

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**



Đề tài

**XÂY DỰNG NỀN TẢNG QUAN TRẮC
MÔI TRƯỜNG NƯỚC DỰA TRÊN
CÔNG NGHỆ BLOCKCHAIN
PHÂN HỆ ỨNG DỤNG MOBILE**

**Sinh viên: Trần Minh Tài
Mã số: B1606931
Khóa: K42**

Cần Thơ, Tháng 1 năm 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

XÂY DỰNG NỀN TẢNG QUAN TRẮC
MÔI TRƯỜNG NƯỚC DỰA TRÊN
CÔNG NGHỆ BLOCKCHAIN
PHÂN HỆ ỨNG DỤNG MOBILE

Cán bộ hướng dẫn:
TS Trần Công Ân

Sinh viên thực hiện:
Trần Minh Tài
Mã số: B1606931
Khóa: K42

Cần Thơ, Tháng 1 năm 2021

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

Cần Thơ, ngày tháng 1 năm 2021

Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Cần Thơ, ngày tháng 1 năm 2021

Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn đến quý thầy cô trường Đại Học Cần Thơ, đặc biệt là các thầy cô Khoa Công nghệ Thông tin và Truyền thông. Cảm ơn các thầy cô đã truyền đạt những kiến thức quý báu trong suốt thời gian em học tại trường.

Em xin gửi lời cảm ơn đến thầy Trần Công Ân đã tận tình giúp đỡ, hướng dẫn, truyền đạt những kinh nghiệm và cho em những lời khuyên trong quá trình thực hiện luận văn tốt nghiệp để em có thể hoàn thành được đề tài này.

Cám ơn đến gia đình và tất cả bạn bè trong nhóm cũng như các bạn lớp Công Nghệ Thông Tin 2 khóa 42 đã hỗ trợ và giúp đỡ em trong suốt quá trình học tập cũng như trong thời gian thực hiện đề tài này.

Mặc dù đã cố gắng hoàn thành đề tài một cách hoàn chỉnh nhất, nhưng chắc chắn không thể tránh khỏi những thiếu sót, hạn chế. Rất mong nhận được sự thông cảm và sự góp ý của quý thầy cô để đề tài được hoàn thiện hơn.

Cần Thơ, ngày tháng 1 năm 2021

Sinh viên thực hiện

Trần Minh Tài

MỤC LỤC

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	i
LỜI CẢM ƠN	iii
DANH MỤC HÌNH ẢNH	vii
DANH MỤC BẢNG	ix
TÓM TẮT	x
ABSTRACT	xi
PHẦN GIỚI THIỆU	1
1. ĐẶT VẤN ĐỀ	1
2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ	1
3. MỤC TIÊU ĐỀ TÀI	1
4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	2
5. NỘI DUNG NGHIÊN CỨU	2
6. BỐ CỤC LUẬN VĂN	3
PHẦN NỘI DUNG	4
1.1. ĐẶC TẢ CHI TIẾT BÀI TOÁN	4
1.1.1. Phạm vi sản phẩm	4
1.1.2. Các chức năng của hệ thống	4
1.1.3. Đặc điểm người sử dụng	5
1.1.4. Môi trường vận hành	5
1.1.5. Các ràng buộc về thực thi và thiết kế	5
1.2. YÊU CẦU KỸ THUẬT GIAO TIẾP	5
1.2.1. Giao diện sử dụng	5
1.2.2. Phần cứng	5
1.2.3. Phần mềm	6
1.2.4. Mạng	6
1.3. CÁC CHỨC NĂNG CỦA HỆ THỐNG	6
1.3.1. Các chức năng cho quản trị viên	8
1.3.2. Các chức năng cho người dùng cuối	13
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	24
2.1. LÝ THUYẾT VỀ BLOCKCHAIN	24
2.2. TỔNG QUAN VỀ HYPERLEDGER FABRIC	27
2.2.1. Giới thiệu về Hyperledger	27
2.2.2. Giới thiệu về Hyperledger Fabric	28
2.2.3. Cấu trúc của một block và blockchain	29
2.2.4. Chức năng mô hình của mạng Hyperledger Fabric	31
2.2.5. Thuật toán đồng thuận trong Hyperledger Fabric	32
2.2.6. Mạng Blockchain Hyperledger Fabric	34
2.2.7. Nền tảng Docker	38

2.2.8.Hyperledger Fabric Client SDK	39
2.3. NODEJS LÀ GÌ.....	39
2.4. TỔNG QUAN VỀ REACTJS	39
2.5. TỔNG QUAN VỀ REACT NATIVE	39
2.6. TỔNG QUAN VỀ FIREBASE	40
CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP	41
3.1. TỔNG QUAN HỆ THỐNG.....	41
3.1.1. Về chức năng	41
3.1.2. Về thiết kế.....	41
3.2. KIẾN TRÚC HỆ THỐNG	41
3.2.1. Sơ đồ tổng quan hệ thống.....	41
3.2.2. Sơ đồ phân rã chức năng	42
3.3. THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG	42
3.3.1. Lưu đồ hoạt động của thiết bị quan trắc.....	42
3.3.2. Mô hình cơ sở dữ liệu và quản lý người dùng	44
3.3.3. Thiết kế và cài đặt Chaincode (Smartcontract)	46
3.3.4. Cài đặt và triển khai mạng Hyperledger Fabric	48
3.4. THIẾT KẾ CHỨC NĂNG	48
3.4.1. Các chức năng cho quản trị viên.....	48
3.4.2. Các chức năng cho người dùng cuối.....	52
CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ.....	60
4.1. GIỚI THIỆU.....	60
4.1.1. Mục tiêu.....	60
4.1.2. Phạm vi kiểm thử	60
4.2. CHI TIẾT KẾ HOẠCH KIỂM THỬ	60
4.2.1. Các chức năng được kiểm thử	60
4.2.2. Cách tiếp cận	60
4.2.3. Môi trường kiểm thử.....	60
4.3. CÁC TRƯỜNG HỢP KIỂM THỬ	61
4.3.1. Trường hợp kiểm thử 1: Thêm thiết bị	61
4.3.2. Trường hợp kiểm thử 2: Thêm quyền truy cập thiết bị	61
4.3.3. Trường hợp kiểm thử 3: Cập nhật quyền truy cập thiết bị.....	62
4.3.4. Trường hợp kiểm thử 4: Kích hoạt thiết bị	62
4.4. KẾT QUẢ KIỂM THỬ	63
4.5. ĐÁNH GIÁ HIỆU NĂNG MẠNG BLOCKCHAIN	64
PHẦN KẾT LUẬN	69
1. KẾT QUẢ ĐẠT ĐƯỢC	69
1.1. Về lý thuyết	69
1.2. Về chương trình	69
2. HẠN CHẾ	69
3. HƯỚNG PHÁT TRIỂN.....	69
TÀI LIỆU THAM KHẢO	70

PHỤ LỤC.....	71
1. HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG	71
1.1. Cài đặt Hyperledger Fabric	71
1.2. Cài đặt ứng dụng.....	71

DANH MỤC HÌNH ẢNH

Hình 1.1 Chức năng người quản trị hệ thống.....	6
Hình 1.2 Chức năng của người dùng.....	7
Hình 2.1 Kiến trúc khối Blockchain.....	24
Hình 2.2 Hệ sinh thái Hyperledger.....	27
Hình 2.3 Sơ đồ bao gồm Blockchain và world state bắt nguồn từ Blockchain	29
Hình 2.4 Cấu trúc khối Hyperledger	29
Hình 2.5 Khối giao dịch.....	30
Hình 2.6 Mô tả hoạt động của cơ chế đồng thuận mạng fabric	33
Hình 2.7 Mô tả quá trình 3 của cơ chế đồng thuận	34
Hình 2.8 Kiến trúc một mạng Hyperledger Fabric cơ bản	34
Hình 2.9 Khởi tạo cơ sở mạng	35
Hình 2.10 Thêm quản trị viên cho mạng	36
Hình 2.11 Định nghĩa 1 nhóm.....	36
Hình 2.12 Tạo 1 kênh cho một nhóm	36
Hình 2.13 Peer và Ledger	37
Hình 2.14 Ứng dụng và hợp đồng thông minh chaincode.....	37
Hình 2.15 Hoàn thành mạng	38
Hình 3.1 Sơ đồ tổng quản hệ thống.....	41
Hình 3.2 Mô tả sự phân phân rã của hệ thống	42
Hình 3.3 Lưu đồ hoạt động thiết bị	43
Hình 3.4 Luồng hoạt động thiết bị	44
Hình 3.5 Thông tin người dùng trong Firebase.....	44
Hình 3.6 Cấu trúc assets	47
Hình 3.7 Chức năng đăng nhập phía admin.....	49
Hình 3.8 Lưu đồ đăng nhập với quản trị viên.....	49
Hình 3.9 Quản lý thiết bị phía admin	50
Hình 3.10 Kích hoạt thiết bị.....	50
Hình 3.11 Lưu đồ kích hoạt thiết bị	51
Hình 3.12 Tìm kiếm thiết bị (Admin)	51
Hình 3.13 Cấp lại token thiết bị	52
Hình 3.14 Lưu đồ chức năng cấp lại token.....	52

Hình 3.15 Đăng nhập phía người dùng di động.....	53
Hình 3.16 Lưu đồ đăng nhập với người dùng.....	53
Hình 3.17 Quản lý người dùng phía client	54
Hình 3.18 Truy xuất dữ liệu theo thời gian thực.....	55
Hình 3.19 Lưu đồ truy xuất dữ liệu realtime	55
Hình 3.20 Truy xuất theo khoản thời gian.....	56
Hình 3.21 Quản lý người dùng được chia sẻ	57
Hình 3.22 Thêm quyền truy xuất thiết bị.....	58
Hình 3.23 Lưu đồ chia sẻ quyền truy cập thiết bị	58
Hình 3.24 Cập nhật quyền truy xuất thiết bị.....	59
Hình 4.1 Biểu đồ giao dịch thành công/thất bại ở các mức	66
Hình 4.2 Biểu đồ độ trễ giao dịch	66

DANH MỤC BẢNG

Bảng 3.1 Cấu trúc collection devices	45
Bảng 3.2 Cấu trúc collection bcAccounts.....	46
Bảng 3.3 Cấu trúc collection fieldRef	46
Bảng 3.4 Cấu trúc collection adminUser	46
Bảng 4.1 Kiểm thử thêm thiết bị	61
Bảng 4.2 Kiểm thử thêm quyền truy cập	62
Bảng 4.3 Kiểm thử thêm quyền truy cập thiết bị	62
Bảng 4.4 Kiểm thử thêm quyền truy cập thiết bị	63
Bảng 4.5 Kết quả kiểm thử	63
Bảng 4.6 Đánh giá mạng blockchain.....	65
Bảng 4.6.7 Thử nghiệm với bộ điều khiển Fixed Load.....	67
Bảng 4.8 Thử nghiệm ở mức 354 giao dịch trên giây	67
Bảng 4.9 Tài nguyên sử dụng	68

TÓM TẮT

Từ xưa đến nay, việc theo dõi và đo lường sự thay đổi nồng độ các chất có môi trường để đưa ra đánh giá ảnh hưởng đến sự sinh trưởng, phát triển của sinh vật sống trong môi trường luôn là vấn đề được nhiều tổ chức, doanh nghiệp cũng như nhiều cá nhân quan tâm. Trước đây, quá trình thu thập, lưu trữ và đánh giá thường theo những cách thủ công nên thường xảy ra chậm trễ trong việc đưa ra những cảnh báo khi có kết quả. Hiện nay, với sự phát triển mạnh mẽ trong lĩnh vực khoa học – công nghệ nói chung, cũng như trong công nghệ thông tin nói riêng đã tác động mạnh đến toàn bộ các mặt trong đời sống xã hội. Việc quan trắc môi trường cũng trở nên dễ dàng và nhanh chóng với những thiết bị có thể tự động đo lường và gửi dữ liệu thông qua nhiều giao thức truyền thông cùng với những ứng dụng hỗ trợ cho việc lưu trữ, quản lý các thông tin quan trắc.

Nền tảng quan trắc môi trường nước hỗ trợ lưu trữ dữ liệu một cách nhanh chóng và lâu dài và giúp quản lý một cách trực quan nhất. Nền tảng cung cấp chức năng phục vụ cho 2 kiểu người dùng: quản trị hệ thống (Admin), người dùng cuối bao gồm chủ thiết bị và người dùng được chia sẻ quyền xem thiết bị. Người dùng cuối có thể gửi yêu cầu tạo định danh thiết bị trên hệ thống, có thể quản lý các thiết bị đã được tạo, quan sát và thống kê dữ liệu lưu trữ trên từng thiết bị, chia sẻ quyền xem từng sensor trong thiết bị cho những người dùng khác. Quản trị hệ thống có nhiệm vụ xác nhận các yêu cầu tạo máy của người dùng và gửi thông tin về thiết bị được tạo trên hệ thống cho người dùng. Nền tảng được xây dựng với: Mạng blockchain Hyperledger Fabric để lưu trữ dữ liệu của các thiết bị, NodeJS và Firebase để xây dựng phía server, ReactJS và React Native để xây dựng giao diện phía client.

Đề tài “**Xây dựng nền tảng quan trắc môi trường dựa trên công nghệ blockchain**” hoàn thiện và đáp ứng được các yêu cầu về thiết kế và cài đặt. Hệ thống đáp ứng đầy đủ các chức năng và hoạt động ổn định.

ABSTRACT

From past to present day, the monitoring and measurement of the changes in the concentration of substances in the environment to make an assessment on the growth and development of organisms living in that environment is always a matter of many organizations, businesses as well as many individuals interested. In the past, the process of gathering, storing and evaluating was often manual, so there was often a delay in issuing warnings when results were available. Currently, with the strong development of the field of science – technology in general, as well as the field of information technology in particular, has a strong impact on all aspects of social life. Environmental monitoring is also easy and fast with devices that can automatically measure and send data through multiple communication protocols along with applications that support the storage and management of monitoring information.

Environmental monitoring platform support fast and long-term data storage and help manage the most intuitive way. The platform provides functionality for two types of users: the system administrator (Admin), the end user includes the device owner, and the user is shared access to view the device. End users can send requests to create device identifiers on the system, manage devices that have been created, view and statistical data stored on each device, and share the right to view each sensor in the device for other users. The system administrator is responsible for confirming user's machine creation requests and sending information about the device created on the system to the user. The platform is built with: Hyperledger Fabric blockchain network for storing device data, NodeJS and Firebase for server-side building, ReactJS and React Native to build client-side interfaces.

The topic “**Building an environmental monitoring platform**” based on blockchain technology is complete and meets the requirements for design and installation. The system is fully functional and stable.

CAM KẾT KẾT QUẢ

Em xin cam kết luận văn này được hoàn thành dựa trên kết quả nghiên cứu của tôi và kết quả nghiên cứu chưa được dùng cho bất cứ luận văn cùng cấp nào khác.

Em xin cam đoan mọi sự giúp đỡ cho việc thực hiện luận văn này đã được cảm ơn và các tài liệu tham khảo trong luận văn đã được ghi rõ nguồn gốc.

Cần Thơ, ngày tháng 1 năm 2020

Sinh viên thực hiện

Trần Minh Tài

PHẦN GIỚI THIỆU

1. ĐẶT VẤN ĐỀ

Trong những năm gần đây ngành nuôi trồng thủy sản ở Việt Nam là một trong những ngành có mức độ phát triển nhanh nhất trên thế giới góp phần thúc đẩy đất nước ngày càng phát triển ngày càng phát triển mặc dù vậy trong quá trình phát triển còn tồn tại nhiều nguy cơ và rủi ro cao tác động trực tiếp đến hiệu suất sản xuất. Bên cạnh đó ô nhiễm môi trường nước, tác động của biến đổi khí hậu theo hướng ngày càng xấu đã và đang ảnh hưởng lớn trong quá trình nuôi trồng và sản xuất. Để có thể khai thác hết tiềm năng của ngành nuôi trồng thủy sản, đồng thời giải quyết được phần nào nguy cơ, rủi ro ô nhiễm môi trường, chống lại tác động của biến đổi khí hậu và dịch bệnh, nâng cao năng suất nuôi trồng thì một hệ thống ghi lại và giám sát chất lượng nước là vô cùng cần thiết. Việc lưu trữ những dữ liệu này là một vấn đề được lưu tâm đến. Dữ liệu được lưu trữ thế nào để an toàn, minh bạch và dễ dàng cũng như truy cập để quản lý một cách trực quan nhất. Vấn đề đặt ra ở đây là cần có một ứng dụng hay nền tảng hỗ trợ được việc đó.

Cùng với đó, cuộc cách mạng 4.0 dần xuất hiện những xu hướng công nghệ mới, nổi trội như trí tuệ nhân tạo AI, nền tảng đám mây... Blockchain cũng là một trong số đó. Blockchain tạo sự chắc chắn và minh bạch về dữ liệu số. Khi dữ liệu được ghi vào trong một cơ sở dữ liệu blockchain nó gần như không thể bị thay đổi hay bị xóa dữ liệu đấy đi. Chính vì vậy, nó tạo được sự tin tưởng của các cá nhân, tổ chức tham gia vào mạng.

Với việc những dữ liệu này không thể bị thay đổi khi lưu vào mạng khiến việc lưu trữ những dữ liệu cần sự chính xác như việc lưu trữ những dữ liệu quan trắc được rõ ràng và được sự tin tưởng từ nhiều người dùng hơn.

Vì vậy, em quyết định ứng dụng công nghệ Blockchain vào đề tài “Xây dựng nền tảng quan trắc môi trường nước”. Cụ thể hơn là sử dụng mạng blockchain Hyperledger Fabric.

2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

Hiện nay, có rất nhiều nền tảng hỗ trợ lưu trữ dữ liệu của các thiết bị IoT và cho phép người dùng quản lý nó như:

- ThingSpeak <https://thingspeak.com>: là một nền tảng mã nguồn mở hỗ trợ thu thập và lưu trữ dữ liệu các cảm biến của bạn lên đám mây.
- Kaa <https://kaaproject.org>: là nguồn mở, đa nền tảng, nền tảng phần mềm trung gian để phát triển hoàn chỉnh các thiết bị IoT từ đầu đến cuối

Dù vậy, em cũng muốn thử áp dụng công nghệ Blockchain để thử sức với đề tài này.

3. MỤC TIÊU ĐỀ TÀI

Mục tiêu của đề tài “Xây dựng nền tảng quan trắc môi trường nước” thỏa mãn những yêu cầu sau:

- Nghiên cứu xây dựng được hệ thống quản lý, lưu trữ dựa trên nền tảng công nghệ blockchain Hyperledger Fabric.

- Nghiên cứu công nghệ Hyperledger Fabric và các ứng dụng mẫu của nó, hiểu được cách thức hoạt động và áp dụng mạng blockchain vào hệ thống.
- Nền tảng đơn giản, trực quan, dễ sử dụng, thân thiện với người dùng.

4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đề tài sẽ nghiên cứu vào các đối tượng sau đây:

- Nghiên cứu xây dựng một mạng Blockchain lưu trữ dữ liệu từ các thiết bị quan trắc gửi lên.
- Nghiên cứu và sử dụng ngôn ngữ JavaScript để xây dựng hợp đồng thông minh (Smart Contract) để xử lý dữ liệu lưu trữ dữ liệu trên mạng blockchain.
- Nghiên cứu và xây dựng một server NodeJS sử dụng Express Framework và các thư viện hỗ trợ.
- Nghiên cứu và ứng dụng Firebase vào hệ thống.
- Nghiên cứu về cách thức hoạt động của một tiến trình quan trắc.
- Nghiên cứu và ứng dụng Framework ReactJS, React Native và các thư viện của nó vào việc xây dựng phía client.

5. NỘI DUNG NGHIÊN CỨU

- Tìm hiểu về quy trình quan trắc.
- Hiểu được cách xây dựng mạng blockchain, cài đặt được mạng Hyperledger Fabric.
- Biết cách sử dụng JavaScript để xây dựng, cài đặt chaincode trong Hyperledger Fabric.
- Tìm hiểu cách được cách xây dựng giao diện website cũng như cách thức xây dựng ứng dụng mobile.
- Tìm hiểu được cách thức vận hành của framework ReactJS, React Native và NodeJS.
- Sử dụng Microsoft Office để viết báo cáo và soạn trình chiếu...
- Quá trình nghiên cứu:
 - Tìm hiểu và thu thập kiến thức: thu thập các tài liệu, các kiến thức cần thiết liên quan đến hệ thống, tìm hiểu các hệ thống sẵn có.
 - Tìm hiểu và lựa chọn công nghệ sử dụng.
 - Đặc tả và thiết kế hệ thống.
 - Thiết kế các chức năng áp dụng cho hệ thống.
 - Lập trình.
 - Cài đặt và kiểm thử.
 - Tổng hợp và viết báo cáo.

6. BỐ CỤC LUẬN VĂN

Nội dung luận văn bao gồm các phần sau:

- Phần đầu bao gồm: Lời cảm ơn, mục lục, các kí hiệu và từ viết tắt, tóm tắt và cam kết kết quả.
- Phần giới thiệu: Nêu lên vấn đề cần giải quyết và phạm vi của vấn đề. Sau đó lên kế hoạch và phương pháp thực hiện. Phần giới thiệu bao gồm những nội dung sau đây: Đặt vấn đề, lịch sử giải quyết vấn đề, mục tiêu đề tài, đối tượng, phạm vi nghiên cứu và nội dung nghiên cứu.
- Phần nội dung: Gồm bốn chương, trình bày nội dung chính của luận văn một cách chi tiết:
 - + **Chương 1:** Đặc tả và yêu cầu nêu lên chi tiết về đề tài và các chức năng, yêu cầu đặt ra của đề tài.
 - + **Chương 2:** Cơ sở lý thuyết về các công nghệ sử dụng trong đề tài
 - + **Chương 3:** Thiết kế và cài đặt giải pháp là mô tả tổng quan hệ thống, thiết kế kiến trúc, cơ sở dữ liệu cũng như giao diện của hệ thống.
 - + **Chương 4:** Kiểm thử và đánh giá với những mục tiêu cùng những trường hợp kiểm thử, sau đó đưa ra đánh giá đối với các chức năng của hệ thống.
- Phần kết luận: Trình bày những kết quả đạt được khi thực hiện đề tài và những hạn chế mà đề tài chưa thực hiện được để đưa ra hướng giải quyết sau này.

PHẦN NỘI DUNG**CHƯƠNG 1: ĐẶC TẢ VÀ YÊU CẦU****1.1. ĐẶC TẢ CHI TIẾT BÀI TOÁN****1.1.1. Phạm vi sản phẩm**

Nền tảng quan trắc môi trường hướng tới đối tượng người dùng là các tổ chức, các doanh nghiệp, các trang trại... Nền tảng có khả năng lưu trữ, quản lý nhiều loại dữ liệu từ nhiều thiết bị khác nhau như những thiết bị bao gồm một hoặc nhiều cảm biến. Hệ thống cung cấp quyền lưu trữ dữ liệu trên mạng blockchain Hyperledger Fabric cho người dùng giúp người dùng đảm bảo được sự minh bạch và chính xác của dữ liệu.

Người dùng có thể truy xuất thông tin trên các thiết bị của họ theo thời gian thực cũng như có thể chia sẻ quyền truy xuất dữ liệu trên thiết bị của mình cho những người dùng khác trong hệ thống.

Đề tài sử dụng công nghệ blockchain Hyperledger Fabric, hệ thống lưu trữ phân tán trên sổ cái ledger, xây dựng chaincode với ngôn ngữ JavaScript.

1.1.2. Các chức năng của hệ thống

Quản trị hệ thống (Admin):

- Đăng nhập, đăng xuất.
- Quản lý thiết bị:
 - + Kích hoạt thiết bị.
 - + Gửi thông tin thiết bị cho người dùng.
 - + Cấp lại token cho người dùng.
 - + Người dùng sở hữu thiết bị:
- Đăng nhập, đăng xuất.
- Gửi yêu cầu tạo định danh thiết bị mới
- Quản lý thiết bị:
 - + Truy xuất dữ liệu thiết bị theo thời gian thực.
 - + Truy xuất dữ liệu theo một khoảng thời gian và đưa ra thông tin được phân tích.
 - + Cấp quyền truy xuất thiết bị cho người dùng khác (bao gồm quyền truy xuất theo từng cảm biến).
 - + Thu hồi quyền truy xuất thiết bị từ những người dùng đã được chia sẻ.

Người dùng được chia sẻ quyền xem thiết bị:

- Đăng nhập, đăng xuất.
- Chức năng bên trong thiết bị được chia sẻ:

- + Truy xuất dữ liệu theo thời gian thực với những trường cảm biến đã được chia sẻ.
- + Truy xuất dữ liệu theo thời gian.

Thiết bị và cảm biến:

- Gửi dữ liệu lên hệ thống.

1.1.3. Đặc điểm người sử dụng

Người quản trị: quản lý thông tin về các thiết bị, cấp quyền hoạt động cho phép các thiết bị được lưu trữ dữ liệu trong mạng, sử dụng tài khoản được cấp quyền riêng biệt. Người dùng cuối tham gia vào hệ thống với phương thức đăng nhập sẵn có.

1.1.4. Môi trường vận hành

Server:

- Mạng blockchain Hyperledger Fabric sử dụng trên hệ điều hành Ubuntu.
- Xây dựng và sử dụng hợp đồng thông minh để tương tác với dữ liệu được lưu trên sổ cái.
- Xây dựng ứng dụng bằng NodeJS với Framework ExpressJS kết hợp với việc lưu trữ thông tin trên Firebase. Những chức năng tác động đến mạng blockchain sẽ do phía Server quản lý thông qua Hyperledger Fabric SDK for Node.js, dữ liệu truy vấn sẽ được trả về cho Client theo những yêu cầu mà phía Client đã gửi cho Server.

Client:

- Về phía Web: sử dụng tốt trên các trình duyệt như Google Chrome, Firefox... được thiết kế bằng framework ReactJS.
- Về phía ứng dụng mobile: Xây dựng với framework React Native sử dụng với các thiết bị chạy Android.

1.1.5. Các ràng buộc về thực thi và thiết kế

- Hệ thống phải được thực thi trên hệ điều hành Ubuntu, và mạng Hyperledger Fabric cùng chaincode phải được cài đặt.
- Phải đảm bảo kết nối Internet để sử dụng được các thư viện cũng như kết nối đến Firebase.
- Hệ thống phải đáp ứng khả năng an toàn, bảo mật.
- Hệ thống phải dễ sử dụng, trực quan.

1.2. YÊU CẦU KỸ THUẬT GIAO TIẾP

1.2.1. Giao diện sử dụng

Bố cục, font chữ đơn giản, rõ ràng, ngôn từ dễ hiểu với người sử dụng.

1.2.2. Phần cứng

Các thiết bị phần cứng phải đáp ứng được những yêu cầu giúp hệ thống hoạt động trơn tru, ổn định.

1.2.3. Phần mềm

Server:

- Hệ điều hành: sử dụng hệ điều hành Ubuntu (Ubuntu 20.04).

Client:

- Sử dụng các trình duyệt web thông dụng như Chrome, Firefox.
- Sử dụng trên các thiết bị chạy hệ điều hành Android.

1.2.4. Mạng

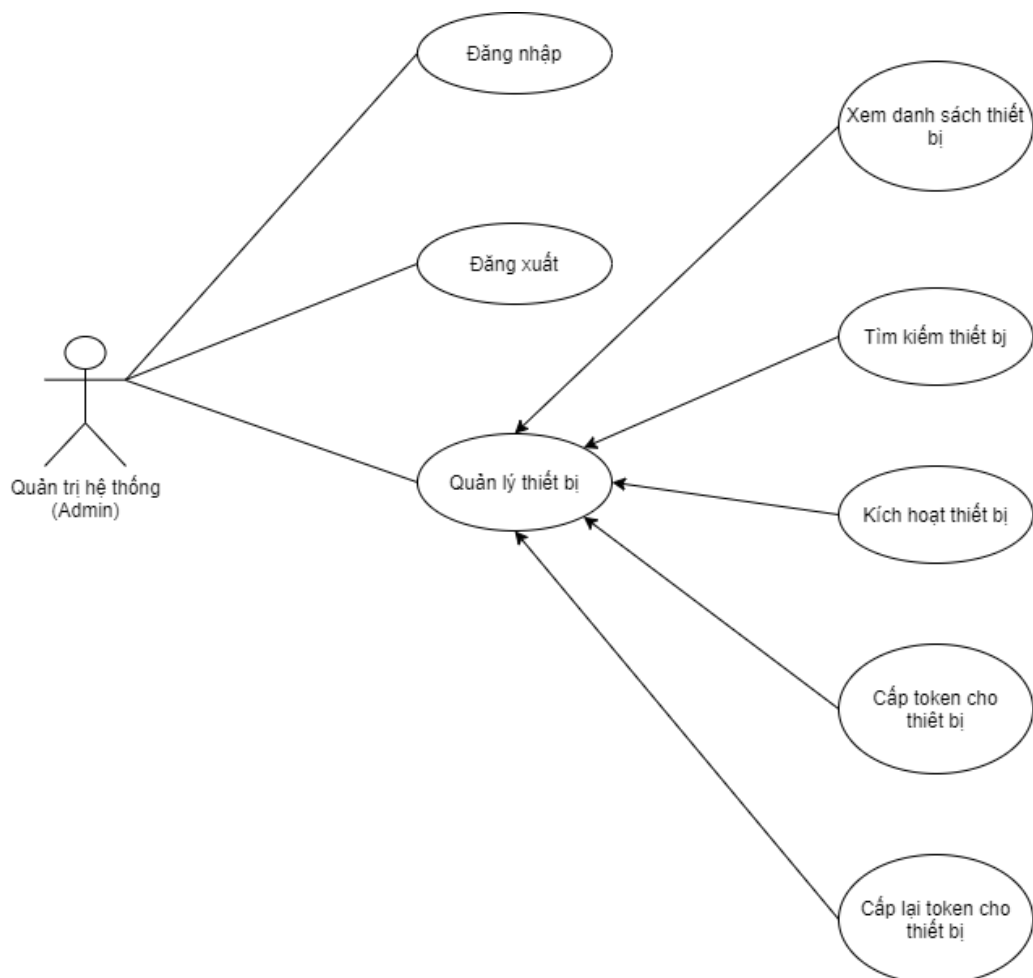
Sử dụng giao thức HTTP.

Sử dụng các phương thức GET, POST để gửi và nhận dữ liệu từ server.

1.3. CÁC CHỨC NĂNG CỦA HỆ THỐNG

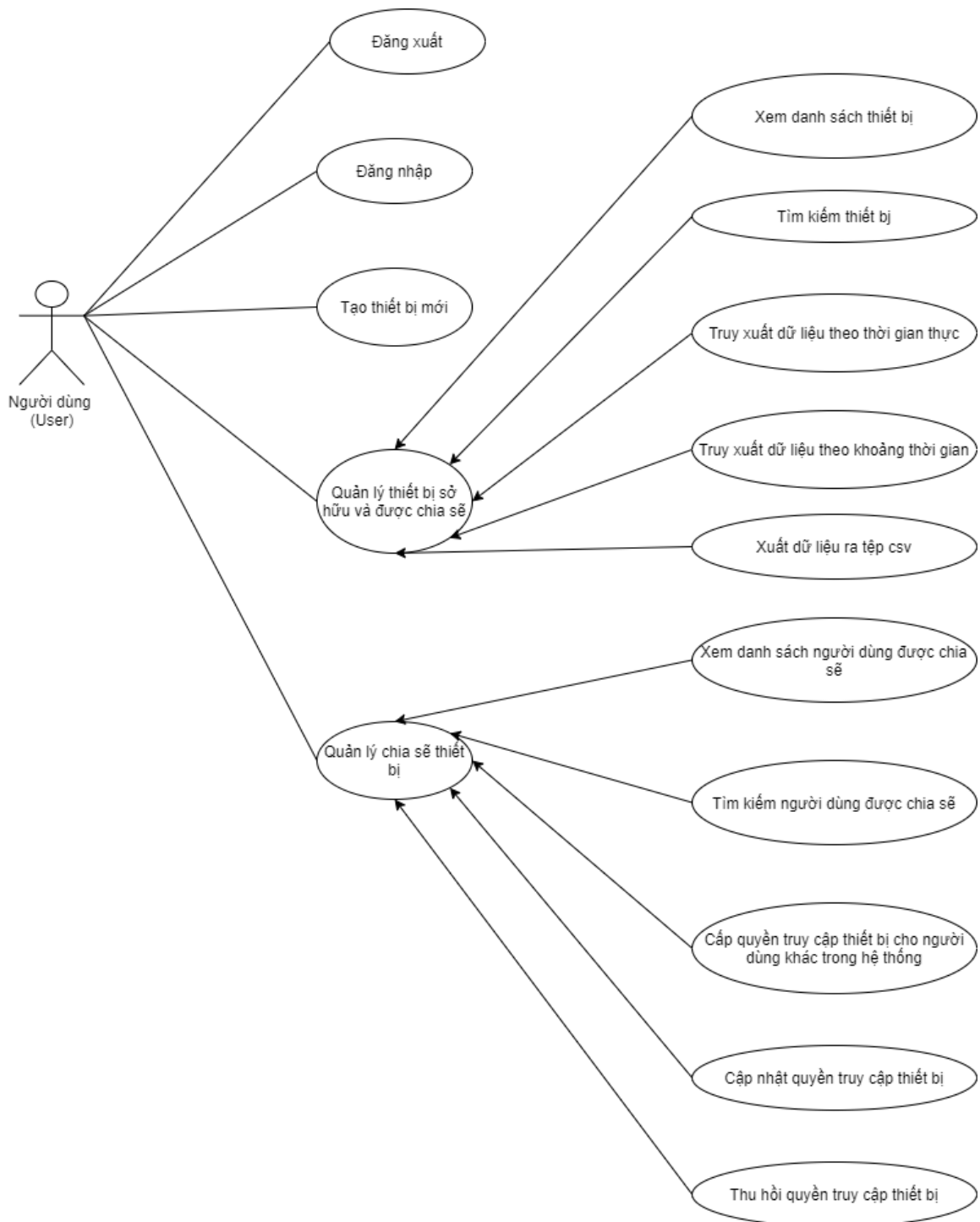
Hệ thống bao gồm 2 đối tượng người dùng bao gồm người dùng quản trị hệ thống và người sử dụng hệ thống. Chức năng mô tả trong các hình sau:

Người dùng quản trị hệ thống sẽ có những chức năng quan trọng như: năng kích hoạt thiết bị, cấp token, cập lại token...



Hình 1.1 Chức năng người quản trị hệ thống

Người dùng cuối sẽ bao gồm những chức năng như: tạo thiết bị mới, truy xuất dữ liệu theo thời gian thực, truy xuất dữ liệu theo khoảng thời gian, cấp quyền truy cập thiết bị cho người dùng khác trong hệ thống, thu hồi và cập nhật quyền truy cập thiết bị của người dùng được chia sẻ...



Hình 1.2 Chức năng của người dùng

1.3.1. Các chức năng cho quản trị viên**1.3.1.1. Đăng nhập****Mô tả và mức độ ưu tiên**

- Mô tả: Chức năng này cho phép người dùng quản trị hệ thống truy cập vào phần quản lý hệ thống để sử dụng các chức năng bắt buộc.
- Độ ưu tiên: Cao

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng quản trị hệ thống.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1. Hệ thống hiển thị giao diện đăng nhập. 2. Người dùng nhập thông tin cần thiết bao gồm: email và mật khẩu. 3. Người dùng nhấn vào nút đăng nhập.
Dòng thay thế	
Dòng sự kiện lỗi	Dòng sự kiện lỗi xảy ra ở bước 4 của dòng sự kiện chính khi thông tin người dùng được hệ thống kiểm tra không chính xác: 4. Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã truy cập vào hệ thống.

1.3.1.2. Quản lý thiết bị

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép quản trị hệ thống xem danh sách các thiết bị bao gồm cả thiết bị đã được kích hoạt và các thiết bị chưa được kích hoạt.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng quản trị hệ thống.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1. Người dùng hoàn thành việc đăng nhập. 2. Hệ thống chuyển sang trang quản lý thiết bị. 3. Người dùng lựa chọn trong danh sách để sử dụng các chức năng tiếp theo
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã hoàn thành việc đăng nhập vào hệ thống với tài khoản quản trị viên.

1.3.1.3. Kích hoạt thiết bị

Mô tả và mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng quản trị viên gửi yêu cầu lên phía Server để kích hoạt định danh cho thiết bị.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng quản trị hệ thống.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Tại giao diện danh sách thiết bị, người dùng nhấn vào nút có kí hiệu là [x]. 2. Giao diện kích hoạt thiết bị sẽ xuất hiện. 3. Người dùng nhập thông tin yêu cầu (nhập lại chính xác ID của thiết bị để xác nhận). 4. Người dùng nhấn vào nút [Đồng ý] để hoàn thành việc kích hoạt thiết bị.
Dòng thay thế	
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi bắt đầu từ bước 4 của dòng sự kiện chính trường hợp dữ liệu nhập vào không hợp lệ:</p> <ol style="list-style-type: none"> 5. Hệ thống thông báo lỗi, người dùng nhập lại thông tin ở bước 3 sau đó tiếp tục thực hiện.

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã hoàn thành việc đăng nhập vào hệ thống với tài khoản quản trị viên.

1.3.1.4. Xem thông tin thiết bị

Mô tả và mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng quản trị viên xem thông tin về các thiết bị.
- Độ ưu tiên: Thấp.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng quản trị hệ thống.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<p>1. Tại giao diện danh sách thiết bị, người dùng nhấn vào nút có kí hiệu là [v].</p> <p>2. Giao diện thông tin thiết bị sẽ xuất hiện.</p>
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã hoàn thành việc đăng nhập vào hệ thống với tài khoản quản trị viên.
- Yêu cầu 2: Thiết bị đã được kích hoạt.

1.3.1.5. Cấp lại token cho thiết bị

Mô tả và mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng quản trị viên cấp lại token cho chủ sở hữu thiết bị.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng quản trị hệ thống.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Tại giao diện danh sách thiết bị, người dùng nhấn vào nút có kí hiệu là [Cấp lại token]. 2. Giao diện cấp lại token thiết bị sẽ xuất hiện. 3. Người dùng nhập thông tin theo yêu cầu. 4. Người dùng nhấn nút [Đồng ý] để hoàn thành.
Dòng thay thế	
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi bắt đầu từ bước 4 của dòng sự kiện chính trường hợp dữ liệu nhập vào không hợp lệ:</p> <ol style="list-style-type: none"> 5. Hệ thống thông báo lỗi, người dùng nhập lại thông tin ở bước 3 sau đó tiếp tục thực hiện.

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã hoàn thành việc đăng nhập vào hệ thống với tài khoản quản trị viên.
- Yêu cầu 2: Thiết bị đã được kích hoạt.

1.3.2. Các chức năng cho người dùng cuối

1.3.2.1. Đăng nhập

Mô tả và mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng đăng nhập vào hệ thống để sử dụng các chức năng bắt buộc.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Chủ thiết bị và người dùng được chia sẻ quyền truy xuất dữ liệu.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Tại giao diện chính, người sử dụng nhấn vào nút Đăng nhập với Google. 2. Một bảng chọn sẽ xuất hiện. 3. Người dùng chọn một trong những tài khoản Google đã được đăng nhập từ trước để tiếp tục. 4. Hoàn thành việc đăng nhập.
Dòng thay thế	<p>Dòng sự kiện thay thế bắt đầu từ bước 2 của dòng sự kiện chính trường hợp chưa đăng nhập tài khoản Google từ trước:</p> <ol style="list-style-type: none"> 3. Yêu cầu đăng nhập Google xuất hiện. 4. Người dùng nhập thông tin tài khoản Google để đăng nhập. 5. Hoàn thành đăng nhập.
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng truy cập vào hệ thống.

1.3.2.2. Tạo định danh thiết bị

Mô tả và mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng yêu cầu khởi tạo vùng lưu trữ dữ liệu trên hệ thống.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Tại giao diện chính, người sử dụng nhấn vào nút thêm thiết bị. 2. Giao diện thêm thiết bị sẽ xuất hiện. 3. Người dùng nhập thông tin của thiết bị vào form. 4. Nhấn vào nút thêm để hoàn thành việc yêu cầu tạo máy.
Dòng thay thế	<p>Dòng sự kiện thay thế bắt đầu từ bước 3 của dòng sự kiện chính:</p> <ol style="list-style-type: none"> 4. Người dùng nhấn vào nút hủy để việc thêm thiết bị sẽ được hủy bỏ
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi xảy ra bắt đầu từ bước 4 của dòng sự kiện chính khi các thông tin không được nhập đầy đủ:</p> <ol style="list-style-type: none"> 5. Yêu cầu người dùng nhập lại thông tin và thực hiện thêm lại.

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống

1.3.2.3. Quản lý thiết bị

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng xem danh sách các thiết bị gồm danh sách thiết bị sở hữu và danh sách thiết bị được chia sẻ
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1. Người dùng hoàn thành việc đăng nhập. 2. Hệ thống chuyển sang trang quản lý thiết bị. 3. Người dùng lựa chọn trong danh sách để sử dụng các chức năng tiếp theo
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.

1.3.2.4. Truy xuất dữ liệu theo thời gian thực**Mô tả mức độ ưu tiên**

- Mô tả: Chức năng cho phép người dùng truy xuất dữ liệu theo dạng biểu đồ và nhận dữ liệu liên tục theo thời gian thực.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1. Người dùng chọn thiết bị để truy xuất từ danh sách thiết bị ở chức năng quản lý thiết bị. 2. Hệ thống chuyển sang giao diện quản lý thiết bị. 3. Người dùng có thể tương tác với các thông số trên các biểu đồ hiển thị của các sensor
Dòng thay thế	Dòng sự kiện thay thế xảy ra ở bước thứ 2 khi thiết bị chưa được kích hoạt: 2. Hệ thống thông báo thiết bị chưa được kích hoạt và ở lại giao diện chính
Dòng sự kiện lỗi	Dòng sự kiện lỗi xảy ra ở bước thứ 2 khi có vấn đề về kết nối: 2. Hệ thống thông báo lỗi và trở về giao diện chính

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng đã có thiết bị hoặc đã được chia sẻ thiết bị.

1.3.2.5. Truy xuất dữ liệu theo khoảng thời gian

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng truy xuất dữ liệu trong một khoảng thời gian nhất định.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Từ giao diện truy xuất dữ liệu thiết bị người dùng nhấn vào nút [Thống kê] 2. Hệ thống chuyển sang giao diện cho người dùng chọn ngày. 3. Người dùng nhấn vào nút [Chấp nhận] hệ thống chuyển sang giao diện với các biểu đồ và đưa ra những giá trị cần thiết.
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng đã truy cập vào một thiết bị.

1.3.2.6. Sao lưu dữ liệu thành tệp CSV**Mô tả mức độ ưu tiên**

- Mô tả: Chức năng cho phép người dùng tải về tập tin dữ liệu để sử dụng cho những mục đích khác nhau.
- Độ ưu tiên: Trung bình.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1. Người dùng nhấn vào nút [Sao lưu]. 2. Yêu cầu người dùng chọn thư mục trên máy tính hoặc điện thoại thông minh để lưu trữ. 3. Người dùng nhấn [Save] để hoàn thành.
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng đã truy xuất dữ liệu trên thiết bị theo thời gian.

1.3.2.7. Quản lý quyền truy xuất vào thiết bị của người dùng

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng quản lý được danh sách người dùng có thể truy xuất vào dữ liệu các cảm biến của thiết bị.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<p>1. Ở giao diện truy xuất dữ liệu thiết bị. Người dùng nhấn vào nút [Quản lý]</p> <p>2. Hệ thống chuyển sang giao diện quản lý người dùng có quyền truy xuất thiết bị. Tại đây người dùng có thể thực hiện tiếp các chức năng như thêm quyền cho một người dùng mới truy cập vào hệ thống hay cập nhật lại quyền của một người dùng đã có.</p>
Dòng thay thế	
Dòng sự kiện lỗi	

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng là chủ sở hữu thiết bị.

1.3.2.8. Cấp quyền truy xuất cho người dùng khác trong hệ thống

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng sở hữu thiết bị thêm quyền truy xuất vào thiết bị cho một người dùng khác trong hệ thống.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Ở giao diện quản lý quyền truy xuất thiết, người dùng chọn vào nút [Thêm]. 2. Hệ thống xuất hiện giao diện để nhập thông tin và chọn các cảm biến sẽ chia sẻ cho người dùng được nhập trong biểu mẫu. 3. Nhấn vào nút thêm để hoàn thành việc thêm. 4. Khi hoàn thành hệ thống sẽ thông báo việc thêm đã hoàn thành.
Dòng thay thế	
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi xảy ra ở bước thứ 3 khi người dùng được thêm vào không tồn tại, hoặc đã được cấp quyền từ trước:</p> <ol style="list-style-type: none"> 5. Hệ thống thông báo việc thêm thất bại và yêu cầu nhập lại biểu mẫu.

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng là chủ sở hữu thiết bị.

1.3.2.9. Cập nhật quyền truy xuất cho người dùng khác trong hệ thống

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng cập nhật lại quyền truy xuất vào từng cảm biến cho người dùng đã được họ chia sẻ quyền truy xuất dữ liệu từ trước.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<p>1.Ở giao diện quản lý quyền truy xuất thiết bị, người dùng chọn vào một người dùng đã được chia sẻ quyền để cập nhật lại quyền của họ.</p> <p>2. Hệ thống hiển thị giao diện cập nhật lại quyền của người dùng được chia sẻ. Chủ sở hữu thiết bị sẽ cập nhật lại các quyền xem các cảm biến trong thiết bị cho người dùng được chia sẻ.</p> <p>3. Chủ sở hữu nhấn vào nút [Cập nhật] để hoàn thành việc cập nhật</p> <p>4. Hệ thống sẽ thông báo việc cập nhật hoàn tất.</p>
Dòng thay thế	
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi xảy ra ở bước thứ 3 khi có vấn đề về kết nối:</p> <p>4. Hệ thống thông báo có lỗi xảy ra và yêu cầu thực hiện lại.</p>

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng sở hữu thiết bị.
- Yêu cầu 3: Người dùng đã chia sẻ quyền truy xuất thiết bị cho ít nhất một người dùng khác.

1.3.2.10. Thu hồi quyền truy xuất cho người dùng khác trong hệ thống

Mô tả mức độ ưu tiên

- Mô tả: Chức năng cho phép người dùng thu hồi lại quyền truy xuất thiết bị từng cấp cho một người dùng khác trong hệ thống.
- Độ ưu tiên: Cao.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	<p>1.Ở giao diện quản lý quyền truy xuất thiết bị, người dùng chọn vào một người dùng đã được chia sẻ quyền để thu hồi quyền truy xuất thiết bị của họ.</p> <p>2. Hệ thống sẽ hiển thị lại thông báo để chắc chắn rằng bạn có muốn thu hồi.</p> <p>3. Người dùng lựa chọn [Đồng ý].</p> <p>4. Hệ thống sẽ hoàn thành việc thu hồi quyền truy xuất.</p>
Dòng thay thế	<p>Dòng sự kiện thay thế xảy ra ở bước thứ 3 khi người dùng lựa chọn [Hủy].</p> <p>4. Hệ thống sẽ bỏ qua việc thu hồi quyền truy xuất</p>
Dòng sự kiện lỗi	<p>Dòng sự kiện lỗi xảy ra ở bước thứ 4 khi có vấn đề về kết nối:</p> <p>4. Hệ thống thông báo có lỗi xảy ra và yêu cầu thực hiện lại.</p>

Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.
- Yêu cầu 2: Người dùng sở hữu thiết bị.
- Yêu cầu 3: Người dùng đã chia sẻ quyền truy xuất thiết bị cho ít nhất một người dùng khác.

1.3.2.11. Tìm kiếm thiết bị**Mô tả mức độ ưu tiên**

- Mô tả: Chức năng cho phép người tìm kiếm thiết bị trong danh sách thiết bị của phần quản lý thiết bị
- Độ ưu tiên: Trung bình.

Tác nhân/ Chuỗi đáp ứng

- Tác nhân: Người dùng cuối.

Bảng chuỗi hoạt động của người sử dụng và đáp ứng của hệ thống:

Dòng sự kiện chính	1.Ở giao diện quản lý thiết bị, người dùng nhập thông tin thiết bị cần tìm kiếm vào trường tìm kiếm. 2. Nhấn vào nút [Tìm kiếm] để thực hiện việc lọc lại danh sách thiết bị
Dòng thay thế	
Dòng sự kiện lỗi	

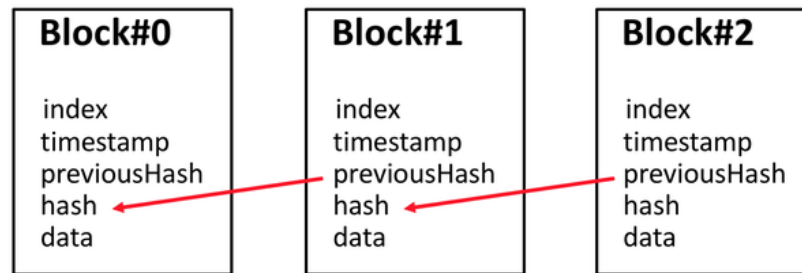
Các yêu cầu chức năng

- Yêu cầu 1: Người dùng đã đăng nhập vào hệ thống.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. LÝ THUYẾT VỀ BLOCKCHAIN

Blockchain (tên ban đầu là block chain) là một cơ sở dữ liệu phân cấp lưu trữ thông tin trong các khối thông tin liên kết với nhau bằng mã hóa và mở rộng theo thời gian. Mỗi khối thông tin đều chứa thông tin về thời gian khởi tạo và được liên kết tới khối trước đó, kèm một mã thời gian và dữ liệu giao dịch. Blockchain được thiết kế để chống lại việc thay đổi của dữ liệu: Một khi dữ liệu đã được mạng lưới chấp nhận thì sẽ không có cách nào thay đổi được đó.



Hình 2.1 Kiến trúc khối Blockchain

Bên trong một khối sẽ có một số thông tin sau đây:

- **index**: vị trí của khối hiện tại, khối genesis sẽ có giá trị index là 0.
- **timestamp**: là thời gian khối được tạo ra.
- **previousHash**: chuỗi hash của khối liền trước.
- **hash**: chuỗi mã hóa không trùng nhau, sử dụng để làm định danh duy nhất cho khối dữ liệu
- **data**: là chuỗi dữ liệu được người dùng lưu trữ vào đó.

Hiện nay đã xuất hiện rất nhiều nền tảng để phát triển, xây dựng mạng blockchain chẳng hạn như:

- **Ethereum**: Ra mắt năm 2015, sử dụng thuật toán đồng thuận Proof-of-work, nền tảng cho phép các nhà phát triển phát triển các ứng dụng dựa trên hợp đồng thông minh.
- **Ripple**: Ra mắt năm 2012, là một nền tảng Blockchain không công khai, xây dựng với mục đích kết nối các ngân hàng, tài sản của công ty và các công ty tài chính thông qua một hệ thống giao dịch toàn cầu có tên là RippleNet. Sử dụng giao thức đồng thuận riêng gọi là Thuật toán đồng thuận giao thức Ripple (RPCA).

Một hệ thống mạng Blockchain sẽ tồn tại rất nhiều nút độc lập, những nút này đều có đầy đủ khả năng xác thực thông tin cũng như chuyển tiếp giao dịch, bảo mật an toàn cao trước khả năng đánh cắp dữ liệu. Khi một nút tham gia vào mạng, nó sẽ nhận được một bản sao của Blockchain (thường gọi là Ledger: sổ cái), điều này sẽ giúp cho mạng được tiếp tục hoạt động khi có những nút của hệ thống gặp vấn đề hoặc ngừng hoạt động.

Cách thức hoạt động:

- **Giao dịch phải được được sinh ra:** tức là phải xảy ra một hoạt động giao dịch nào đó.
- **Giao dịch phải được phát đến một mạng ngang hàng và được xác minh:** tức là mọi thông tin về giao dịch sẽ được ghi lại bao gồm thời gian, vị trí và ai tham gia vào giao dịch...
- **Giao dịch hợp lệ phải được lưu trữ trong khối:** khi giao dịch đã hoàn tất, ta có thể tra cứu lại giao dịch bất cứ lúc nào.
- **Khối (Block) phải nhận được mã băm (hash):** chỉ khi nhận được mã băm (hash) thì một khối mới được thêm vào mạng Blockchain.

Các đặc tính của Blockchain:

- Một mạng lưới minh bạch: Blockchain đảm bảo tính minh bạch trong giao dịch thông qua chuỗi mật mã tinh vi. Những mã này có tính duy nhất.
- Quyền lực được phân phối: Mạng Blockchain phân phối quyền lực thông qua một mạng ngang hàng. Không một tổ chức nào có thể đơn phương tắt hệ thống giao dịch. Dù một tổ chức nào đó tham gia vào hệ thống bị loại bỏ thì hệ thống vẫn tồn tại.
- Tính bảo mật: Các tiêu chuẩn an toàn được nhúng toàn mạng mà không có bất kỳ điểm chịu lỗi nào sẽ không chỉ cung cấp được khả năng bảo mật cao mà còn có cả tính chính xác.
- Quyền sở hữu: Kết hợp với hạ tầng khóa công khai, mạng Blockchain không chỉ ngăn chặn việc lập chi mà còn khẳng định quyền sở hữu của mỗi sản phẩm được lưu thông, cũng như việc đảm bảo mỗi giao dịch là bất biến và không thể hủy bỏ.
- Hợp đồng thông minh (Smart Contract): Là một thuật ngữ diễn tả khả năng tự động tạo các điều kiện và tiến hành của hệ thống máy tính bằng cách sử dụng công nghệ Blockchain. Tất cả quá trình hoạt động của Smart Contract đều được thực thi một cách tự động và không có sự can thiệp nào từ bên ngoài.

Hệ thống Blockchain thường được phân chia thành 3 loại: Public, Private và Permissioned trong đó:

- **Public:** đây là loại Blockchain mà bất kỳ ai cũng có quyền đọc, ghi dữ liệu. Quá trình xác thực giao dịch trên hệ thống mạng Blockchain này đòi hỏi phải có hàng nghìn (hoặc có thể nhiều hơn) nút (Node) tham gia. Vì vậy, hệ thống Blockchain này được đánh giá khá an toàn (vì chi phí để thực hiện một vụ tấn công vào hệ thống là khá cao). Điển hình cho loại Blockchain này là các đồng tiền điện tử như Bitcoin, Ethereum...
- **Private:** loại Blockchain này chỉ cho phép người dùng được quyền đọc dữ liệu, còn quyền ghi thuộc về một tổ chức thứ 3 với độ tin cậy tuyệt đối. Bên thứ 3 này sẽ có toàn quyền quyết định mọi thay đổi trên Blockchain. Thời gian xác thực giao dịch của một mạng Private Blockchain sẽ xảy ra nhanh (vì hệ thống mạng này chỉ cần một lượng nhỏ thiết bị tham gia vào giao dịch). Ripple là

một dạng Blockchain kiểu này, hệ thống này chỉ cần 80% các nút hoạt động ổn định là có thể tiến hành giao dịch.

- **Permissioned:** Hay còn có tên gọi khác là Consortium, là một mạng Blockchain dạng Private được bổ xung thêm một số tính năng nhất định, kết hợp giữa “niềm tin” khi tham gia vào Public và “niềm tin tuyệt đối” khi tham gia vào mạng Private. Ví dụ: Các ngân hàng, các tổ chức tài chính liên doanh sẽ sử dụng mạng Blockchain này.

Cơ chế đồng thuận phân tán đồng đẳng:

Cơ chế đồng thuận phân tán đồng đẳng (hay còn gọi là cơ chế đồng thuận phân quyền) trái ngược với mô hình cổ điển là cơ chế đồng thuận tập trung – có nghĩa là một cơ sở dữ liệu tập trung được dùng để xác định giao dịch.

Một sơ đồ phân tán đồng đẳng chuyển giao quyền lực và sự tin tưởng cho một mạng lưới phân tán đồng đẳng và cho phép các nút trong mạng đẩy liên tục lưu trữ các giao dịch trên một khối (block) công cộng, tạo thêm một chuỗi (chain) độc nhất: Chuỗi khối (blockchain).

Mỗi khối kế tiếp chứa một mã băm của khối trước đó; vì vậy, mã hóa (thông qua hàm hash) được sử dụng để đảm bảo tính xác thực của nguồn giao dịch và loại bỏ sự cần thiết phải có một trung gian tập trung. Sự kết hợp của mã hóa và công nghệ Blockchain đảm bảo rằng một giao dịch sẽ không bao giờ được thực hiện 2 lần.

Các cơ chế đồng thuận trong thuật toán Blockchain:

- **Proof of Work (Bằng chứng công việc):** Cơ chế này xuất hiện trong Bitcoin, Ethereum và hầu hết các loại tiền điện tử đang có trên thị trường. Cơ chế này tốn khá nhiều năng lượng điện.
- **Proof of Stake (Bằng chứng Cổ phần):** có trong Decred, Peercoin. Đặc điểm của nó là phi tập trung hơn nữa, tiêu hao ít năng lượng và khó bị phá hoại.
- **Proof of Authority (Bằng chứng ủy nhiệm):** đây là mô hình tập trung thường thấy trong POA.Network, Ethereum Kovan testnet. Hiệu suất cao, có khả năng mở rộng tốt.
- **Byzantine Fault Tolerance (Đồng thuận chống gian lận):** Thường thấy trong các mạng Hyperledger, Stellar, Dispatch và Ripple. Có hiệu suất cao, chi phí thấp, có khả năng mở rộng tốt nhưng chưa được tin tưởng hoàn toàn.

Ứng dụng Blockchain

- Ứng dụng của Blockchain mà chúng ta thường nghe nói đến nhất là tiền điện tử Bitcoin.
- Trong chuỗi Sản xuất – cung ứng, Blockchain đáp ứng tốt việc truy xuất nguồn gốc sản phẩm. Giúp tăng sự tin tưởng từ người dùng. Ví dụ về những ứng dụng như: iCheck.
- Áp dụng trong ngành y tế, lưu trữ thông tin về kết quả. Blockchain giúp đảm bảo được sự chính xác của thông tin.

Những vấn đề của Blockchain:

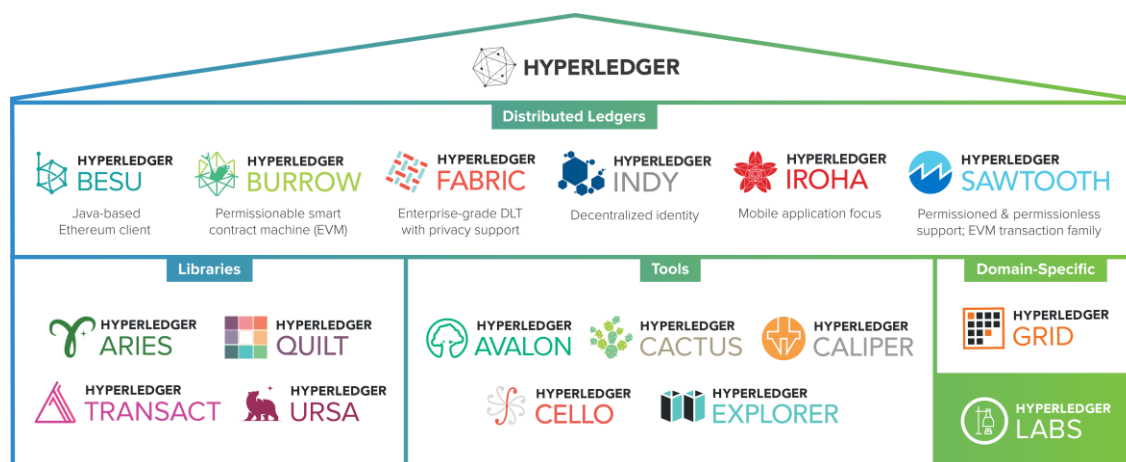
- Tiêu tốn năng lượng điện
- Tốn nhiều không gian lưu trữ: Blockchain cần rất nhiều bản sao chính nó phân tán lên các nút, nên đòi hỏi một khả năng lưu trữ dữ liệu lớn. Tính đến 2019, mạng Blockchain của Bitcoin tiêu tốn đến khoảng 200GB lưu trữ và sẽ ngày càng tăng lên.
- Cần thời gian triển khai: Công nghệ Blockchain có thể thích hợp đối với một số lĩnh vực nhưng triển khai rộng rãi sẽ tốn nhiều thời gian, đặc biệt với các ngành có những cách thức và quy định riêng.
- Sửa đổi dữ liệu: Tính không thể phá vỡ của Blockchain vừa là một điểm lợi cũng vừa là một điểm bất lợi

Tóm lại, Blockchain là một sổ cái giao dịch bất biến, được duy trì trong một mạng lưới phân tán của các nút ngang hàng. Các nút này duy trì một bản sao của sổ cái bằng cách áp dụng các giao dịch đã được xác thực bằng giao thức đồng thuận.

2.2. TỔNG QUAN VỀ HYPERLEDGER FABRIC

2.2.1. Giới thiệu về Hyperledger

Hyperledger là một mã nguồn mở của nền tảng Blockchain, được khởi xướng bởi Linux Foundation. Với Hyperledger, Linux Foundation nhằm mục đích tạo ra môi trường để các nhà phát triển phần mềm phối hợp để xây dựng các framework Blockchain. Hyperledger hợp tác với nhiều hãng công nghệ lớn, nổi bật là IBM. Hyperledger được xây dựng như một hệ sinh thái các giải pháp và người dùng trên nền tảng Blockchain nhằm giải quyết các vấn đề trong các ngành ngân hàng, tài chính, sản xuất, chuỗi cung ứng, công nghệ và cả IoT. Cùng với mục đích ứng dụng Blockchain vào nhiều trường hợp khác nhau và vẫn đảm bảo tính tin cậy, minh bạch của công nghệ.



Hình 2.2 Hệ sinh thái Hyperledger

Các dự án của Hyperledger:

- Hyperledger Sawtooth: là một bộ khối chuỗi mô-đun được phát triển bởi Intel, sử dụng thuật toán đồng thuận mới có tên là Proof of Elapsed Time (PoET) có khả năng xử lý hàng triệu transaction/giây.

- Hyperledger Fabric: cung cấp kiến trúc mô-đun với phân định vai trò giữa các nút trong mạng, thực hiện các hợp đồng thông minh và sự đồng thuận có thể cấu hình và dịch vụ thành viên.
- Hyperledger Iroha: được xây dựng cho cá dự án sử dụng di động, sử dụng thuật toán đồng thuận riêng, mô hình quyền và vai trò phong phú hỗ trợ đa chữ ký.
- Hyperledger Burrow: hoạt động như một công cụ ứng dụng hợp đồng thông minh được cấp phép, và thực hiện theo các tiêu chuẩn của Ethereum Virtual Machine (EVM).
- Hyperledger Indy: hỗ trợ xác định danh tính người dùng.

Các công cụ hỗ trợ phổ biến của Hyperledger:

- Hyperledger Caliper: sử dụng đánh giá hiệu năng mạng Blockchain.
- Hyperledger Cello: hỗ trợ cấu hình mạng cho Blockchain.
- Hyperledger Explorer: hỗ trợ việc quản lý mạng Blockchain.

2.2.2. Giới thiệu về Hyperledger Fabric

Hyperledger Fabric là nền tảng công nghệ sổ cái phân tán (DLT) được cấp phép cấp doanh nghiệp, được thiết kế để sử dụng trong bối cảnh doanh nghiệp, mang lại một số khả năng khác biệt chính so với một số nền tảng sổ cái hoặc Blockchain phổ biến khác. Nổi bật là kiến trúc mô-đun mang lại mức độ bảo mật cao, tính linh hoạt cũng như khả năng mở rộng cao. Với cơ cấu là một pluggable blockchain, nó sẽ cung cấp một set các peer node với những quyền truy cập khác nhau để thao tác với một sổ cái chung.

Hyperledger Fabric có tính mô đun hóa (**modularity**) khá cao nên cho phép các doanh nghiệp dễ dàng **plug and play** để xây dựng một mạng Blockchain phù hợp với yêu cầu của mình.

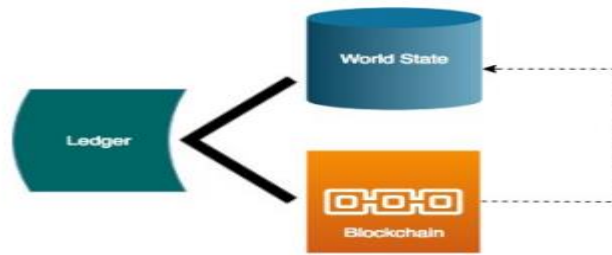
Hyperledger Fabric được xây dựng với công nghệ container để lưu trữ những bộ “chaincode” với bên trong là những luật được cài đặt để vận hành hệ thống.

Hyperledger Fabric hỗ trợ các giao thức đồng thuận kết hợp. Cho phép người dùng xây dựng và bắt đầu mạng được phép chọn cơ chế đồng thuận để đem lại được những điều kiện tốt nhất. Những cơ chế đồng thuận của một mạng Fabric bao gồm: **SOLO, Kafka, Raft**.

Một mạng Fabric cho phép xây dựng nhiều kênh với mỗi kênh là những sổ cái riêng biệt cho hoạt động giao dịch cùng với cơ chế phân quyền cho các tổ chức vào kênh giúp tạo sự bảo mật cho những kênh riêng biệt.

2.2.3. Cấu trúc của một block và blockchain

Một sổ cái của mạng Hyperledger Fabric bao gồm 2 thành phần cụ thể nhưng có liên quan với nhau:



Hình 2.3 Sổ cái bao gồm Blockchain và world state bắt nguồn từ Blockchain

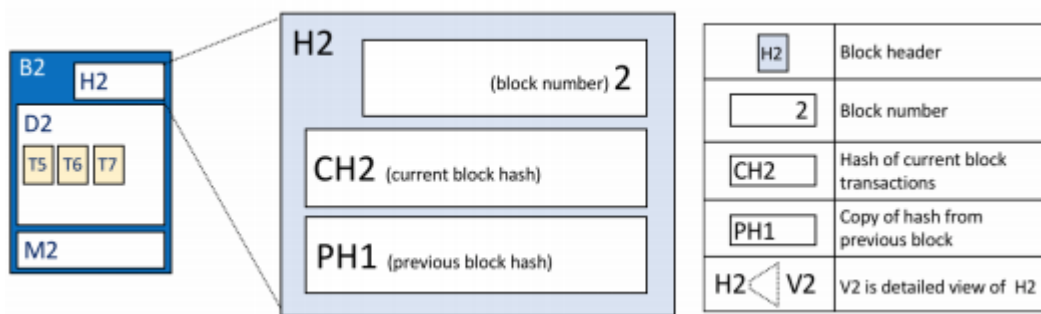
2.2.3.1. World State

Trạng thái thế giới là một cơ sở dữ liệu lưu trữ các giá trị hiện tại của các trạng thái khác nhau của Sổ cái. Giá trị hiện tại của các trạng thái này có thể dễ dàng đọc từ trạng thái Thế giới, tránh việc duyệt qua toàn bộ nhật ký giao dịch và do đó tiết kiệm thời gian. Trạng thái bản ghi hoặc sổ cái được biểu thị dưới dạng các cặp giá trị khóa được cập nhật, tạo và xóa thường xuyên.

2.2.3.2. Chuỗi khối (Blockchain)

Nó là nhật ký của tất cả các giao dịch dẫn đến trạng thái thế giới. Các giao dịch này được tổ chức thành các khối được liên kết với nhau để tạo thành Blockchain cho phép chúng ta ghi lại tất cả lịch sử dẫn đến tình trạng thế giới hiện tại. Không giống như trạng thái thế giới, nó chứa các khối bất biến chứa một tập hợp các giao dịch có thứ tự

2.2.3.3. Các khối (Blocks)



Hình 2.4 Cấu trúc khối Hyperledger

Khối của Hyperledger Fabric bao gồm 3 phần:

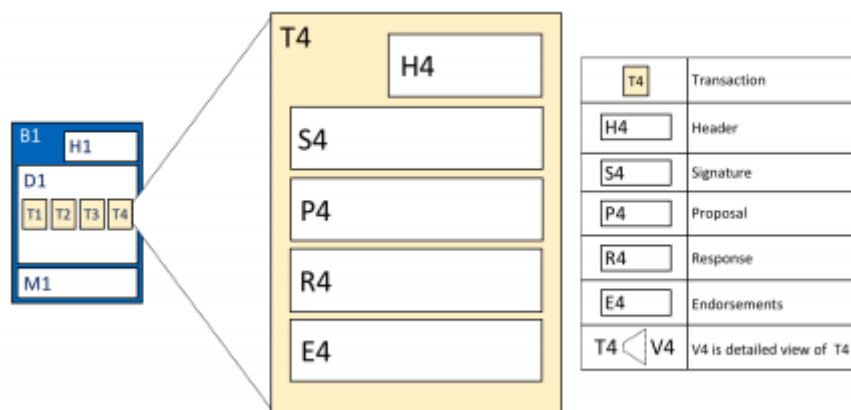
Tiêu đề khối H2 của khối B2, chứa băm của dữ liệu khối hiện tại của D2, số tiêu khối hiện tại là 2 và bản sao với băm của khối trước đó PH1 có số khối là 1.

- Tiêu đề khối (Block Header): chứa 3 trường được ghi khi tạo một khối.
- + Số khối: là một số nguyên bắt đầu từ 0 (đối với khối khởi nguyên) và được tăng thêm 1 cho mỗi khối mới được gắn vào mạng Blockchain.

- + Băm của khối trước đó: chứa giá trị băm của khối trước đó trong Blockchain, điều cần thiết giúp duy trì liên kết giữa các khối.
- + Băm của khối hiện tại: chứa giá trị băm của tất cả giao dịch có trong khối hiện tại.
- Siêu dữ liệu khối (Block Metadata): Siêu dữ liệu khối chứa thời gian khối được thực thi, khóa công khai, chữ ký và chứng chỉ của người viết khối. Sau đó, trình xác nhận khối cũng thêm một cờ cho biết giao dịch có hợp lệ/không hợp lệ hay không. Thông tin về giao dịch hợp lệ/không hợp lệ này không được sử dụng trong hàm băm vì khi tạo khối và cờ này được thêm vào khi ghi vào sổ cái.
- Dữ liệu khối (Block Data): phần này chứa danh sách giao dịch đã đặt hàng. Các giao dịch này được thêm vào dữ liệu khối khi được tạo. Các giao dịch này có cấu trúc chi tiết đơn giản và dễ hiểu.

2.2.3.4. Các giao dịch (Transactions)

Các thay đổi đối với trạng thái thế giới được ghi lại bằng giao dịch. Cấu trúc dữ liệu của khối chứa các giao dịch.



Hình 2.5 Khối giao dịch

Giao dịch T4 bên trong dữ liệu khối D1 của khối B1 chứa chữ ký (signature) giao dịch S4, tiêu đề (header) giao dịch H4, phản hồi (response) giao dịch R4, đề xuất (proposal) giao dịch P4 và danh sách chứng thực (endorsements) E4. Các trường khác nhau của cấu trúc Blockdata chứa các giao dịch như sau:

- Tiêu đề: đại diện bởi H4, chứa siêu dữ liệu quan trọng về giao dịch, ví dụ: tên và phiên bản của mã chaincode.
- Đề xuất: được đại diện bởi P4, bao gồm các tham số đầu vào được cung cấp cho chaincode bởi một ứng dụng tạo bản cập nhật được đề xuất của sổ cái. Đề xuất này cung cấp một tập hợp các tham số đầu vào khi chaincode chạy, cùng với trạng thái thế giới hiện tại, xác định trạng thái thế giới mới.
- Chứng thực: được trình bày trong E4, đây là danh sách các phản hồi giao dịch của mỗi tổ chức đã được ký kết và đủ để đáp ứng chính sách chứng thực (endorsement policy). Chỉ có một phản hồi giao dịch được bao gồm trong giao

dịch mặc dù có thể có nhiều xác nhận. Điều này là do mỗi chứng thực mã hóa một cách hiệu quả phản hồi giao dịch cụ thể từ tổ chức của nó, có nghĩa là việc bao gồm bất kỳ phản hồi giao dịch nào không khớp với sự chứng thực đầy đủ sẽ là một sự lãng phí tài nguyên vì nó sẽ bị từ chối và bị đánh dấu là không hợp lệ và sẽ không cập nhật trạng thái thế giới của sổ cái.

- Chữ ký: chứa một chữ ký mật mã được tạo bởi ứng dụng của máy khách. Nó được sử dụng để kiểm tra xem các chi tiết của giao dịch không bị sửa đổi vì nó yêu cầu khóa riêng của ứng dụng để tạo ra nó.
- Phản hồi: ghi lại các giá trị trước và sau của trạng thái thế giới dưới dạng tập hợp Đọc-Ghi (RW-set). Đây là đầu ra của chaincode và nếu giao dịch được xác thực thành công việc cập nhật trạng thái thế giới sẽ được áp dụng vào sổ cái.

2.2.4. Chức năng mô hình của mạng Hyperledger Fabric

Tính module: Mạng Hyperledger được xây dựng theo hướng mô-đun với các mô-đun sau:

- Một mô-đun Ordering làm nhiệm vụ đồng thuận các giao dịch được tích hợp sẵn các giao thức đồng thuận.
- Một mô-đun về dịch vụ cung cấp thành viên (Membership service provider – MSP) chịu trách nhiệm liên kết các thực thể trong mạng.
- Một mô-đun dịch vụ dịch vụ nhắn tin ngang hàng.
- Một mô-đun hợp đồng thông minh (chaincode) chạy trong môi trường Docker.
- Sổ cái (Ledger).
- Một plugin liên kết các chính sách xác thực.

Quản lý danh tính

Fabric cung cấp dịch vụ nhận dạng thành viên để quản lý ID người dùng và xác thực tất cả người dùng tham gia mạng.

Quyền riêng tư và bảo mật

Fabric sử dụng phương thức bảo mật thông qua kiến trúc của nó. Người dùng tham gia trên Fabric có thể thiết lập các kênh giữa tập con của những người tham gia mà đã được cấp quyền để thực hiện giao dịch. Những người dùng khi tham gia vào mạng nhưng không được cấp quyền trên kênh nhất định thì sẽ không có bất cứ thông tin nào về kênh, thông tin giao dịch hay thành viên của kênh đó.

Assets

Assets có thể bao gồm từ hữu hình đến vô hình, từ bất động sản, phần cứng đến những hợp đồng, tài sản trí tuệ. Hyperledger Fabric sửa đổi assets bằng cách sử dụng cái giao dịch trên chaincode.

Trong Hyperledger Fabric các assets được trình bày dưới dạng tập hợp các cặp key-value, với các thay đổi trạng thái được ghi lại dưới dạng giao dịch trên một sổ cái và được biểu diễn dưới dạng nhị phân hoặc JSON.

Chaincode

Chaincode là phần mềm định nghĩa asset và hướng dẫn giao dịch để thay đổi các asset đó, hiểu theo cách khác nó là logic kinh doanh. Chaincode thực thi các quy tắc để đọc hoặc thay đổi giá trị các cặp key-value hoặc thông tin cá cơ sở dữ liệu trạng thái khác. Các hàm chaincode thực thi dựa trên cơ sở dữ liệu trạng thái hiện tại của sổ cái và được khởi tạo thông qua một đề xuất giao dịch.

Chaincode hoạt động như một ứng dụng phân tán đáng tin cậy có sự bảo mật cao.

Đặc điểm:

- Có thể có nhiều chaincode chạy song song trên mạng.
- Có thể triển khai dễ dàng và linh hoạt.

Hầu như những nền tảng công nghệ Blockchain đều có thể triển khai chaincode theo cấu trúc giao thức đồng thuận xác nhận giao dịch sau đó truyền cho các nút ngang hàng, những nút này mỗi nút sẽ thực hiện các giao dịch tuần tự.

Chaincode có thể được xây dựng bằng các ngôn ngữ lập trình như JavaScript, TypeScript, Go hay Java, thay vì sử dụng ngôn ngữ dành riêng cho nó (DSL).

Ledger

Sổ cái (Ledger) là bản ghi trình tự, chống giả mạo của các chuyển đổi trạng thái là kết quả của tất cả sự thay đổi trạng thái trong kết cấu. Sự thay đổi của trạng thái là kết quả của các lệnh gọi mã giao dịch do các bên tham gia gửi.

Sổ cái bao gồm một chuỗi khối ("chain") để lưu trữ bản ghi bất biến, được lưu trữ trong khối, cũng như là cơ sở để duy trì trạng thái kết cấu hiện tại.

Peer

Peer nhận các cập nhật trạng thái của Ledger theo thứ tự các block từ Orderer. Peer duy trì trạng thái của mạng và một bản sao của sổ cái. Có hai loại Peer khác nhau: **Endorsers** (chứng thực) và **Committers** (cam kết)

- Endorsers: Mô phỏng và thực hiện giao dịch.
- Committers: xác minh xác nhận và xác nhận kết quả giao dịch, trước khi thực hiện giao dịch với Blockchain.

2.2.5. Thuật toán đồng thuận trong Hyperledger Fabric

Sự đồng thuận của Fabric được tạo thành từ ba bước riêng biệt:

- Chứng thực giao dịch.
- Đặt hàng.
- Xác nhận và cam kết.

Fabric hoạt động ở chế độ cấp phép (Permissioned) với cơ chế kiểm soát truy cập và đồng thuận một cách linh hoạt, điều này khiến phạm vi ứng dụng của nó trở nên lớn hơn. Xuất phát từ sự cần riêng tư từ những tổ chức tham gia.

Fabric là một cuốn sổ cái có tính chất trao quyền, có khả năng chia sẻ. Kiến trúc dựa trên nhiều sổ cái hoạt động độc lập nhưng vẫn cho phép giao dịch của một sổ cái có thể tìm kiếm và sử dụng các giao dịch trên sổ cái khác.

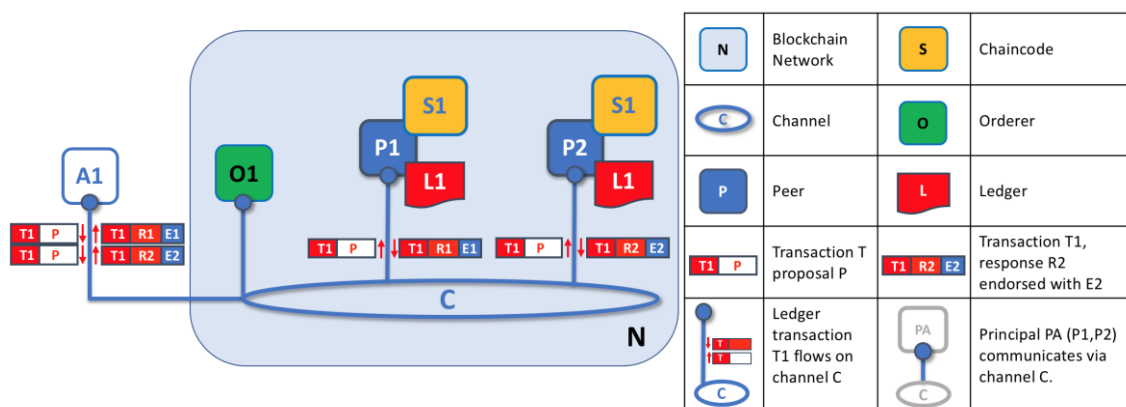
Cơ chế đồng thuận của Hyperledger Fabric sẽ phủ khắp toàn bộ các quá trình giao dịch. Các nút được phân chia những vai trò cùng nhiệm vụ khác nhau trong quá trình đồng thuận. Với 3 vai trò của các nút: client (phía ứng dụng người dùng), Orderer, Peer.

- Client: thực hiện các hành động tạo, hủy các giao dịch, giao tiếp với orderer và peer.
- Peer: có nhiệm vụ thực hiện duy trì sổ cái của mạng, nhận các tín hiệu trạng thái từ orderer, đưa các giao dịch mới vào sổ cái.
- Orderer: có chức năng cung cấp kênh giao tiếp giữa Peer và client, đảm bảo đầy đủ những tín hiệu hoạt động giao dịch được diễn ra đầy đủ trên kênh giao tiếp. Đảm bảo các Peer tham gia sẽ nhận được đầy đủ và phải chính xác cùng một thông điệp theo thứ tự logic.

Cách hoạt động của một cơ chế đồng thuận

Giai đoạn 1: Đề xuất

Ở giai đoạn này của quy trình tạo giao dịch liên quan đến sự tương tác giữa một ứng dụng và một tập hợp các ứng dụng ngang hàng – nó không liên quan đến orderer. Để bắt đầu, các ứng dụng tạo một đề xuất giao dịch mà chúng gửi đến từng nhóm Peer trong mạng để yêu cầu chứng thực. Sau đó, mỗi endorsing peer (peer xác nhận) này sẽ thực thi độc lập cá mã bằng cách sử dụng đề xuất giao dịch để tạo phản hồi đề xuất giao dịch. Nó không áp dụng bản cập nhật này cho sổ cái, mà chỉ đơn giản là ký nó và trả nó vào ứng dụng. Khi ứng dụng đã nhận đủ số lượng phản hồi đề xuất đã ký, giai đoạn đầu tiên sẽ hoàn thành.



Hình 2.6 Mô tả hoạt động của cơ chế đồng thuận mạng fabric

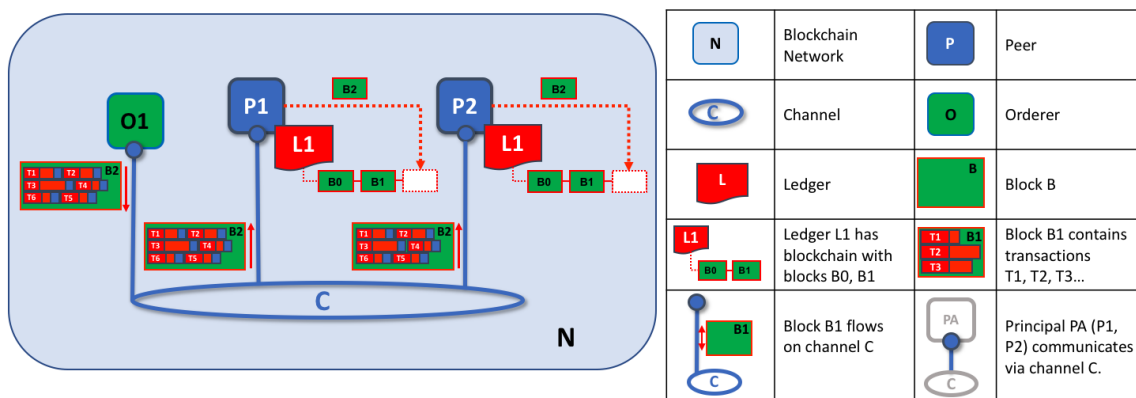
Trong mô hình, ứng dụng A1 tạo giao dịch T1 đề xuất P sau đó gửi lên đến các peer P1 và P2 trên kênh C. P1 thực hiện chaincode S1 bằng cách sử dụng đề xuất giao dịch T1 tạo ra giao dịch T1 kèm phản hồi R1 mà nó xác nhận với E1. Cùng với đó, P2 thực hiện S1 bằng cách sử dụng giao dịch T1 tạo ra giao dịch T1 kèm phản hồi R2 mà nó xác nhận với E2. Ứng dụng A1 nhận lại hai phản hồi xác nhận cho giao dịch T1 từ P1 và P2, với tên là E1 và E2.

Giai đoạn 2: Ordering và đóng gói các giao dịch thành các khối

Giai đoạn này là giai đoạn đóng gói. Orderer đóng vai trò quan trọng trong quá trình này – nó nhận các giao dịch chứa các phản hồi đề xuất giao dịch được xác nhận từ nhiều ứng dụng và sắp xếp các giao dịch lại thành các khối.

Giai đoạn 3: Xác thực và cam kết

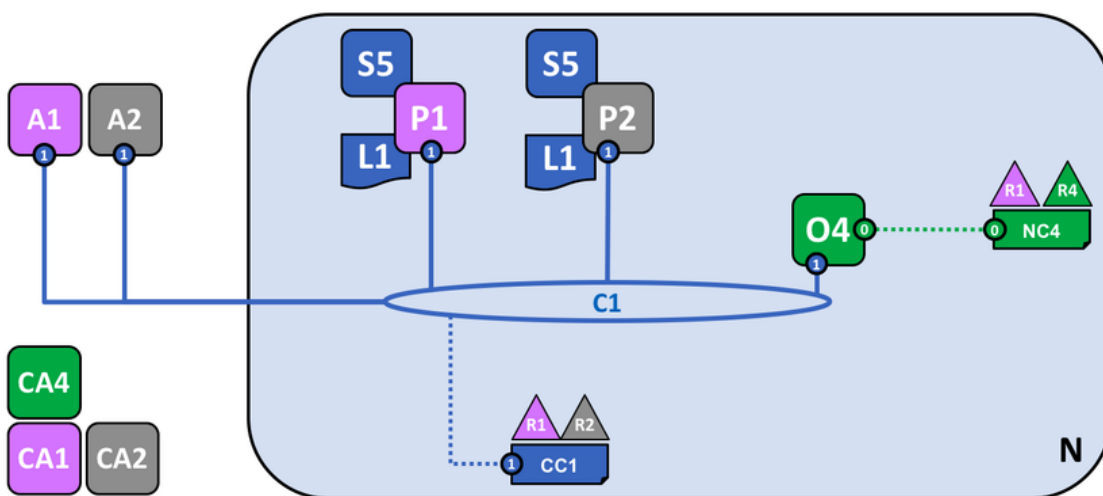
Giai đoạn này là giai đoạn cuối cùng của quy trình, nó liên quan đến việc phân phối và xác nhận tiếp theo của các khối từ orderer đến các peer, tại đây chúng có thể được cam kết với sổ cái. Cụ thể hơn, tại mỗi peer, mọi giao dịch trong khối đều được xác thực để đảm bảo rằng nó đã được các tổ chức có liên quan xác nhận một cách nhất quán trước khi được cam kết vào sổ cái. Các giao dịch thất bại được giữ lại để kiểm tra, nhưng không được cam kết vào sổ cái.



Hình 2.7 Mô tả quá trình 3 của cơ chế đồng thuận

Trong mô hình này, orderer O1 phân phối khối B2 đến peer P1 và P2. Peer P1 xử lý khối B2, dẫn đến một khối mới được thêm vào sổ cái L1 trên P1. Song song đó, peer P2 xử lý khối B2, một khối mới được thêm vào L1 trên P2. Khi quá trình này hoàn tất, sổ cái L1 đã được cập nhật nhất quán trên các peer P1 và P2 và mỗi bên thông báo cho các ứng dụng được kết nối rằng giao dịch đã được xử lý.

2.2.6. Mạng Blockchain Hyperledger Fabric



Hình 2.8 Kiến trúc một mạng Hyperledger Fabric cơ bản

Trong kiến trúc trên ta sẽ thấy được:

- **N** (Network): Mạng Hyperledger Fabric.
- **NC** (Network Config): Cấu hình của mạng.
- **C** (Channel): Kênh, là tập hợp của các tổ chức có vai trò nhất định trong một quy trình hoạt động. Ví dụ, một kênh về mua bán xe hơi sẽ bao gồm 2 tổ chức là: Nhà sản xuất xe hơi và nhà phân phối xe hơi.
- **CC** (Channel Configuration): Cấu hình của kênh.
- **R** (Organization): Tổ chức trong mạng.
- **O** (Orderer Node): Nếu như trong Public Blockchain, tất cả các nút của mạng đều tham gia vào quá trình đồng thuận, thì ở Hyperledger Fabric chỉ có Orderer tham gia vào quá trình đó.
- **P** (Peer Node): là điểm tương tác giữa các thành viên trong tổ chức tương ứng với kênh, mọi hành động của người dùng đều phải đi qua peer.
- **S** (Smart Contract hay Chaincode): được cài đặt trên kênh, định nghĩa rõ các cấu trúc, các hành động mà người dùng có thể thực hiện để tương tác trạng thái của cấu trúc được lưu trong sổ cái.
- **L** (Ledger): sổ cái lưu trạng thái của các đối tượng.
- **CA** (Certificate Authority): phát hành các Identity cho người dùng hoặc nút của tổ chức tương ứng.
- **A** (Application): Ứng dụng hay giao diện (Web, mobile app) giúp người dùng tương tác với hệ thống dễ dàng hơn.

Quy trình xây dựng mạng

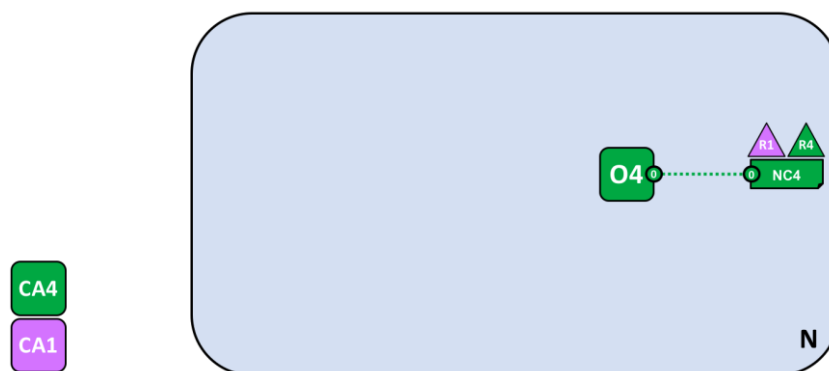
Bước 1: Khởi tạo cơ sở mạng



Hình 2.9 Khởi tạo cơ sở mạng

Mạng được hình thành khi một Orderer được bắt đầu. Trong hình trên, Network, dịch vụ Ordering bao gồm 1 nút duy nhất, orderer O4, được cấu hình theo cấu hình mạng NC4, cấp quyền quản trị cho tổ chức R4.

Bước 2: Thêm quản trị viên mạng



Hình 2.10 Thêm quản trị viên cho mạng

Tổ chức R4 cập nhật cấu hình mạng để tổ chức R1 cũng trở thành quản trị viên của mạng. Sau thời điểm này, tổ chức R1 và R4 có quyền ngang nhau với cấu hình mạng. Cùng với đó, tổ chức phát hành chứng chỉ CA1 cũng được thêm vào, nó được sử dụng để xác định người dùng của tổ chức R1.

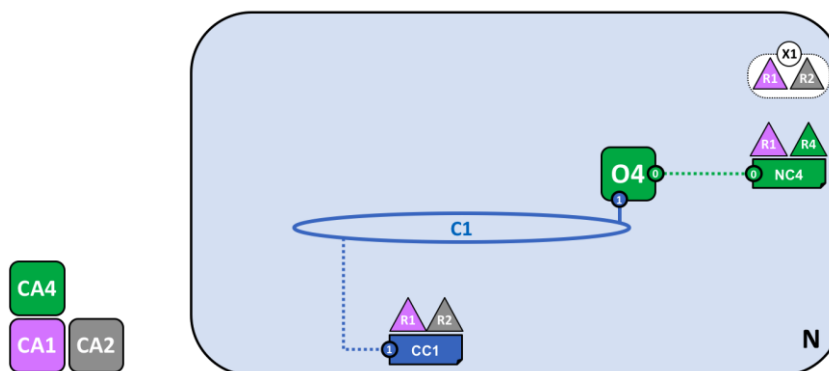
Bước 3: Định nghĩa 1 nhóm



Hình 2.11 Định nghĩa 1 nhóm

Quản trị viên mạng xác định nhóm X1 có 2 thành viên là tổ chức R1 và R2. Định nghĩa nhóm (consortium) này được lưu trong cấu hình mạng NC4 và sẽ được sử dụng ở giai đoạn phát triển mạng tiếp theo. CA1 và CA2 là 2 tổ chức phát hành chứng chỉ tương ứng cho các tổ chức này.

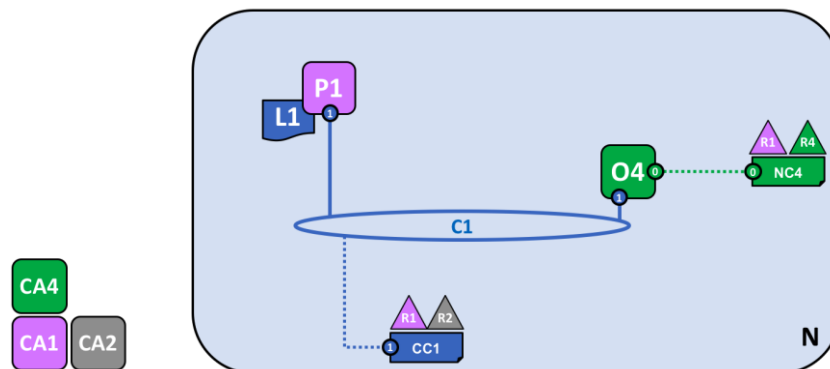
Bước 4: Tạo 1 kênh cho một nhóm



Hình 2.12 Tạo 1 kênh cho một nhóm

Một kênh C1 được tạo ra cho 2 tổ chức R1 và R2 bằng cách sử dụng định nghĩa nhóm X1. Kênh được điều chỉnh bởi cấu hình kênh CC1, hoàn toàn tách biệt với cấu hình mạng. CC1 được quản lý bởi R1 và R2, những người có quyền ngang nhau đối với C1. R4 không có quyền nào trong CC1.

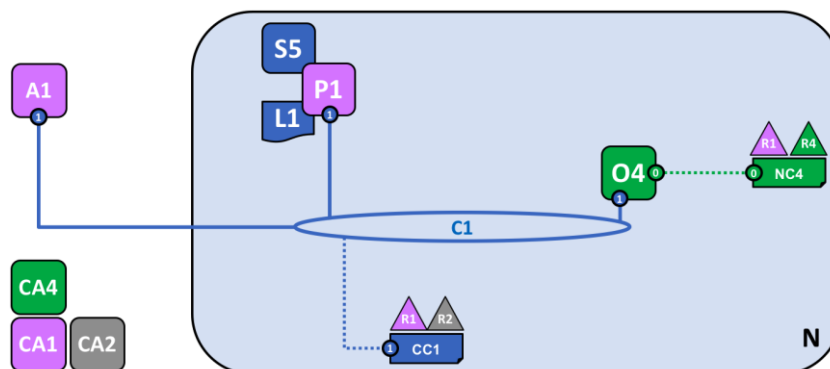
Bước 5: Peer và sổ cái



Hình 2.13 Peer và Ledger

Một nút peer P1 đã tham gia vào kênh C1. P1 về mặt vật lý lưu trữ một bản sao của sổ cái L1. P1 và O4 có thể giao tiếp với nhau qua kênh C1.

Bước 6: Ứng dụng và hợp đồng thông minh chaincode



Hình 2.14 Ứng dụng và hợp đồng thông minh chaincode

Một hợp đồng thông minh S5 đã được cài đặt trên P1. Ứng dụng A1 trong tổ chức R1 có thể sử dụng S5 để truy cập vào sổ cái thông qua nút ngang hàng P1. A1, P1 và O4 đều được kết nối với kênh C1, tức là chúng đều có thể sử dụng các phương tiện liên lạc do kênh đó cung cấp.

Cài đặt một smart contract

Sau khi smart contract S5 được code xong, quản trị viên trong tổ chức R1 phải cài đặt nó vào peer P1. Sau đó, P1 có thể thấy logic triển khai của S5 – code mà nó sử dụng để truy cập vào sổ cái L1.

Khi một tổ chức có nhiều peer trên một kênh, họ có thể chọn các peer mà họ muốn để cài đặt smart contract. Không cần phải cài đặt một smart contract trên mọi peer.

Khởi tạo một smart contract

Để các thành phần khác được kết nối với kênh C1 biết về smart contract vừa được cài đặt, ta phải khởi tạo nó trên kênh C1. Ở đây, chỉ có 1 peer P1, một quản trị viên trong tổ chức R1 phải khởi tạo S5 trên kênh C1 bằng cách sử dụng P1. Sau khi khởi tạo, mọi thành phần trên kênh C1 đều biết về sự tồn tại của S5. Có nghĩa là bây giờ S5 có thể được gọi bởi ứng dụng khách A1

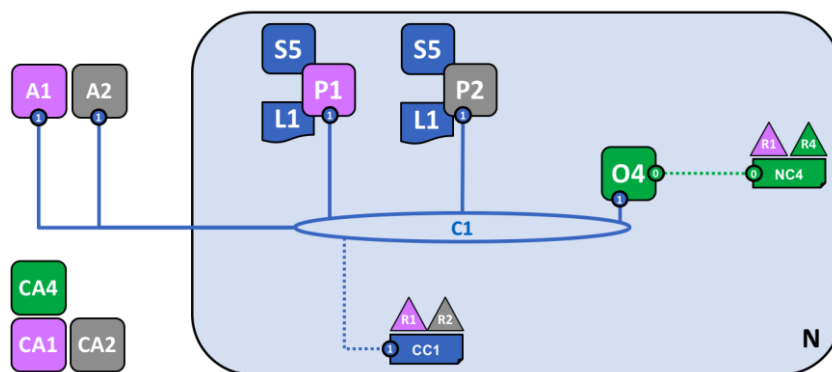
Thiết lập chính sách xác thực

Mô tả các tổ chức nào phải phê duyệt các giao dịch trước khi chúng được các tổ chức khác chấp nhận vào bản sau của sổ cái. Định nghĩa như kiểu R1 AND R2 hoặc R1 OR R2

Gọi hợp đồng thông minh

Khi một smart contract được cài đặt trên một peer và được khởi tạo trên 1 kênh, nó có thể được gọi bởi một ứng dụng. Các ứng dụng thực hiện điều này bằng cách gửi các đề xuất giao dịch cho các peer thuộc sở hữu của các tổ chức được chỉ định bởi chính sách chứng thực. Những đề xuất giao dịch đóng vai trò là tham số đầu vào cho smart contract, sử dụng nó để tạo ra phản hồi giao dịch được chứng thực, được trả lại bởi peer cho ứng dụng.

Bước 7: Hoàn thành mạng



Hình 2.15 Hoàn thành mạng

Mạng đã phát triển thông qua việc bổ sung cơ sở hạ tầng mạng cho tổ chức R2. Cụ thể, R2 đã thêm nút ngang hàng P2 lưu trữ một bản sao của sổ cái L1 và chaincode S5. R2 phê duyệt cùng một định nghĩa chaincode tương tự R1. P2 cũng tham gia vào kênh C1, cũng như ứng dụng A2. A2 và P2 được xác định bằng chứng chỉ từ CA2. Tất cả điều này có nghĩa là cả hai ứng dụng A1 và A2 đều có thể gọi chaincode S5 trên C1 bằng cách sử dụng các nút peer P1 hoặc P2.

2.2.7. Nền tảng Docker

Docker là một nền tảng cho developer và sysadmin để phát triển, triển khai và chạy ứng dụng với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần triển khai lên bất kỳ server nào chỉ cần chạy container của Docker thì ứng dụng của bạn sẽ được khởi chạy ngay lập tức

Lợi ích của Docker

- Không như máy ảo Docker chỉ bắt đầu và dừng chỉ trong vài giây.
- Bạn có thể khởi chạy container trên mỗi hệ thống bạn muốn.
- Container có thể build và loại bỏ nhanh hơn máy ảo.
- Dễ dàng thiết lập môi trường làm việc. Chỉ cần cấu hình một lần duy nhất và không bao giờ phải cài đặt những phụ thuộc ràng buộc. Nếu bạn thay đổi máy hoặc có người mới tham gia vào dự án bạn chỉ cần lấy cấu hình đó và đưa cho họ.
- Nó giữ cho word-space của bạn được sạch sẽ hơn khi bạn xóa môi trường mà ảnh hưởng đến các phần khác.

2.2.8. Hyperledger Fabric Client SDK

Hyperledger Fabric Client SDK giúp bạn có thể sử dụng các API để tương tác với mạng Blockchain Hyperledger Fabric.

- fabric-network: Gói này đóng gói các API để kết nối với mạng Fabric, gửi các giao dịch và thực hiện truy vấn đối với sổ cái ở mức cao hơn thông thường.
- fabric-ca-client: Cho phép các ứng dụng đăng ký Peer và ứng dụng người dùng để thiết lập danh tính đáng tin cậy trên mạng Blockchain.

2.3. NODEJS LÀ GÌ

NodeJS là một nền tảng phát triển độc lập được xây dựng trên Javascript Runtime của Chrome để xây các ứng dụng nhanh, có độ lớn. NodeJS sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

ExpressJS là một framework được xây dựng trên nền tảng của NodeJS, nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. ExpressJS hỗ trợ các method HTTP và middleware tạo ra các API vô cùng mạnh mẽ và dễ sử dụng.

2.4. TỔNG QUAN VỀ REACTJS

ReactJs là một thư viện JavaScript có tính hiệu quả và linh hoạt để xây dựng các thành phần giao diện người dùng (UI) có thể sử dụng lại. ReactJS giúp phân chia các UI phức tạp thành phần nhỏ (được gọi là component). ReactJS được dùng để xây dựng các ứng dụng [single page application] (SPA).

2.5. TỔNG QUAN VỀ REACT NATIVE

React Native là một framework được tạo ra bởi Facebook, cho phép developer xây dựng các ứng dụng di động trên cả Android và iOS chỉ với một ngôn ngữ duy nhất: JavaScript.

Sự ra đời của React Native là lời giải cho bài toán liên quan đến hiệu năng Hybrid và sự phức tạp khi viết nhiều loại ngôn ngữ native cho từng nền tảng di động trước đó.

Ưu điểm của React Native:

- Sử dụng ít native code hơn.
- Rút ngắn thời gian phát triển ứng dụng.
- Tiết kiệm chi phí nhân lực cho doanh nghiệp.
- Khả năng tái sử dụng code cao.
- Không cần nhiều kiến thức nền.

Nhược điểm:

- Hiệu năng thấp hơn so với các ứng dụng thuần native code
- Độ bảo mật thấp
- Khả năng tùy biến không thực sự tốt với một vài module.

2.6. TỔNG QUAN VỀ FIREBASE

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên tầng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác trên cơ sở dữ liệu

Firebase cung cấp nhiều dịch vụ khác nhau.

Cloud Firestore là một cơ sở dữ liệu linh hoạt dễ mở rộng, hỗ trợ đồng bộ dữ liệu một cách nhanh chóng.

CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

3.1. TỔNG QUAN HỆ THỐNG

3.1.1. Về chức năng

Nền tảng quan trắc môi trường nước dựa trên công nghệ Blockchain cung cấp hai loại người dùng quản trị hệ thống và người dùng cuối:

- Quản trị hệ thống: Xem danh sách thiết bị trong hệ thống, kích hoạt thiết bị cho phép tham gia vào mạng Blockchain, thông báo cho người dùng qua email khi đã kích hoạt thành công và cấp lại token cho thiết bị khi người dùng yêu cầu.
- Người dùng cuối: thêm thiết bị, quản lý các thiết bị đang có, gửi dữ liệu từ các thiết bị lên hệ thống, xem thiết bị dữ liệu cảm biến, có thể chia sẻ thiết bị cho người dùng khác trên từng cảm biến có trên thiết bị, xuất dữ liệu ra file,...

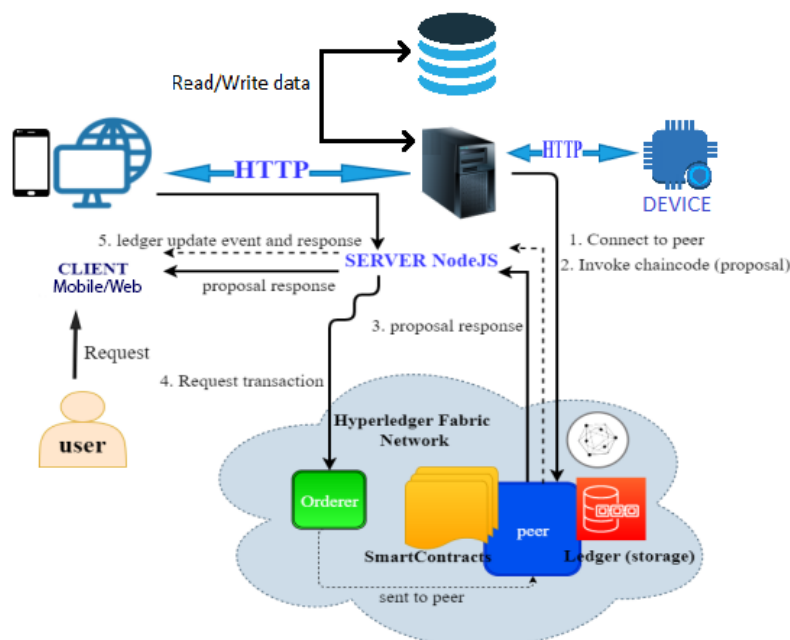
3.1.2. Về thiết kế

Nền tảng được xây dựng từ các thành phần chính: mạng Hyperledger Fabric được triển khai các mã chaincode để người dùng tương tác với sổ cái, server NodeJS dựa trên framework ExpressJS, dịch vụ cơ sở dữ liệu Firebase để lưu trữ và xác thực người dùng và ReactJS, React Native để xây dựng giao diện người dùng.

3.2. KIẾN TRÚC HỆ THỐNG

3.2.1. Sơ đồ tổng quan hệ thống

Hệ thống xây dựng bao gồm các thành phần: Server NodeJS, mạng Hyperledger Fabric, Firebase và giao diện người dùng (Web và ứng dụng di động) được thể hiện như hình bên dưới:



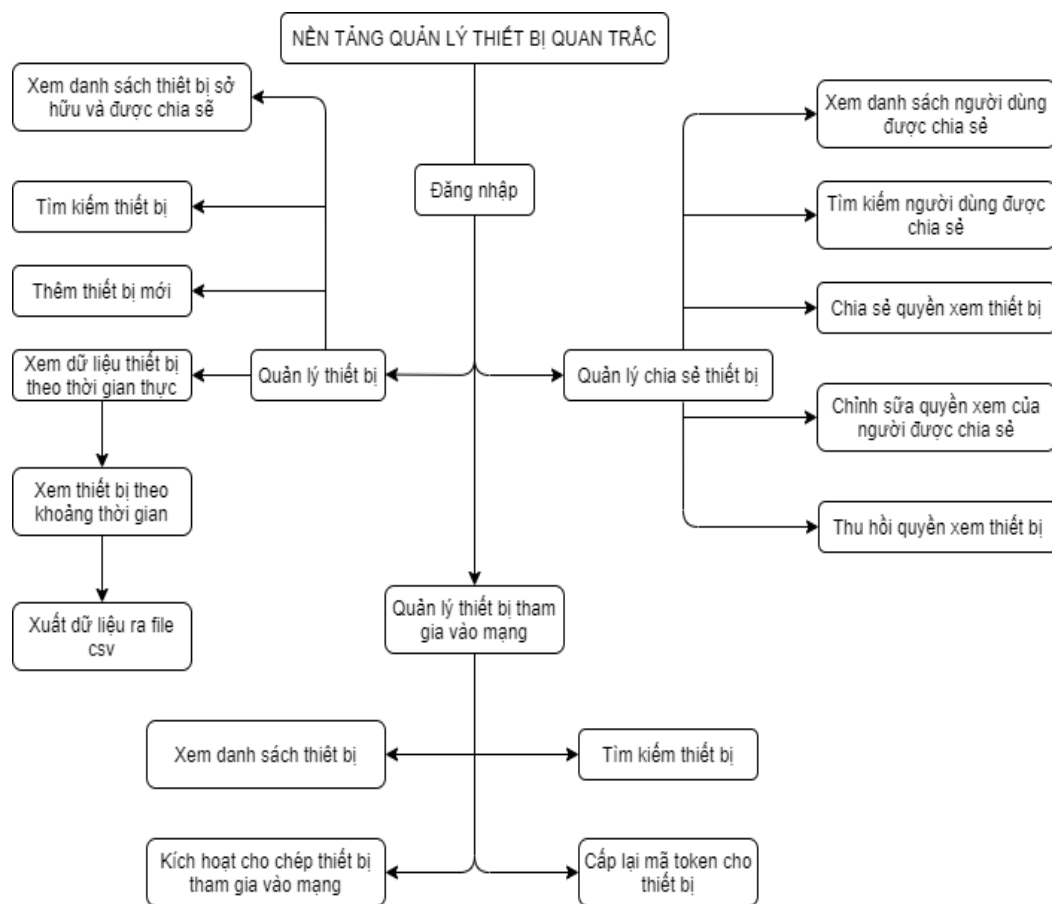
Hình 3.1 Sơ đồ tổng quản hệ thống

Người dùng truy cập hệ thống thông qua trình duyệt Web hoặc ứng dụng di động để yêu cầu xác thực người dùng bằng cách gọi API tương ứng lên server NodeJS, tại

đây server sẽ tương tác với cơ sở dữ liệu đám mây Firebase để thực hiện xác thực người dùng. Khi người dùng đã được xác thực thành công sẽ có thể thực hiện các chức năng của mình như: xem dữ liệu thiết bị, thêm thiết bị... Khi người dùng cần đọc dữ liệu của thiết bị sẽ gửi yêu cầu đến server NodeJS sau đó server sẽ tương tác với mạng Hyperledger Fabric để xác thực định danh thiết bị trên mạng và trả về dữ liệu cho người dùng. Đối với thiết bị quan trắc khi gửi dữ liệu lên mạng cũng phải thông qua server NodeJS để xác thực thiết bị bằng mã token được kí bởi thông tin định danh thiết bị trong mạng Hyperledger Fabric khi xác thực thành công thì sẽ cập nhật dữ liệu vào mạng

3.2.2. Sơ đồ phân rã chức năng

Hệ thống được xây dựng bao gồm các chức năng chính như quản lý thiết bị, quản lý chia sẻ thiết bị và quản lý thiết bị tham gia vào mạng. Các chức năng được mô tả như hình bên dưới đây:



Hình 3.2 Mô tả sự phân rã của hệ thống

3.3. THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG

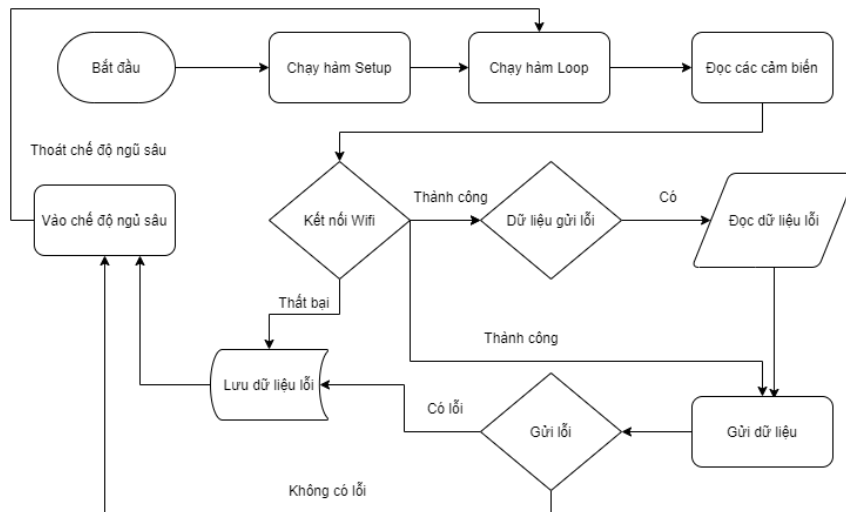
3.3.1. Lưu đồ hoạt động của thiết bị quan trắc

Lưu đồ hoạt động mô của thiết bị biểu diễn qui trình hoạt động của thiết bị. Người dùng có thể hiểu được hoạt động của nó một cách dễ dàng.

Theo lưu đồ, qui trình hoạt động của thiết bị sẽ diễn ra như sau:

- Bắt đầu, thiết bị sẽ chạy hàm Setup và chạy hàm Loop

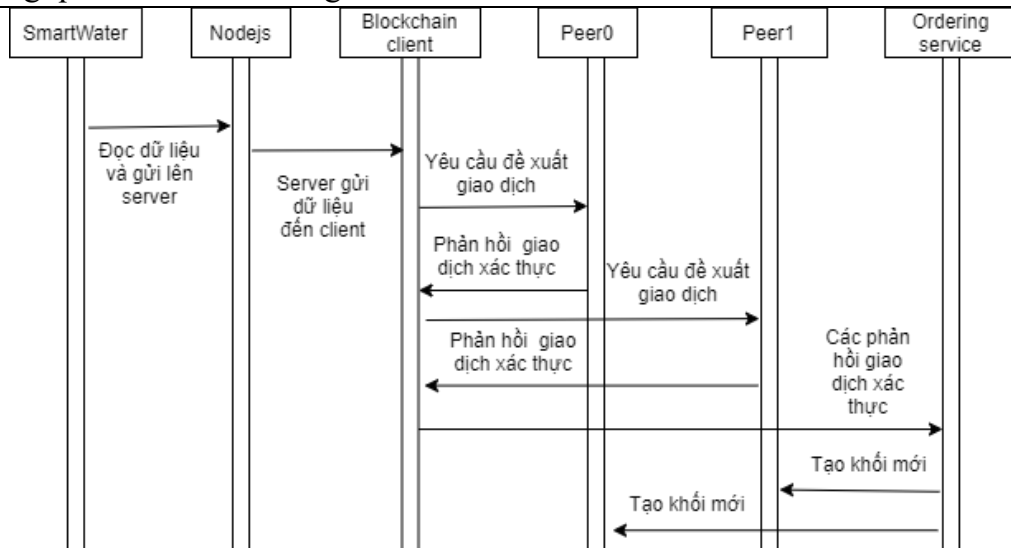
- Đọc các dữ liệu từ các cảm biến và kết nối đến wifi.
- Sau đó sẽ xảy ra 2 trường hợp:
 - + Kết nối wifi thất bại: Lưu lại dữ liệu lỗi và vào chế độ ngủ sâu
 - + Kết nối wifi thành công: Thiết bị sẽ tiếp tục 2 hoạt động. Đầu tiên, nó sẽ kiểm tra có dữ liệu lỗi không để gửi cùng dữ liệu mới. Sau đấy, thực hiện gửi nếu lỗi sẽ lưu lại dữ liệu lỗi và vào chế độ ngủ sâu. Nếu không lỗi gửi thành công và vào chế độ ngủ sâu.



Hình 3.3 Lưu đồ hoạt động thiết bị

Trình tự giao tiếp như sau:

- Thiết bị SmartWater đọc các giá trị từ cảm biến.
- Dữ liệu được đóng vào dấu thời gian và sau đó gửi đến server NodeJS.
- Sau đó, tải trọng này được xử lý bởi ứng dụng Nodejs và một đề xuất giao dịch được tạo.
- Giao dịch được đề xuất sau đó được ký bởi ít nhất một đồng nghiệp từ mỗi tổ chức.
- Sau đó, đề xuất giao dịch đã ký này được đệ trình lên mạng Blockchain.
- Bên trong mạng Blockchain, chaincode được gọi và các chức năng thích hợp được gọi.
- Sau đó, chaincode thực hiện các cập nhật cho Sổ cái và trả về một phản hồi.
- Sau đó, một phản hồi được tạo bởi chaincode và trả về ứng dụng.



Hình 3.4 Luồng hoạt động thiết bị

3.3.2. Mô hình cơ sở dữ liệu và quản lý người dùng

3.3.2.1. Quản lý và xác thực người dùng

Hệ thống sử dụng dịch vụ Firebase Authentication do Firebase cung cấp cho phép người dùng đã sở hữu tài khoản Google đều có thể đăng nhập vào hệ thống. Khi người dùng đã truy cập vào hệ thống thì Firebase sẽ tạo ra một UID duy nhất và các thông tin liên quan đến tài khoản Google lưu vào Firebase mọi hoạt động của người dùng sẽ được quản lý dựa trên các thông tin này.

Search by email address, phone number, or user UID					Add user		
Identifier	Providers	Created	Signed In	User UID			
taib1606931@student.ctu.e...		Nov 20, 2020	Jan 2, 2021	0eJGpZ5a3fVanlcj7wAgw8Wb9oz1			
tranminhtai1998@gmail.com		Nov 19, 2020	Jan 2, 2021	GsUT8FNT05ZTHChdFZrsQB9mcs...			
iot.blockchain.2020@gmail...		Nov 20, 2020	Jan 2, 2021	ZCXgDCVTv1PFetiH70XIza7TzWw2			
minhtai@admin.iotfabric.com		Dec 13, 2020	Dec 23, 2020	oeASxd87fSgtHazTdCoQTGKKCKt1			
Rows per page: 50					1 - 4 of 4	<	>

Hình 3.5 Thông tin người dùng trong Firebase

3.3.2.2. Cơ sở dữ liệu

Firebase cung cấp dịch vụ Cloud Firestore hoạt động như một cơ sở dữ liệu NoSQL và hỗ trợ truy vấn thời gian thực giúp đồng bộ hóa dữ liệu phía client một cách nhanh chóng và dễ dàng.

Xây dựng cơ sở dữ liệu với 4 Collection:

- devices: collection chứa thông tin thiết bị.
- bcAccounts: collection chứa định danh của thiết bị trong mạng Blockchain.
- fieldRef: collection chứa thông tin các cảm biến được chia sẻ của người dùng.
- adminUser: chứa thông tin người dùng quản trị.

Cấu trúc của collection **devices**:

STT	Tên thuộc tính		Kiểu dữ liệu	Diễn giải
1	_id		ObjectID	ID tài liệu (Tự động tạo)
2	auth		String	UID chủ sở hữu thiết bị
3	date		String	Ngày giờ thêm thiết bị
4	desc		String	Mô tả thiết bị
5	email		String	Email chủ sở hữu thiết bị
6	name		String	Tên thiết bị
7	actived		Boolean	Trạng thái kích hoạt
8	refUser		Array	Mảng ID người dùng được chia sẻ
9	warning		Boolean	Tùy chọn cảnh báo
10	phoneNumber		String	Số điện thoại nhân cảnh báo
11	dataFields	fieldDisplay	String	Tên hiển thị của cảm biến
		fieldName	String	Tên cảm biến
		fieldUnit	String	Đơn vị của cảm biến

Bảng 3.1 Cấu trúc collection devices

Cấu trúc collection **bcAccounts**:

STT	Tên thuộc tính		Kiểu dữ liệu	Diễn giải
1	_id		ObjectID	ID tài liệu (Tự động tạo)
2	auth		String	UID người sở hữu
3	bcIdentity		String	Định danh người dùng trong mạng Blockchain
4	deviceId		String	ID thiết bị
5	provider(tùy chọn)		String	ID người cung cấp định danh

Bảng 3.2 Cấu trúc collection bcAccounts

Cấu trúc collection **fieldRef**:

STT	Tên thuộc tính		Kiểu dữ liệu	Miêu tả
1	_id		ObjectId	ID tài liệu (Tự động tạo)
2	auth		String	UID người sở hữu
3	deviceId		String	ID thiết bị
4	dataFields	fieldDisplay	String	Tên hiển thị của cảm biến
		fieldName	String	Tên cảm biến
		fieldUnit	String	Đơn vị của cảm biến
		share	Boolean	Trạng thái được chia sẻ

Bảng 3.3 Cấu trúc collection fieldRef

Cấu trúc collection **adminUser**:

STT	Tên thuộc tính	Kiểu dữ liệu	Miêu tả
1	_id	ObjectId	ID tài liệu
2	admin	Boolean	Là admin hay không

Bảng 3.4 Cấu trúc collection adminUser

3.3.3. Thiết kế và cài đặt Chaincode (Smartcontract)

Xây dựng cấu trúc của dữ liệu của các asset sẽ được ghi vào ledger. Dữ liệu của mạng sẽ bao gồm giá trị của các cảm biến từ thiết bị gửi lên cùng với thông tin thiết bị và thời gian các giá trị này tạo thành.

Các thiết bị có thể có những cảm biến khác nhau do những mục đích sử dụng khác nhau của người dùng nên cấu trúc dữ liệu của một thiết bị sẽ không cố định. Vì vậy ở đây cấu trúc của một asset sẽ bao gồm ID của thiết bị, điểm thời gian và một phần dữ liệu chung với những trường tùy chọn tương ứng với các cảm biến cho mỗi thiết bị.

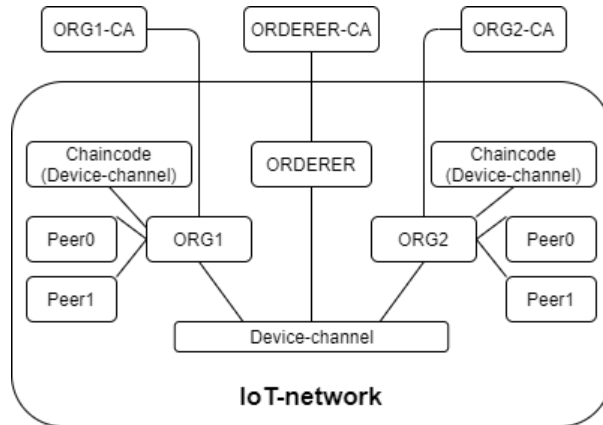
Một ví dụ cụ thể ở đây, thiết bị G có các cảm biến nhiệt độ, độ ẩm được đặt tên tương ứng là temperature, humidity dữ liệu sẽ được lưu lại bao gồm ID thiết bị, điểm thời gian và một đối tượng chứa temperature và humidity.



Hình 3.6 Cấu trúc assets

3.3.4. Cài đặt và triển khai mạng Hyperledger Fabric

Mạng Hyperledger IoT-network được thiết lập bao gồm 2 tổ chức Org1 và Org2, trong mỗi tổ chức sẽ bao gồm hai nút peer0 và peer1. Với mỗi tổ chức sẽ có một cơ quan cấp phát chứng chỉ tương ứng CA1 cho Org1 và CA2 cho Org2. Miền Org1 và Org2 được sử dụng các nút của 2 tổ chức khác nhau với những chức năng của tổ chức cũng khác nhau. Tổ chức Orderer và nút Orderer sử dụng cơ chế đồng thuận SOLO làm phương thức orderer.



Mạng Blockchain được thiết lập, tạo kênh và thêm các nút vào kênh, sau cùng chaincode sẽ được triển khai lên kênh theo cách sau:

- **Bước 1:** Cài đặt Hyperledger Fabric.
- **Bước 2:** Thiết lập và cấu hình mạng IoT-network để khởi tạo các Docker container cần thiết kênh thêm các peer vào kênh đăng ký các CA cần thiết cho chúng.
- **Bước 3:** Cài đặt chaincode lên các nút peer trong kênh.
- **Bước 4:** Commit và approved chaincode vào kênh

3.4. THIẾT KẾ CHỨC NĂNG

3.4.1. Các chức năng cho quản trị viên

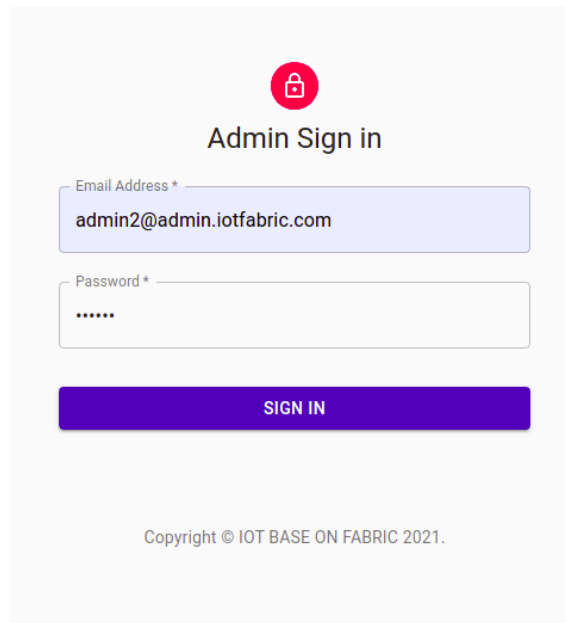
3.4.1.1. Chức năng đăng nhập

Mục đích

- Cho phép quản trị viên đăng nhập vào hệ thống.
- Điều kiện tiên quyết: Người dùng đã có tài khoản

Giao diện

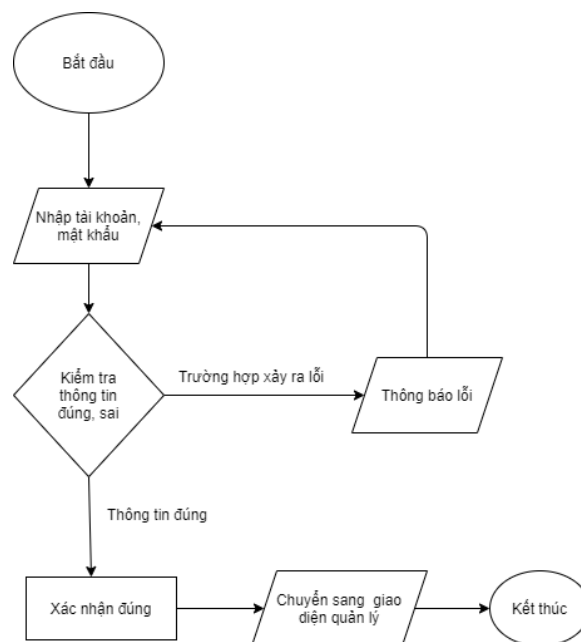
Giao diện với 2 text-field và nút để Sign-in khi người dùng nhập xong thông tin



The image shows a web form titled "Admin Sign in". At the top, there is a red lock icon. Below the title, there are two input fields: "Email Address *" with the value "admin2@admin.iotfabric.com" and "Password *" with masked characters "*****". A blue "SIGN IN" button is positioned below the password field. At the bottom, there is a copyright notice: "Copyright © IOT BASE ON FABRIC 2021."

Hình 3.7 Chức năng đăng nhập phía admin

Sơ đồ xử lý



Hình 3.8 Lưu đồ đăng nhập với quản trị viên

3.4.1.2. Chức năng quản lý thiết bị

Mục đích

- Xem danh sách thiết bị và thực hiện các thao tác quản lý thiết bị.
- Điều kiện tiên quyết: Đã đăng nhập vào hệ thống quản trị.

Giao diện

Giao diện thiết bị là một bảng với danh sách các thiết bị cùng những chức năng bên trong.

QUẢN LÝ THIẾT BỊ

Nhập từ khóa để tìm kiếm

Thời gian	ID thiết bị	Tên thiết bị	Chủ sở hữu	Thông tin thiết bị	Trạng thái	Làm mới token
12/7/2020, 5:57:33 PM	7bcQzH8tlwiqCcCnCGew	Waspmost SW1	taib1606931@student.ctu.edu.vn	Hồ cá giống 105:11		
12/7/2020, 6:10:46 PM	7x0U5ikwNYunS2lgHRCm	Waspmost SW2	taib1606931@student.ctu.edu.vn	Hồ cá giống 105:23		
12/7/2020, 6:10:46 PM	DDViqU5it8Qn2antSvOd	Waspmost SW3	tranminhtai1998@gmail.com	Ao sen 1		
12/7/2020, 6:10:46 PM	eSPwLdlIHieVwrUMsk5	Waspmost Test1	taib1606931@student.ctu.edu.vn	Hồ thủ nước 1		
12/7/2020, 6:10:46 PM	y1ad3JCYbqonY5zGCA93	Waspmost SW4	tranminhtai1998@gmail.com	Ao sen 2		

Rows per page: 10 1-5 of 5

Hình 3.9 Quản lý thiết bị phía admin

3.4.1.3. Chức năng kích hoạt thiết bị

Mục đích

- Kích hoạt thiết bị tạo định danh lưu trữ dữ liệu trên hệ thống.
- Điều kiện tiên quyết: Đã đăng nhập vào hệ thống.

Giao diện

Giao diện là một form gồm 1 text-field để xác nhận chính xác ID thiết bị cùng 2 nút Hủy Bỏ và Đồng Ý

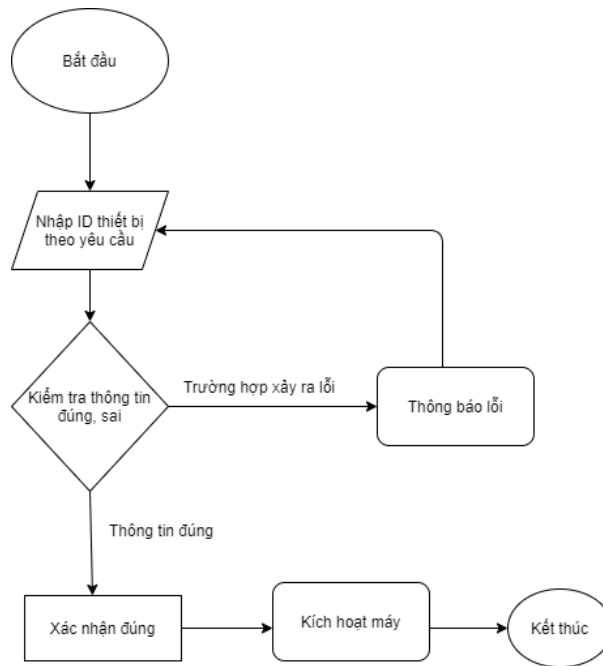
ID thiết bị	Tên thiết bị	Chủ sở hữu
7bcQzH8tlwiqCcCnCGew	Waspmost SW1	taib1606931@student.ctu.edu.v
7x0U5ikwNYunS2lgHRCm	Waspmost SW2	taib1606931@student.ctu.edu.v
DDViqU5it8Qn2antSv		om
eSPwLdlIHieVwrUMs		edu.v
y1ad3JCYbqonY5zGCA93	Waspmost SW4	tranminhtai1998@gmail.com

ID Thiết bị: 7bcQzH8tlwiqCcCnCGew

Vui lòng nhập 7bcQzH8tlwiqCcCnCGew để xác nhận tạo máy.

HỦY BỎ **ĐỒNG Ý**

Hình 3.10 Kích hoạt thiết bị



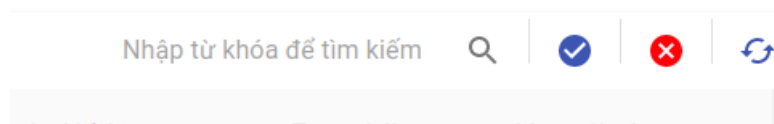
Hình 3.11 Lưu đồ kích hoạt thiết bị

3.4.1.4. Chức năng tìm kiếm thiết bị

Mục đích

- Tìm kiếm thiết bị trong một danh sách với nhiều thiết bị.
- Điều kiện tiên quyết: Đã đăng nhập vào hệ thống.

Giao diện









Hình 3.12 Tìm kiếm thiết bị (Admin)

3.4.1.5. Chức năng cấp lại token cho thiết bị

Mục đích

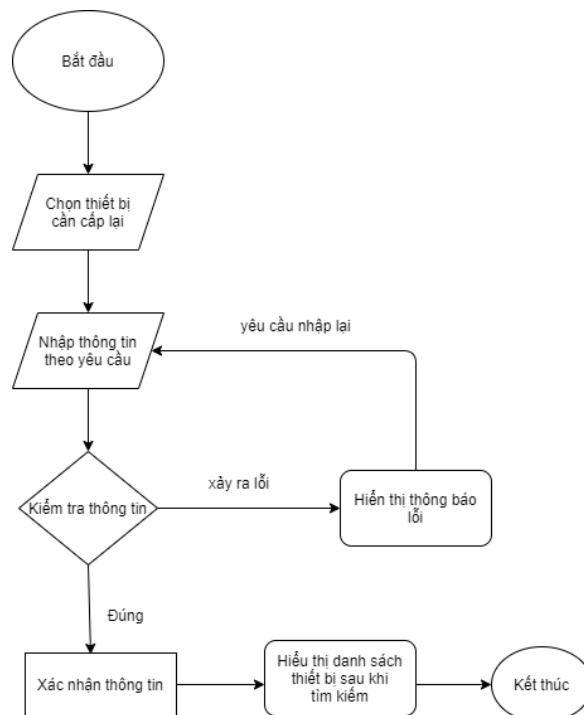
- Cấp lại token cho người dùng
- Điều kiện tiên quyết: đã đăng nhập vào hệ thống.

Giao diện

Tên thiết bị	Trạng thái	Làm mới token
giống 105:11		
giống 105:23		
n 1		

Hình 3.13 Cập lại token thiết bị

Sơ đồ xử lý



Hình 3.14 Lưu đồ chức năng cập lại token

3.4.2. Các chức năng cho người dùng cuối

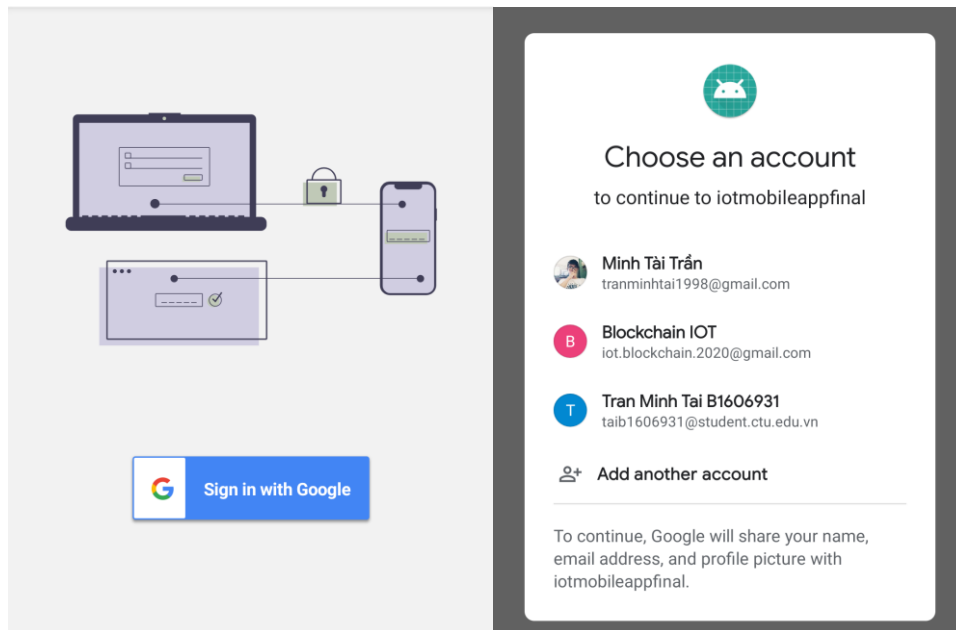
3.4.2.1. Chức năng đăng nhập

Mục đích

- Giúp người dùng truy cập vào hệ thống.
- Điều kiện tiên quyết: người dùng đã truy cập vào ứng dụng.

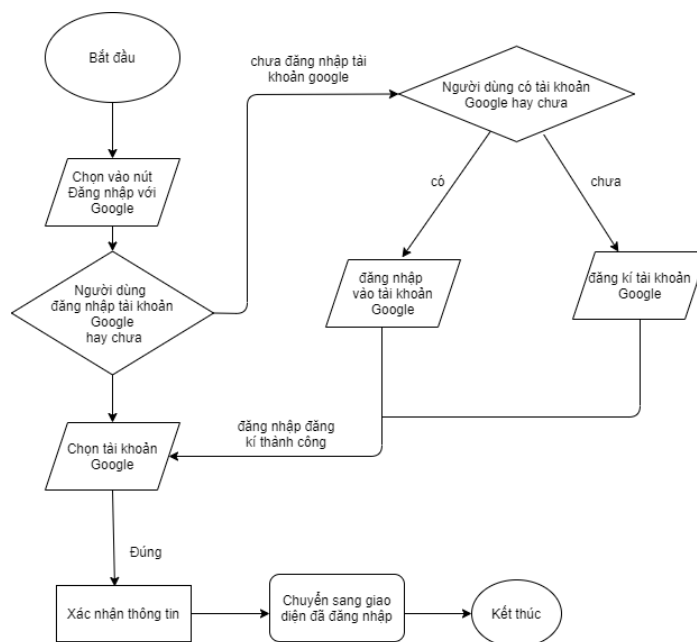
Giao diện

Giao diện sẽ bao gồm 1 nút nhấn để hiển thị SignIn với Google, một popup chọn tài khoản với Google nếu người dùng đã đăng nhập Google từ trước hoặc có thể đăng nhập và đăng ký nếu cần thiết.



Hình 3.15 Đăng nhập phía người dùng di động

Sơ đồ xử lý



Hình 3.16 Lưu đồ đăng nhập với người dùng

3.4.2.2. Chức năng tạo thiết bị

Mục đích

- Gửi yêu cầu tạo định danh thiết bị quan trắc.
- Điều kiện tiên quyết: yêu cầu người dùng đăng nhập vào hệ thống.

3.4.2.3. Chức năng quản lý thiết bị

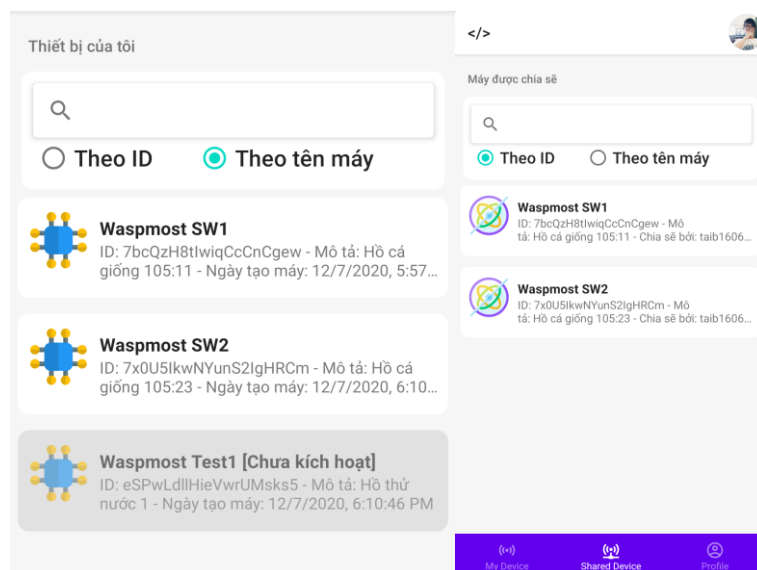
Mục đích

- Quản lý danh sách thiết bị và thực hiện các chức năng.

- Điều kiện tiên quyết: yêu cầu người dùng đăng nhập vào hệ thống.

Giao diện

Giao diện quản lý thiết bị sẽ bao gồm 2 tab: Thiết bị của tôi và Máy được chia sẻ. Mỗi tab sẽ bao gồm danh sách thiết bị cùng những chức năng bên trong.



Hình 3.17 Quản lý người dùng phía client

3.4.2.4. Chức năng tìm kiếm thiết bị

Mục đích

- Tìm kiếm nhanh thiết bị trong danh sách
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống

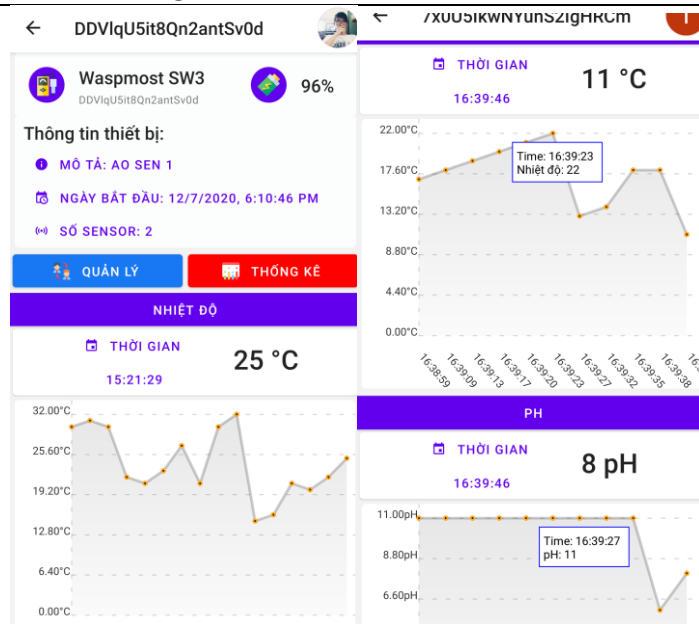
3.4.2.5. Chức năng truy xuất dữ liệu theo thời gian thực

Mục đích

- Truy xuất thông tin về thiết bị, dữ liệu trên thiết bị cập nhật liên tục theo thời gian thực.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống.

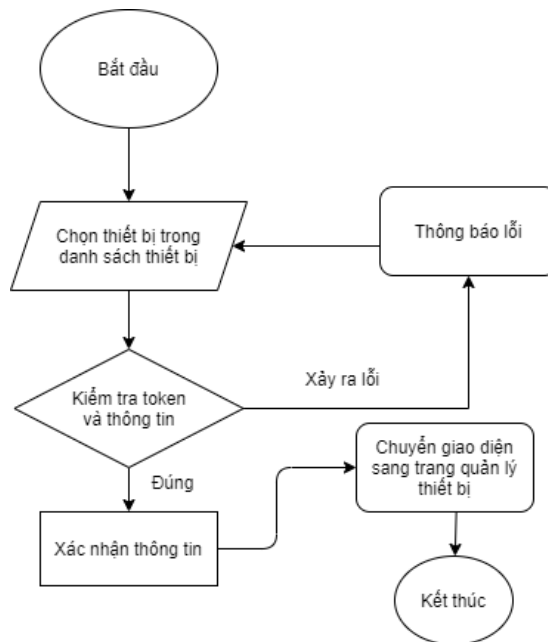
Giao diện

Giao diện bao gồm thông tin về thiết bị ID, tên thiết bị, mô tả thiết bị... cùng với các nút chọn chức năng phía trong. Cùng các biểu đồ dữ liệu bên trong thiết bị được cập nhật realtime.



Hình 3.18 Truy xuất dữ liệu theo thời gian thực

Sơ đồ xử lý



Hình 3.19 Lưu đồ truy xuất dữ liệu realtime

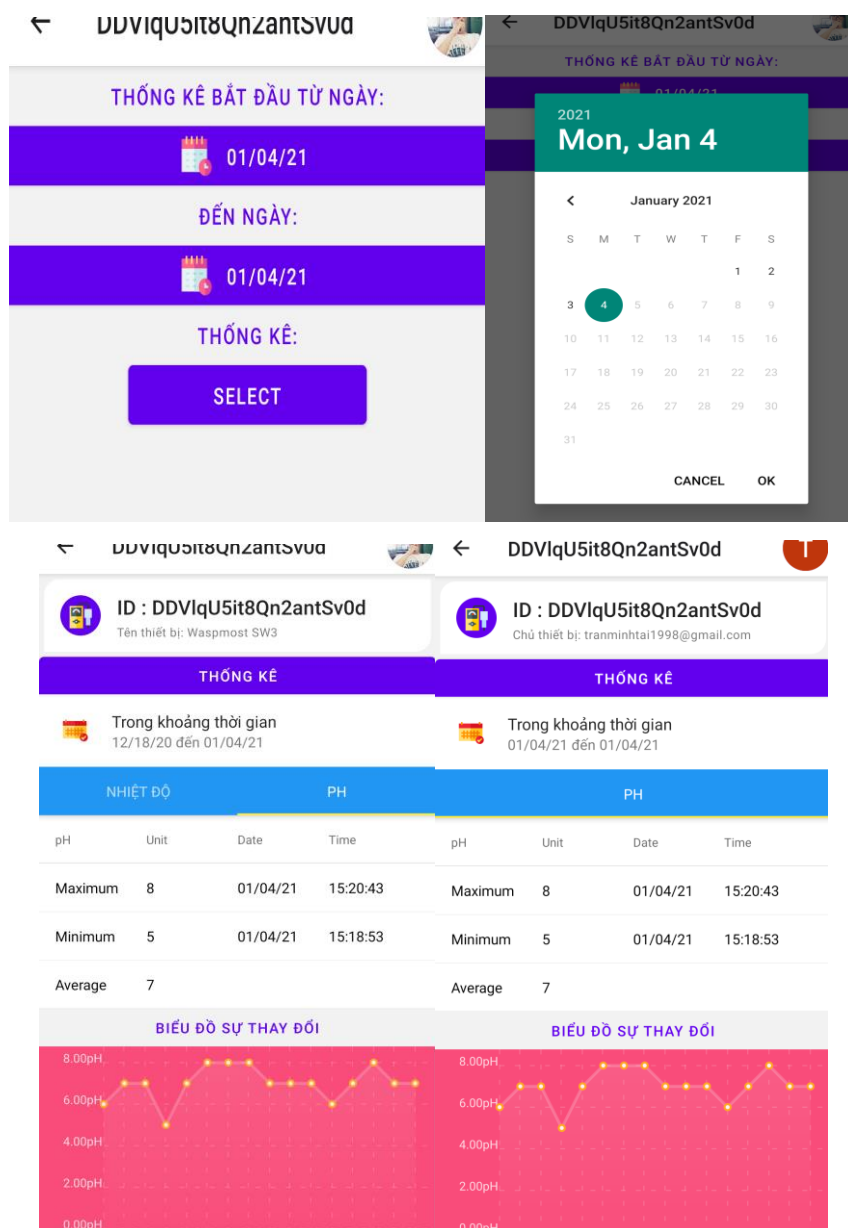
3.4.2.6. Chức năng truy xuất dữ liệu theo khoảng thời gian

Mục đích

- Truy xuất được dữ liệu của thiết bị trong một khoảng thời gian nhất định đưa ra được các thông số trong khoảng thời gian đó.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống và chọn một thiết bị thực hiện chức năng truy xuất.

Giao diện

Ở bước đầu tiên giao diện sẽ bao gồm 2 trường chọn thời gian và một nút nhấn để hoàn thành. Sau đó, giao diện bên trong sẽ bao gồm các thông số của các cảm biến cùng với các thông tin được tính toán.



Hình 3.20 Truy xuất theo khoản thời gian

3.4.2.7. Chức năng xuất dữ liệu ra file CSV

Mục đích

- Sao lưu dữ liệu trong khoảng thời gian được chọn để phục vụ cho những công việc khác.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống, trong giao diện truy xuất dữ liệu theo khoảng thời gian.

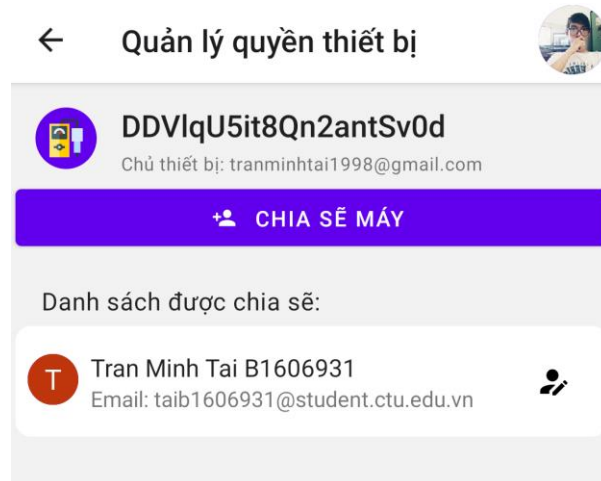
3.4.2.8. Chức năng quản lý người dùng được chia sẻ

Mục đích

- Quản lý được quyền bên trong thiết bị.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống, người dùng là chủ thiết bị, người dùng đang ở giao diện chi tiết một thiết bị.

Giao diện

Quản lý thiết bị sẽ bao gồm danh sách thiết bị cùng với các nút nhấn chức năng bên trong.



Hình 3.21 Quản lý người dùng được chia sẻ

3.4.2.9. Chức năng cấp quyền cho người dùng khác trong mạng

Mục đích

- Cấp quyền truy cập vào thiết bị cho người dùng khác trong mạng với các cảm biến cho phép.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống, người dùng là chủ thiết bị, người dùng đang ở giao diện quản lý chia sẻ thiết bị.

Giao diện

Giao diện sẽ bao gồm một text-field để nhập email của người được chia sẻ cùng các switch Tắt/Mở các cảm biến chia sẻ.

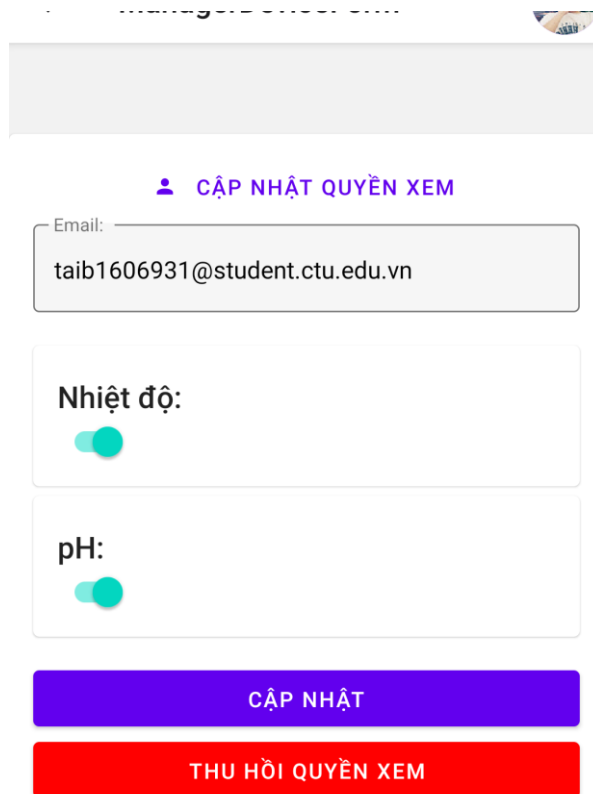
3.4.2.10. Chức năng cập nhật quyền cho người dùng khác trong mạng

Mục đích

- Cập nhật lại quyền truy cập vào thiết bị cho người dùng khác trong mạng với các cảm biến cho phép.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống, người dùng là chủ thiết bị, người dùng đang ở giao diện quản lý chia sẻ thiết bị.

Giao diện

Giao diện tương tự như thêm quyền truy xuất. Trường text-field sẽ bị vô hiệu hóa cùng nút nhấn thu hồi quyền xem



Hình 3.24 Cập nhật quyền truy xuất thiết bị

3.4.2.11. Chức năng thu hồi quyền của người dùng khác trong mạng

Mục đích

- Thu hồi lại quyền truy cập vào thiết bị đã cấp cho người dùng khác trong mạng.
- Điều kiện tiên quyết: yêu cầu người dùng đã đăng nhập vào hệ thống, người dùng là chủ thiết bị, người dùng đang ở giao diện quản lý chia sẻ thiết bị.

CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ

4.1. GIỚI THIỆU

4.1.1. Mục tiêu

- Xác định các trường hợp lỗi có thể xảy ra.
- Kiểm thử về giao diện có đáp ứng thiết kế đặt ra hay không.
- Nhằm xác định hệ thống có đáp ứng được nhu cầu đặt ra hay không.
- Phát hiện lỗi của chương trình để tiến hành bảo trì cần thiết.
- Tạo tài liệu cho phát triển phần mềm và bảo trì phần mềm sau này.

4.1.2. Phạm vi kiểm thử

- Kiểm tra thiết kế và hệ thống thỏa mãn đặc tả.
- Kiểm tra chức năng xử lý có đúng dữ liệu hay không.
- Tìm và sửa các lỗi xảy ra khi kiểm thử.

4.2. CHI TIẾT KẾ HOẠCH KIỂM THỬ

4.2.1. Các chức năng được kiểm thử

- Thêm thiết bị.
- Thêm quyền truy cập thiết bị.
- Truy xuất theo khoảng thời gian.
- Kích hoạt thiết bị.

4.2.2. Cách tiếp cận

- Thực hiện các kiểm thử đơn vị.
- Kiểm thử tích hợp.
- Kiểm thử chức năng.
- Kiểm thử chấp nhận.

4.2.3. Môi trường kiểm thử

Trên trình duyệt:

- Tốc độ CPU: 2.6GHz.
- Dung lượng bộ nhớ RAM: 8GB.
- Hệ điều hành: Ubuntu 20.04.
- Trình duyệt: Chrome.

Trên thiết bị di động

- Tốc độ CPU: 1.2GHz
- Dung lượng bộ nhớ RAM: 3GB.

- Hệ điều hành: Android 9.

4.3. CÁC TRƯỜNG HỢP KIỂM THỬ

4.3.1. Trường hợp kiểm thử 1: Thêm thiết bị

Mục tiêu

- Kiểm tra hoạt động có tốt hay không.

Trường hợp và kết quả

Mã trường hợp	Nội dung được nhập vào	Kết quả mong đợi	Kết quả thực tế	Ghi chú
TH1_01	Không nhập gì cả	Hiển thị thông báo và yêu cầu nhập vào	Hiển thị thông báo và yêu cầu nhập vào	Thành công
TH1_02	Nhập sai định dạng	Hiển thị thông báo và yêu cầu nhập lại	Hiển thị thông báo và yêu cầu nhập lại	Thành công
TH1_03	Đánh dấu vào cảnh báo và nhập đầy đủ	Thông báo thêm thành công	Thông báo thêm thành công	Thành công
TH1_04	Nhập đầy đủ	Thông báo thêm thành công	Thông báo thêm thành công	Thành công

Bảng 4.1 Kiểm thử thêm thiết bị

4.3.2. Trường hợp kiểm thử 2: Thêm quyền truy cập thiết bị

Mục tiêu

- Kiểm tra hoạt động có tốt hay không.

Trường hợp và kết quả

Mã trường hợp	Nội dung được nhập vào	Kết quả mong đợi	Kết quả thực tế	Ghi chú
TH2_01	Không nhập người dùng	Hiển thị thông báo và yêu cầu nhập vào	Hiển thị thông báo và yêu cầu nhập vào	Thành công
TH2_02	Nhập người dùng không tồn tại hoặc đã	Hiển thị thông báo lỗi và yêu	Hiển thị thông báo lỗi và yêu	Thành công

	được thêm từ trước	cầu nhập lại	cầu nhập lại	
TH2_03	Nhập chính xác	Thông báo thêm thành công	Thông báo thêm thành công	Thành công

Bảng 4.2 Kiểm thử thêm quyền truy cập

4.3.3. Trường hợp kiểm thử 3: Cập nhật quyền truy cập thiết bị

Mục tiêu

- Kiểm tra hoạt động có tốt hay không.

Trường hợp và kết quả

Mã trường hợp	Nội dung được nhập vào	Kết quả mong đợi	Kết quả thực tế	Ghi chú
TH3_01	Không chọn bất kỳ cảm biến nào	Thông báo lỗi	Thông báo yêu cầu chọn ít nhất 1 cảm biến	Thành công
TH3_02	Chọn đầy đủ	Thông báo thành công	Thông báo cập nhật quyền thành công	Thành công

Bảng 4.3 Kiểm thử thêm quyền truy cập thiết bị

4.3.4. Trường hợp kiểm thử 4: Kích hoạt thiết bị

Mục tiêu

- Kiểm tra hoạt động có tốt và ổn định hay không

Trường hợp và kết quả

Mã trường hợp	Nội dung được nhập vào	Kết quả mong đợi	Kết quả thực tế	Ghi chú
TH4_01	Không nhập ID thiết bị	Yêu cầu nhập vào ID	Không thể nhấn submit	Thành công
TH4_02	Nhập sai ID thiết bị	Thông báo lỗi	Thông báo kích hoạt thất bại	Thành công
TH4_03	Nhập chính xác ID thiết bị	Thông báo thành công	Thông báo thành công cập nhật lại trạng thái thiết bị	Thành công

Bảng 4.4 Kiểm thử thêm quyền truy cập thiết bị

4.4. KẾT QUẢ KIỂM THỬ

STT	Tên chức năng	Số trường hợp kiểm thử	Số trường hợp thành công	Số trường hợp thất bại	Ghi chú
1	Thêm thiết bị	4	4	0	
2	Cập nhật quyền truy xuất thiết bị	3	3	0	
3	Cập nhật quyền truy cập thiết bị	2	2	0	
4	Kích hoạt thiết bị	3	3	0	

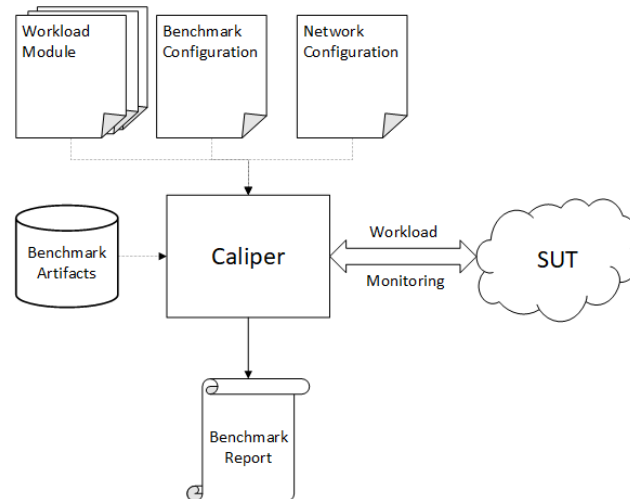
Bảng 4.5 Kết quả kiểm thử

4.5. ĐÁNH GIÁ HIỆU NĂNG MẠNG BLOCKCHAIN

Hyperledger Caliper là một framework dùng để đánh giá hiệu suất mạng blockchain, cho phép người dùng thử nghiệm các giải pháp blockchain khác nhau với các trường hợp sử dụng tùy chỉnh và nhận được một tập hợp các kết quả kiểm tra hiệu suất.

Caliper tạo ra một khối lượng công việc dựa trên một hệ thống cụ thể đang được thử nghiệm và liên tục theo dõi phản hồi của nó. Cuối cùng, nó sẽ tạo ra một báo cáo dựa trên các phản hồi đã quan sát được.

Mô hình hoạt động của Caliper



Tốc độ mà các giao dịch được nhập vào hệ thống blockchain là một yếu tố trong các bài kiểm tra hiệu suất. Caliper cho phép tùy chỉnh các thông số kỹ thuật của bộ điều khiển để người dùng thử nghiệm theo cơ chế tùy chỉnh:

Fixed Rate: là bộ điều khiển hỗ trợ gửi số giao dịch ở thời gian cố định được chỉ định từ trước với đơn vị là TPS (số giao dịch mỗi giây).

Fixed Load: bộ điều khiển tự động điều chỉnh hỗ trợ phù hợp với cấu hình thiết bị. Giúp quá trình đánh giá không quá tải, không đánh giá chính xác được khả năng chịu tải tối đa của mạng blockchain.

Sử dụng Hyperledger Caliper để thử nghiệm đánh giá với các trường hợp:

Cấu hình máy chủ khởi chạy mạng:

- CPU: I7 6700HQ
- RAM: 16GB
- Môi trường docker phiên bản 20.10.1

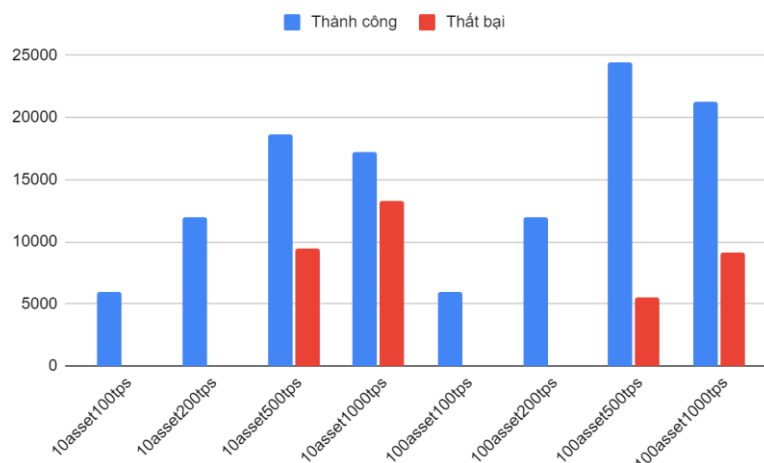
Các thiết lập chạy thử nghiệm:

- Số worker sử dụng: 8.
- Thời gian: 60 giây.
- Bộ điều khiển tỉ lệ: fixed-rate
- + 100 giao dịch/giây

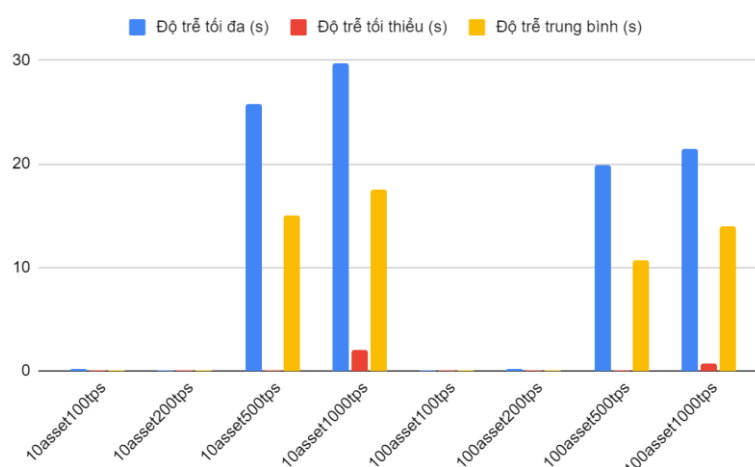
- + 200 giao dịch/giây
- + 500 giao dịch/giây
- + 1000 giao dịch/giây

Trường hợp	Thành công	Thất bại	Lưu lượng gửi (TPS)	Độ trễ cao nhất (s)	Độ trễ thấp nhất (s)	Độ trễ trung bình (s)	Thông lượng (TPS)
10asset100tps	6008	0	100.1	0.14	0.01	0.04	100.0
10asset200tps	12008	0	200.1	0.16	0.01	0.02	200.0
10asset500tps	24417	5518	498.7	19.92	0.03	10.73	466.9
10asset1000tps	21276	9096	506.1	21.51	0.77	13.97	472.2
100asset100tps	6008	0	100.1	0.17	0.01	0.04	100.1
100asset200tps	12008	0	200.1	0.11	0.01	0.01	200.0
100asset500tps	18616	9493	468.1	25.84	0.07	15.05	431.7
100asset1000tps	17205	13247	507.4	29.79	2.08	17.53	470.6

Bảng 4.6 Đánh giá mạng blockchain



Hình 4.1 Biểu đồ giao dịch thành công/thất bại ở các mức



Hình 4.2 Biểu đồ độ trễ giao dịch

- Trong 4 trường hợp thử nghiệm ở mức dưới 200 giao dịch/giây, độ trễ cao nhất chỉ ở khoảng 0.17 giây. Tất cả các giao dịch thực hiện đều thành công.
- Với trường hợp 10asset500tps cùng mức 500 giao dịch/giây, độ trễ cao nhất ở mức 19.92 giây, độ trễ trung bình khoảng 10.73. Số giao dịch thành công là 24417 chiếm tỉ lệ khoảng 81.5% tổng số giao dịch.
- Trường hợp 10asset1000tps với mức thiết lập 1000 giao dịch/giây, độ trễ cao nhất ở mức 21.51 giây, độ trễ trung bình ở mức 13.97 giây. Thông lượng gửi đạt 506.1 so với 1000 giao dịch/giây ở mức thiết lập. Số giao dịch thành công là 21276 chiếm 70% tổng số giao dịch.
- Trong trường hợp 100asset500tps thử nghiệm ở mức 500 giao dịch/giây, các giao dịch bắt đầu xuất hiện độ trễ cao nhất ở mức 25.84 giây, độ trễ trung bình nằm ở mức 15.05 giây. Lưu lượng gửi đạt 468.1 so với mức thiết lập 500 giao dịch/giây. Trường hợp này, số giao dịch thất bại là 9493 chiếm khoảng 33.7% tổng giao dịch.
- Trường hợp 100asset1000tps thử nghiệm ở mức 1000 giao dịch/giây, độ trễ xuất hiện ở mức cao 29.79, độ trễ trung bình cũng nằm ở mức cao 17.53. Lưu

lượng chỉ đạt ở mức 507 so với 1000 ở mức thiết lập. Trường hợp này số giao dịch thất bại là 13247 chiếm đến 43% tổng giao dịch.

Thử nghiệm với bộ điều khiển Fixed Load cùng với các thông số tương tự

Trường hợp	Thành công	Thất bại	Lưu lượng gửi (TPS)	Độ trễ cao nhất (s)	Độ trễ thấp nhất (s)	Độ trễ trung bình (s)	Thông lượng (TPS)
fixedLoad	15067	0	251.1	0.17	0.01	0.04	249.9

Bảng 4.6.7 Thử nghiệm với bộ điều khiển Fixed Load

- Caliper tự động điều chỉnh lưu lượng gửi ở mức 251.1, độ trễ ở mức thấp. Điều này cho thấy được khả năng chịu tải có thể cao hơn mức đề xuất của bộ điều khiển.

Tiếp tục thử nghiệm với Fixed Rate

Trường hợp	Thành công	Thất bại	Lưu lượng gửi (TPS)	Độ trễ cao nhất (s)	Độ trễ thấp nhất (s)	Độ trễ trung bình (s)	Thông lượng (TPS)
total	21248	0	354.1	3.21	0.01	0.29	354.0

Bảng 4.8 Thử nghiệm ở mức 354 giao dịch trên giây

- Dựa vào kết quả 70% các giao dịch thành công ở mức thiết lập 1000 giao dịch/giây với thông lượng gửi khoảng 506.1 giao dịch/giây, thiết lập ở mức 354 (70% thành công của 506.1) giao dịch/giây để thử nghiệm. Trong trường hợp này, độ trễ đã ở mức ổn định. Tất cả các giao dịch đều thành công cho ta thấy được khả năng chịu tải ở mức 354 giao dịch/giây là chấp nhận được.

Container Docker	CPU% cao nhất	CPU% trung bình	RAM cao nhất	RAM trung bình
dev-peer0.org1.example.com-basic_2.0	80.73	12.04	85.9	71.2
dev-peer1.org1.example.com-basic_2.0	72.97	11.65	82.2	69.8
peer1.org1.example.com	155.55	27.16	382	375
peer0.org1.example.com	137.25	27.18	373	366

Bảng 4.9 Tài nguyên sử dụng

Thử nghiệm ở nhiều vòng cho thấy được khả năng chịu tải của mạng Blockchain, giúp đánh giá và cải thiện, nâng cấp hệ thống khi cần thiết.

PHẦN KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC

1.1. Về lý thuyết

- Nắm được quy trình xây dựng phát triển ứng dụng.
- Hiểu được kiến thức về Blockchain, Hyperledger Fabric.
- Ứng dụng được công nghệ Blockchain vào hệ thống.
- Nâng cao khả năng phân tích, thiết kế...
- Bổ sung nhiều kiến thức về ReactJS, React Native, NodeJS.
- Vận dụng được các kỹ thuật và kiến thức học được áp dụng vào thực tiễn về lập trình, tư duy logic và kỹ năng làm việc nhóm.

1.2. Về chương trình

- Bước đầu xây dựng được nền tảng quan trắc.
- Đáp ứng được những yêu cầu cơ bản của.
- Xây dựng, cài đặt và triển khai thành công mạng blockchain Hyperledger Fabric ở mức độ cơ bản.
- Giao diện tương đối dễ sử dụng

2. HẠN CHẾ

- Công nghệ mới, nguồn tài liệu còn hạn chế.
- Các chức năng chưa thể hoàn thiện được tốt.
- Chưa hiểu rõ hết toàn bộ đề tài, nhiều chức năng chưa đúng yêu cầu.
- Hệ thống còn đơn giản, chưa tối ưu.

3. HƯỚNG PHÁT TRIỂN

- Nghiên cứu thêm về mạng blockchain Hyperledger Fabric để ứng dụng được hết tính năng của nó, đồng thời cải thiện lại những vấn đề chưa giải quyết được ở lúc này, cải tiến chaincode tối ưu hơn.
- Phát triển thêm các chức năng mới, mở rộng hệ thống mạng blockchain.
- Cải thiện lại giao diện người dùng trở nên tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] Hyperledger Fabric 2.2: <https://hyperledger-fabric.readthedocs.io/en/release-2.2>
- [2] NodeJS: <https://nodejs.org/en/docs>
- [3] React JS: <https://reactjs.org>
- [4] React Native: <https://reactnative.dev>
- [5] Hyperledger Fabric SDK for Node.js: <https://hyperledger.github.io/fabric-sdk-node/release-2.2/module-fabric-network.html>
- [6] Deepstream.io: <https://deepstream.io/docs>
- [7] Tài liệu về Firebase: <https://firebase.google.com/docs>
- [8] Sinh viên Đặng Xuân Mai, Luận văn tốt nghiệp đại học ngành Công Nghệ Thông Tin *Xây dựng hệ thống quản lý văn bản trên nền tảng Blockchain*, Khoa CNTT & TT – Đại học Cần Thơ, 2019

PHỤ LỤC

1. HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

1.1. Cài đặt Hyperledger Fabric

Yêu cầu:

- Hệ điều hành: Ubuntu 20.04
- Docker: phiên bản 20.10.1
- Docker-compose: phiên bản 1.27.4
- Node: phiên bản 15.x
- Npm: phiên bản 7.x
- Git: phiên bản 2.25.1

Bước 1: Cài đặt các phần mềm tiên quyết.

Mở terminal để thực hiện các lệnh sau:

```
sudo apt install curl
sudo apt install git
sudo apt install docker
sudo apt install docker-compose
sudo apt install nodejs & npm install -g npm@latest
```

Bước 2: Cài đặt các tệp nhị phân và docker image của Hyperledger Fabric

Mở terminal thực hiện các lệnh sau:

```
curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.2.0 1.4
```

1.2. Cài đặt ứng dụng

Bước 1: Tải về source của ứng dụng tại github:

```
git clone https://github.com/nhuthuy-2972/iotbaseblockchain.git
```

Bước 2: Khởi động mạng Blockchain. Điều hướng đến thư mục network/iot-network, thực hiện lệnh sau

```
./iot-network-start.sh
```

Nếu bị từ chối chạy lệnh sau: `chmod a+x iot-network-start.sh`

Bước 3: Khởi động server API. Điều hướng vào thư mục Server-API thực hiện các lệnh sau:

```
npm install
./start-api-server.sh
```

Bước 4: Khởi động web server:

- Điều hướng tới thư mục webui/web-client, thực hiện các lệnh sau:

```
yarn install  
./start-webclient.sh
```

- Điều hướng tới thư mục webui/web-client, thực hiện các lệnh sau:

```
npm install  
./start-webadmin.sh
```

Bước 5: Sử dụng:

Truy cập vào <http://localhost:3000> để truy cập vào giao diện người dùng.

Truy cập vào <http://localhost:3001> để truy cập vào giao diện quản trị.