chế nên đề tài không tránh khỏi những thiếu sót kính mong quý Thầy Cô cho ý kiến để đề tài của em được hoàn thiện hơn.

Sau cùng, em xin kính chúc quý Thầy Cô trong Khoa Công Nghệ Thông Tin và ThS. Huỳnh Trọng Thừa thật dồi dào sức khỏe, niềm tin để tiếp tục sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

TP.Hồ Chí Minh, tháng 12 năm 2016

**Sinh viên thực hiện**

Hồ Ngọc Khánh Trình

## MỤC LỤC

<b>MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. ĐẶT VẤN ĐỀ</b> .....	2
1.1 Mục tiêu, phạm vi nghiên cứu: .....	2
1.2 Đối tượng nghiên cứu: .....	2
1.3 Lợi ích của hệ thống : .....	2
1.4 Các tính năng của ứng dụng cần xây dựng: .....	2
<b>CHƯƠNG 2 CƠ SỞ LÝ THUYẾT</b> .....	3
2.1 Giới thiệu giao thức MQTT:.....	3
2.1.1 Giới thiệu khái quát: .....	3
2.2.2 Các định nghĩa thành phần của giao thức: .....	3
2.2 Giới thiệu các công nghệ áp dụng vào đề tài:.....	4
2.2.1 NodeJs:.....	4
2.2.2 ExpressJs:.....	4
2.2.3 MongoDB: .....	4
2.2.4 AngularJs: .....	5
2.2.5 Nền tảng đám mây IBM Bluemix: .....	7
<b>CHƯƠNG 3 PHÂN TÍCH, THIẾT KẾ VÀ CÀI ĐẶT</b> .....	8
3.1 Phân tích, thiết kế:.....	8
3.1.1. Khái quát hệ thống: .....	8
3.1.2. Cơ sở dữ liệu: .....	9
3.1.3 Sơ đồ Use case:.....	17
3.1.4 Sơ đồ tuần tự: .....	23
3.2 Cài đặt ứng dụng nhúng: .....	26
3.2.1 Cài đặt và cấu hình thiết bị:.....	26
3.3 Cài đặt ứng dụng Web: .....	27
3.3.1 Thiết lập môi trường chạy ứng dụng: .....	27
3.3.2 Cài đặt các chức năng của ứng dụng: .....	29
3.4 Kết quả thu được: .....	31
3.4.1 Màn hình chức năng:.....	31
<b>KẾT LUẬN</b> .....	50
<b>DANH MỤC TÀI LIỆU THAM KHẢO</b> .....	51

## **DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

BẢNG 3.1 Collection User

BẢNG 3.2 Collection Gardens

BẢNG 3.3 Collection Seasons

BẢNG 3.4 Collection Tasks

BẢNG 3.5 Collection Productionitems

BẢNG 3.6 Collection Devicenodes

BẢNG 3.7 Collection Taskcategories

BẢNG 3.8 Collection Histories

BẢNG 3.9 Collection Images

SƠ ĐỒ 3.1 Khái quát hệ thống

SƠ ĐỒ 3.2 Sơ đồ quan hệ cơ sở dữ liệu

SƠ ĐỒ 3.3 Sơ đồ Use Case đăng ký tài khoản

SƠ ĐỒ 3.4 Sơ đồ Use Case đăng nhập

SƠ ĐỒ 3.5 Sơ đồ Use Case quản lý sản xuất

SƠ ĐỒ 3.6 Sơ đồ Use Case quản lý đối tượng sản xuất

SƠ ĐỒ 3.7 Sơ đồ Use Case quản lý duyệt vườn

SƠ ĐỒ 3.8 Sơ đồ Use Case quản lý thiết bị sản xuất

SƠ ĐỒ 3.9 Sơ đồ Use Case danh sách loại tác vụ

SƠ ĐỒ 3.10 Sơ đồ Use Case truy xuất thông tin sản phẩm của hệ thống.

SƠ ĐỒ 3.11 Sơ đồ Use Case quản lý thông tin cá nhân

SƠ ĐỒ 3.12 Sơ đồ Use Case quản lý người dùng của quản trị viên

SƠ ĐỒ 3.13 Sơ đồ tuần tự đăng ký vườn rau

SƠ ĐỒ 3.14 Sơ đồ tuần tự liệt kê danh sách vườn rau

SƠ ĐỒ 3.15 Sơ đồ tuần tự chỉnh sửa vườn rau

SƠ ĐỒ 3.16 Sơ đồ tuần tự xóa vườn rau

HÌNH 3.1 Cấu hình IBM IoT Watson Platform

HÌNH 3.2 Kết quả gửi dữ liệu từ ứng dụng nhúng lên Bluemix

HÌNH 3.3 Cấu hình kết nối cơ sở dữ liệu

HÌNH 3.4 Cấu trúc cây thư mục phía client

HÌNH 3.5 Cấu trúc cây thư mục phía server

HÌNH 3.6 Màn hiển thị danh sách sản phẩm

HÌNH 3.7 Màn hình thông tin sản phẩm

HÌNH 3.8 Màn hình đăng ký tài khoản

HÌNH 3.9 Màn hình đăng nhập

HÌNH 3.10 Màn hình đăng ký vườn rau

HÌNH 3.11 Màn hình chỉnh sửa vườn.

HÌNH 3.12 Màn hình hiển thị danh sách vườn rau

HÌNH 3.13 Màn hình bản đồ vị trí các vườn rau

- HÌNH 3.14 Màn hình thông tin vườn rau
- HÌNH 3.15 Màn hình thiết bị sử dụng của vườn rau
- HÌNH 3.16 Màn hình đối tượng sản xuất của vườn rau
- HÌNH 3.17 Màn hình hình ảnh hoạt động của vườn
- HÌNH 3.18 Màn hình xét duyệt đăng ký vườn rau
- HÌNH 3.19 Màn hình tạo mùa vụ
- HÌNH 3.20 Màn hình hiển thị danh sách mùa vụ
- HÌNH 3.21 Màn hình chi tiết mùa vụ
- HÌNH 3.22 Màn hình hiển thị danh sách đối tượng sản xuất.
- HÌNH 3.23 Màn hình thêm đối tượng sản xuất
- HÌNH 3.24 Màn hình chi tiết đối tượng sản xuất
- HÌNH 3.25 Màn hình chỉnh sửa đối tượng sản xuất
- HÌNH 3.26 Màn hình hiển thị danh sách thiết bị sản xuất
- HÌNH 3.27 Màn hình thêm thiết bị sản xuất
- HÌNH 3.28 Màn hình chi tiết thiết bị sản xuất
- HÌNH 3.29 Màn hình chỉnh sửa thiết bị sản xuất
- HÌNH 3.30 Màn hình hiển thị danh sách tác vụ sản xuất
- HÌNH 3.31 Màn hình thêm loại tác vụ sản xuất
- HÌNH 3.32 Màn hình chi tiết thiết bị sản xuất
- HÌNH 3.33 Màn hình chỉnh sửa loại tác vụ sản xuất
- HÌNH 3.34 Màn hình hiển thị thông tin cá nhân
- HÌNH 3.35 Màn hình chỉnh sửa thông tin cá nhân
- HÌNH 3.36 Màn hình thay đổi ảnh đại diện
- HÌNH 3.37 Màn hình hiển thị danh sách người dùng
- HÌNH 3.38 Màn hình hiển thị thông tin người dùng
- HÌNH 3.39 Màn hình chỉnh sửa thông tin người dùng
- HÌNH 3.40 Màn hình thống kê tỉ lệ sản phẩm

## THUẬT NGỮ, CHỮ VIẾT TẮT

COLLECTION	: Bộ sưu tập, tương ứng với bảng trong cơ sở dữ liệu quan hệ.
CLIENT	: Máy khách.
SERVER	: Máy chủ.
API	: Application Programming Interface – Giao diện lập trình ứng dụng.
MQTT	: Message Queuing Telemetry Transport.
MASTER – SLAVE	: Mô hình chủ - tớ.
NoSQL	: Cơ sở dữ liệu phi quan hệ.
MVC	: Model – View – Controll.
SCHEMA	: Lược đồ.
QoS	: Qualities of service.
PUBLIC	: Công khai.
SUBSCRIBE	: Đăng kí theo dõi.
RETAIN	: Giữ lại.
LWT	: Last will and testament

## MỞ ĐẦU

Hiện nay với sự phát triển của hệ thống thông tin trực tuyến, việc tìm kiếm thông tin và mua bán trực tuyến đang ngày một phổ biến.

Từ tầng lớp tri thức hay đến những người lao động tay chân đều đã và đang tiếp cận dần với công nghệ thông tin. Thêm vào đó là ngày nay nhu cầu an toàn vệ sinh thực phẩm luôn được mọi người quan tâm chú trọng. Ai ai cũng muốn chọn cho mình những thực phẩm tươi ngon, đặc biệt là mặt hàng rau củ rất dễ bị nhiễm thuốc trừ sâu hay sử dụng chất kích thích tăng trưởng. Do đó việc có những nguồn thông tin, những hệ thống tra cứu trực tuyến về những loại rau củ là vô cùng cần thiết.

Hiểu được vấn đề đó cũng như mong muốn xây dựng ứng dụng Web cho người tiêu dùng có thể truy xuất thông tin rau củ quả cũng như hỗ trợ chủ vườn rau quản lý thông tin vườn rau của mình. Vì vậy em thực hiện đề tài: **Xây dựng hệ thống thông tin tra cứu quá trình chăm sóc vườn rau trên nền Web.**

## CHƯƠNG 1. ĐẶT VẤN ĐỀ

### 1.1 Mục tiêu, phạm vi nghiên cứu:

- Mục tiêu xây dựng một hệ thống thông tin có quy mô nhỏ giúp quản lý vườn rau, quá trình canh tác các đối tượng sản xuất của vườn và hỗ trợ tra cứu thông tin rau của quả.

### 1.2 Đối tượng nghiên cứu:

- Vườn rau: các thông tin cơ bản của vườn như tên vườn, chủ vườn, địa chỉ, ngày tạo.
- Nghiệp vụ quản lý và chăm sóc vườn rau.
- Các đặc tính cơ bản của các đối tượng sản xuất.
- Các công nghệ để triển khai hệ thống trên nền Web.

### 1.3 Lợi ích của hệ thống :

- Giúp người tiêu dùng truy xuất được nguồn gốc và thông tin kỹ thuật của sản phẩm rau củ quả.
- Giúp chủ vườn có thể đăng ký thông tin vườn rau của mình.
- Giúp người quản trị có thể quản lý thông tin và hỗ trợ cho chủ vườn về vườn rau, mùa vụ, sản phẩm rau củ quả.

### 1.4 Các tính năng của ứng dụng cần xây dựng:

- Gửi dữ liệu nhiệt độ, độ ẩm từ ứng dụng nhúng lên nền tảng đám mây Bluemix
- Nhận dữ liệu từ nền tảng đám mây thông qua khai thác API của nền tảng Bluemix.
- Đăng ký, đăng nhập.
- Đăng ký thông tin vườn rau.
- Quản lý sản xuất đối với chủ vườn.
- Xét duyệt vườn rau dành cho người quản trị.
- Quản lý thông tin cá nhân.
- Quản lý người dùng.
- Quản lý đối tượng sản xuất.
- Quản lý thiết bị.
- Quản lý tác vụ chăm sóc vườn rau.
- Truy xuất thông tin sản phẩm của vườn rau.
- Thống kê



## CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

### 2.1 Giới thiệu giao thức MQTT:

#### 2.1.1 Giới thiệu khái quát:

MQTT (Message Queuing Telemetry Transport) là một giao thức gửi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.

MQTT cũng là giao thức sử dụng trong Facebook Messenger.

#### 2.2.2 Các định nghĩa thành phần của giao thức:

##### - Publish, subscribe:

Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là mqtt client - gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như “/client1/channel1”, “/client1/channel2”. Quá trình đăng ký này gọi là “subscribe”. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký. Khi một client gửi dữ liệu tới kênh đó, gọi là “publish”.

##### - QoS (Qualities of service):

Ở đây có 3 tùy chọn khi “publish” và “subscribe”:

- + QoS0 Broker/client sẽ gửi dữ liệu đúng 1 lần.
- + QoS1 Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
- + QoS2 Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.

Một gói tin có thể được gửi ở bất kỳ QoS nào, và các client cũng có thể subscribe với bất kỳ yêu cầu QoS nào. Có nghĩa là client sẽ lựa chọn QoS tối đa mà nó có thể nhận tin. Ví dụ, nếu 1 gói dữ liệu được publish với QoS2, và client subscribe với QoS0, thì gói dữ liệu được nhận về client này sẽ được broker gửi với QoS0, và 1 client khác đăng ký cùng kênh này với QoS 2, thì nó sẽ được Broker gửi dữ liệu với QoS2.

Một ví dụ khác, nếu 1 client subscribe với QoS2 và gói dữ liệu gửi vào kênh đó publish với QoS0 thì client đó sẽ được Broker gửi dữ liệu với QoS0. QoS càng cao thì càng đáng tin cậy, đồng thời độ trễ và băng thông đòi hỏi cũng cao hơn.

##### - Retain:

Nếu **RETAIN** được set bằng 1, khi gói tin được publish từ Client, Broker phải lưu trữ lại gói tin với QoS, và nó sẽ được gửi đến bất kỳ Client nào subscribe cùng kênh trong tương lai. Khi một Client kết nối tới Broker và subscribe, nó sẽ nhận được gói tin cuối cùng có RETAIN = 1 với bất kỳ topic nào mà nó đăng ký trùng. Tuy nhiên, nếu Broker nhận được gói tin mà có QoS = 0 và RETAIN = 1, nó sẽ hủy tất cả

các gói tin có RETAIN = 1 trước đó. Và phải lưu gói tin này lại, nhưng hoàn toàn có thể hủy bất kỳ lúc nào.

Khi publish một gói dữ liệu đến Client, Broker phải set RETAIN = 1 nếu gói được gửi như là kết quả của việc subscribe mới của Client (giống như tin nhắn ACK báo subscribe thành công). RETAIN phải bằng 0 nếu không quan tâm tới kết quả của việc subscribe.

## LWT

Gói tin LWT (last will and testament) không thực sự biết được Client có trực tuyến hay không, cái này do gói tin KeepAlive đảm nhận. Tuy nhiên gói tin LWT như là thông tin điều gì sẽ xảy ra đến sau khi thiết bị ngoại tuyến.

## 2.2 Giới thiệu các công nghệ áp dụng vào đề tài:

### 2.2.1 NodeJs:

NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng C++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.

Tại thời điểm này, nó được rất nhiều nhà phát triển ứng dụng ưa chuộng và còn là một nền tảng rất mới mẻ.

NodeJS đang là một nền tảng mới và thú vị cho việc phát triển các ứng dụng web, ứng dụng Server. NodeJS cũng cho thấy rằng nó hứa hẹn là một nền tảng hấp dẫn có thể thay thế được các nền tảng truyền thống như Apache, PHP, Python . . . NodeJS là một nền tảng cho việc viết ứng dụng Javascript phía server. Với ngôn ngữ Javascript và nền tảng nhập xuất bất đồng bộ, nó là một nền tảng mạnh mẽ để phát triển các ứng dụng thời gian thực.

### 2.2.2 ExpressJs:

ExpressJs là một web application framework cho NodeJS, cung cấp các tính năng mạnh mẽ cho việc xây dựng một ứng dụng web đúng nghĩa hoặc lai.

ExpressJS là framework phổ biến và được sử dụng rộng rãi nhất của NodeJS, được xây dựng trên cấu trúc ngữ pháp của Sinatra. Ý tưởng đằng sau ExpressJS là đưa đến một framework nhẹ, dễ dàng tiếp cận để phát triển các ứng dụng web từ nhỏ đến lớn hay ứng dụng lai.

Express cũng có thể sử dụng để xây dựng một API mạnh mẽ và thân thiện với người dùng, vì nó cung cấp rất nhiều tiện ích HTTP và middleware cho việc kết nối.

### 2.2.3 MongoDB:

MongoDB là một cơ sở dữ liệu có dạng NoSQL.

NoSQL là một thể hệ cơ sở dữ liệu có các đặc điểm chính là không ràng buộc phân tán, mã nguồn mở, khả năng co giãn theo chiều ngang có thể lưu trữ, xử lý từ một lượng rất nhỏ cho tới dữ liệu cực lớn, lên đến hàng petabytes dữ liệu trong hệ thống cần có độ chịu tải, chịu lỗi cao với những yêu cầu về tài nguyên phần cứng thấp.

Mục tiêu chính của Mongo là giữ lại các thuộc tính thân thiện của SQL. Do đó các câu truy vấn khá giống với SQL nên MongoDB khá thích hợp cho các lập trình viên đã quen với ngôn ngữ truy vấn SQL. MongoDB có một khối lượng tính năng lớn và hiệu năng cao. Với các loại dữ liệu phong phú, nhiều truy vấn và việc giảm thời gian phát triển trong việc mô hình hóa các đối tượng.

MongoDB được sử dụng tốt nhất với nhu cầu cần truy vấn động, cần tốc độ nhanh cho một cơ sở dữ liệu lớn vì MongoDB ngoài tốc độ đọc nhanh ra thì tốc độ ghi của nó rất nhanh. MongoDB hỗ trợ việc tìm theo trường, khoảng kết quả tìm và tìm theo cú pháp. Các truy vấn có thể trả về các trường được qui định trong văn bản và cũng có thể bao gồm các hàm Javascript mà người dùng chưa định nghĩa. Cũng giống như các cơ sở dữ liệu quan hệ, bất cứ một trường nào trong MongoDB đều được đánh chỉ mục.

MongoDB còn có hỗ trợ theo mô hình chủ – tớ (master – slave), mảnh, vùng dữ liệu (Sharding).

MongoDB sử dụng một quá trình xử lý để xử lý các yêu cầu về dữ liệu, quản lý định dạng dữ liệu, thực hiện các hoạt động quản lý bên dưới là mongod, đây là trình xử lý chính. Trong việc mở rộng theo chiều ngang sử dụng mô hình mảnh lưu trữ, MongoDB cung cấp dịch vụ xử lý các truy vấn từ tầng ứng dụng, xác định vị trí dữ liệu trong cụm các node phân mảnh được gọi là mongos.

#### **2.2.4 AngularJs:**

AngularJS là một framework ứng dụng web mã nguồn mở. Nó được phát triển lần đầu năm 2009 bởi Misko Hevery và Adam Abrons. Hiện tại nó được duy trì bởi Google.

AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó chạy trên tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.

Các tính năng chung của AngularJS:

- AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
- AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.
- Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.
- AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.

Ưu điểm của AngularJS:

- AngularJS cung cấp khả năng tạo ra các Single Page Application một cách rất rõ ràng và dễ dàng để duy trì.
- AngularJS cung cấp khả năng Data binding tới HTML do đó giúp người dùng cảm giác linh hoạt, thân thiện.
- AngularJS code dễ dàng khi unit test.
- AngularJS sử dụng dependency injection.
- AngularJS cung cấp khả năng tái sử dụng các component (thành phần).
- Với AngularJS, lập trình viên sẽ viết ít code hơn, với nhiều chức năng hơn.
- Với AngularJS, view là thành phần trong trang HTML thuần, trong khi controller được viết bởi JavaScript với quá trình xử lý nghiệp vụ.

Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

- Không an toàn : Là một JavaScript framework, ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
- Nếu người sử dụng ứng dụng của bạn vô hiệu hóa JavaScript thì họ chỉ nhìn được trang cơ bản, không thấy gì thêm.

**2.2.5 Nền tảng đám mây IBM Bluemix:**

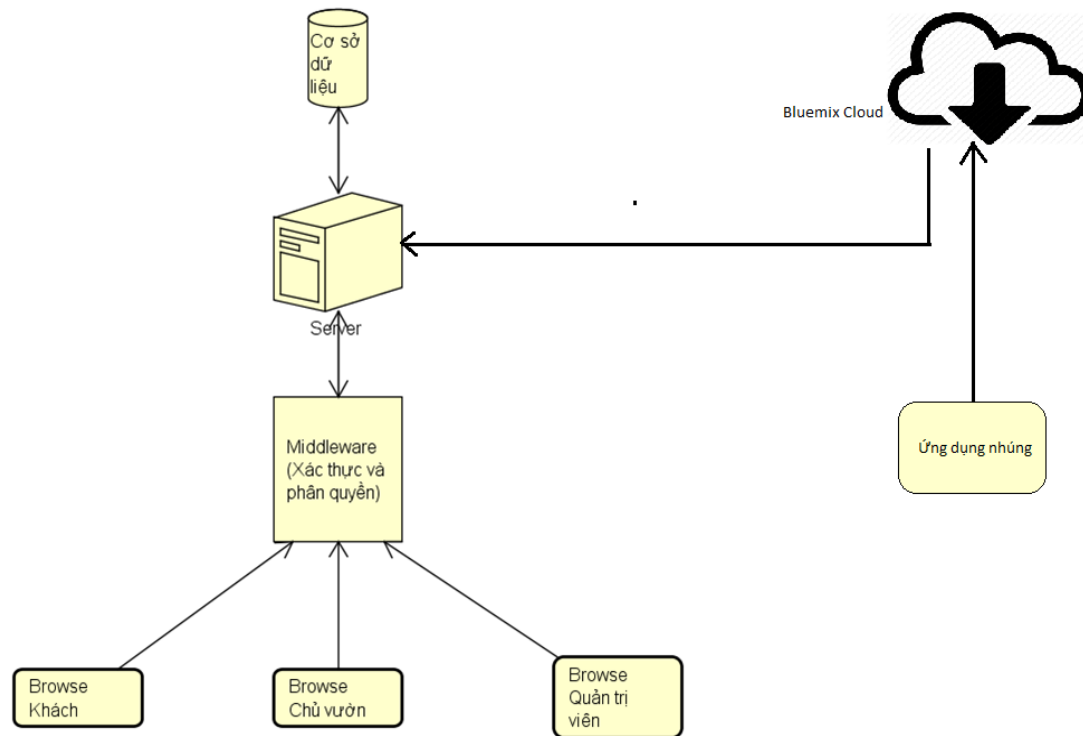
- Bluemix là nền tảng trên đám mây của IBM, nó cho phép nhà phát triển phần mềm: thiết kế, phát triển, triển khai và theo dõi các ứng dụng Web và ứng dụng di động của mình trên đám mây một cách nhanh chóng. Người phát triển có thể sử dụng các ngôn ngữ lập trình Java, JavaScript (Node.js), PHP, Python, Go, Ruby, và rất nhiều các API của IBM, của các công ty khác và của cộng đồng nguồn mở có sẵn cho các ứng dụng của mình.
- Trong đề tài này, em đã sử dụng dịch vụ IBM IoT Watson Platform từ nền tảng Bluemix để phát triển ứng dụng nhúng để gửi dữ liệu từ thiết bị nhúng thông tin về nhiệt độ và độ ẩm lên nền tảng đám mây. Sau đó, khai thác API của dịch vụ cung cấp để nhận dữ liệu về máy chủ ứng dụng và cung cấp cho máy khách.

## CHƯƠNG 3 PHÂN TÍCH, THIẾT KẾ VÀ CÀI ĐẶT

### 3.1 Phân tích, thiết kế:

#### 3.1.1. Khái quát hệ thống:

- Sơ đồ khái quát hệ thống:



Sơ đồ 3.1 Khái quát hệ thống.

- Hệ thống cho phép người tiêu dùng ( guest ) truy xuất thông tin quá trình canh tác của các đối tượng sản xuất.
- Cho phép chủ vườn ( user) đăng ký và quản lý thông tin vườn rau của mình và theo dõi dữ liệu quá trình canh tác của vườn rau qua từng mùa vụ.
- Cho phép quản trị viên ( admin ) quản lý thông tin người dùng, duyệt đăng ký vườn và tiến hành tạo mùa vụ cho từng vườn, cập nhật nguồn tài nguyên của hệ thống (danh sách rau củ quả).
- Ứng dụng nhúng thu thập thông tin nhiệt độ và độ ẩm gửi lên nền tảng đám mây Bluemix, server của ứng dụng sẽ nhận dữ liệu theo thời gian thực lưu trữ vào cơ sở dữ liệu.

**3.1.2. Cơ sở dữ liệu:**

- **Collection (tương ứng với bảng trong hệ quản trị cơ sở dữ liệu quan hệ) User:**

+ Dùng để lưu trữ thông tin người dùng của hệ thống, bao gồm thông tin cá nhân, tên đăng nhập, mật khẩu (sau khi băm) kèm theo quyền truy cập (roles). Người dùng ở mức User là chủ vườn rau, người dùng ở mức Admin là quản trị của hệ thống.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId của người dùng
displayName	String		Tên hiển thị
username	String	Required, Unique	Tên đăng nhập
email	String	Required, Unique	Thư điện tử của người dùng
roles	Array	Required	Quyền của người dùng
createDate	Date	Default: Date.now()	Ngày tạo tài khoản
profileImageURL	String		Địa chỉ URL ảnh đại diện
password	String	Required	Mật khẩu
lastName	String		Họ của người dùng
firstName	String		Tên của người dùng

*Bảng 3.1 Collection User*

**- Collection Gardens:**

+ Dùng để lưu trữ thông tin vườn rau, liên kết với Collection User thông qua trường User, liên kết với Collection ProductionItems thông qua trường productionItem.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId của vườn rau
user	ObjectId	Reference	ObjectId chủ vườn
location	Array	Required	Vị trí của vườn rau
deviceNode	ObjectId	Reference	ObjectId thiết bị sản xuất
description	String		Mô tả vườn
created	Date	Default: Date.now()	Ngày tạo vườn
productionItem	Array	Reference	ObjectId đối tượng sản xuất
name	String	Required, Unique	Tên vườn
address	String	Required	Địa chỉ vườn
area	Number	Required	Diện tích vườn
imageUrl	String		URL ảnh đại diện của vườn
approved	Boolean	Default : false	Vườn đã được duyệt hay chưa
isDeleted	Boolean	Default : false	Vườn đã được xóa hay chưa
deleteDate	Date		Ngày xóa vườn

*Bảng 3.2 Collection Gardens*



**- Collection Seasons:**

+ Dùng để lưu trữ thông tin mùa vụ sản xuất của vườn rau, bao gồm thông tin cơ bản của vườn rau và thông tin lịch sử quá trình canh tác của vườn rau; Liên kết với Collection Gardens thông qua trường Garden, liên kết với Collection ProductionItems thông qua trường productionItem.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId của mùa vụ
user	ObjectId	Reference	ObjectId chủ vườn
garden	ObjectId	Reference	ObjectId vườn rau
productionItem	ObjectId	Reference	ObjectId thiết bị sản xuất
startDate	Date		Ngày bắt đầu mùa vụ
endDate	Date		Ngày thu hoạch dự kiến
quantity	Number		Số lượng sản phẩm thu hoạch
name	String	Required, Unique	Tên mùa vụ
seedQuantity	Number		Số lượng giống
created	Date	Default : Date.now()	Ngày tạo mùa vụ
status	Number	Default : 0	Tình trạng mùa vụ
isDeleted	Boolean	Default : false	Mùa đã được xóa hay chưa
deleteDate	Date		Ngày xóa mùa vụ

Bảng 3.3 Collection Seasons

**- Collection Tasks:**

+ Dùng để lưu trữ thông tin quá trình chăm sóc vườn rau thông qua những tác vụ.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId của tác vụ
user	ObjectId	Reference	ObjectId chủ vườn
season	ObjectId	Reference	Vị trí của vườn rau
type	ObjectId	Reference	Loại tác vụ
date	Date		Ngày thực hiện tác vụ
created	Date	Default: Date.now()	Ngày tạo tác vụ
status	Number	Default	Tình trạng tác vụ
isDeleted	Boolean	Default : false	Tác vụ đã được xóa hay chưa
deleteDate	Date		Ngày xóa tác vụ

*Bảng 3.4 Collection Tasks*

**- Collection Productionitems:**

+ Dùng để lưu trữ dữ liệu tên, đường dẫn hình ảnh, id đối tượng sản xuất có trên hệ thống để người dùng có thể chọn trong lúc đăng ký thông tin vườn rau.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId đối tượng sản xuất
imgUrl	String		URL của ảnh đối tượng sản xuất
name	String	Required, Unique	Tên đối tượng sản xuất
description	String		Mô tả đối tượng
created	Date	Default: Date.now()	Ngày tạo đối tượng sản xuất
isDeleted	Boolean	Default : false	Đối tượng đã được xóa hay chưa
deleteDate	Date		Ngày xóa đối tượng

*Bảng 3.5 Collection Productionitems*

**- Collection Devicenodes:**

+ Dùng để lưu trữ thông tin về thiết bị được dùng trong vườn rau để người dùng có thể chọn lúc đăng ký thông tin vườn rau.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId thiết bị sản xuất
name	String	Required, Unique	Tên thiết bị sản xuất
description	String		Mô tả thiết bị
type	Number	Default: 0	Kiểu thiết bị: 0 là dùng chung, 1 là dùng riêng
apiKey	String	Required	API Key của thiết bị
apiToken	String	Required	API Token của thiết bị
created	Date	Default: Date.now()	Ngày tạo thiết bị
isDeleted	Boolean	Default : false	Thiết bị đã được xóa hay chưa
deleteDate	Date		Ngày xóa thiết bị

*Bảng 3.6 Collection Devicenodes*

**- Collection Taskcategories:**

+ Dùng để lưu trữ thông tin các loại tác vụ chăm sóc vườn rau có trên hệ thống.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId loại tác vụ
name	String	Required, Unique	Tên loại tác vụ
description	String		Mô tả loại tác vụ
type	Number	Required, Unique	Kiểu tác vụ
imgUrl	String		URL hình ảnh của tác vụ
created	Date	Default: Date.now()	Ngày tạo loại tác vụ
isDeleted	Boolean	Default : false	Loại tác vụ đã được xóa hay chưa
deleteDate	Date		Ngày xóa loại tác vụ

*Bảng 3.7 Collection Taskcategories*

**- Collection Histories:**

+ Dùng để lưu trữ dữ liệu nhiệt độ, độ ẩm của vườn rau.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId loại tác vụ
apiKey	String		API Key thiết bị sử dụng
temperature	Number		Nhiệt độ
humidity	Number		Độ ẩm
created	Date	Default: Date.now()	Ngày tạo thông tin
isDeleted	Boolean	Default : false	Thông tin đã được xóa hay chưa
deleteDate	Date		Ngày xóa thông tin

*Bảng 3.8 Collection Histories*

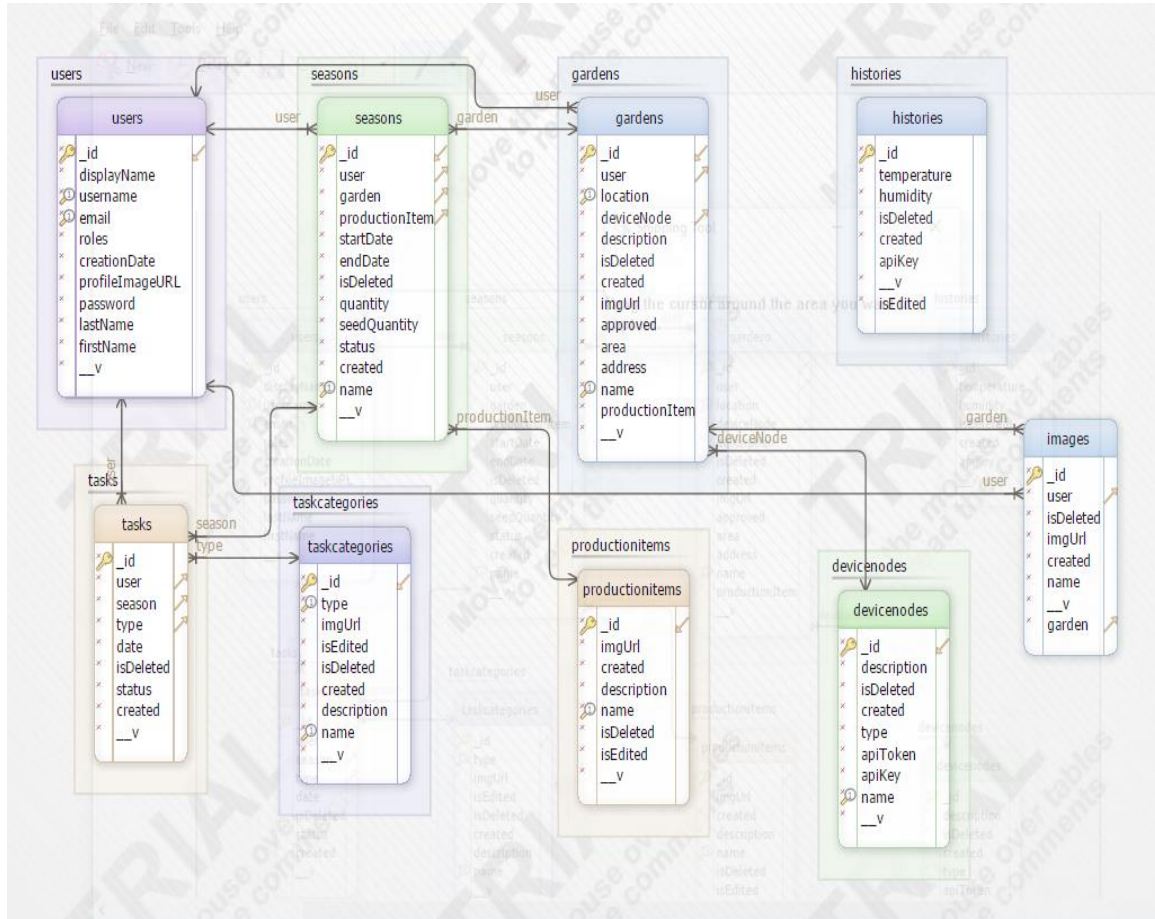
**- Collection Images:**

+ Dùng để lưu trữ dữ liệu hình ảnh hoạt động của vườn rau.

Tên	Kiểu	Ràng buộc	Mô tả
_id	ObjectId	Primary Key, Auto Increment	ObjectId của ảnh
user	ObjectId	Reference	ObjectId người dùng tải ảnh lên.
garden	ObjectId	Reference	ObjectId của vườn rau
name	String		Tên ảnh
imgUrl	String	Required	URL của ảnh
created	Date	Default: Date.now()	Ngày tạo ảnh
isDeleted	Boolean	Default : false	Ảnh đã được xóa hay chưa
deleteDate	Date		Ngày xóa ảnh

*Bảng 3.9 Collection Images*

### - Quan hệ giữa các collection:



Sơ đồ 3.2 Sơ đồ quan hệ cơ sở dữ liệu

- + Mỗi chủ vườn ( User) có thể có nhiều vườn rau (Gardens) , nhưng mỗi vườn rau chỉ có một chủ vườn.
- + Mỗi vườn rau có thể có nhiều mùa vụ sản xuất (Seasons), nhưng mỗi mùa vụ sản xuất chỉ thuộc một vườn rau.

### 3.1.3 Sơ đồ Use case:

#### - Các nhóm chức năng hệ thống:

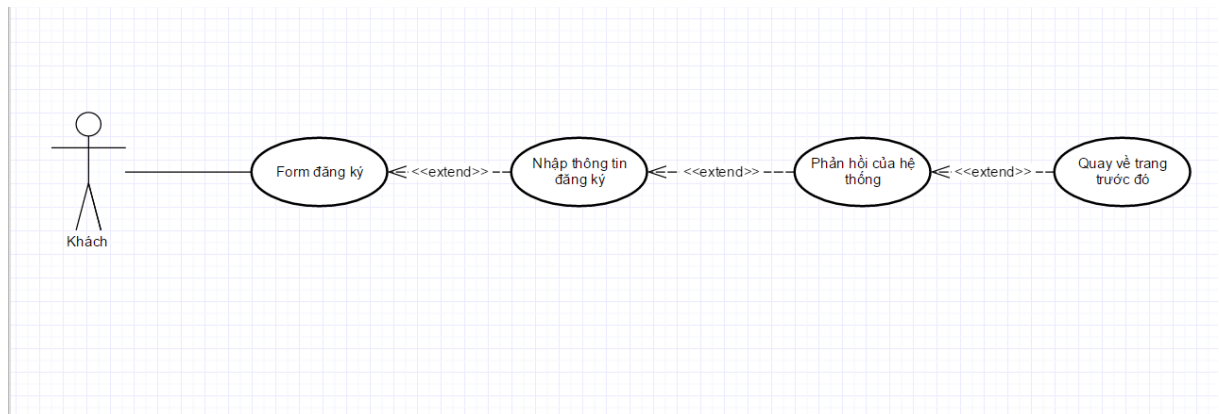
Chức năng của hệ thống có thể chia làm những nhóm chức năng chính như sau:

- + Nhóm chức năng đăng ký, đăng nhập thành viên.
- + Nhóm chức năng dành cho người dùng bao gồm quản lý thông tin cá nhân, quản lý vườn rau, quản lý mùa vụ.
- + Nhóm chức năng quản lý dành cho người quản trị bao gồm quản lý xét duyệt đăng ký vườn rau, quản lý các tài khoản, quản lý các đối tượng sản xuất, thiết bị sản xuất, danh sách loại tác vụ.

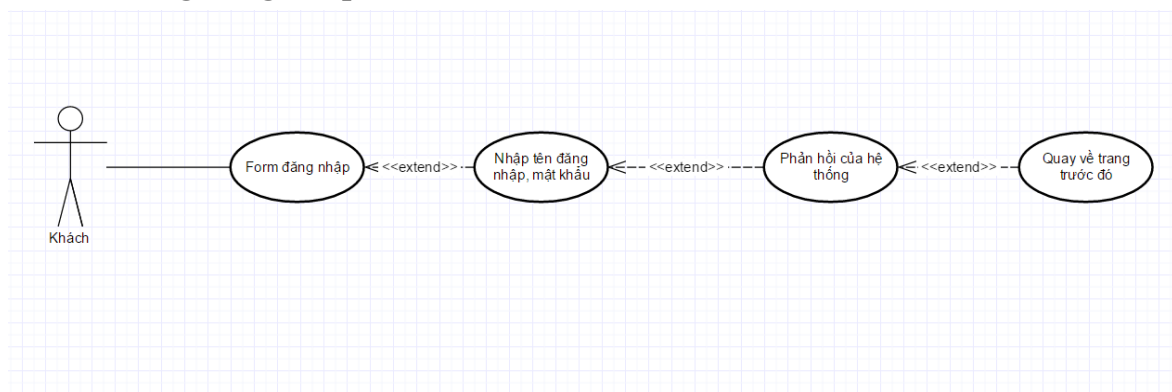
**- Xác định các tác nhân:**

Dựa vào các mô tả trên, có thể xác định những tác nhân chính như sau:

- + Khách (người tiêu dùng): người sử dụng hệ thống.
- + Chủ vườn: người quản lý vườn rau, kiểm soát thông tin và các chức năng của từng vườn rau và từng mùa vụ tương ứng.
- + Quản trị hệ thống: người điều hành, quản lý và theo dõi mọi hoạt động của hệ thống.

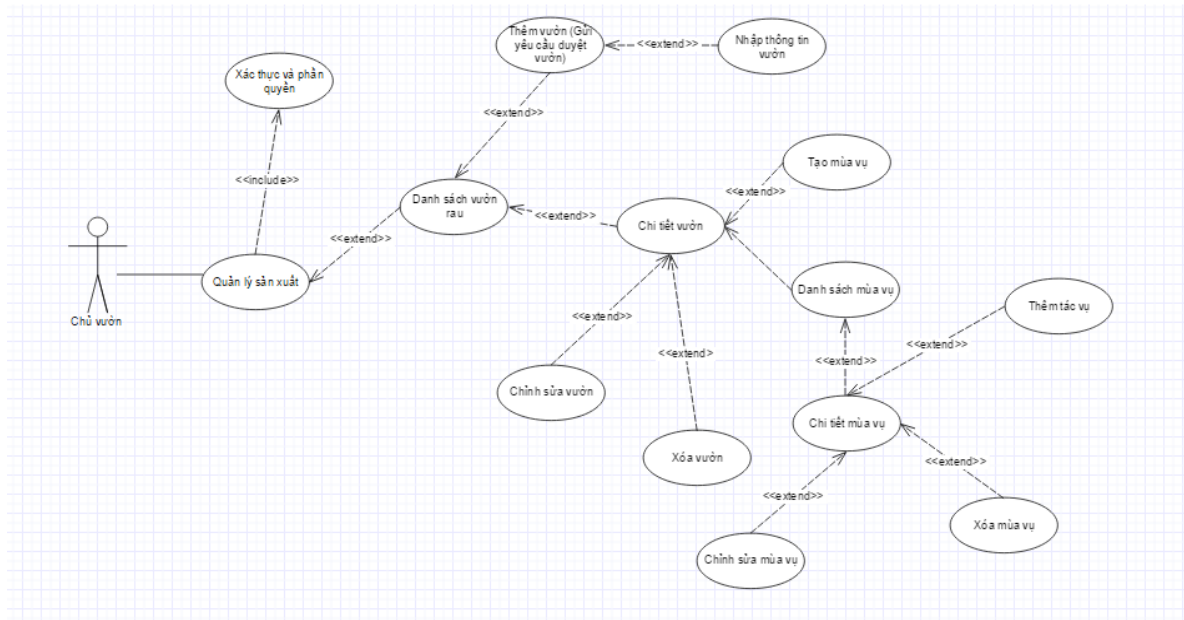
**- Các sơ đồ Use Case chính:****+ Chức năng đăng ký tài khoản:**

Sơ đồ 3.3 Sơ đồ Use Case đăng ký tài khoản

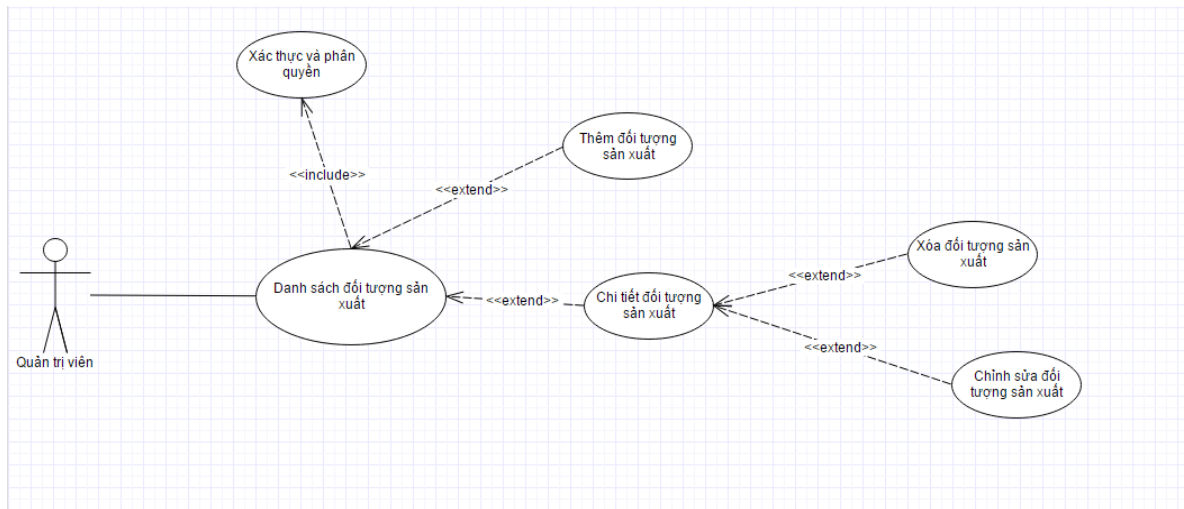
**+ Chức năng đăng nhập:**

Sơ đồ 3.4 Sơ đồ Use Case đăng nhập



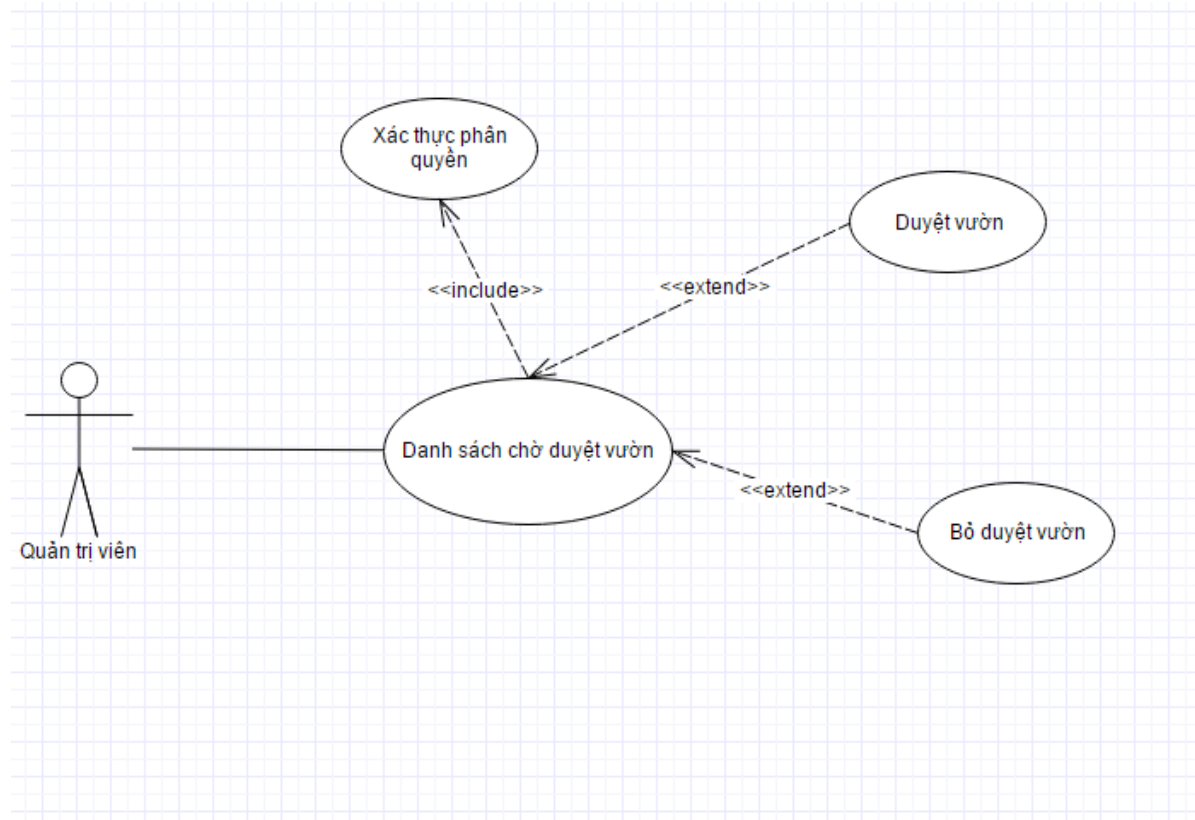
**+ Chức năng quản lý sản xuất:**

Sơ đồ 3.5 Sơ đồ Use Case quản lý sản xuất

**+ Chức năng quản lý đối tượng sản xuất:**

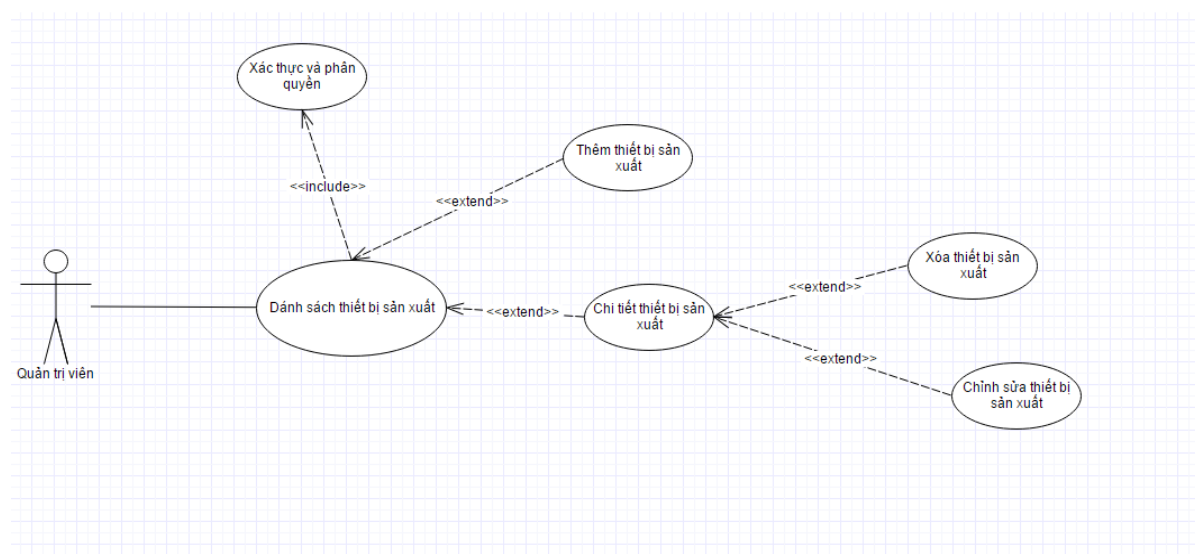
Sơ đồ 3.6 Sơ đồ Use Case quản lý đối tượng sản xuất.

**+ Chức năng quản lý duyệt vườn:**

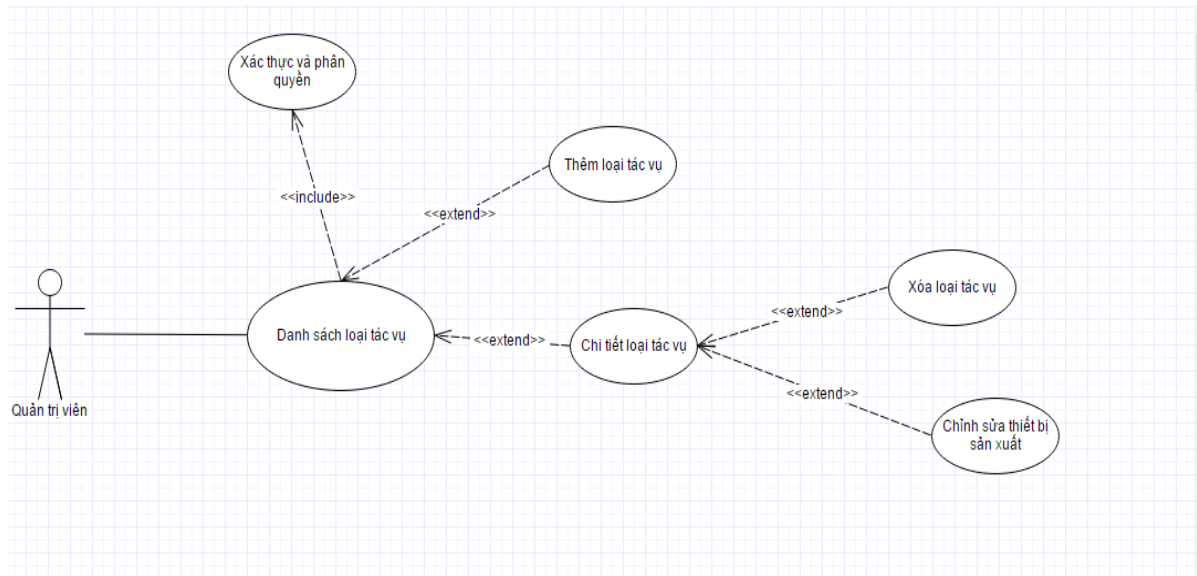
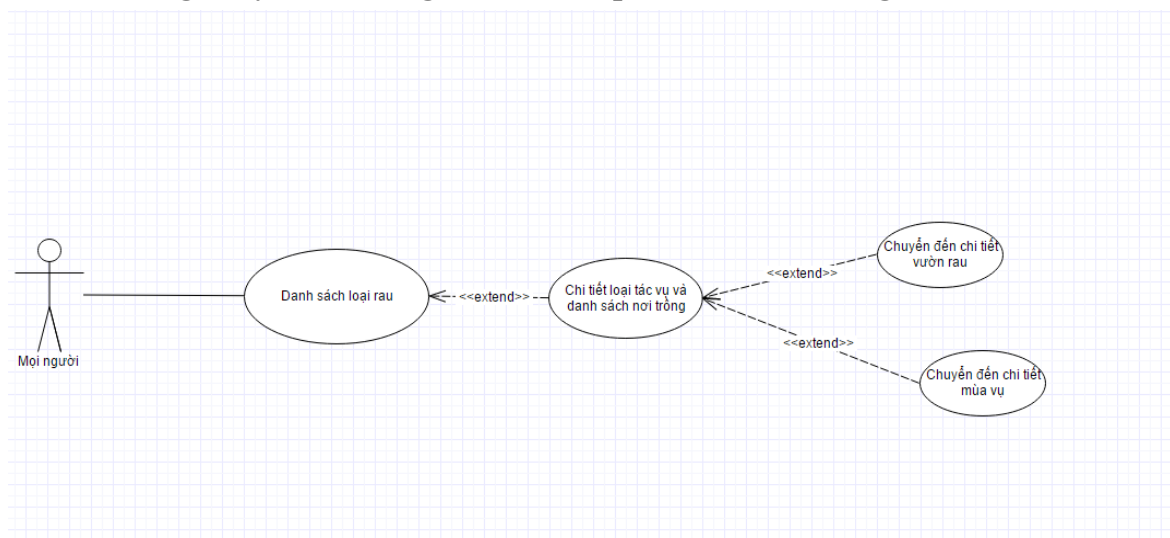


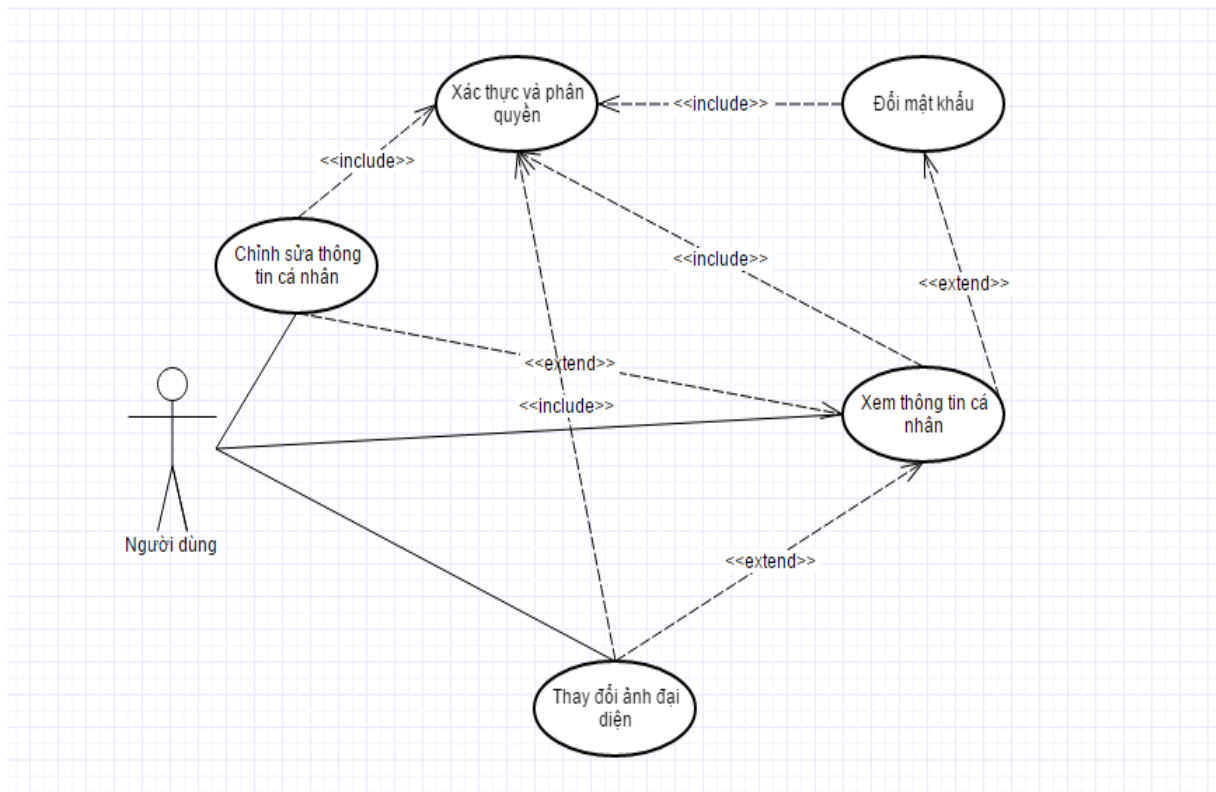
Sơ đồ 3.7 Sơ đồ Use Case quản lý duyệt vườn.

**+ Chức năng quản lý thiết bị sản xuất:**

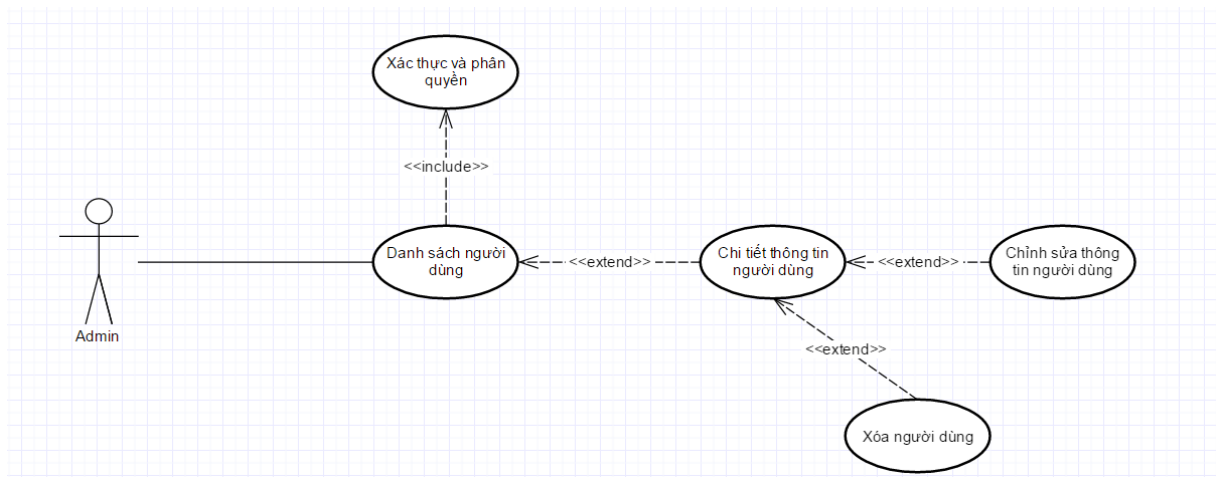


Sơ đồ 3.8 Sơ đồ Use Case quản lý thiết bị sản xuất.

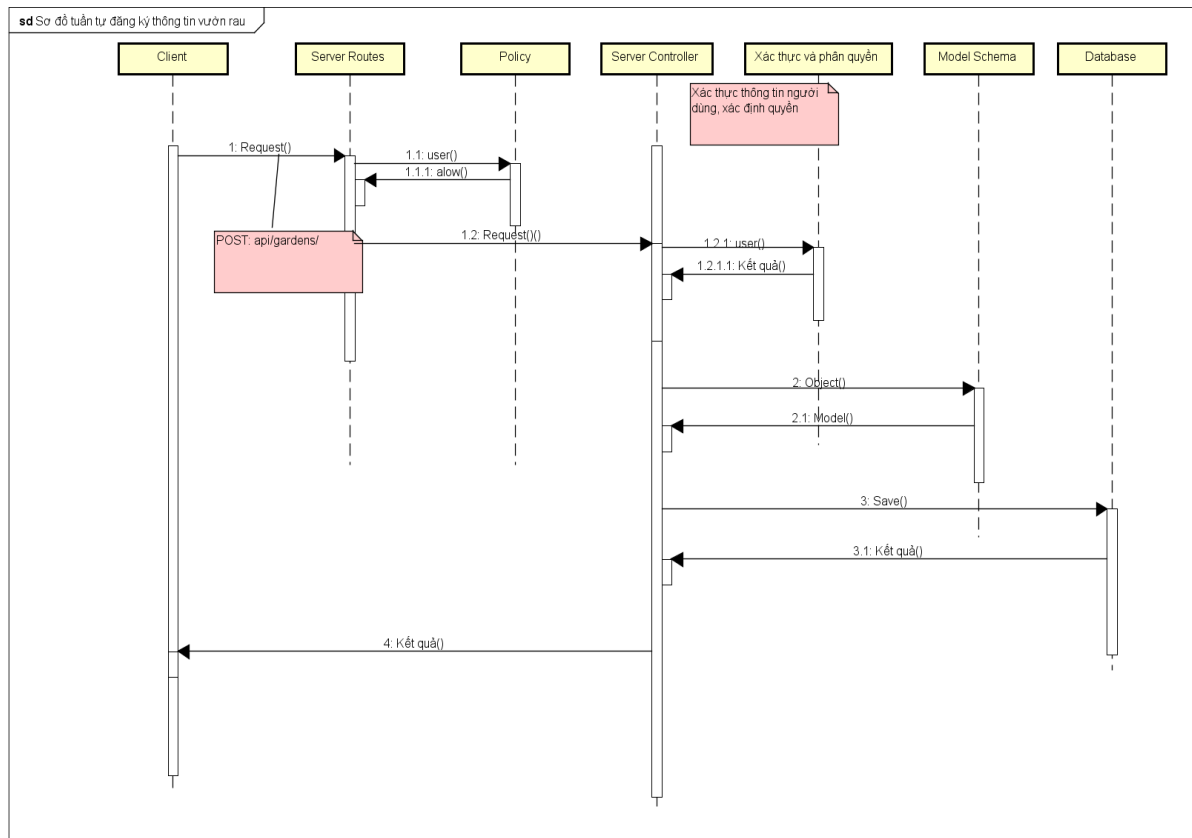
**+ Chức năng quản lý danh sách loại tác vụ:***Sơ đồ 3.9 Sơ đồ Use Case danh sách loại tác vụ***+ Chức năng truy xuất thông tin các sản phẩm của hệ thống vườn rau:***Sơ đồ 3.10 Sơ đồ Use case truy xuất thông tin sản phẩm của hệ thống.*

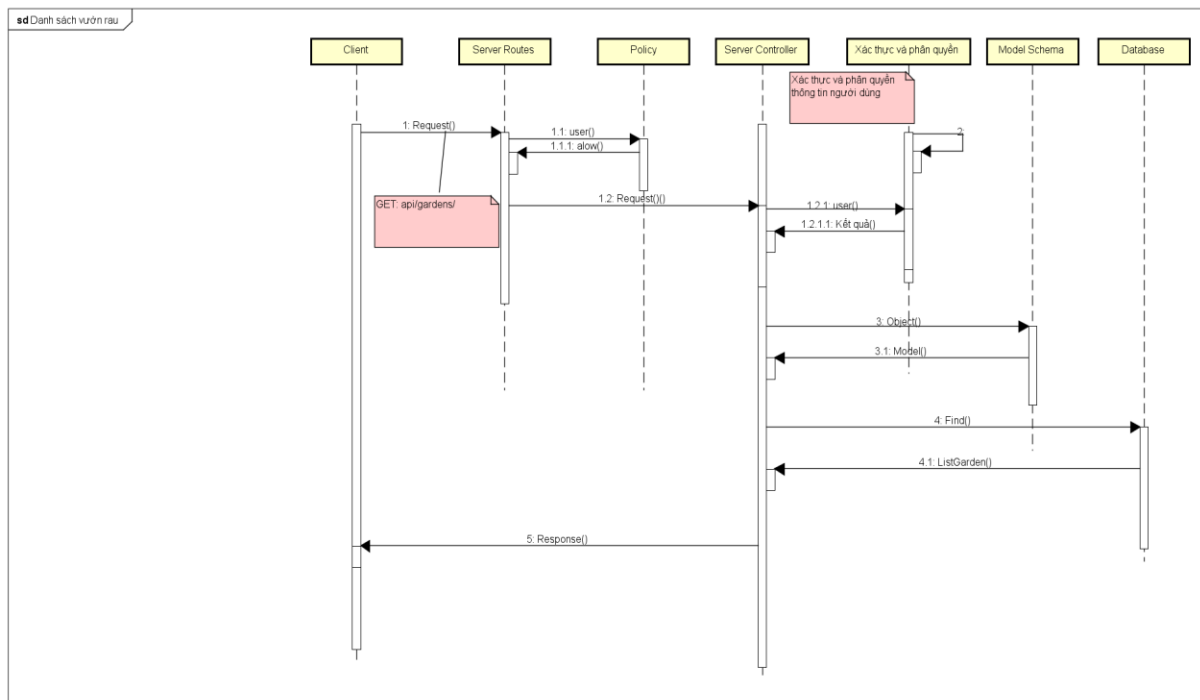
**+ Chức năng quản lý thông tin cá nhân:**

Sơ đồ 3.11 Sơ đồ Use Case quản lý thông tin cá nhân

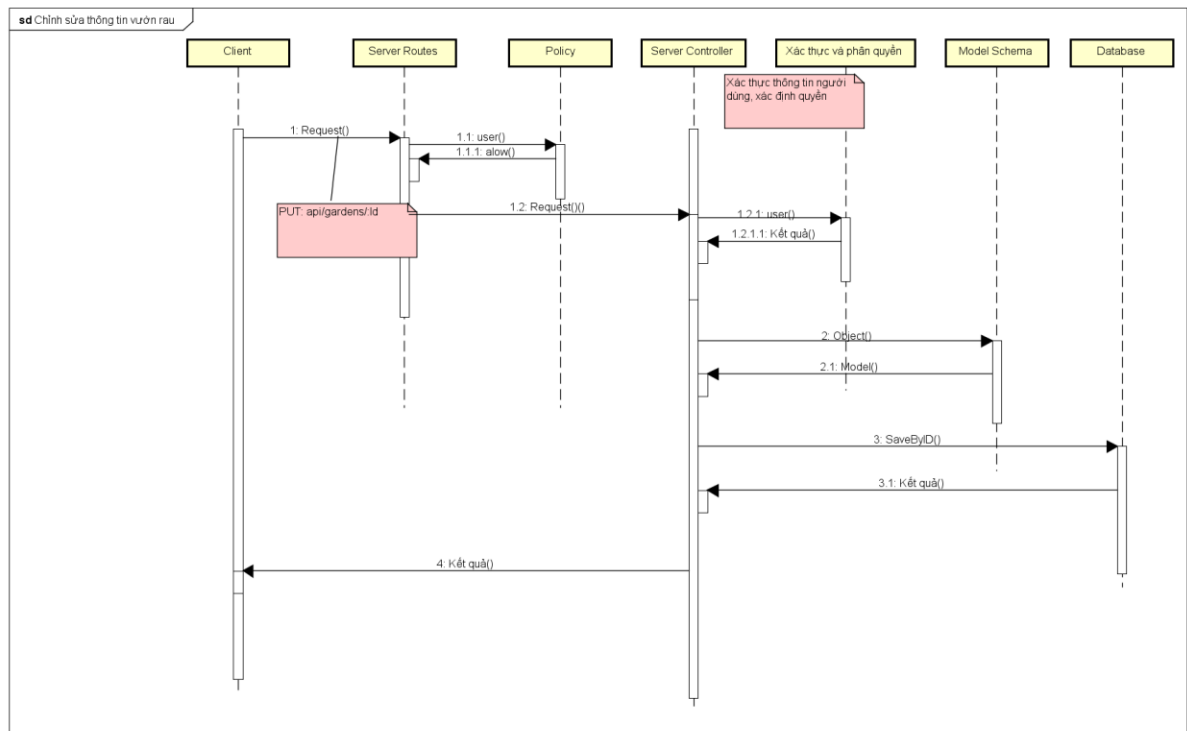
**+ Chức năng quản lý người dùng của quản trị viên:**

Sơ đồ 3.12 Sơ đồ Use Case quản người dùng của quản trị viên

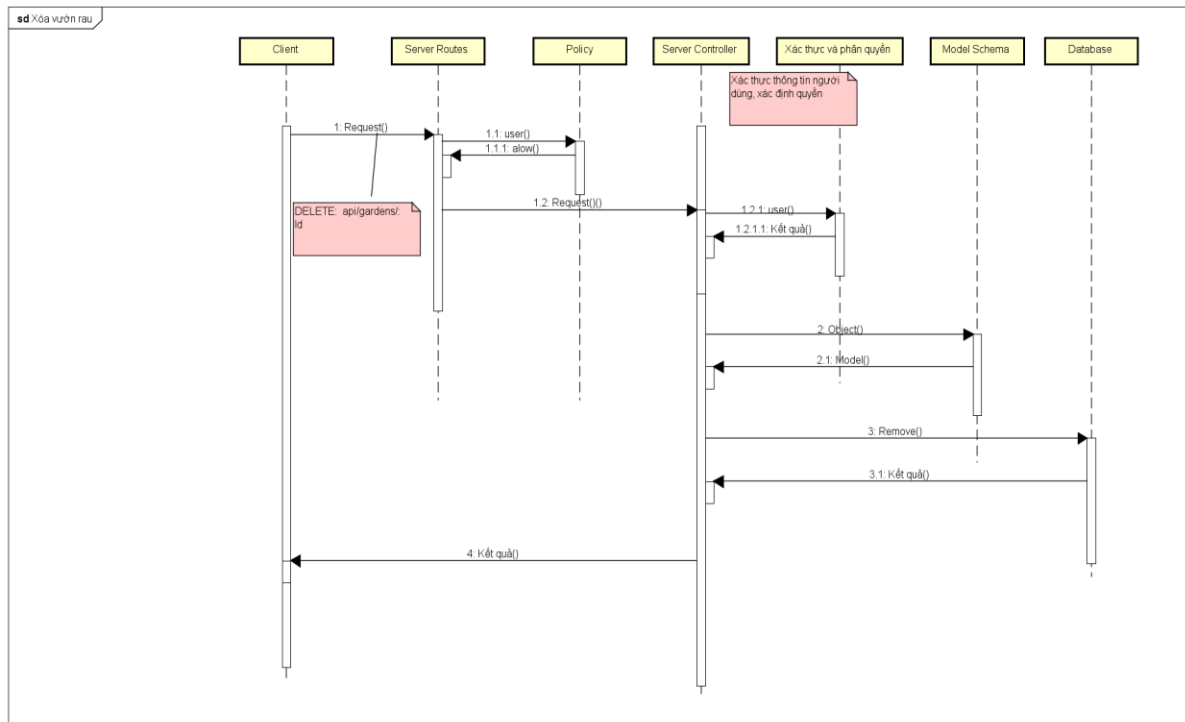
**3.1.4 Sơ đồ tuần tự:****Sơ đồ tuần tự của một số chức năng chính:****- Đăng ký thông tin vườn rau:***Sơ đồ 3.13 Sơ đồ tuần tự đăng ký vườn rau*

**- Liệt kê danh sách vườn rau:**

Sơ đồ 3.14 Sơ đồ tuần tự liệt kê danh sách vườn rau

**- Chỉnh sửa thông tin vườn rau:**

Sơ đồ 3.15 Sơ đồ tuần tự chỉnh sửa thông tin vườn rau

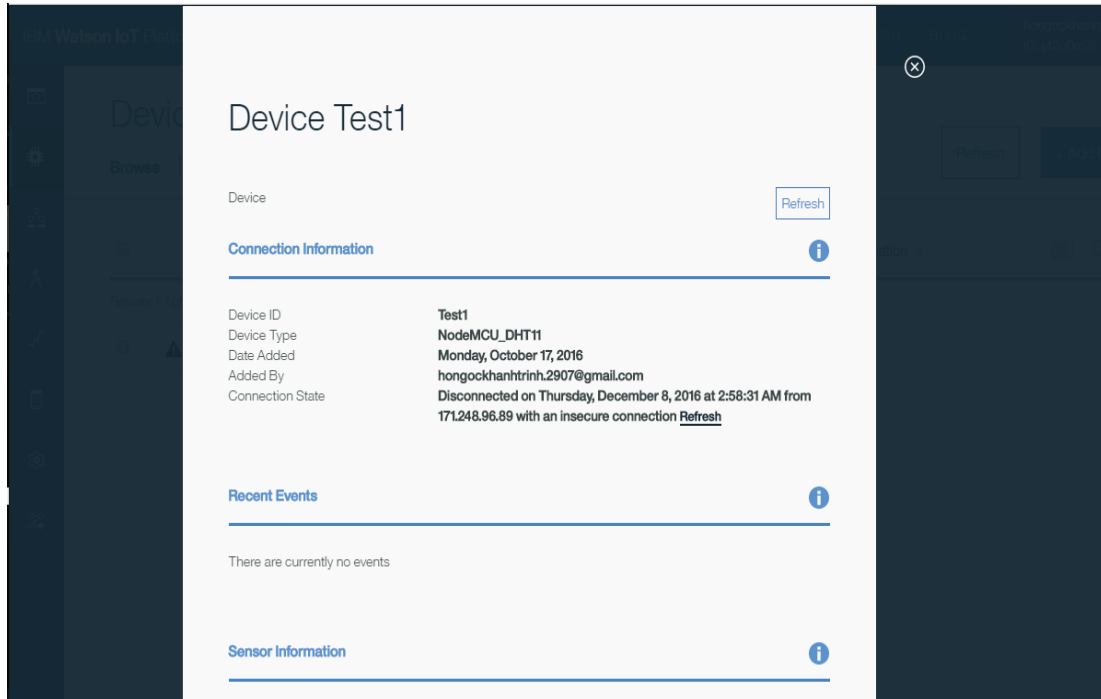
**- Xóa vườn rau:**

Sơ đồ 3.16 Sơ đồ tuần tự xóa vườn rau

### 3.2 Cài đặt ứng dụng nhúng:

#### 3.2.1 Cài đặt và cấu hình thiết bị:

- Thiết bị sử dụng: Node MCU, Cảm biến nhiệt độ độ ẩm DHT11.
  - Trình biên dịch: Arduino IDE.
  - Nền tảng đám mây: IBM Bluemix.
  - Các bước thực hiện:
- + Đăng ký tài khoản IBM Bluemix và tạo dịch vụ IBM IoT Watson Platform.



Hình 3.1 Cấu hình IBM IoT Watson Platform

+ Cấu hình kết nối với IBM IoT Watson Platform:

```
#define ORG "42o0m3"
#define DEVICE_TYPE "NodeMCU_DHT11"
#define DEVICE_ID "Test1"
#define TOKEN "Qk71CVHJ8xlm9s82pC"
```

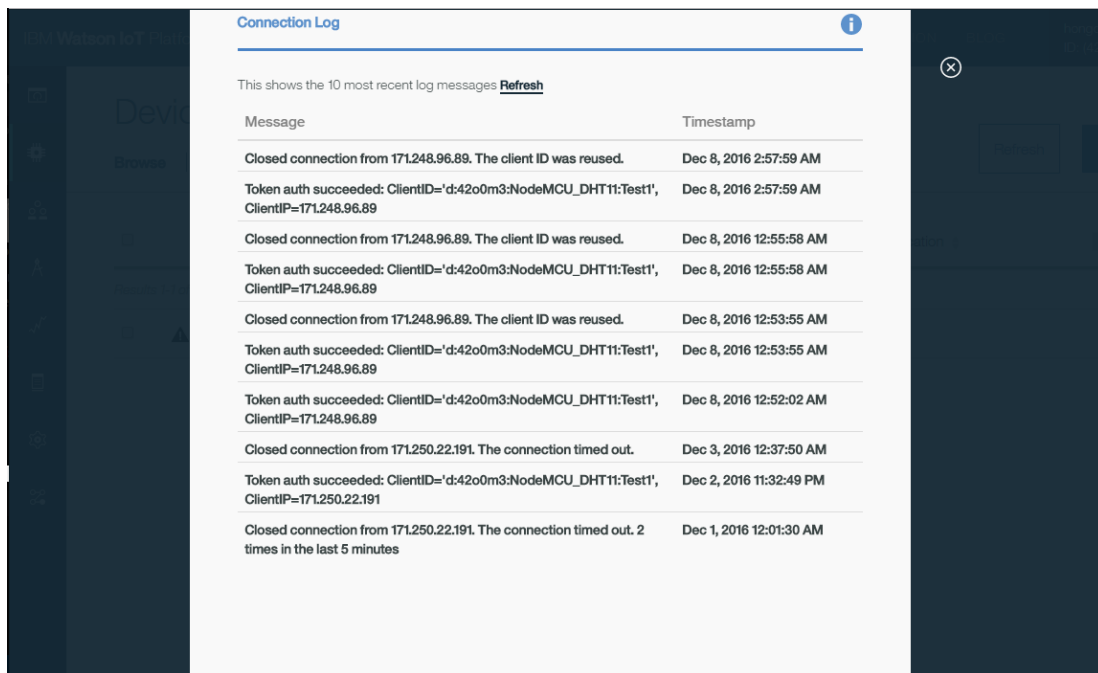
+ Cấu hình kết nối wifi :

```
const char* ssid = "*ssid*";
const char* password = "*password*";
```

+ Tiến hành đọc dữ liệu từ cảm biến DHT11 sau đó chèn vào payload để gửi lên server cứ mỗi 5 giây một lần.



+ Kết quả:



The screenshot shows the 'Connection Log' interface in Bluemix. It displays a table of the 10 most recent log messages. The messages include connection status updates, token authentication success, and connection timeouts. The table has two columns: 'Message' and 'Timestamp'.

Message	Timestamp
Closed connection from 171.248.96.89. The client ID was reused.	Dec 8, 2016 2:57:59 AM
Token auth succeeded: ClientID='d:42o0m3:NodeMCU_DHT11:Test1', ClientIP=171.248.96.89	Dec 8, 2016 2:57:59 AM
Closed connection from 171.248.96.89. The client ID was reused.	Dec 8, 2016 12:55:58 AM
Token auth succeeded: ClientID='d:42o0m3:NodeMCU_DHT11:Test1', ClientIP=171.248.96.89	Dec 8, 2016 12:55:58 AM
Closed connection from 171.248.96.89. The client ID was reused.	Dec 8, 2016 12:53:55 AM
Token auth succeeded: ClientID='d:42o0m3:NodeMCU_DHT11:Test1', ClientIP=171.248.96.89	Dec 8, 2016 12:53:55 AM
Token auth succeeded: ClientID='d:42o0m3:NodeMCU_DHT11:Test1', ClientIP=171.248.96.89	Dec 8, 2016 12:52:02 AM
Closed connection from 171.250.22.191. The connection timed out.	Dec 3, 2016 12:37:50 AM
Token auth succeeded: ClientID='d:42o0m3:NodeMCU_DHT11:Test1', ClientIP=171.250.22.191	Dec 2, 2016 11:32:49 PM
Closed connection from 171.250.22.191. The connection timed out. 2 times in the last 5 minutes	Dec 1, 2016 12:01:30 AM

Hình 3.2 Kết quả gửi dữ liệu từ ứng dụng nhúng lên Bluemix.

### 3.3 Cài đặt ứng dụng Web:

#### 3.3.1 Thiết lập môi trường chạy ứng dụng:

##### - Cài đặt NodeJs:

Trước tiên, truy cập website [nodejs.org](http://nodejs.org) để download phần mềm về máy. Sau khi tải về chúng ta tiến hành cài đặt bình thường như những phần mềm khác.

Để kiểm tra đã cài thành công hay chưa các bạn hãy vào cửa sổ cmd, di chuyển tới thư mục cài đặt và chạy dòng lệnh sau để kiểm tra phiên bản cài đặt

```
node -v
```

##### - Cài đặt và cấu hình MongoDB:

MongoDB dành cho Windows Server 2008 R2 edition (Ví dụ: 2008R2) chỉ chạy trên Windows Server 2008 R2, Windows 7 64-bit, và các phiên bản mới hơn của Windows. Các phiên bản này tận dụng các cải tiến gần đây cho nền tảng Windows và không thể hoạt động trên các phiên bản cũ của Windows.

MongoDB dành cho Windows 64-bit chạy trên bất kỳ phiên bản 64-bit của Windows mới hơn so với Windows XP, bao gồm Windows Server 2008 R2 và Windows 7 64-bit.

MongoDB dành cho Windows 32-bit chạy trên bất kỳ phiên bản 32-bit của Windows mới hơn so với Windows XP. Phiên bản 32-bit của MongoDB chỉ được dành cho hệ thống cũ và để sử dụng trong thử nghiệm và phát triển hệ thống. Phiên bản 32-bit của MongoDB chỉ hỗ trợ cơ sở dữ liệu nhỏ hơn 2GB.

+ Tiến hành tải tập tin cài đặt MongoDB tại đường dẫn:  
<https://www.mongodb.com/download-center#community>

+ Sau khi tải xong, tiến hành cài đặt theo các bước như mọi phần mềm bình thường khác.

+ Cấu hình MongoDB:

Chúng ta tạo một thư mục chứa dữ liệu của Database, thư mục chứa file log, file cấu hình

Tạo thư mục:

D:/MongoStore

Sau đó tạo 2 thư mục con

**data**

**log**

Và một file cấu hình:

**config.txt**

Tập tin config :

**config.txt**

```
1  ##store data
2  dbpath=D:\MongoStore\data
3
4  ##all output go here
5  logpath=D:\MongoStore\log\mongo.log
```

Vào cmd và gõ dòng lệnh cd tới thư mục bin của mongodb.

*cd C:\DevPrograms\MongoDB\bin*

Chạy file cấu hình mongodb với câu lệnh:

*mongod.exe --config D:\MongoStore\config.txt*

Tới đây việc cấu hình MongoDB thành công.

Có thể tải công cụ RoboMongo để sử dụng MongoDB một cách trực quan tại đường dẫn:

<http://robomongo.org/download.html>

**- Cài đặt gói thư viện:**

Các gói thư viện của đồ án được lưu trong tập tin package.json của thư mục chứa mã nguồn.

Ta tiến hành kết nối mạng, mở màn hình CMD, di chuyển đến thư mục tập tin package.json và chạy lệnh

**npm install**

Sau khi chạy xong thì ta đã cài đặt đầy đủ các gói thư viện để chạy chương trình demo

### 3.3.2 Cài đặt các chức năng của ứng dụng:

Hệ thống được chia ra làm hai ứng dụng là client và server. Ứng dụng client sẽ chịu trách nhiệm về mặt tương tác với người dùng thông qua thành phần giao diện người dùng và thành phần xử lý để giao tiếp với ứng dụng server thông qua REST API mà ứng dụng server cung cấp. Ứng dụng server là nơi thực hiện các xử lý nghiệp vụ của hệ thống, xử lý các phiên làm việc với client cũng như giao tiếp với hệ quản trị cơ sở dữ liệu.

#### - Cấu hình kết nối cơ sở dữ liệu MongoDB:

A screenshot of a code editor window titled 'db.js'. The code is in JavaScript and shows the setup for connecting to a MongoDB database using the mongoose library. The code includes the following lines:

```
1 var mongoose = require('mongoose');
2 var url = process.env.MONGODB_URI || 'mongodb://127.0.0.1:27017/vegetable-dev';
3 module.exports = {
4   init: init
5 };
6
7 function init() {
8   mongoose.connect(url); // connect to database
9 }
```

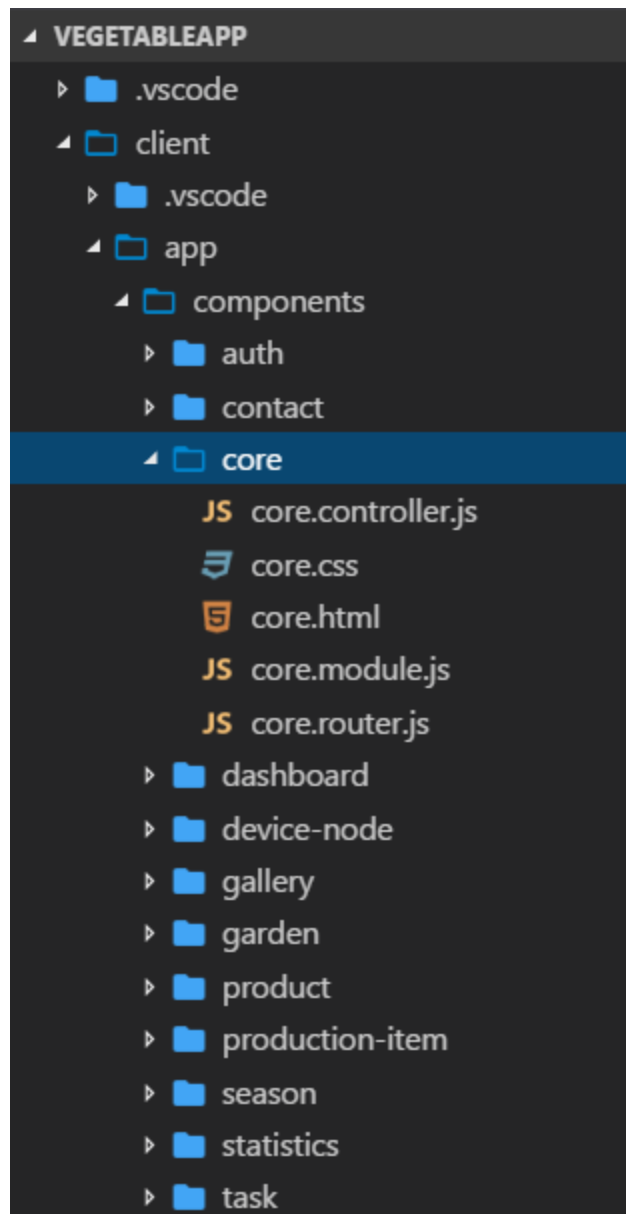
Hình 3.3 Cấu hình kết nối cơ sở dữ liệu

Dựa vào địa chỉ uri để kết nối đến cơ sở dữ liệu MongoDB tại localhost.

Việc quản lý kết nối sẽ do mô-đun mongoose đảm nhiệm.

#### - Cấu trúc cây thư mục phía client :

+ Ứng dụng được chia nhỏ ra thành từng mô-đun nhỏ. Mỗi thư mục mô-đun chứa các tập tin xử lý theo mô hình MVC.

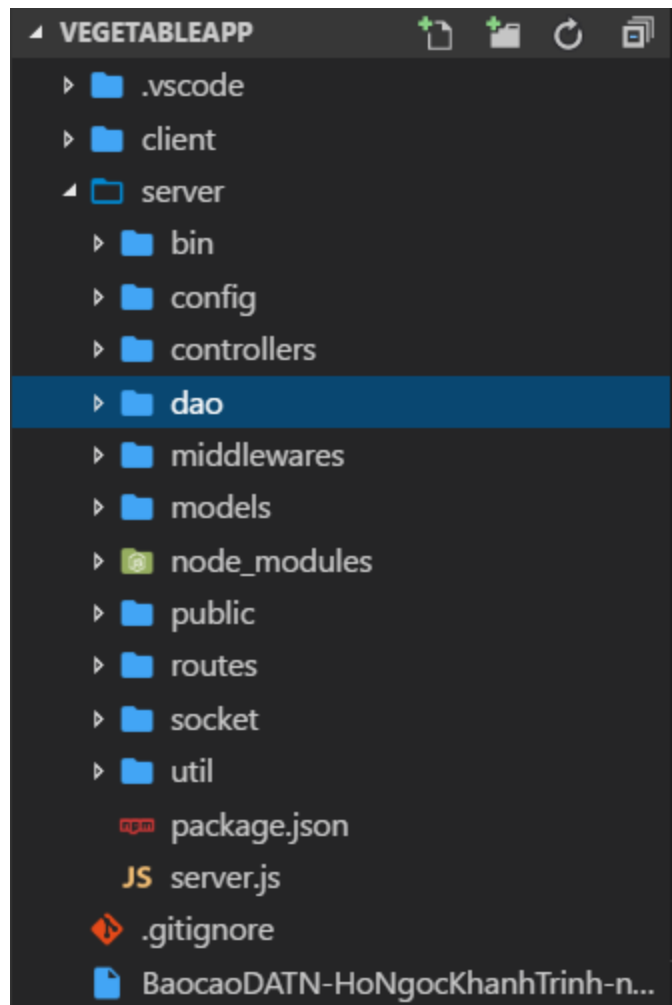


Hình 3.4 Cấu trúc cây thư mục client

#### - Cấu trúc cây thư mục phía server:

Các thư mục chính:

- + Thư mục “**config**” chứa các tập tin xử lý cấu hình kết nối cơ sở dữ liệu, cấu hình server của ứng dụng.
- + Thư mục “**controllers**” chứa các tập tin xử lý lô-gíc nghiệp vụ của ứng dụng.
- + Thư mục “**dao**” chứa các tập tin xử lý truy cập dữ liệu của ứng dụng.
- + Thư mục “**models**” chứa các tập tin xử lý schema của các collection của cơ sở dữ liệu.
- + Thư mục “**routes**” chứa các tập tin xử lý chia các REST API của ứng dụng.



Hình 3.5 Cấu trúc cây thư mục phía server

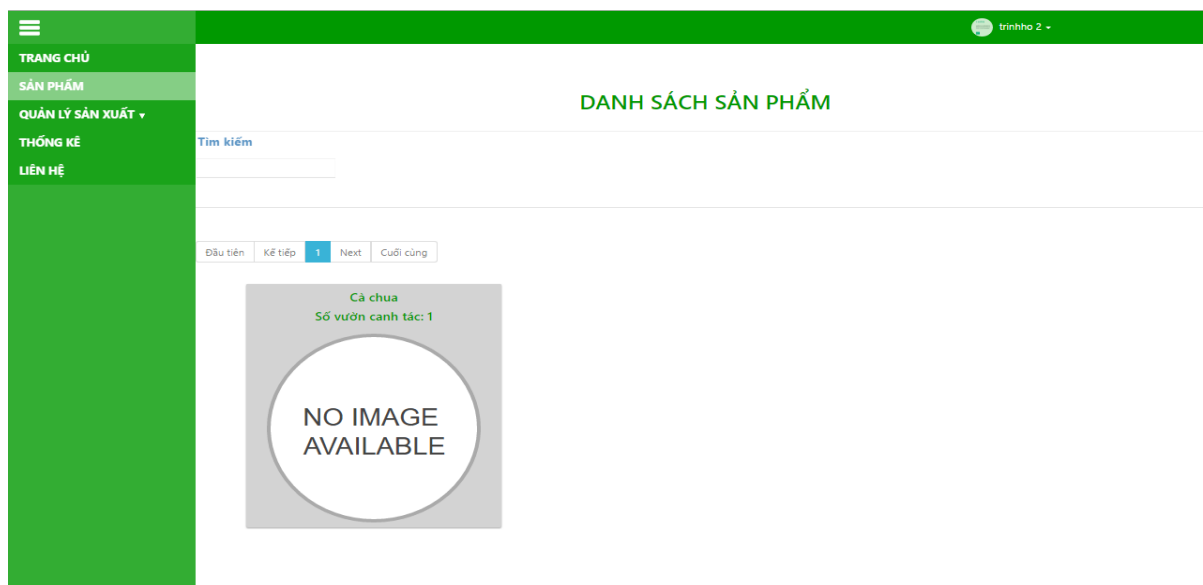
### 3.4 Kết quả thu được:

#### 3.4.1 Màn hình chức năng:

##### - Màn hình hiển thị danh sách sản phẩm:

Tại màn hình này người dùng sẽ xem được danh sách sản phẩm hiện có kèm theo thông tin cơ bản của sản phẩm.

Người dùng cũng có thể tìm kiếm sản phẩm khi nhập từ khóa vào ô tìm kiếm.



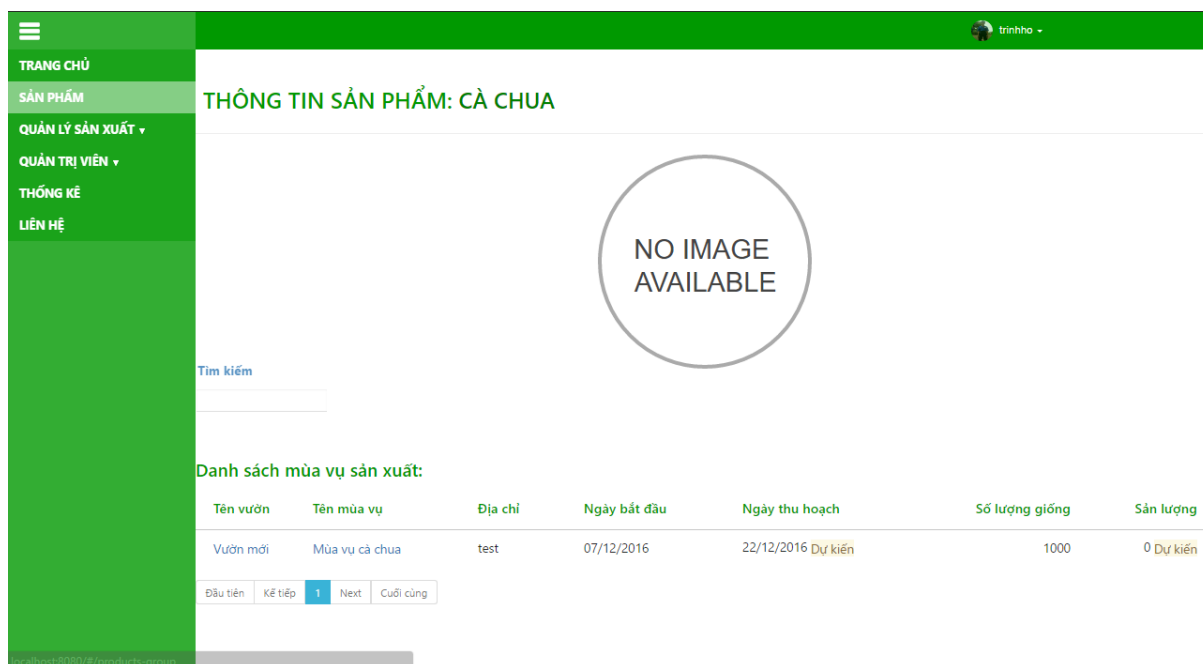
Hình 3.6 Màn hình thị danh sách sản phẩm.

#### - Màn hình thông tin sản phẩm:

Tại màn hình này người dùng có thể xem được thông tin của sản phẩm theo từng mùa vụ sản xuất, từ đó biết được nguồn gốc của sản phẩm.

Người dùng có thể xem thông tin vườn cũng như mùa vụ đang sản xuất sản phẩm bằng cách nhấp vào liên kết ở tên vườn và tên mùa vụ.

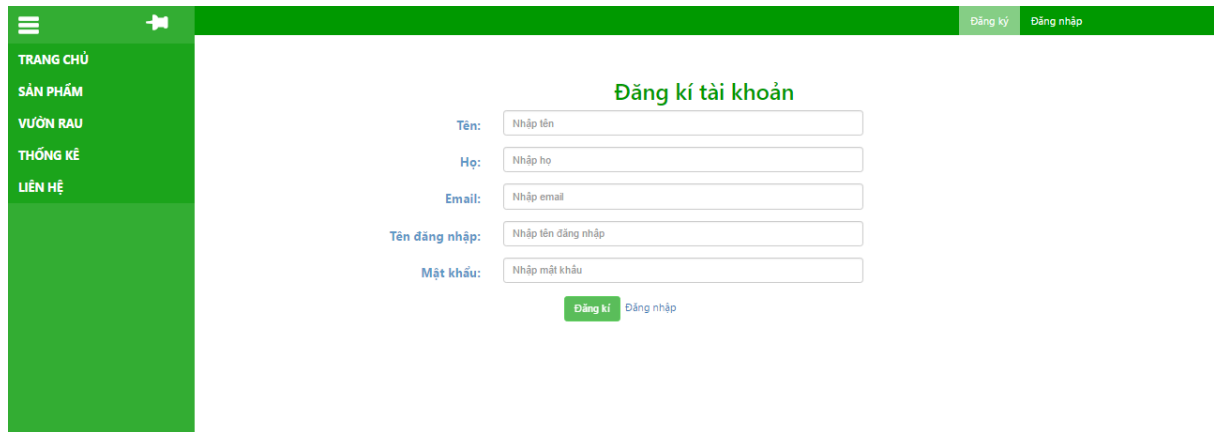
Người dùng cũng có thể tìm kiếm thông tin sản phẩm theo từ khóa.



Hình 3.7 Màn hình thông tin sản phẩm.

**- Đăng ký tài khoản:**

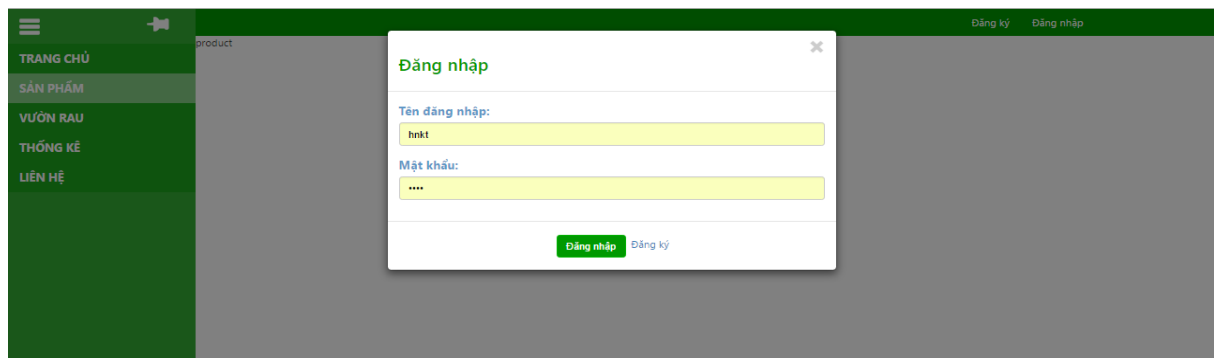
Người dùng có thể đăng ký tài khoản sau khi nhập đầy đủ thông tin và nhấn nút đăng ký.

The screenshot shows a web application interface with a green sidebar on the left containing menu items: TRANG CHỦ, SẢN PHẨM, VƯỜN RAU, THỐNG KÊ, and LIÊN HỆ. The main content area has a green header with 'Đăng ký' and 'Đăng nhập' links. The central form is titled 'Đăng ký tài khoản' and includes five input fields: 'Tên:' (Name), 'Họ:' (Surname), 'Email:', 'Tên đăng nhập:' (Username), and 'Mật khẩu:' (Password). Each field has a placeholder text 'Nhập [field name]'. At the bottom of the form are two buttons: 'Đăng ký' (green) and 'Đăng nhập' (blue).

Hình 3.8 Màn hình đăng ký tài khoản

**- Đăng nhập:**

Người dùng nhập tên đăng nhập và mật khẩu để đăng nhập vào hệ thống, nếu nhập sai thông tin thì hệ thống sẽ hiển thị lỗi cho người dùng biết, nếu đăng nhập thành công, hệ thống sẽ chuyển về trang trước đó.

The screenshot shows the same web application interface as Figure 3.8, but with a modal window titled 'Đăng nhập' (Login) open in the center. The modal contains two input fields: 'Tên đăng nhập:' (Username) with the value 'hnkt' and 'Mật khẩu:' (Password) with masked characters '\*\*\*\*'. At the bottom of the modal are two buttons: 'Đăng nhập' (green) and 'Đăng ký' (blue). The background of the page is dimmed.

Hình 3.9 Màn hình đăng nhập

**- Đăng ký vườn rau:**

Chủ vườn rau nhập đầy đủ thông tin vườn, chọn vị trí vườn của mình trên bản đồ để gửi yêu cầu đăng ký vườn rau của mình lên hệ thống.

**Tạo vườn**

Tên vườn: (\*)

Vĩ độ: (\*)

Kinh độ: (\*)

Địa chỉ: (\*)

Diện tích canh tác (m²): (\*)

Chọn loại rau canh tác:

Chọn loại thiết bị:

Mô tả:

**Tạo**

Hình 3.10 Màn hình đăng ký vườn rau

- **Chỉnh sửa vườn:**

**CHỈNH SỬA VƯỜN**

Tên vườn

Vĩ độ

Kinh độ

Địa chỉ

Diện tích canh tác m²

Chọn loại rau canh tác:

Chọn loại thiết bị:

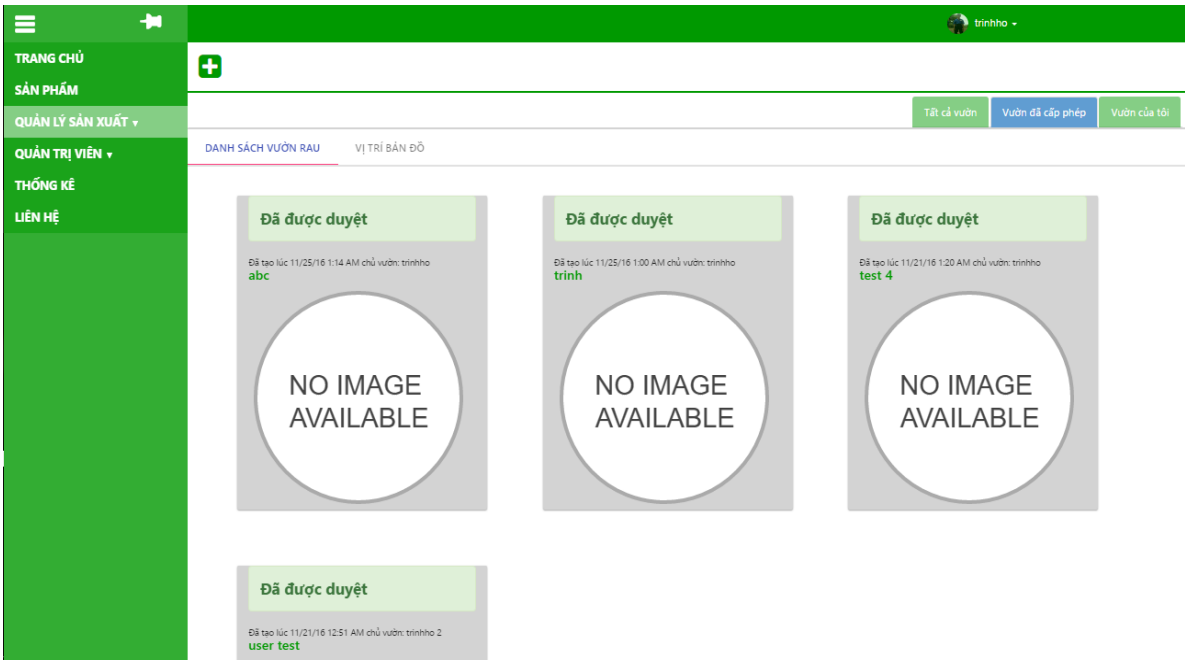
Mô tả

**Cập nhật**

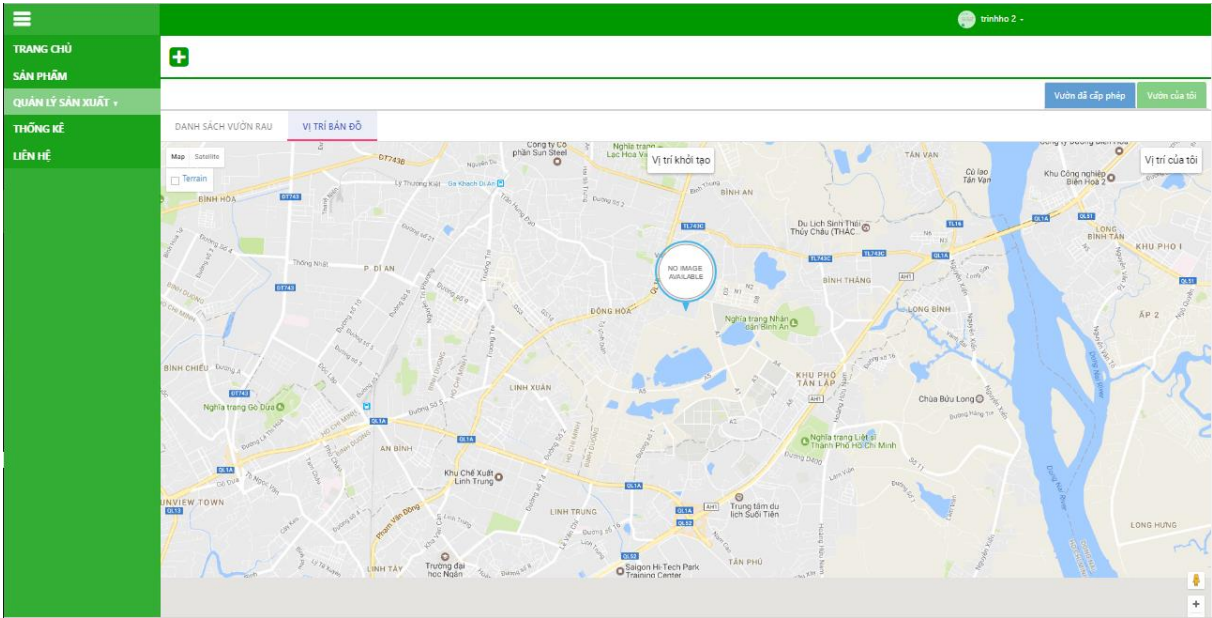
Hình 3.11 Màn hình chỉnh sửa vườn.



- Danh sách vườn rau:

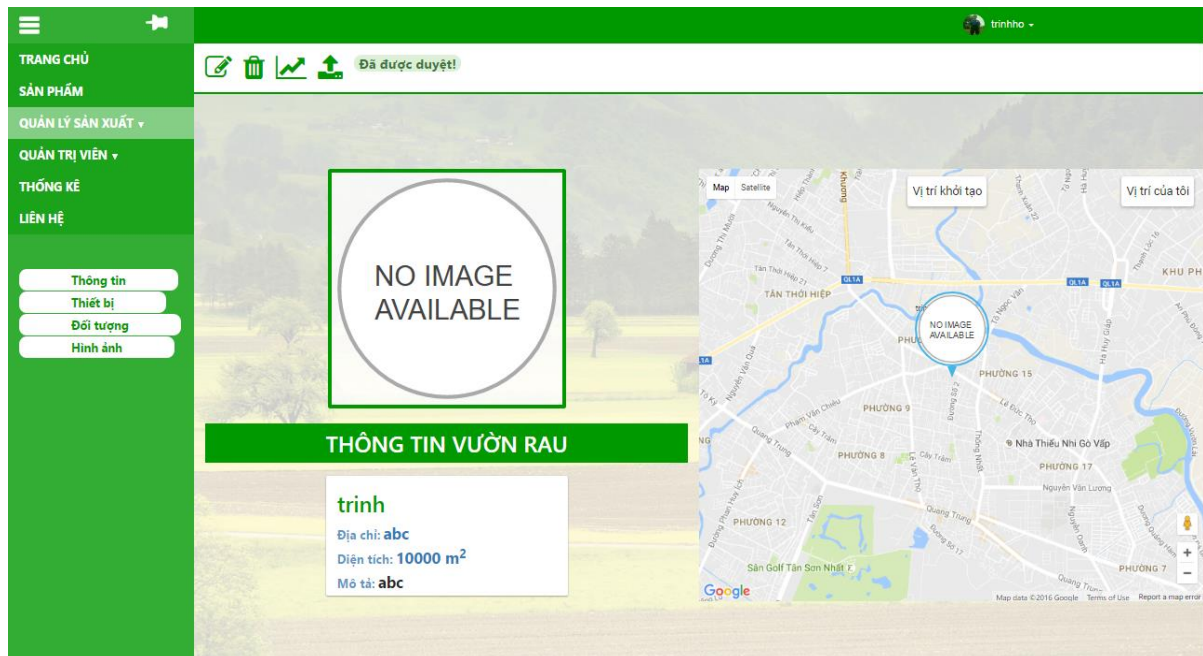


Hình 3.12 Màn hình hiển thị danh sách vườn rau.



Hình 3.13 Màn hình bản đồ vị trí các vườn rau.

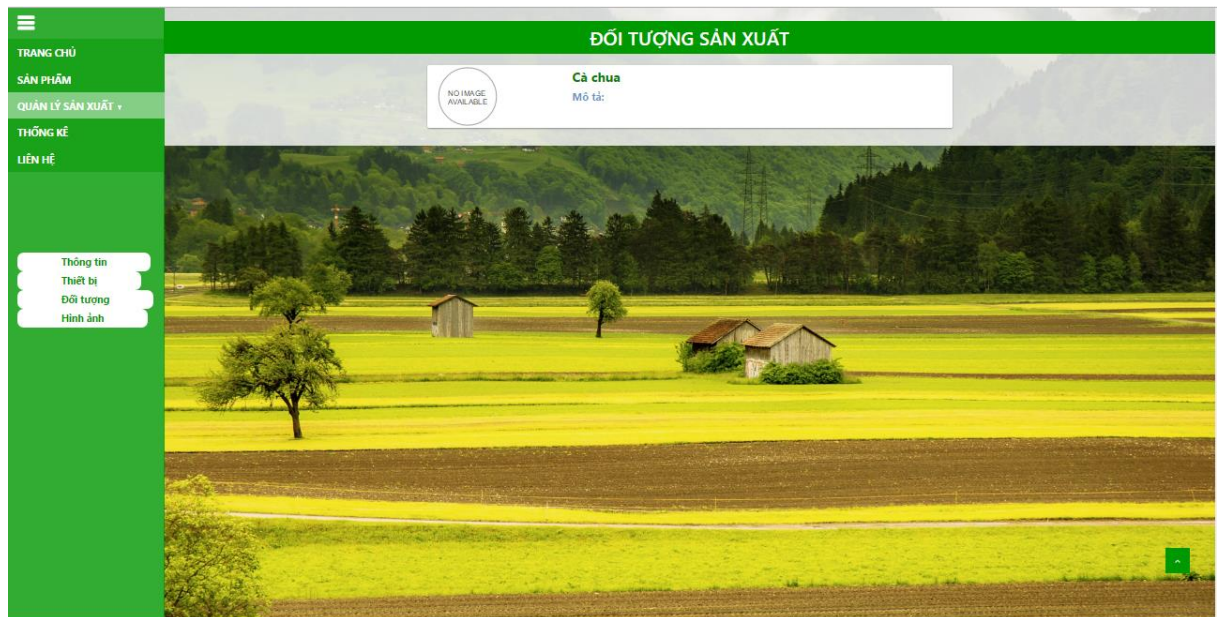
- Chi tiết vườn:



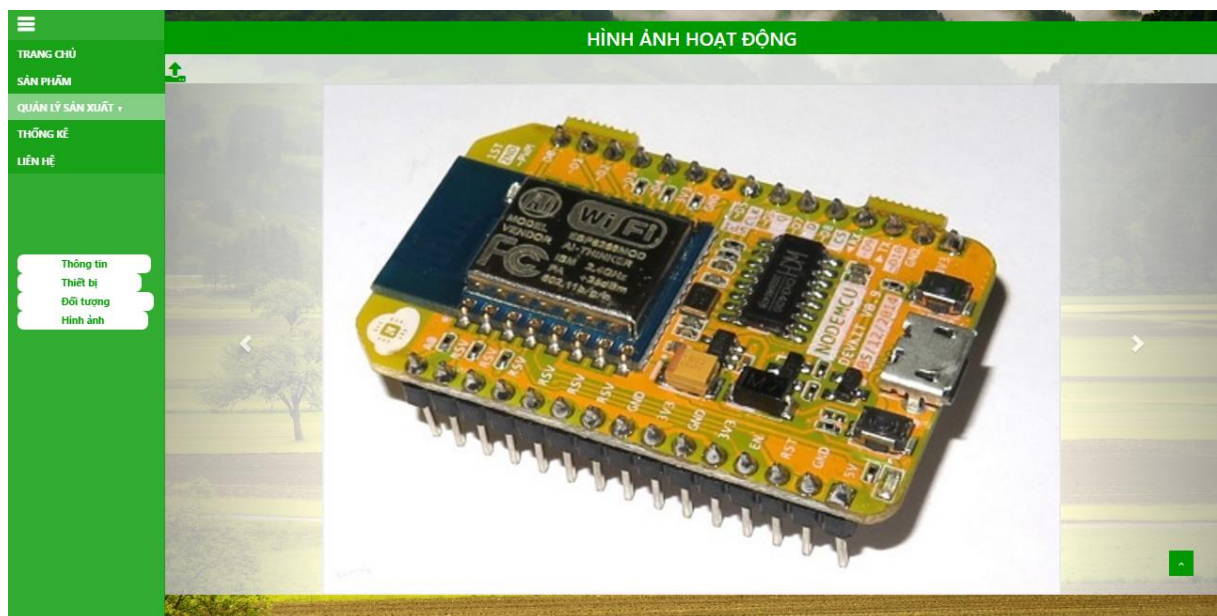
Hình 3.14 Màn hình thông tin vườn rau.



Hình 3.15 Màn hình thiết bị sử dụng của vườn rau.



Hình 3.16 Màn hình đối tượng sản xuất của vườn rau.



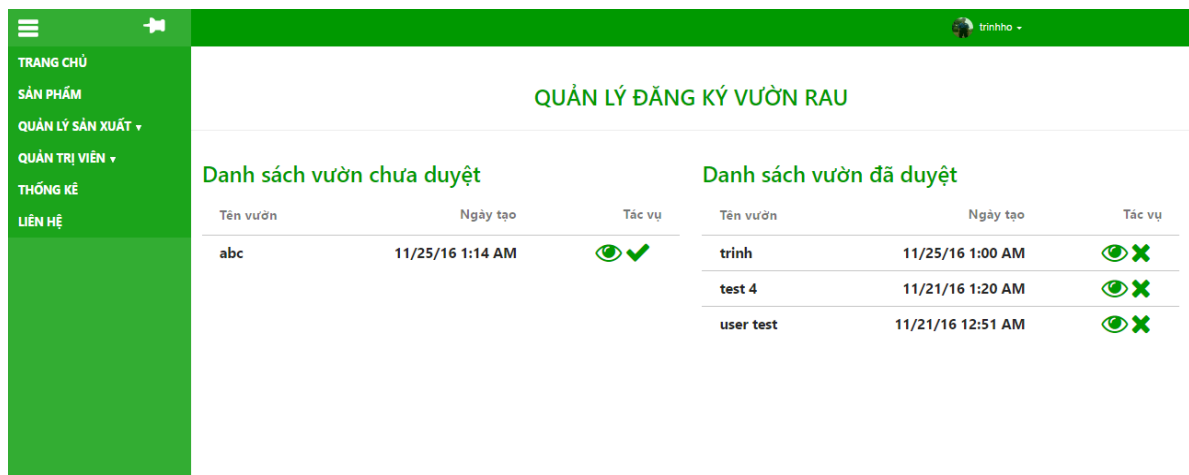
Hình 3.17 Màn hình hình ảnh hoạt động của vườn.











**- Xét duyệt đăng ký vườn rau:**

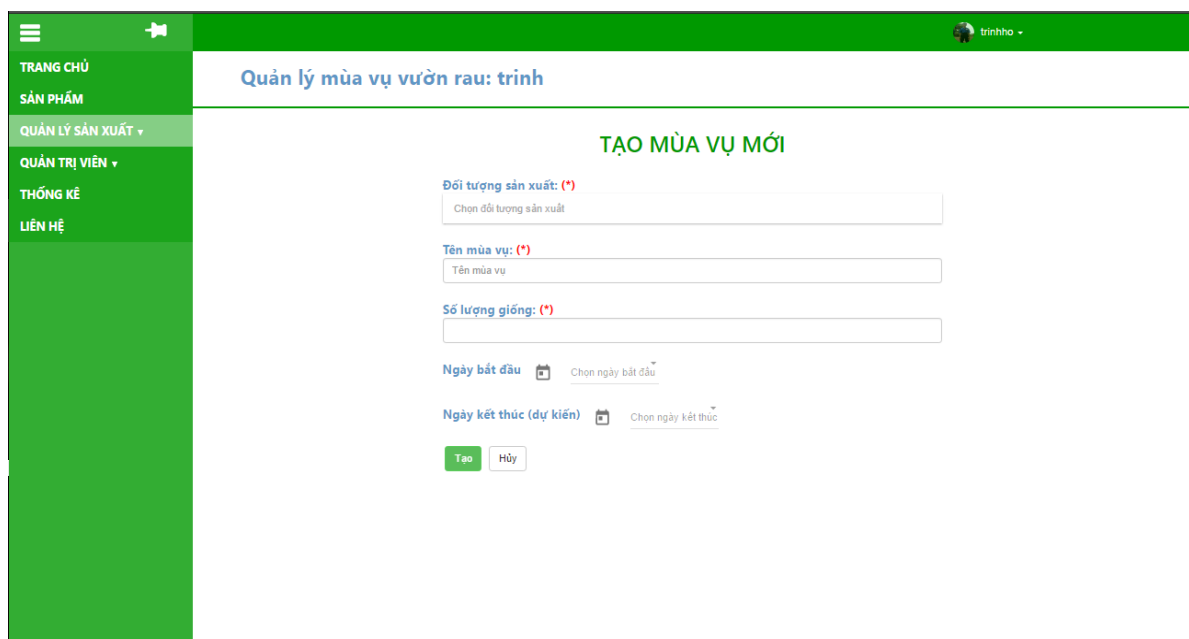
Hệ thống sẽ hiển thị danh sách vườn đang chờ duyệt ở bên trái và danh sách vườn đã duyệt bên phải.

Quản trị viên có thể duyệt vườn hoặc bỏ duyệt vườn, sau khi thực hiện thao tác thì hệ thống tự động cập nhật danh sách vườn chưa duyệt và đã duyệt.



Danh sách vườn chưa duyệt			Danh sách vườn đã duyệt		
Tên vườn	Ngày tạo	Tác vụ	Tên vườn	Ngày tạo	Tác vụ
abc	11/25/16 1:14 AM	 	trinh	11/25/16 1:00 AM	 
			test 4	11/21/16 1:20 AM	 
			user test	11/21/16 12:51 AM	 

Hình 3.18 Màn hình xét duyệt đăng ký vườn rau.

**- Quản lý mùa vụ:**


**Quản lý mùa vụ vườn rau: trinh**


**TẠO MÙA VỤ MỚI**

Đối tượng sản xuất: (\*)  
Chọn đối tượng sản xuất

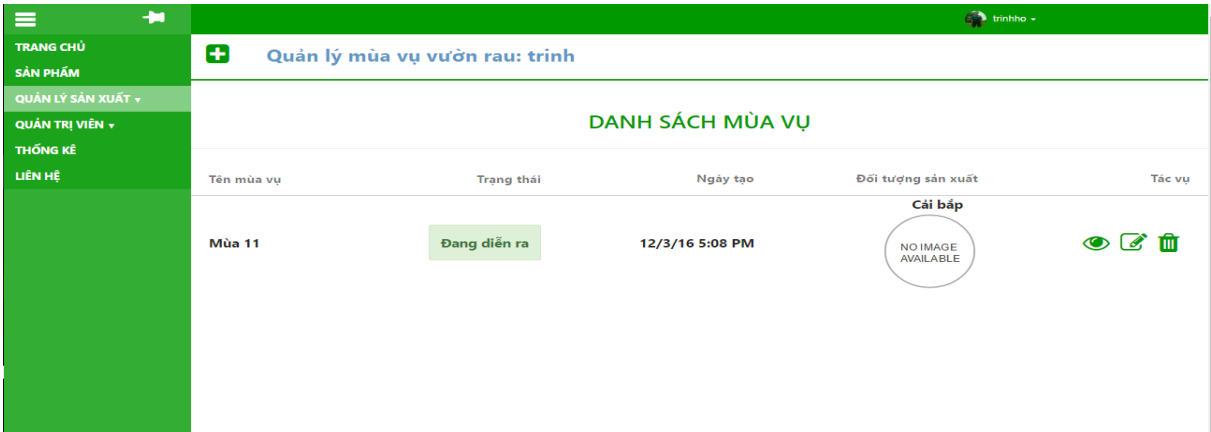
Tên mùa vụ: (\*)  
Tên mùa vụ

Số lượng giống: (\*)

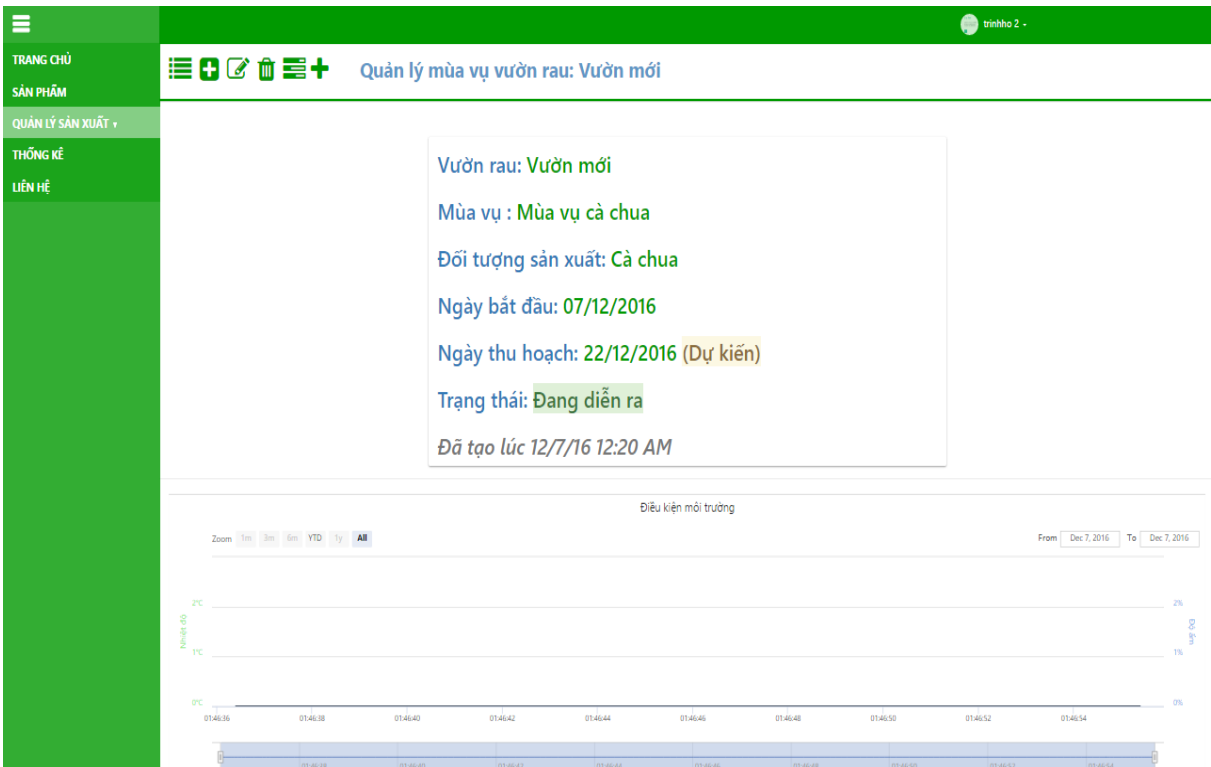
Ngày bắt đầu  Chọn ngày bắt đầu

Ngày kết thúc (dự kiến)  Chọn ngày kết thúc

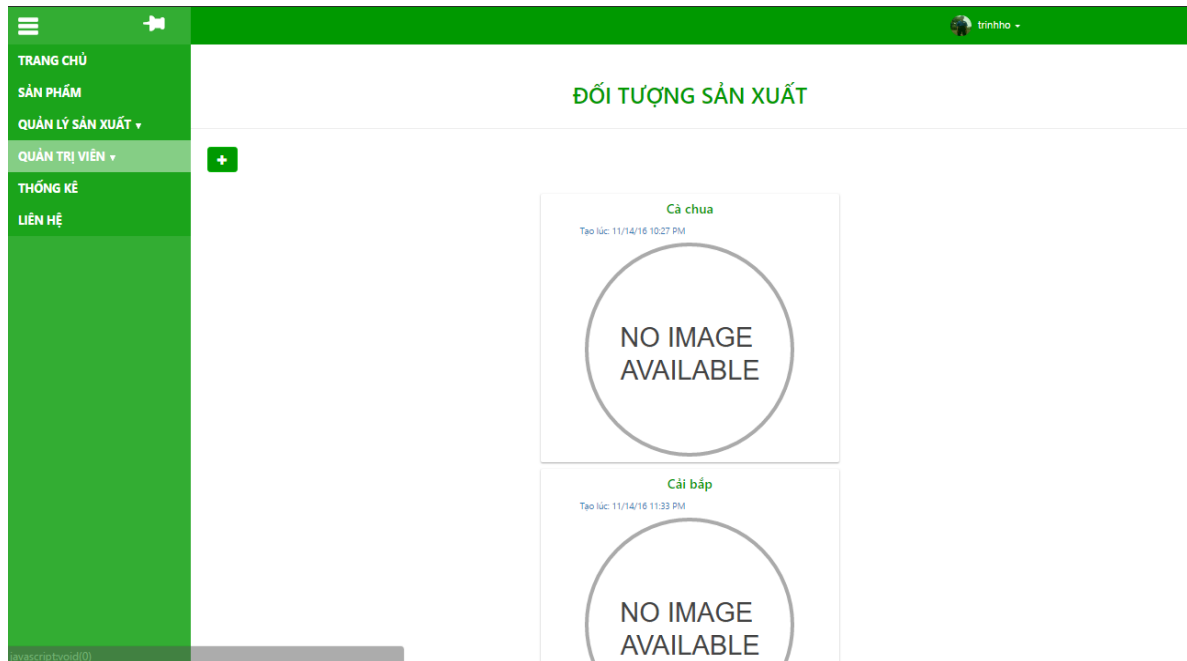
Hình 3.19 Màn hình tạo mùa vụ.



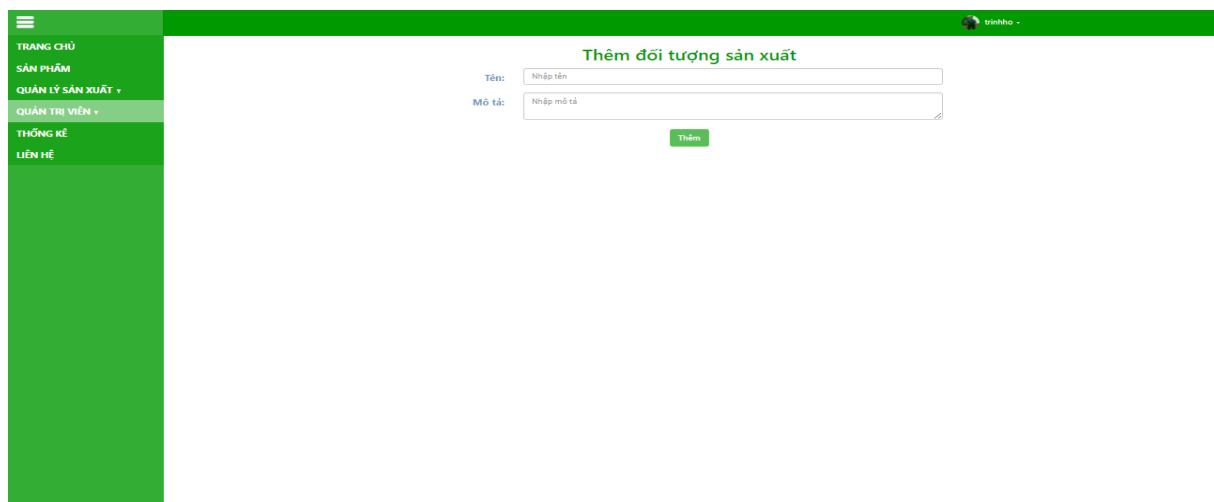
Hình 3.20 Màn hình hiển thị danh sách mùa vụ.



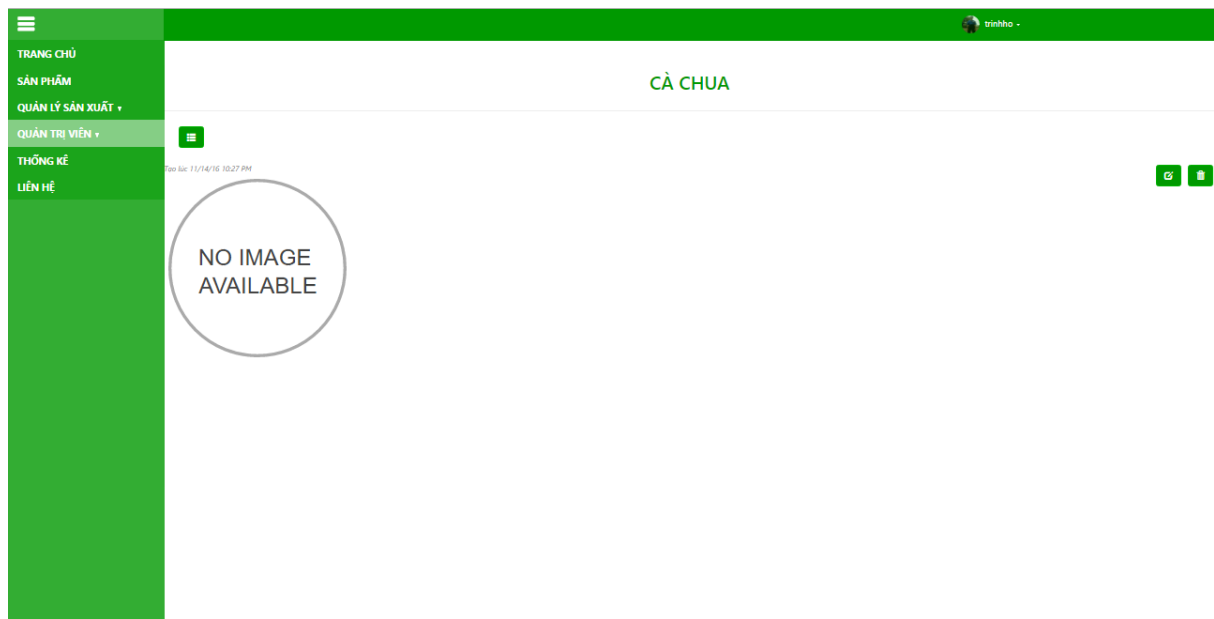
Hình 3.21 Màn hình chi tiết mùa vụ.

**- Quản lý đối tượng sản xuất:**

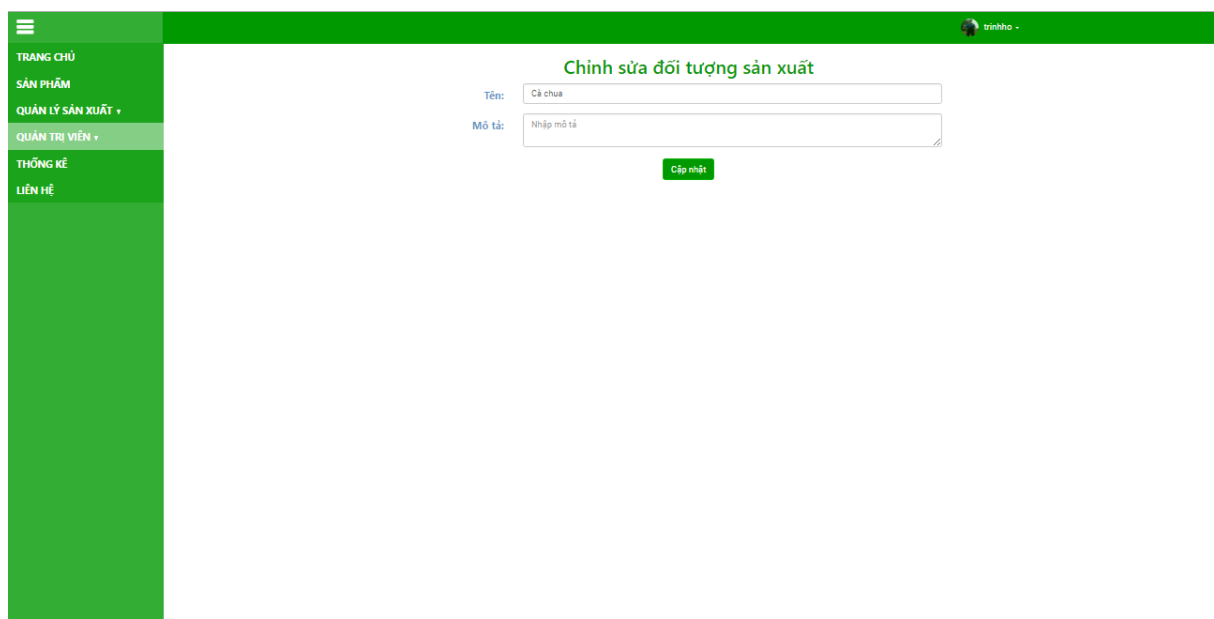
Hình 3.22 Màn hình hiển thị danh sách đối tượng sản xuất.



Hình 3.23 Màn hình thêm đối tượng sản xuất.



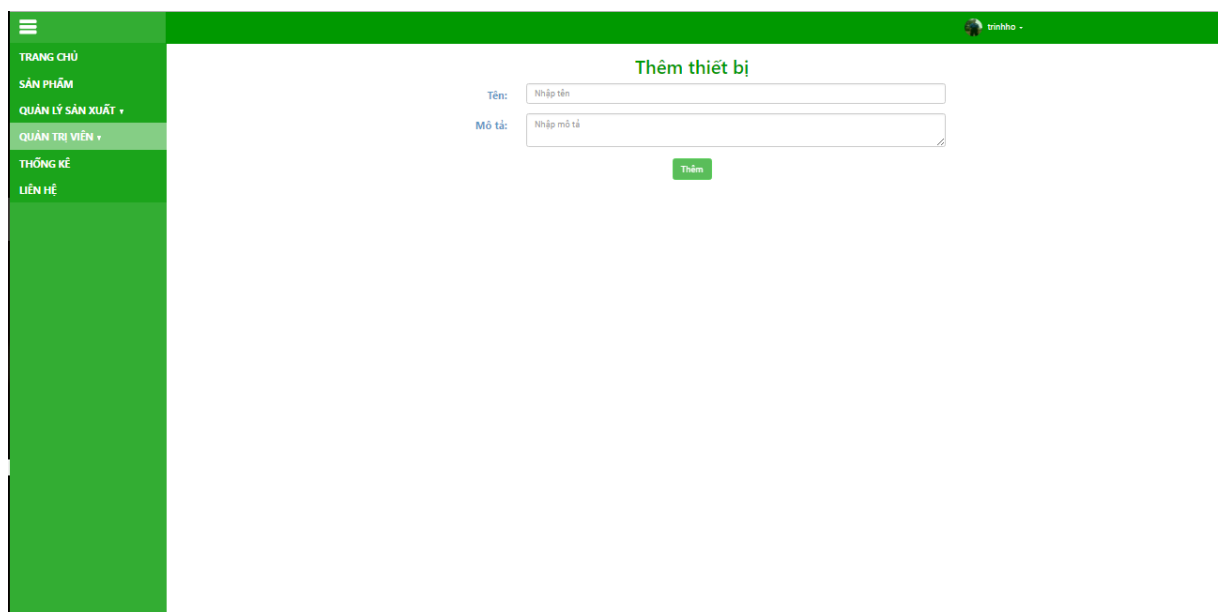
Hình 3.24 Màn hình chi tiết đối tượng sản xuất.



Hình 3.25 Màn hình chỉnh sửa đối tượng sản xuất

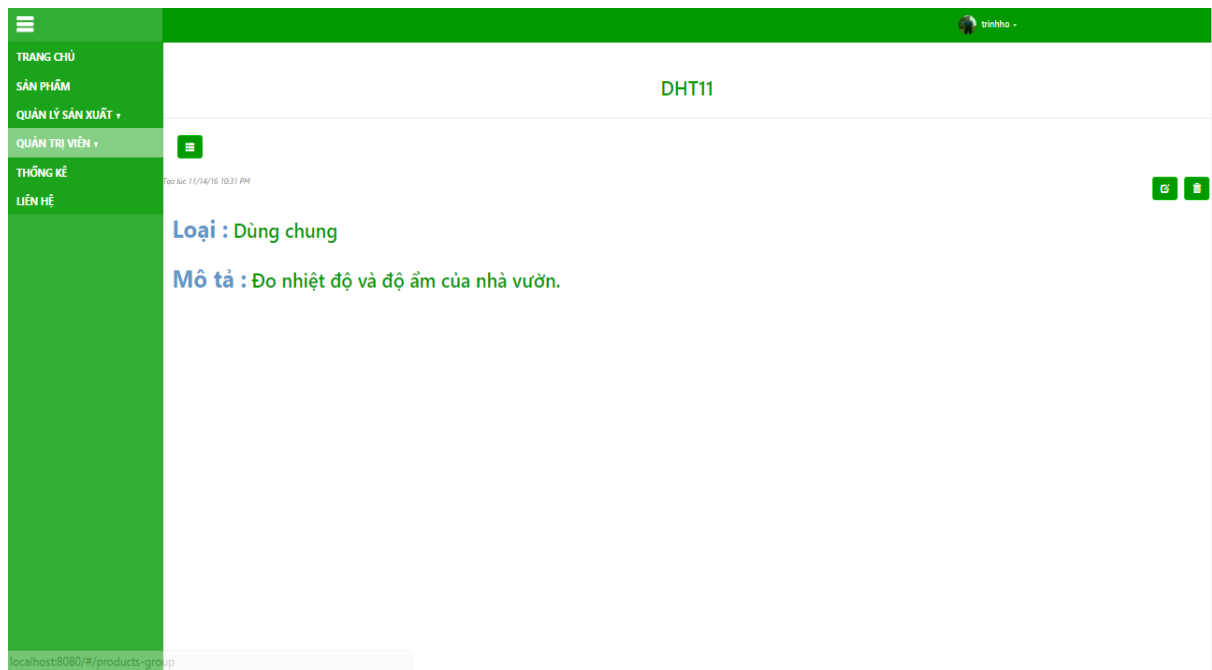
**- Quản lý thiết bị sản xuất:**

Hình 3.26 Màn hình hiển thị danh sách thiết bị sản xuất

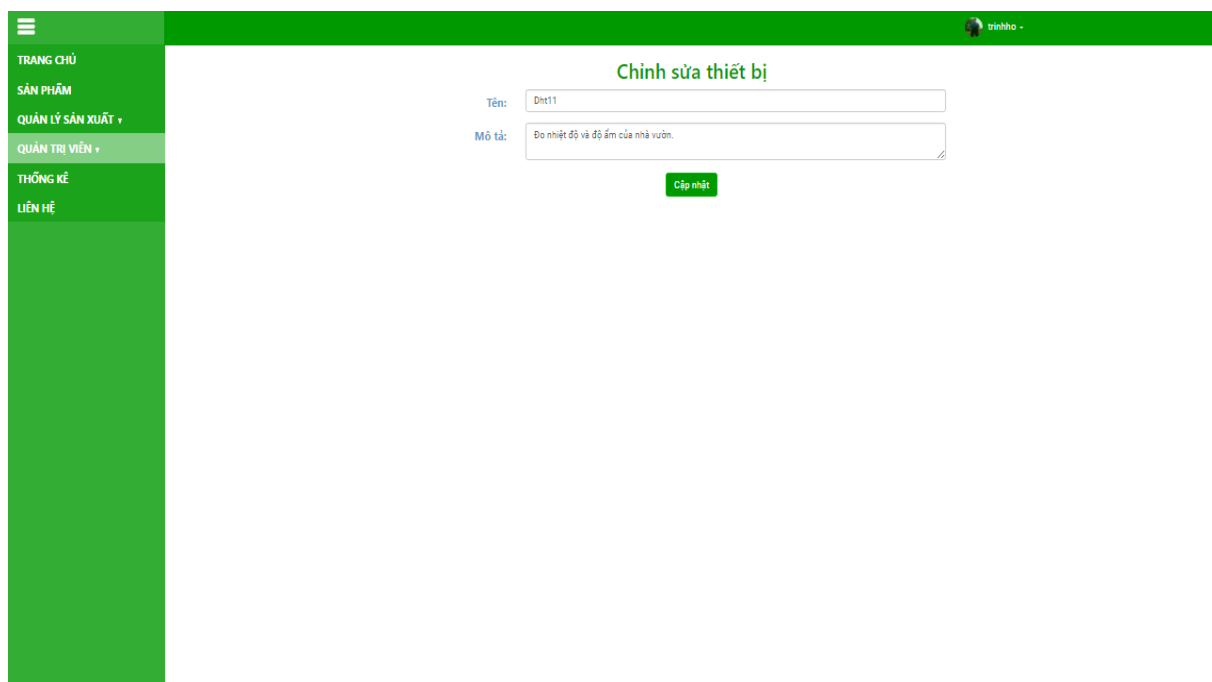


Hình 3.27 Màn hình thêm thiết bị sản xuất

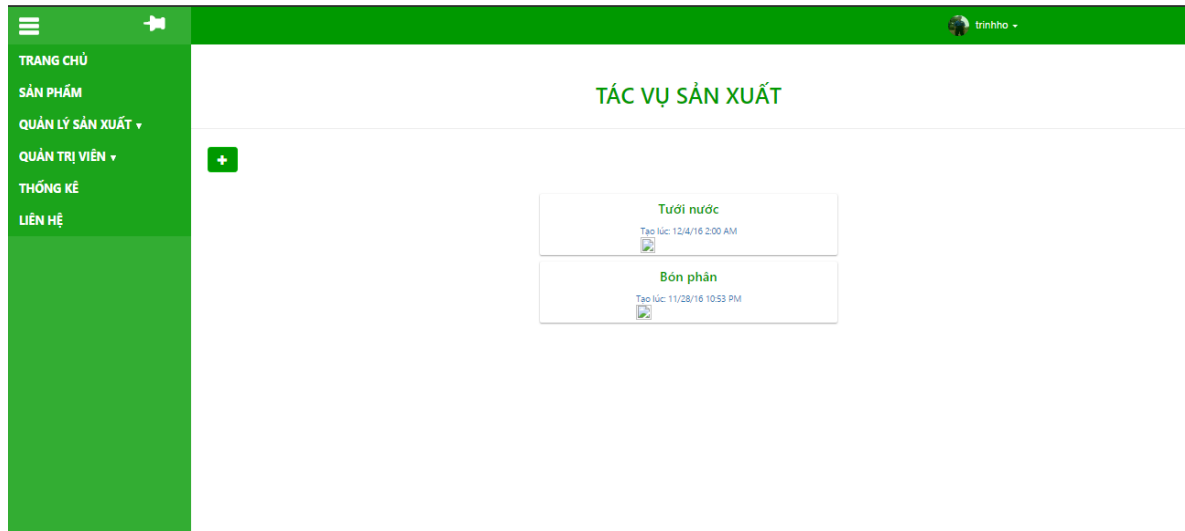




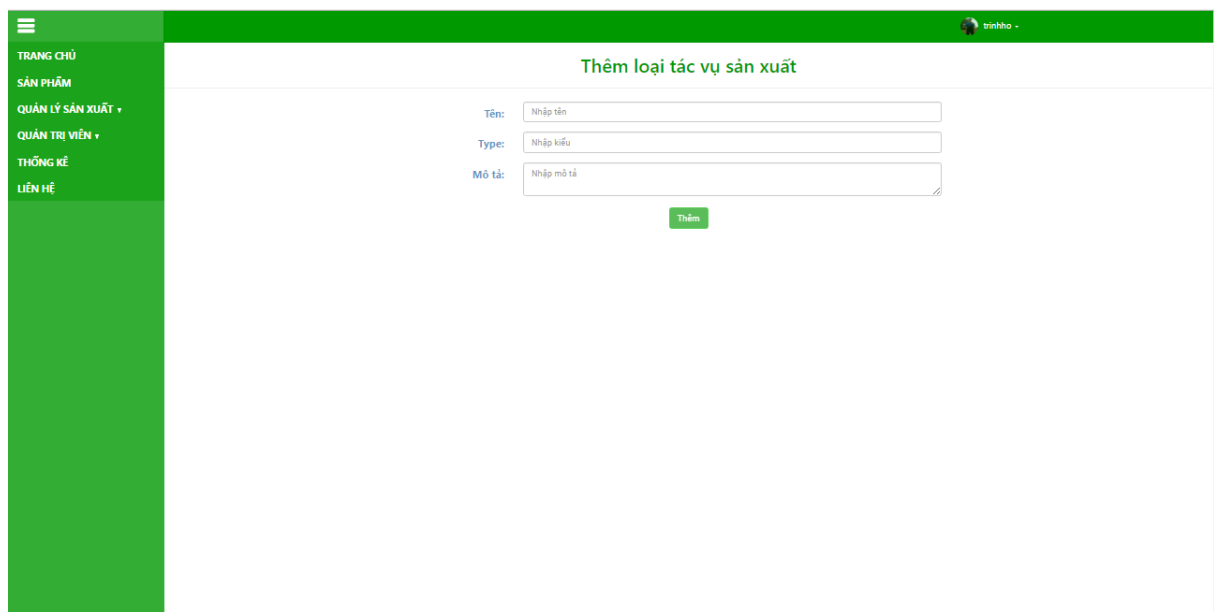
Hình 3.28 Màn hình chi tiết thiết bị sản xuất



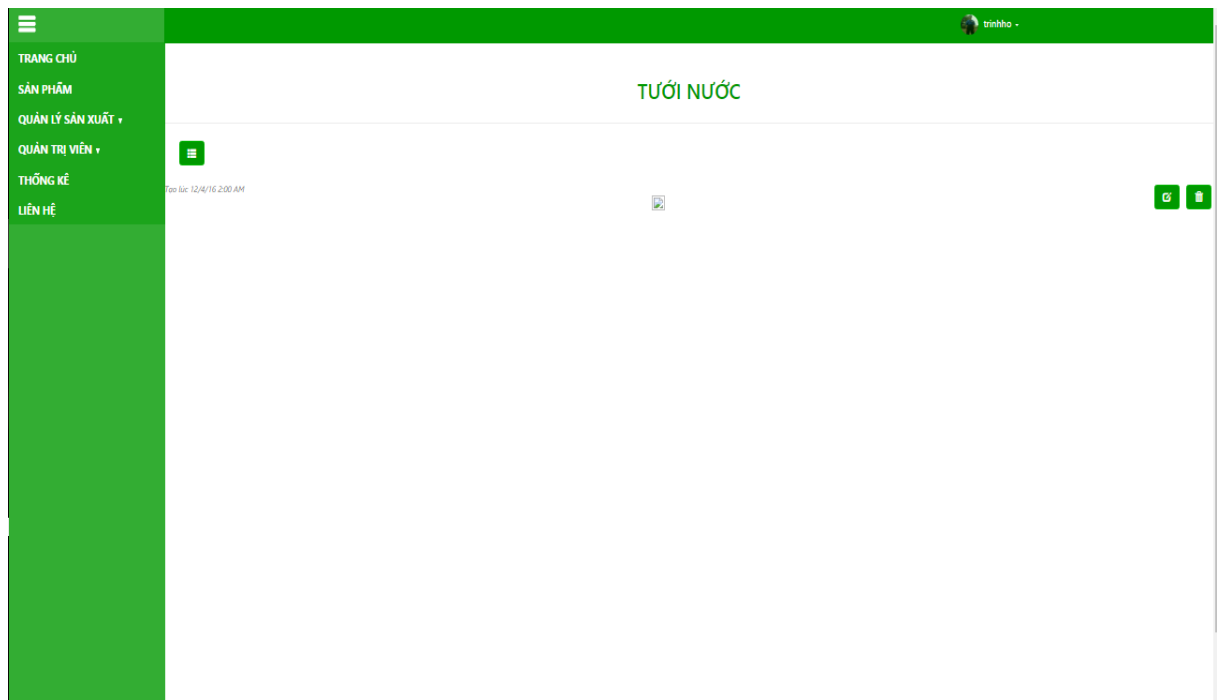
Hình 3.29 Màn hình chỉnh sửa thiết bị sản xuất

**- Quản lý danh sách loại tác vụ chăm sóc vườn rau:**

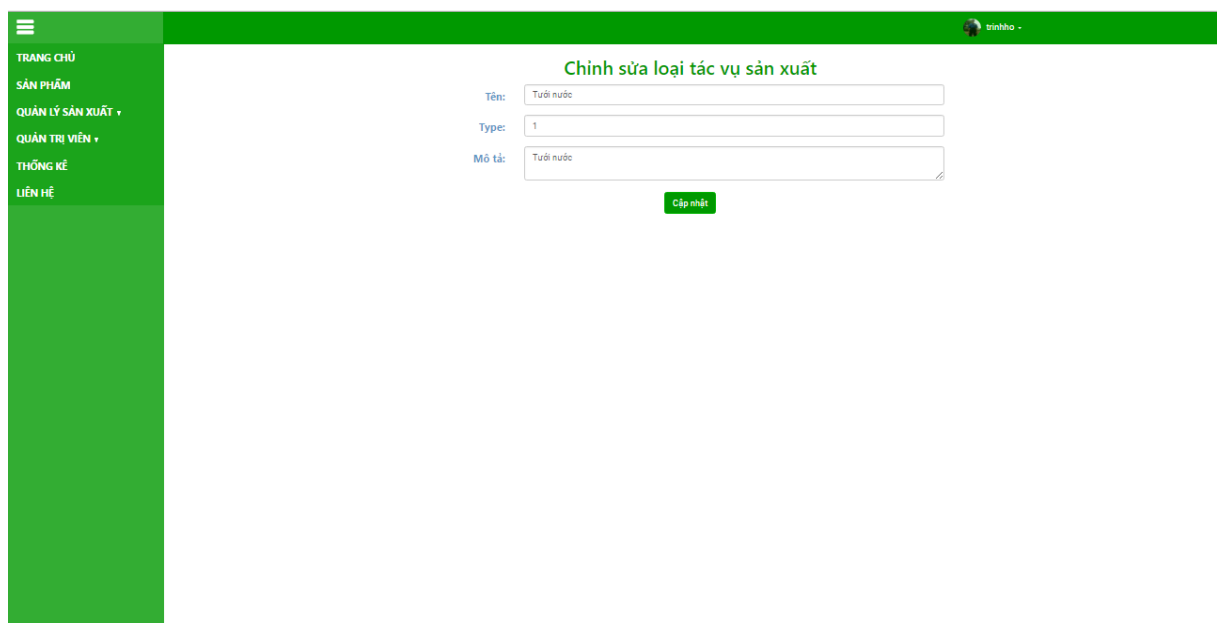
Hình 3.30 Màn hình hiển thị danh sách tác vụ sản xuất



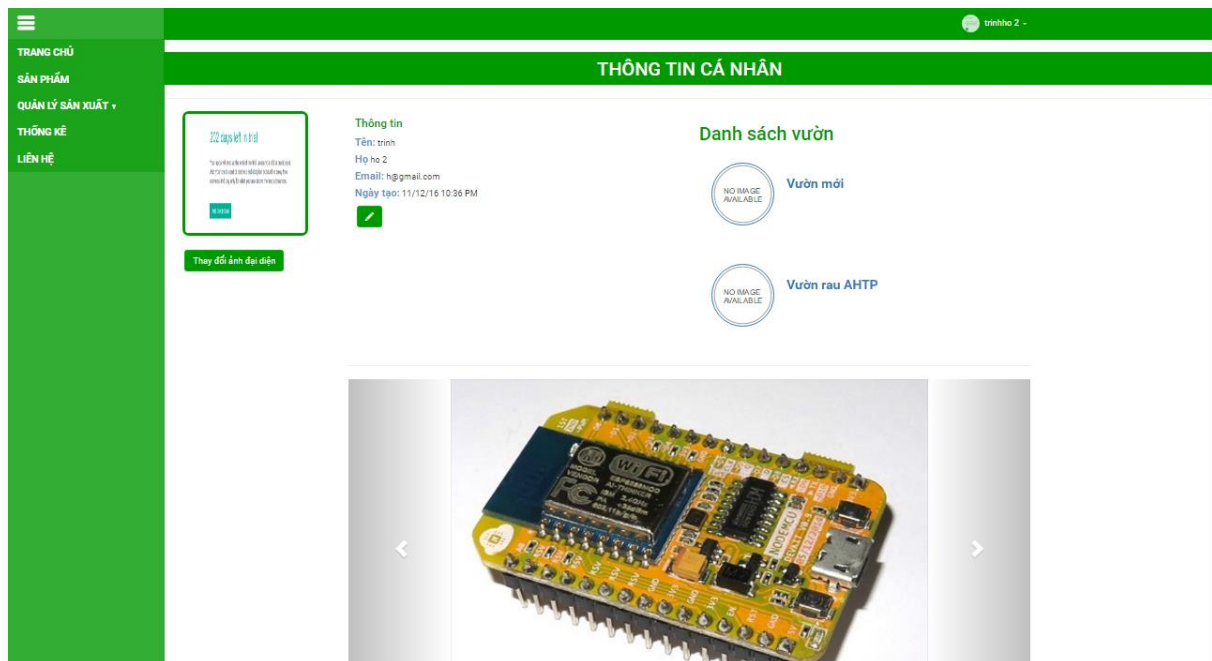
Hình 3.31 Màn hình thêm loại tác vụ sản xuất.



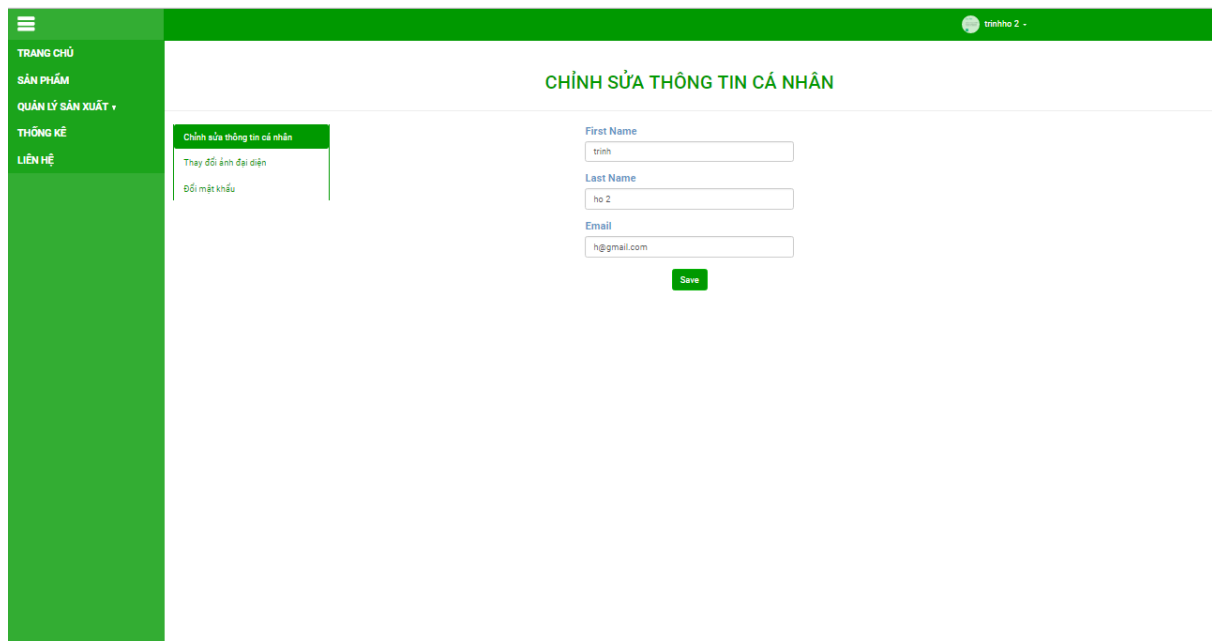
Hình 3.32 Màn chi tiết loại tác vụ sản xuất.



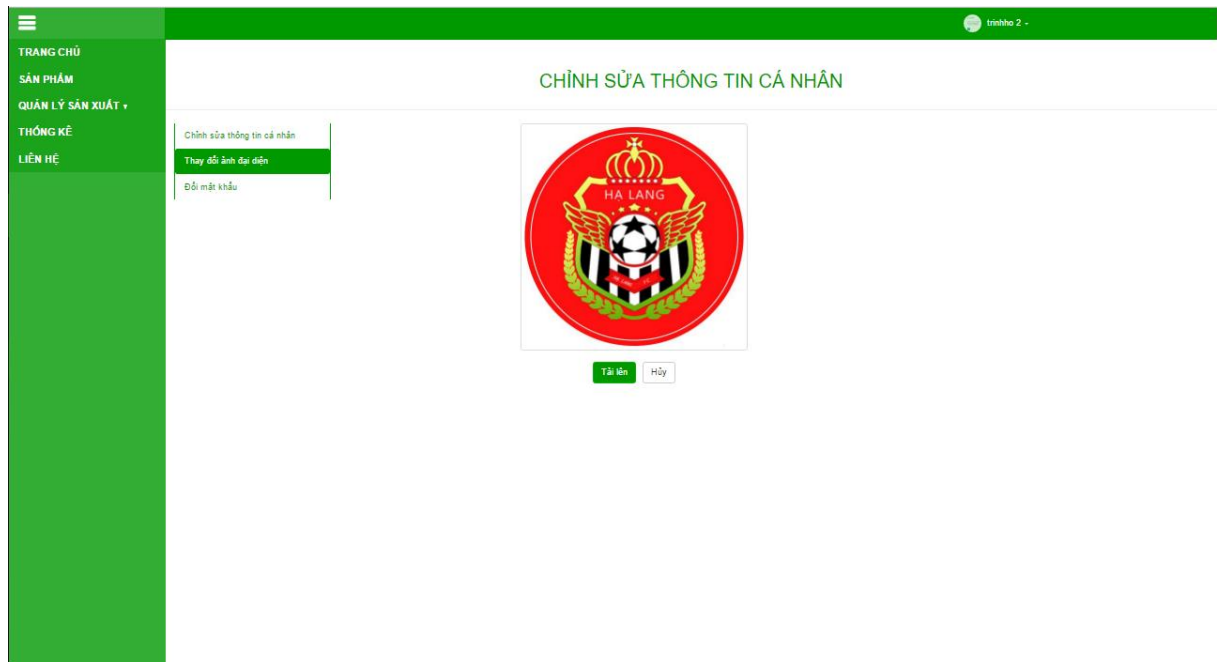
Hình 3.33 Màn hình chỉnh sửa loại tác vụ sản xuất.

**- Quản lý thông tin cá nhân:**

Hình 3.34 Màn hình hiển thị thông tin cá nhân.

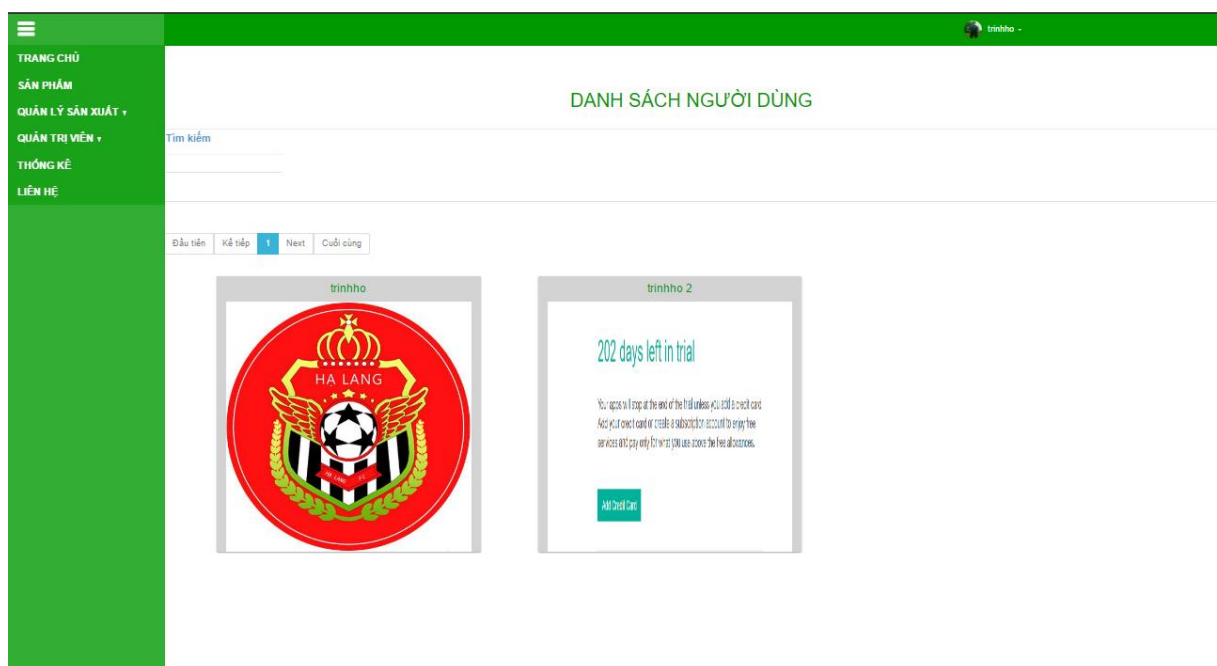


Hình 3.35 Màn hình chỉnh sửa thông tin cá nhân.

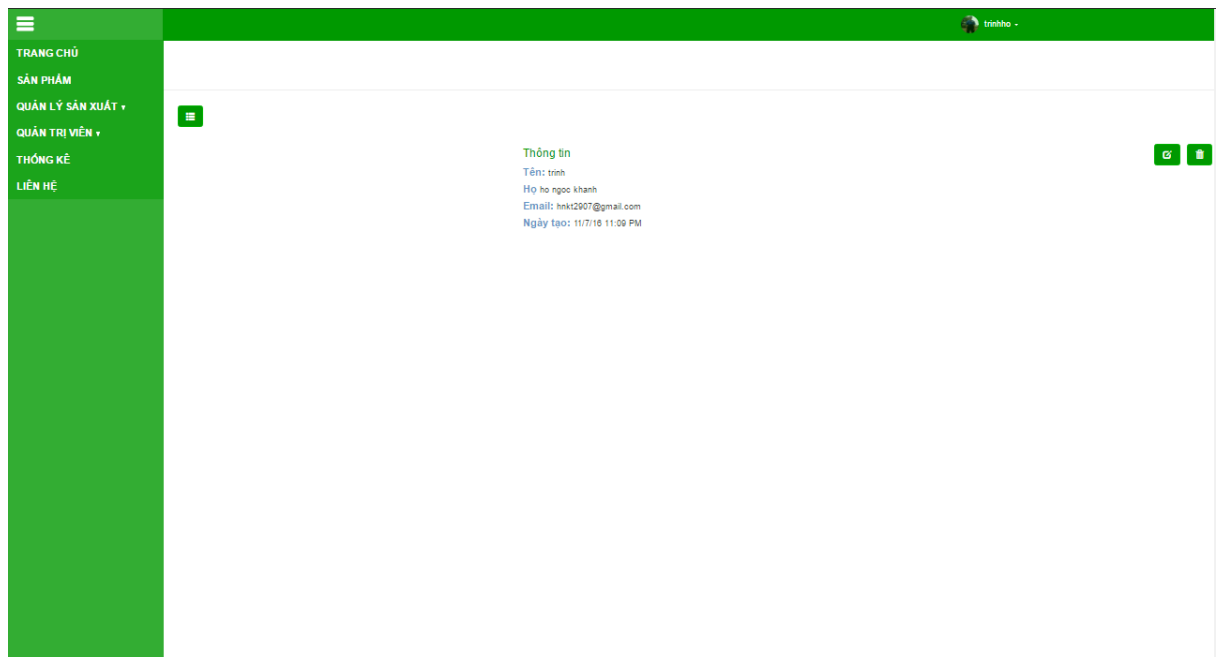


Hình 3.36 Màn hình thay đổi ảnh đại diện

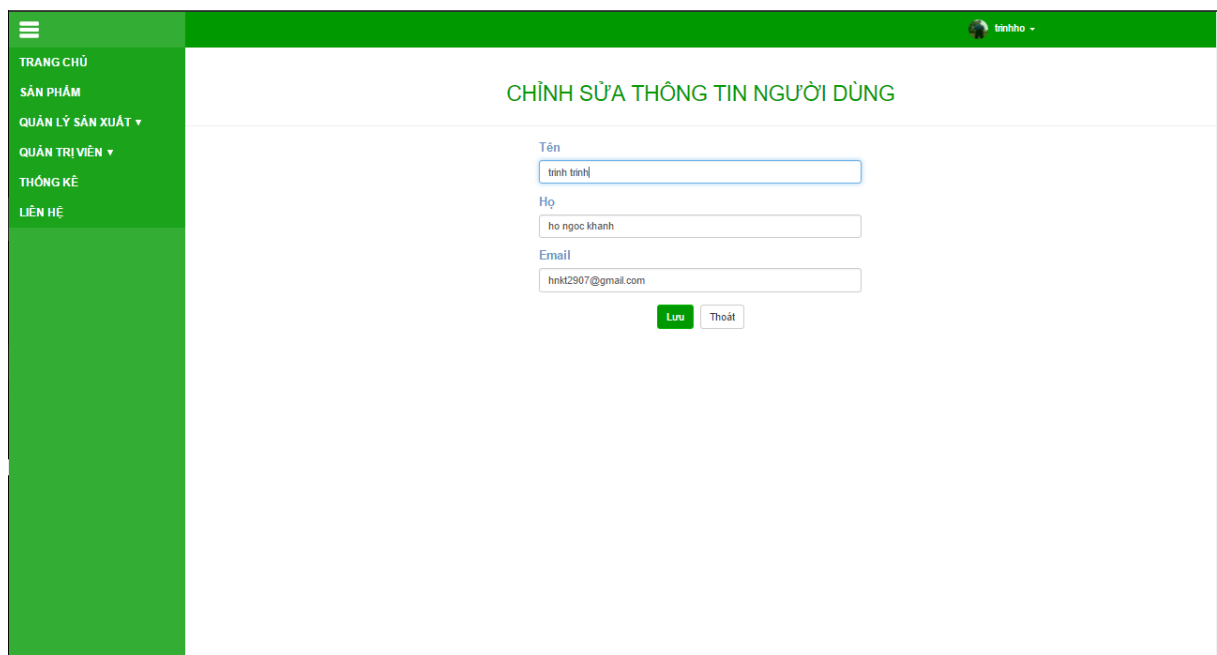
- Quản lý người dùng của quản trị viên:



Hình 3.37 Màn hình hiển thị danh sách người dùng.

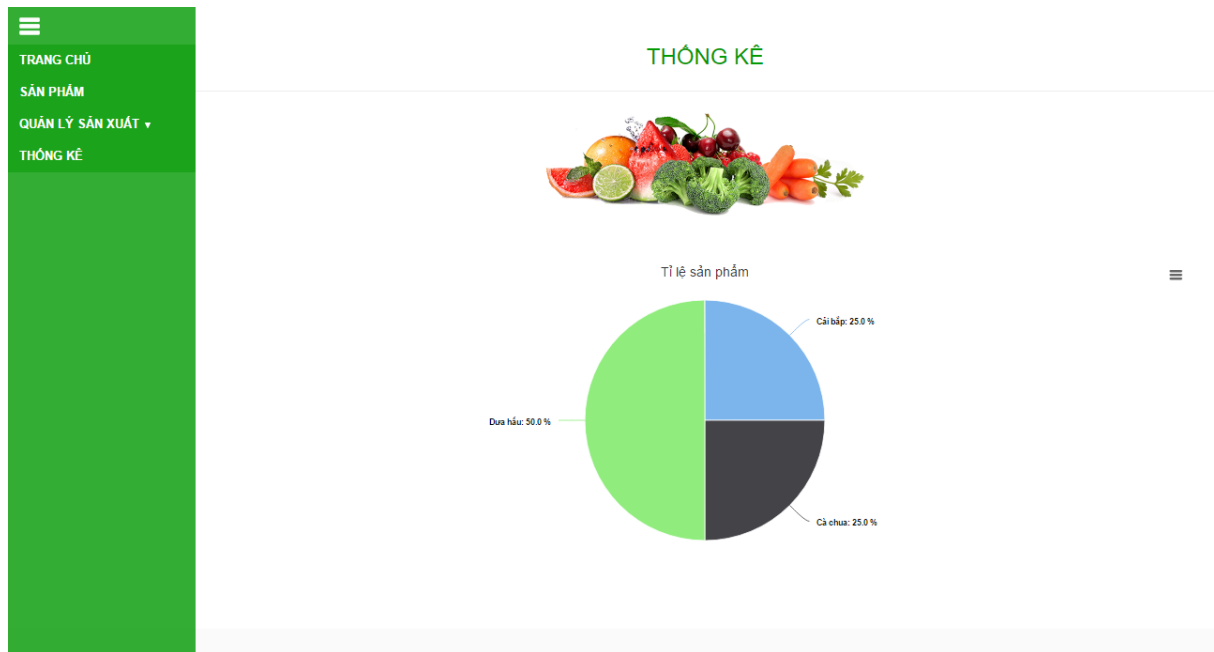


Hình 3.38 Màn hình hiển thị thông tin người dùng



Hình 3.39 Màn hình chỉnh sửa thông tin người dùng

## - Thống kê



Hình 3.40 Thống kê tỉ lệ sản phẩm

## KẾT LUẬN

### - Hạn chế:

- + Chưa khai thác hết được các ưu điểm của công nghệ sử dụng, giao diện người dùng còn khá đơn giản.
- + Nghiệp vụ quản lý canh tác vườn rau chưa đầy đủ.

### - Kết quả đạt được:

- + Áp dụng các công nghệ triển khai xây dựng được một hệ thống cho phép người dùng truy xuất thông tin của rau củ quả.
- + Người dùng là chủ vườn có thể đăng ký và quản lý thông tin vườn rau của mình, theo dõi biểu đồ nhiệt độ độ ẩm môi trường vườn của mình theo thời gian thực, quản lý tác vụ chăm sóc vườn rau của từng mùa vụ.
- + Người quản trị có thể duyệt đăng ký vườn, tạo và cập nhật dữ liệu đối tượng sản xuất, thiết bị sản xuất, danh sách loại tác vụ, quản lý người dùng.

### - Hướng phát triển:

- + Tối ưu hóa giao diện người dùng.
- + Khai thác triệt để sức mạnh của nền tảng điện toán đám mây và công nghệ IoT để có thể áp dụng vào thực tế.



## DANH MỤC TÀI LIỆU THAM KHẢO

Tham khảo các tài liệu trực tuyến:

Tiếng Anh:

- <https://nodejs.org/en/docs/>
- <https://docs.mongodb.com>
- <https://docs.angularjs.org/guide>
- <http://meanjs.org/docs.html>
- <http://mqtt.org/>