

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG ĐIỆN – ĐIỆN TỬ**



**BÁO CÁO BÀI TẬP LỚN**  
**LẬP TRÌNH ỨNG DỤNG DI ĐỘNG**

**ĐỀ TÀI:**  
**HỆ THỐNG IOT QUẢN LÝ CÁC THIẾT BỊ TRONG NHÀ**  
**Mã lớp : 133395**

**Sinh viên thực hiện:**

**Mã số sinh viên**

Nguyễn Đình Cảnh

20182383

Nguyễn Đức Lâm

20182626

Nguyễn Long Thủy

20182814

**Giảng viên hướng dẫn: ThS. Vũ Song Tùng**

**Hà Nội , 7/2022**

# MỤC LỤC

<b>MỤC LỤC.....</b>	<b>1</b>
<b>DANH SÁCH HÌNH VẼ.....</b>	<b>3</b>
<b>LỜI MỞ ĐẦU .....</b>	<b>5</b>
<b>CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI .....</b>	<b>6</b>
<b>1.1 Hệ thống Internet of thing.....</b>	<b>6</b>
1.1.1 Khái niệm về IoT .....	6
1.1.2 Cấu trúc hệ thống IoT .....	7
1.1.3 Các dự án IoT phổ biến hiện nay .....	7
<b>1.2 Đặt vấn đề.....</b>	<b>8</b>
<b>1.3 Các phần mềm , công cụ hỗ trợ.....</b>	<b>9</b>
1.3.1 Ngôn ngữ Java .....	9
1.3.2 Android Studio .....	9
1.3.3 Google Firebase .....	10
1.3.4 Aduino IDE.....	11
<b>1.4 Kết luận .....</b>	<b>12</b>
<b>CHƯƠNG 2. THIẾT KẾ HỆ THỐNG.....</b>	<b>13</b>
<b>2.1 Xây dựng yêu cầu hệ thống.....</b>	<b>13</b>
2.1.1 Yêu cầu chức năng .....	13
2.1.2 Yêu cầu phi chức năng.....	13
<b>2.2 Xác định các Actor.....</b>	<b>14</b>
2.2.1 Người dùng (User) .....	14
2.2.2 Admin.....	14
<b>2.3 Use Case Diagram.....</b>	<b>14</b>
<b>2.4 Class Diagram.....</b>	<b>15</b>
<b>2.5 Activity Diagram .....</b>	<b>16</b>
2.5.1 Đăng nhập .....	17
2.5.2 Điều khiển thiết bị.....	18
2.5.3 Cảnh báo cháy.....	19
2.5.4 Xem nhiệt độ - độ ẩm.....	20

2.5.5 Thêm mới , xóa thiết bị .....	21
2.5.6 Lập báo cáo thống kê .....	22
2.5.7 Quản lý người dùng .....	23
<b>2.6 Sequence Diagram .....</b>	<b>23</b>
2.6.1 Đăng nhập .....	24
2.6.2 Điều khiển thiết bị.....	25
2.6.3 Cảnh báo cháy.....	25
2.6.4 Xem nhiệt độ - độ ẩm.....	26
2.6.5 Thêm mới , xóa thiết bị.....	27
2.6.6 Lập báo cáo thống kê .....	28
2.6.7 Quản lý người dùng .....	29
<b>2.7 Kết luận .....</b>	<b>30</b>
<b>CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ THỰC NGHIỆM.....</b>	<b>30</b>
<b>3.1 Phần mềm.....</b>	<b>30</b>
3.1.1 Yêu cầu .....	30
3.1.2 Real-time Database .....	30
3.1.3 Giao diện phần mềm .....	30
<b>3.2 Phần cứng.....</b>	<b>33</b>
3.2.1 ESP8266.....	33
3.2.2 Cảm biến nhiệt độ, độ ẩm .....	33
3.2.3 Cảm biến ánh sáng .....	34
3.2.4 Cảm biến khí gas.....	35
<b>3.3 Kiểm tra trên bo mạch trắng .....</b>	<b>37</b>
<b>KẾT LUẬN.....</b>	<b>38</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>39</b>
<b>SOURCE CODE .....</b>	<b>40</b>

# DANH SÁCH HÌNH VẼ

Hình 1.1 Cấu trúc hệ thống IoT .....	7
Hình 1.2 Giao diện lập trình Android Studio.....	10
Hình 1.3 Firebase realtime database.....	11
Hình 1.4 Giao diện Aduino IDE .....	12
Hình 2.1 Use Case Diagram .....	15
Hình 2.2 Class Diagram .....	16
Hình 2.3 Activity Diagram - Đăng nhập.....	17
Hình 2.4 Activity Diagram - Điều khiển thiết bị.....	18
Hình 2.5 Activity Diagram - Cảnh báo cháy .....	19
Hình 2.6 Activity Diagram - Xem nhiệt độ - độ ẩm.....	20
Hình 2.7 Activity Diagram - Thêm mới , xóa thiết bị.....	21
Hình 2.8 Activity Diagram - Lập báo cáo thống kê.....	22
Hình 2.9 Activity Diagram - Quản lý người dùng .....	23
Hình 2.10 Sequence Diagram - Đăng nhập .....	24
Hình 2.11 Sequence Diagram - Điều khiển thiết bị .....	25
Hình 2.12 Sequence Diagram - Cảnh báo cháy .....	25
Hình 2.13 Sequence Diagram - Xem nhiệt độ - độ ẩm .....	26
Hình 2.14 Sequence Diagram - Thêm mới , xóa thiết bị .....	27
Hình 2.15 Sequence Diagram - Lập báo cáo thống kê .....	28
Hình 2.16 Sequence Diagram - Quản lý người dùng.....	29
Hình 3.1 Giao diện đăng nhập.....	31
Hình 3.2 Giao diện trang chính .....	31
Hình 3.3 Giao diện xem nhiệt độ độ ẩm .....	32
Hình 3.4 Giao diện điều khiển ánh sáng.....	32
Hình 3.5 Sơ đồ chân ESP8266 .....	33

<b>Hình 3.6 DHT 11 .....</b>	<b>34</b>
<b>Hình 3.7 Cảm biến ánh sáng.....</b>	<b>35</b>
<b>Hình 3.8 Cảm biến khí gas.....</b>	<b>36</b>
<b>Hình 3.9 Test trên bo mạch trắng.....</b>	<b>37</b>

## LỜI MỞ ĐẦU

Những năm gần đây, khoa học công nghệ phát triển nhanh như vũ bão, đặc biệt là lĩnh vực công nghệ thông tin. Có thể nói, công nghệ thông tin đã len lỏi vào hầu hết tất cả mọi lĩnh vực trong cuộc sống của chúng ta. Trong đó, công nghệ phần mềm luôn luôn đóng vai trò tiên phong, dẫn đầu trong quá trình đó. Các phần mềm đã đang và sẽ được ứng dụng ngày càng nhiều trong công việc cũng như cuộc sống thường ngày của con người. Cùng với đó là sự phát triển ngày càng nhiều của các ngôn ngữ lập trình với xu thế là lập trình hướng đối tượng, giúp người lập trình dễ dàng thao tác với các đối tượng mong muốn cũng như dễ dàng trong việc sửa chữa nâng cấp và bảo trì.

Nắm bắt được xu hướng đó, em đã quyết định tìm hiểu và thiết kế một phần mềm đơn giản và gần gũi với thực tế: “Hệ thống IOT quản lí và thu thập dữ liệu các thiết bị trong nhà “ , với mục đích tìm hiểu thêm về lĩnh vực kỹ thuật số và nâng cao kiến thức của mình. Đồng thời, việc thực hiện đề tài này cũng giúp em nâng cao khả năng làm việc cá nhân, khả năng thiết kế, thực hành và rút ra những kinh nghiệm thực tiễn trong quá trình thiết kế một sản phẩm kỹ thuật. Chi tiết thiết kế sản phẩm được em trình bày rõ hơn ở phần sau.

Cuối cùng, nhóm chúng em xin chân thành cảm ơn thầy **Vũ Song Tùng** đã hướng dẫn tận tình trong quá trình học tập và thực hiện đề tài. Trong quá trình thực hiện đề tài không thể tránh khỏi những sai sót, nhóm em rất mong nhận được sự nhận xét đánh giá, góp ý từ thầy để bài báo cáo của em được hoàn thiện hơn.

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

Chương này sẽ giới thiệu tổng quan về các khái niệm có trong đề tài của nhóm thực hiện, đưa ra các vấn đề cần giải quyết.

## 1.1 Hệ thống Internet of thing

### 1.1.1 Khái niệm về IoT

IoT viết tắt của Internet of Things, nghĩa là Internet vạn vật, một hệ thống các thiết bị tính toán, máy móc cơ khí và kỹ thuật số hoặc con người có liên quan với nhau và khả năng truyền dữ liệu qua mạng mà không yêu cầu sự tương tác giữa con người với máy tính.

Hệ thống IoT sẽ bao gồm các đặc trưng như sau:

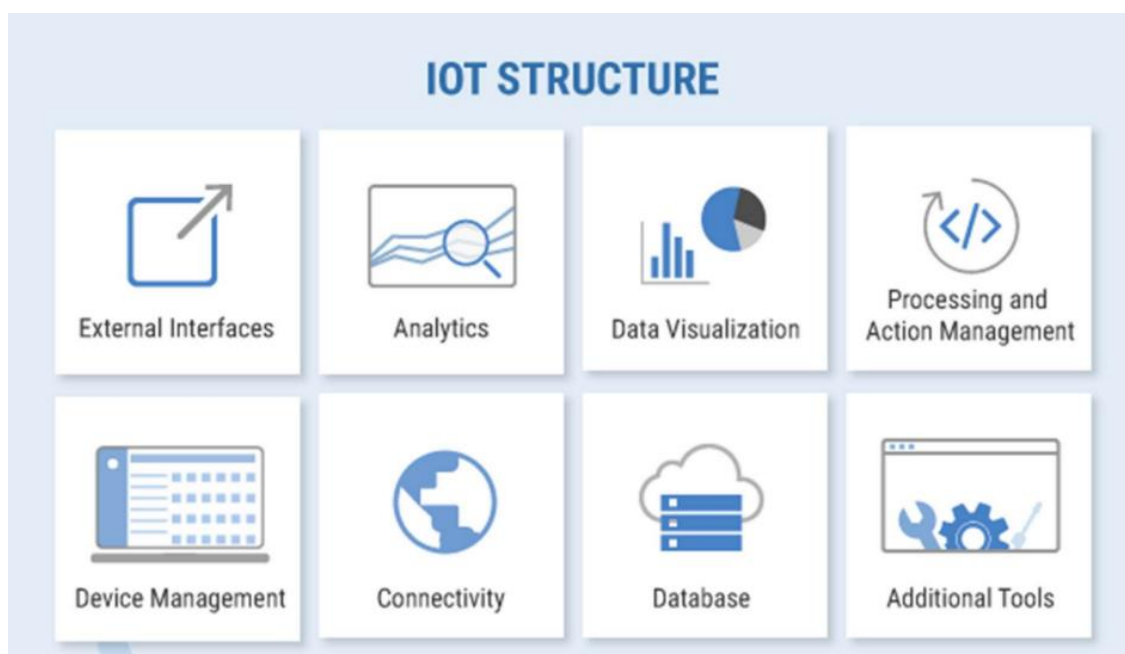
- Không đồng nhất: Các thiết bị trong IoT thường không đồng nhất vì sở hữu phần cứng, cũng như hệ thống khác nhau và có thể tương tác với nhau nhờ vào sự liên kết của các hệ thống.
- Kết nối liên thông (Interconnectivity): Đối với hệ thống IoT, các vật dụng, thiết bị đa dạng đều có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.
- Dịch vụ liên quan đến “Things”: Hệ thống IoT có khả năng cung cấp các dịch vụ liên quan đến “Things” chẳng hạn như bảo vệ sự riêng tư và nhất quán giữa thiết bị vật lý (Physical Thing) và phần mềm (Virtual Thing).
- Có quy mô lớn: Số lượng rất lớn các thiết bị, máy móc sẽ được quản lý và giao tiếp với nhau với quy mô lớn hơn nhiều so với số lượng máy tính kết nối Internet hiện nay. Số lượng thông tin được truyền bởi các thiết bị sẽ lớn hơn nhiều so với con người thực hiện.
- Có thể thay đổi linh hoạt: Các trạng thái của các thiết bị điện tử, máy móc có thể tự động thay đổi như tắt và bật, kết nối hoặc ngắt, truy xuất vị trí,...

IoT giúp cuộc sống thông minh hơn, tiện lợi và kết nối tốt hơn. Hệ thống Internet kết nối vạn vật này cung cấp cho các doanh nghiệp cái nhìn chi tiết về mọi thứ từ thời gian, hiệu suất của máy móc đến chuỗi cung ứng và hoạt động hậu cần. IoT giúp công ty tự động

hóa các quy trình, giảm chi phí lao động, giảm chất thải, cải thiện dịch vụ, làm cho việc sản xuất và giao hàng ít tốn kém hơn và đồng thời mang lại sự minh bạch trong các giao dịch của khách hàng.

### 1.1.2 Cấu trúc hệ thống IoT

Một hệ thống IoT sẽ bao gồm 4 thành phần chính bao gồm: Thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng (Network and Cloud), bộ phân tích và xử lý dữ liệu (Services-creation and Solution Layers).



Hình 1.1 Cấu trúc hệ thống IoT

### 1.1.3 Các dự án IoT phổ biến hiện nay

IoT được ứng dụng trong nhiều khía cạnh cuộc sống và phục vụ cho đa dạng đối tượng trong đời sống hiện nay :

- Ứng dụng cho doanh nghiệp :

Lợi ích của IoT cho doanh nghiệp phụ thuộc vào việc triển khai cụ thể dựa trên việc truy cập vào nguồn dữ liệu về các sản phẩm và hệ thống nội bộ của họ. Các nhà sản xuất đang bổ sung các cảm biến vào các thành phần của sản phẩm để giúp nó có thể kết nối và truyền tải dữ liệu, từ đó giúp phát hiện ra lỗi trước khi thiệt hại xảy ra. Việc sử dụng IoT cho doanh nghiệp có thể được chia thành hai phân khúc:



- ✓ Các dịch vụ dành riêng cho ngành như cảm biến trong nhà máy phát điện hoặc thiết bị định vị thời gian thực để chăm sóc sức khỏe.
- ✓ Các thiết bị IoT được sử dụng trong tất cả các ngành công nghiệp như điều hòa không khí thông minh hoặc hệ thống an ninh.
- Ứng dụng cho người dùng

IoT sẽ làm cho nhà, văn phòng và phương tiện trở nên thông minh hơn, dễ đo lường hơn và tự động tốt hơn. Các thiết bị thông minh như trợ lý ảo như Google Assistant giúp tiết kiệm sức lực, hỗ trợ các công việc hằng ngày như phát nhạc, bật điều hoà,... Bên cạnh đó, nhờ vào IoT có thể giúp chúng ta biết được mức độ ô nhiễm của môi trường ngay trên điện thoại hay máy tính thông qua các chỉ số đo lường, điều khiển xe tự lái và thành phố thông minh có thể thay đổi cách chúng ta xây dựng và quản lý không gian công cộng .

Tuy nhiên, nhiều trong số những đổi mới này có thể có ảnh hưởng lớn đến quyền riêng tư cá nhân của chúng ta, cụ thể có thể đến là rò rỉ dữ liệu cá nhân và bị "theo dõi" qua hệ thống camera an ninh, giám sát,...

## **1.2 Đặt vấn đề**

Khi công nghệ phát triển, ngành thiết kế cũng thay đổi theo xu hướng thời đại, đem đến nhiều giá trị đột phá và đóng góp cho xã hội. Con người dần tăng cường tương tác với nhau qua các thiết bị công nghệ, cảm nhận sự thuận tiện và nhanh chóng rõ rệt so với các phương thức truyền thống. Đi kèm với đó, sự phát triển mạnh mẽ của các ngành khoa học kỹ thuật và kỹ thuật điều khiển tự động, giúp cung cấp các giải pháp công nghệ, đặc biệt là IoT, ngày càng hiện đại, đa dạng và có khả năng ứng dụng thực tiễn vào cuộc sống nhằm cải thiện chất lượng sống của con người .

Mô hình nhà thông minh của Việt Nam và thế giới đang được phát triển hiện nay ,các thiết bị trong nhà có thể hoạt động tự động hoặc dưới sự điều khiển của con người ,các thiết bị được điều khiển từ xa thông qua SmartPhone, Web..., có thể nắm được tình hình ngôi nhà mình và điều khiển chúng từ một nơi nào đó cách xa thông qua internet.

## 1.3 Các phần mềm , công cụ hỗ trợ

### 1.3.1 Ngôn ngữ Java

Java là một ngôn ngữ lập trình, được phát triển bởi **Sun Microsystem** vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

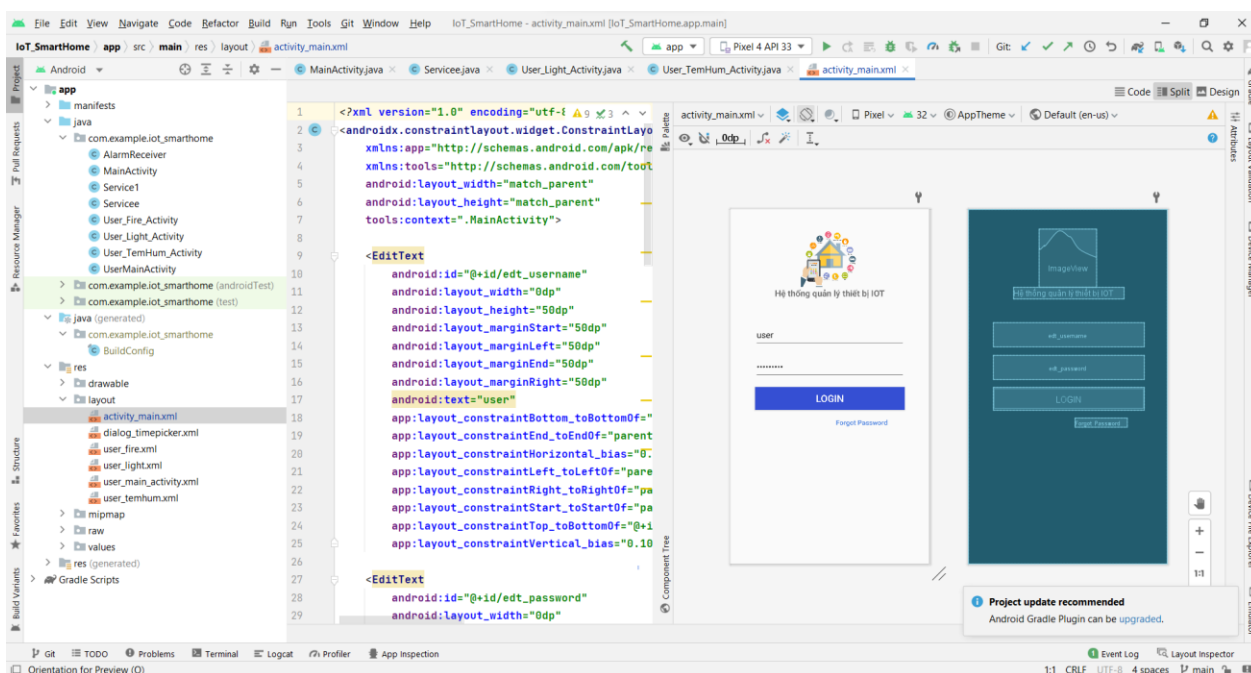
Ngày nay Java được sử dụng với các mục đích sau: phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn , tạo các trang web có nội dung động (*web applet*), nâng cao chức năng của server, phát triển nhiều loại ứng dụng khác nhau: cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

### 1.3.2 Android Studio

Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE Java IntelliJ của hãng JetBrains (đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).

Một số tính năng nổi bật :

- Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
- Chức năng dò và sửa lỗi nhanh, hướng Android.
- Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
- Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
- Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

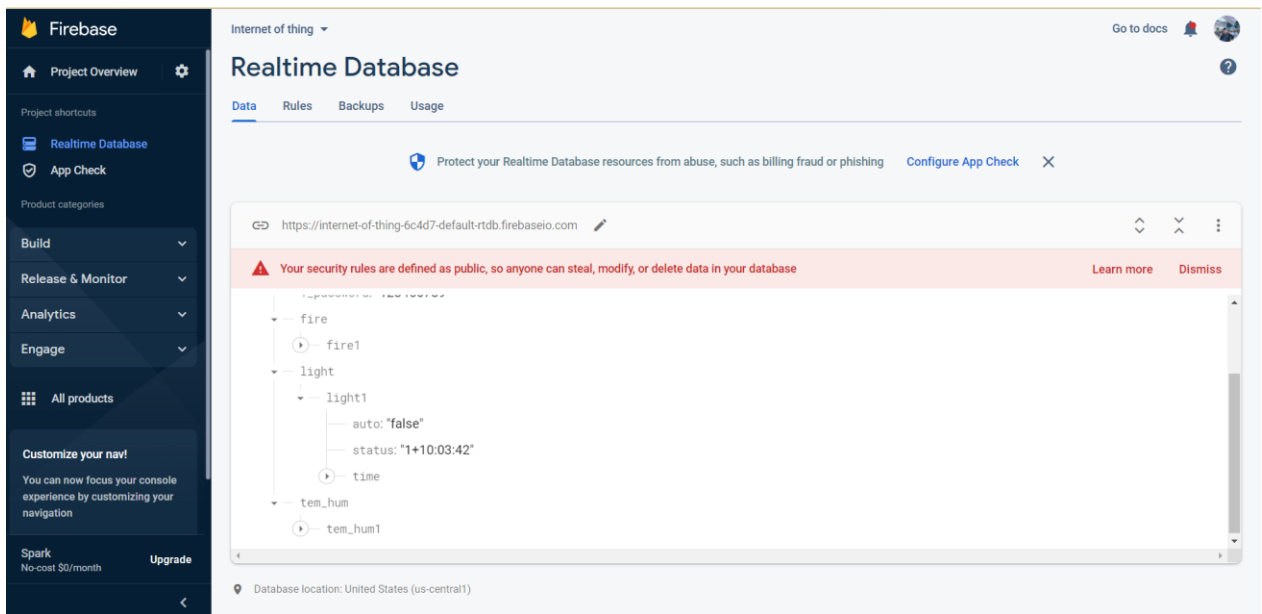


Hình 1.2 Giao diện lập trình Android Studio

### 1.3.3 Google Firebase

Firebase realtime database là một cơ sở dữ liệu thời gian thực, NoSQL được lưu trữ đám mây cho phép bạn lưu trữ và đồng bộ dữ liệu. Dữ liệu được lưu trữ dưới dạng cây Json, và được đồng bộ theo thời gian thực đối với mọi kết nối.

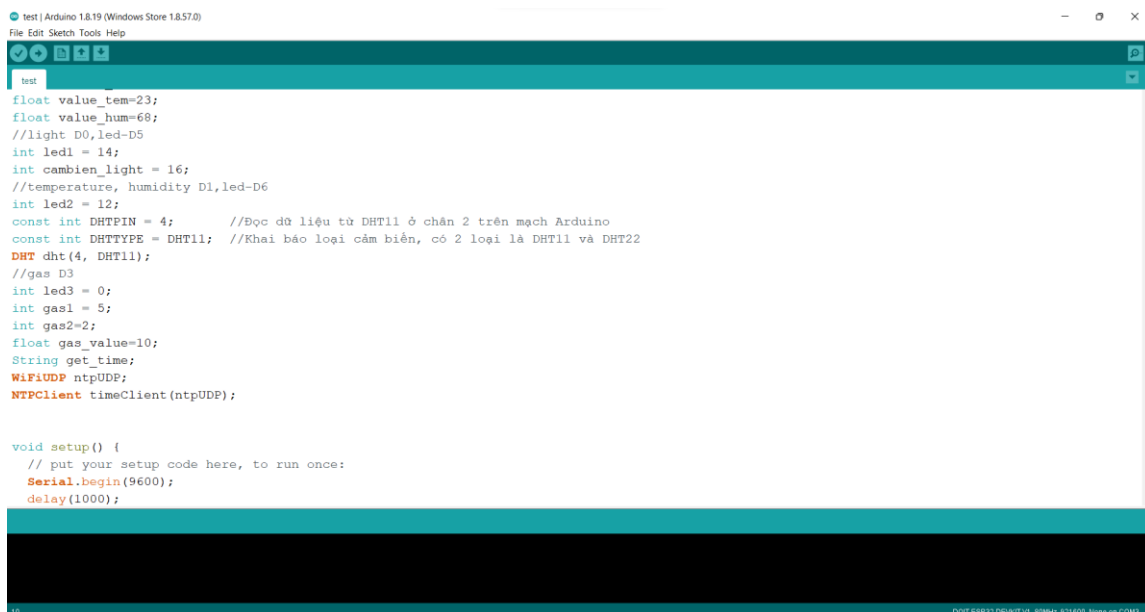
Khi bạn xây dựng những ứng dụng đa nền tảng như Android, IOS và Web App, tất cả các client của bạn sẽ kết nối trên cùng một cơ sở dữ liệu Firebase và tự động cập nhật dữ liệu mới nhất khi có sự thay đổi cả một cơ sở dữ liệu là một cây json lớn, với độ trễ thấp, Firebase realtime database cho phép bạn xây dựng các ứng dụng cần độ realtime như app chat, hệ thống IoT.



Hình 1.3 Firebase realtime database

### 1.3.4 Aduino IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.



Hình 1.4 Giao diện Aduino IDE

## 1.4 Kết luận

Từ những khái niệm trên, sau khi đưa ra được vấn đề cần giải quyết và các phần mềm công cụ hỗ trợ, đề tài sẽ thực hiện việc truyền tải dữ liệu từ các thiết bị, cảm biến, thông qua các nền tảng như: web, ESP32, ứng dụng mobile, ... và đẩy lên một server nhằm lưu trữ dữ liệu (trong đề tài này nhóm chọn Google Firebase), tiếp theo sẽ thực hiện thiết kế hệ thống cho đề tài.

## CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

Chương này sẽ đưa ra các yêu cầu về hệ thống, tiến hành xây dựng các sơ đồ cơ bản UML là Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram.

### 2.1 Xây dựng yêu cầu hệ thống

#### 2.1.1 Yêu cầu chức năng

Yêu cầu xây dựng hệ thống quản lý và thu thập dữ liệu các thiết bị trong nhà:

- Người dùng có thể nắm bắt được trạng thái của thiết bị thông qua các ứng dụng điện thoại Android.
- Người dùng có thể đăng nhập bằng hệ thống nhận diện khuôn mặt.
- Người dùng có thể điều khiển thiết bị từ xa thông qua các ứng dụng điện thoại Android.

Việc sử dụng hệ thống điều khiển thiết bị từ xa được mô tả như sau:

Người dùng truy cập vào các ứng dụng được chỉ định để tiến hành kết nối với các thiết bị thông qua mạng. Người dùng tiến hành đăng nhập vào các app để có thể truy cập vào hệ thống. Sau khi truy cập, người dùng chọn các chức năng hiển thị thông qua giao diện của ứng dụng. Sau đó hệ thống sẽ trả về các dữ liệu thu thập được như là nhiệt độ, độ ẩm. Nếu thiết bị được kết nối thành công thì hệ thống sẽ trả về trạng thái đang hoạt động của thiết bị cần điều khiển. Người dùng có thể tiến hành thay đổi trạng thái của thiết bị thông qua giao diện của ứng dụng.

#### 2.1.2 Yêu cầu phi chức năng

Để hoàn thiện được hệ thống và đưa tới người tiêu dùng thì hệ thống có các yêu cầu phi chức năng như sau:

- Hệ thống có độ phản hồi là nhỏ nhất xấp xỉ 1ms.
- Nhỏ gọn, thiết bị dễ dàng sử dụng với người dùng.
- An toàn về mặt kỹ thuật điện.
- Cập nhật liên tục trạng thái của thiết bị, tầm 1 phút 1 lần.

## **2.2 Xác định các Actor**

### **2.2.1 Người dùng (User)**

- Là người trực tiếp sử dụng hệ thống điều khiển thiết bị từ xa thông qua các ứng dụng nhất định nhằm mục đích phục vụ cá nhân hoặc chung.
- Mỗi người dùng đều có tài khoản cá nhân, có thể thay đổi các trạng thái của các thiết bị thông qua phần mềm ( tắt/bật, xem nhiệt độ , độ ẩm...).

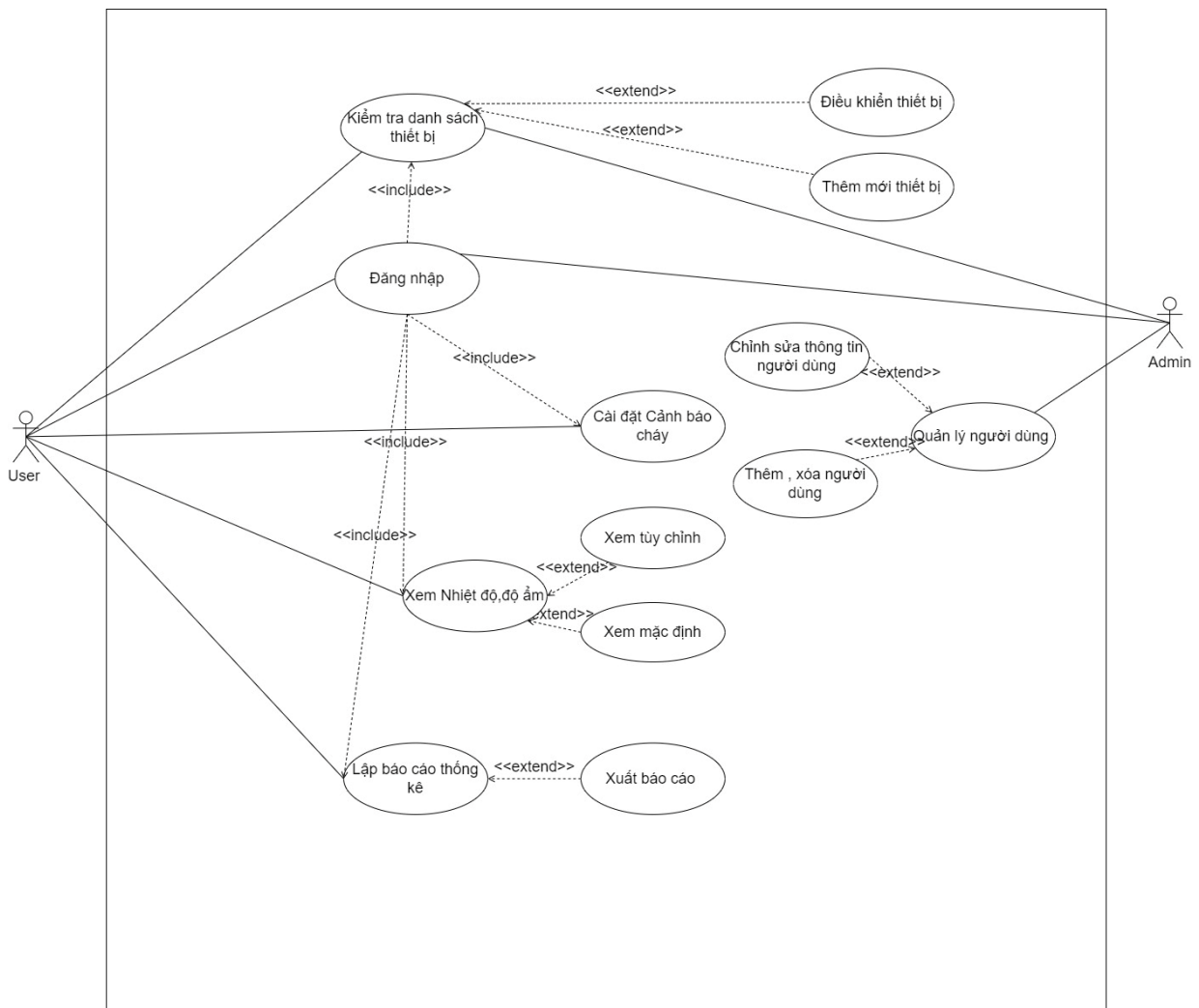
### **2.2.2 Admin**

- Là người tham gia vào hệ thống điều khiển từ xa thông qua các ứng dụng nhất định.
- Là người quản lý thiết lập các tài khoản của người dùng, đồng thời quản lý số thiết bị điều khiển trên phần mềm.

## **2.3 Use Case Diagram**

Các use case của người dùng(User) và Admin :

- Đăng nhập : Use case này cho phép người sử dụng đăng nhập vào hệ thống .
- Kiểm tra danh sách thiết bị : Use case này cho phép người dùng kiểm tra các thiết bị , thêm mới hoặc điều khiển thiết bị đó .
- Xem nhiệt độ , độ ẩm : Use case này cho phép người dùng xem trực tiếp nhiệt độ , độ ẩm trong nhà
- Cài đặt cảnh báo cháy : Use case này cho phép người dùng thiết lập cảnh báo cháy trong nhà .
- Lập báo cáo thống kê : Use case này cho phép người dùng lập báo cáo thống kê theo ngày , xuất file báo cáo .
- Quản lý người dùng : Use case này cho phép Admin có thể quản lý người dùng , chỉnh sửa thông tin người dùng .



**Hình 2.1 Use Case Diagram**

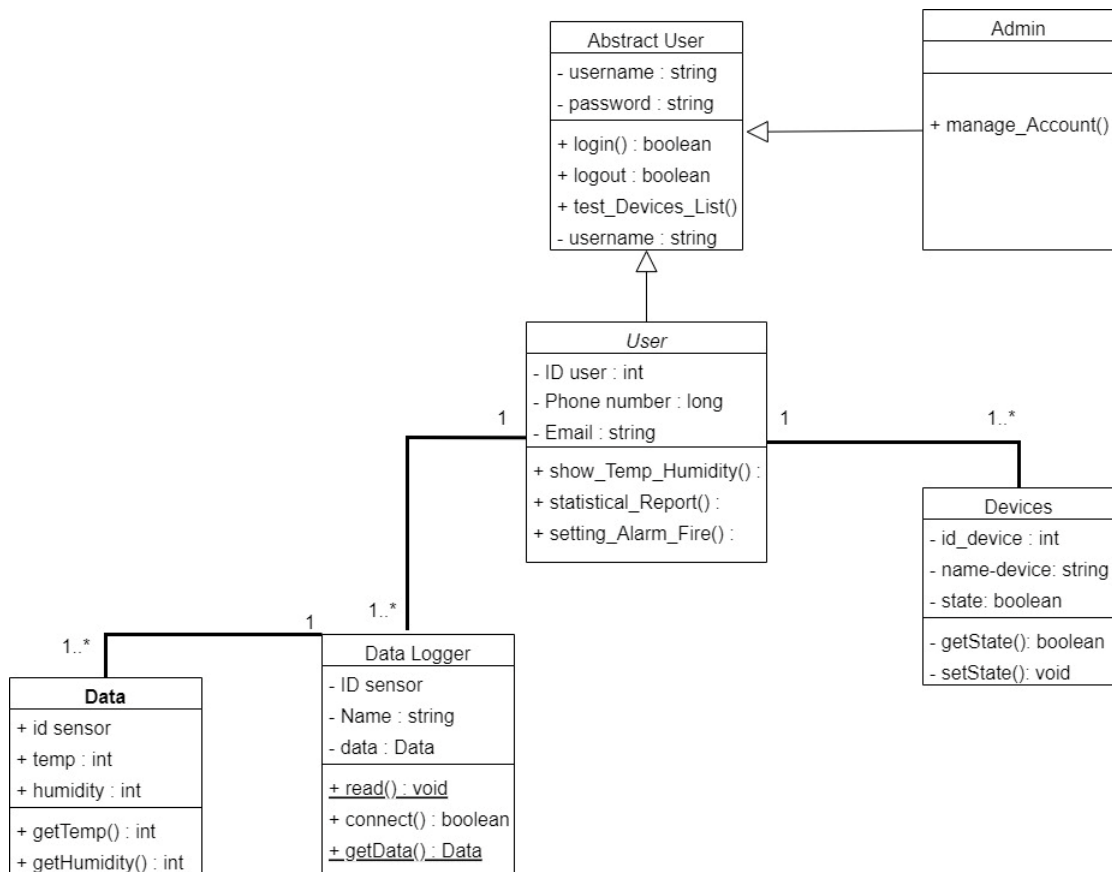
## 2.4 Class Diagram

Class diagram sẽ cho ta biết các class có trong hệ thống , từ đó sẽ tiến hành xây dựng , lập trình cho hệ thống dễ dàng hơn .

Các class có trong hệ thống :

- AbstractUser : là lớp trừu tượng tổng quát cho lớp Admin và User có thể kế thừa .
- User : là lớp đại diện cho người dùng .
- Admin : là lớp đại diện cho Admin
- Devices : là lớp đại diện các thiết bị trong nhà như : tủ lạnh , quạt
- Data : lớp đại diện cho các cảm biến để thu thập dữ liệu



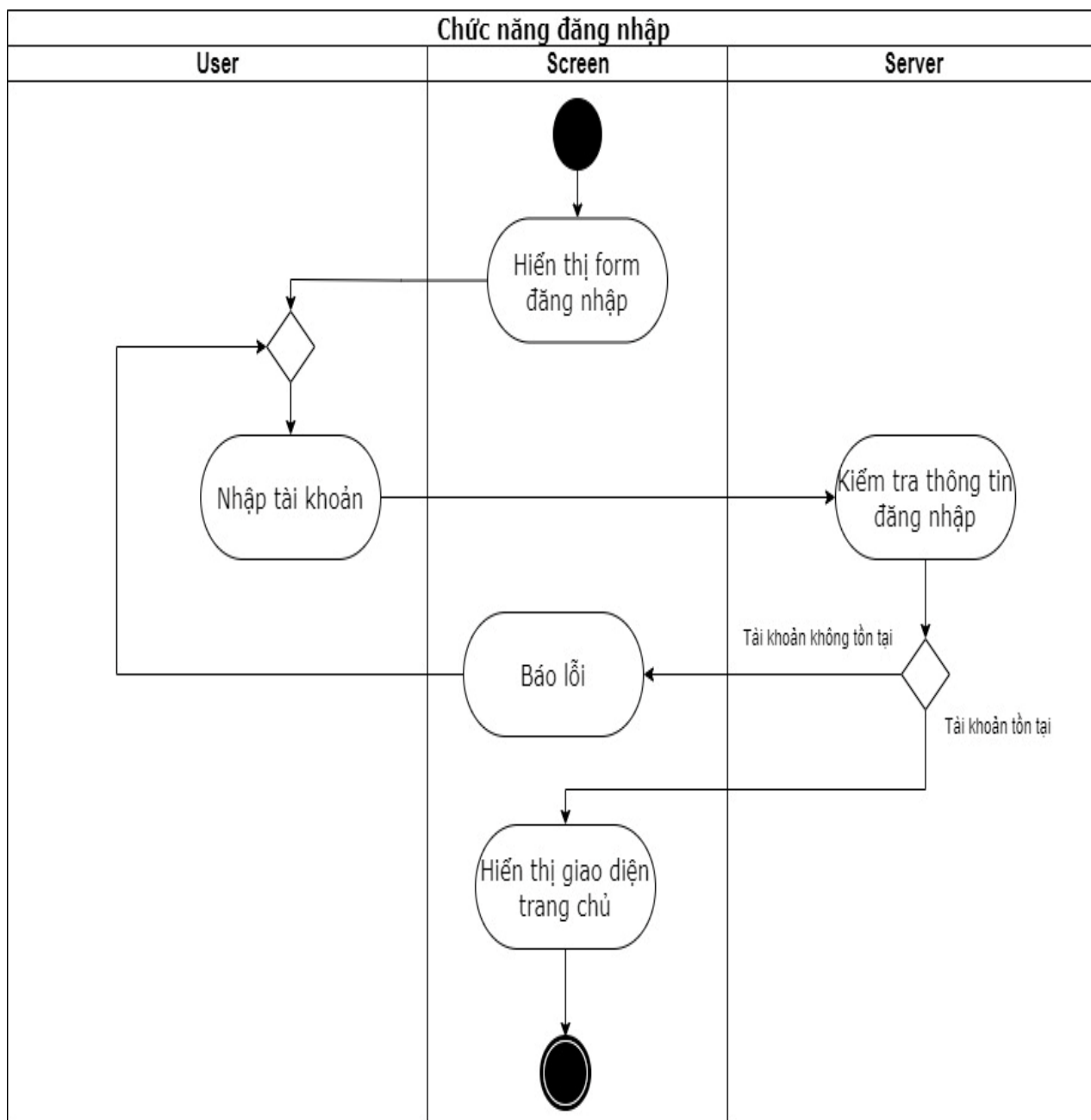


Hình 2.2 Class Diagram

## 2.5 Activity Diagram

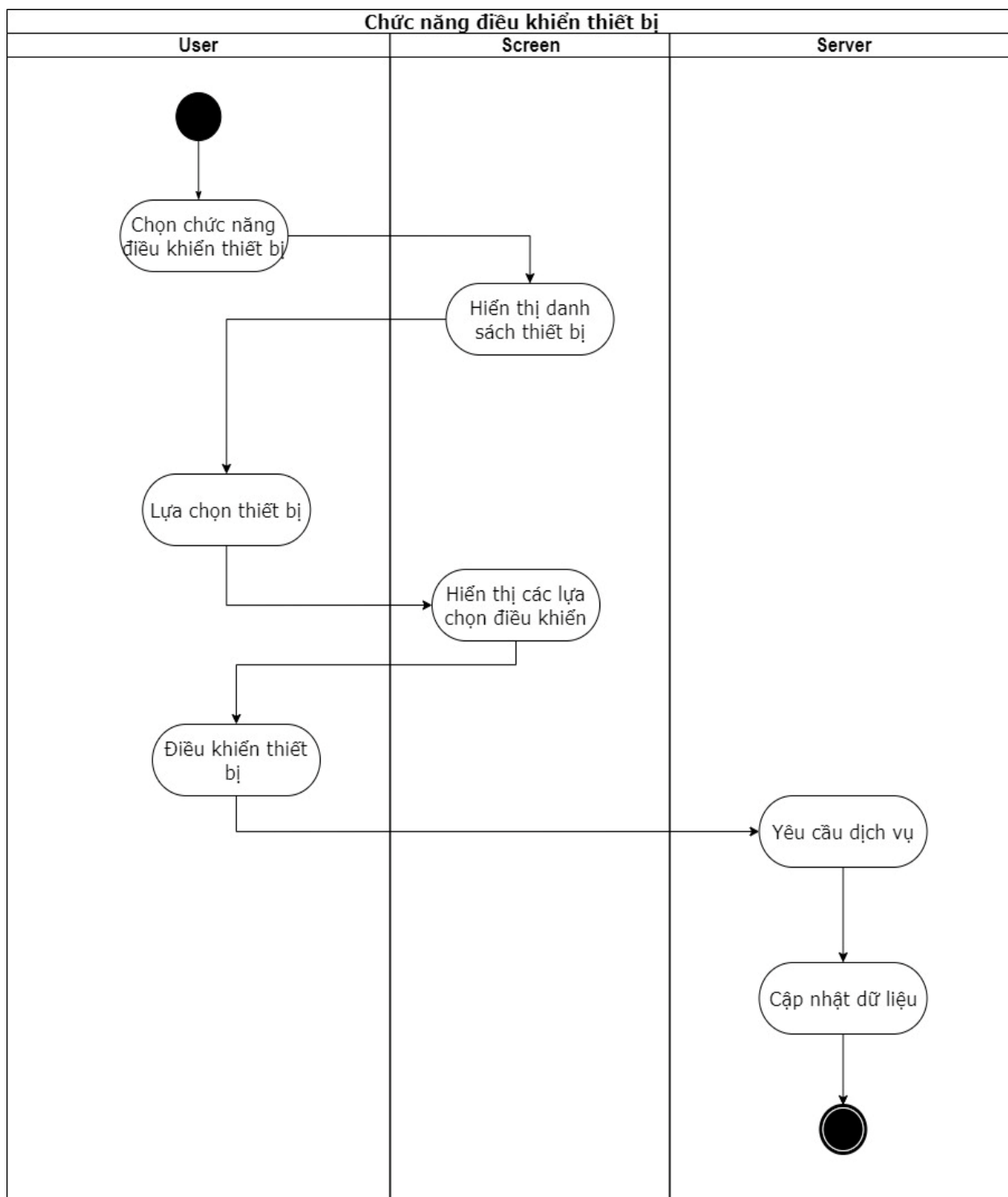
Activity Diagram(Sơ đồ hoạt động) mô tả cụ thể hoạt động của các chức năng trong sơ đồ use case , từ đó chúng ta sẽ hình dung ra được các bước cần làm của mỗi hoạt động .

### 2.5.1 Đăng nhập



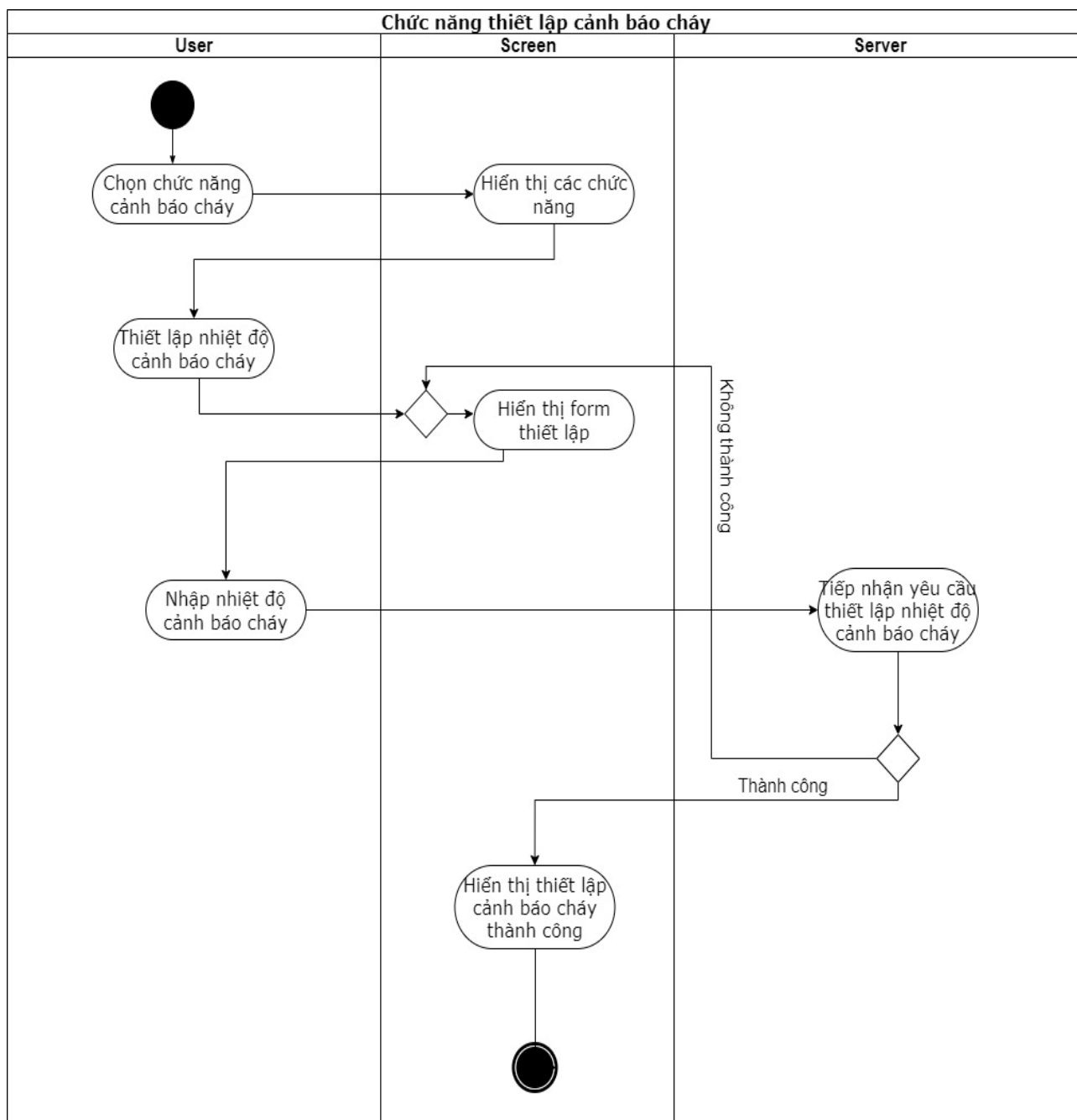
Hình 2.3 Activity Diagram - Đăng nhập

## 2.5.2 Điều khiển thiết bị



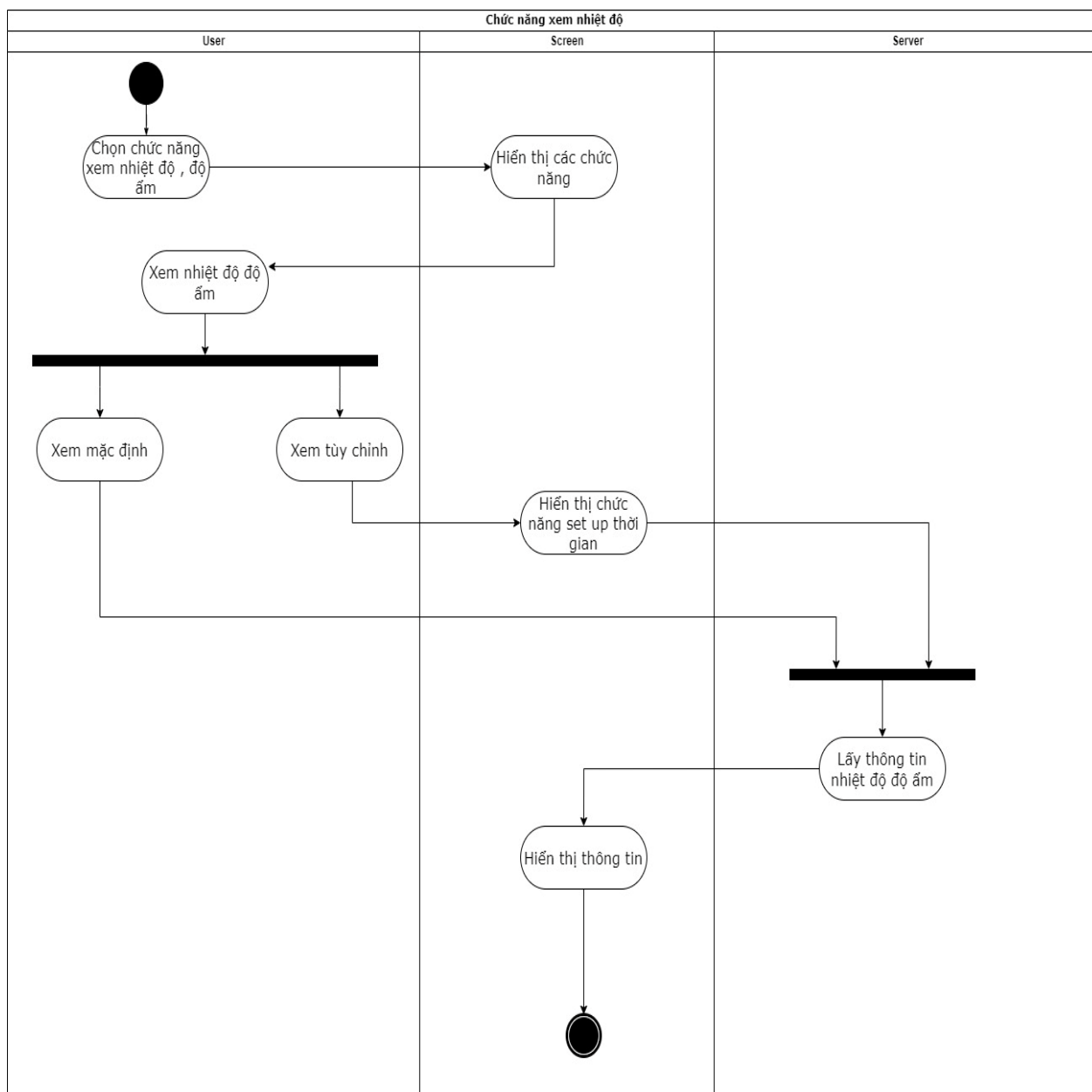
Hình 2.4 Activity Diagram - Điều khiển thiết bị

### 2.5.3 Cảnh báo cháy



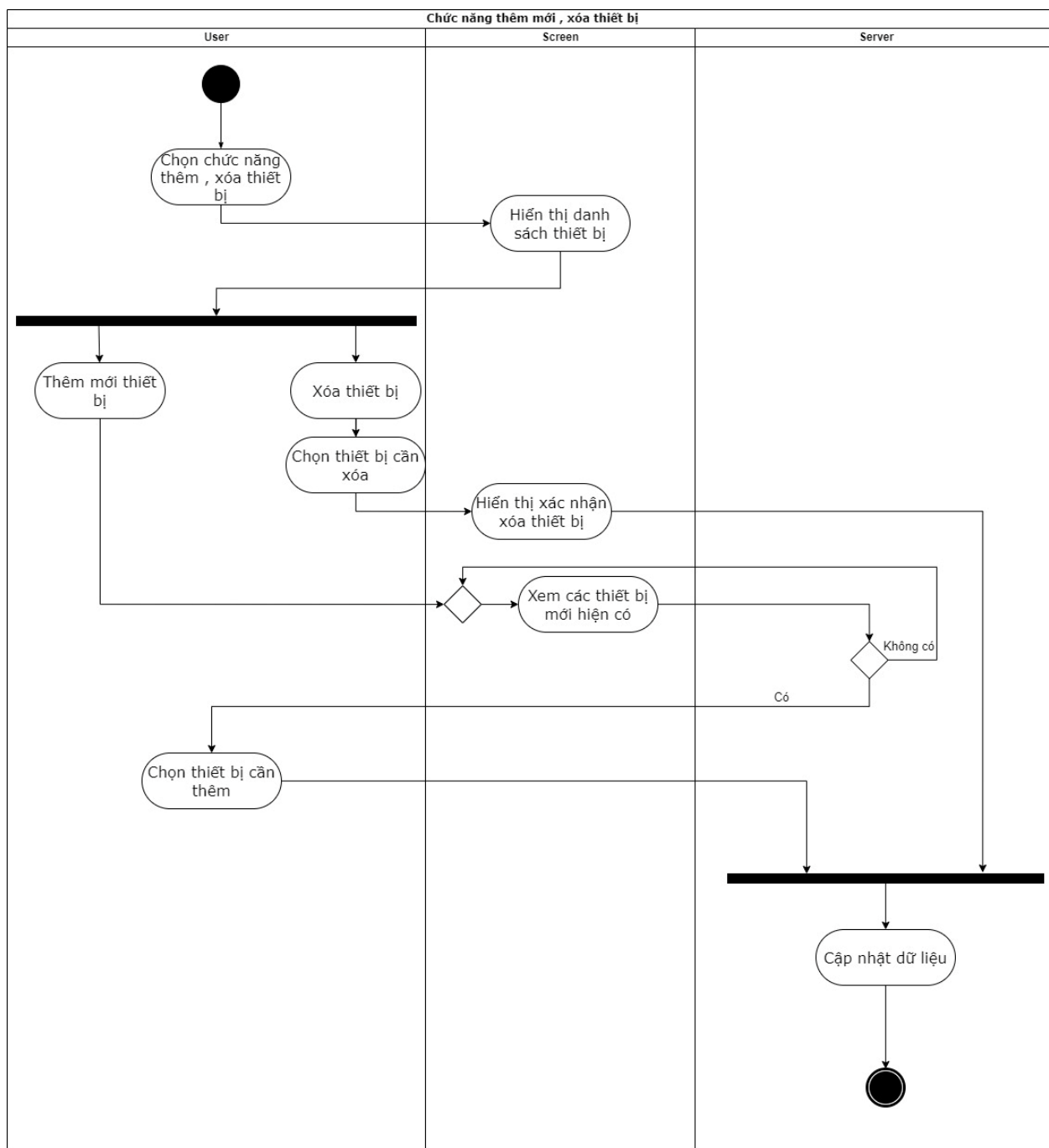
Hình 2.5 Activity Diagram - Cảnh báo cháy

## 2.5.4 Xem nhiệt độ - độ ẩm



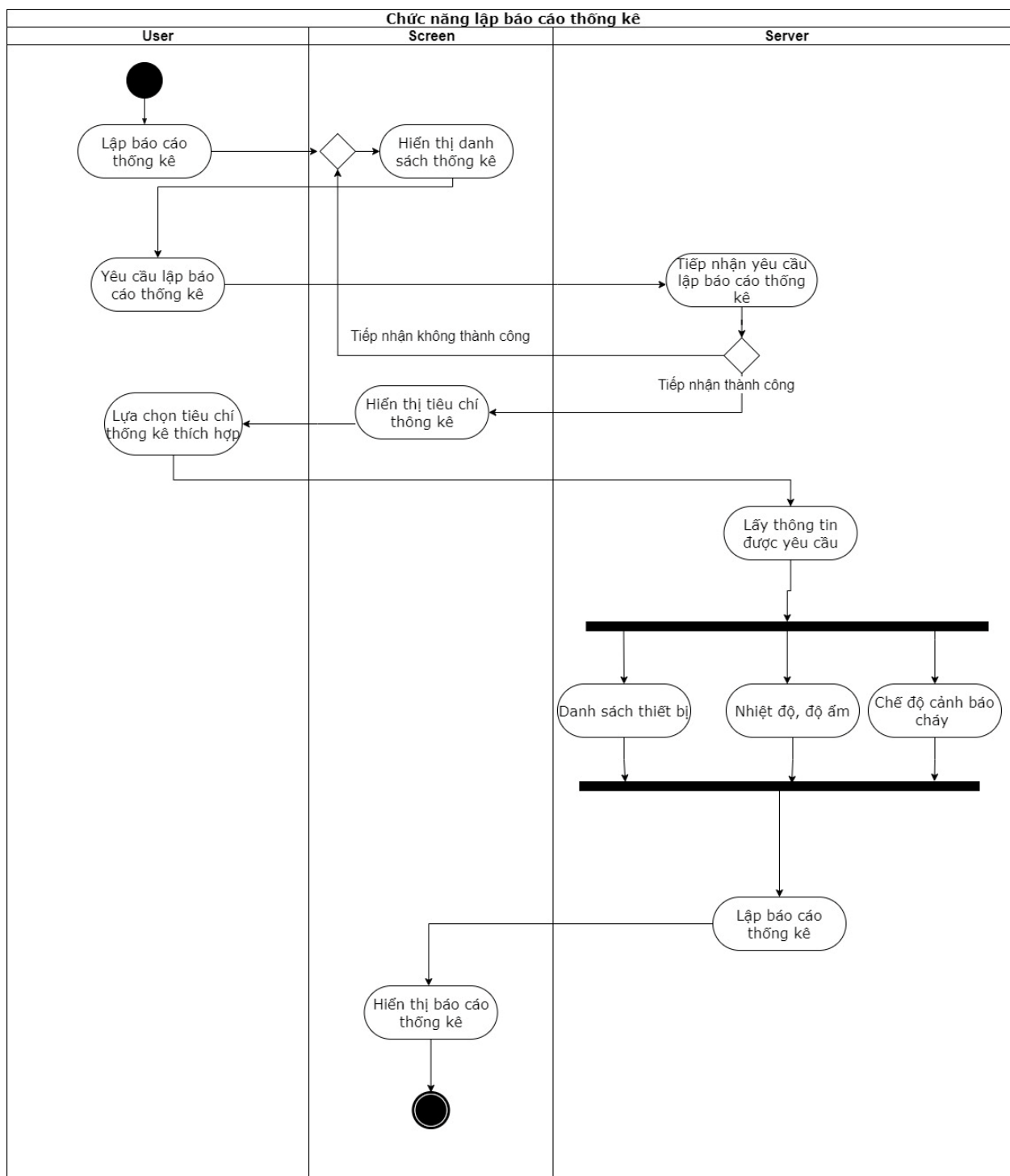
Hình 2.6 Activity Diagram - Xem nhiệt độ - độ ẩm

## 2.5.5 Thêm mới , xóa thiết bị



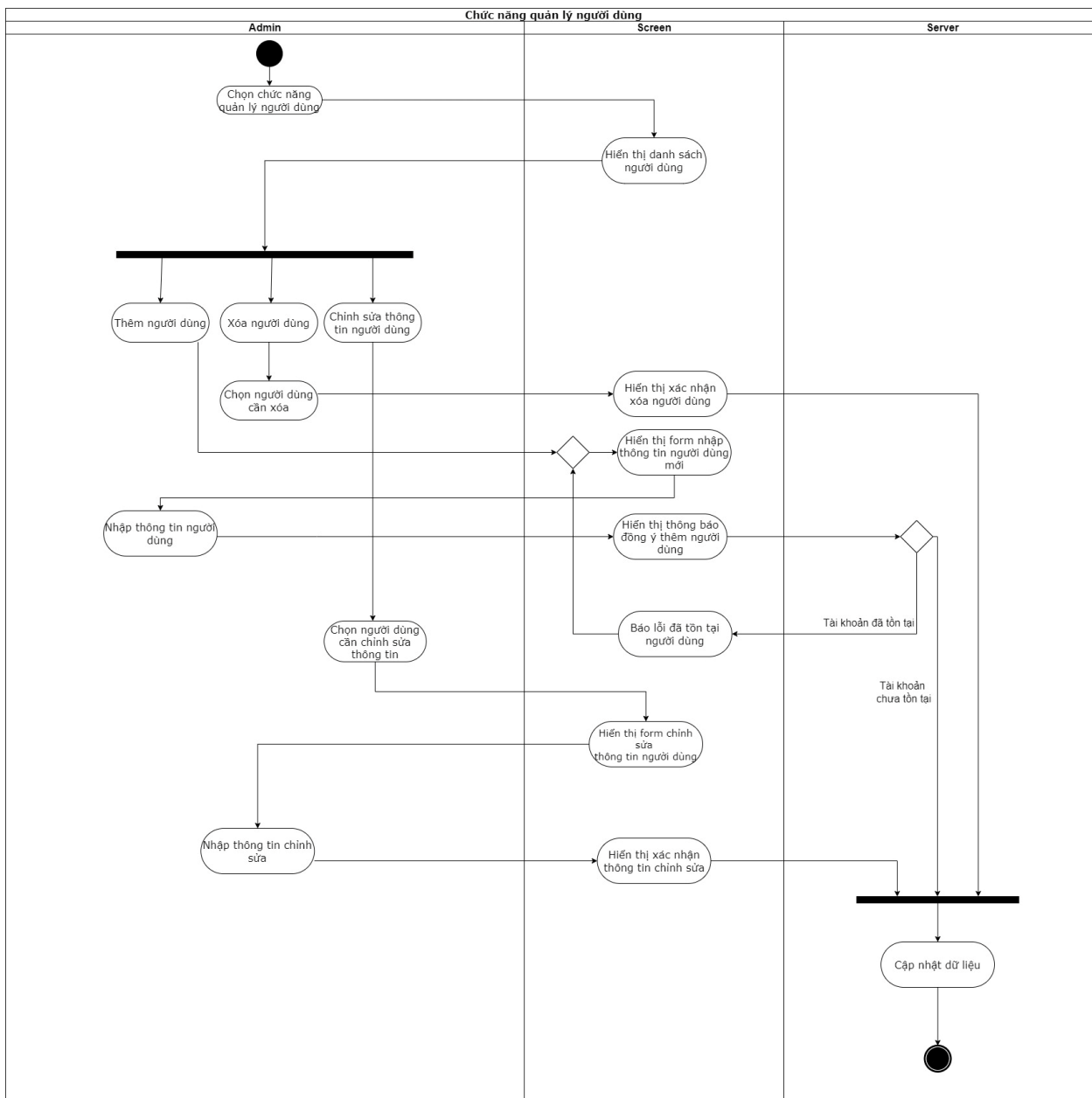
Hình 2.7 Activity Diagram - Thêm mới , xóa thiết bị

## 2.5.6 Lập báo cáo thống kê



Hình 2.8 Activity Diagram - Lập báo cáo thống kê

## 2.5.7 Quản lý người dùng



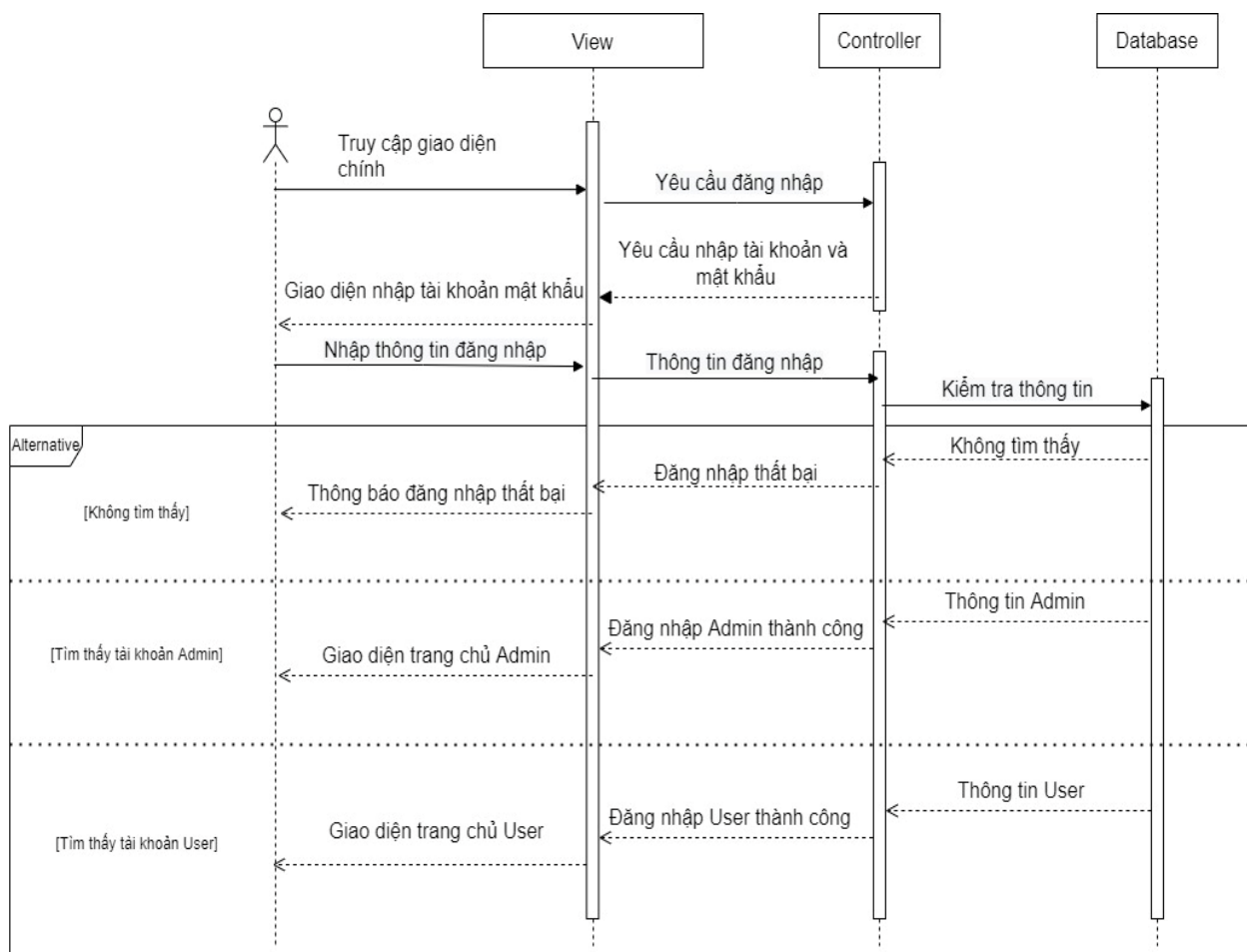
Hình 2.9 Activity Diagram - Quản lý người dùng

## 2.6 Sequence Diagram

Sequence Diagram(Sơ đồ tuần tự) dùng để xác định các trình tự diễn ra sự kiện của user và admin . Nó miêu tả chi tiết các thông điệp được gửi và nhận giữa các đối tượng đồng thời cũng chú trọng đến việc trình tự về mặt thời gian gửi và nhận các thông điệp đó.

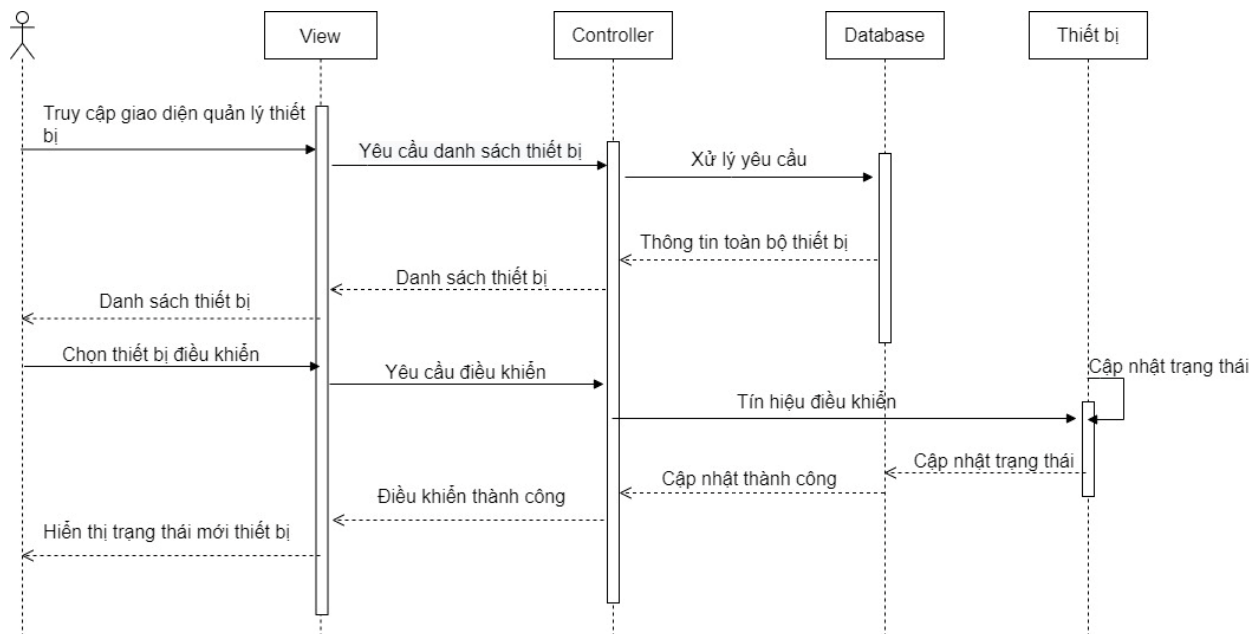


## 2.6.1 Đăng nhập



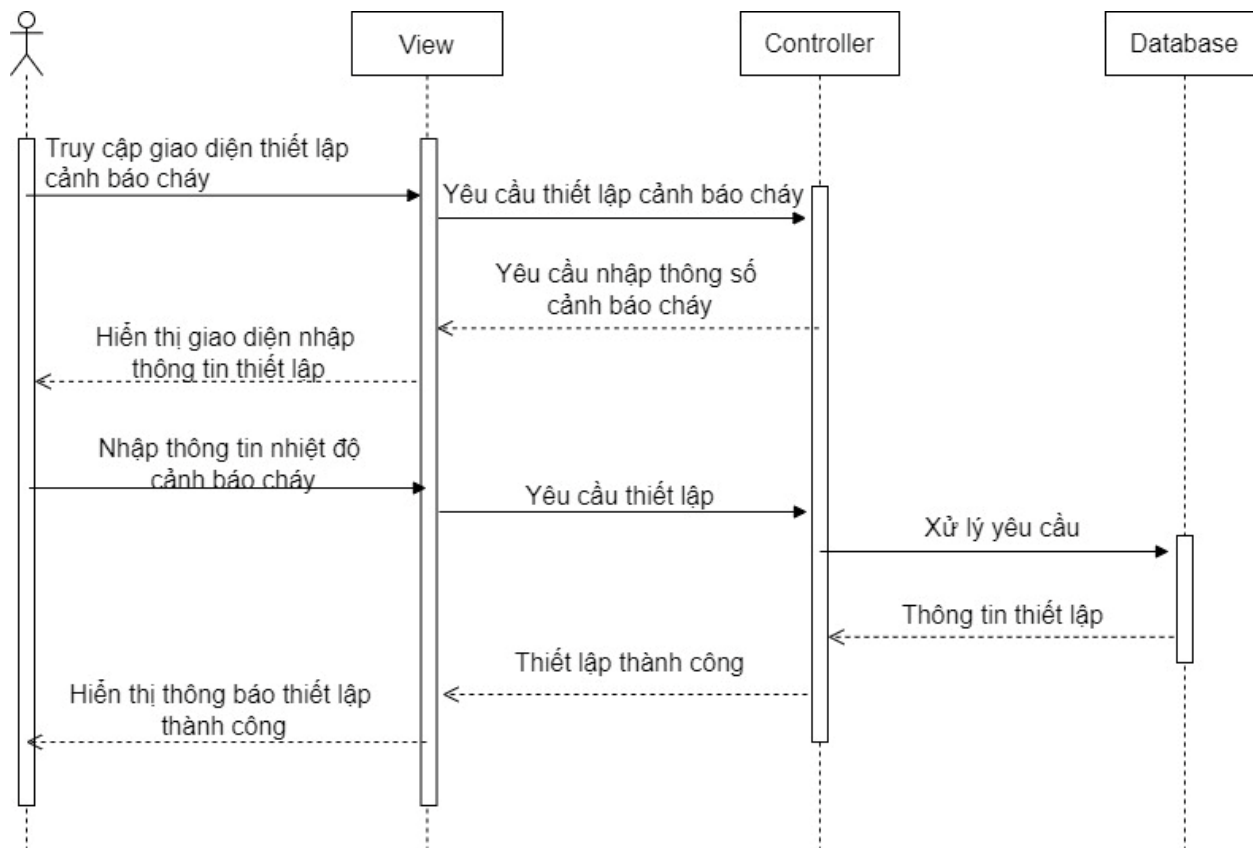
Hình 2.10 Sequence Diagram - Đăng nhập

## 2.6.2 Điều khiển thiết bị



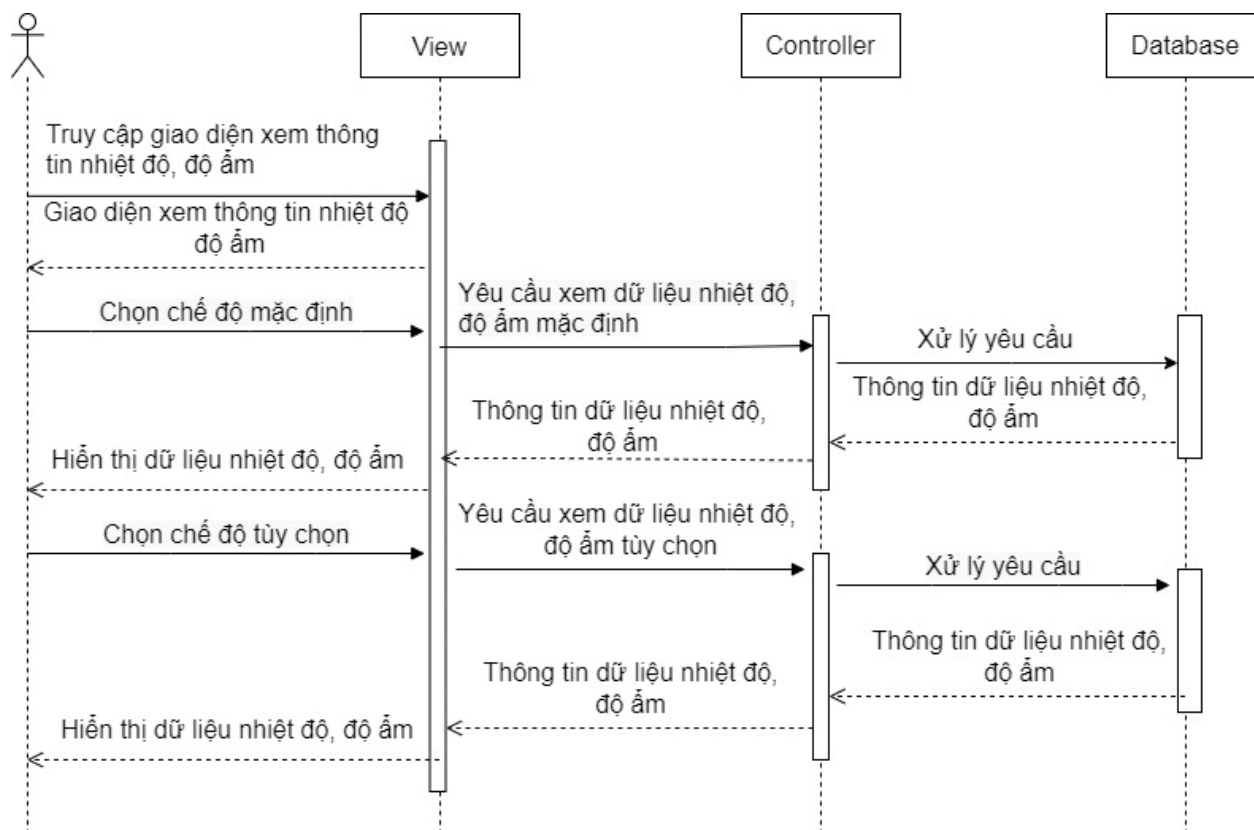
Hình 2.11 Sequence Diagram - Điều khiển thiết bị

## 2.6.3 Cảnh báo cháy



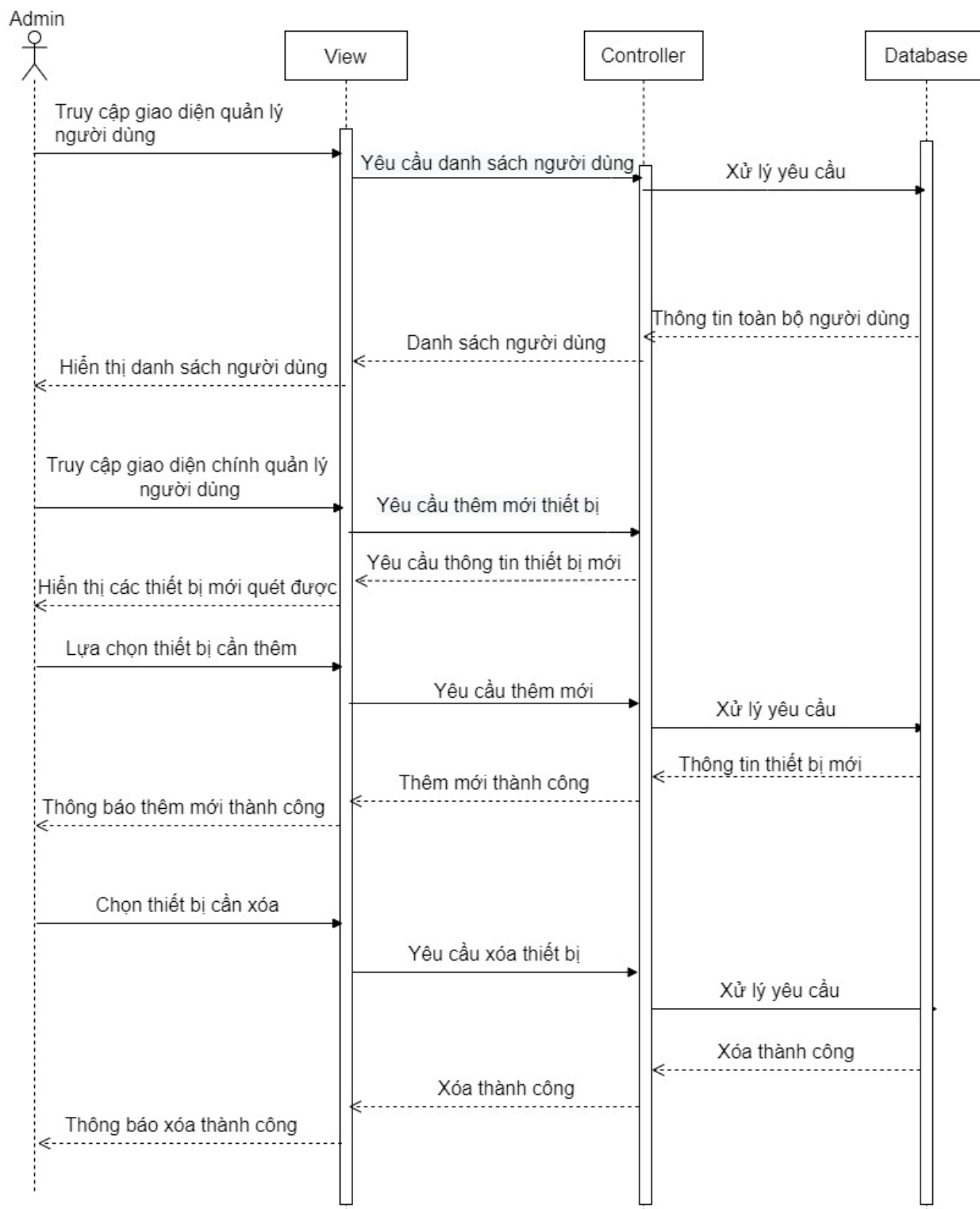
Hình 2.12 Sequence Diagram - Cảnh báo cháy

## 2.6.4 Xem nhiệt độ - độ ẩm



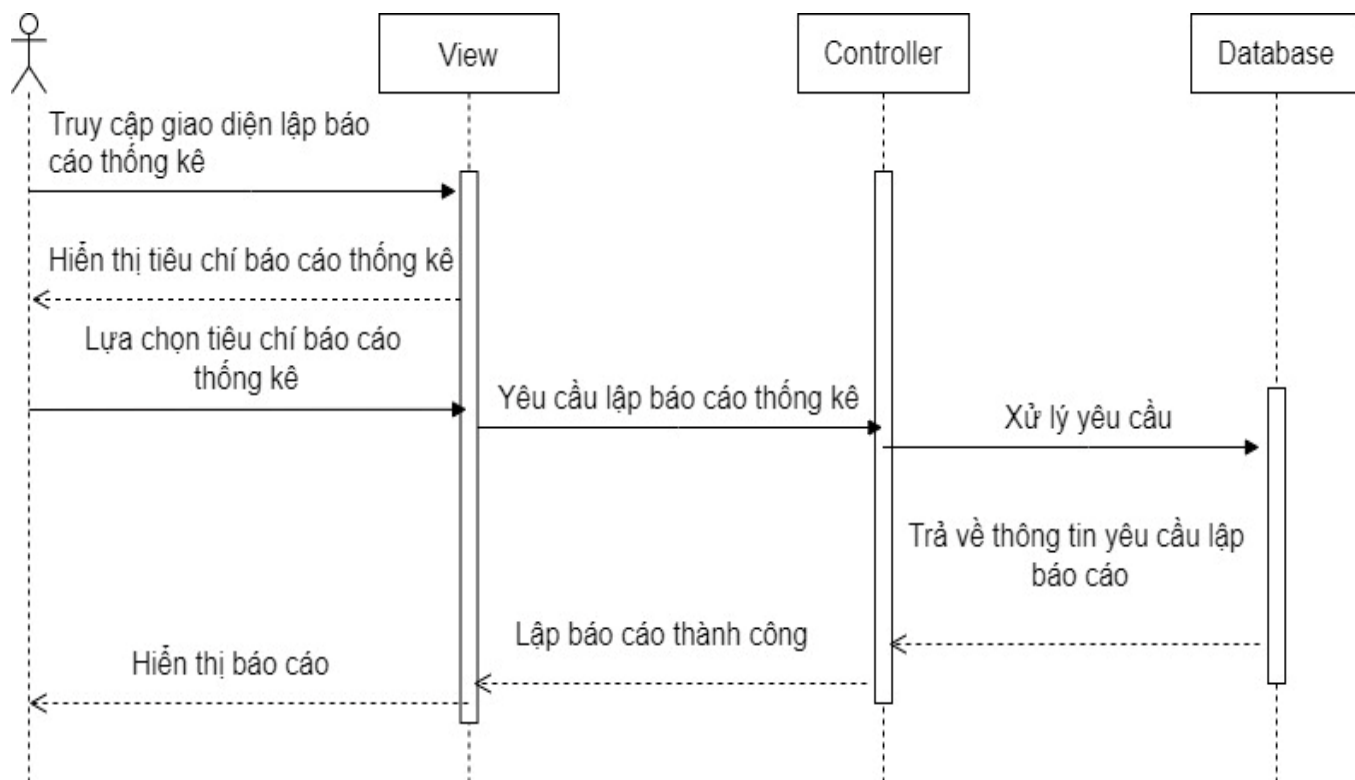
Hình 2.13 Sequence Diagram - Xem nhiệt độ - độ ẩm

## 2.6.5 Thêm mới , xóa thiết bị



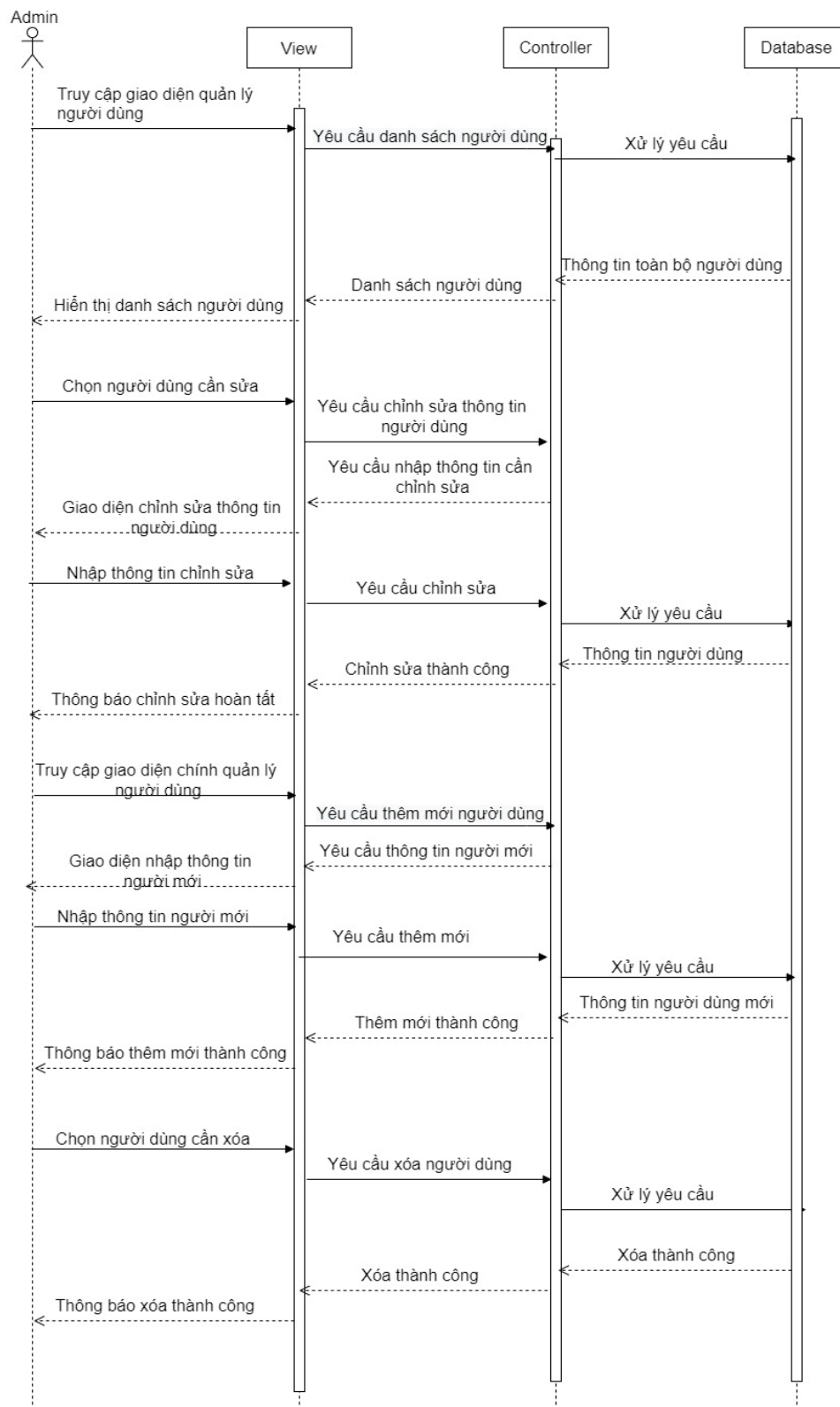
Hình 2.14 Sequence Diagram - Thêm mới , xóa thiết bị

### 2.6.6 Lập báo cáo thống kê



Hình 2.15 Sequence Diagram - Lập báo cáo thống kê

## 2.6.7 Quản lý người dùng



Hình 2.16 Sequence Diagram - Quản lý người dùng

## 2.7 Kết luận

Sau khi đã đưa ra các yêu cầu và sơ đồ hệ thống, tiếp đến sẽ tiến hành triển khai lập trình xây dựng ứng dụng trên điện thoại và chuẩn bị mạch demo.

# CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ THỰC NGHIỆM

## 3.1 Phần mềm

### 3.1.1 Yêu cầu

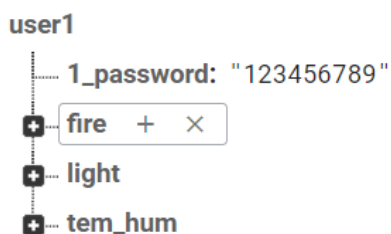
Gồm có Giao diện người dùng App trên SmartPhone.

Giao diện xây dựng cần đảm bảo các yêu cầu sau:

- Dễ nhìn, dễ sử dụng, thích hợp với nhiều đối tượng người dùng
- Độ ổn định cao
- Kết nối được với hệ thống qua internet
- Hiển thị thông số gửi từ hệ thống

### 3.1.2 Real-time Database

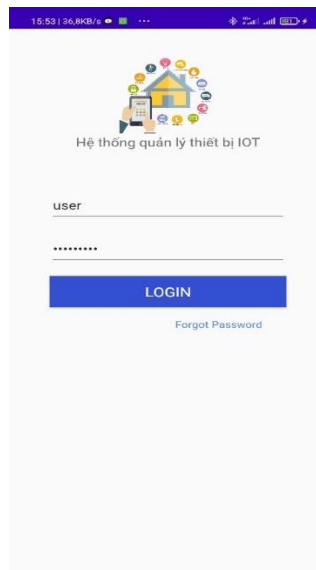
[internet-of-thing-6c4d7-default-rtdb](#) > [user1](#)



### 3.1.3 Giao diện phần mềm

#### 3.1.3.1 Đăng nhập

Cho phép người dùng có thể đăng nhập vào hệ thống , hiện tại với tài khoản cố định là “user1” và mật khẩu là “12345678”.



**Hình 3.1 Giao diện đăng nhập**

### 3.1.3.2 Trang chính

Cho phép người dùng lựa chọn một trong ba chức năng chính : Điều khiển bật tắt đèn , xem nhiệt độ độ ẩm , cảnh báo cháy

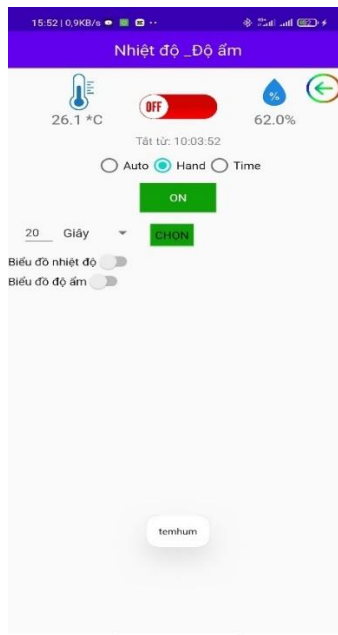


**Hình 3.2 Giao diện trang chính**

### 3.1.3.3 Xem nhiệt độ độ ẩm

Cho phép người dùng xem nhiệt độ độ ẩm hiện tại , xem biểu đồ , và cài đặt thời gian thu thập dữ liệu .

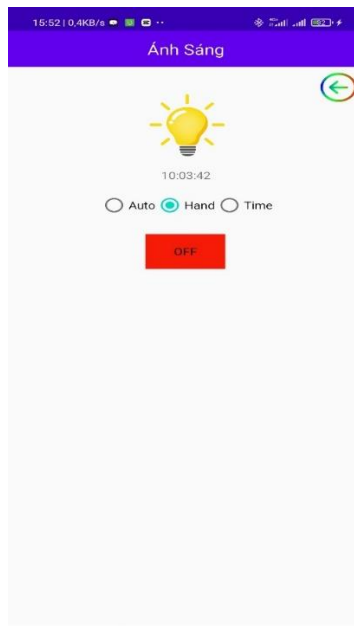




**Hình 3.3 Giao diện xem nhiệt độ độ ẩm**

#### 3.1.3.4 Điều khiển bật tắt ánh sáng

Cho phép người dùng điều khiển bật tắt ánh sáng bằng nút nhấn ON/OFF .

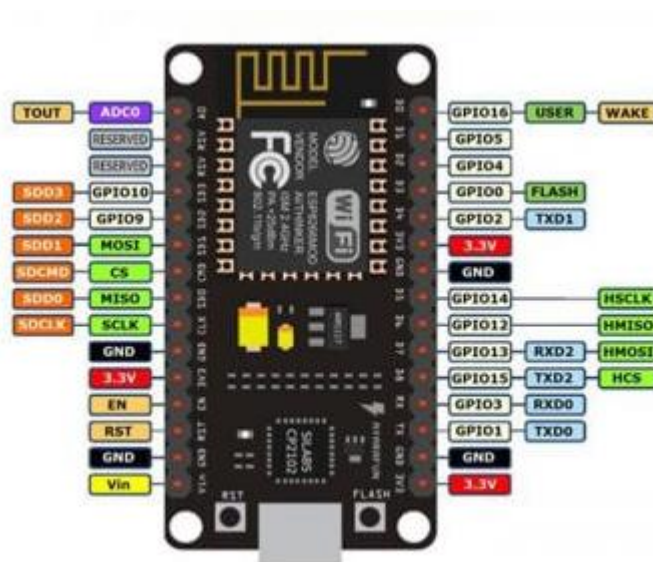


**Hình 3.4 Giao diện điều khiển ánh sáng**

## 3.2 Phần cứng

### 3.2.1 ESP8266

ESP8266 là một vi xử lý được thiết kế bởi Espressif System tích hợp sẵn wifi. Có nhiều loại board được thiết kế sử dụng vi xử lý ESP8266. Sự khác nhau của các loại board là số lượng các GPIO (GPIO là các cổng được sử dụng cho input hoặc output). Đây là một board rất hữu dụng để mọi người có thể có những bước tiếp cận tốt nhất đến thế giới IOT- Internet Of Things.

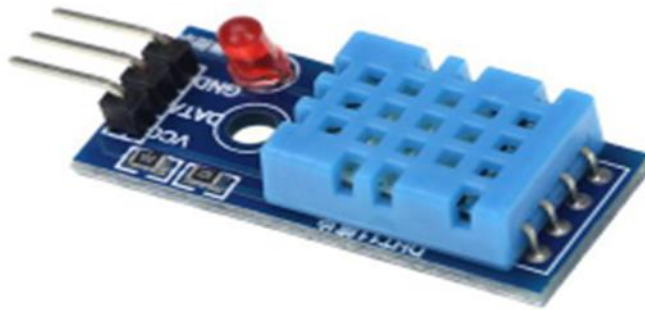


Hình 3.5 Sơ đồ chân ESP8266

### 3.2.2 Cảm biến nhiệt độ, độ ẩm

Thông số kỹ thuật :

- Điện áp: 3-5V (DC)
- Dải độ ẩm 20-90% RH, sai số 5%RH
- Dải nhiệt độ 0-50 độ C, sai số 2 độ C
- Khoảng cách truyền 20m.



**Hình 3.6 DHT 11**

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất).

### **3.2.3 Cảm biến ánh sáng**

Ưu điểm:

- Nhỏ gọn
- Chính xác
- Linh hoạt trong việc điều chỉnh độ nhạy của cảm biến

Các chân chức năng:

- VCC Cấp nguồn dương cho Cảm biến (3V3-5V)
- GND Nối Mass
- D0 Ngõ ra tín hiệu Digital I/O



**Hình 3.77 Cảm biến ánh sáng**

Cảm biến ánh sáng là các thiết bị quang điện chuyển đổi năng lượng ánh sáng (photon) cho dù ánh sáng nhìn thấy được hay tia hồng ngoại thành tín hiệu điện (electron).

Một cảm biến ánh sáng tạo ra tín hiệu đầu ra cho biết cường độ ánh sáng bằng cách đo năng lượng bức xạ tồn tại trong một dải tần số rất hẹp về cơ bản được gọi là “ánh sáng” và tần số từ “Hồng ngoại” đến “Có thể nhìn thấy” tới “Tia cực tím ” quang phổ ánh sáng.

### **3.2.4 Cảm biến khí gas**

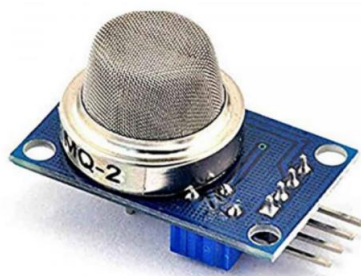
Thông số kỹ thuật:

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao
- Mạch đơn giản
- Ổn định khi sử dụng trong thời gian dài

Cảm biến khí gas MQ-2 sử dụng phần tử  $\text{SnO}_2$  có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của  $\text{SnO}_2$  sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện.

Cảm biến khí gas MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan ( $\text{CH}_4$ ) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.

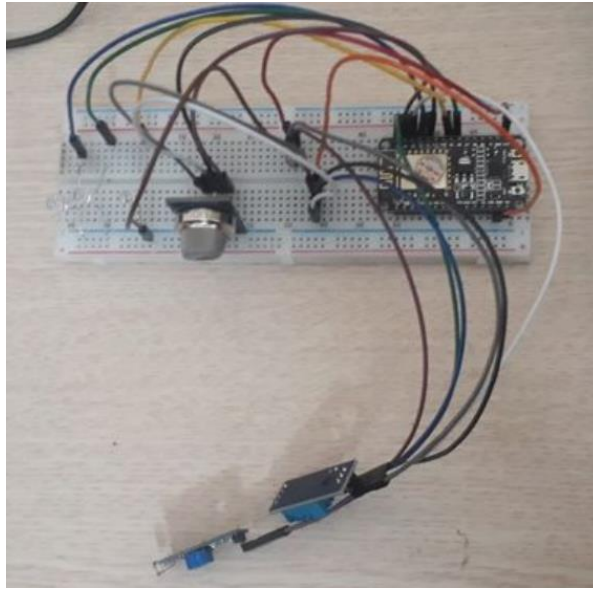
Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở.



**Hình 3.88** Cảm biến khí gas

### 3.3 Kiểm tra trên bo mạch trắng

Em thực hiện demo mạch trên bo mạch trắng và sử dụng ứng dụng di động để thực hiện các chức năng đã nêu trên .



**Hình 3.99 Test trên bo mạch trắng**

## KẾT LUẬN

Sau khi tiến hành thực hiện đề tài ,chúng em đã hoàn thành một số mục tiêu đề ra với những tự đánh giá như sau:

### **Đạt được:**

- Điều khiển được thiết bị qua SmartPhone.
- Giám sát trạng thái và hiển thị.
- Nhận diện được người lạ trong nhà.
- Hình thành được một hệ thống nhà thông minh cơ bản.

### **Hạn Chế:**

- Độ chính xác cần được cải thiện.
- Chức năng của hệ thống chỉ đáp ứng được ở mức độ nền tảng
- Tồn tại một số thiếu sót trong mô hình
- Giao diện người dùng còn đơn giản, hạn chế

Hướng phát triển: Trong tương lai, với cơ sở đã có, nhóm em sẽ phát triển đề tài theo các hướng như :

- Hoàn thiện giao diện người dùng App
- Phát triển hệ thống của thông minh sử dụng sinh trắc học .
- Phát triển hệ thống giám sát Camera .
- Tích hợp thêm nhiều thiết bị vào hệ thống điều khiển

Do còn hạn chế về thời gian và kiến thức nên đề tài của nhóm em còn nhiều thiếu sót. Nhóm mong nhận được sự góp ý chỉ bảo của thầy để đề tài của em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

- [1] N. H. Phước, “Điện tử Việt,” [Trực tuyến]. Available: <https://dientuviet.com/giao-tiep-dht11-voi-nodemcu-esp8266/>.
- [2] HowKTeam, “HowKTeam,” [Trực tuyến]. Available: <https://howkteam.vn/Course/Android-Basics/Gioi-thieu-lap-trinh-Android-11>.
- [3] ITNavi, “ITNavi,” [Trực tuyến]. Available: <https://itnavi.com.vn/blog/firebase-la-gi>.
- [4] H. V. Trinh, “Viblo,” [Trực tuyến]. Available: <https://viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-1-PjxMe6yNG4YL>.



## SOURCE CODE

Link github : [https://github.com/canhr101802/Project\\_App\\_Android\\_IOT.git](https://github.com/canhr101802/Project_App_Android_IOT.git)

Activity	Tên file code	Nội dung
Đăng nhập	<a href="#">activity_main.xml</a>	Bố cục giao diện đăng nhập
	<a href="#">MainActivity.java</a>	Khởi tạo activity đăng nhập, gọi hiển thị bố cục giao diện đăng nhập, truy xuất tới các id View của file xml, xử lý sự kiện trên giao diện đăng nhập
Trang chủ	<a href="#">user_main_activity.xml</a>	Bố cục giao diện trang chủ bao gồm 3 chức năng chính : Ánh sáng, Xem nhiệt độ, độ ẩm, Cảnh báo cháy
	<a href="#">UserMainActivity.java</a>	Gọi hiển thị bố cục giao diện trang chủ, xử lý sự kiện chuyển sang các chức năng khác
Ánh sáng	<a href="#">user_light.xml</a>	Bố cục giao diện bật tắt đèn
	<a href="#">User Light Activity.java</a>	Xử lý sự kiện bật tắt đèn gửi dữ liệu lên Google firebase
Xem nhiệt độ, độ ẩm	<a href="#">user_temhum.xml</a>	Bố cục giao diện xem nhiệt độ, độ ẩm trong nhà
	<a href="#">User TemHum Activity.java</a>	Xử lý sự kiện xem nhiệt độ, độ ẩm gửi dữ liệu lên Google firebase
Cảnh báo cháy	<a href="#">user_fire.xml</a>	Bố cục giao diện cảnh báo cháy

	<a href="#">User Fire Activity.java</a>	Xử lý sự kiện cảnh báo cháy
	<a href="#">dialog_timepicker.xml</a>	Bộ cục giao diện đồng hồ hẹn giờ bật tắt thiết bị
	<a href="#">Servicee.java</a>	Thực hiện chạy ngầm các dịch vụ nhằm mục đích có thể vận hành nhiều ứng dụng một lúc
	<a href="#">test.ino</a>	Kết nối phần cứng với google firebase, xử lý các yêu cầu gửi từ server bằng ngôn ngữ C