

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ THÔNG TIN



BÁO CÁO

TIỂU LUẬN CHUYÊN NGÀNH

Đề tài: Quản lý và điều khiển thiết bị IOT

SVTH: Lê Công Nghĩa 16110032

Nguyễn Việt Anh 16110007

GVHD: Ths. Trần Công Tú

TP.Hồ Chí Minh, tháng 12 năm 2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ THÔNG TIN



BÁO CÁO

TIỂU LUẬN CHUYÊN NGÀNH

Đề tài: Quản lý và điều khiển thiết bị IOT

SVTH:	Lê Công Nghĩa	16110032
	Nguyễn Việt Anh	16110007
GVHD:	Ths. Trần Công Tú	

TP. Hồ Chí Minh, tháng 12 năm 2019

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giáo viên hướng dẫn

(ký và ghi họ tên)

.....

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giáo viên phản biện

(ký và ghi họ tên)

.....

LỜI CẢM ƠN

Trong quá trình thực hiện đề án tiểu luận chuyên ngành Công nghệ thông tin, nhóm thực hiện đã may mắn nhận được rất nhiều sự quan tâm giúp đỡ của thầy cô và bạn bè.

Nhóm thực hiện xin gửi lời chân thành cảm ơn đến thầy Trần Công Tú, giảng viên khoa Công nghệ thông tin – trường đại học Sư phạm Kỹ thuật TP Hồ Chí Minh đã tận tình hướng dẫn, đóng góp ý kiến và chỉ bảo nhiệt tình trong suốt quá trình nhóm làm đề án.

Nhóm cũng xin chân thành cảm ơn các thầy cô trường Đại học Sư phạm Kỹ thuật TP Hồ Chí Minh nói chung, và các thầy cô của khoa Công nghệ thông tin và Đào tạo Chất lượng cao nói riêng đã dạy cho nhóm kiến thức về các môn học để nhóm có được nền tảng, cơ sở vững vàng hoàn thành đề án lần này.

Cuối cùng, nhóm xin cảm ơn các bạn khóa 16 ngành Công nghệ thông tin, đã luôn giúp đỡ trong suốt quá trình học tập và hoàn thành đề án.

TP.HCM, ngày 14 tháng 11 năm 2019

Nhóm thực hiện

TÓM TẮT

Đồ án tiểu luận chuyên ngành mà nhóm thực hiện là trang web gồm các chức năng: kết nối và điều khiển các thiết bị IOT trong gia đình, công ty và xí nghiệp, cài đặt hẹn giờ thời gian bật tắt thiết bị, theo dõi, thống kê và báo cáo về mức điện tiêu thụ của từng thiết bị, từng khu vực, cho phép chủ nhà chia sẻ quyền điều khiển một số thiết bị được chỉ định cho người khác. Điều này sẽ giúp người dùng có được một hệ thống toàn diện từ việc điều khiển, giám sát thiết bị điện của mình đến việc tối ưu hóa cách sử dụng các thiết bị đó sao cho có hiệu quả kinh tế và tiết kiệm chi phí. Đặc biệt từ việc giảm thiểu nguồn điện tiêu thụ sẽ góp phần rất lớn trong việc bảo vệ môi trường.

Hiện tại, phiên bản web của sản phẩm đang được phát triển theo MERN stack gồm ReactJs cho phần front end và NodeJs, MQTT ở back end cùng với mongoDB. Sản phẩm này ở phiên bản ứng dụng điện thoại đang được nghiên cứu và phát triển, sẽ có thêm các tính năng mới như live camera, ứng dụng machine learning để đưa ra thông báo và tự động xử lý.

TP.HCM, ngày 04 tháng 11 năm 2019

Nhóm thực hiện

MỤC LỤC

PHẦN MỞ ĐẦU.....	1
1. Lý do chọn đề tài	1
2. Mục tiêu của đề tài.....	1
3. Phạm vi nghiên cứu	2
4. Phương pháp nghiên cứu	2
PHẦN NỘI DUNG	3
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	3
1.1. Tổng quan về REST API	3
1.1.1. REST API là gì?	3
1.1.2. RESTful API hoạt động như thế nào?.....	3
1.1.3. Ưu điểm và nhược điểm	4
1.2. Tổng quan về NodeJs.....	4
1.2.1. NodeJs là gì?	4
1.2.2. Ứng dụng của Nodejs	4
1.2.3. Ưu điểm và nhược điểm	5
1.3. Tổng quan về MQTT	5
1.3.1. MQTT là gì?.....	5
1.3.2. Ứng dụng của MQTT trong IoT.....	5
1.4. Tổng quan về MongoDB.....	6
1.4.1. MongoDB là gì?	6
1.4.2. Ưu điểm và nhược điểm	6
1.5. Tổng quan về ReactJs	6
1.5.1. ReactJs là gì?.....	6
1.5.2. Sử dụng Components trong ReactJs?	7
CHƯƠNG 2: KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU	8
2.1. Khảo sát hiện trạng	8

2.2. Xác định yêu cầu	9
2.2.1. Giới thiệu về ứng dụng	9
2.2.2. Mô tả chức năng	9
2.2.3. Mô tả phi chức năng	9
2.3. UseCase Diagram	10
2.3.1. Lược đồ UseCase tổng quan	10
2.3.2. UseCase Login	10
2.3.3. UseCase Logout.....	11
2.3.4. UseCase Manage Devices	12
2.3.5. UseCase Control Devices	13
2.3.6. UseCase State Devices	14
2.3.7. UseCase Info Devices	15
2.3.8. UseCase History Devices.....	16
2.3.9. UseCase Add New Device	17
CHƯƠNG 3: THIẾT KẾ PHẦN MỀM	18
3.1. Thiết kế hệ thống	18
3.1.1. Lược đồ Lớp	18
3.1.2. Lược đồ sequence	20
3.2. Thiết kế dữ liệu.....	24
3.2.1. Bảng Users	24
3.2.2. Bảng Devices.....	25
3.2.3. Bảng Areas	26
3.3. Thiết kế giao diện	26
3.3.1. Dashboard	26
3.3.2. Kết nối thiết bị.....	27
3.3.3. Quản lý thiết bị	28
3.3.4. Chia sẻ thiết bị.....	28

3.3.5. Lịch sử thiết bị.....	29
3.3.6. Quản lý tài khoản.....	29
CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ.....	30
4.1. Cài đặt ứng dụng.....	30
4.2. Kiểm thử ứng dụng	30
4.2.1. Kiểm thử giao diện	30
4.2.2. Kiểm thử chức năng.....	31
PHẦN KẾT LUẬN	32
1. Những kết quả đạt được	32
2. Ưu điểm của đề tài.....	32
3. Hạn chế của đề tài.....	32
4. Hướng phát triển	32
TÀI LIỆU THAM KHẢO	33

DANH MỤC HÌNH VẼ

Hình 1.1-1. Mô hình RESTful API	3
Hình 1.1-2. Sơ đồ hoạt động của Web API	3
Hình 2.3-1. UseCase hệ thống.....	10
Hình 2.3-2. UseCase Login	10
Hình 2.3-3. UseCase Logout	11
Hình 2.3-4. UseCase ManageDevices	12
Hình 2.3-5. UseCase Control Devices	13
Hình 2.3-6. UseCase State Devices	14
Hình 2.3-7. UseCase Info Devices	15
Hình 2.3-8. UseCase Devices History	16
Hình 2.3-9. UseCase Add New Device	17
Hình 3.3-1. Class List Devices Diagram	18
Hình 3.3-2. Class Info Devices Diagram	18
Hình 3.3-3. Class Add Device Diagram	19
Hình 3.3-4. Class Share Devices Diagram	19
Hình 3.3-5. Login Sequence Diagram	20
Hình 3.3-6. Manage Devices Sequence Diagram	21
Hình 3.3-7. Control Devices Sequence Diagram	22
Hình 3.3-8. Add Devices Sequence Diagram	23
Hình 3.3-9. Share Devices Sequence Diagram	24
Hình 3.5-1. Giao diện DashBoard	27
Hình 3.5-2. Giao diện Kết nối thiết bị	27
Hình 3.5-3. Giao diện Quản lý thiết bị	28
Hình 3.5-4. Giao diện Chia sẻ thiết bị	28
Hình 3.5-5. Giao diện Lịch sử thiết bị	29
Hình 3.5-6. Giao diện Quản lý tài khoản	29

DANH MỤC BẢNG BIỂU

Bảng 2.3-1.Mô tả UC Login	11
Bảng 2.3-2.Mô tả UC Logout	11
Bảng 2.3-3.Mô tả UC Manage Devices.....	12
Bảng 2.3-4.Mô tả UC Control Devices	13
Bảng 2.3-5.Mô tả UC State Devices	14
Bảng 2.3-6.Mô tả UC Device Info	15
Bảng 2.3-7.Mô tả UC Devices History.....	16
Bảng 2.3-8.Mô tả UC Add New Device.....	17
Bảng 3.4-1.Bảng Users	24
Bảng 3.4-2.Bảng Devices.....	25
Bảng 3.4-3.Bảng Areas	26
Bảng 4.4-1. Test Case GUI.....	30
Bảng 4.4-2. Test Case Functions.....	31

DANH MỤC CÁC TỪ VIẾT TẮT

UC: Use Case

HTML: Hypertext Markup Language

REST: REpresentational State Transfer Application Programming Interface

API: Application Program Interface

IoT: Internet of Things

HTTP: HyperText Transfer Protocol

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, cùng với sự phát triển mạnh mẽ của công nghệ thì những ứng dụng của các giải pháp thông minh với các công nghệ mới nhất ngày càng được sử dụng rộng rãi không chỉ ở các công ty, xí nghiệp lớn mà ngay ở các gia đình cũng có thể sử dụng được. Cuộc cách mạng công nghệ 4.0 đã mang đến một làn gió mới về cách giao tiếp của con người với vạn vật xung quanh bằng trí tuệ nhân tạo (AI) và Internet of things (IoT). Có thể nói các hệ thống IoT đã đơn giản hóa cuộc sống của con người trong việc sử dụng các thiết bị được tích hợp trong hệ thống, các thiết bị cũng có thể tự động trao đổi thông tin và thực hiện các công việc giúp nâng cao chất lượng sống của con người.

Vì vậy, nhóm thực hiện quyết định sử dụng các kiến thức đã được học ở trường cùng với những công nghệ đang được sử dụng hiện nay trong lĩnh vực IoT để xây dựng hệ thống quản lý và điều khiển các thiết bị IoT.

2. Mục tiêu của đề tài

Thông qua đề tài “Xây dựng hệ thống quản lý và điều khiển thiết bị IoT”, nhóm thực hiện mong muốn đạt được những mục tiêu sau:

- Tìm hiểu nguyên lý và cách vận hành của hệ thống IoT
- Phân tích và thiết kế, quản lý một ứng dụng theo đúng quy trình phát triển phần mềm đã được học.
- Tìm hiểu và xây dựng được ứng dụng bằng MERN stack
- Áp dụng các công nghệ mới, mạnh mẽ đang được sử dụng rộng rãi hiện nay
- Đặc biệt là sự đóng góp của ứng dụng trong việc giải quyết các vấn đề về tối ưu hóa việc sử dụng các thiết bị IoT hiện nay: tiêu thụ năng lượng, tăng tuổi thọ thiết bị điện, ...

3. Phạm vi nghiên cứu

- Tìm hiểu nguyên lý hoạt động và cách xây dựng hệ thống IoT
- Tìm hiểu các công nghệ được sử dụng trong đề tài:
 - Nodejs
 - MQTT
 - ReactJs
 - MongoDB
 - RESTful API
 - Json Web Token
- Tìm hiểu và triển khai nghiệp vụ và quy trình phát triển phần mềm trong thực tế

4. Phương pháp nghiên cứu

- Phương pháp điều tra:

Nhóm thực hiện đã khảo sát các hệ thống và ứng dụng có mục đích tương tự trên thị trường để phân tích và xác định các năng phù hợp với hệ thống của mình.

- Phương pháp chuyên gia:

Nhóm thực hiện đã tham khảo ý kiến của giảng viên hướng dẫn và những giảng viên giảng dạy trong khoa cũng như các chuyên gia đã làm việc với các hệ thống IoT nhằm có được những giải pháp tối ưu nhất khi thực hiện.

- Phương pháp mô hình hóa:

Sau khi nghiên cứu và phân tích các ưu điểm và nhược điểm của hệ thống khác sau đó mô hình hóa và xây dựng lại các chức năng quan trọng nhất.[1]

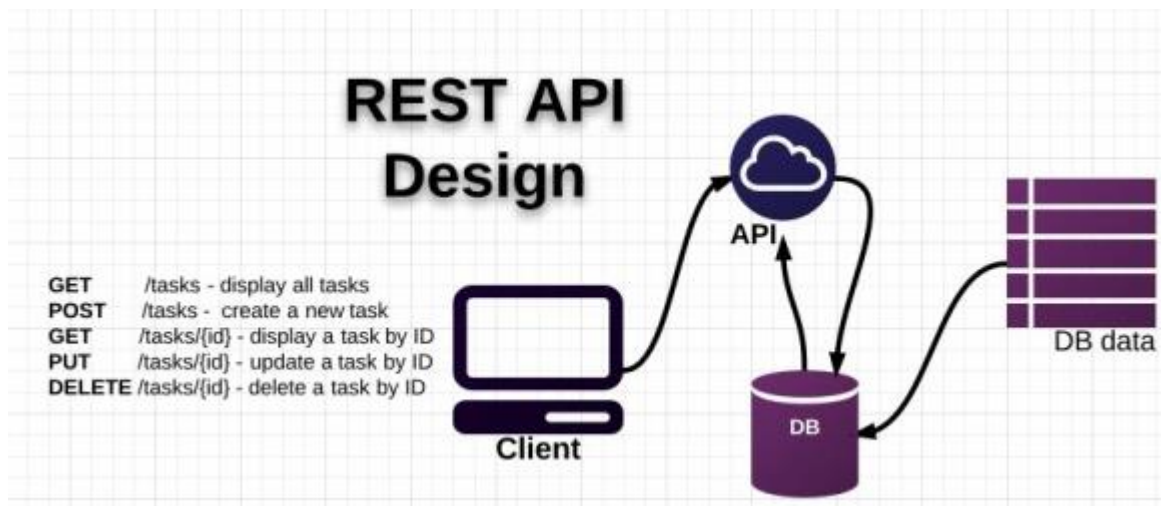
PHẦN NỘI DUNG

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về REST API

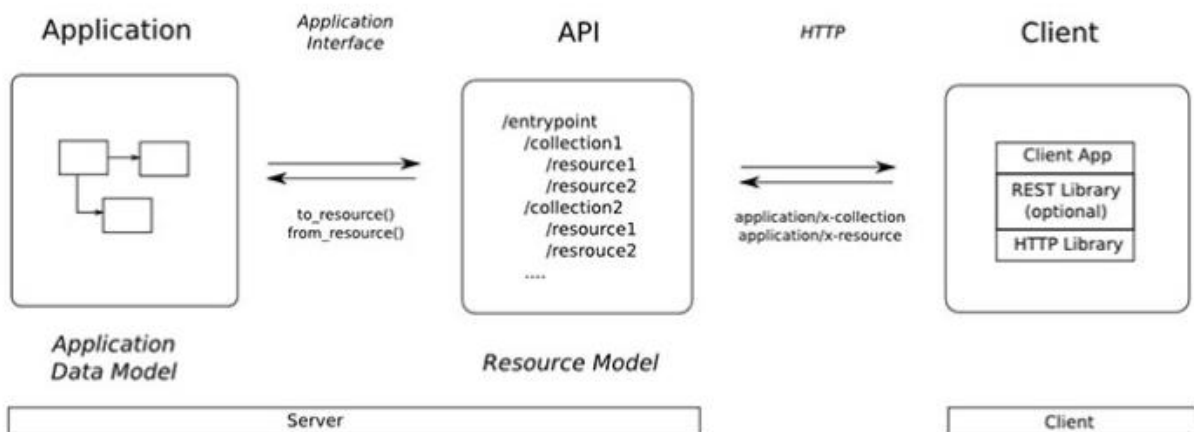
1.1.1. REST API là gì?

REST hay RESTful API là một tiêu chuẩn trong việc thiết kế API cho các ứng dụng web(Web services) để tiện cho việc quản lý resources.[2]



Hình 1.1-1. Mô hình RESTful API [2]

1.1.2. RESTful API hoạt động như thế nào?



Hình 1.1-2. Sơ đồ hoạt động của Web API [2]

Hiện nay, REST API thường sử dụng json là format chính thức để gửi và trả dữ liệu. Các hoạt động cơ bản như thêm, sửa, xóa dữ liệu thao tác trên một URL sử dụng những phương thức HTTP riêng: [3]

- **GET:** trả về dữ liệu dạng json
- **POST:** tạo mới dữ liệu
- **PUT:** cập nhật thông tin mới cho dữ liệu
- **DELETE:** xóa dữ liệu

1.1.3. Ưu điểm và nhược điểm

- **Ưu điểm:**
 - REST URL đại diện cho resource chứ không phải hành động
 - Trả về nhiều định dạng khác nhau: json, xml,..
 - Hiệu suất tốt, dễ phát triển
- **Nhược điểm:**
 - Bảo mật kém hơn SOAP
 - Dễ gây tốn tài nguyên nếu khối lượng thông tin lớn.
 - Chỉ hoạt động trên giao thức HTTP

1.2. *Tổng quan về NodeJs*

1.2.1. NodeJs là gì?

NodeJs là một mã nguồn mở sử dụng V8 engine JavaScript để thực thi mã, là một môi trường cho các máy chủ hay ứng dụng mạng. Nodejs chứa thư viện build-in cho phép các ứng dụng hoạt động như một webserver mà không cần các phần mềm như Nginx, IIS hay Apache.[4]

1.2.2. Ứng dụng của Nodejs

Các ứng dụng Nodejs thì viết bằng JavaScript theo kiến trúc hướng sự kiện và non – blocking để tối ưu hóa thông lượng ứng dụng: [4]

- Xây dựng websocket server (Chat server)

- Hệ thống thông báo (Notification như facebook hay twitter)
- Ứng dụng chạy thời gian thực.

1.2.3. Ưu điểm và nhược điểm

- **Ưu điểm:**

- Streaming data: nodejs sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
- Nodejs rất hiệu quả cho việc xây dựng các ứng dụng thời gian thực với sự hỗ trợ của HTML 5.
- Shelling tools unix Nodejs sẽ tận dụng tối đa Unix để hoạt động nên nó có thể xử lý hàng nghìn process và trả ra 1 luồng khiến cho hiệu suất hoạt động đạt mức tối đa và hiệu quả nhất.[4]

- **Nhược điểm:**

- Đối với các ứng dụng nặng tốn tài nguyên như encoding video, convert file, decoding encryption thì không nên sử dụng nodejs vì nodejs phải thông qua trình biên dịch nên sẽ tốn thêm thời gian rất lâu.[4]

1.3. Tổng quan về MQTT

1.3.1. MQTT là gì?

MQTT(Message Queue Telemetry Transport) là một giao thức truyền tin gọn nhẹ được thiết kế để giao tiếp giữa các thiết bị và hệ thống máy tính có khả năng hoạt động trong điều kiện đường truyền không ổn định. Vì vậy MQTT rất thích hợp để sử dụng cho các hệ thống IoT.[5]

MQTT trong một hệ thống IoT được hoạt động dựa trên cơ chế publish/subscribe. Quá trình này giống như chúng ta đăng kí và subscribe trên kênh Youtube. Mỗi client khi subscribe sẽ nhận được dữ liệu khi bất kỳ trạm khác gửi dữ liệu vào kênh đã đăng kí. [5]

1.3.2. Ứng dụng của MQTT trong IoT

Các ứng dụng của MQTT trong thực tế:

- Facebook Messenger sử dụng MQTT để tạo ra cuộc trò chuyện trực tuyến.
- Amazon sử dụng MQTT trong dự án Amazon IoT trên AWS
- Microsoft Azure IoT Hub sử dụng MQTT làm giao thức chính cho các tin nhắn từ xa.

1.4. Tổng quan về MongoDB

1.4.1. MongoDB là gì?

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở được viết bằng C++ và là một dạng NoSQL hàng triệu người dùng, sử dụng đa nền tảng cung cấp tốc độ truy xuất, hiệu năng cao, có khả năng mở rộng dễ dàng. [6]

1.4.2. Ưu điểm và nhược điểm

•Ưu điểm:

- Vì schema được sinh ra được sinh ra để nhóm các đối tượng vào cụm, dễ quản lý.
- Cấu trúc của đối tượng rõ ràng
- Không có các join phức tạp
- Khả năng mở rộng cực lớn, cho phép thực hiện replication và sharding.
- Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy xuất dữ liệu nhanh hơn. Việc cập nhật thực hiện nhanh

•Nhược điểm:

- Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu bộ nhớ RAM lớn.
- Mọi thay đổi về dữ liệu mặc định đều được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao

1.5. Tổng quan về ReactJs

1.5.1. ReactJs là gì?

ReactJs là một thư viện của Javascript do facebook phát triển để xây dựng front end có các thành phần có khả năng tái sử dụng. Nó cho phép những nhà phát triển web

tạo ra giao diện người dùng một cách nhanh chóng. Phần render của ReactJs thường sử dụng các Component chứa các thẻ HTML.

Reactjs có khả năng cập nhật nhanh những thay đổi về dữ liệu. react không phải là framework nên nó giảm các ràng buộc thư viện với nhau. Tuy nhiên, React chỉ phục cho tầng View nên khi muốn áp dụng vào các framework MVC sẵn có thì phải cấu hình lại. [7]

1.5.2. Sử dụng Components trong ReactJs?

React được xây dựng xung quanh các component, chúng ta có thể tái sử dụng các component ở nhiều nơi với các trạng thái(state) và thuộc tính(prop) khác nhau. Trong các component thì hàm render dùng để xử lý việc tạo thẻ html thông qua virtual DOM. Do đó, mọi thay đổi của dữ liệu ở bất kỳ thời điểm nào thì cũng được cập nhật và Virtual DOM update ngay lập tức. [8]

CHƯƠNG 2. KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU

2.1. *Khảo sát hiện trạng*

Cuộc cách mạng công nghệ 4.0 mở ra rất nhiều cơ hội trong lĩnh vực IoT, trong đó nhu cầu sử dụng nhà thông minh ngày một tăng mạnh. Có thể nói thị trường sản xuất các thiết bị thông minh cũng như việc cung cấp các giải pháp thông để tối ưu việc sử dụng các thiết bị thông minh đó cũng rất lớn và hấp dẫn.

Hiện nay trên thị trường đang có các sản phẩm, giải pháp IoT khá hoàn thiện và ổn định tuy nhiên vẫn còn một số điểm hạn chế mà nhóm đã khảo sát và thống kê lại được:

- Nổi bật trong việc cung cấp giải pháp toàn diện về IoT có thể kể đến AWS, hệ thống AWS IoT cung cấp các dịch vụ phục vụ từ nhà máy xí nghiệp lớn cho tới nhà thông minh, khả năng bảo mật nhiều lớp và khả năng tích hợp API vượt trội. AWS cung cấp phần mềm thiết bị (FreeRTOS, GreenGrass,...), dịch vụ kiểm soát (IoT core, Device Defender, Things graph), dịch vụ dữ liệu (IoT Analytics, IOT Events,...)
- Cũng đến từ một ông lớn khác đó là hệ thống Cisco IoT với các giải pháp phù hợp chủ yếu cho doanh nghiệp trong các lĩnh vực:
 - Năng lượng: cải thiện hiệu suất, giảm chi phí, tăng an toàn.
 - Chế tạo: kết nối máy móc, giám sát và quản lý từ xa.
 - Giao thông vận tải: tối ưu hóa hoạt động đội xe, giảm chi phí,..
- Ở thị trường Việt Nam, tập đoàn công nghệ SBD cũng cung cấp các giải pháp IoT như:
 - EasyLink Platform: cung cấp nền tảng phát triển ứng dụng IoT nhanh chóng,...
 - Smart-Rom Monitoring: giám sát phòng thông minh sử dụng công nghệ Zigbee.
 - E-parking: bãi giữ xe thông minh giúp quản lý đồng bộ nhiều bãi xe cùng lúc, dễ dàng tìm kiếm chỗ trống, thanh toán linh hoạt.
 - eGOV: hệ thống chính quyền số.

2.2. Xác định yêu cầu

2.2.1. Giới thiệu về ứng dụng

Ứng dụng quản lý và điều khiển thiết bị IoT là một giải pháp áp dụng cho những khu vực có sử dụng các thiết bị điện và cần sự quản lý thông minh để giảm thiểu năng lượng tiêu hao và giúp tiết kiệm chi phí vận hành.

Nhóm thực hiện xây dựng trên nền tảng web nên dễ dàng cho người dùng có thể truy cập vào hệ thống từ mọi thiết bị có kết nối internet bằng tài khoản đã đăng kí trên hệ thống.

2.2.2. Mô tả chức năng

Người dùng sau khi đăng nhập có thể sử dụng được các chức năng sau:

- Điều khiển các thiết bị với các thao tác bật tắt, tăng giảm cường độ một cách dễ dàng
- Thêm mới hoặc xóa bỏ các thiết bị IoT trên hệ thống
- Chia sẻ quyền sử dụng các thiết bị giữa các người dùng trong hệ thống
- Xem báo cáo về lịch sử của thiết bị và mức năng lượng đã tiêu thụ, cũng như có biểu đồ so sánh trực quan theo thời gian.

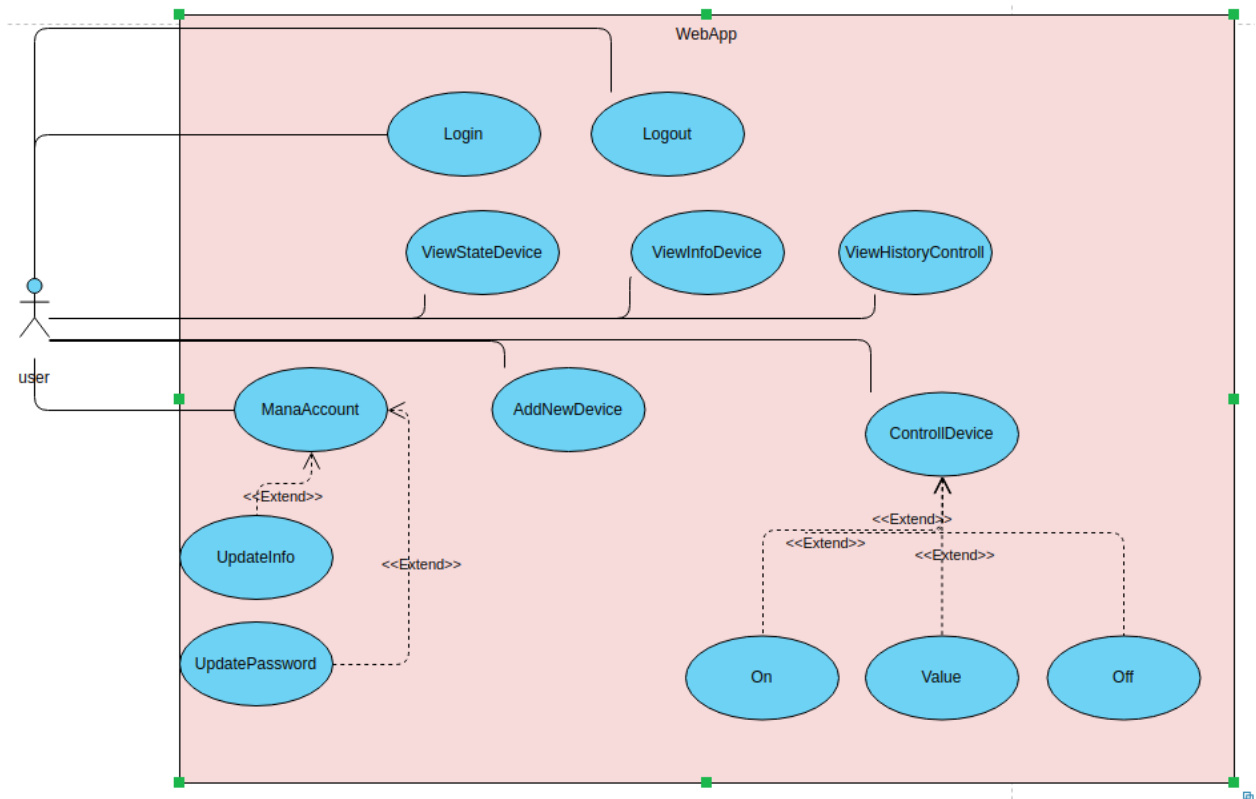
2.2.3. Mô tả phi chức năng

Do hệ thống đòi hỏi xử lý thời gian thực và sẵn sàng phục vụ lượng lớn người dùng và thiết bị nên các yêu cầu về hiệu năng, UX/UI, chi phí duy trì cũng được chú trọng:

- Hiệu năng: Tốc độ load trang dưới 3s, thao tác trên giao diện có độ trễ phản hồi dưới 2s
- UI: giao diện đơn giản với màu sắc tương phản
- UX: người dùng chỉ cần tốn rất ít thao tác đơn giản thì đã nhanh chóng đạt được mục đích mà mình cần.

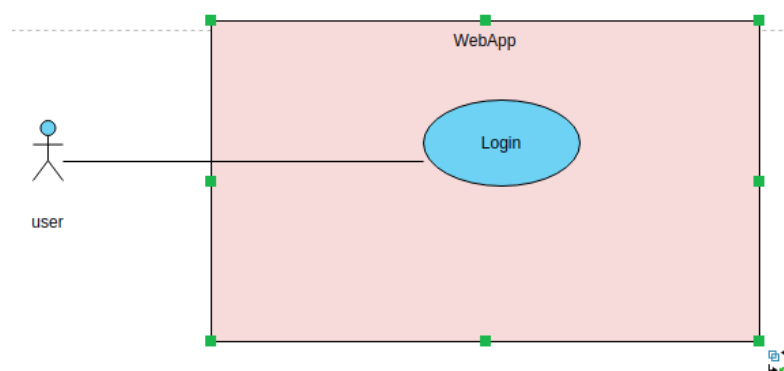
2.3. UseCase Diagram

2.3.1. Lược đồ UseCase tổng quan



Hình 2.3-1. UseCase hệ thống

2.3.2. UseCase Login

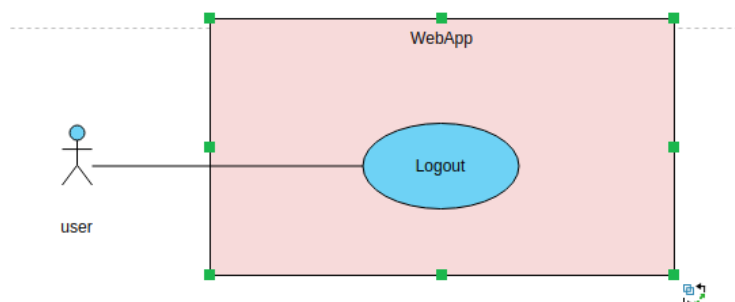


Hình 2.3-2. UseCase Login

Bảng 2.3-1.Mô tả UC Login

User Case ID	UC_1
Name	Login
Goal	Đăng nhập vào hệ thống
Actors	User
Pre-conditions	Load được form đăng nhập
Post-conditions	
Main Flow	<ol style="list-style-type: none"> 1. Nhập user name và password 2. Click nút đăng nhập
Exception	Người dùng không tồn tại Mật khẩu sai
Open Issues	N/A

2.3.3. UseCase Logout



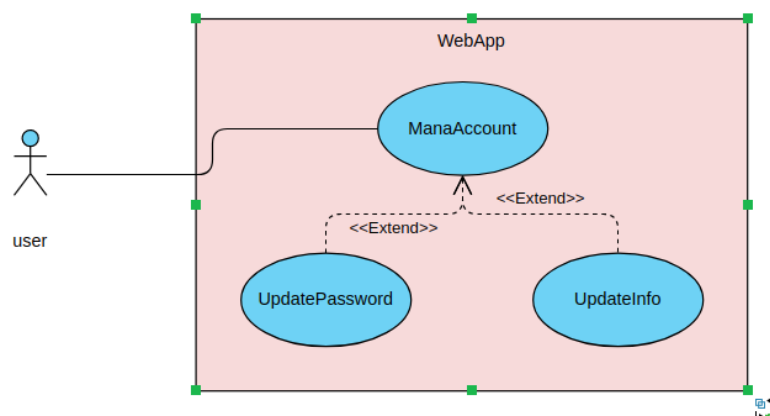
Hình 2.3-3. UseCase Logout

Bảng 2.3-2.Mô tả UC Logout

User Case ID	UC_2
Name	Logout

Goal	Đăng xuất khỏi hệ thống
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống
Post-conditions	
Main Flow	1. Click vào Logout
Exception	Server không phản hồi
Open Issues	N/A

2.3.4. UseCase Manage Devices



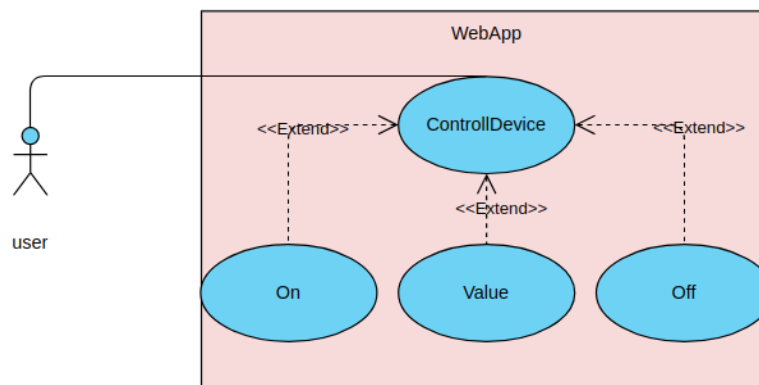
Hình 2.3-4. UseCase ManageDevices

Bảng 2.3-3.Mô tả UC Manage Devices

User Case ID	UC_3
Name	Manage Devices
Goal	Quản lý thiết bị
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống

Post-conditions	
Main Flow	1. Click vào Quản lý thiết bị
Exception	Server không phản hồi
Open Issues	N/A

2.3.5. UseCase Control Devices

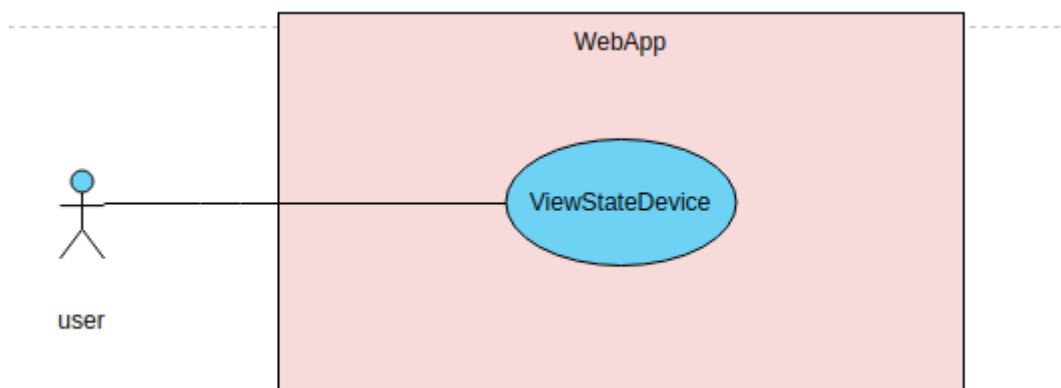


Hình 2.3-5. UseCase Control Devices

Bảng 2.3-4.Mô tả UC Control Devices

User Case ID	UC_3
Name	Control Devices
Goal	Điều khiển thiết bị
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống
Post-conditions	
Main Flow	1. Click vào Quản lý thiết bị
Exception	Server không phản hồi
Open Issues	N/A

2.3.6. UseCase State Devices

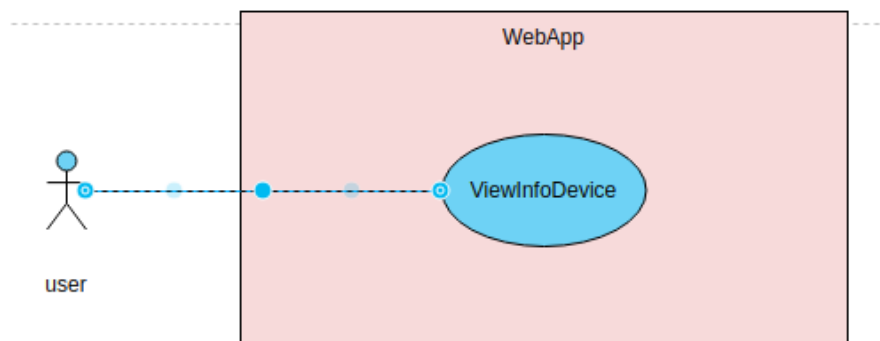


Hình 2.3-6. UseCase State Devices

Bảng 2.3-5. Mô tả UC State Devices

User Case ID	UC_4	
Name	State Devices	
Goal	Xem trạng thái thiết bị	
Actors	User	
Pre-conditions	Đã đăng nhập vào hệ thống	
Post-conditions		
Main Flow	1. Click vào Dashboard	
Exception	Server không phản hồi	
Open Issues	N/A	

2.3.7. UseCase Info Devices

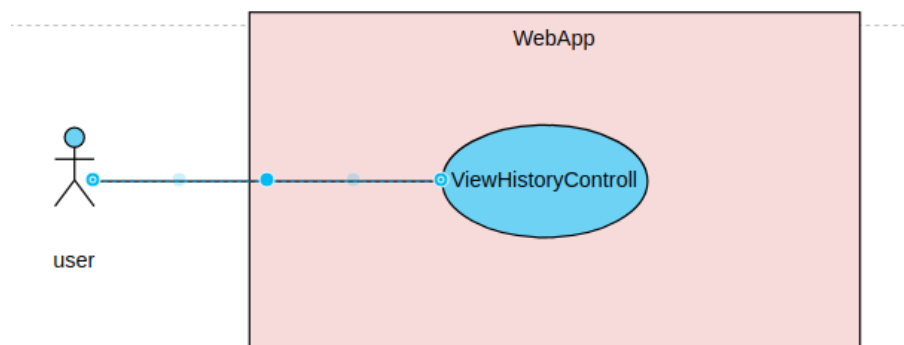


Hình 2.3-7. UseCase Info Devices

Bảng 2.3-6.Mô tả UC Device Info

User Case ID	UC_5
Name	Info Devices
Goal	Xem thông tin thiết bị
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống
Post-conditions	
Main Flow	1. Click vào Thông tin thiết bị
Exception	Server không phản hồi
Open Issues	N/A

2.3.8. UseCase History Devices

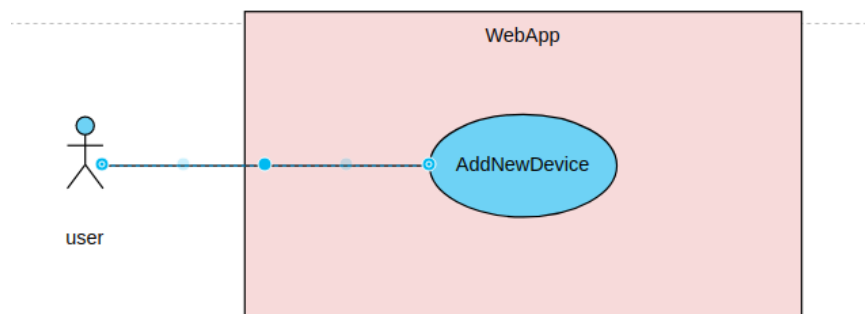


Hình 2.3-8. UseCase Devices History

Bảng 2.3-7.Mô tả UC Devices History

User Case ID	UC_6
Name	History Devices
Goal	Xem lịch sử của thiết bị
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống
Post-conditions	
Main Flow	1. Click vào Lịch sử thiết bị
Exception	Server không phản hồi
Open Issues	N/A

2.3.9. UseCase Add New Device



Hình 2.3-9. UseCase Add New Device

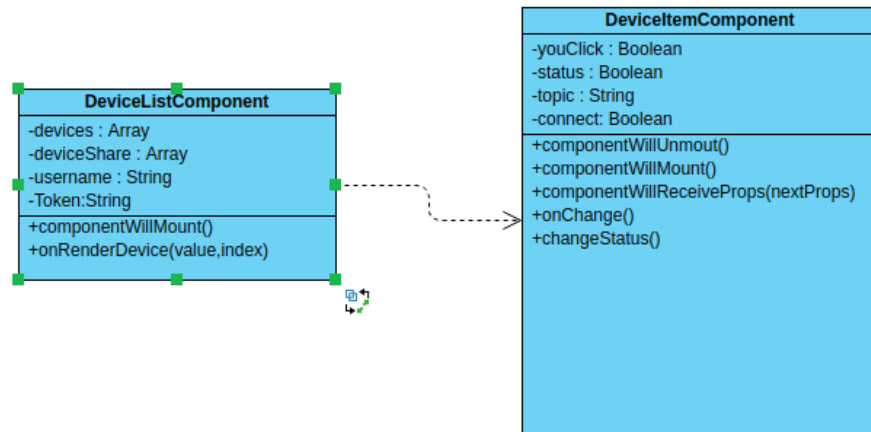
Bảng 2.3-8.Mô tả UC Add New Device

User Case ID	UC_7
Name	Add New Devices
Goal	Xem trạng thái thiết bị
Actors	User
Pre-conditions	Đã đăng nhập vào hệ thống
Post-conditions	
Main Flow	1. Click vào Thêm thiết bị
Exception	Server không phản hồi
Open Issues	N/A

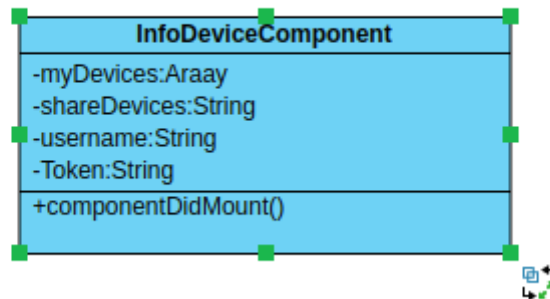
CHƯƠNG 3. THIẾT KẾ PHẦN MỀM

3.1. Thiết kế hệ thống

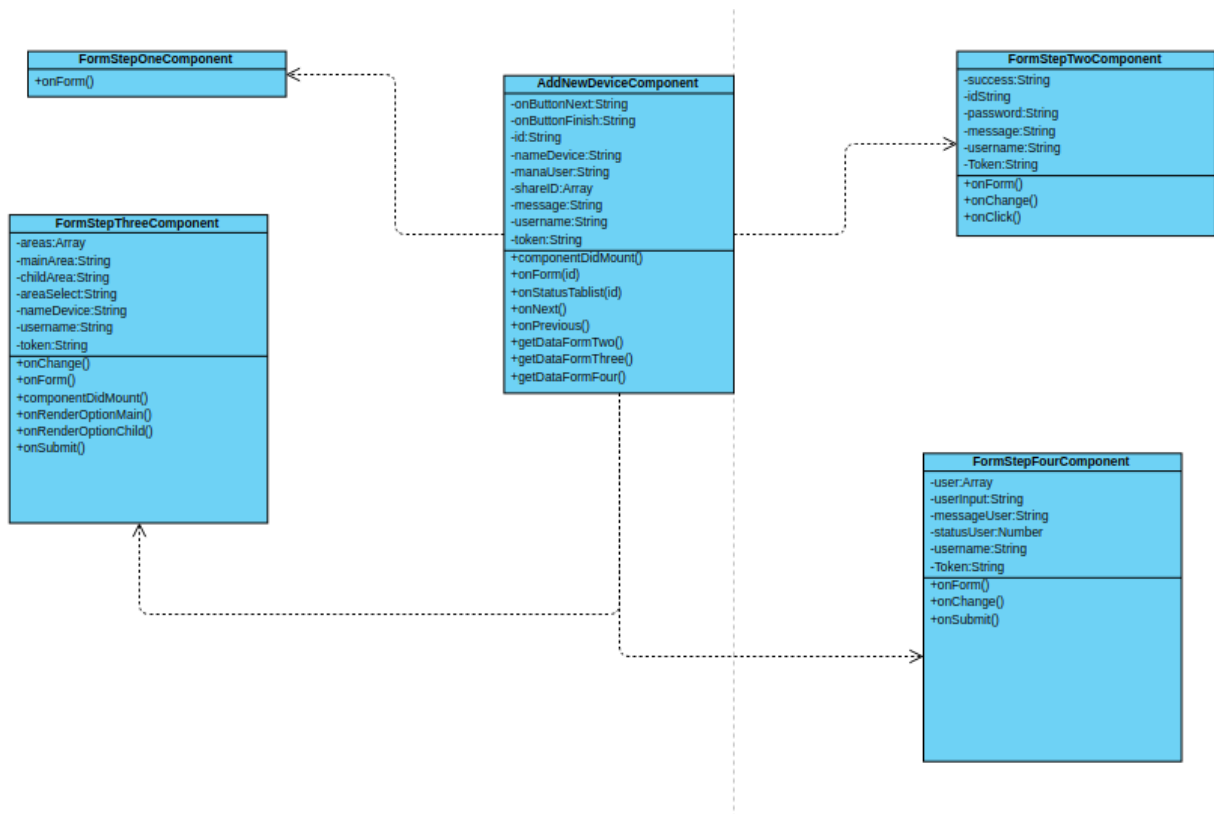
3.1.1. Lược đồ Lớp



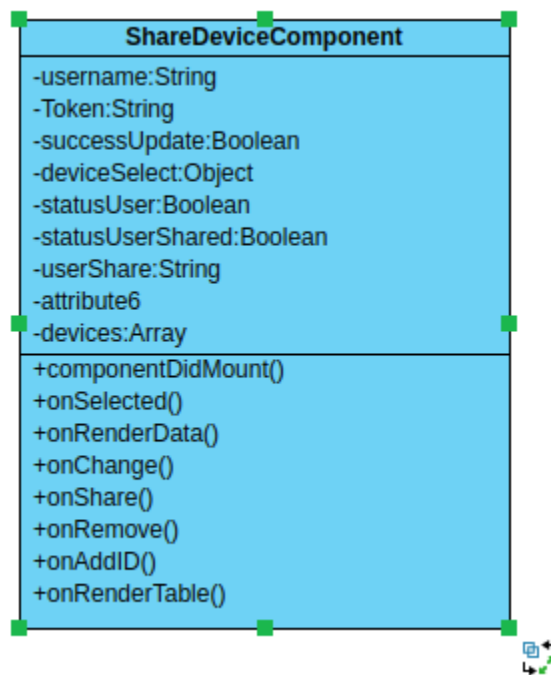
Hình 3.1-1. Class List Devices Diagram



Hình 3.1-2. Class Info Devices Diagram

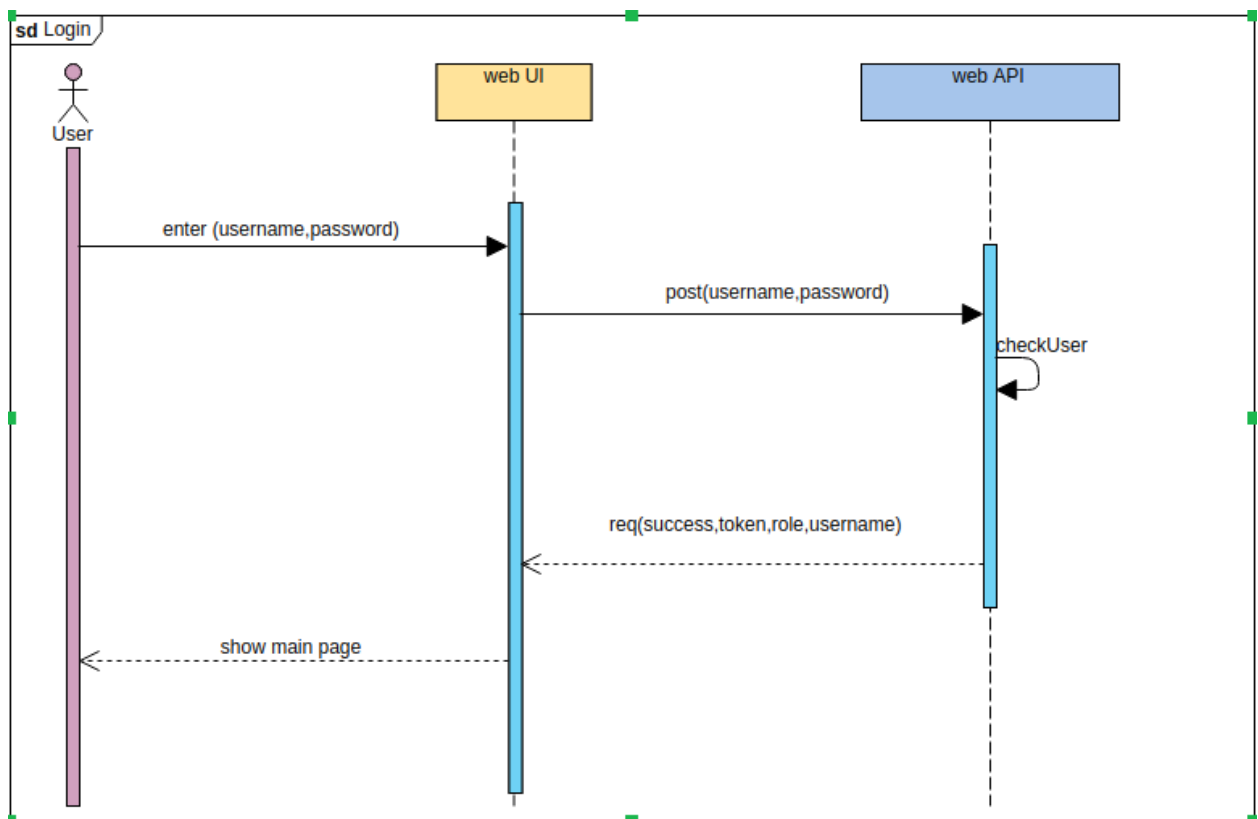


Hình 3.1-3.Class Add Device Diagram

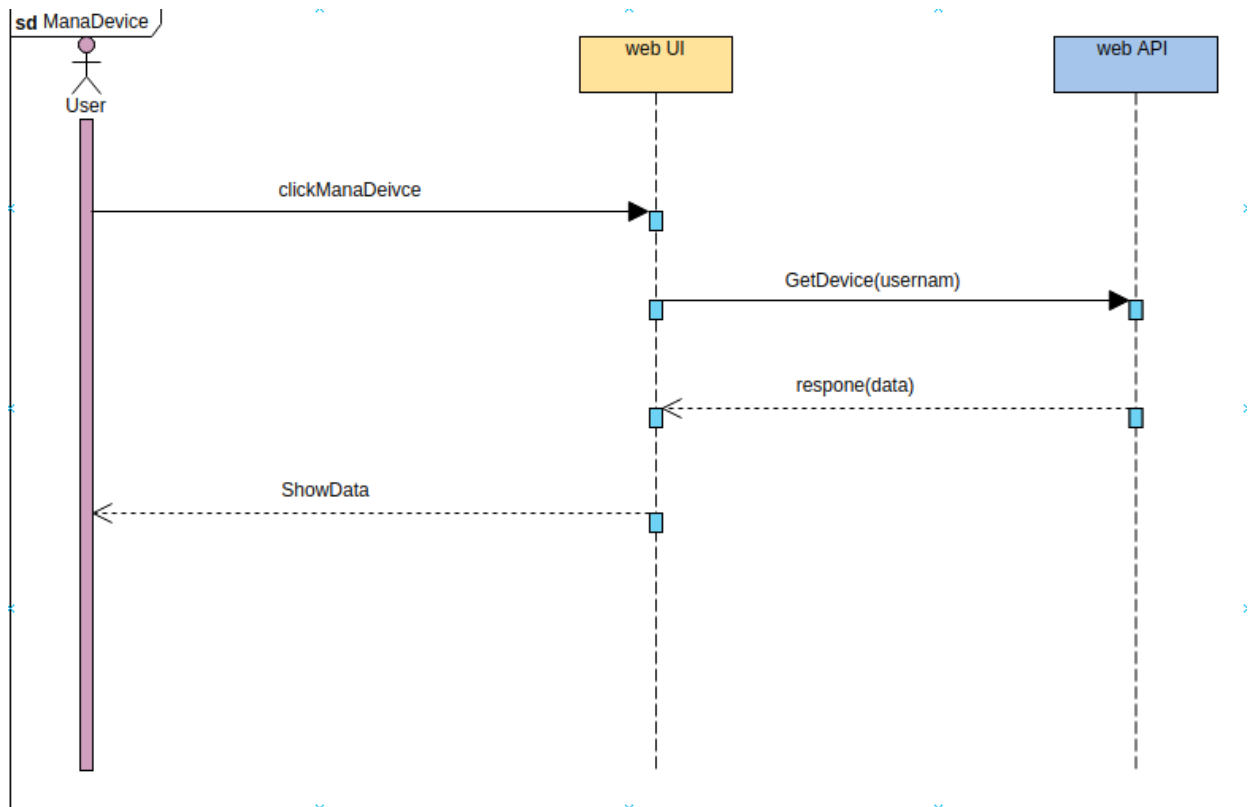


Hình 3.1-4. Class Share Devices Diagram

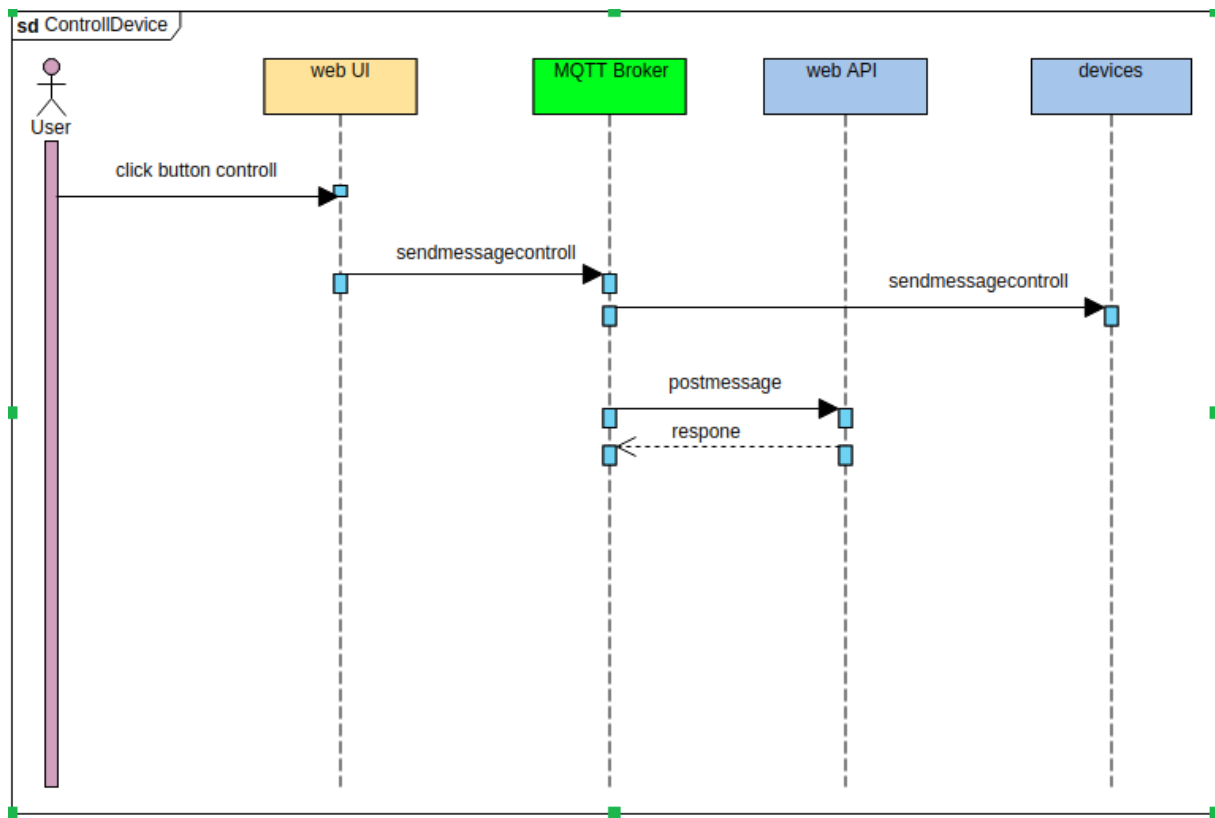
3.1.2. Lược đồ sequence



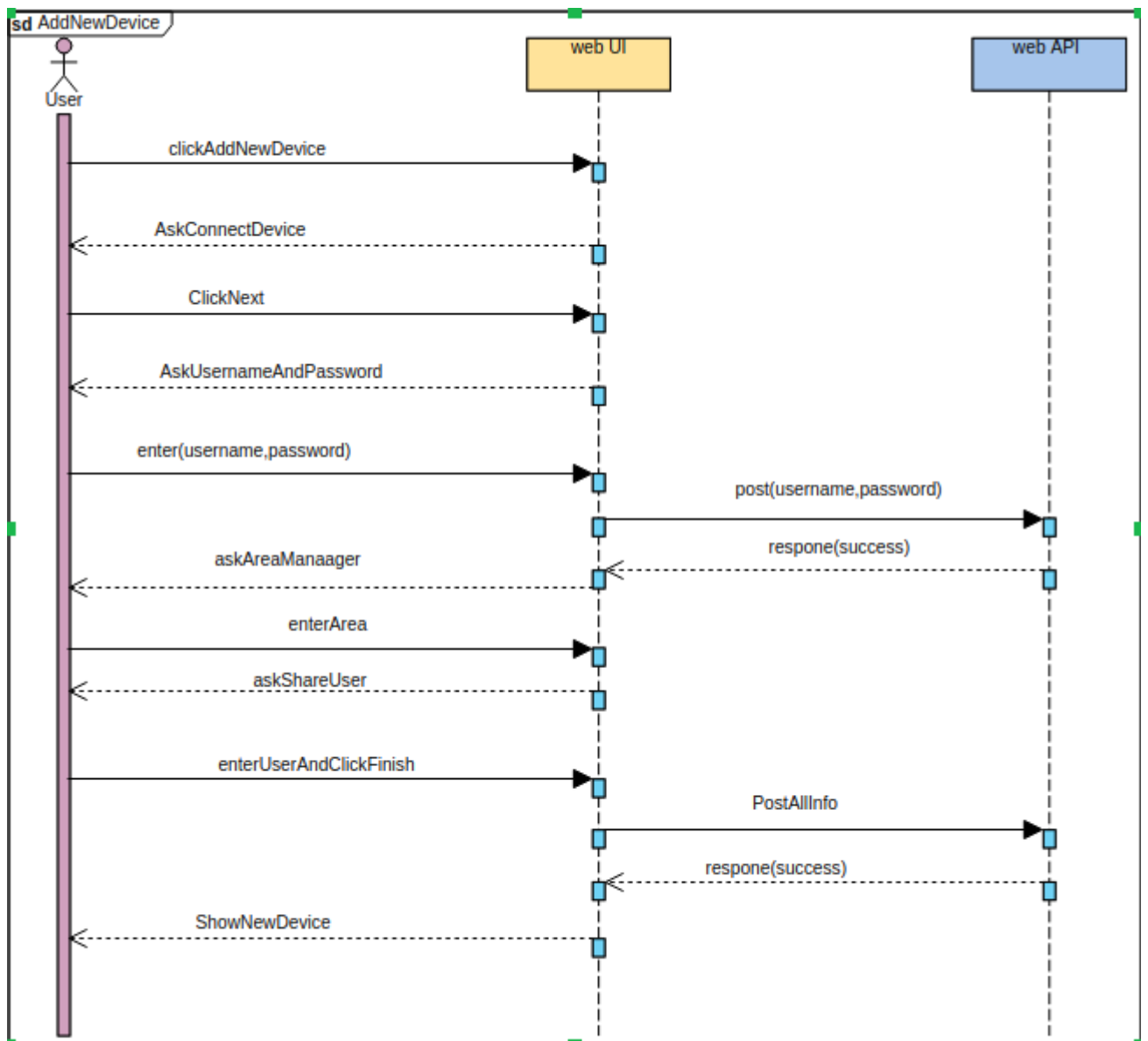
Hình 3.1-5. Login Sequence Diagram



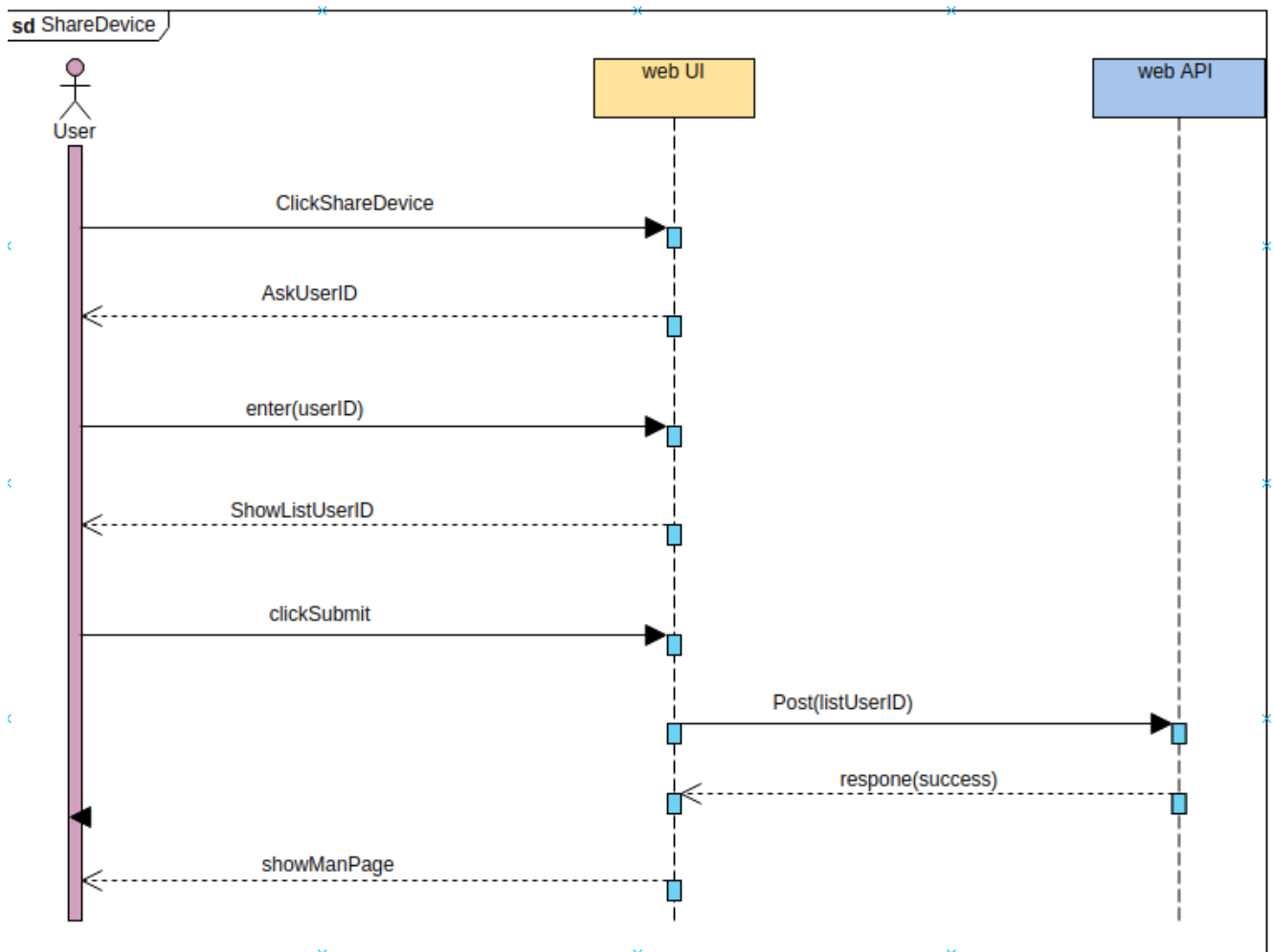
Hình 3.1-6. Manage Devices Sequence Diagram



Hình 3.1-7. Control Devices Sequence Diagram



Hình 3.1-8. Add Devices Sequence Diagram



Hình 3.1-9. Share Devices Sequence Diagram

3.2. Thiết kế dữ liệu

3.2.1. Bảng Users

Bảng 3.2-1. Bảng Users

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	_id	Đặc trưng cho mỗi user trong hệ thống	ObjectId	Khóa chính
2	Username	Tên đăng nhập	String	Not null
3	Password	Mật khẩu	String	Not null

4	Address	Địa chỉ	String	
5	City	Thành phố	String	
6	Email	Địa chỉ mail	String	
7	Firstname	Họ	String	Not null
8	Lastnam	Tên	String	Not null
9	Phone	SĐT	String	
10	State	Trạng thái		

3.2.2. Bảng Devices

Bảng 3.2-2.Bảng Devices

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	_id	Đặc trưng cho mỗi user trong hệ thống	ObjectId	Khóa chính
2	areaControl	Danh sách khu vực	Array	
3	shareID	UserId được chia sẻ	Array	
4	Code	Mã thiết bị	String	Not null
5	password	Mật khẩu thiết bị	String	Not null
6	Type	Loại thiết bị	String	
7	Data	Thông tin thiết bị	Object	Not null
8	Connect	Trạng thái kết nối	Boolean	Not null
9	manaUser	User sở hữu thiết bị	String	
10	Name	Tên thiết bị	String	Not null

3.2.3. Bảng Areas

Bảng 3.2-3. Bảng Areas

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	_id	Đặc trưng cho mỗi user trong hệ thống	ObjectId	Khóa chính
2	RoomId	Mã phòng	String	Not null
3	Name	Tên phòng	String	Not null
4	DeviceControl	Danh sách thiết bị	Array	
5	ManaUser	Chủ sở hữu	String	
6	ShareUser	User được chia sẻ	String	

3.3. *Thiết kế giao diện*

3.3.1. Dashboard

Sau khi đăng nhập thành công với tài khoản đã đăng kí trên hệ thống, người dùng sẽ ngay lập tức vào được giao diện dashboard. Màn hình hiển thị các thông số thống kê về số thiết bị kết nối, lượng điện đã tiêu thụ, hiệu suất hoạt động của thiết bị,...

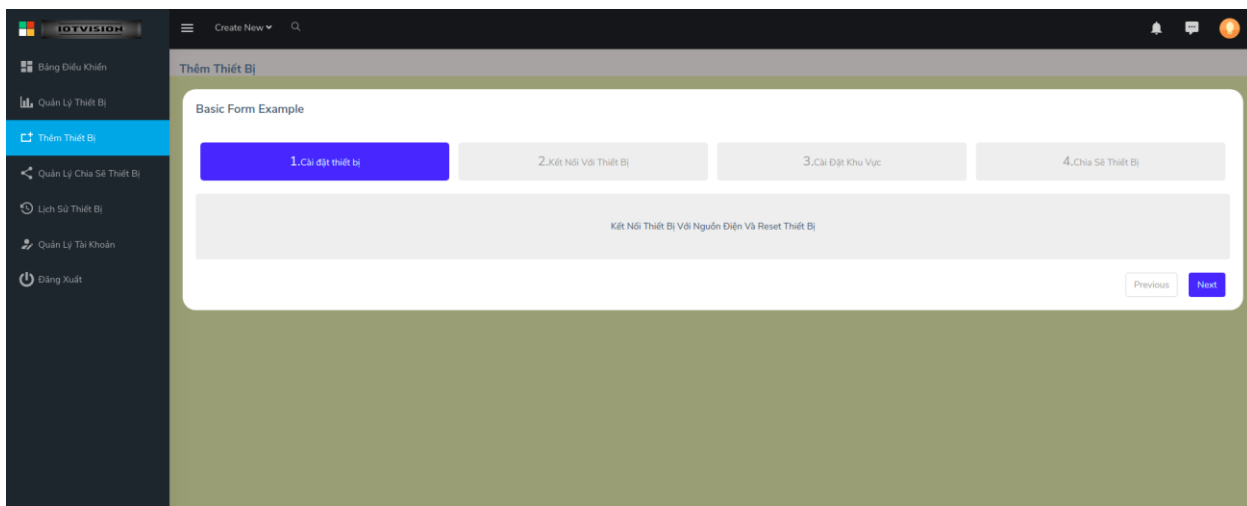
Tình trạng của các thiết bị điện hiển thị phía dưới được chia theo phân vùng, các thiết bị đã được kết nối sáng lên thì có thể điều khiển các thiết bị đó.



Hình 3.3-1. Giao diện DashBoard

3.3.2. Kết nối thiết bị

Các thiết bị đã được kết nối với nguồn điện ở trạng thái sẵn sàng kết nối thì người dùng có thể kết nối qua các bước sau:



Hình 3.3-2. Giao diện Kết nối thiết bị

3.3.3. Quản lý thiết bị

Các thiết bị đã được kết nối và hệ thống và được liệt kê theo thiết bị của mình sở hữu và thiết bị được chia sẻ:

Thiết Bị Của Tôi						
Mã	Mật khẩu thiết bị	Tên	Trạng thái	Loại	Mã người tạo	Tên khu vực
TB010	tbioT010	máy lạnh phòng ngủ	Đang kết nối	air-conditioner	vietanh98	Chùa Thuộc Khu Vực
TB011	tbioT011	đèn nhà bếp	Đang kết nối	light	vietanh98	Chùa Thuộc Khu Vực

Thiết Bị Được Chia Sẻ						
Mã	Mật khẩu thiết bị	Tên	Trạng thái	Loại	Mã người tạo	Tên khu vực
TB002	tbioT002	quạt phòng khách	Đang kết nối	fan	lcnghia95	Phòng Khách
TB003	tbioT003	đèn phòng ngủ	Đang kết nối	lamp	lcnghia95	Phòng Ngủ
TB004	tbioT004	máy lạnh phòng ngủ	Đang kết nối	air-conditioner	lcnghia95	Phòng Ngủ

Hình 3.3-3. Giao diện Quản lý thiết bị

3.3.4. Chia sẻ thiết bị

Người dùng có thể chia sẻ các thiết bị của mình thông qua username mà người được chia sẻ đăng ký với hệ thống.

Chia Sẻ Thiết Bị

Tài Khoản Muốn Chia Sẻ *

Chọn Thiết Bị Muốn Chia Sẻ

máy lạnh phòng ngủ

Thêm

Danh Sách Đã Chia Sẻ

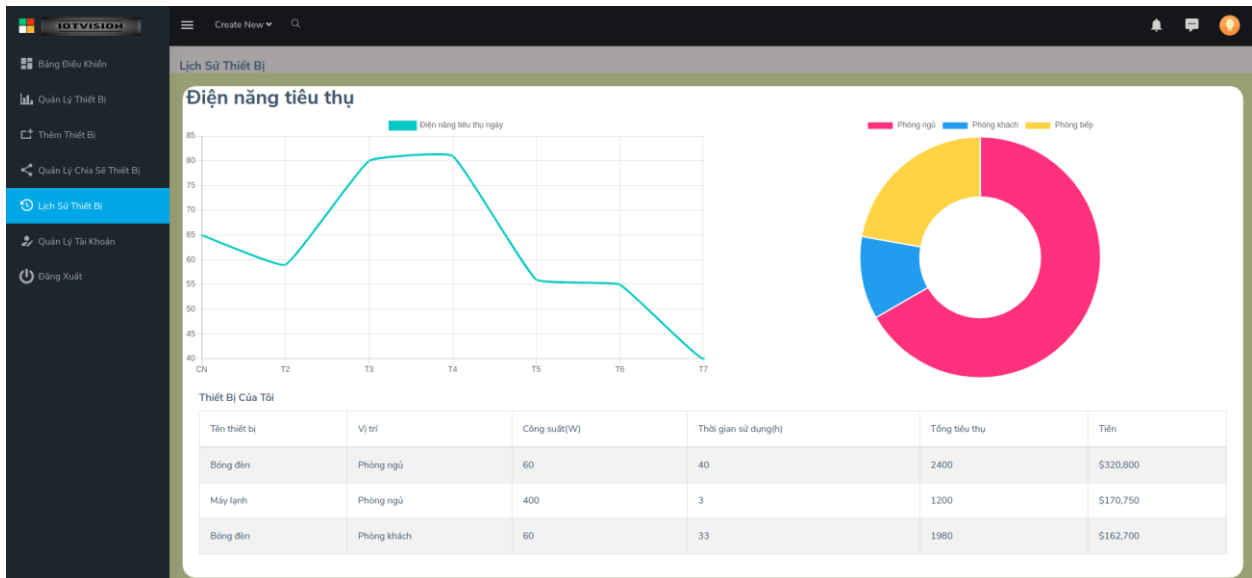
Tên Thiết Bị	Tên Người Dùng	Xóa Chia Sẻ
máy lạnh phòng ngủ	lcnghia95	Xóa

Lưu Thay Đổi

Hình 3.3-4. Giao diện Chia sẻ thiết bị

3.3.5. Lịch sử thiết bị

Thời lượng sử dụng các thiết bị được ghi lại và thống kê theo dạng bảng biểu và biểu đồ.



Hình 3.3-5. Giao diện Lịch sử thiết bị

3.3.6. Quản lý tài khoản

Những thông tin về tài khoản người dùng sau khi đăng kí có thể khai báo thêm hoặc chỉnh sửa ở màn hình này.

Thông Tin Tài Khoản

Tên Tài Khoản: vietanh98 | My Name Is: Admin | Tên: Admin

Email: superAdmin@gmail.com | Số Điện Thoại: 0333865088

Thành Phố: Hồ Chí Minh | Quận/Huyện: Quận 12 | Địa Chỉ: 65/1A Tô 16, Khu Phố 1 (gần đường An Phú Đông 10)

[Edit](#) [Lưu](#)

Hình 3.3-6. Giao diện Quản lý tài khoản

CHƯƠNG 4. CÀI ĐẶT VÀ KIỂM THỬ

4.1. Cài đặt ứng dụng

Ứng dụng được chia thành 3 repositories là: Backend, Frontend, MQTT và mỗi repository được deploy độc lập. Các packages đã được sử dụng trong từng repository:

Bảng 4-1. Dependency Packages

Back-end	Front-end
<ul style="list-style-type: none">○ Body-parser: 1.19.0○ Express: 4.17.1○ Mongoose: 5.7.3○ Nodemon: 1.19.3	<ul style="list-style-type: none">○ Axios: 0.19.0○ Mqtt-react: 0.0.3○ React-redux: 7.1.3○ Redux: 4.0.4○ React-chartjs-2: 2.8.0

Người dùng truy cập vào địa chỉ: <http://iotvision.ddnsking.com/> để sử dụng.

4.2. Kiểm thử ứng dụng

Sau khi deploy sản phẩm, nhóm thực hiện tiến hành test lại các màn hình giao diện và chức năng của các component có đúng với thiết kế và mong muốn hay không.

4.2.1. Kiểm thử giao diện

Bảng 4.2-1. Test Case GUI

Test case	Expected Result	Actual Result	Status	Notes
UI_LoginPage	Màn hình login	Match thiết kế	Pass	
UI_Dashboard	Hình 3.3-1	Màn hình dashboard	Pass	
UI_ManagePage	Hình 3.3-3	Trang Quản lý thiết bị	Pass	
UI_SharePage	Hình 3.3-4	Trang Chia sẻ	Pass	
UI_AddPage	Hình 3.3-2	Trang thêm mới	Pass	

UI_HistoryPage	Hình 3.3-5	Trang lịch sử	Pass	
----------------	------------	---------------	------	--

4.2.2. Kiểm thử chức năng

Bảng 4.2-2. Test Case Functions

Test case	Expected Result	Actual Result	Status	Notes
Func_Login	Chuyển đến dashboard	Chuyển đến dashboard	Pass	
Func_AddNew	Thêm thành công	Thêm thành công	Pass	
Func_ShareDevices	Chia sẻ thành công	Chia sẻ thành công	Pass	

PHẦN KẾT LUẬN

1. *Những kết quả đạt được*

Theo như kế hoạch ban đầu có đặt ra các tính năng cơ bản như đăng kí tài khoản, quản lý tài khoản người dùng, kết nối và điều khiển thiết bị IOT, quản lý được các thiết bị đã kết nối, chia sẻ quyền điều khiển với người dùng khác trong hệ thống, có thể xem lại được lịch sử sử dụng các thiết bị và quan sát thống kê dưới dạng biểu đồ để dễ dàng có sự so sánh và điều chỉnh sao cho mức tiêu thụ điện là ít nhất mà vẫn đáp ứng đủ nhu cầu sử dụng của gia đình.

2. *Ưu điểm của đề tài*

Thao tác trên ứng dụng điều khiển khá dễ dàng sử dụng mà không cần người dùng phải có trải nghiệm trước đó với hệ thống. Các nút bấm có kích thước khá lớn và màu sắc tương phản nổi bật các phân chức năng. Thiết kế có tính liên mạch thống nhất.

Nhóm cũng đã làm được một mô hình nhà đã được lắp đặt các thiết bị IOT mô phỏng lại ngôi nhà thật ngoài thực tế. Ứng dụng đã có thể điều khiển được các thiết bị bóng đèn và quạt, thực thi ngay lập tức sau khi click, độ trễ gần như là rất nhỏ.

3. *Hạn chế của đề tài*

Ngoài những tính năng mà nhóm đã làm được thì vẫn còn một số hạn chế về mặt thời gian và resource nên nhóm vẫn chưa thể triển khai một số tính năng trong phiên bản lần này: tìm kiếm, thêm options bộ lọc cho phần biểu đồ, thông báo cho người dùng nếu số lượng kết nối quá nhiều hoặc vượt quá định mức cho phép.

4. *Hướng phát triển*

Nhóm sẽ tiếp tục phát triển các tính năng chưa làm được và cải thiện các tính năng đã có. Tuy nhiên nhóm sẽ phát triển phiên bản cho mobile trên cả 2 nền tảng IOS và Android và thêm mới các tính năng như live camera, báo cháy, chống trộm có ứng dụng machine learning trong việc sử lý tự động.

TÀI LIỆU THAM KHẢO

- [1] <https://cachhoc.net/2013/12/03/school-ppnckh-he-thong-cac-phuong-phap-nghien-cuu-khoa-hoc/>
- [2] <https://topdev.vn/blog/restful-api-la-gi/>
- [3] <https://viblo.asia/p/tim-hieu-ve-restful-924lJM4zZPM>
- [4] <https://viblo.asia/p/tim-hieu-ve-node-js-co-ban-ojaqG0dGEKwZ>
- [5] <https://smartfactoryvn.com/technology/internet-of-things/giao-thuc-mqtt-la-gi-nhung-ung-dung-cua-mqtt-nhu-the-nao/>
- [6] <https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGlY3>
- [7] <https://viblo.asia/p/buoc-dau-tim-hieu-ve-reactjs-1VgZvwWYlAw>
- [8] <https://viblo.asia/p/tim-hieu-ve-reactjs-Az45bg3VKxY>