TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ────────

**BÀI TẬP LỚN**

IOT VÀ ỨNG DỤNG

**XÂY DỰNG HỆ THỐNG   
GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Mã sinh viên** |
| Nguyễn Trọng Hải | 20183730 |
| Phạm Minh Hiệp | 20183738 |
| Trần Đức Ngọc | 20183806 |
| Ngô Đình Sáng | 20183819 |
|  |  |
| Lớp | : **Kỹ thuật máy tính – Khóa 63** |
| Giảng viên hướng dẫn | : TS. Phạm Ngọc Hưng |

*Hà Nội, tháng 7 năm 202**1*  
MỤC LỤC

[CHƯƠNG 1. MÔ TẢ ĐỀ TÀI 4](#_Toc77229514)

[I. Lý do chọn đề tài 4](#_Toc77229515)

[II. Mô tả bài toán 4](#_Toc77229516)

[1. Mục đích sử dụng 4](#_Toc77229517)

[2. Kịch bản sử dụng 4](#_Toc77229518)

[3. Các đặc điểm quan trọng 4](#_Toc77229519)

[4. Các yêu cầu cần đạt được 5](#_Toc77229520)

[5. Các tác nhân sử dụng hệ thống 5](#_Toc77229521)

[III. Các chức năng và dịch vụ 5](#_Toc77229522)

[IV. Kế hoạch thực hiện 5](#_Toc77229523)

[V. Thành viên nhóm 6](#_Toc77229524)

[CHƯƠNG 2. MÔ TẢ HỆ THỐNG 7](#_Toc77229525)

[I. Mô hình hóa hệ thống 7](#_Toc77229526)

[1. Các giao thức lõi (Core protocols) 7](#_Toc77229527)

[2. Các trạm đo đạc (Measurements) 7](#_Toc77229528)

[3. Trạm giám sát (Monitoring) 8](#_Toc77229529)

[4. Máy chủ và Cơ sở dữ liệu (Server & Database) 8](#_Toc77229530)

[5. Người dùng (Clients) 8](#_Toc77229531)

[CHƯƠNG 3. GIẢI PHÁP CÔNG NGHỆ 9](#_Toc77229532)

[I. Mô hình hóa các công nghệ 9](#_Toc77229533)

[II. Chi tiết hóa hệ thống 10](#_Toc77229534)

[1. Giao thức MQTT và xây dựng MQTT broker 10](#_Toc77229535)

[2. Cảm biến khí gas MQ-135 11](#_Toc77229536)

[3. Vi xử lý ESP32 12](#_Toc77229537)

[4. Website giám sát 14](#_Toc77229538)

[5. Máy chủ NodeJS 15](#_Toc77229539)

[6. Cơ sở dữ liệu MongoDB 16](#_Toc77229540)

[7. Mobile App cho người dùng 17](#_Toc77229541)

[CHƯƠNG 4. KẾT QUẢ THỰC HIỆN 18](#_Toc77229542)

[I. Cảm biến và MQTT broker 18](#_Toc77229543)

[1. Đọc kết quả từ cảm biến 18](#_Toc77229544)

[2. Xây dựng MQTT broker 20](#_Toc77229545)

[II. Website giám sát 21](#_Toc77229546)

[1. Chức năng “Hiển thị nồng độ không khí realtime” 21](#_Toc77229547)

[2. Chức năng “chọn trạm đo” 22](#_Toc77229548)

[3. Chức năng “Cảnh báo chất lượng không khí” 23](#_Toc77229549)

[4. Chức năng thống kê chất lượng không khí trong vòng 10 phút 23](#_Toc77229550)

[III. App cho người dùng 24](#_Toc77229551)

[1. Chức năng “Subscribe more” 25](#_Toc77229552)

[2. Chức năng “Subscribe list” 26](#_Toc77229553)

[3. Chức năng “Historic warning” 28](#_Toc77229554)

[CHƯƠNG 5. KẾT LUẬN 30](#_Toc77229555)

[I. Khó khăn và bài học 30](#_Toc77229556)

[II. Kết luận 30](#_Toc77229557)

[III. Hướng phát triển 30](#_Toc77229558)

[TÀI LIỆU THAM KHẢO 31](#_Toc77229559)

**CHƯƠNG 1. MÔ TẢ ĐỀ TÀI**

## I. Lý do chọn đề tài

Trong không khí có nhiều khi độc hại như: CO, CO2, NOx, Amonia, Acetona, sulfide, hơi benzen, khói, … gây ảnh hưởng tới sức khỏe con người khi hít phải.



Tuy nhiên, những khí này hầu hết đều không màu và tương đối loãng nên khi ta hít phải sẽ rất khó nhận biết nồng độ đã đạt đến ngưỡng nguy hiểm hay chưa?

Chúng ta cần có một hệ thống thông minh để đo đạc chính xác nồng độ các khí và đưa ra cảnh báo tránh tiếp xúc nhiều trong môi trường có chứa nhiều khí độc hại để tránh ảnh hướng nhiều đến sức khỏe.

## II. Mô tả bài toán

### 1. Mục đích sử dụng

Chúng ta muốn biết được nồng độ các khí độc hại trong không khí và xem bản đồ ô nhiễm không khí để tránh đi qua khu vực ô nhiễm.

### 2. Kịch bản sử dụng

Kịch bản 1: Xem bản đồ ô nhiễm không khí để đăng ký thông tin từ các trạm quan sát ô nhiễm không khí gần khu vực muốn chọn

* Kịch bản 2: Xem thông tin real-time từ các địa điểm đã ấn chọn đăng ký và xem các cảnh báo gửi về.
* Kịch bản 3: Xem bản đồ phân tích mức độ ô nhiễm của từng khí đo được.

### 3. Các đặc điểm quan trọng

Hỗ trợ thao tác ngay trên bản đồ của Google.

Hỗ trợ thời gian thực.

Một App cho người dùng và một Web để giám sát chất lượng không khí.

### 4. Các yêu cầu cần đạt được

Trễ nhỏ hơn 0.5s để đảm bảo trong thời gian thực.

App và Web phải chính xác và tương ứng với nhau, không được sai lệch.

### 5. Các tác nhân sử dụng hệ thống

Người giám sát sử dụng Web.

Người dùng sử dụng App.

## III. Các chức năng và dịch vụ

Đăng ký/ Đăng nhập.

Thiết lập thêm trạm đo.

Xem bản đồ nồng độ chất khí và đăng ký trạm đo.

Xem thông tin chi tiết từng khí và độ ô nhiễm.

Xem các cảnh báo ô nhiễm.

## IV. Kế hoạch thực hiện

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Người tham gia** | **Thời điểm**  **bắt đầu** | **Thời điểm**  **kết thúc** |
| Mô tả chi tiết bài toán | Nguyễn Trọng Hải | 08/05/2021 | 13/06/2021 |
| Sử dụng ESP 32 đọc cảm biến MQ-135 | Phạm Minh Hiệp | 12/05/2021 | 18/05/2021 |
| Xây dựng MQTT broker | Phạm Minh Hiệp | 18/05/2021 | 24/05/2021 |
| Xây dựng Web | Trần Đức Ngọc | 12/05/2021 | 13/06/2021 |
| Xây dựng App | Nguyễn Trọng Hải  Ngô Đình Sáng | 01/06/2021 | 15/06/2021 |
| Demo Web + Đo đạc | Phạm Minh Hiệp  Trần Đức Ngọc | 13/06/2021 | 15/06/2021 |
| Demo App | Nguyễn Trọng Hải  Ngô Đình Sáng | 13/06/2021 | 15/06/2021 |
| Làm Báo cáo | Nguyễn Trọng Hải  Ngô Đình Sáng  Trần Đức Ngọc | 01/06/2021 | 15/06/2021 |
| Làm Slide | Nguyễn Trọng Hải | 18/05/2021 | 15/06/2021 |

## V. Thành viên nhóm

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Email** |
| Nguyễn Trọng Hải | 20183730 | [hai.nt183730@sis.hust.edu.vn](mailto:hai.nt183730@sis.hust.edu.vn) |
| Phạm Minh Hiệp | 20183738 | [hiep.pm183738@sis.hust.edu.vn](mailto:hiep.pm183738@sis.hust.edu.vn) |
| Trần Đức Ngọc | 20183806 | [ngoc.td183806@sis.hust.edu.vn](mailto:ngoc.td183806@sis.hust.edu.vn) |
| Ngô Đình Sáng | 20183819 | [sang.nd183819@sis.hust.edu.vn](mailto:sang.nd183819@sis.hust.edu.vn) |

**CHƯƠNG 2. MÔ TẢ HỆ THỐNG**

## I. Mô hình hóa hệ thống

Hệ thống gồm 5 thành phần chính như hình bên dưới:

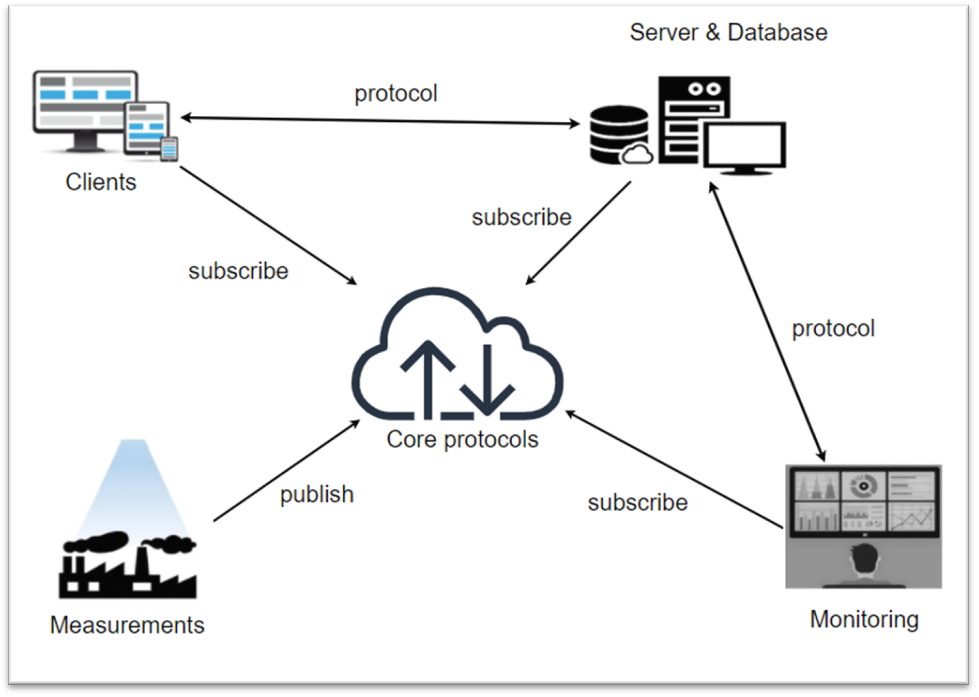
+ Các giao thức lõi (Core protocols).

+ Các trạm đo đạc (Measurements).

+ Trạm giám sát (Monitoring).

+ Máy chủ và Cơ sở dữ liệu (Server & Database).

+ Người dùng (Clients).



### 1. Các giao thức lõi (Core protocols)

Giao thức nằm ở trung tâm của hệ thống, là hạt nhân quan trọng nhất trong hệ thống. Các giao thức lõi có vai trò truyền các thông điệp, tín hiệu nhận được từ trạm đo đến với các phần còn lại trong hệ thống (nhứng thành phần ấy đã đăng ký với các giao thức lõi từ trước).

### 2. Các trạm đo đạc (Measurements)

Các trạm đo được đặt tại những khu vực khác nhau (mỗi khu vực một trạm) để lấy thông số nồng độ các chất khí có trong không khí của khu vực đó. Những thông số đo được sẽ được đưa qua một cảm biến có kết nối Wifi để truyền tín hiệu, thông điệp qua các giao thức lõi đến các thành phần khác có trong hệ thống.

### 3. Trạm giám sát (Monitoring)

Trạm quan sát nhận dữ liệu thời gian thực của tất cả các khu vực qua các giao thức lõi để hiển thị những thông số trong thời gian thực.

Trạm quan sát truy vấn dữ liệu qua server để hiển thị những biều đồ phân tích nồng độ các khí có trong không khí của từng khu vực.

### 4. Máy chủ và Cơ sở dữ liệu (Server & Database)

Trạm quan sát nhận dữ liệu thời gian thực của tất cả các khu vực qua các giao thức lõi để xử lý và lưu những thông tin cần thiết vào Cơ sở dữ liệu.

Xử lý những yêu cầu truy vấn của trạm quan sát và người dùng khi cần thiết.

### 5. Người dùng (Clients)

Người dùng nhận dữ liệu thời gian thực của những khu vực mà đã được đăng ký qua các giao thức lõi để hiển thị những thống số trong thời gian thực.

Người dùng truy vấn dữ liệu qua server để hiển thị những biều đồ phân tích nồng độ các khí có trong không khí của từng khu vực.

**CHƯƠNG 3.** **GIẢI PHÁP CÔNG NGHỆ**

## I. Mô hình hóa các công nghệ

Hệ thống gồm 7 thành phần chính như hình bên dưới:

+ Giao thức lõi là MQTT.

+ Cảm biến khí gas MQ-135.

+ Vi xử lý ESP32.

+ Website giám sát.

+ Máy chủ NodeJS.

+ Cơ sở dữ liệu MongoDB.

+ Mobile App cho người dùng.



## II. Chi tiết hóa hệ thống

### 1. Giao thức MQTT và xây dựng MQTT broker

#### 1.1 Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp/thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt [2].

Sử dụng giao thức MQTT cho bài toán này là bởi vì đây là một bài toán không đòi hỏi xử lý ở cường độ cao, những tính toán khá đơn giản và lượng dữ liệu không lớn.

#### 1.2 Xây dựng MQTT broker

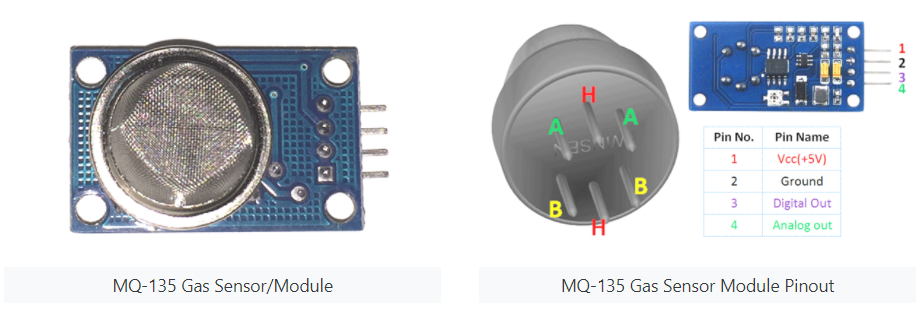
Ban đầu, nhóm dự định xây dựng riêng 1 broker. Tuy nhiên hiệu quả của việc sử dụng broker này chưa được cao, độ trễ lớn. Việc publish/subcribe data phục vụ cho nhu cầu của nhóm cũng khá đơn giản, cho nên nhóm đã quyết định dùng brokder có sẵn.

Broker có sẵn: EMQX Broker

Link tham khảo : https://www.emqx.io/

2. Cảm biến khí gas MQ-135

#### 2.1 Cảm biến khí gas MQ-135



Cảm biến MQ-135 có độ nhạy cao đối với các chất khí như CO, CO2, NOx amonia, acetona, …, cho biết sự thay đổi của tỉ số (RS/R0) (trong đó RS là biến trở của sensor, R0 là điện trở khi không khí trong lành). Tỉ số này là nguyên liệu đầu vào quan trọng để vi xử lý tính toán ra nồng độ từng khí theo công thức cho trước.

#### 2.2 Cách đo

Công thức tính RS



Trong đó, VC là điện áp đầu vào.

VRL là điện áp 2 đầu cuộn dây (gồm R và L).

RL là điện trở giữa 2 đầu cuộn dây.

Dùng tỉ số RS/R0 kết hợp với tính toán để suy ra nồng độ từng khí [3]

Trong đó,

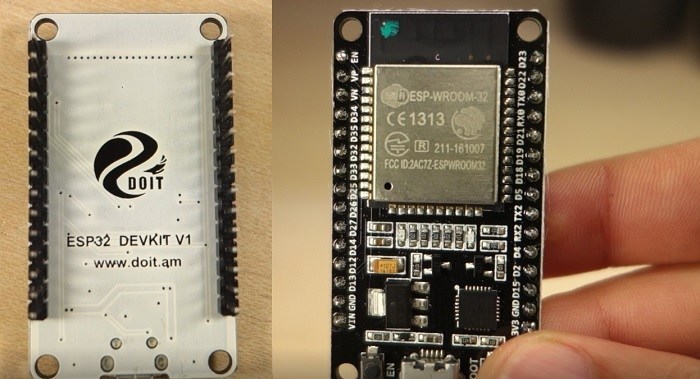
Ro: điện trở của cảm biến ở 100ppm của NH3 trong không khí sạch.

Rs: điện trở của cảm biến ở các nồng độ khí khác nhau.

### 3. Vi xử lý ESP32

#### 3.1 Xi xử lý ESP32

ESP32 là thế hệ tiếp theo của ESP8266, có cải tiến là dual-core và khả năng kết nối Wi-fi, Bluetooth [4].



ESP32 là thế hệ tiếp theo của ESP8266, có cải tiến là dual-core và khả năng kết nối Wi-fi, Bluetooth.

Thông số ESP32 DEVKIT DOIT board

+ Core: dual-core

+ Wifi: 2.4GHz to 150Mbits/s

+ Bluetooth: BLE (Bluetooh Low Energy) và thừa kế đặc trưng Bluetooth

+ Kiến trúc: 32bit

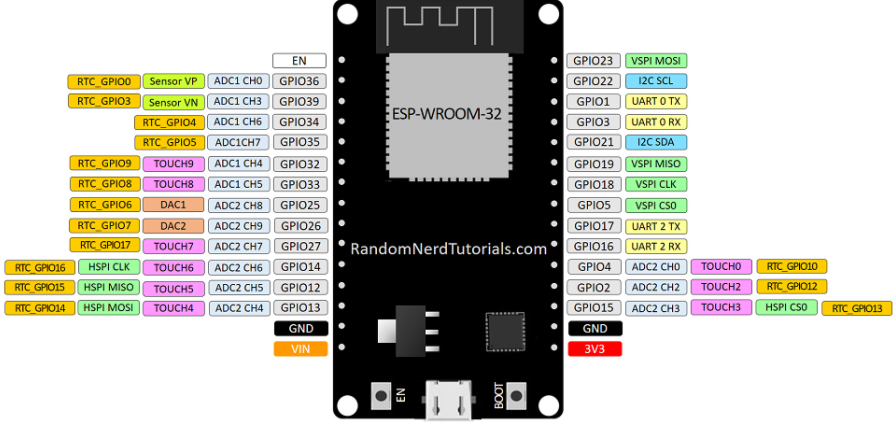
+ Xung lên tới 240 MHz

+ RAM: 512kB

+ Pins: 30 – 36 tùy mẫu

+ Mức logic là 3.3V

+ Nhiều kết nối ngoại vi



#### 3.2 Tính toán dữ liệu từ sensor

Vi xử lý ESP32 nhận dữ liệu chỉ số RS/R0 từ cảm biến khí gas MQ-135 để đưa ra tính toán chỉ số PPM (parts per million – tỉ lệ phần triệu) cho từng chất khí.

Nhờ vào khả năng kết nối Wifi, ESP32 có thể dễ dàng kết nối đến MQTT broker đã được dựng sẵn để đưa dữ liệu thời gian thực về cho các thành phần đã đăng ký như Máy chủ, người dùng và Trạm quan sát.

### 4. Website giám sát

#### 4.1 ReactJS

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

#### 4.2 Rechart

Thư viện cho phép biểu diễn dữ liệu dưới dạng biểu đồ

#### 4.3 React-Leaftlet

Leaflet là một thư viện JavaScript mã nguồn mở cho việc xây dựng một ứng dụng map có tính tương tác. Đây là một thư viện khá nhẹ, chỉ khoảng 38KB cho phần script nhưng lại có đầy đủ tất cả các tính năng mà hầu hết các developer cần.

Leaflet được thiết kế chú trọng tới sự đơn giản, hiệu suất, và khả năng sử dụng. Nó cũng hoạt động hiệu quả trên cả nền tảng máy tính lẫn mobile, có thể được mở rộng với cả tá plugin, ngoài ra nó còn có một trang document API khá đẹp, đơn giản nhưng dễ đọc.

#### 4.4 Mqtt Client

Thư viện cho phép subscribe vào broker để theo dõi dữ liệu một cách realtime phía client

#### 4.5 Material UI

Material UI đem đến cho bạn và trang web của bạn một giao diện hoàn toàn mới, với những button, textfield, toogle... được design theo một phong cách mới lạ, thay vì việc nhà nhà người người dùng Bootstrap như hiện nay.

### 5. Máy chủ NodeJS

#### 5.1. ExpressJS

Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile

Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.

Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.

Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function

#### 5.2. ScheduleJob

Do yêu cầu nhiều trạm đo nhưng chỉ có 1 sensor vật lý, dùng ScheduleJob để fake sensor publish data lên mqtt broker.

### 6. Cơ sở dữ liệu MongoDB

Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB và Node.js. Nó quản lý mối quan hệ giữa dữ liệu, cung cấp sự xác nhận giản đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn các đối tượng trong MongoDB.

MongoDB là một cơ sở dữ liệu NoSQL. Bạn có thể lưu trữ các JSON trong đó, và cấu trúc của các tài liệu này có thể thay đổi vì nó không bắt buộc như các cơ sở dữ liệu SQL. Đây là một trong những lợi thế của việc sử dụng NoSQL vì nó tăng tốc độ phát triển ứng dụng và giảm sự phức tạp của việc triển khai.

### 7. Mobile App cho người dùng

#### 7.1 Các công nghệ sử dụng trong App

Sử dụng Android Studio IDE để lập trình App. Để đảm bảo dữ liệu thời gian thực, App có sử dụng AsynTask [9] để chia luồng thực thi và luồng hiển thị. App sử dụng Google API [10] để lấy địa điểm thực tế các trạm đo, cho phép người dùng tương tác và search ngay trên bản đồ. PahoMQTT[8] được sử dụng để App đăng ký dữ liệu đi qua MQTT broker.

#### 7.2 Các chức năng trong App

Tìm kiếm, tương tác với bản đồ để tìm trạm đo gần nhất mới địa điểm người dùng mong muốn, người dùng có thể đăng ký hoặc hủy đăng ký trạm đo này.

Người dùng có thể xem danh sách những trạm đo mình đã đăng ký và nhấn vào để xem chi tiết các thông số tại một trạm đo nào đó.

Người dùng có thể xem các cảnh báo được gửi về cho mình.

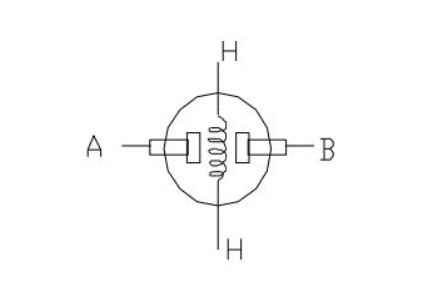
# CHƯƠNG 4. KẾT QUẢ THỰC HIỆN

## I. Cảm biến và MQTT broker

### 1. Đọc kết quả từ cảm biến

#### 1.1 Mô tả cảm biến MQ-135

Cảm biến được tích hợp một biến trở có giá trị thay đổi theo nồng độ các chất khí. Nếu nồng độ cao, giá trị của biến trở sẽ giảm. Nếu nồng độ thấp, giá trị của biến trở sẽ tăng.



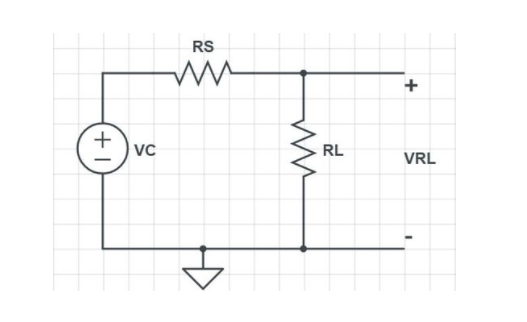
Giữa A và B có tích hợp 1 biến trở, thay đổi tùy thuộc vào nồng độ khí. Giữa H và H có lắp sẵn 1 điện trở nhiệt.

Nồng độ của 1 khí được tính theo phần triệu (ppm) theo tỉ lệ điện trở của cảm biến (RS/R0). Trong đó, RS là điện trở của cảm biến thay đổi tùy thuộc nồng độ khí và R0 là điện trở của cảm biến ở một nồng độ đã biết mà không có sự hiện diện của các khí khác hoặc trong điều kiện không khí trong lành.

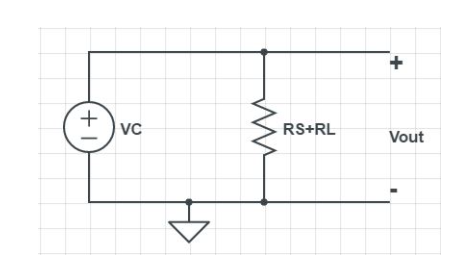
*Tính toán*

Từ đinh luật Ôm, ta có : I = V / R

Với V là điện áp, I là dòng điện và R là điện trở. Đơn giản hóa mạch, ta có hình dưới đây:

**

Kết hợp 2 điện trở trong chuỗi (RS và RL kết hợp)

**

Giá trị của dòng điện trong mạch: I = VC / (RS + RL)

VRL = [VC / (RS + RL)] \* RL

Biến đổi công thức, ta tính được RS = [(VC \* RL) / VRL] – RL

Từ công thức trên, ta có thể tính được giá trị điện trở của cảm biến đối với các loại khí khác nhau.

#### 1.2 Material UI

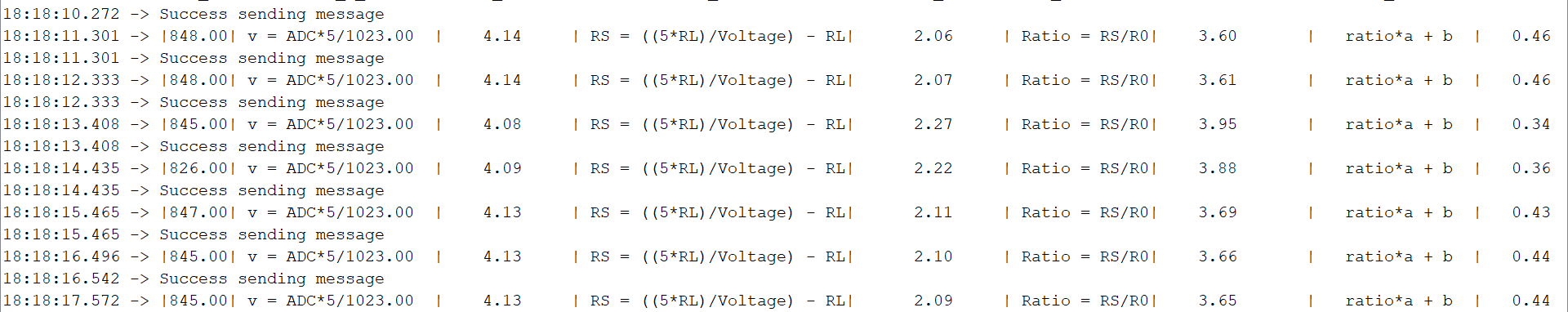
Thực hiện đo nồng độ của các chất khí sau: CO, CO2, NH4, Acetona.

Công thức tính giá trị nồng độ của các chất khí (ppm): a \* ratio^ b

Với a và b là các hằng số có giá trị trong bảng dưới đây:

|  |  |  |
| --- | --- | --- |
| Khí | a | b |
| CO | 605.18 | -3.937 |
| CO2 | 110.47 | -2.862 |
| NH4 | 102.2 | -2.473 |
| Acetona | 34.668 | -3.369 |

Đọc tín hiệu analog từ chân số 33 của ESP32 để lấy điện áp vào cho Sensor MQ135. Từ đó tính được các giá trị điện trở. Kết quả thực nghiệm được biểu diễn bởi hình dưới đây:



### 2. Xây dựng MQTT broker

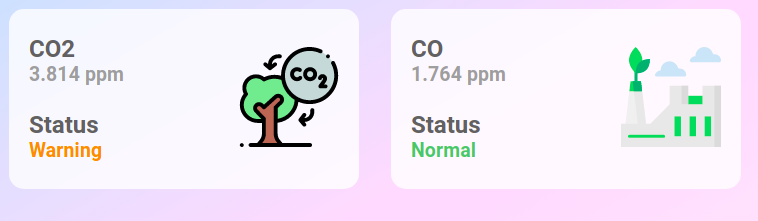
Sử dụng Broker có sẵn: EMQX Broker

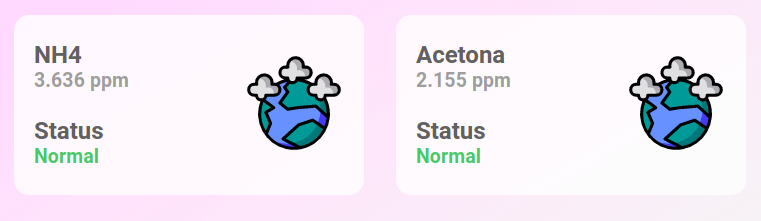
Link tham khảo: https://www.emqx.io

Data thu được từ sensor MQ135 được đóng gói dưới dạng JSON và publish lên broker. Client subcribe vào topic chỉ định để lấy data và visualize real time data vừa lấy được.

## II. Website giám sát

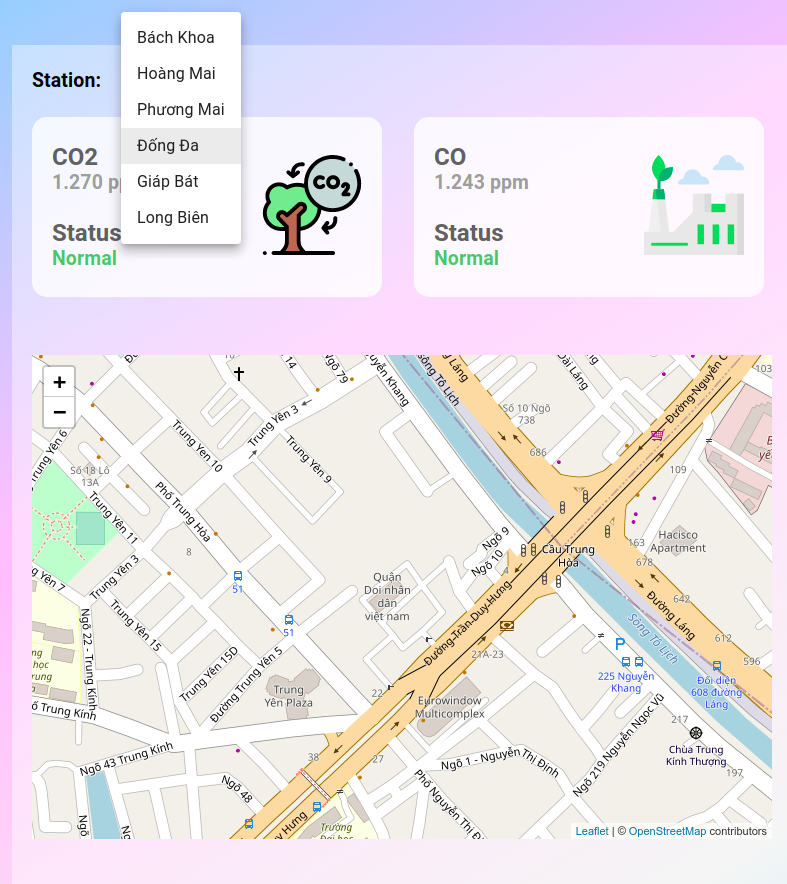
### 1. Chức năng “Hiển thị nồng độ không khí realtime”





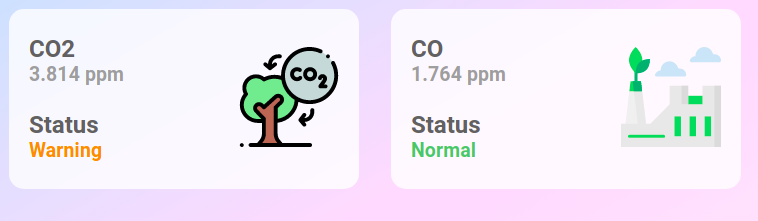
Khi truy cập vào web, người dùng sẽ nhận được thông tin chất lượng không khí gồm các khí CO2, CO, NH4, Acetona. Mặc định trạm đo (station) là ở Hoàng Mai (cảm biến thật).

### 2. Chức năng “chọn trạm đo”



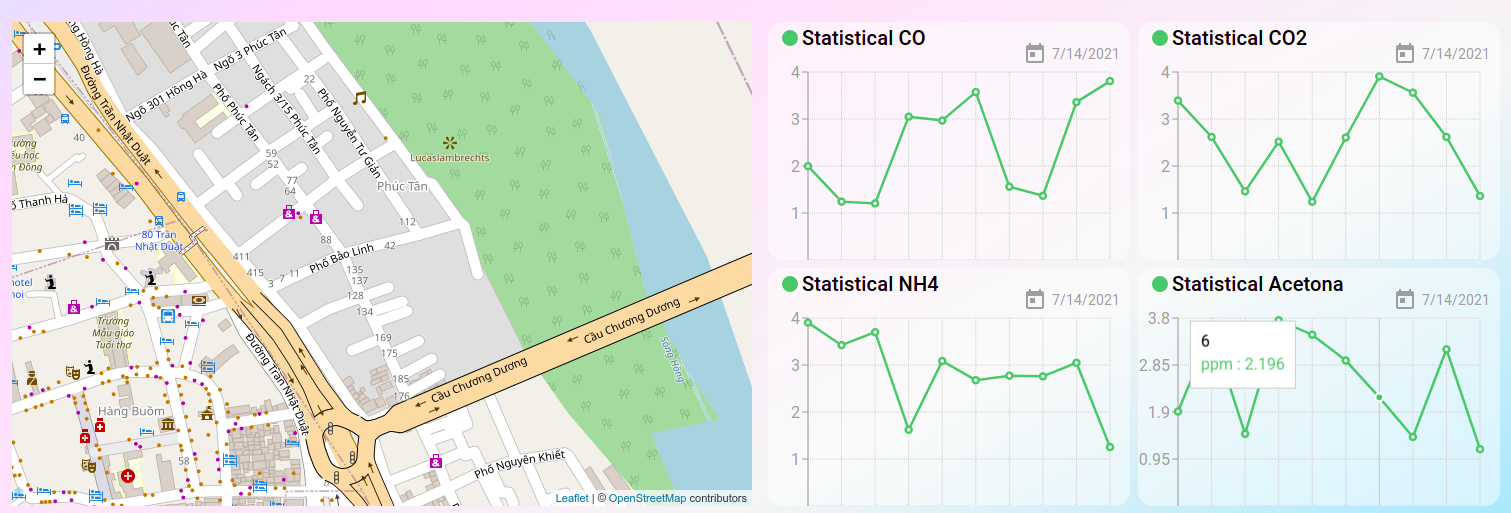
Bên cạnh trạm đo Hoàng Mai (sử dụng dữ liệu từ cảm biến thật). Trang web còn cho phép người dùng lựa chọn các trạm đo khác như Bách Khoa, Phương Mai, Đống Đa, Giáp Bát...

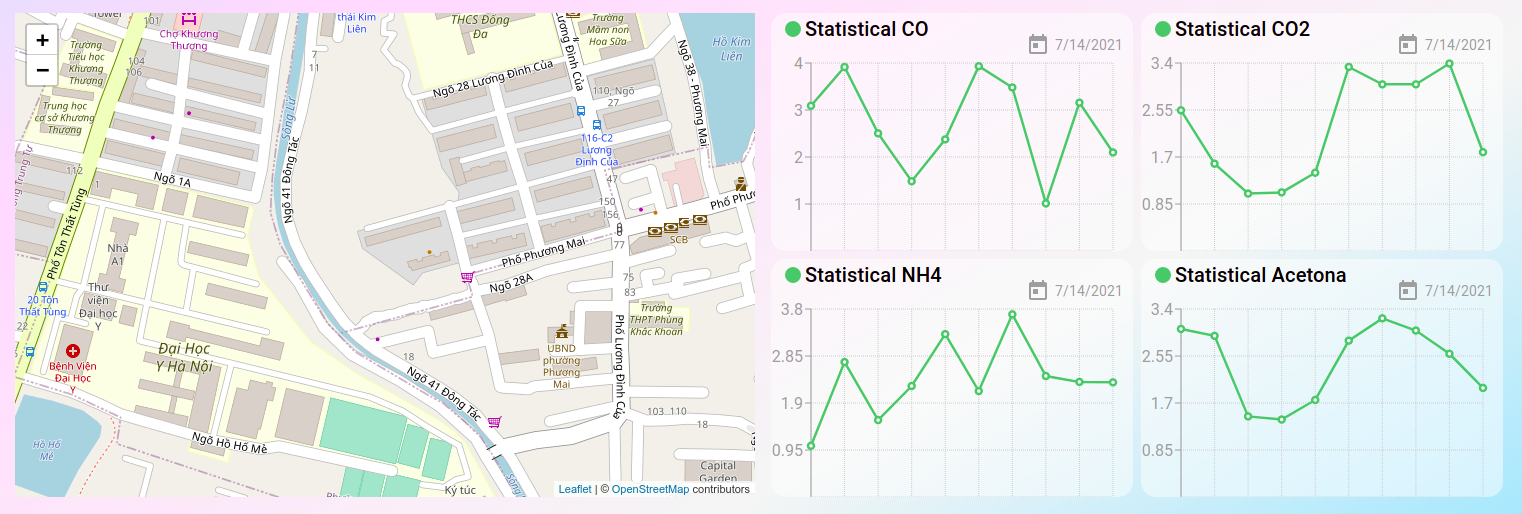
### 3. Chức năng “Cảnh báo chất lượng không khí”



Bên cạnh việc đo đạc chất lượng không khí, trang web cũng đưa ra cảnh báo tương ứng với số liệu đo đạc được. Có 3 trạng thái: Normal, Warning và Dangerous.

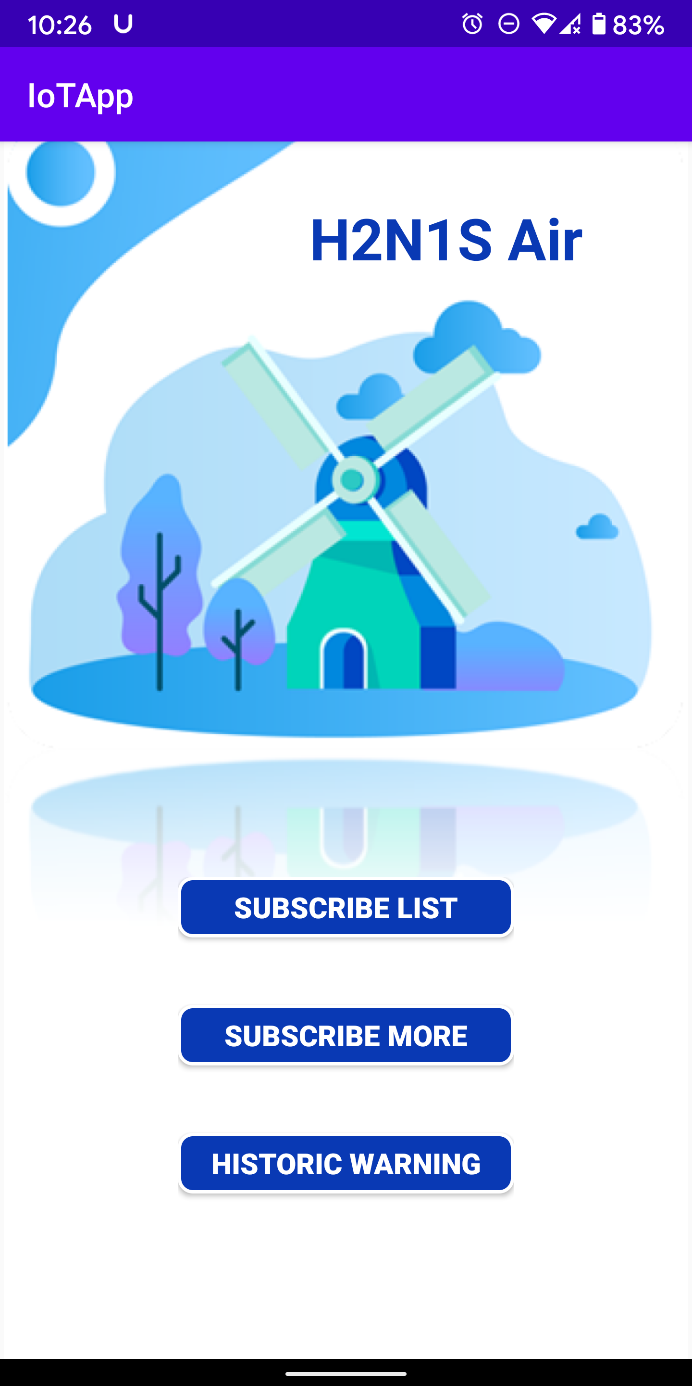
### 4. Chức năng thống kê chất lượng không khí trong vòng 10 phút





Cứ 1 phút trang web ghi lại thông tin của chất lượng không khí đo đạc được từ đó đưa ra thống kê về sự biến đổi của chất lượng không khí

**III. App cho người dùng**



Tìm kiếm, tương tác với bản đồ để tìm trạm đo gần nhất mới địa điểm người dùng mong muốn, người dùng có thể đăng ký hoặc hủy đăng ký trạm đo này.

Người dùng có thể xem danh sách những trạm đo mình đã đăng ký và nhấn vào để xem chi tiết các thông số tại một trạm đo nào đó.

Người dùng có thể xem các cảnh báo được gửi về cho mình.

**1. Chức năng “Subscribe more”**

|  |  |
| --- | --- |
| Không có mô tả. |  |

Khi chọn chức năng “**Subscride more**”, bản đồ Google Map hiện ra với 4 điểm màu đỏ như hình bên trái, đó là 4 điểm đặt các trạm đo. Người dùng có thể ấn vào để xem tên trạm đo, ví dụ như là trạm “**Bách Khoa Hà Nội**”.

Khi chưa tìm kiếm hay tương tác gì với bản đồ thì sẽ có gợi ý “**Choose the place nearest your location, you can subcribe it!**”

Khi người dùng tìm kiếm hoặc tương tác chạm trên bản đồ, ví dụ là tìm kiếm địa điểm “Quỳnh Lôi”, hệ thống sẽ thông báo điểm gần nhất để người dùng có thể đăng ký, ví dụ như trong hình là “**The place nearest this location is Bạch Mai, you can subcribe it**”.

Người dùng chọn “**Subcribe/Unsubscribe**” để Đăng ký/Hủy đăng ký địa điểm được chọn.

Nếu không có địa điểm nào gần với nơi được chọn sẽ có thông báo “**Sorry! There is no place near the location you’ve chosen**”.

### 2. Chức năng “Subscribe list”

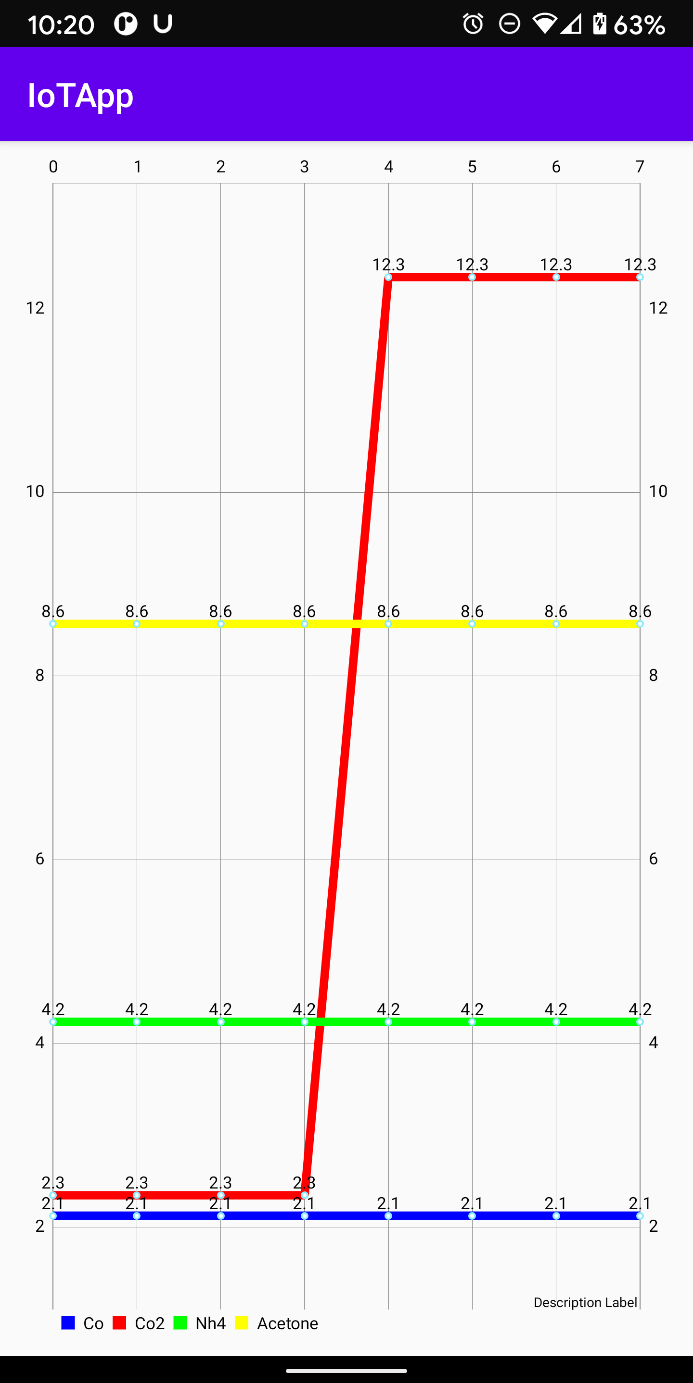
|  |  |
| --- | --- |
|  | Không có mô tả. |

Khi chọn chức năng “**Subscrided list**”, danh sách các địa điểm mà người dùng đã đăng ký sẽ hiện ra như hình bên trái. Người dùng trượt để tìm kiếm địa điểm muốn xem thông số trong thời gian thực Khi lựa chọn một địa điểm nào đó, ví dụ như “**Lăng Bác**” sẽ hiện ra giao diện như hình bên phải.

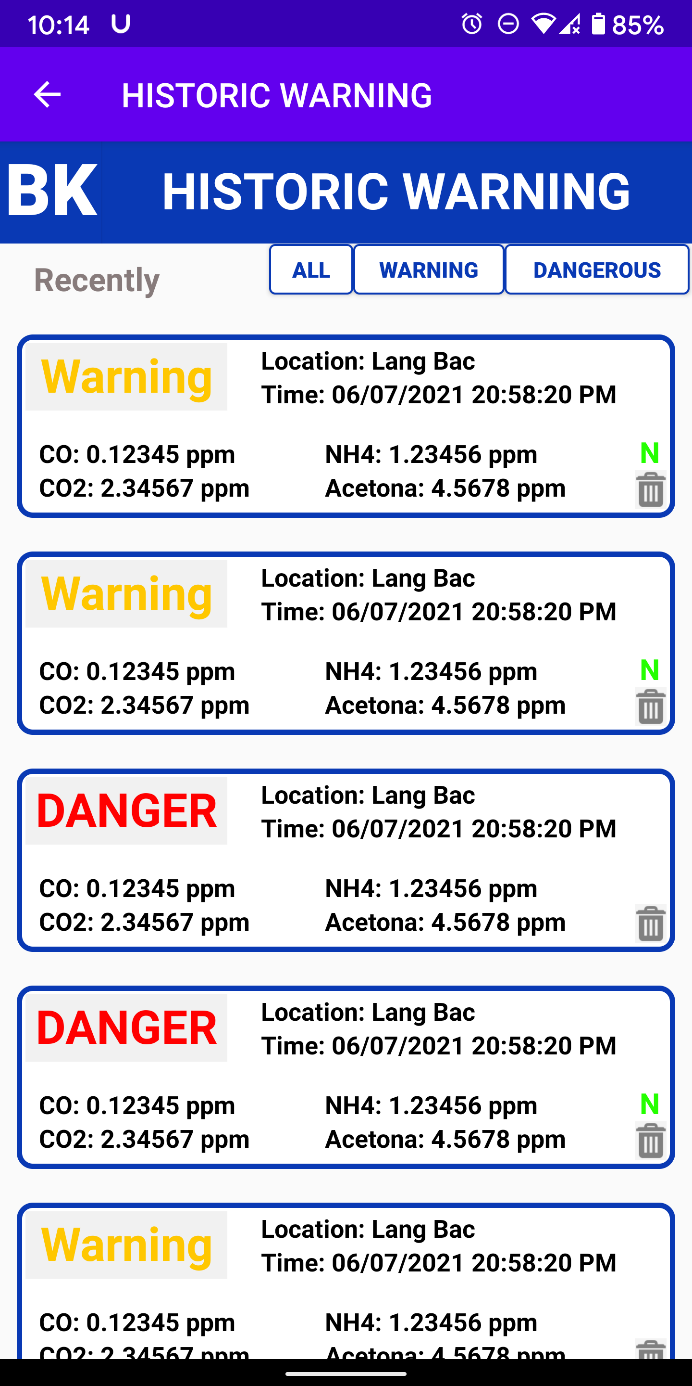
Giao diện “**Air Quality**” hiện lên để người dùng có thể xem chi tiết từng nồng độ chất khí tại địa điểm đã chọn trong thời gian thực. Như trong hình là 4 loại khí CO, CO2, NH4, Acetona, tương ứng với mỗi khí là chỉ số PPM và trạng thái (gồm 3 trạng thái Normal, Warning và Dangerous).

Người dùng chọn “**Subcribe/Unsubscribe**” để Đăng ký/Hủy đăng ký địa điểm được chọn.

Bên cạnh đó, người dùng có thể chọn “**Hictoric air quality graph**” để xem biểu đồ các chất khí theo các khoảng thời gian liên tục theo thời gian.



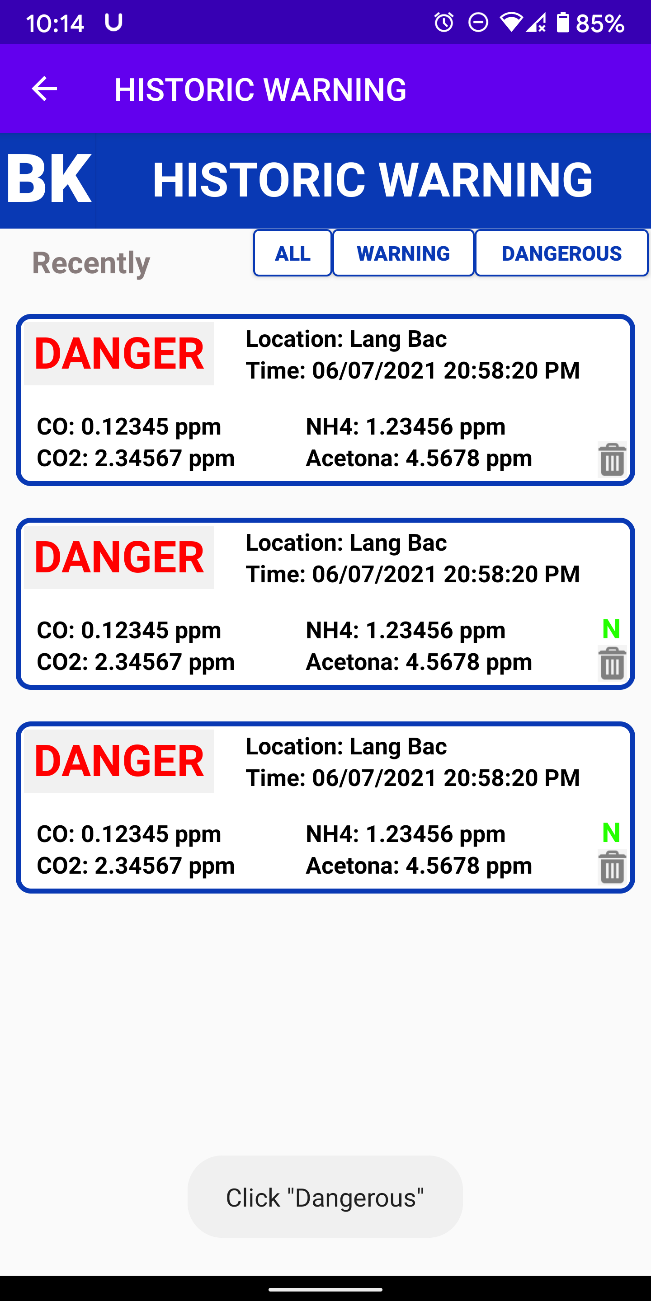
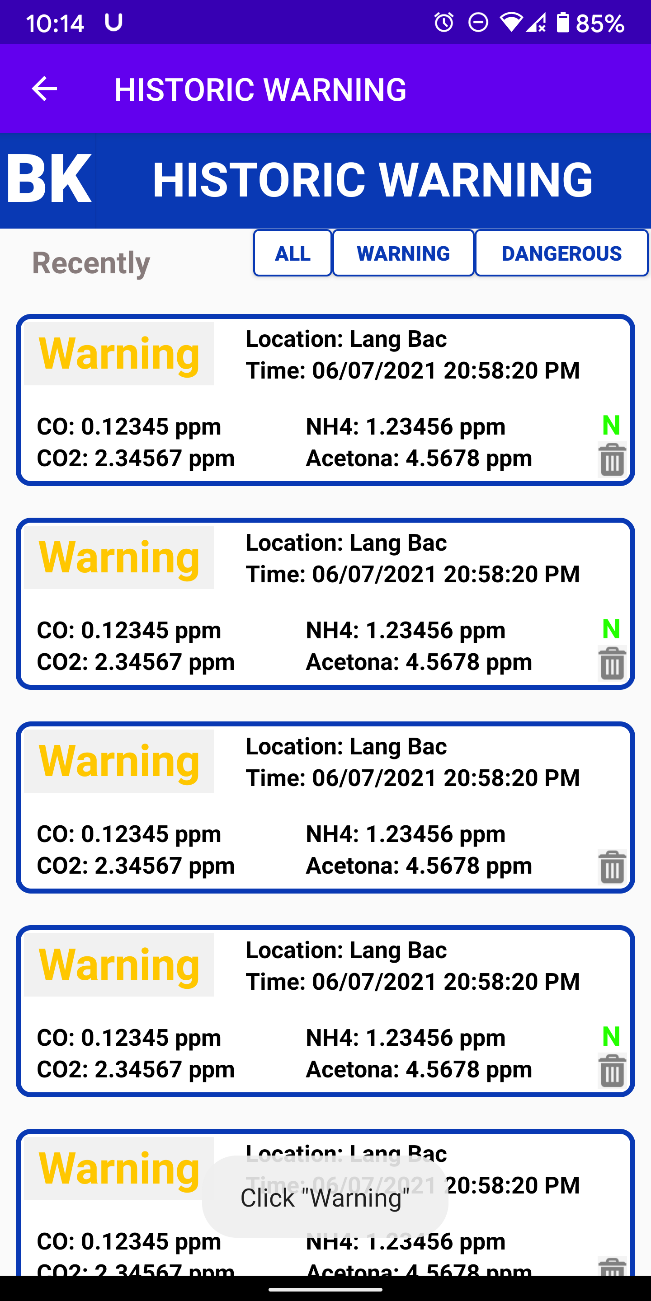
### 3. Chức năng “Historic warning”



Khi chọn chức năng “**Historic warning**”, danh sách các cảnh báo sẽ được hiện ra theo thứ tự càng gần đây thì càng ở trên.

Mỗi cảnh bảo đều có trạng thái (Warning hoặc Dangerous), có địa điểm, thời gian, nồng độ 4 chất và một icon xóa ở góc dưới bên phải để người dùng muốn xóa nếu không muốn nhìn thấy nữa. Bên cạnh đó, một số cảnh báo trên icon xóa còn có chữ “N” thể hiện là còn mới, chưa được xem, người dùng muốn xóa chữ “N” chỉ cần nhấp vào xem Warning là được.

Bên cạnh đó, phần bên cạnh chữ “**Recently**” còn có 3 lựa chọn “**All**”, “**Warning**”, “**Dangerous**” để lọc theo từng trạng thái (Warning hoặc Dangerous).



Khi chọn “Warning” hoặc “Dangerous” các trạng thái tương ứng sẽ được lọc riêng ra như 2 hình bên trên.

# CHƯƠNG 5. KẾT LUẬN

## I. Khó khăn và bài học

Do dịch bệnh Covid-19 vẫn đang diễn ra hết sức phức tạp nên việc gặp mặt trực tiếp các thành viên là rất khó khăn (chỉ được khoảng 2, 3 lần) nên hầu hết những buổi trao đổi đều diễn ra Online thông qua mạng xã hội Facebook và phần mềm MS Teams. May mắn là cảm biến MQ-135 và ESP32 đã chuẩn bị từ trước và gửi cho các bạn vẫn đang ở Hà Nội.

Cảm biến và ESP32 không mất quá nhiều thời gian để tìm hiểu nhưng việc xây dựng Web, App, Database và Server mất rất nhiều thời gian. Nhóm đã chia ra mỗi người làm một phần riêng nhưng phải có sự hỗ trợ lần nhau mới có thể hoàn thành, một người làm độc lập thực sự rất khó.

Hệ thống gồm nhiều thành phần đòi hỏi tính liên kết và đồng bộ cao nên quá trình đồng bộ đòi hỏi các thành viên phải liên hệ với nhau rất nhiều để kiểm tra và sửa lỗi.

## II. Kết luận

Khoảng 10 tuần học lý thuyết đã trang bị cho cả 4 thành viên những kiến thức cơ bản nhất về hệ thống IoT và các thành phần cơ bản trong hệ thống. Nhìn chung, cả nhóm đều nắm được kiến thức cơ bản và có khả năng vận dụng vào bài tập lớn. Nhóm chia mỗi người phụ trách 1,2 thành phần trong hệ thống để tự phát triển rồi sau đó đồng bộ với nhau.

Cảm biến đo và dữ liệu được ESP32 đẩy về MQTT broker liên tục và ổn định nhưng chưa thử trong thời gian lâu dài. MQTT broker hoạt động nhanh và chính xác, không gặp lỗi trong quá trình sử dụng.

Website và Android App được hoàn thành với các chức năng thực hiện tốt yêu cầu đề ra. NodeJS Server và Database MongoDB chạy vẫn rất ổn định.

## III. Hướng phát triển

Đầu tiên, nhóm cần hoàn thiện những chức năng còn thiếu, chưa hoàn thiện. Đây là công việc cần thiết nhất vì thời gian làm bài tập lớn không đủ nhiều để hoàn thiện tất cả các chức năng. Các chức năng đến hiện tại là tối giản để đảm bảo minh họa những dịnh vụ quan trọng nhất mà người dùng cần.

Triển khai App trên GooglePlay, Webside trên tên miền thật để đánh giá kiểm tra, đánh giá các lỗi để tiến hành sửa chữa.

Lắp đặt hệ thống trong thực tế để cảnh báo ô nhiễm không khí.

**TÀI LIỆU THAM KHẢO**

[1] Bài giảng học phần IoT và Ứng dụng – TS. Phạm Ngọc Hưng.

[2] Giao thức MQTT:

<https://viblo.asia/p/mqtt-la-gi-vai-tro-cua-mqtt-trong-iot-V3m5WL3bKO7>

[3] Cảm biến MQ-135:

<https://components101.com/sensors/mq135-gas-sensor-for-air-quality>

[4] Vi xử lý ESP32:

<https://randomnerdtutorials.com/getting-started-with-esp32/>

[8] Paho MQTT cho Android:

<https://github.com/eclipse/paho.mqtt.android>

[9] AsynTask:

<https://developer.android.com/reference/android/os/AsyncTask>

[10] Search Map, Google Map API:

<https://www.youtube.com/watch?v=iWYsBDCGhGw>

[11] Search Map, Google Map API:

<https://www.youtube.com/watch?v=iWYsBDCGhGw>