ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**Ảnh có chứa văn bản, ký hiệu

Mô tả được tạo tự động**

**BÁO CÁO BÀI TẬP LỚN**

**Đề tài: Quan trắc chất lượng không khí**

|  |  |
| --- | --- |
| Lớp                                 : | 136838 |
| Học phần                        : | IoT và ứng dụng |
| Mã học phần                   : | IT3190 |
| Giảng viên hướng dẫn    : | TS. Phạm Ngọc Hưng |

Danh sách thành viên nhóm:

|  |  |
| --- | --- |
| Họ và tên | Mã số sinh viên |
| Nguyễn Xuân Mạnh Hùng | 20194292 |
| Trương Văn Hiển | 20194276 |
| Vũ Ngọc Quang Huy | 20194301 |
| Hà Trung Kiên | 20190078 |

*Hà Nội, tháng 3 năm 2023*

**MỤC LỤC**

[LỜI NÓI ĐẦU 4](#_Toc129158601)

[Chương 1: Giới thiệu đề tài 5](#_Toc129158602)

[1.1. Đặt vấn đề 5](#_Toc129158603)

[1.2. Mục tiêu, phạm vi đề tài 5](#_Toc129158604)

[Chương 2: Phân tích yêu cầu bài toán 7](#_Toc129158605)

[2.1. Phân tích yêu cầu tổng quan 7](#_Toc129158606)

[2.2. Sơ đồ tổng quan hệ thống 7](#_Toc129158607)

[2.3. Phân tích yêu cầu chức năng 8](#_Toc129158608)

[2.3.1. Các tác nhân chính 8](#_Toc129158609)

[2.3.2. Biểu đồ use case tổng quan 8](#_Toc129158610)

[2.3.3. Biểu đồ phân rã use case 8](#_Toc129158611)

[Chương 3: Phân tích thiết kế bài toán 20](#_Toc129158612)

[3.1. Biểu đồ trình tự tương tác hệ thống 20](#_Toc129158613)

[3.2. Thiết kế chi tiết lớp 23](#_Toc129158614)

[3.2.1. Sơ đồ gói 23](#_Toc129158615)

[3.2.2. Sơ đồ tổng quan gói Model 23](#_Toc129158616)

[3.2.3. Sơ đồ tổng quan gói Controller 25](#_Toc129158617)

[3.2.4. Sơ đồ tổng quan gói View 28](#_Toc129158618)

[3.3. Thiết kế cơ sở dữ liệu 31](#_Toc129158619)

[3.3.1. Sơ đồ bảng quan hệ dữ liệu 31](#_Toc129158620)

[3.3.2. Tổng quan bảng dữ liệu 32](#_Toc129158621)

[3.3.3. Thiết kế chi tiết các bảng dữ liệu 32](#_Toc129158622)

[Chương 4: Công nghệ và công cụ sử dụng 33](#_Toc129158623)

[4.1. ReactJS 33](#_Toc129158624)

[4.2. Nodejs 33](#_Toc129158625)

[4.3. MongoDB 34](#_Toc129158626)

[4.4. NodeMCU ESP8266 35](#_Toc129158627)

[4.5. MQ Telemetry Transport (MQTT) 35](#_Toc129158628)

[4.6. Hypertext Transfer Protocol (HTTP) 36](#_Toc129158629)

[4.7. Github 37](#_Toc129158630)

[4.8. Arduino IDE 37](#_Toc129158631)

[4.9. Visual Studio Code 37](#_Toc129158632)

[Chương 5: Triển khai cài đặt 38](#_Toc129158633)

[Chương trình và kết quả thử nghiệm 38](#_Toc129158634)

[5.1. Triển khai cài đặt 38](#_Toc129158635)

[5.2. Chương trình minh hoạ 38](#_Toc129158636)

[Chương 6: Kết luận và hướng phát triển 42](#_Toc129158637)

[6.1. Phân chia công việc thành viên nhóm 42](#_Toc129158638)

[6.2. Kết luận 43](#_Toc129158639)

[6.3. Hạn chế 43](#_Toc129158640)

[6.4. Hướng phát triển 44](#_Toc129158641)

[TÀI LIỆU THAM KHẢO 45](#_Toc129158642)

# **LỜI NÓI ĐẦU**

Trong vài năm qua, Internet of Things đã trở thành một trong những công nghệ quan trọng nhất của thế kỷ 21**.** Internet of Things, hay IoT**,** Internet vạn vật đề cập đến hàng tỷ thiết bị vật lý trên khắp thế giới hiện được kết nối với internet, tất cả đều thu thập và chia sẻ dữ liệu. Nhờ sự xuất hiện của chip máy tính siêu rẻ và sự phổ biến của mạng không dây, có thể biến bất cứ thứ gì, từ thứ nhỏ như viên thuốc đến thứ lớn như máy bay, thành một phần của IoT. Việc kết nối tất cả các đối tượng khác nhau này và thêm các cảm biến vào chúng sẽ tăng thêm mức độ thông minh kỹ thuật số cho các thiết bị vật lý, cho phép chúng giao tiếp dữ liệu thời gian thực mà không cần đến con người. Internet of Things đang làm cho cấu trúc của thế giới xung quanh chúng ta trở nên thông minh hơn và phản ứng nhanh hơn, hợp nhất công nghệ vật lý và kỹ thuật số.

Với sự phát triển mạnh mẽ của Internet và công nghệ, IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Xu hướng IOT hiện nay đang dần trở nên phổ biến trên thế giới bới tính tiện lợi, đa dạng và hiện đại của nó.

Tác động của IoT rất đa dạng, trên các lĩnh vực: quản lý hạ tầng, y tế, xây dựng và tự động hóa, giao thông, trồng trọt, chăn nuôi… Với mục đích học tập, nghiên cứu công nghệ trên lĩnh vực IOT và hoàn thành học phần **IOT và ứng dụng***,* nhóm 14 chúng em lựa chọn đề tài **“Quan trắc chất lượng không khí”** làm đề tài Bài tập lớn của học phần.

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức và sai sót trong kiểm thử. Chúng em rất mong nhận được những đóng góp cũng như nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, nhóm chúng em xin được gửi lời cảm ơn đến **TS. Phạm Ngọc Hưng** 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**

## **Đặt vấn đề**

Theo một nghiên cứu của Tổ chức Y tế Thế giới (WHO) thì ô nhiễm không khí là nguyên nhân ra 7 triệu ca tử vong mỗi năm trên toàn cầu. Cũng theo nghiên cứu này có tới 97% thành phố ở các quốc gia thu nhập thấp và trung bình không đáp ứng các tiêu chuẩn về chất lượng không khí do WHO đề ra.

Các quốc gia đang phát triển và có dân số đông như Trung Quốc, Ấn Độ,... đang là những nước có mức ô nhiễm không khí nặng nề nhất. Các nước phát triển tình trạng ô nhiễm không khí chỉ ít nghiêm trọng hơn chứ không thực sự khả quan quan lắm. Tại các nước châu âu, ô nhiễm không khí là một trong những nguyên nhân chính gây ra các bệnh về hô hấp như tim mạch, ung thư…

Thời gian gân đây tình trạng ô nhiễm không khí đã được cải thiện do các nước đã quan tâm, cam kết và có những hành động tích cực để cải thiện môi trường không khí. Tuy nhiên vẫn chưa đủ, vẫn cần nhiều sự quan tâm hơn, những hành động quyết liệt hơn với tình trạng ô nhiễm không khí như hiện nay.

## **Mục tiêu, phạm vi đề tài**

Hiện nay ở Việt Nam đã có những trang web cho phép do lường chất lượng không khí nhưng vẫn còn rất nhiều hạn chế và chưa hỗ trợ việc thêm thiết bị của người dùng, điều này không hỗ trợ được người dùng muốn tìm hiểu chất lượng không khí tại một hoặc nhiều địa điểm mà người dùng muốn biết dựa trên phần cứng của bản thân.

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 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**

## **Phân tích yêu cầu tổng quan**

Để có thể thiết kế được một hệ thống hỗ trợ đo chất lượng không khí, trước tiên chúng ta cần hiều được các luồng sự kiện có thể diễn ra trong hệ thống.

Sau đây là quy trình nghiệp vụ chính của người dùng.

Sau khi truy cập vào hệ thống:

* Hệ thống sẽ thông qua trình duyệt brower 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.

## **Sơ đồ tổng quan hệ thống**

Chart

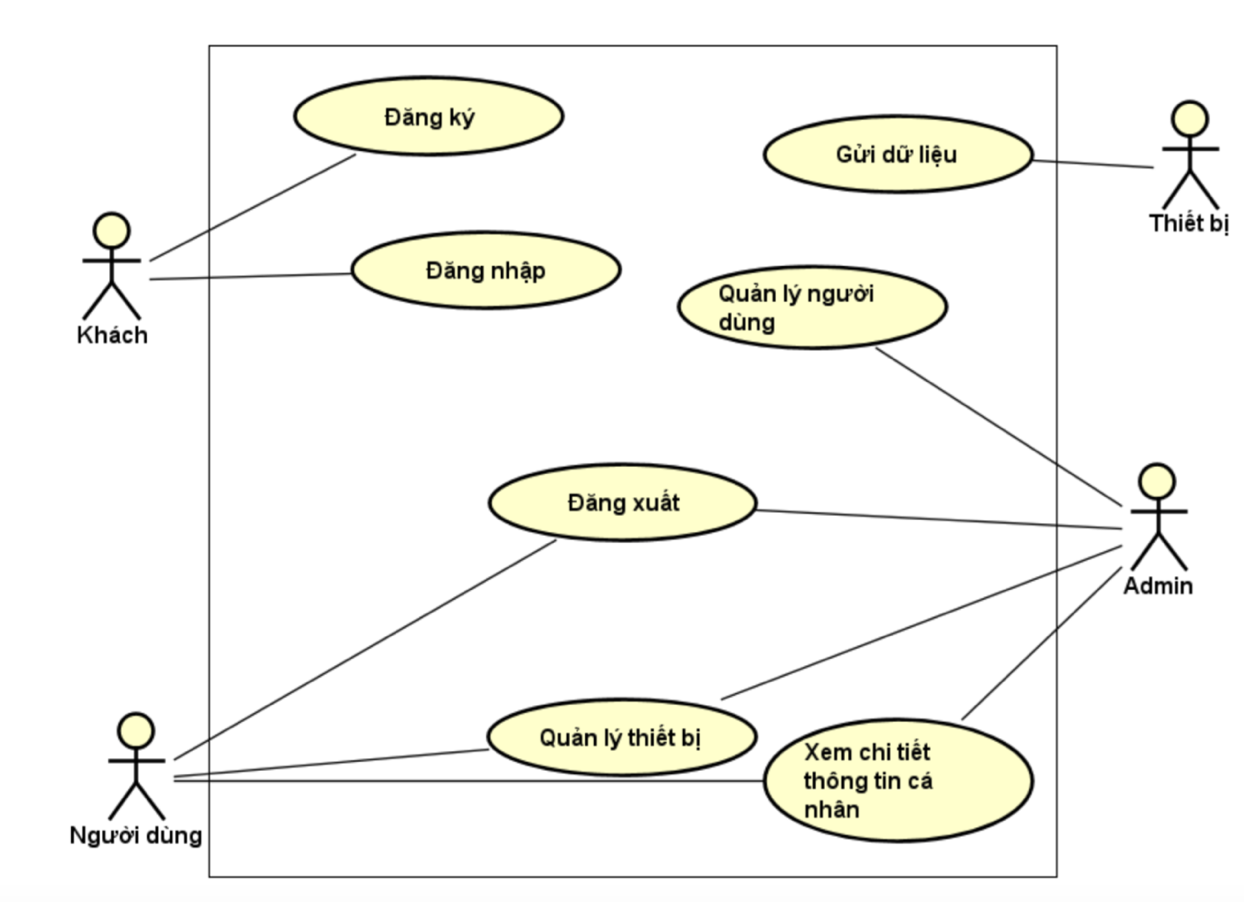
Description automatically generated

## **Phân tích yêu cầu chức năng**

### **Các tác nhân chính**

Hệ thống gồm 2 tác nhân chính:

1. Người dùng
2. Khách
   * 1. **Biểu đồ use case tổng quan**



* + 1. **Biểu đồ phân rã use case**

#### **Nhóm tác nhân Khách**

Diagram

Description automatically generated

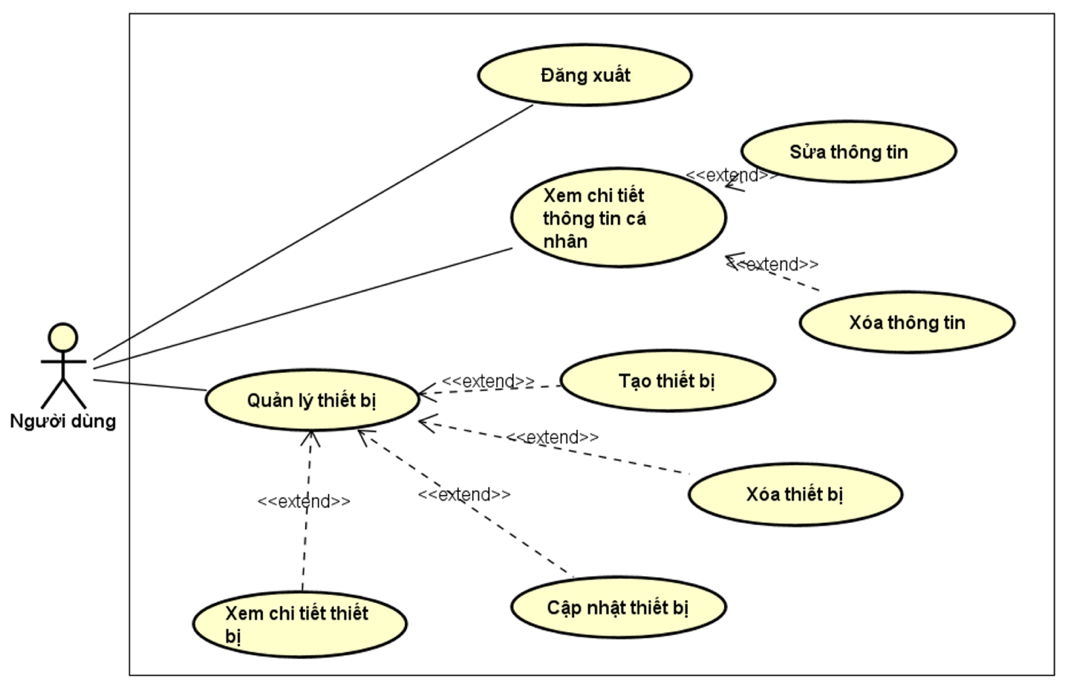
* Đặc tả use case UC001: “Đăng ký”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC001 | Tên Use case | Đăng ký |
| Mục đích | Để khách có thể đăng nhập thành người dùng | | |
| Tác nhân | Khách | | |
| Sự kiện  kích hoạt | Khi khách bấm vào “Đăng ký” | | |
| Điều kiện  tiên quyết | Khách chưa có tài khoản | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách | Chọn chức năng Đăng ký | | 2 | Hệ thống | Hiển thị giao diện trang đăng ký | | 3 | Khách | Nhập đầy đủ thông tin đăng ký | | 4 | Khách | Gửi yêu cầu đăng ký | | 5 | Hệ thống | Kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa | | 6 | Hệ thống | Kiểm tra username có bị trùng hay không | | 7 | Hệ thống | Kiểm tra email có hợp lệ hay không | | 8 | Hệ thống | Kiểm tra mật khẩu nhập lại có trùng khớp với mật khẩu nhập ở trên không | | 9 | Hệ thống | Lưu vào database và thông báo đăng ký thành công | | 10 | Hệ thống | Điều hướng sang trang chủ | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu | | 7a | Hệ thống | Thông báo lỗi: username đã tồn tại | | 8a | Hệ thống | Thông báo lỗi: Email không hợp lệ hoặc đã được sử dụng | | 9a | Hệ thống | Thông báo lỗi: Mật khẩu nhập lại không khớp | | | |
| Hậu điều kiện | Thông báo đăng ký thành công | | |

* Đặc tả use case UC002: “Đăng nhập”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC002 | Tên Use case | Đăng nhập |
| Mục đích | Giúp khách thành người dùng của hệ thống | | |
| Tác nhân | Khách | | |
| Sự kiện  kích hoạt | Khi khách bấm vào “Đăng nhập” | | |
| Điều kiện  tiên quyết | Khách đã có tài khoản trong hệ thống | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách | Chọn chức năng Đăng nhập | | 2 | Hệ thống | Hiển thị giao diện trang đăng nhập | | 3 | Khách | Nhập username và password để đăng nhập | | 4 | Khách | Gửi yêu cầu đăng nhập | | 5 | Hệ thống | Kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa | | 6 | Hệ thống | Kiểm tra username và password có hợp lệ so với dữ liệu đã lưu trong cơ sở dữ liệu của hệ thống hay không | | 7 | Hệ thống | Mở giao diện trang chủ ứng với loại tài khoản đăng nhập | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu | | 7a | Hệ thống | Thông báo lỗi: username hoặc password không chính xác nếu không tìm thấy tài khoản trong hệ thống | | | |
| Hậu điều kiện | Hiển thị giao diện trang chủ ứng với loại tài khoản người dùng | | |

#### **Nhóm tác nhân Người dùng**

****

* Đặc tả use case UC003: “Đăng xuất”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC003 | Tên Use case | Đăng xuất |
| Mục đích | Người dùng đăng xuất khỏi hệ thống | | |
| Tác nhân | Người dùng | | |
| Sự kiện  kích hoạt | Khi người dùng bấm vào “Đăng xuất” | | |
| Điều kiện  tiên quyết | Khách đã có tài khoản trong hệ thống | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng Đăng xuất | | 2 | Hệ thống | Xóa token và đăng xuất người dùng | | 3 | Hệ thống | Mở giao diện đăng nhập | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2a | Hệ thống | Thông báo lỗi: không thể đăng xuất | | 3a | Hệ thống | Giữ nguyên màn hìn hiện tại | | | |
| Hậu điều kiện | Hiển thị giao diện trang chủ ứng với loại tài khoản người dùng | | |

* Đặc tả usecase UC004: “Sửa thông tin”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC004 | Tên Use case | Sửa thông tin |
| Mục đích | Người dùng muốn thay đổi thông tin cá nhân | | |
| Tác nhân | Người dùng | | |
| Sự kiện  kích hoạt | Người dùng ấn vào “Chỉnh sửa thông tin cá nhân” | | |
| Điều kiện  tiên quyết | Người dùng đang ở trang thông tin cá nhân | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn “Chỉnh sửa thông tin cá nhân” | | 2 | Hệ thống | Hiển thị giao diện chỉnh sửa thông tin cá nhân | | 3 | Người dùng | Sửa các trường dữ liệu cần thay đổi | | 4 | Người dùng | Nhấn nút lưu thông tin | | 5 | Hệ thống | Kiểm tra xem đã nhập các trường bắt buộc nhập hay chưa | | 6 | Hệ thống | Kiểm tra thông tin các trường có đúng định dạng hay không | | 7 | Hệ thống | Cập nhập lại database | | 8 | Hệ thống | Thông báo thay đổi thông tin thành công | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu nhập thiếu | | 7a | Hệ thống | Thông báo lỗi: Thông tin chưa nhập đúng định dạng | | | |
| Hậu điều kiện | Hiển thị thông báo thay đổi thông tin thành công | | |

* Đặc tả usecase UC005: “Xem chi tiết thông tin cá nhân”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC005 | Tên Use case | Xem chi tiết thông tin cá nhân |
| Mục đích | Giúp người dùng thay đổi thông tin cá nhân | | |
| Tác nhân | Người dùng | | |
| Sự kiện  kích hoạt | Người dùng ấn vào xem thông tin cá nhân | | |
| Điều kiện  tiên quyết | Không | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Ấn xem thông tin cá nhân | | 2 | Hệ thống | Hiển thị thông tin cá nhân người dùng   * Nếu muốn sửa thông tin thì thực hiện nghiệp vụ “Sửa thông tin” | | | |
| Luồng sự kiện thay thế | Không có | | |
| Hậu điều kiện | Hiển thị thông tin cá nhân cho người dùng | | |

* Đặc tả UC006: “Quản lý thiết bị”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC006 | Tên Use case | Quản lý thiết bị |
| Mục đích | Người dùng xem danh sách các thiết bị mà mình đã đăng ký | | |
| Tác nhân | Người dùng | | |
| Sự kiện  kích hoạt | Người dùng chọn chức năng quản lý thiết bị | | |
| Điều kiện  Tiên quyết | Không có | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn xem danh sách thiết bị | | 2 | Hệ thống | Hiển thị danh sách thiết bị người dùng đã đăng ký   * Nếu muốn sửa thông tin thiết bị thì thực hiện nghiệp vụ “sửa thiết bị” * Nếu muốn xóa thiết bị thì thực hiện nghiệp vụ “Xóa thiết bị” * Nếu muốn thêm thiết bị thì thực hiện nghiệp vụ “Thêm thiết bị” | | | |
| Luồng sự kiện thay thế | Không có | | |
| Hậu điều kiện | Không có | | |

* Đặc tả usecase UC007: “Tạo thiết bị”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC007 | Tên Use case | Tạo thiết bị |
| Mục đích | Người dùng tạo mới thiết bị | | |
| Tác nhân | Người dùng | | |
| Sự kiện  kích hoạt | Người dùng chọn chức năng “Thêm thiết bị” | | |
| Điền kiện tiên quyết | Không có | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng “Thêm thiết bị” | | 2 | Hệ thống | Hiện thị trường nhập thông tin thiết bị | | 3 | Người dùng | Nhập các thông tin cần thiết, nhấn gửi | | 4 | Hệ thống | Kiểm tra thông tin thiết bị đã có trước đó hay chưa | | 5 | Hệ thống | Thêm thiết bị vào database và load lại trang với danh sách thiết bị mới | | 6 | Hệ thống | Thông báo thêm thành công | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo lỗi: không thể thêm thiết bị do đã tồn tại | | 3a | Hệ thống | Giữ nguyên màn hìn hiện tại | | | |
| Hậu điều kiện | Hiển thị thông báo thêm thành công | | |

* Đặc tả usecase UC008: “Xóa thiết bị”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC008 | Tên Use case | Xóa thiết bị |
| Mục đích | Người dùng xóa thiết bị đã thêm | | |
| Tác nhân | Người dùng | | |
| Sự kiện kích hoạt | Người dùng chọn chức năng “Xóa thiết bị” | | |
| Điều kiện tiên quyết | Không có | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng xóa thiết bị | | 2 | Hệ thống | Xóa thiết bị | | 3 | Hệ thống | Load lại trang, thông báo xóa thành công | | | |
| Luồng sự kiện thay thế | Không có | | |
| Hậu điều kiện | Hiển thị thông báo xóa thành công | | |

* Đặc tả usecase UC009: “Cập nhật thiết bị”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC009 | Tên Use case | Cập nhật thiết bị |
| Mục đích | Người dùng sửa thông tin thiết bị của mình | | |
| Tác nhân | Người dùng | | |
| Sự kiện kích hoạt | Người dùng chọn chức năng “Cập nhật thiết bị” | | |
| Điều kiện  Tiên quyết | Không có | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng sửa thiết bị | | 2 | Hệ thống | Hiển thị trường nhập thông tin thiết bị với mặc định là các thông tin đã thêm trước đó | | 3 | Người dùng | Sửa các thông tin | | 4 | Hệ thống | Cập nhật thông tin, thông báo thành công và load lại trang | | | |
| Luồng  sự kiện thay thế | Không có | | |
| Hậu điều kiện | Hiển thị thông báo thay đổi thông tin thiết bị thành công | | |

* Đặc tả usecase UC010: “Xem chi tiết thiết bị”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC010 | Tên Use case | Xem chi tiết thiết bị |
| Mục đích | Người dùng xem dữ liệu thiết bị thu thập được | | |
| Tác nhân | Người dùng | | |
| Sự kiện kích hoạt | Người dùng chọn chức năng “Xem chi tiết” | | |
| Điều kiện  tiên quyết | Không có | | |
| Luồng  sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng xem chi tiết | | 2 | Hệ thống | Lấy vị trí hiện tại của thiết bị gửi lên và hiện thị trên màn hình map, đọc dữ liệu thiết bị thu thập dược biểu diễn trên biểu đồ | | | |
| Luồng sự kiện thay thế | Không có | | |
| Hậu điều kiện | Hiển thị thông báo thay đổi thông tin thành công | | |

#### **Nhóm tác nhân Thiết bị**

Đặc tả use case UC011: “Gửi dữ liệu”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC011 | Tên Use case | Gửi dữ liệu |
| Mục đích sử dụng | Thiết bị gửi dữ liệu thi thập được cho hệ thống | | |
| Tác nhân | Thiết bị | | |
| Sự kiện kích hoạt | Thiết bị gửi dữ liệu thông qua publisher MQTT của hệ thống | | |
| Điều kiện  tiên quyết | Không có | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3a | Hệ thống | Xóa dữ liệu mà thiết bị đã gửi đi | | | |
| Hậu điều kiện | Không có | | |

# **Chương 3: Phân tích thiết kế bài toán**

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

* Đăng nhập:

Diagram, schematic

Description automatically generated

* Đăng xuất:

Diagram

Description automatically generated

* Sửa thông tin:

Diagram

Description automatically generated

* Quản lý thiết bị:

Diagram

Description automatically generated

* Tạo thiết bị:

Diagram

Description automatically generated

* Xoá thiết bị:

Diagram

Description automatically generated

* Cập nhật thiết bị:

Diagram

Description automatically generated

* Xem chi tiết thiết bị:

Diagram

Description automatically generated

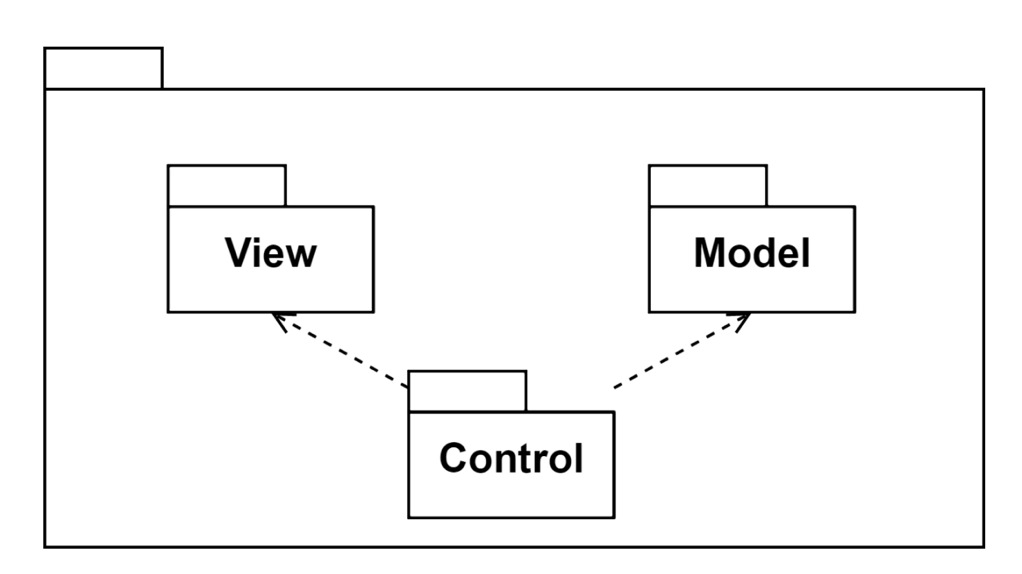
* Gửi dữ liệu:

Diagram

Description automatically generated

## **Thiết kế chi tiết lớp**

* + 1. **Sơ đồ gói**

****

* + 1. **Sơ đồ tổng quan gói Model**

Diagram

Description automatically generated

* + - 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 |
| \_id | String | Private | Mã thiết bị lưu trên database |
| embedId | String | Private | Mã thiết bị phần cứng |
| deviceName | String | Private | Tên thiết bị |
| connectState | String | Private | Trang thái hoạt động |
| userId | Sting | Private | Mã người tạo |
| stateHistory | List<stateHistory> | Private | Dữ liệu thiết bị |
| isPublic | Boolean | Private | Trang thái công cộng thiết bị |

* + - 1. **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 |
| usename | String | Private | Tên người dùng |
| password | String | Private | Mật khẩu người dùng |
| role | String | Private | Phân loại người dùng |

* + - 1. **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 |
| Temperatuer | Int | Private | Nhiệt độ |
| Co2 | Int | Private | Nồng độ co2 |
| Humidity | Int | Private | Độ ẩm |
| dust | int | Private | Độ bụi |

* + 1. **Sơ đồ tổng quan gói Controller**

Text

Description automatically generated with low confidence

* + - 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 |
| validatorForValidId | String | Void | Public | Kiểm tra hợp lệ id của thiết bị |
| ValidatorForDeviceExists | String | Void | Public | Kiểm tra thiết bị đã tồn tại chưa |
| getAllDevice | Void | List<device> | Public | Trả danh sách thiết bị |
| getDeviceByDeviceId | String | device | Public | Lấy thiết bị theo id của thiết bị trên database |
| getDeviceByEmbedId | String | Device | Public | Lấy thiết bị theo id phần cứng của thiết bị |
| createDeviceByUserId | String, device | Device | Public | Tạo mới thiết bị |
| updateDeviceByDeviceId | String, device | Device | Public | Cập nhật thiết bị |
| updateStateHistoryByDeviceId | String, device | Device | Public | Cập nhật trạng thái thiết bị |
| deleteDeviceByDeviceId | String | void | Public | Xóa thiết bị |

* + - 1. **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<user> | 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 |

* + 1. **Sơ đồ tổng quan gói View**

A picture containing diagram

Description automatically generated

* + - 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 |

* + - 1. **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 |

* + - 1. **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 phức:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

* + - 1. **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 |
| getDeviceByDeviceId | 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 |

* + - 1. **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 rí thiết bị |
| getDeviceByDeviceId | Void | Void | Public | Lấy thông tin chi tiết thiết bị và dữ liệu thiết bị thu thập |

* + - 1. **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 thiện đăng nhập |

## **Thiết kế cơ sở dữ liệu**

* + 1. **Sơ đồ bảng quan hệ dữ liệu**

Graphical user interface, application

Description automatically generated

* + 1. **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 |

* + 1. **Thiết kế chi tiết các bảng dữ liệu**
       1. **Bảng Device**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Nullable | Ràng buộc | Mô tả |
|  | \_id | String | No | PK | Id định danh thiết bị |
|  | embedId | String | No |  | Id phần cứng thiết bị |
|  | isPulic | Boolean | No |  | Trang thái công cộng của thiết bị |
|  | deviceName | String | No |  | Tên thiết bị |
|  | connectState | String | No |  | Trạng thái kết nối thiết bị |
|  | Locaiton | String | no |  | Vị trí thiết bị |
|  | userId | String | no | FK | Id của người tạo |

* + - 1. **Bảng User**

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

* + - 1. **Bảng StateHistory**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Nullable | Ràng buộc | Mô tả |
|  | at | Date | No |  | Thời gian thiết bị nhận dữ liệu |
|  | Temperature | Int | Yes |  | Nhiệt độ |
|  | Co2 | Int | Yes |  | Mức độ co2 |
|  | Humidity | Int | Yes |  | Độ ẩm |
|  | dust | Int | Yes |  | Độ bụi |
|  | deviceId | string | no | FK | Id thiết bị nhận |

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

## **4.1. ReactJS**

**React** là một thư viện do Facebook phát triển, React có thể được sử dụng để xây dựng các thành phần UI có tính tương tác cao. React rất dễ dàng phối hợp được với các thư viện Javascript, nó cho phép các [lập trình viên](http://niithanoi.vn/chuong-trinh-dao-tao/380/lap-trinh-vien-quoc-te-niit.html) nhúng code HTML vào code Javascript bằng JSX.

Điểm hay nhất của React là nó không chỉ hoạt động ở phía Client mà còn có thể render phía server và có thể dễ dàng kết nối với nhau.

Ưu điểm:

* Dễ dàng sử dụng, tạo được các component nhẹ.
* API thanh lịch.
* Hỗ trợ cộng đồng lớn.
* Phổ biến trong giới StartUp.
* Rất nhiều tiện ích nguồn mở.

Nhược điểm:

* Điều chỉnh cho JSX.
* Các giải pháp hoàn chỉnh yêu cầu thư viện của bên thứ ba.
* Tính khả dụng của các tùy chọn có thể gây nhầm lẫn.

## **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**](https://jobs.hybrid-technologies.vn/blog/mongodb-la-gi/) 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: I2C, SPI, I²S, UART
* Đặc biệt: 10bit ADC

## **4.5. MQ Telemetry Transport (MQTT)**

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

Description automatically generated

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 các thông tin dạng siêu văn bản như HTML, hoạt động theo mô hình client – server

Graphical user interface, application

Description automatically generated

Được sử dụng rộng rãi trong môi trường web, là phương thức cực kì 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/TruongVanHien194276/ProjectIOT.git>

* 1. Folder Hardware: Open Arduino IDE

Kết nối thiết bị và chạy code hệ thống phần cứng với file Hardware.ino

* 1. Floder Backend\_IOT: Open Terminal

npm install

node server.js

* 1. Folder Frontend\_IOT: Open Terminal

npm install

npm run start

Cổng localhost:8088

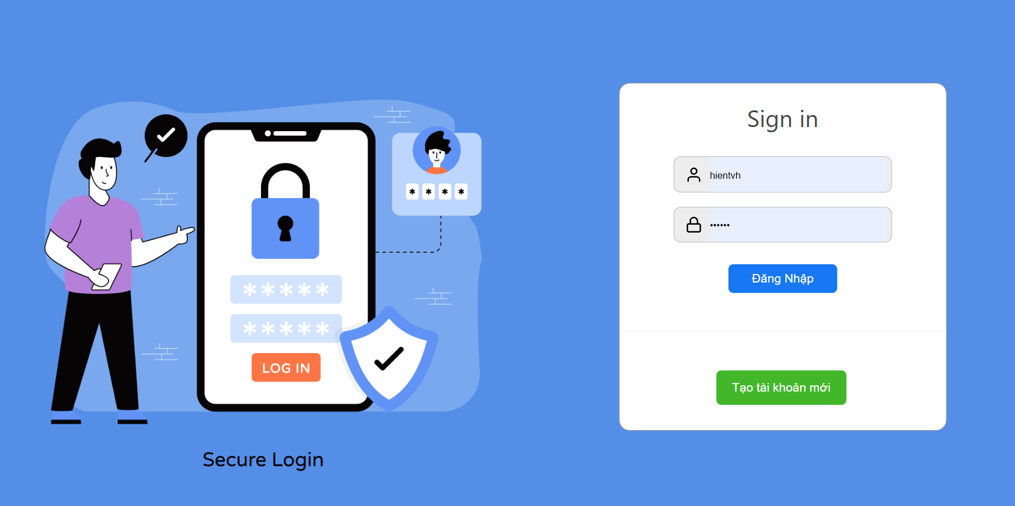
## **5.2. Chương trình minh hoạ**

* Đăng ký

Graphical user interface, application

Description automatically generated

* Đăng nhập



* Giao diện trang chủ

Graphical user interface, application, Word, website

Description automatically generated

* Thông tin cá nhân

Graphical user interface, application, map

Description automatically generated

* Thêm thiết bị

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

* Xem thiết bị

Graphical user interface, application

Description automatically generated

* Đăng xuất

Graphical user interface, application

Description automatically generated

# **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 domain logic bài toán (quan trắc chất lượng không khí) | Cả nhóm |
| Lên kế hoạch triển khai |
| Thiết kế hệ thống | Thiết kế các use case sử dụng | Nguyễn Xuân Mạnh Hùng |
| Thiết kế tổng quan hệ thống  (tổ chức các thành phần và chia luồng hệ thống) | Cả nhóm |
| Thiết kế cơ sở dữ liệu | Hà Trung Kiên |
| Triển khai công nghệ phía phần cứng | Tìm hiểu các thiết bị phần cứng | Cả nhóm |
| Đặt mua các thiết bị | Trương Văn Hiển |
| Code cho cảm biến và ESP |
| Triển khai server | Hosting MQTT broker | Trương Văn Hiển |
| Hosting API server  (cho phía database) | Hà Trung Kiên |
| Triển khai công nghệ phía phần mềm | Tìm hiểu và lựa chọn công nghệ | Nguyễn Xuân Mạnh Hùng  Vũ Ngọc Quang Huy |
| Xây dụng cấu trúc thư mục |
| Code các UI components |
| Triển khai MQTT | Trương Văn Hiển |
| Triển khai HTTP | Hà Trung Kiên |
| Làm slide thuyết trình |  | Hà Trung Kiên |
| Viết báo cáo |  | Trương Văn Hiển  Vũ Ngọc Quang Huy |
| Tổng hợp và đánh giá kết quả | Đánh giá kết quả thực hiện được | Cả nhóm |
| Đánh giá công việc thành viên |

## **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.
* Kĩ năng làm việc nhóm, học nhóm.

## **6.3. Hạn chế**

* Thiết bị cồng kềnh.
* 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 emsẽ 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” – IT3190 – TS. Phạm Ngọc Hưng
2. <https://niithanoi.vn/chi-tiet-tin/1830/react-la-gi-no-co-uu-diem-va-nhuoc-diem-la-gi.html#.ZAeY8y8RqRt>
3. <https://vn.got-it.ai/blog/tim-hieu-ve-nodejs-va-nhung-uu-nhuoc-diem-cua-nodejs>
4. <https://jobs.hybrid-technologies.vn/blog/uu-nhuoc-diem-mongodb/>
5. Arduino Documentation: <https://docs.arduino.cc>
6. Cách kết nối thiết bị với sever thông quan giao thức mqtt: [New MQTT Integration - How to connect IoT Devices using a Third-Party Broker and send Downlinks - YouTube](https://www.youtube.com/watch?v=VoOlvANzUcU)
7. API public của AirVisual API: [AirVisual API (iqair.com)](https://api-docs.iqair.com/)
8. Cách hiển thị vị trí dạng map trên react: [Maps, geocoding, and navigation APIs & SDKs | Mapbox](https://www.mapbox.com/)