

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**

**Đề tài: Hệ thống đo lường nhiệt độ, độ ẩm không khí**

Lớp : 144947

Học phần : IoT và ứng dụng

Mã học phần : IT4735

Giảng viên hướng dẫn : TS.Nguyễn Đình Thuận

Danh sách thành viên nhóm:

Họ và tên	Mã số sinh viên
Dương Kim Nam	20204765
Nguyễn Văn Thắng	20204785
Dương Tiến Hoàng	20204832

*Hà Nội, tháng 1 năm 2024*

<b>LỜI NÓI ĐẦU</b>	<b>2</b>
<b>Chương 1: Giới thiệu đề tài</b>	<b>3</b>
1.1. Đặt vấn đề	3
<b>Chương 2: Phân tích yêu cầu bài toán</b>	<b>5</b>
2.1. Phân tích yêu cầu tổng quan	5
2.2. Sơ đồ tổng quan hệ thống	5
2.3. Phân tích yêu cầu chức năng	6
2.3.1. Các tác nhân chính	6
2.3.2. Biểu đồ use case tổng quan	6
2.3.3. Biểu đồ phân rã use case	7
2.3.4. Đặc tả chức năng	7
<b>Chương 3: Phân tích thiết kế hệ thống</b>	<b>16</b>
3.1. Biểu đồ trình tự tương tác hệ thống	16
3.2. Thiết kế chi tiết lớp	21
3.2.1. Sơ đồ gói	21
3.2.2. Sơ đồ tổng quan gói Model	22
3.2.3. Sơ đồ tổng quan gói Controller	24
3.2.4. Sơ đồ tổng quan gói View	25
3.3. Thiết kế cơ sở dữ liệu	29
3.3.1. Sơ đồ bảng quan hệ dữ liệu	29
3.3.2. Tổng quan bảng dữ liệu	29
3.3.3. Thiết kế chi tiết các bảng dữ liệu	29
<b>Chương 4: Công nghệ và công cụ sử dụng</b>	<b>31</b>
<b>Chương 5: Triển khai cài đặt</b>	<b>36</b>
<b>Chương trình và kết quả thử nghiệm</b>	<b>36</b>
5.1. Triển khai cài đặt	36
5.2. Chương trình minh họa	36

# LỜI NÓI ĐẦU

Trong những năm gần đây, Internet of Things (IoT) đã trở thành một trong những công nghệ quan trọng nhất của thế kỷ 21, đem lại những đổi mới đáng kể trong nhiều lĩnh vực khác nhau. Với việc tích hợp cảm biến và kết nối thông qua Internet, IoT cho phép thu thập và chia sẻ dữ liệu một cách tự động. Trong lĩnh vực công nghiệp, IoT giúp tối ưu hóa quy trình sản xuất và quản lý chuỗi cung ứng. Trong y tế, IoT đã mang lại những tiện ích tuyệt vời đối với việc theo dõi sức khỏe và chăm sóc bệnh nhân. Trong các đô thị thông minh, IoT đóng vai trò quan trọng trong việc giải quyết các vấn đề liên quan đến giao thông, an ninh và quản lý năng lượng.

Sau một thời gian học tập, nghiên cứu các công nghệ liên quan đến lĩnh vực IoT, và để hoàn thành học phần IoT và ứng dụng, nhóm chúng em lựa chọn đề tài “Hệ thống đo lường nhiệt độ, độ ẩm không khí” làm đề tài Bài tập lớn của học phần.

Mặc dù đã dành nhiều thời gian để cố gắng hoàn thiện sản phẩm, nhưng cũng không thể tránh khỏi những thiếu sót trong kiến thức và quá trình kiểm thử. Chúng em rất mong nhận được những đóng góp cũng như nhận xét thẳng thắn, chi tiết đến từ thầy để có thể tiếp tục hoàn thiện hơn nữa. Cuối cùng, chúng em xin được gửi lời cảm ơn đến TS. Nguyễn Đình Thuận đã hướng dẫn chúng em trong suốt quá trình hoàn thiện Bài tập lớn. Nhóm chúng em xin chân thành cảm ơn thầy.

# **Chương 1: Giới thiệu đề tài**

## **1.1. Đặt vấn đề**

Ngày nay, vấn đề ô nhiễm không khí trên toàn cầu ngày càng trở nên nghiêm trọng, là một thách thức lớn đối với toàn nhân loại. Theo báo cáo của Tổ chức Y tế Thế giới (WHO), ô nhiễm không khí ngoài trời là nguyên nhân gây tử vong sớm thứ tư trên thế giới, cướp đi sinh mạng của khoảng 7 triệu người mỗi năm. Các chất ô nhiễm trong không khí có thể gây ra nhiều vấn đề sức khỏe, bao gồm bệnh tim mạch, đột quỵ, ung thư phổi và các bệnh hô hấp khác. Ngoài ra, ô nhiễm không khí còn có thể ảnh hưởng đến hệ sinh thái, làm giảm năng suất nông nghiệp và gây ra các vấn đề về tầm nhìn.

Để giải quyết vấn đề này, cần có sự chung tay của cả cộng đồng. Hiện nay, các chính phủ đã ban hành các chính sách và biện pháp mạnh mẽ để giảm thiểu khí thải, giảm tác động của ô nhiễm không khí đến môi trường. Tuy nhiên những hành động như vậy là chưa đủ, vẫn cần nhiều sự quan tâm và động động quyết liệt hơn, xuất phát từ cả các cơ quan và các cá nhân, để nhanh chóng đẩy lùi vấn đề ô nhiễm không khí như hiện nay.

## **1.2. Mục tiêu, phạm vi đề tài**

Mặc dù hiện nay ở Việt Nam đã có những trang web cho phép đo lường chất lượng không khí, nhưng các trang web này vẫn còn nhiều hạn chế, chẳng hạn như chưa hỗ trợ việc thêm thiết bị của người dùng. Điều này khiến cho những người dùng muốn tìm hiểu chất lượng không khí tại một hoặc nhiều địa điểm cụ thể bằng phần cứng của bản thân gặp khó khăn.

Qua khảo sát và phân tích trên, bài tập lớn của chúng em hướng đến xây dựng hệ thống hỗ trợ đo, hiển thị dữ liệu không khí (cụ thể là nhiệt độ và độ ẩm) thông qua thiết bị phần cứng với các mục tiêu chính sau:

- Người dùng nhận được kết quả chất lượng không khí, nhiệt độ tại nơi đang truy cập vào website.

- Người dùng có thể thêm các thiết bị của bản thân để nhận dữ liệu đo được từ các thiết bị đó.
- Người dùng có thể xem chi tiết các thiết bị qua thời gian thực, dữ liệu mà thiết bị nhận được.
- Người dùng có thể xóa, sửa thiết bị.

## Chương 2: Phân tích yêu cầu bài toán

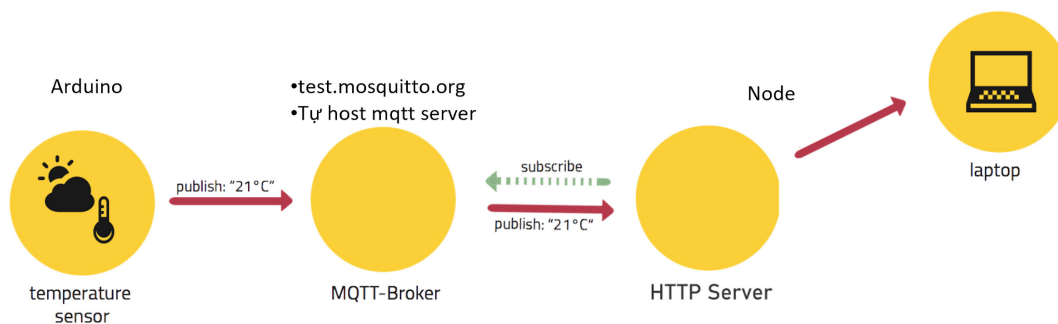
### 2.1. Phân tích yêu cầu tổng quan

Trước tiên, chúng ta sẽ đi tìm hiểu về các luồng sự kiện chính diễn ra trong hệ thống này.

Đối với người dùng, sau khi truy cập vào hệ thống:

- Hệ thống sẽ thông qua trình duyệt browser của người dùng xác định vị trí đứng hiện tại của người dùng từ đó hiển thị vị trí đó trên map.
- Hệ thống thông qua địa chỉ IP của người dùng để lấy được dữ liệu chất lượng không khí từ đó đưa ra các thông tin và cảnh báo.
- Người dùng muốn thêm thiết bị vào hệ thống để nhận được dữ liệu từ thiết bị đó.
- Sau khi thêm được thiết bị thì người dùng có thể xem chi tiết dữ liệu mà thiết bị đó nhận được và vị trí của thiết bị đó.

### 2.2. Sơ đồ tổng quan hệ thống



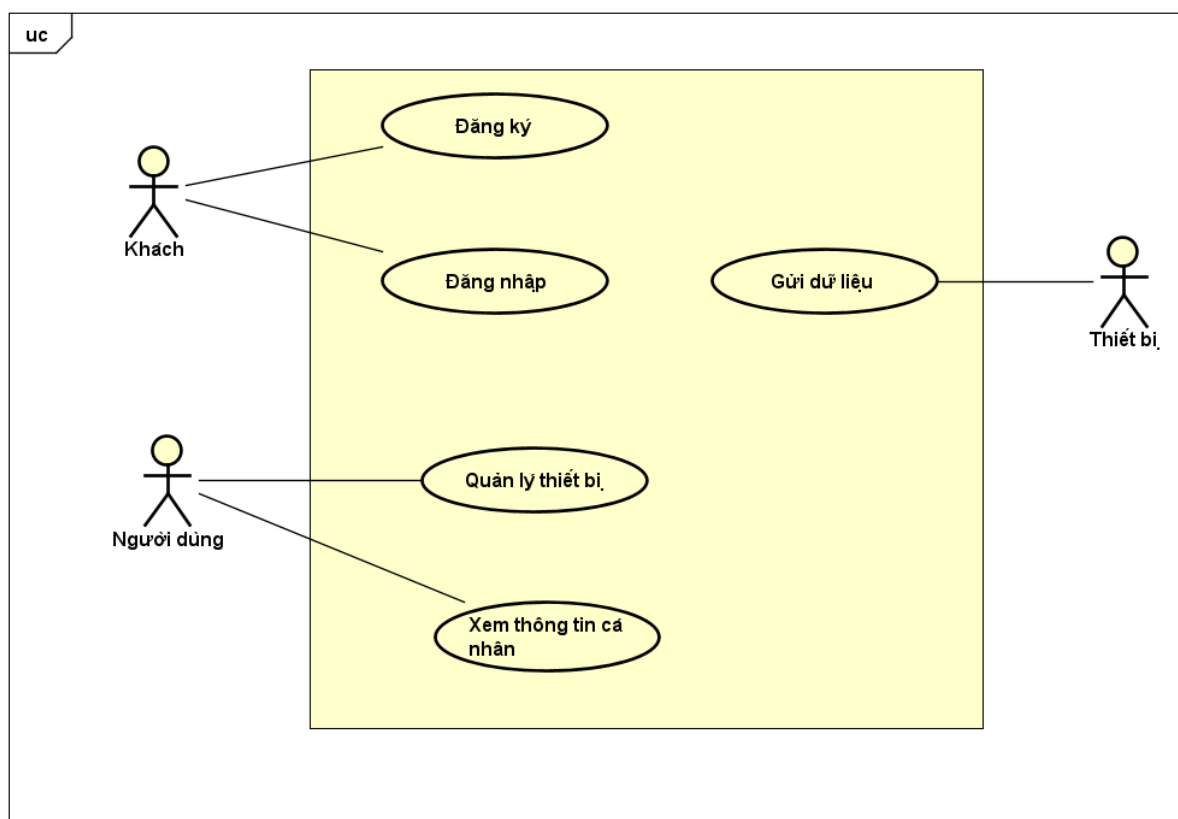
## 2.3. Phân tích yêu cầu chức năng

### 2.3.1. Các tác nhân chính

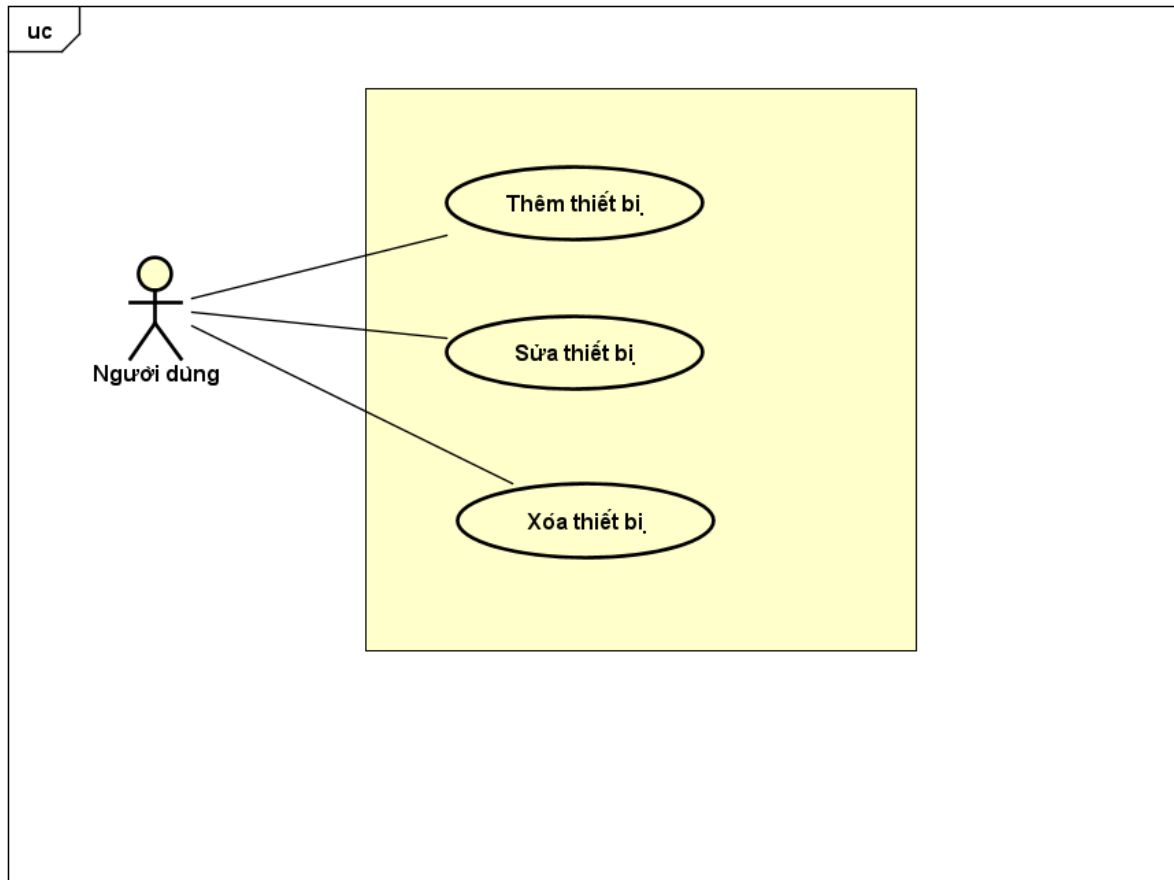
Hệ thống gồm 2 tác nhân chính:

1. Người dùng
2. Khách

### 2.3.2. Biểu đồ use case tổng quan



### 2.3.3. Biểu đồ phân rã use case



### 2.3.4. Đặc tả chức năng

#### 2.3.4.1. Nhóm tác nhân khách

Đặc tả use case UC-01: Đăng nhập

<b>Mã Use case</b>	UC-01
<b>Tên Use case</b>	Đăng nhập
<b>Mục đích Use case</b>	Đăng nhập vào hệ thống.
<b>Tác nhân</b>	Khách
<b>Sự kiện kích hoạt</b>	Không có



<b>Tiền điều kiện</b>	Đã có tài khoản đăng nhập từ trước.		
<b>Hậu điều kiện</b>	Đăng nhập thành công, quay về giao diện chính theo vai trò của tác nhân.		
<b>Luồng sự kiện chính</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Khách	Điền thông tin đăng nhập (email, mật khẩu)
	2.	Khách	Nhấn nút Đăng nhập
	3.	Hệ thống	Kiểm tra kiểu dữ liệu đầu vào có hợp lệ hay không
	4.	Hệ thống	Truy xuất cơ sở dữ liệu kiểm tra tài khoản mật khẩu.
	5.	Hệ thống	Truy xuất cơ sở dữ liệu kiểm tra tài khoản mật khẩu.
	6.	Hệ thống	Đăng nhập thành công chuyển đến giao diện chính
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	4.a.	Hệ thống	Kiểu dữ liệu đầu vào sai (email thiếu đuôi, mật khẩu không đủ 8 ký tự,...). Hiển thị thông báo tại vị trí sai.
	5.b.	Hệ thống	Không tìm thấy tài khoản hoặc mật khẩu sai. Hiển thị đăng nhập không thành công.

Đặc tả use case UC-02: Đăng ký

<b>Mã Use case</b>	UC-02
<b>Tên Use case</b>	Đăng ký
<b>Mục đích Use case</b>	Khách đăng ký tài khoản sử dụng hệ thống.
<b>Tác nhân</b>	Khách

<b>Sự kiện kích hoạt</b>	Không có		
<b>Tiền điều kiện</b>	Khách chưa có tài khoản và muốn đăng ký mới.		
<b>Hậu điều kiện</b>	Tài khoản đăng ký thành công.		
<b>Luồng sự kiện chính</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Khách	Chọn “Tạo tài khoản mới” ở giao diện đăng nhập.
	2.	Hệ thống	Hiển thị biểu mẫu đăng ký tài khoản
	3.	Người dùng	Điền thông tin đăng ký (username, email, password)
	4.	Hệ thống	Kiểm tra tài khoản, mật khẩu có đúng kiểu dữ liệu hoặc đã tồn tại tài khoản hay chưa.
	5.	Hệ thống	Kiểm tra các trường bắt buộc đã điền đủ chưa, kiểm tra kiểu dữ liệu đã đúng chưa
	6.	Hệ thống	Lưu tài khoản vào cơ sở dữ liệu.
	7.	Hệ thống	Thông báo đăng ký thành công, chuyển về giao diện đăng nhập.
<b>Luồng sự kiện thay thế</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	4.a.	Hệ thống	Nếu sai kiểu dữ liệu hoặc đã tồn tại thì hiển thị tài khoản không hợp lệ, yêu cầu điền lại.

	5.b.	Hệ thống	Nếu chưa điền đầy đủ trường thông tin bắt buộc hoặc sai kiểu dữ liệu, yêu cầu điền lại.
--	------	----------	---

Đặc tả use case UC-03: Đăng xuất

<b>Mã Use case</b>	UC-03		
<b>Tên Use case</b>	Đăng xuất		
<b>Mục đích Use case</b>	Người dùng đăng xuất khỏi hệ thống		
<b>Tác nhân</b>	Người dùng		
<b>Sự kiện kích hoạt</b>	Người dùng chọn nút “Log out” trong giao diện chính.		
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập.		
<b>Hậu điều kiện</b>	Người dùng đăng xuất khỏi hệ thống.		
<b>Luồng sự kiện chính</b>	<b>ST</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Chọn nút “Log out” trong giao diện chính
	2.	Hệ thống	Thoát tài khoản của người dùng

#### 2.3.4.2. Nhóm tác nhân người dùng

Đặc tả use case UC-04: Thêm thiết bị

<b>Mã Use case</b>	UC-04		
<b>Tên Use case</b>	Thêm thiết bị		
<b>Mục đích Use case</b>	Giúp người dùng thêm 1 thiết bị mới		
<b>Tác nhân</b>	Người dùng		
<b>Sự kiện kích hoạt</b>	Quản trị viên nhấn nút “Add Device” trong giao diện quản lý thiết bị		
<b>Tiền điều kiện</b>	Không có		
<b>Hậu điều kiện</b>	Hệ thống thông báo thêm môn thiết bị mới thành công		
<b>Luồng sự kiện chính</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Nhấn nút “Add Device”.
	2.	Hệ thống	Hiển thị biểu mẫu thêm thiết bị mới.
	3.	Người dùng	Điền thông tin thiết bị mới
	4.	Người quản lý	Nhấn nút “Add”.
	5.	Hệ thống	Kiểm tra tính hợp lệ của các trường thông tin đã điền
	6.	Hệ thống	Cập nhật cơ sở dữ liệu.

	7.	Hệ thống	Thông báo thêm thiết bị thành công.
<b>Luồng sự kiện thay thế</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	6.a.	Hệ thống	Thông báo thông tin thiết bị không hợp lệ
	7.b.	Hệ thống	Yêu cầu nhập lại biểu mẫu thêm thiết bị.

Đặc tả use case UC-05: Xóa thiết bị

<b>Mã Use case</b>	UC-05		
<b>Tên Use case</b>	Xóa thiết bị		
<b>Mục đích Use case</b>	Quản trị viên xóa các thiết bị trong danh sách các thiết bị		
<b>Tác nhân</b>	Người dùng		
<b>Sự kiện kích hoạt</b>	Quản trị viên chọn “Xóa thiết bị” ở giao diện quản lý thiết bị		
<b>Tiền điều kiện</b>	Thiết bị đã tồn tại		
<b>Hậu điều kiện</b>	Xóa thành công thiết bị		
<b>Luồng sự kiện chính</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>

	1.	Người dùng	Chọn biểu tượng “Xóa thiết bị” trong danh sách thiết bị
	2.	Hệ thống	Cập nhật lại cơ sở dữ liệu
	3.	Hệ thống	Thông báo xóa thiết bị thành công
<b>Luồng sự kiện thay thế</b>	Không có		

Đặc tả use case UC-06: Sửa thông tin thiết bị

<b>Mã Use case</b>	UC-06		
<b>Tên Use case</b>	Sửa thông tin thiết bị		
<b>Mục đích Use case</b>	Giúp người dùng sửa lại thông tin thiết bị		
<b>Tác nhân</b>	Người dùng		
<b>Sự kiện kích hoạt</b>	Không có		
<b>Tiền điều kiện</b>	Thiết bị đã tồn tại		
<b>Hậu điều kiện</b>	Sửa thành công thông tin của thiết bị		
<b>Luồng sự kiện chính</b>	<b>ST</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Nhấn nút “Sửa thông tin thiết bị”

	2.	Hệ thống	Hiển thị biểu mẫu sửa thông tin thiết bị
	3	Người dùng	Điền thông tin cần sửa của thiết bị
	4.	Người dùng	Nhấn nút “Save”
	5.	Hệ thống	Lưu thông tin vào cơ sở dữ liệu
	6.	Hệ thống	Thông báo sửa thiết bị thành công
<b>Luồng sự kiện thay thế</b>	Không có		

#### 2.3.4.1. Nhóm tác nhân thiết bị

Đặc tả use case UC-07: “Gửi dữ liệu”

<b>Mã Use case</b>	UC-07
<b>Tên Use case</b>	Gửi dữ liệu
<b>Mục đích Use case</b>	Thiết bị gửi dữ liệu thu thập được cho hệ thống
<b>Tác nhân</b>	Thiết bị
<b>Sự kiện kích hoạt</b>	Không có
<b>Tiền điều kiện</b>	Không có
<b>Hậu điều kiện</b>	Thiết bị gửi dữ liệu lên server

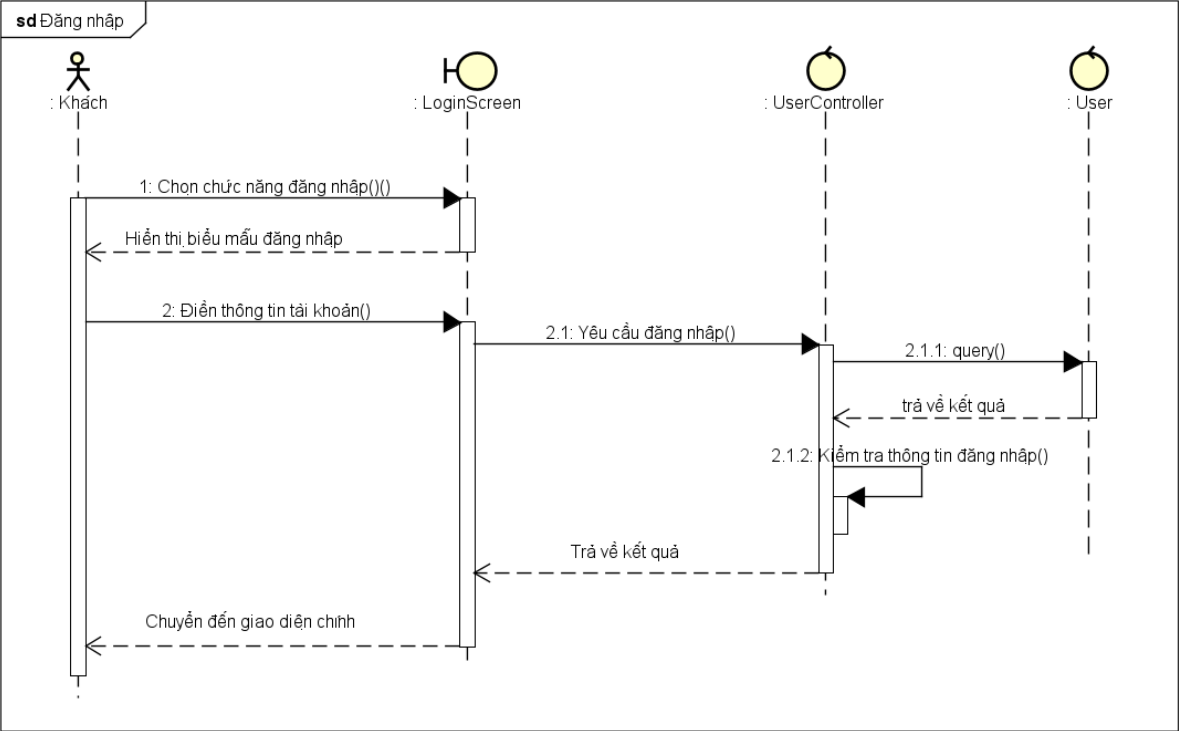
<b>Luồng sự kiện chính</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Thiết bị	Gửi dữ liệu thông qua giao thức MQTT
	2.	Hệ thống	Kiểm tra thiết bị đã tồn tại hay chưa
	3	Hệ thống	Thêm dữ liệu thiết bị gửi vào database
<b>Luồng sự kiện thay thế</b>	<b>ST T</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	3.a.	Hệ thống	Xóa dữ liệu mà thiết bị đã gửi đi



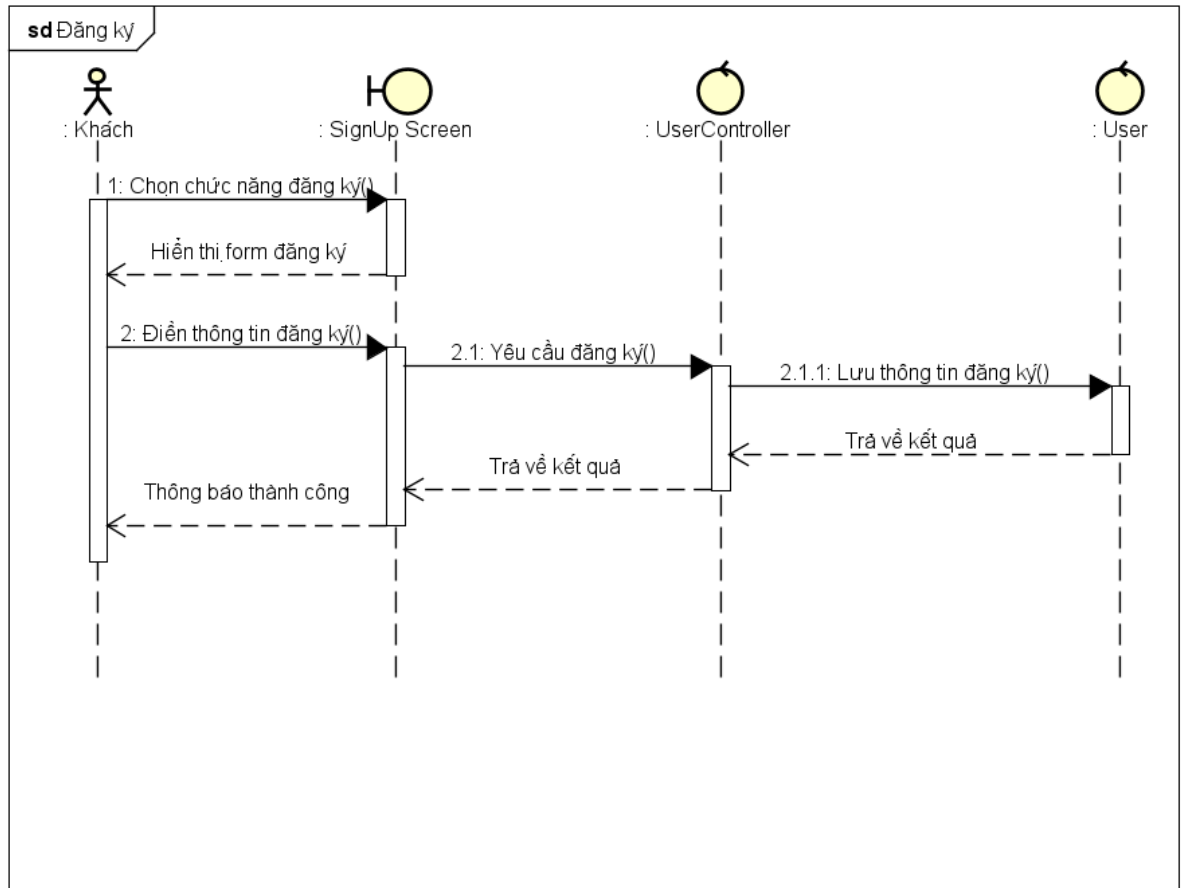
# Chương 3: Phân tích thiết kế hệ thống

## 3.1. Biểu đồ trình tự tương tác hệ thống

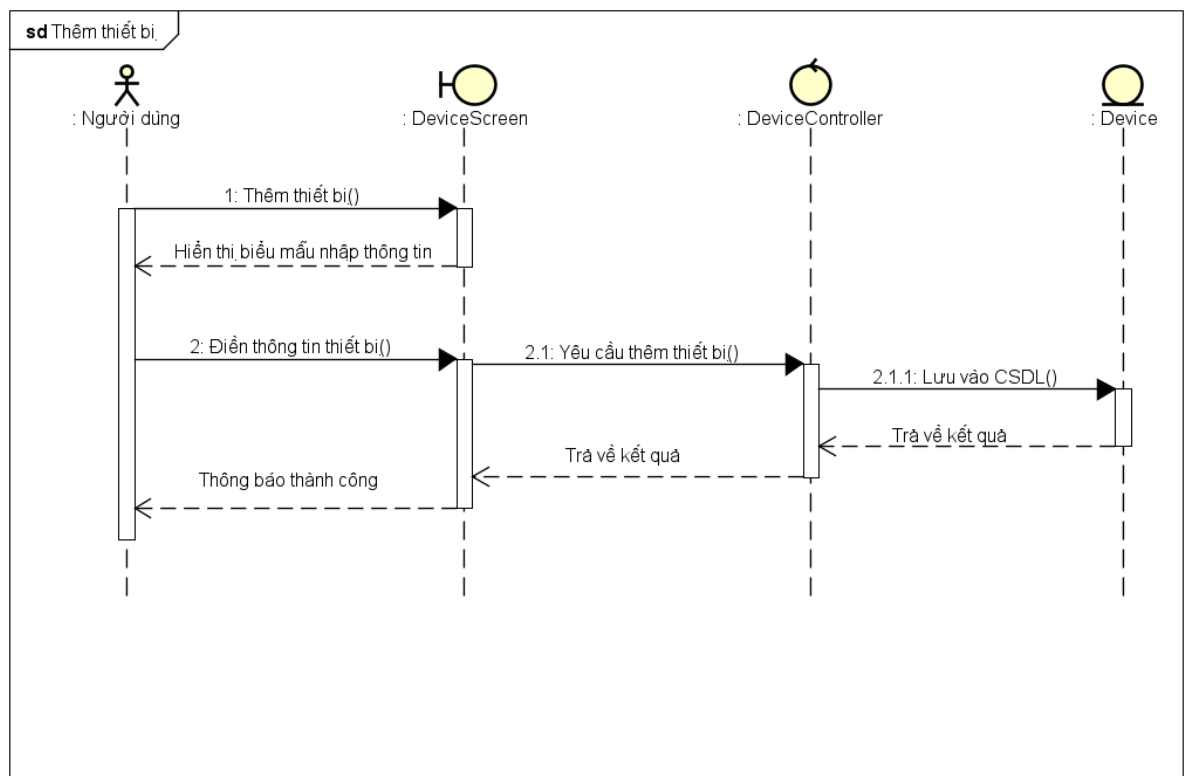
- Đăng nhập



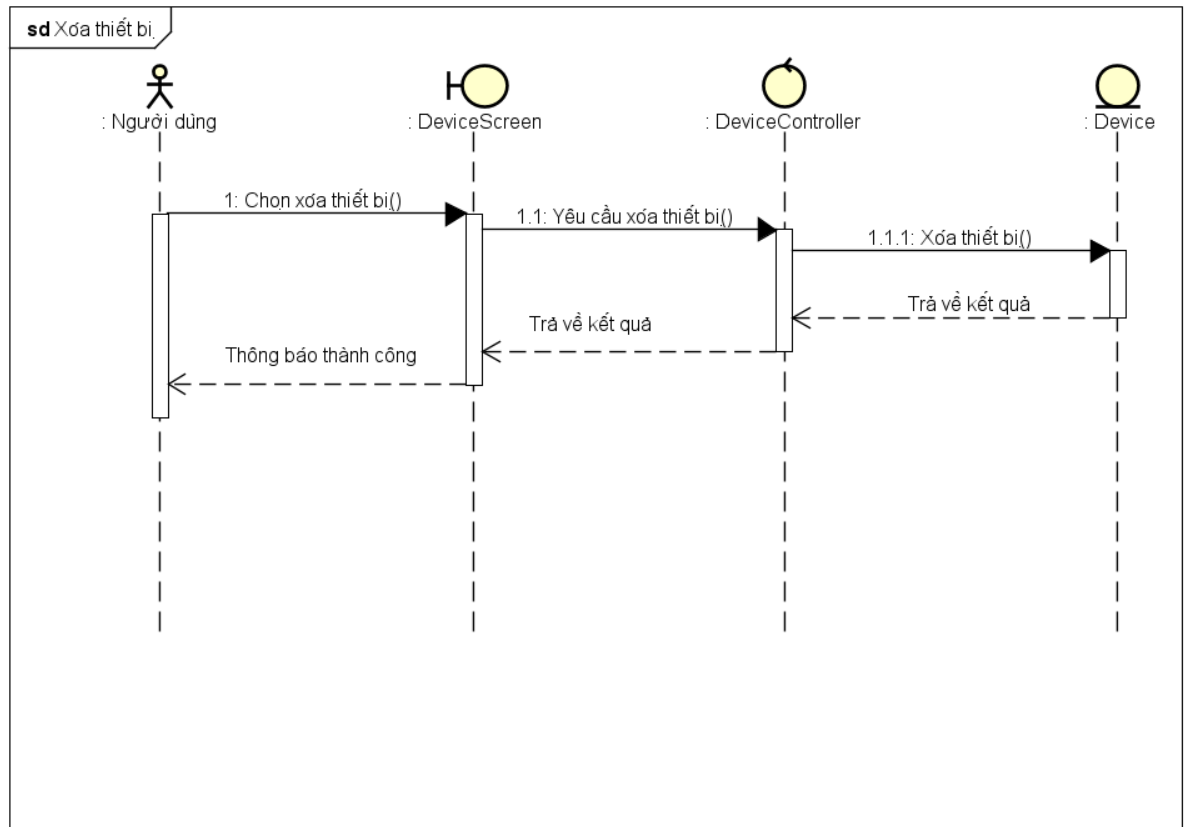
- Đăng ký



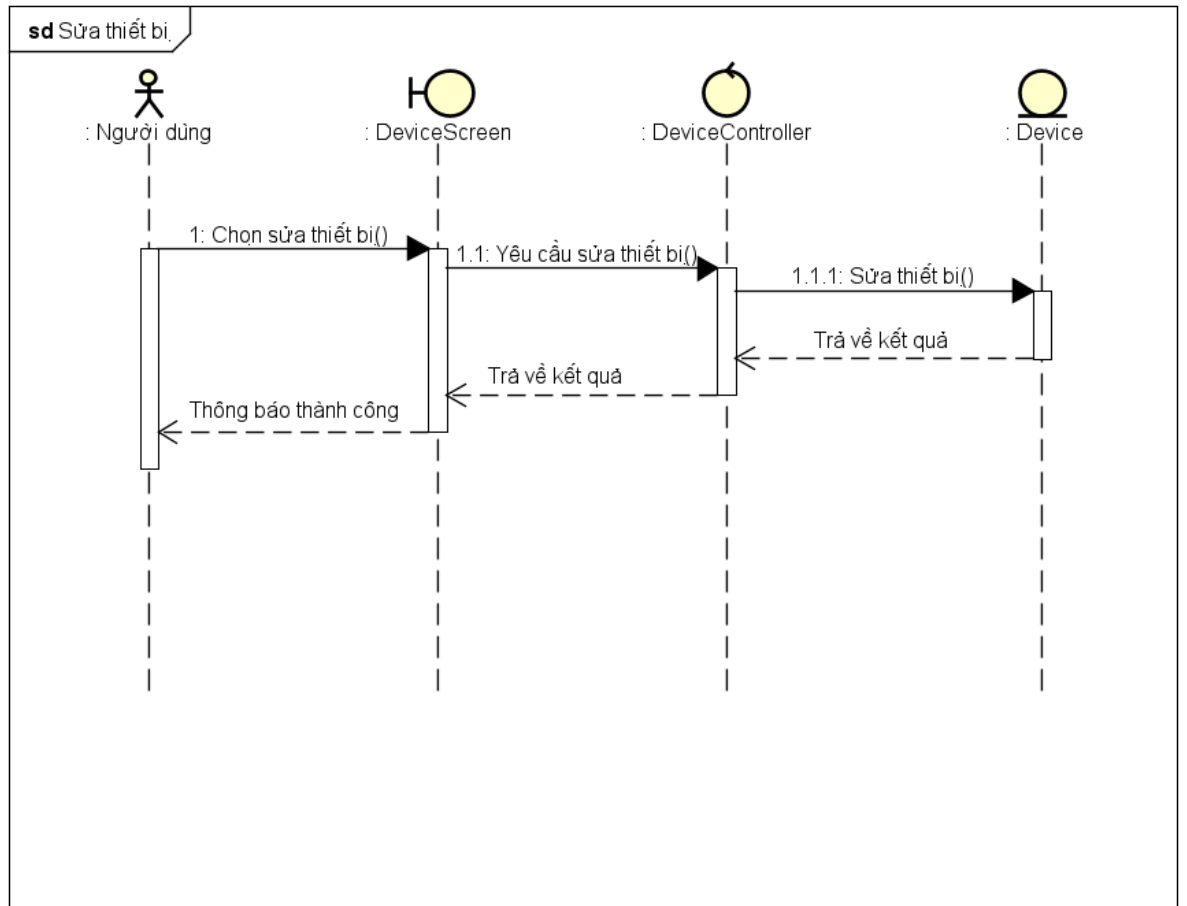
- 
- Thêm thiết bị



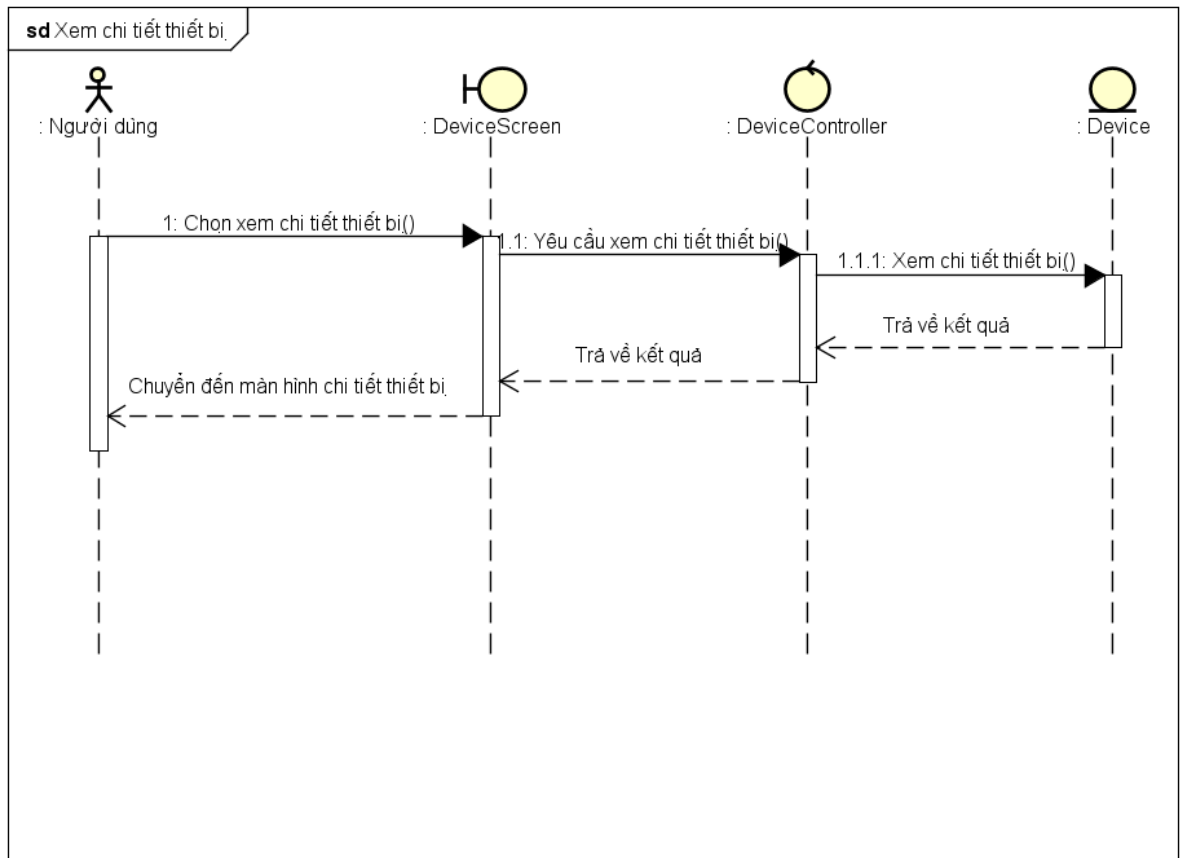
- Xóa thiết bị



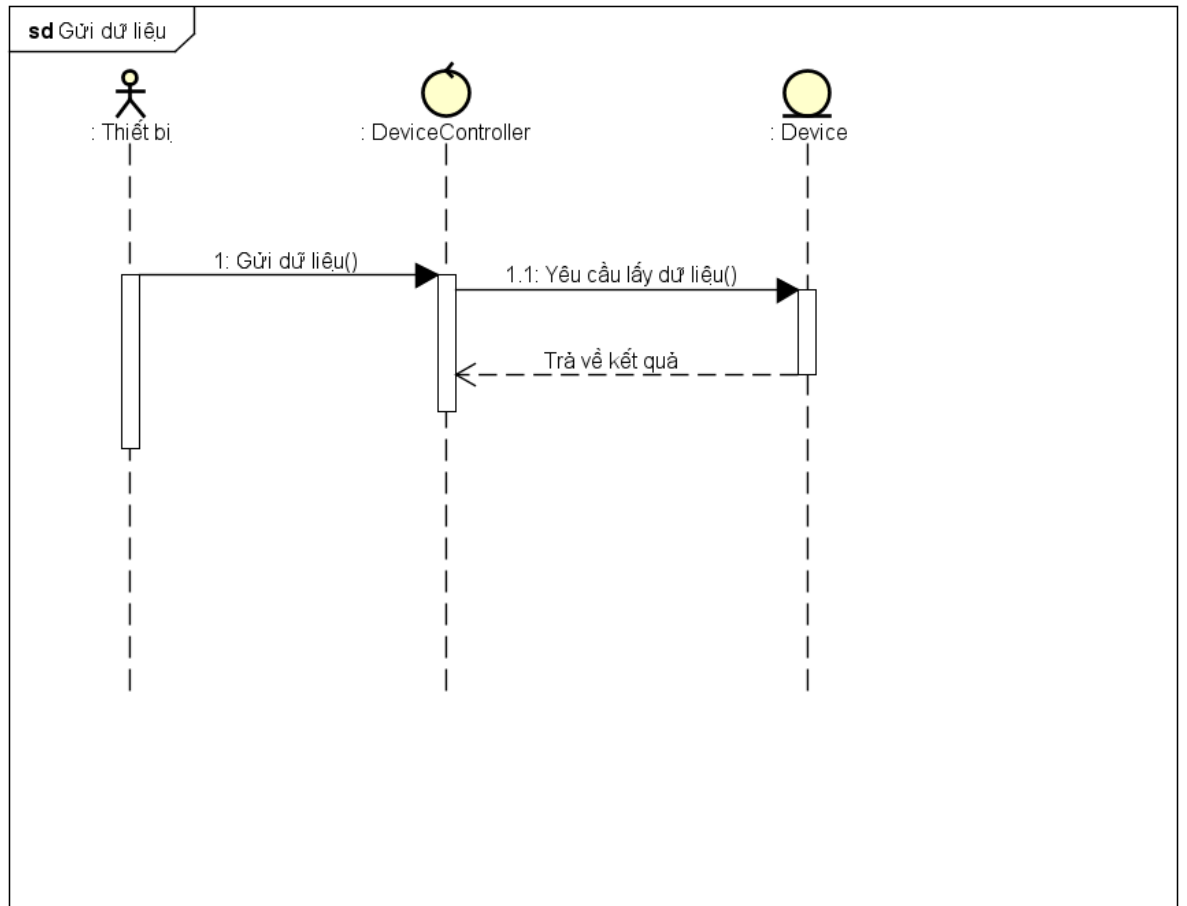
- Sửa thiết bị



- Xem chi tiết thiết bị

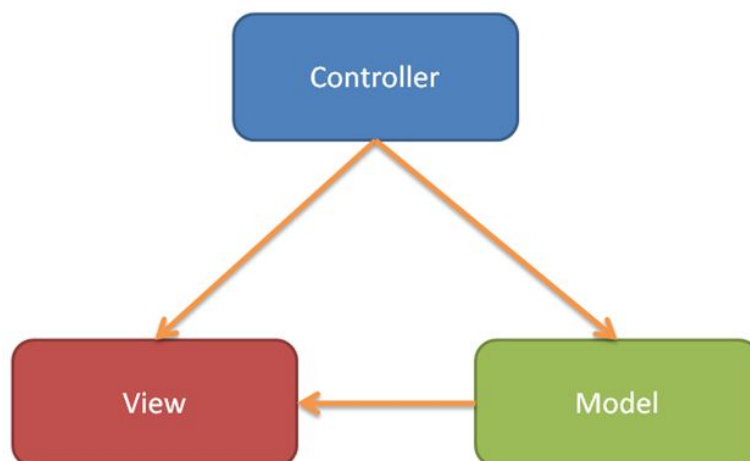


- Gửi dữ liệu

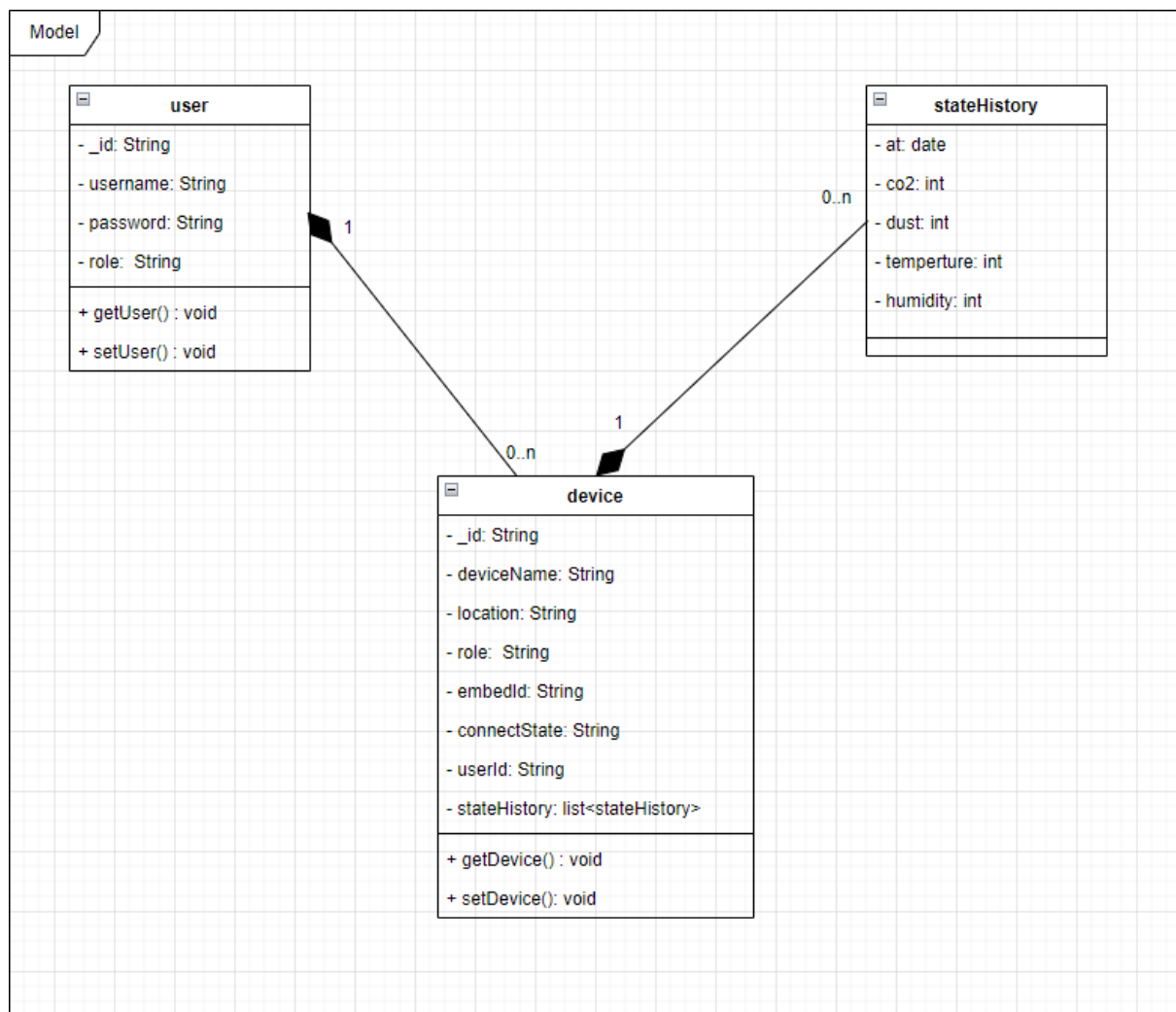


## 3.2. Thiết kế chi tiết lớp

### 3.2.1. Sơ đồ gói



### 3.2.2. Sơ đồ tổng quan gói Model



#### 3.2.2.1. Lớp device

Mô tả: lớp khai báo các thuộc tính thiết bị

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
<code>_id</code>	String	Private	Mã thiết bị lưu trên database
<code>embedId</code>	String	Private	Mã thiết bị phân cứng
<code>deviceName</code>	String	Private	Tên thiết bị
<code>connectState</code>	String	Private	Trạng thái hoạt động
<code>userId</code>	String	Private	Mã người tạo

stateHistory	List<stateHistory>	Private	Dữ liệu thiết bị
isPublic	Boolean	Private	Trạng thái công cộng thiết bị

### 3.2.2.2. Lớp user

Mô tả: lớp khai báo thuộc tính của người dùng

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
_id	String	Private	Mã người dùng
username	String	Private	Tên người dùng
password	String	Private	Mật khẩu người dùng
role	String	Private	Phân loại người dùng

### 3.2.2.3. Lớp stateHistory

Mô tả: lớp mô tả dữ liệu thiết bị nhận được

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
at	Date	Private	Thời điểm dữ liệu nhận data
Temperature	Int	Private	Nhiệt độ
Co2	Int	Private	Nồng độ CO2
Humidity	Int	Private	Độ ẩm
dust	Int	Private	Độ bụi



### 3.2.3. Sơ đồ tổng quan gói Controller

#### 3.2.3.1. Lớp deviceController

Mô tả: lớp mô tả deviceController

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
Device	device	Private	Thực thể device

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
validateForVal idId	String	Void	Public	Kiểm tra hợp lệ id của thiết bị
validatorForDe viceExists	String	Void	Public	Kiểm tra tra thiết bị đã tồn tại chưa
getAllDevice	Void	List<device >	Public	Trả danh sách thiết bị
getDeviceByD eviceId	String	device	Public	Lấy thiết bị theo id của thiết bị trên database
getDeviceByE mbedId	String	Device	Public	Lấy thiết bị theo id phần cứng của thiết bị
createDeviceB yUserId	String, device	Device	Public	Tạo thiết bị mới
updateDevice ByDeviceId	String, device	Device	Public	Cập nhật thiết bị

updateStateHistoryByDeviceId	String, device	Device	Public	Cập nhật trạng thái thiết bị
deleteDeviceById	String	void	Public	Xóa thiết bị

### 3.2.3.2. Lớp userController

Mô tả: lớp mô tả userController

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
User	User	Private	Thực thể user

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
getAllUser	Void	List	Public	Lấy tất cả tài khoản
signUp	User	Void	Public	Đăng nhập
Logout	User	Void	Public	Đăng xuất
getUserAndDevices	String	Void	Public	Lấy toàn bộ thiết bị dự trên userId
getCurrentUser	Void	Void	Public	Lấy thông tin người dùng đăng nhập hiện tại
updateUser	void	void	Public	Cập nhật thông tin người dùng

### 3.2.4. Sơ đồ tổng quan gói View

### 3.2.4.1. Lớp Trang chủ

Mô tả: lớp mô tả trang chủ

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
getLocation	void	void	Public	Lấy vị trí hiện tại của người dùng
getAirQunlity	void	void	Public	Lấy thông tin chất lượng không khí hiện tại của người dùng

### 3.2.4.2. Lớp Thông tin cá nhân

Mô tả: lớp mô tả thông tin chi tiết người dùng đang đăng nhập

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
Username	String	Private	Tên người dùng
Email	String	Private	Email người dùng

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
getUserByUserId	Void	Void	Public	Lấy thông tin người dùng dựa trên id người dùng

updateUser	Void	Void	Public	Cập nhật thông tin người dùng
------------	------	------	--------	-------------------------------

### 3.2.4.3. Lớp Đăng ký

Mô tả: lớp nhận thông tin người dùng và thực hiện đăng ký tài khoản

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
Username	String	Private	Tên khách muốn đăng ký
Email	String	Private	Email khách muốn đăng ký
Password	String	Private	Mật khẩu người dùng

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
signUp	Void	Void	Public	Đăng ký tài khoản
login	Void	Void	Public	Đăng nhập

### 3.2.4.4. Lớp Danh sách thiết bị

Mô tả: lớp thể hiện thông tin thiết bị và cung cấp các service với thiết bị

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
deviceName	String	Private	Tên thiết bị
embedId	String	Private	Id phần cứng thiết bị
state	Boolean	Private	Trạng thái kết nối thiết bị

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
getAllDeviceByUserId	Void	Void	Public	Lấy thông tin toàn bộ các thiết bị
getDeviceById	Void	Void	Public	Lấy thông tin của thiết bị dựa trên id
createDevice	Void	Void	Public	Tạo mới thiết bị
deleteDevice	Void	Void	Public	Xóa thiết bị
updateDevice	Void	Void	Public	Cập nhật thiết bị

### 3.2.4.5. Lớp Chi tiết thiết bị

Mô tả: lớp thể hiện chi tiết thông tin thiết bị

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
getLocation	Void	Void	Public	Hiện thị vị trí thiết bị
getDeviceById	Void	Void	Public	Lấy thông tin chi tiết thiết bị và dữ liệu thiết bị thu thập

### 3.2.4.6. Lớp Đăng nhập

Mô tả: lớp mô tả các thông tin đăng nhập và thực hiện đăng nhập

Các thuộc tính:

Tên thuộc tính	Kiểu dữ liệu	Phạm vi truy cập	Mô tả
----------------	--------------	------------------	-------

Username	String	Private	Tên đăng nhập
Password	String	Private	Mật khẩu của người dùng

Các phương thức:

Tên phương thức	Đầu vào	Đầu ra	Phạm vi truy cập	Mô tả
Login	Void	Void	Public	Thực hiện đăng nhập

### 3.3. Thiết kế cơ sở dữ liệu

#### 3.3.1. Sơ đồ bảng quan hệ dữ liệu

#### 3.3.2. Tổng quan bảng dữ liệu

Tên bảng dữ liệu	Mô tả
Device	Mô tả thiết bị
User	Mô tả người dùng
stateHistory	Mô tả dữ liệu thiết bị

#### 3.3.3. Thiết kế chi tiết các bảng dữ liệu

##### 3.3.3.1. Bảng Device

STT	Trường dữ liệu	Kiểu dữ liệu	Nullable	Ràng buộc	Mô tả
1	_id	String	No	PK	Id định danh thiết bị
2	embedId	String	No		Id phần cứng thiết bị
3	isPulic	Boolean	No		Trạng thái công cộng của thiết bị
4	deviceName	String	No		Tên thiết bị
5	connectState	String	No		Trạng thái kết nối thiết bị
6	Locaiton	String	No		Vị trí thiết bị
7	userId	String	No	FK	Id của người tạo

### 3.3.3.2. Bảng User

STT	Trường dữ liệu	Kiểu dữ liệu	Nullable	Ràng buộc	Mô tả
1	_id	String	No	PK	Id định danh người dùng
2	Username	String	No		Tên đăng nhập
3	password	String	No		Mật khẩu người dùng
4	Role	String	No		Phân loại người dùng

### 3.3.3.3. Bảng stateHistory

STT	Trường dữ liệu	Kiểu dữ liệu	Nullable	Ràng buộc	Mô tả
1	at	Date	No		Thời gian thiết bị

					nhận dữ liệu
2	Temperature	Int	Yes		Nhiệt độ
3	Co2	Int	Yes		Mức độ co2
4	Humidity	Int	Yes		Độ ẩm
5	dust	Int	Yes		Độ bụi
6	deviceId	String	No	FK	Id thiết bị nhận

## Chương 4: Công nghệ và công cụ sử dụng



## 4.1. ReactJS

ReactJS là một thư viện Javascript phổ biến được phát triển bởi Facebook, chủ yếu được sử dụng để xây dựng giao diện người dùng động và hiệu suất cao trong các ứng dụng web. Được biết đến với kiến trúc component và cách quản lý trạng thái linh hoạt. ReactJS mang lại nhiều lợi ích cho nhà phát triển.

Ưu điểm:

- Kiến trúc Component: ReactJS giúp tổ chức ứng dụng theo các module và tái sử dụng linh hoạt thông qua việc sử dụng component.
- Virtual DOM: sử dụng Virtual DOM giúp tối ưu hiệu suất, giảm tải cho máy chủ và cải thiện trải nghiệm người dùng.
- Thư viện Redux: Được tích hợp tốt với Redux để quản lý trạng thái ứng dụng một cách hiệu quả.
- Cộng đồng mạnh mẽ: ReactJS có một cộng đồng lớn và nhiệt tình, giúp đội ngũ phát triển có nguồn hỗ trợ và tài liệu đa dạng.

Nhược điểm:

- Ecosystem phức tạp: Với một số lựa chọn và thư viện phong phú, việc lựa chọn thư viện phù hợp có thể làm phức tạp hóa quá trình phát triển.

## 4.2. NodeJS

Nodejs là một nền tảng được phát triển độc lập trên V8 JavaScript Engine – trình thông thực thi mã JavaScript. Nhờ Nodejs mà việc xây dựng các ứng dụng Web trở nên đơn giản và dễ dàng hơn rất nhiều. Ngoài ra, Nodejs còn được biết tới là một mã nguồn mở và là một môi trường cho các máy chủ và ứng dụng mạng.

Ưu điểm:

- Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ.

- Giúp bạn dễ dàng mở rộng khi có nhu cầu phát triển website.
- Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử Nodejs lý nhanh hơn rất nhiều.
- Có khả năng xử lý nhiều Request cùng một lúc trong thời gian ngắn nhất.
- Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.
- Phù hợp để xây dựng những ứng dụng thời gian thực như các ứng dụng chat, mạng xã hội ...

Nhược điểm: Nodejs gây hao tốn tài nguyên và thời gian.

### 4.3. MongoDB

MongoDB là một cơ sở dữ liệu NoSQL hướng đối tượng, đơn giản, linh động và có thể mở rộng.

Nó dựa trên mô hình lưu trữ NoSQL document.mNgôn ngữ MongoDB là triển khai một kho lưu trữ dữ liệu cung cấp hiệu suất cao, tính sẵn sàng cao và tự động mở rộng. MongoDB sử dụng JSON hoặc BSON document để lưu trữ dữ liệu.

Ưu điểm:

- Document oriented.
- Hiệu suất cao.
- Tính sẵn sàng cao – Nhân rộng.
- Khả năng mở rộng cao – Sharding.
- Năng động – Không có lược đồ cứng nhắc.
- Linh hoạt – thêm / xóa trường có ít hoặc không ảnh hưởng.
- Dữ liệu không đồng nhất.
- Không joins.
- Phân phối được.
- Biểu diễn dữ liệu trong JSON hoặc BSON.
- Hỗ trợ không gian địa lý.

- Tích hợp dễ dàng với BigData Hadoop.
- Ngôn ngữ truy vấn dựa trên tài liệu mạnh mẽ như SQL.

Nhược điểm:

- Một nhược điểm của NoSQL là hầu hết các giải pháp đều không tuân thủ ACID mạnh mẽ (Atomic, Consistency, Isolation, Durability) như các hệ thống RDBMS được thiết lập tốt hơn.
- Giao dịch phức tạp.
- Không có chức năng hoặc thủ tục lưu trữ tồn tại nơi bạn có thể liên kết logic.

## 4.4. NodeMCU ESP8266

ESP8266 là một chip của Espressif Systems có tích hợp công nghệ Wi-Fi với đặc tính giá rẻ, tương thích với nhiều nền tảng.

Thông số của ESP8266:

- Tần số hoạt động: 80 MHz
- Bộ nhớ flash: 1MB
- 17 GPIO pins
- Giao tiếp: I 2C, SPI, I<sup>2</sup>S, UART
- Đặc biệt: 10bit ADC

## 4.5. Message Queue Telemetry Transport (MQTT)

Giao thức truyền thông điệp (message) theo mô hình publish/subscribe

Là giao thức được sử dụng rộng rãi trong lĩnh vực IoT vì có nhiều ưu điểm như:

- Nhẹ và hiệu quả.
- Truyền thông 2 hướng.
- Khả năng mở rộng tới hàng triệu thiết bị.
- Truyền tin tin cậy (theo 3 mức độ): MQTT hỗ trợ 3 mức QoS:
  - 0 – nhiều nhất một lần
  - 1 – ít nhất một lần

- 2 – duy nhất một lần
- Hỗ trợ môi trường mạng không ổn định.
- Hỗ trợ nâng cao bảo mật trong truyền tin với giao thức TLS và các giao thức xác thực tin cậy như OAuth

## 4.6. Hypertext Transfer Protocol (HTTP)

Giao thức hoạt động ở tầng application dùng để truyền thông tin dạng siêu văn bản như HTML, hoạt động theo mô hình client-server.

Được sử dụng rộng rãi trong môi trường web, là phương thức cực kỳ phổ biến giúp kết nối các trình duyệt web và các web servers trên khắp thế giới.

## 4.7. Github

Github là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

## 4.8. Arduino IDE

Arduino IDE là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino. Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác.

Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin dưới dạng mã. Mã chính, còn được gọi là sketch, được tạo trên nền tảng IDE sẽ tạo ra một file Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên bo.

Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch, phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino. Môi trường này hỗ trợ cả ngôn ngữ C và C++.

## 4.9. Visual Studio Code

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor. Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

## Chương 5: Triển khai cài đặt

### Chương trình và kết quả thử nghiệm

#### 5.1. Triển khai cài đặt

1. Clone repo link từ github:  
<https://github.com/kira210402/IoT-Project.git>
2. Mở Arduino IDE, kết nối máy tính với thiết bị phần cứng và tiến hành chạy nạp code vào phần cứng với file hardware.ino

3. Chuyển đến folder Backend, và open Terminal

npm i

npm start

4. Chuyển đến folder Frontend, và mở Terminal

npm i

npm start

Mở cổng localhost:8088

## 5.2. Chương trình minh họa

- Đăng ký

Đăng Ký

×

Username \*

Email \*

Password \*

ĐĂNG KÝ ➤

- Đăng nhập

## Đăng nhập

Tài Khoản

thang

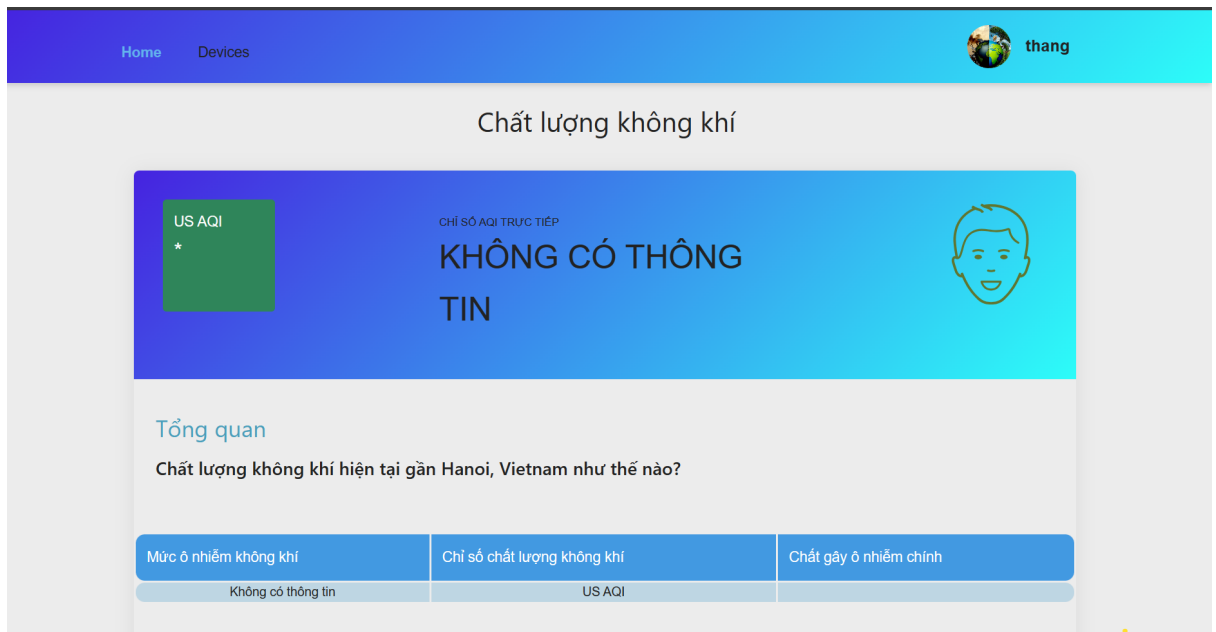
Mật khẩu

\*\*\*\*\*

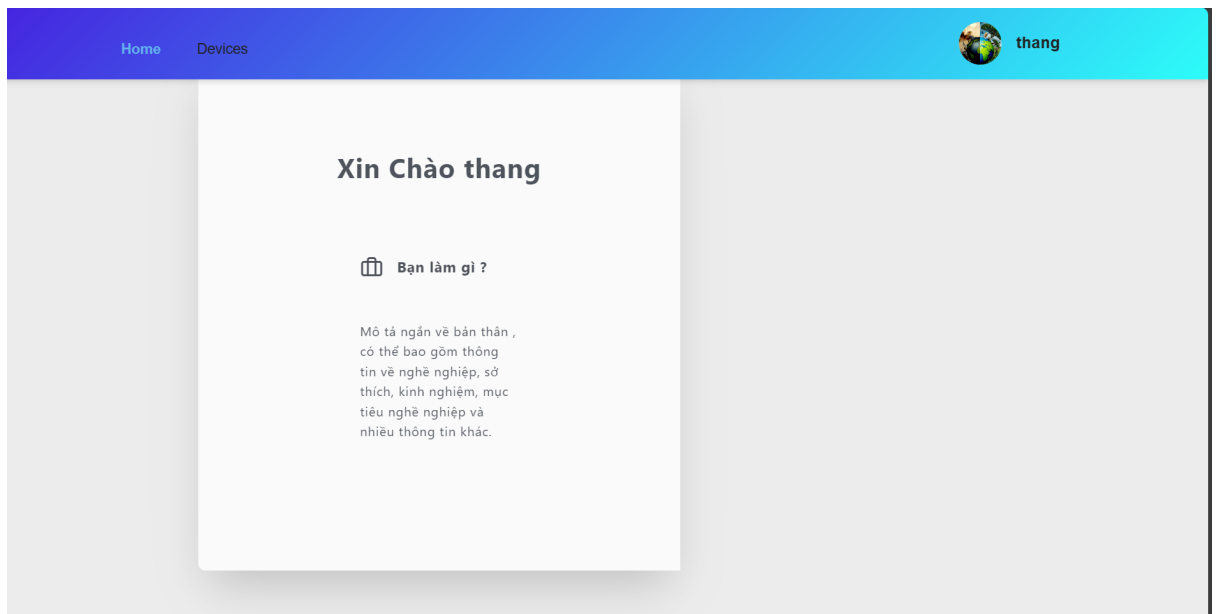
Đăng Nhập

Tạo tài khoản mới

### - Giao diện trang chủ



### - Thông tin cá nhân



- Thêm thiết bị

×

Device

Device Id

Embed ID

Device Name

Device Name

Device State


☐

Add

- Xem thiết bị



[Home](#) [Devices](#)

 thang

Device's id: 659a3429301303d45084678b

Device's name: frerrr


Device's userid: 65996f4f02edef1f1f75415f

Tổng quan

Chất lượng không khí hiện tại vị trí thiết bị frerrr như thế nào?

Device's id	Device's name	Device's userid
659a3429301303d45084678b	frerrr	65996f4f02edef1f1f75415f

Thời tiết



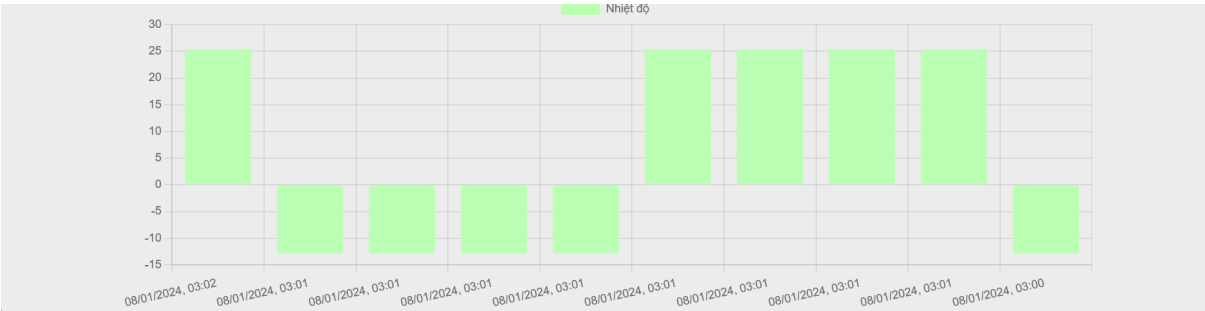
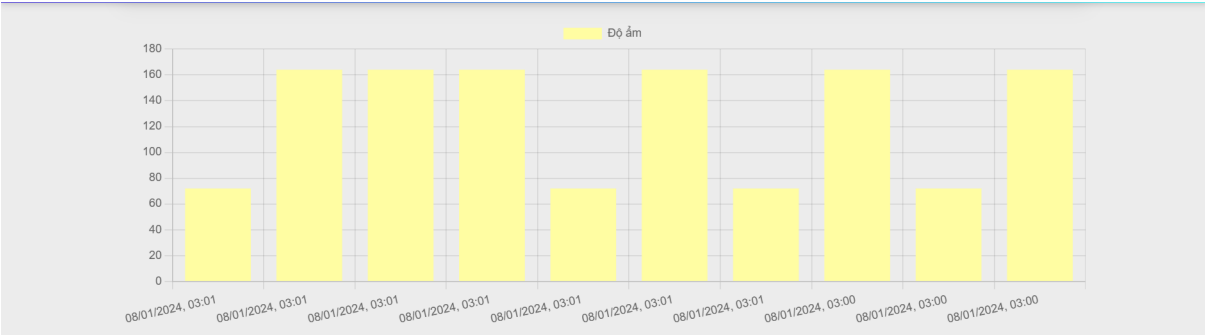
Thời tiết hiện tại ở vị trí thiết bị frerrr ra sao?

Nhiệt độ

25.4

Độ ẩm

72



- Đăng xuất



**thang**



Infomation



Logout

## Chương 6: Kết luận và hướng phát triển

### 6.1. Phân chia công việc thành viên nhóm

Công việc	Mô tả chi tiết	Thành viên thực hiện
Tìm hiểu bài toán	Tìm hiểu logic bài toán + Lập kế hoạch triển khai	Cả nhóm
Thiết kế hệ thống	Thiết kế use-case, tổng quan hệ thống, cơ sở dữ liệu	Cả nhóm
Triển khai phần cứng	Tìm hiểu, đặt mua, lắp đặt và code phần cứng	Dương Kim Nam
Triển khai backend		Nguyễn Văn Thắng
Triển khai frontend		Dương Tiến Hoàng
Viết báo cáo		Cả nhóm
Làm slide thuyết trình		Dương Kim Nam

### 6.2. Kết luận

Trong quá trình thực hiện đề tài, nhóm chúng em đã cố gắng hết sức để tìm hiểu nghiệp vụ, xây dựng thiết kế và tìm hiểu công nghệ IoT để xây dựng và cài đặt chương trình nhưng vì thời gian có hạn nên chưa thể giải quyết được toàn bộ các vấn đề. Chúng em mong nhận được sự cảm thông và góp ý của thầy. Nhóm chúng em xin đánh giá một vài kết quả thu được như sau:

- Xây dựng được căn bản một hệ thống quan trắc có sự tham gia của các yếu tố công nghệ phía nhúng, phía phần mềm và server.
- Nắm được cách hai giao thức HTTP và MQTT hoạt động trong môi trường IoT.
- Hệ thống chạy ổn định.
- Giao diện web đẹp, trực quan, thân thiện với người dùng.
- Thiết bị dễ dàng kết nối, giá thành rẻ.

- Thấy được rõ hơn việc ứng dụng Internet of Things vào thực tế, rõ hơn workflow xử lý của một bài toán IoT.
- Kỹ năng làm việc nhóm, học nhóm.

### **6.3. Hạn chế**

- Thiết bị công kênh.
- Kiến thức về domain logic của bài toán ô nhiễm không khí, quan trắc không khí chưa nhiều.
- Các thiết bị đo chưa chính xác do giá thành thấp nên chất lượng không được cao, dẫn đến dữ liệu gửi về có sai số lớn, bị ảnh hưởng nhiều bởi nhiễu.
- Hệ thống chưa đáp ứng được lượng dữ liệu lớn

### **6.4. Hướng phát triển**

Với những nhược điểm nêu trên, nhóm chúng em sẽ cần phát triển bài toán tốt hơn, với định hướng như sau:

- Cần làm thêm các chức năng phía quản trị viên để giúp quản lý cấu hình các thiết bị phần cứng.
- Cần chú ý tới một số vấn đề như bảo mật cho hệ thống IoT.
- Cần tối ưu hơn về hiệu năng của hệ thống.
- Tìm hiểu sâu hơn về domain logic của bài toán như các thuật toán đánh giá chất lượng không khí, các thông số quan trọng gây ảnh hưởng tới chất lượng không khí.
- Thu gọn thiết bị phần cứng.
- Tăng khả năng xử lý dữ liệu.
- Có nhiều loại thống kê hơn

# TÀI LIỆU THAM KHẢO

1. Bài giảng học phần “IoT và ứng dụng” - IT4735 - TS.Phạm Ngọc Hưng
2. [Arduino Documentation](#)
3. [Reading Temperature Wirelessly over WiFi with DHT11, DHT22 and ESP8266 NodeMCU by Robojax](#)
4. [Cách kết nối thiết bị với server thông qua giao thức MQTT](#)