

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**LÝ TUẤN AN – 52000620
PHÙNG PHÚC HẬU – 52000443
TRẦN ĐĂNG QUANG – 52000118
LÊ NGỌC NGUYỄN MINH – 52000498**

**ỨNG DỤNG BLOCKCHAIN
TRONG TRUY XUẤT NGUỒN
GỐC SẢN PHẨM**

**BÁO CÁO CUỐI KỲ
BLOCKCHAIN
VÀ CÔNG NGHỆ SỔ CÁI PHÂN TÁN**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**LÝ TUẤN AN – 52000620
PHÙNG PHÚC HẬU – 52000443
TRẦN ĐĂNG QUANG – 52000118
LÊ NGỌC NGUYỄN MINH – 52000498**

ỨNG DỤNG BLOCKCHAIN TRONG TRUY XUẤT NGUỒN GỐC SẢN PHẨM

BÁO CÁO CUỐI KÌ BLOCKCHAIN VÀ CÔNG NGHỆ SỞ CẢI PHÂN TÁN

Người hướng dẫn
TS. Phạm Thái Kỳ Trung

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Thầy Phạm Thái Kỳ Trung đã hỗ trợ cho chúng em trong suốt quá trình học tập môn Blockchain. Nhờ có những kiến thức của thầy, chúng em mới có đủ kiến thức để hoàn thành bài báo cáo này.

Chúng em xin chân thành cảm ơn Thầy.

TP. Hồ Chí Minh, ngày 29 tháng 12 năm 2023

Tác giả

An

Lý Tuấn An

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Phạm Thái Kỳ Trung. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 29 tháng 12 năm 2023

Tác giả

An

Lý Tuấn An

TRUY XUẤT NGUỒN GỐC SẢN PHẨM ỨNG DỤNG BLOCKCHAIN

TÓM TẮT

Bài báo cáo này trình bày ứng dụng công nghệ blockchain trong việc xây dựng hệ thống truy xuất nguồn gốc của sản phẩm.

Báo cáo sẽ giới thiệu cơ bản về công nghệ blockchain và tiềm năng ứng dụng trong lĩnh vực truy xuất thông tin. Sau đó, báo cáo trình bày phân tích yêu cầu, thiết kế kiến trúc hệ thống và cách lưu trữ thông tin nguồn gốc sản phẩm dựa trên Hyperledger Fabric.

Để minh họa, báo cáo sẽ hướng dẫn xây dựng demo với các chức năng: nhập thông tin sản phẩm, tìm kiếm và tra cứu thông tin trên blockchain.

Báo cáo nhằm mục đích ứng dụng công nghệ blockchain vào bài toán thực tế, góp phần nâng cao tính minh bạch và an toàn cho người tiêu dùng.

MỤC LỤC

DANH MỤC HÌNH VẼ	vii
DANH MỤC BẢNG BIỂU	viii
DANH MỤC CÁC CHỮ VIẾT TẮT.....	ix
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Các nội dung tìm hiểu	1
1.2 Mục tiêu thực hiện đề tài.....	1
1.3 Công cụ sử dụng.....	1
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	2
2.1 Giới thiệu về Blockchain.....	2
2.1.1 Khái niệm Blockchain.....	2
2.1.2 Các phiên bản của công nghệ Blockchain.....	2
2.1.3 Cấu trúc của Blockchain.....	3
2.1.4 Phân loại hệ thống Blockchain.....	3
2.1.5 Cơ chế hoạt động của Blockchain	5
2.1.6 Ưu điểm của Blockchain	7
2.1.7 Nhược điểm của Blockchain	8
2.1.8 Các ứng dụng và lĩnh vực ứng dụng phổ biến của Blockchain	8
2.2 Giới thiệu về Hyperledger Fabric	10
2.2.1 Định nghĩa.....	10
2.2.2 Ưu điểm.....	10
2.2.3 Nhược điểm.....	10

2.2.4 Đặc điểm của Hyperledger Fabric	11
2.2.5 Kiến trúc đơn giản của một mạng Hyperledger Fabric	11
2.2.6 Quy trình giao dịch Hyperledger Fabric	12
2.2.7 Ứng dụng Hyperledger Fabric	13
2.2.8 Triển vọng và xu hướng phát triển Hyperledger Fabric	13
2.2.9 So sánh công nghệ Blockchain giữa Hyperledger Fabric, Ethereum và Corda	14
CHƯƠNG 3. THIẾT KẾ MÔ HÌNH HỆ THỐNG.....	17
3.1 Mô hình	17
3.2 Các chức năng	17
3.3 Đặc tả Usecase:	18
3.3.1 Consumer	18
3.3.2 Farmer	19
3.3.3 Company	23
CHƯƠNG 4. QUÁ TRÌNH THỰC HIỆN DEMO.....	28
4.1 Lời nói đầu	28
4.2 Điều kiện tiên quyết	28
4.2.1 WSL2	28
4.2.2 Docker	29
4.2.3 Microsoft VS Code (Optional)	32
4.2.4 Git For Windows (Optional)	32
4.2.5 Npm & Node	32
4.3 Cài đặt ứng dụng Web.....	33

4.4 Video chi tiết quá trình thực hiện.....	34
CHƯƠNG 5. KẾT LUẬN.....	35
5.1 Tổng kết quá trình triển khai.....	35
5.2 Kết quả đạt được	35
5.3 Đánh giá ý nghĩa, giá trị khoa học, thực tiễn của hệ thống.....	35
TÀI LIỆU THAM KHẢO	36

DANH MỤC HÌNH VẼ

Hình 2.1 Cấu trúc Blockchain.....	3
Hình 2.2 Public Blockchain	4
Hình 2.3 Private Blockchain	4
Hình 2.4 Hybrid Blockchain	5
Hình 2.5 Hyperledger Fabric.....	11
Hình 3.1 Mô hình hệ thống	17
Hình 3.2 Usecase Diagram.....	18

DANH MỤC BẢNG BIỂU

No table of figures entries found.

DANH MỤC CÁC CHỮ VIẾT TẮT

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Các nội dung tìm hiểu

Báo cáo này tập trung vào nghiên cứu các khía cạnh quan trọng của Blockchain để ứng dụng trong truy xuất nguồn gốc sản phẩm. Các nội dung tìm hiểu bao gồm:

- Tìm hiểu cơ bản về công nghệ blockchain: định nghĩa, đặc điểm, cấu trúc, ...
- Nghiên cứu các công nghệ blockchain phổ biến hiện nay.
- Tìm hiểu cách thiết kế và xây dựng một đề án trên nền tảng Hyperledger Fabric.
- Tìm hiểu các ứng dụng/case study về xây dựng hệ thống truy xuất nguồn gốc sản phẩm.

1.2 Mục tiêu thực hiện đề tài

- Trang bị kiến thức cơ bản về công nghệ blockchain và đặc điểm ưu việt của nó.
- Xây dựng được demo/ứng dụng trên nền tảng Hyperledger Fabric để minh họa tính năng truy xuất nguồn gốc.
- Rèn luyện khả năng phân tích và giải quyết bài toán thực tế bằng công nghệ blockchain.
- Chuẩn bị kiến thức để áp dụng blockchain vào thực tiễn sau này.

1.3 Công cụ sử dụng

- Hyperledger Fabric Framework.
- NodeJs.
- IDE Visual Code.
- Docker
- WSL2

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Blockchain

2.1.1 Khái niệm Blockchain

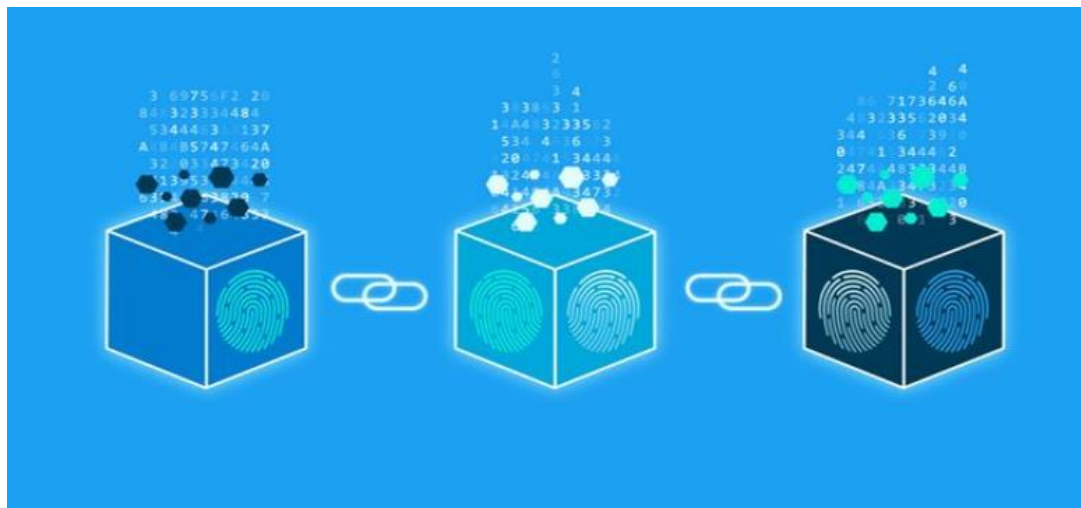
- Blockchain là thuật ngữ mô tả một trong những công nghệ chuỗi khối.
- Là một sổ cái kỹ thuật số có nhiệm vụ phân tán, lưu trữ tất cả các dữ liệu 1 cách nhanh chóng và hiệu quả.
- Blockchain có tính chất phi tập trung.

2.1.2 Các phiên bản của công nghệ Blockchain

- Công nghệ Blockchain 1.0: Tiền tệ và Thanh toán.
 - + Tiền điện tử là ứng dụng chính.
 - + Các tính năng: chuyển đổi tiền tệ, chuyển tiền và tạo hệ thống thanh toán kỹ thuật số.
- Công nghệ Blockchain 2.0: Tài chính và Thị trường.
 - + Phát triển các ứng dụng xử lý tài chính và ngân hàng.
 - + Các tính năng: Mở rộng quy mô chuỗi khối, sử dụng các ứng dụng tài chính và thị trường.
 - + Các loại tài sản: cổ phiếu, nợ phải trả, quyền sở hữu và toàn bộ những yếu tố liên quan đến hợp đồng.
- Công nghệ Blockchain 3.0: Theo dõi Thiết kế và Vận hành.
 - + Vượt ra khỏi ranh giới tài chính, phá vỡ vùng an toàn ban đầu.
 - + Mở rộng ở những lĩnh vực khác như giáo dục, chính phủ, y tế và nghệ thuật.
- Công nghệ Blockchain 4.0: Doanh nghiệp.
 - + Phát triển tập trung vào các doanh nghiệp.
 - + Tạo ra và chạy các ứng dụng giao dịch một cách nhanh chóng, an toàn và hiệu quả hơn.

- + Các tính năng: Lưu trữ thông tin, phù hợp với các ứng dụng trong việc thanh toán hóa đơn hay vấn đề tiền lương của các doanh nghiệp.

2.1.3 Cấu trúc của Blockchain



Hình 2.1 Cấu trúc Blockchain

- Mỗi Block sẽ có 3 phần:

- + Dữ liệu: Đây là thông tin về bản ghi đã xác thực, mã hóa trong khối. Hệ thống liên tục update các dữ liệu đó và bảo vệ bởi những thuật toán đồng thuận.
- + Hash của khối hiện tại: Chuỗi ký tự khác biệt tạo ra ngẫu nhiên. Chúng đại diện cho mỗi một Block. Mã hàm này có nhiệm vụ là phát hiện đổi thay trong khối của mỗi chuỗi.
- + Hash khối liền kề: Thông tin giúp các Block liên kết với nhau trong chuỗi.

2.1.4 Phân loại hệ thống Blockchain

- Công khai:

Public Blockchain

All Are Validators



Hình 2.2 Public Blockchain

- + Tất cả người dùng đều được phép đọc và ghi dữ liệu trên hệ thống Blockchain.
 - + Quá trình xác thực giao dịch yêu cầu có sự xuất hiện của nhiều nút.
 - + Quá trình tấn công sẽ không khả thi và tốn kém.
 - + Ví dụ: Bitcoin, Ethereum là Blockchain công khai.
- Riêng tư:



Hình 2.3 Private Blockchain

- + Xuất hiện bên thứ 3 đáng tin cậy và giữ vai trò lưu trữ dữ liệu.

- + Người sử dụng chỉ có thể đọc mà không có quyền ghi.
- + Chỉ cho phép một số thiết bị tham gia vào xác thực giao dịch.
- + Tối ưu hóa thời gian hoàn thành giao dịch hơn.
- + Ví dụ: Ripper là một dạng Blockchain riêng tư.

-Quyền Permissioned:



Hình 2.4 Hybrid Blockchain

- + Được thiết lập dưới dạng riêng tư.
- + Được kết hợp giữa công cộng và tư nhân.
- + Ví dụ: Các ngân hàng và tổ chức tài chính liên doanh sử dụng Blockchain của riêng mình.

2.1.5 Cơ chế hoạt động của Blockchain

- Blockchain hoạt động trên nguyên lý sau:

- + Nguyên lý mã hoá:
 - Để thực hiện giao dịch ở Blockchain, chúng ta cần phần mềm cho phép lưu trữ, trao đổi điện tử. Phần mềm đó chính là Ví điện tử. Loại ví này được bảo vệ bởi nguyên lý mã hóa đặc biệt là dùng cặp khóa bảo mật công khai (Public key) và riêng tư (Private key).

- Khi mã hóa thông điệp bằng Public key: Chủ sở hữu khóa riêng tư - cặp của khóa công khai này mới giải mã, đọc được nội dung trong thông điệp.
- Khi mã hóa bởi Private key: Lúc này bạn đã tạo chữ ký điện tử để máy tính trong Blockchain tiến hành kiểm tra chủ thể cũng như tính xác thực giao dịch. Nếu một ký tự đơn ở giao dịch này hay đổi, chữ ký điện tử cũng thay đổi theo. Chính vì vậy các Hacker khó lòng sửa được số Bitcoin muốn gửi hay giao dịch.

+ Quy tắc của sổ cái:

- Mỗi một node trong Blockchain đều được lưu giữ bản sao sổ cái kế toán. Chính vì vậy những node này đều biết tài khoản của bạn có số dư là bao nhiêu. Tuy nhiên, Blockchain không theo dõi số dư tài khoản, chỉ ghi lịch sử của những giao dịch được yêu cầu.
- Để biết trên ví điện tử có số dư bao nhiêu, bạn cần xác nhận mọi giao dịch liên quan đến ví đã diễn ra trên Blockchain.

+ Nguyên lý tạo khối:

- Những giao dịch khi đã gửi lên Blockchain đều được phân vào các khối. Giao dịch trong khối đều cùng xảy ra ở một thời điểm. Ngược lại, giao dịch chưa được xác nhận khi không thực hành trong một khối.
- Mỗi nút chúng ta có thể nhóm giao dịch cùng nhau vào khối và gửi đến mạng lưới như ản ý cho các khối kế tiếp được phép gắn vào sau đó. Bất cứ nút nào cũng tạo được khối mới.
- Để có thể thêm Blockchain, mỗi khối cần chứa quãng mã tạo bằng hàm hash không thể đảo ngược. Cách để xử lý phương trình này đó chính là đoán số ngẫu nhiên, đồng thời kết hợp nội dung khối trước, từ đó tạo ra kết quả đúng.

+ Thuật toán bảo mật Blockchain:

- Những giao dịch trong Blockchain được bảo mật bởi những thuật toán chúng tôi chia sẻ bên trên. Khi khối chứa một tham chiếu tới Block trước đó, đây sẽ là vấn đề toán học cần giải quyết để có thể truyền Block sau đến mạng lưới.
- Chính vì thế, muốn tính toán trước các Block là điều rất khó bởi nó cần tính số lượng lớn những số ngẫu nhiên để giải quyết khối và đặt trên Blockchain. Cho nên, những giao dịch trên Blockchain ngày càng an toàn hơn.

2.1.6 Ưu điểm của Blockchain

- Không thể làm giả hoặc phá hủy chuỗi Blockchain:

- Gần như những chuỗi Blockchain đều không phá hủy được. Theo những nghiên cứu trước đó, chỉ Quantum Computer (máy tính lượng tử) mới can thiệp được và có thể giải mã chuỗi Blockchain.

- Bất biến:

- Một khi thông tin, dữ liệu đã được nhập vào Blockchain, gần như không sửa được.
- Các thông tin chỉ thêm vào được khi mọi thành viên trong hệ thống chấp thuận. Chỉ có người tạo ra dữ liệu đó mới có thể sửa đổi và được lưu trữ mãi mãi trên Blockchain.

- Bảo mật dữ liệu:

- Dữ liệu trong Blockchain được bảo mật gần như tuyệt đối. Bởi các thông tin này được lưu trữ ở những khối (Block) được liên kết bởi thuật toán phức tạp.
- Khi quy mô của chuỗi (Chain) càng lớn, việc để lộ những thông tin trong hệ thống sẽ càng khó. Từ đó loại bỏ được những tình trạng đánh cắp, sửa đổi sai lệch thông tin.

- Minh bạch:

- Bất cứ ai đều có thể theo dõi được dữ liệu ở Blockchain từ địa chỉ này đến địa chỉ khác.

- Không những vậy, công nghệ còn giúp bạn theo dõi, thống kê lịch sử trên những địa chỉ đó.
- Hợp đồng thông minh:
 - Những kỹ thuật số đều được tạo bằng đoạn code IFTTT (if-this-then-that) trong Blockchain. Chính vì thế, hợp đồng cho phép hệ thống thực thi tất cả mọi thứ, không cần tới sự tham gia của thứ ba.
 - Những điều khoản ở hợp đồng thông minh thực thi khi điều kiện trước đó được đáp ứng, không ai có thể hủy bỏ hoặc ngăn chặn nó.

2.1.7 Nhược điểm của Blockchain

- Dễ bị dòm ngó bởi Hacker: Khi Blockchain được tin tưởng, có nhiều giao dịch sẽ trở thành “con mồi béo bở” của các Hacker thông qua tấn công 51%.
- Một khi các dữ liệu được thêm vào Blockchain, việc sửa đổi chúng là rất khó.
- Ngoài ra, khi Private – key bị mất, tiền của người dùng bị mất theo và không thể lấy lại được.

2.1.8 Các ứng dụng và lĩnh vực ứng dụng phổ biến của Blockchain

- Tài chính ngân hàng:
 - Xác thực khả năng tín dụng cũng như thông tin khách hàng: Công nghệ cho phép thực hiện các giao dịch dù không có trung gian xác minh.
 - Mạng lưới tiến hành xác minh, thanh toán giao dịch ngang hàng.
 - Hạn chế và quản lý rủi ro khi thanh toán bởi vỡ nợ hay trục trặc kỹ thuật trước khi giao dịch.
 - Quản lý thông minh: Công nghệ cho phép đổi mới liên tục, cải tiến và lặp lại dựa vào sự đồng thuận ở mạng lưới.
- Thương mại điện tử:
 - Quản lý dữ liệu khách hàng.
 - Theo dõi tình trạng, thông tin về sản phẩm/hàng hóa qua serial, QR.
 - Xây dựng trên sàn thương mại điện tử hay website hệ thống thanh toán, ví điện tử, thẻ quà tặng, khách hàng thân thiết, tri ân khách hàng...

- Vận hành, đồng thời quản lý chuỗi cung ứng.

- Giáo dục:

- Quản lý đánh giá uy tín khi nghiên cứu khoa học.
- Ghi lại những dữ liệu bảo mật về điểm số, học tập cho hệ thống học trực tuyến. Cùng với đó là đánh giá năng lực cá nhân dựa vào yêu cầu tuyển sinh.
- Lưu trữ và theo dõi bảng điểm, bằng cấp của mỗi sinh viên và thông tin đơn vị đào tạo.
- Xem xét ứng viên phù hợp việc giảng dạy không để đưa ra quyết định mời họ làm việc.

- Truyền thông, viễn thông:

- Chống gian lận khi chuyển vùng: Những thỏa thuận chuyển vùng của các nhà khai thác sẽ minh bạch. Bởi nút được chỉ định trong khối đóng vai trò là người khai thác để tiến hành xác minh giao dịch phát trên mạng.
- Chuyển đổi 5G: Thỏa thuận giữa các mạng và quy tắc sẽ ở dạng hợp đồng thông minh. Tức là hệ thống sẽ tự kết nối thiết bị với nhà mạng gần nhất và tiến hành tính phí dịch vụ cũng như đánh giá tính liên tục của kết nối.
- Kết nối Internet: Tạo môi trường truyền tải mọi dữ liệu an toàn qua việc tạo mạng lưới ngang hàng tự quản.

- Sản xuất:

- Theo dõi số lượng hàng hóa nhập – bán cùng lịch trình sản xuất.
- Quản lý kho bãi, hàng tồn kho.
- Truy xuất được nguồn gốc của hàng hóa qua các khâu.
- Theo dõi nguồn nguyên liệu sản xuất.

- Y tế:

- Theo dõi, quản lý bệnh lý và tăng cường quản lý về chất lượng.
- Quản lý thiết bị y tế, chuỗi cung ứng thuốc: Theo dõi nguồn gốc, hạn dùng của vật tư Y tế.
- Tăng tính minh bạch trong xuất xứ xét nghiệm lâm sàng, tự động hóa giao dịch khám chữa bệnh, quyền sở hữu dữ liệu bệnh nhân.

- Nông nghiệp:

- Quản lý chuỗi phân phối sản phẩm tồn kho, cung ứng sản phẩm.
- Lưu trữ các thông tin về hàng hóa, tiêu chuẩn thực phẩm sạch, quy trình chăm sóc.
- Truy xuất vòng đời nông sản và nguồn gốc.

- Logistic:

- Sở hữu nhiều tính năng thông minh, việc ứng dụng Blockchain vào Logistic là điều đúng đắn. Blockchain góp phần giúp chúng ta đẩy mạnh quá trình vận chuyển nguyên vật liệu, hàng hóa. Chính vì thế, bạn sẽ tiết kiệm được chi phí nhân công, vận chuyển, phát sinh như cầu cảng, kho bãi... và thời gian.
- Bên cạnh đó việc kết hợp công nghệ này còn giúp chúng ta quản lý được lịch trình di chuyển, rà soát chuyến đi của xe hàng.
- Ngoài ra Blockchain còn được ứng dụng trong nhiều ngành khác như bất động sản, du lịch, từ thiện....

2.2 Giới thiệu về Hyperledger Fabric

2.2.1 Định nghĩa

- Hyperledger Fabric là một private/permissioned blockchain framework nguồn mở, được thiết kế để phát triển các ứng dụng blockchain cho doanh nghiệp.
- Fabric cung cấp cơ sở hạ tầng blockchain module, có khả năng mở rộng cao, hỗ trợ các mô hình triển khai linh hoạt.

2.2.2 Ưu điểm

- Hiệu năng xử lý giao dịch cao, có khả năng mở rộng tốt.
- Hỗ trợ thiết lập quyền riêng tư và kiểm soát truy cập ở mức tinh vi.
- Có cơ chế đồng thuận Modular, linh hoạt hỗ trợ nhiều mô hình triển khai.

2.2.3 Nhược điểm

- Quy trình phức tạp, khó sử dụng đối với nhà phát triển nhỏ lẻ.
- Đòi hỏi máy chủ và băng thông mạng lớn để vận hành ổn định.

2.2.4 Đặc điểm của Hyperledger Fabric

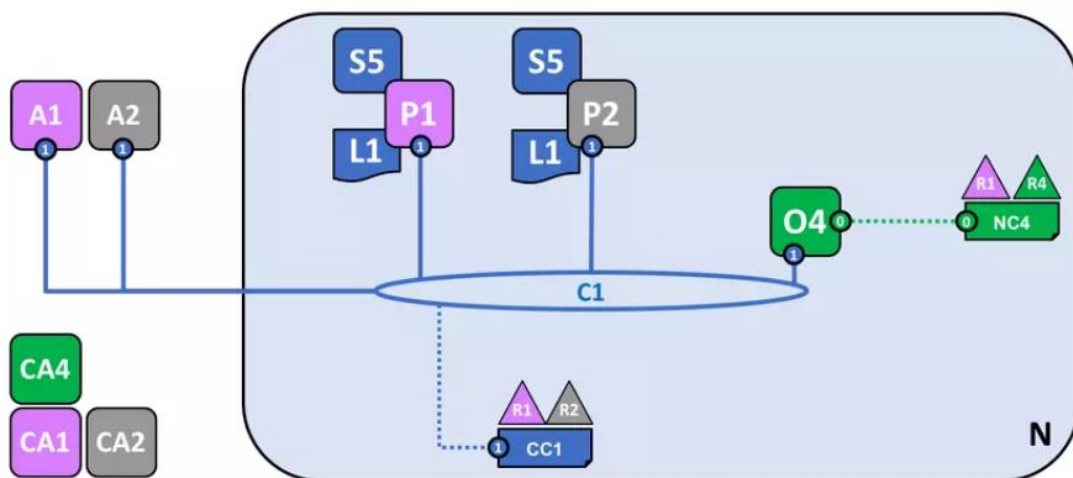
- Hyperledger Fabric được phát triển bởi Hyperledger, trong đó phải kể đến Fabcar. Fabcar có tính năng hướng dẫn thông tin cho Hyperledger Fabric. Fabcar được tạo nên từ 2 phần chính:

- Ứng dụng Chaincode: được dùng để triển khai và thực thi lệnh bên trong mạng fabric.
- Ứng dụng Client: được dùng để các traders tương tác với chaincode trên mạng fabric này.

- Hyperledger Fabric được xem là có tính mô đun tương đối cao. Hơn nữa Hyperledger Fabric hoạt động khá linh hoạt và nó còn cung cấp cho doanh nghiệp một bộ khung để xây dựng Private Blockchain. Đó là lý do tại sao các doanh nghiệp có thể dễ dàng xây dựng một ứng dụng Private Blockchain phù hợp các yêu cầu nghiệp vụ của mình.

- Tuy nhiên, Hyperledger Fabric sẽ không dễ sử dụng cho người mới. Bởi vì nó được hoạt động như một framework mô-đun, nơi mà các ứng dụng có thể dễ dàng mở rộng ở mọi cấp độ. Hyperledger Fabric hi vọng sẽ trở thành nhà cung cấp các dịch vụ blockchain minh bạch, phân cấp và bảo mật.

2.2.5 Kiến trúc đơn giản của một mạng Hyperledger Fabric



Hình 2.5 Hyperledger Fabric

N: (Network) Mạng.

NC: Network Configuration (Cấu hình của mạng).

C: Channel (Kênh), tập hợp các tổ chức có vai trò nhất định trong cùng một quy trình kinh doanh. Ví dụ, trong một channel về mua bán xe hơi sẽ gồm có 2 tổ chức là: Nhà sản xuất xe hơi, Nhà phân phối xe hơi.

CC: Channel Configuration (Cấu hình của kênh).

R: Organization (Tổ chức).

O: Orderer Node: Nếu như trong Public Blockchain, tất cả các node 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, 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 (Chaincode), được cài đặt trên kênh, định nghĩa rõ các struct, các hành động mà người dùng có thể thực hiện để tương tác trạng thái của struct được lưu trong sổ cái.

L: Ledger (Sổ cái), lưu trữ trạng thái của các đối tượng.

CA: Certificate Authority, phát hành identity cho người dùng hoặc node của tổ chức tương ứng. Ví dụ, người dùng A là thành viên của Tổ chức R1, khi muốn tham gia vào mạng thì sẽ gửi yêu cầu đến CA1, sau đó CA1 sẽ tạo ra một identity gồm private-key, public-key và các đặc tính liên quan khác, sau đó trả về cho người dùng A, từ đó về sau A dùng identity đó để thực hiện các tương tác với mạng, mạng sẽ tự động biết đó là người dùng A đến từ tổ chức R1.

A: Application, ứng dụng hay giao diện (web, mobile app) giúp người dùng tương tác với hệ thống dễ dàng hơn.

2.2.6 Quy trình giao dịch Hyperledger Fabric

- Mô hình client gửi giao dịch.

- Client gửi transaction proposal tới các endorsing peers trong channel.
- Các endorsing peer thực thi chaincode và trả về kết quả.

- Xác thực và kiểm soát quyền truy cập.
 - MSP định danh, xác thực người dùng khi submit transaction.
 - ACL kiểm soát người dùng có quyền truy cập vào channel, chaincode, ... không.
- Gom giao dịch và xác nhận.
 - Client gom các kết quả trả về từ peers và tạo cn_Proposal.
 - Orderer nhóm các cn_Proposal vào block
 - Orderer xác nhận block và phân phối đến committer peers.
- Lưu lại vĩnh viễn giao dịch trên ledger.
 - Transaction được validate và commit lên ledger của peers.
 - Trạng thái mới cập nhật cho tất cả.

2.2.7 Ứng dụng Hyperledger Fabric

- Chuỗi cung ứng: Walmart sử dụng để theo dõi nguồn gốc thực phẩm.
- Ngân hàng: Blockchain Proof of Concept sử dụng Fabric để giảm thiểu rủi ro tài chính.
- Chính phủ: Cơ quan y tế singapore sử dụng trong chia sẻ hồ sơ y khoa.
- Viễn thông: Tập đoàn BT đang thử nghiệm Fabric trong quản lý SIM điện thoại.

2.2.8 Triển vọng và xu hướng phát triển Hyperledger Fabric

- Trở thành nền tảng blockchain hàng đầu cho doanh nghiệp, tổ chức.
- Phát triển các công cụ và framework để đơn giản hóa việc xây dựng/triển khai ứng dụng enterprise blockchain.
- Được ứng dụng trong các ngành:
 - Tài chính (ngân hàng, bảo hiểm).
 - Chuỗi cung ứng.
 - Chính phủ điện tử.
 - Viễn thông.

2.2.9 So sánh công nghệ Blockchain giữa Hyperledger Fabric, Ethereum và Corda

- Về mục tiêu:

- Hyperledger Fabric hướng đến việc cung cấp một kiến trúc mô-đun có khả năng mở rộng cao. Cấu trúc này sẽ được sử dụng rộng rãi trong các ngành công nghiệp khác nhau.
- Ethereum không xây dựng theo cấu trúc mô-đun. Bản chất Ethereum là một nền tảng cho các loại giao dịch và ứng dụng.
- Cũng tương tự như Hyperledger Fabric, mục tiêu của Corda là hướng đến các trường hợp sử dụng cụ thể.

- Ngôn ngữ lập trình:

- Đối với Hyperledger Fabric: ngôn ngữ được sử dụng trong Smart contract được gọi là Chaincode. Chaincode được viết bằng ngôn ngữ Java hoặc Golang.
- Đối với Ethereum: Ngôn ngữ lập trình được xây dựng là Solidity. Đó là một dạng ngôn ngữ cấp cao được dùng để thực hiện các Smart contract. Ưu điểm lớn nhất của Solidity là tính bền vững, dễ hiểu và dễ thực hiện.
- Đối với Corda: ngôn ngữ lập trình trong Corda được gọi là Kotlin. Lý do chính cho việc chọn Kotlin là mức độ tích hợp cao.

- Cơ chế hoạt động: Cơ chế đồng thuận

- Hyperledger Fabric:
 - Cơ chế đồng thuận của Hyperledger Fabric được mở rộng và bao gồm toàn bộ giao dịch.
 - Các nút được giả định đóng vai trò và nhiệm vụ khác nhau trong quá trình đồng thuận. Cụ thể là: khách (client), người đặt hàng (orderer) hay thành viên (peer).
 - Cơ chế hoạt động trong một giao dịch như sau: một khách hàng gửi một giao dịch đến những người xác nhận để khởi tạo một quá trình

cập nhật sổ cái. Sau khi đạt được sự đồng thuận, giao dịch này sẽ được chuyển tiếp cho các thành viên nắm giữ sổ cái để thực hiện giao dịch.

- Hyperledger Fabric cho phép kiểm soát chi tiết sự đồng thuận và hạn chế truy cập vào các giao dịch. Điều đó càng mang lại sự hiệu quả về khả năng mở rộng, đồng thời nâng cao hiệu suất và tính riêng tư của mạng.
- Tuy nhiên, quyền truy cập vào các giao dịch tại Hyperledger Fabric chỉ được giới hạn cho các bên liên quan. Vậy nên, sự đồng thuận chỉ có thể đạt được ở mức giao dịch.
- Ethereum:
 - Hiện tại sử dụng cơ chế đồng thuận Proof of Work (PoW). Tuy nhiên, trong tương lai Ethereum sẽ ra mắt thuật toán đồng thuận Proof of Stake (PoS).
 - Vai trò và nhiệm vụ của các nút trong quá trình đồng thuận là giống hệt nhau.
 - Quyền truy cập vào các giao dịch đạt đến mức sổ cái.
 - Tất cả những người tham gia phải đồng ý với trạng thái của blockchain và đều có quyền truy cập vào tất cả các khối giao dịch đã tồn tại trong blockchain.
 - Mức độ bảo mật thông tin cá nhân không được đánh giá cao. Bởi vì mặc dù dữ liệu được lưu trữ ẩn danh trên blockchain, nhưng chúng vẫn có thể được truy cập bởi mọi người tham gia.
- Corda:
 - Corda là một permissioned blockchain được kiểm soát và phát triển bởi R3 và các tổ chức tham gia vào giao dịch.
 - Sự đồng thuận trong Corda cũng chỉ đạt được ở mức giao dịch bởi các thành viên có liên quan.

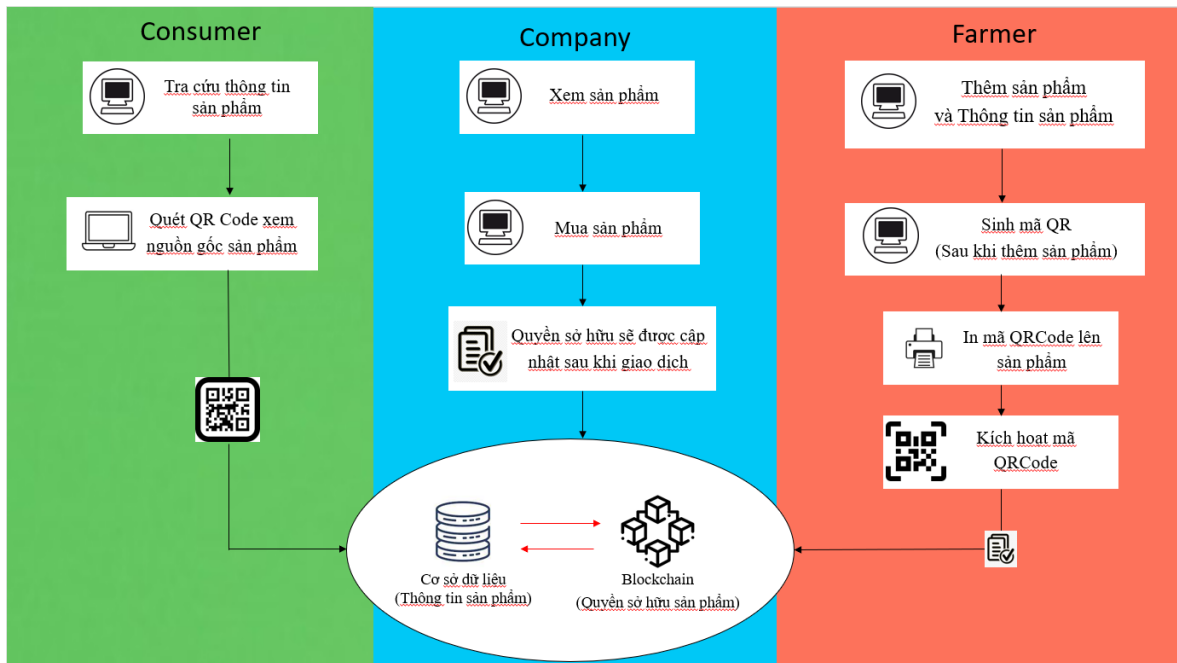
- Corda xây dựng hệ thống hạn chế quyền truy cập vào dữ liệu của người dùng. Khi đó chỉ những người có quyền hoặc có đặc quyền hợp lý mới được truy cập và chia sẻ dữ liệu cá nhân. Điều đó nhằm mục đích đảm bảo tính nhất quán trong một hệ thống chung.
- Tính mô đun và khả năng tương tác của Corda tương đối cao. Điều này cho phép các tổ chức tích hợp các thiết lập hiện có, chẳng hạn như cơ sở dữ liệu và vào mạng Corda.

- Các token hay tiền tệ xây dựng sẵn trong blockchain:

- Hyperledger Fabric và Corda không yêu cầu sự đồng thuận thông qua khai thác vì vậy cũng không tồn tại đơn vị tiền tệ dựng sẵn trong blockchain. Thế nhưng, Hyperledger Fabric vẫn có thể phát triển các đơn vị tiền tệ riêng bằng cách thực hiện chaincode. Còn đối với Corda thì không thể phát triển đơn vị tiền tệ riêng.
- Mặt khác, Ethereum lại cho phép các chức năng thanh toán tiền tệ. Hơn nữa người dùng có thể xây dựng các token riêng bằng cách triển khai một smart contract. Đây cũng là cách mà tiền mã hóa hoặc các tài sản số trên Ethereum được tạo ra.

CHƯƠNG 3. THIẾT KẾ MÔ HÌNH HỆ THỐNG

3.1 Mô hình



Hình 3.1 Mô hình hệ thống

- Mô hình bao gồm 3 User chính:

- + Farmer.
- + Company.
- + Consumer.

3.2 Các chức năng

- Farmer

- + Thêm sản phẩm (Ghi thông tin cần thiết của sản phẩm).
- + Tự động tạo mã QR Code cho sản phẩm được thêm.
- + Xem kho sản phẩm.

- Company

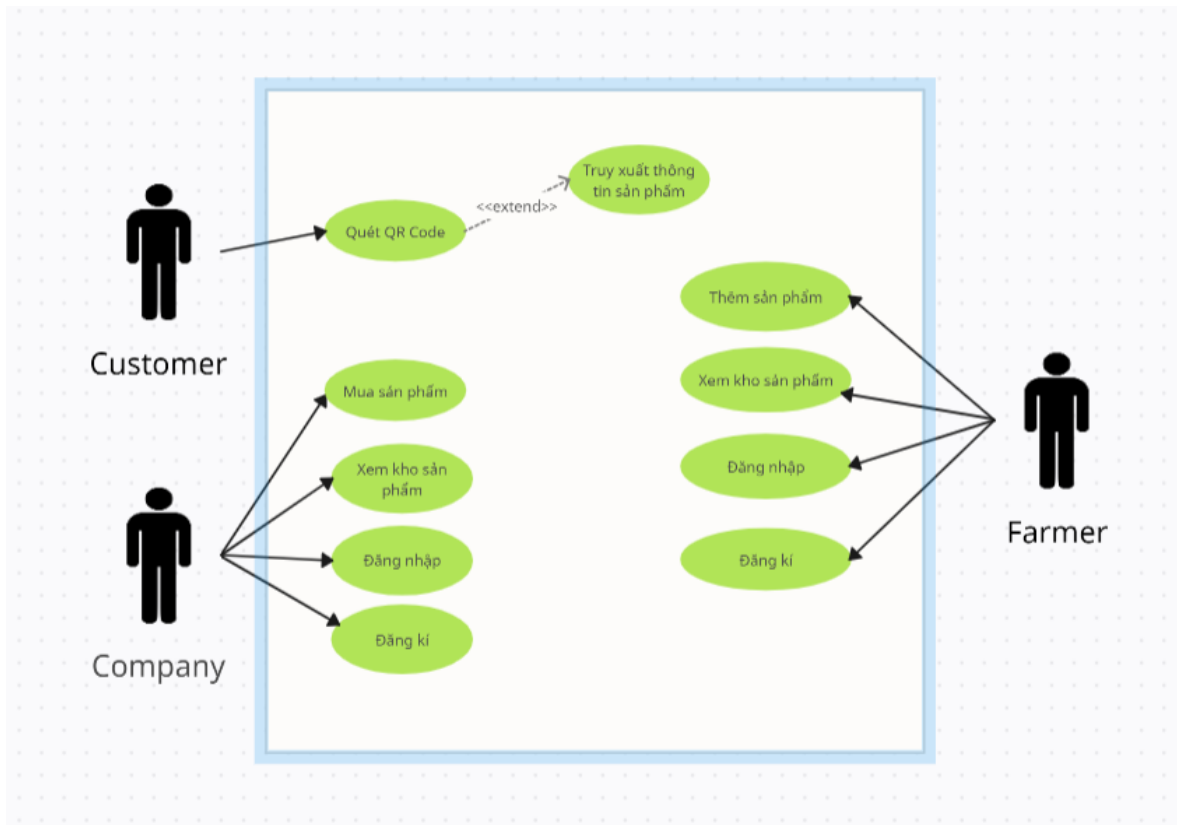
- + Mua sản phẩm (nhập sản phẩm từ Farmer).
- + Xem kho sản phẩm.

- Consumer

+ Truy cập web tra cứu thông tin sản phẩm bằng cách quét mã QR Code.

3.3 Đặc tả Usecase:

Usecase Diagram:



Hình 3.2 Usecase Diagram

3.3.1 Consumer

3.3.1.1 Quét QR Code truy xuất nguồn gốc

Use case ID	UC-3.3.1.1
Use case name	Quét QR Code truy xuất nguồn gốc sản phẩm.
Description	Khách hàng có thể quét mã QR Code của sản phẩm để xem thông tin nguồn gốc sản phẩm.
Priority	Bắt buộc có.

Trigger	Người dùng quét mã QR Code.
Pre-condition(s)	<ul style="list-style-type: none"> - Màn hình chức năng của consumer hiển thị. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Khách hàng quét QR Code của sản phẩm.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng bấm xác minh là consumer. 2. Người dùng bấm chức năng quét QR Code. 3. Ứng dụng yêu cầu quyền truy cập camera. 4. Người dùng cho phép quyền truy cập camera. 5. Người dùng quét mã QR Code có in trên tem sản phẩm. 6. Hệ thống đọc mã QR của sản phẩm và hiển thị thông tin nguồn gốc sản phẩm.
Alternative flow	4a. Người dùng không cho phép quyền truy cập camera: Không thể sử dụng được chức năng quét mã QR.
Exception flow	Mã QR không hợp lệ hoặc không có thông tin: Thông báo lỗi.
Bussiness rules	Chỉ xác minh là cosumer mới sử dụng chức năng này
Non-functional requirement	

3.3.2 Farmer

3.3.2.1 Đăng kí

Use case ID	UC-3.3.2.1
Use case name	Đăng ký
Description	Người dùng nhập thông tin cá nhân để đăng ký tài khoản sử dụng chức năng của farmer.
Priority	Bắt buộc có.

Trigger	Bấm vào đăng kí
Pre-condition(s)	<ul style="list-style-type: none"> - Người dùng chưa có tài khoản. - Màn hình đăng kí hiển thị. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Tài khoản người dùng được tạo thành công trong hệ thống. - Thông tin người dùng như tên, địa chỉ, email, mật khẩu, vai trò được lưu trữ trong database. - Người dùng có thể sử dụng email và mật khẩu để đăng nhập tài khoản.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng bấm xác minh không phải là consumer. 2. Người dùng bấm sang trang Đăng kí. 3. Người dùng nhập thông tin đăng ký gồm tên, địa chỉ, vai trò, email, mật khẩu. 4. Người dùng chọn nút đăng ký. 5. Hệ thống kiểm tra thông tin hợp lệ. 6. Hệ thống tạo tài khoản mới cho người dùng. 7. Hệ thống hiển thị thông báo đăng ký thành công.
Alternative flow	<p>5a. Thông tin đăng ký không hợp lệ:</p> <ul style="list-style-type: none"> • Hệ thống hiển thị thông báo lỗi • Quay lại bước 3
Exception flow	Địa chỉ mail không hợp lệ: Thông báo lỗi yêu cầu nhập lại
Bussiness rules	Email đăng kí là duy nhất, không được trùng với tài khoản đã có.
Non-functional requirement	

3.3.2.2 Đăng nhập

Use case ID	UC-3.3.2.2
Use case name	Đăng nhập

Description	Người dùng đăng nhập vào để sử dụng các chức năng của farmer.
Priority	Bắt buộc có.
Trigger	Người dùng bấm xác minh không phải là consumer.
Pre-condition(s)	<ul style="list-style-type: none"> - Tài khoản đã được tạo. - Các quyền đã được cấp. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Người dùng đăng nhập thành công
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn không để xác minh danh tính không phải là consumer. 2. Người dùng nhập email và mật khẩu để truy cập trang cho farmer. 3. Người dùng sử dụng các chức năng dành cho farmer. 4. Người dùng thoát ra khỏi hệ thống.
Alternative flow	1a. Người dùng chọn có thì chuyển sang giao diện của consumer
Exception flow	Địa chỉ mail hoặc mật khẩu không đúng: Thông báo lỗi, yêu cầu nhập lại.
Bussiness rules	Mật khẩu phải đúng với tài khoản mail đã tạo.
Non-functional requirement	

3.3.2.3 Xem thông tin cá nhân

Use case ID	UC-3.3.2.3
Use case name	Xem thông tin cá nhân.
Description	Người dùng xem – chỉnh thông tin bản thân.
Priority	Bắt buộc có.
Trigger	Chọn mục xem thông tin cá nhân.
Pre-condition(s)	<ul style="list-style-type: none"> - Tài khoản đã được tạo. - Thông tin đã được điền.

	- Đã kết nối mạng.
Post-condition(s)	- Người dùng đã chọn chức năng xem.
Basic flow	1. Người dùng chọn xem thông tin. 2. Người dùng xem thông tin. 3. Người dùng thoát ra khỏi chức năng.
Alternative flow	2a. Người dùng chọn sửa thông tin. 2a1. Màn hình chuyển sang phần sửa thông tin. 2a2. Người dùng lưu thông tin. Quay lại bước 2.
Exception flow	
Bussiness rules	Người dùng chỉ được chỉnh sửa lại thông tin lưu sau 24h.
Non-functional requirement	

3.3.2.4 Thêm sản phẩm

Use case ID	UC-3.3.2.4
Use case name	Thêm sản phẩm.
Description	Người dùng thêm sản phẩm vào danh sách.
Priority	Bắt buộc có.
Trigger	Chọn mục thêm sản phẩm.
Pre-condition(s)	- Tài khoản đã được tạo. - Quyền đã được cấp. - Đã kết nối mạng.
Post-condition(s)	- Người dùng đã chọn chức năng thêm sản phẩm.
Basic flow	1. Người dùng chọn thêm sản phẩm. 2. Người dùng điền đầy đủ các thông tin. 3. Người dùng nhấn submit để thêm. 4. Người dùng thoát khỏi chức năng.
Alternative flow	3a. Người dùng không muốn thêm sản phẩm thì thoát ra.

Exception flow	Người dùng nhập thiếu thông tin -> màn hình hiển thị thông báo và yêu cầu nhập đầy đủ.
Bussiness rules	Người dùng phải nhập chính xác các thông tin của sản phẩm.
Non-functional requirement	

3.3.2.5 Xem kho

Use case ID	UC-3.3.2.5
Use case name	Xem kho.
Description	Người dùng xem danh sách các sản phẩm.
Priority	Bắt buộc có.
Trigger	Chọn mục xem kho.
Pre-condition(s)	<ul style="list-style-type: none"> - Kho có sản phẩm. - Quyền đã được cấp. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Người dùng đã thêm ít nhất 1 sản phẩm.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn xem danh sách. 2. Người dùng thoát khỏi chức năng.
Alternative flow	
Exception flow	
Bussiness rules	Kho sẽ tự động cập nhật dữ liệu sau 15 giây.
Non-functional requirement	

3.3.3 Company

3.3.3.1 Đăng kí

Use case ID	UC-3.3.3.1
Use case name	Đăng ký

Description	Người dùng nhập thông tin cá nhân để đăng ký tài khoản sử dụng chức năng của company.
Priority	Bắt buộc có.
Trigger	Người dùng bấm vào phần đăng kí.
Pre-condition(s)	<ul style="list-style-type: none"> - Người dùng chưa có tài khoản. - Màn hình đăng kí hiển thị. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Tài khoản người dùng được tạo thành công trong hệ thống. - Thông tin người dùng như tên, địa chỉ, email, mật khẩu, vai trò được lưu trữ trong database. - Người dùng có thể sử dụng email và mật khẩu để đăng nhập tài khoản.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng bấm xác minh không phải là consumer. 2. Người dùng bấm sang trang Đăng kí. 3. Người dùng nhập thông tin đăng ký gồm tên, địa chỉ, vai trò, email, mật khẩu. 4. Người dùng chọn nút đăng ký. 5. Hệ thống kiểm tra thông tin hợp lệ. 6. Hệ thống tạo tài khoản mới cho người dùng. 7. Hệ thống hiển thị thông báo đăng ký thành công.
Alternative flow	<p>5a. Thông tin đăng ký không hợp lệ:</p> <ul style="list-style-type: none"> • Hệ thống hiển thị thông báo lỗi • Quay lại bước 3
Exception flow	Địa chỉ mail không hợp lệ: Thông báo lỗi yêu cầu nhập lại
Bussiness rules	Email đăng kí là duy nhất, không được trùng với tài khoản đã có.
Non-functional requirement	

3.3.3.2 Đăng nhập

Use case ID	UC-3.3.3.2
Use case name	Đăng nhập
Description	Người dùng đăng nhập vào để sử dụng các chức năng của company.
Priority	Bắt buộc có.
Trigger	Người dùng bấm xác minh không phải là consumer.
Pre-condition(s)	<ul style="list-style-type: none"> - Tài khoản đã được tạo. - Các quyền đã được cấp. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Người dùng đăng nhập thành công
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn không để xác minh danh tính không phải là consumer. 2. Người dùng nhập email và mật khẩu để truy cập trang cho company. 3. Người dùng sử dụng các chức năng dành cho company. 4. Người dùng thoát ra khỏi hệ thống.
Alternative flow	1a. Người dùng chọn có thì chuyển sang trang của customer.
Exception flow	Địa chỉ mail hoặc mật khẩu không đúng: Thông báo lỗi, yêu cầu nhập lại.
Bussiness rules	Người dùng nhập sai tài khoản 5 lần thì khóa tài khoản.
Non-functional requirement	

3.3.3.3 Xem thông tin cá nhân

Use case ID	UC-3.3.3.3
Use case name	Xem thông tin.
Description	Người dùng xem – chỉnh thông tin bản

	thân.
Priority	Bắt buộc có.
Trigger	Chọn mục xem thông tin cá nhân.
Pre-condition(s)	<ul style="list-style-type: none"> - Tài khoản đã được tạo. - Thông tin đã được điền. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Người dùng đã chọn chức năng xem.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn xem thông tin. 2. Người dùng xem thông tin. 3. Người dùng thoát ra khỏi chức năng.
Alternative flow	<p>2a. Người dùng chọn sửa thông tin.</p> <p>2a1. Màn hình chuyển sang phần sửa thông tin.</p> <p>2a2. Người dùng lưu thông tin.</p> <p>Quay lại bước 2.</p>
Exception flow	
Bussiness rules	Người dùng chỉ được chỉnh sửa lại thông tin lưu sau 24h.
Non-functional requirement	

3.3.3.4 Mua sản phẩm

Use case ID	UC-3.3.3.4
Use case name	Mua sản phẩm.
Description	Người dùng mua sản phẩm trên hệ thống.
Priority	Bắt buộc có.
Trigger	Người dùng chọn mua sản phẩm.
Pre-condition(s)	<ul style="list-style-type: none"> - Đã có tài khoản và đăng nhập. - Sản phẩm có sẵn trong mục. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Giao dịch mua đã thành công.

Basic flow	<ol style="list-style-type: none"> 1. Chọn chức năng mua sản phẩm 2. Chọn sản phẩm cần mua. 3. Người dùng nhấn Mua để mua sản phẩm 4. Người dùng thoát khỏi chức năng.
Alternative flow	<p>3a. Người dùng không nhấn Mua mà thoát ra khỏi chức năng.</p> <p>3a1. Không có giao dịch xảy ra.</p> <p>Tiếp tục bước 4</p>
Exception flow	Người dùng chọn sản phẩm đã bán: Hiện thông báo đã bán.
Bussiness rules	
Non-functional requirement	

3.3.3.5 Xem kho

Use case ID	UC-3.3.3.5
Use case name	Xem kho.
Description	Người dùng xem danh sách các sản phẩm.
Priority	Bắt buộc có.
Trigger	Chọn mục xem kho.
Pre-condition(s)	<ul style="list-style-type: none"> - Kho có sản phẩm. - Quyền đã được cấp. - Đã kết nối mạng.
Post-condition(s)	<ul style="list-style-type: none"> - Người dùng đã mua ít nhất 1 sản phẩm.
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn xem danh sách. 2. Người dùng thoát khỏi chức năng.
Alternative flow	
Exception flow	
Bussiness rules	Kho sẽ tự động cập nhật dữ liệu sau 15 giây.
Non-functional requirement	

CHƯƠNG 4. QUÁ TRÌNH THỰC HIỆN DEMO

4.1 Lời nói đầu

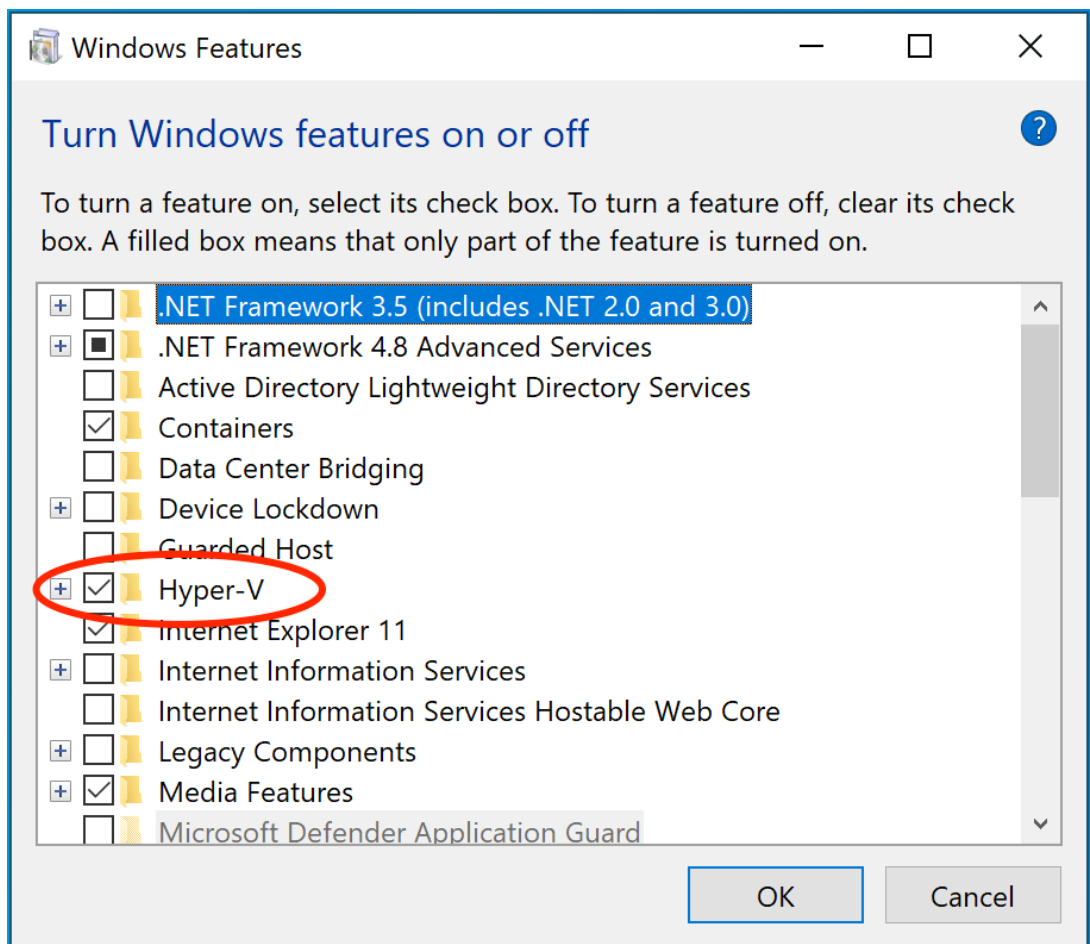
Trong chương này, chúng tôi sẽ trình bày quá trình thực hiện demo cũng như hướng dẫn chi tiết để mọi người có thể thực hiện được tương tự.

4.2 Điều kiện tiên quyết

Cần có các điều kiện tiên quyết sau để chạy mạng thử nghiệm Fabric dựa trên Docker trên máy cục bộ của bạn.

4.2.1 WSL2

Bước 0: Kiểm tra máy tính bạn đã bật tính năng này chưa. Nếu tính năng này là “Enabled”, hãy bỏ qua các bước bên dưới và chuyển đến mục tiếp theo.



Bước 1: Mở PowerShell hoặc Command Prompt của Windows ở chế độ Administrator, nhập lệnh `wsl --install`, sau đó khởi động lại máy của bạn.

Bước 2: Kiểm tra phiên bản WSL nào bạn đang chạy.

Bạn có thể xem các bản phân phối Linux đã cài đặt của mình và kiểm tra phiên bản WSL mà mỗi bản được đặt thành bằng cách nhập lệnh: `wsl -l -v` trong PowerShell hoặc Windows Command Prompt của Windows.

Để đặt phiên bản mặc định thành WSL 2 khi cài đặt bản phân phối Linux mới, hãy sử dụng lệnh: `wsl --set-default-version 2`.

Bước 3: Kiểm tra và bật tính năng như ở Bước 0.

Tài liệu chi tiết: [WSL](#)

4.2.2 Docker

Trong hướng dẫn cài đặt Docker này cũng cần có các điều kiện tiên quyết theo như nhà phát hành để Docker Desktop hoạt động chính xác, bao gồm:

- Nền tảng máy ảo
- Windows Subsystem for Linux (WSL)
- Hypervisor được kích hoạt khi khởi động Windows (Hyper-V)
- Ảo hóa được kích hoạt trong BIOS

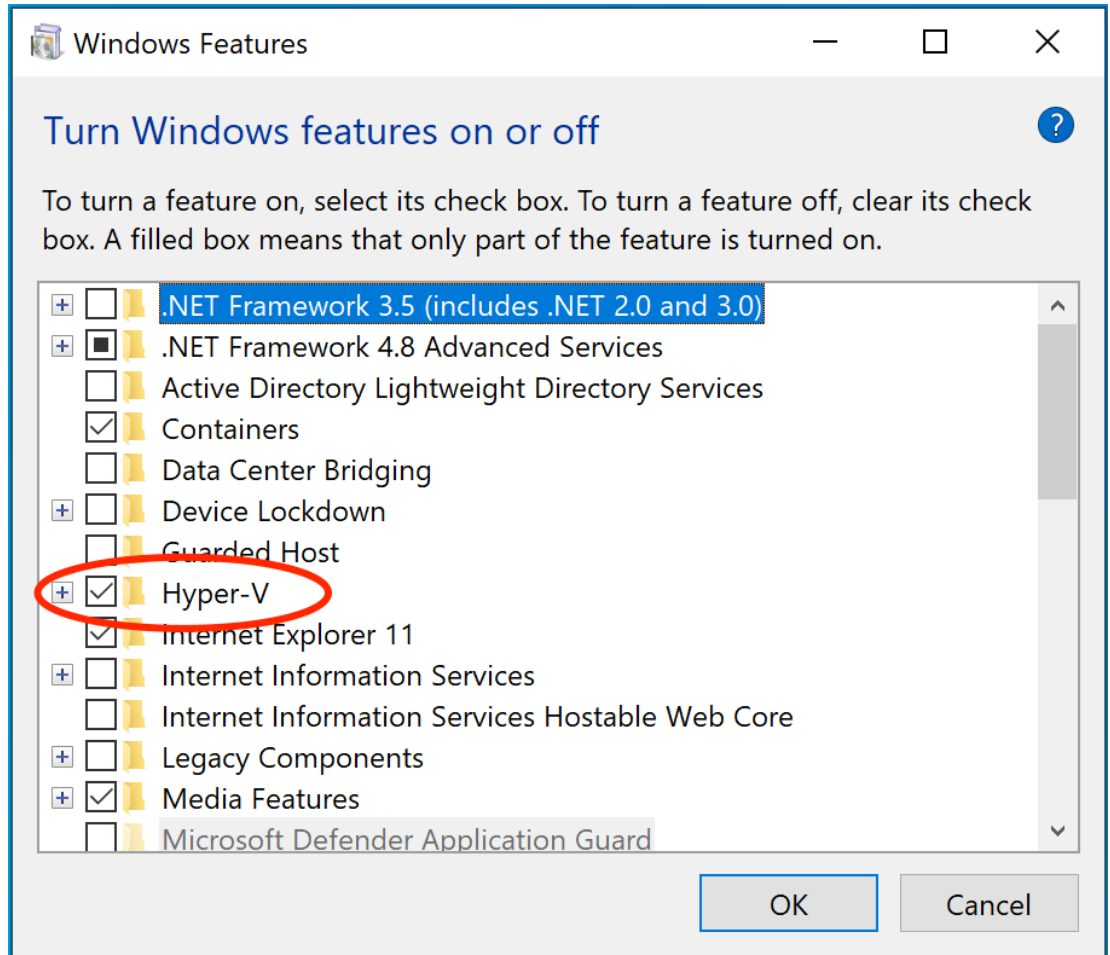
Nếu bạn đã đáp ứng đủ các điều kiện trên hãy bỏ qua các mục hướng dẫn bên dưới và tiến hành cài đặt: [Docker](#)

4.2.2.1 WSL

Ở phần trước chúng tôi đã hướng dẫn cài đặt WSL2.

4.2.2.3 Hyper-V

Bước 0: Kiểm tra máy tính bạn đã bật tính năng này chưa. Nếu tính năng này là “Enabled”, hãy bỏ qua các bước bên dưới và chuyển đến mục tiếp theo.



Bước 1: Cài đặt file [Enable Hyper-V.bat](#)

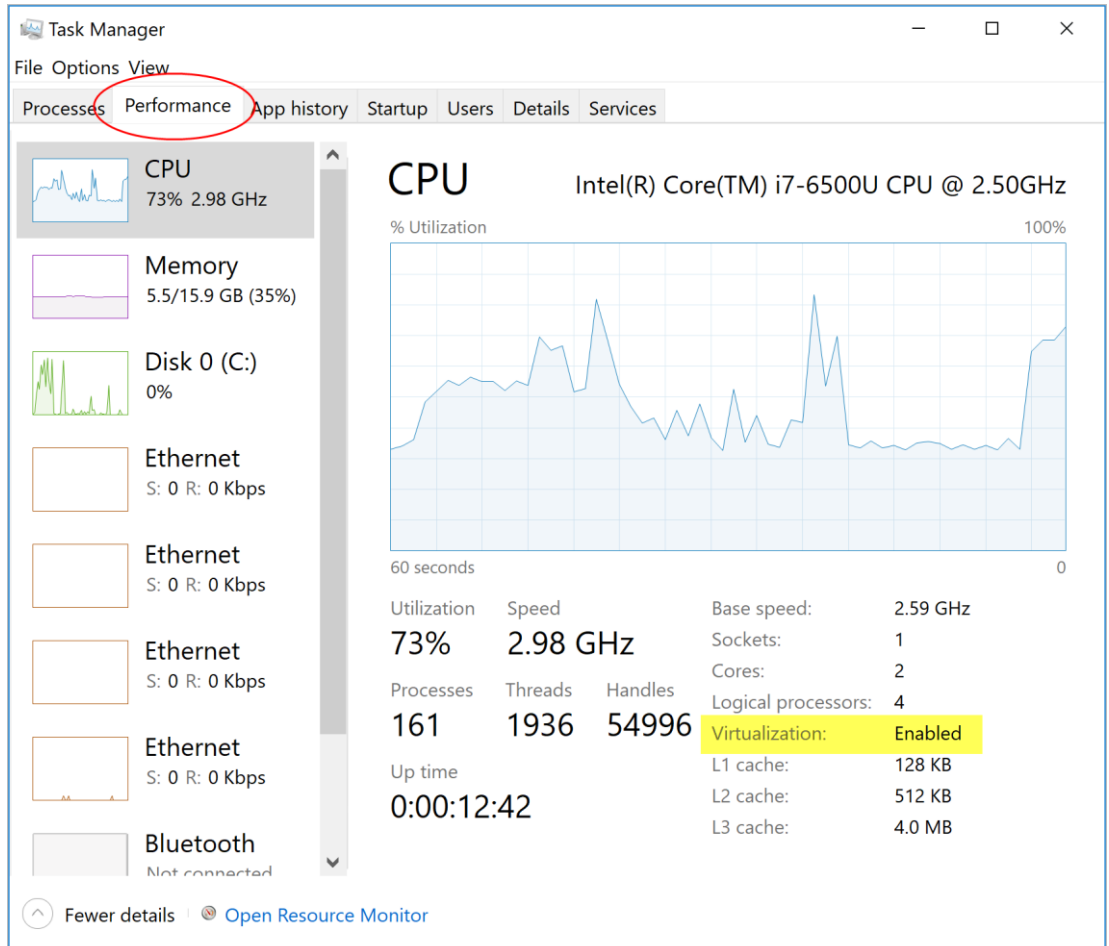
Bước 2: Chọn Run As Administrator (sẽ mất một khoản thời để cài đặt chương trình). Sau khi hoàn thành, bạn khởi động lại máy tính của mình.

Bước 3: Kiểm tra và bật tính năng như ở Bước 0.

Tài liệu chi tiết: [Hyper-V](#), [Download Hyper-V](#)

4.2.2.2 Kích hoạt ảo hóa

Bước 0: Kiểm tra máy tính bạn đã bật tính năng này chưa bằng cách mở “Task Manager” → “Performance” → “CPU”. Nếu tính năng này là “Enabled”, hãy bỏ qua các bước bên dưới và chuyển đến mục tiếp theo.



Bước 1: Khởi động lại máy tính của bạn và truy cập BIOS.

Bước 2: Khi đã vào BIOS, bạn cần tìm phần cấu hình CPU của mình. Tùy thuộc vào hệ thống của bạn, bạn sẽ cần tìm menu có tên Cấu hình CPU, bộ xử lý, Northbridge hoặc Chipset. Bạn có thể đến menu này bằng cách nhấp vào nút tương tự, chẳng hạn như “Advanced” hoặc “Advanced Mode”.

Bước 3: Sau khi tìm thấy phần cấu hình CPU, bạn cần tìm menu hoặc tùy chọn nơi nó cho phép bạn kích hoạt ảo hóa phần cứng. Ảo hóa phần cứng được kích hoạt trong phần tăng tốc. Tùy thuộc vào PC của bạn, hãy tìm bất

kỳ tên nào trong số này hoặc các tên tương tự như Hyber-V, Vanderpool, SVM, AMD-V, Intel Virtualization Technology hoặc VT-X.

Bước 4: Sau khi chọn tùy chọn bật ảo hóa và “Enabled”, hãy tìm tùy chọn cho phép bạn lưu những thay đổi này. Trước khi lưu nó, trước tiên bạn có thể phải thoát khỏi menu và sau đó nhấp vào tùy chọn lưu thay đổi. Bây giờ bạn đã kích hoạt thành công ảo hóa phần cứng trên máy tính của mình.

Bước 5: Thoát BIOS và khởi động lại máy tính của bạn.

Bước 6: Kiểm tra bật tính năng như ở Bước 0.

Tài liệu chi tiết: [Virtualization](#)

4.2.3 Microsoft VS Code (Optional)

Microsoft VS Code cung cấp một IDE tích hợp chặt chẽ với các bản phân phối WSL2 Linux. Tìm kiếm Microsoft Marketplace trong Mã VS cho gói tiện ích mở rộng Phát triển từ xa để biết thêm thông tin. Gói này bao gồm, cùng với những thứ khác, extension “Remote – WSL” và extension “Remote – Containers”.

4.2.4 Git For Windows (Optional)

Mặc dù không bắt buộc, nhưng nếu bạn quyết định cài đặt Git trên Windows và quản lý kho lưu trữ Fabric nguyên bản (trái ngược với trong WSL2 và cài đặt Git của nó), thì hãy đảm bảo bạn định cấu hình Git như sau:

```
git config --global core.autocrlf false
git config --global core.longpaths true
```

4.2.5 Npm & Node

Chúng tôi quyết định lựa chọn ExpressJS làm Framework cho ứng dụng Web với những module được cung cấp và hỗ trợ bởi Fabric SDK Node. Vì vậy bạn hãy cài đặt Npm & Node cùng phiên bản trên cả Windows và Linux (WSL2).

- Cài đặt trên Windows:

Bước 1: Tải bản cài đặt Latest LTS [Nodejs](#)

Bước 2: Mở tệp cài đặt và làm theo các bước hướng dẫn trên màn hình

Bước 3: Kiểm tra xem Npm & Node đã được cài đặt thành công bằng cách mở Command Prompt và chạy các lệnh:

```
npm -v
node -v
```

- Cài đặt trên WSL:

Bước 1: Mở terminal trong WSL.

Bước 2: Chạy lệnh sau để tải và cài đặt NVM (Node Version Manager)

```
curl -o- https://raw.githubusercontent.com/nvm-
sh/nvm/v0.39.0/install.sh | bash
```

Bước 3: Sau khi cài đặt xong, đóng và mở lại terminal hoặc chạy lệnh sau để tải nvm:

```
source ~/.bashrc
```

Bước 4: Sử dụng nvm để cài đặt Node.js và npm với phiên bản bạn mong muốn

```
nvm install 20.10.0
nvm use 20.10.0
```

Bạn cũng có thể cài đặt phiên bản cụ thể của npm:

```
npm install -g npm@10.2.3
```

Bước 5: Kiểm tra xem Npm & Node đã được cài đặt bằng cách chạy các lệnh:

```
npm -v
node -v
```

Lưu ý: Chúng tôi khuyến khích bạn cài đặt phiên bản Npm **v10.2.3** & Node **v20.10.0**, vì đây là phiên bản mà chúng tôi sử dụng cho ứng dụng Web này)

4.3 Cài đặt ứng dụng Web

Bước 1: Mở WSL và Clone code

```
git clone https://github.com/devilkun1/Blockchain_Origin_Tracking.git
```

```
cd Blockchain-Origin_Tracking
```

Bước 2: Khởi động Docker Desktop. Sau đó di chuyển đến đường dẫn 'hyperledger-fabric' và chạy tệp bash trong terminal

```
cd hyperledger-fabric
```

```
./install-fabric.sh -f '2.5.4' -c '1.5.7' d s b
```

Bước 3: Sau khi cài đặt xong. Tiếp tục di chuyển đến đường dẫn 'fabric-samples/test-network'.

```
cd fabric-samples/test-network
```

Bước 4: Dừng chạy container/image để tránh xung đột lệnh

```
./network.sh down
```

Bước 5: Tạo kênh (channel) với chứng chỉ CA (Certificate Authorities)

```
./network.sh up createChannel -c mychannel -ca
```

Bước 6: Triển khai Chaincode

```
./network.sh deployCC -ccn basic -ccp ../../chaincode-custom/ -ccl  
javascript
```

Bước 7: Quay lại thư mục dự án (Blockchain-Origin_Tracking), cài đặt tất cả các gói trong file pack.json

```
cd ~/Blockchain-Origin_Tracking
```

```
npm install
```

Bước 8: Khởi động và truy cập Web

```
npm start
```

```
http://localhost:3000
```

4.4 Video chi tiết quá trình thực hiện

Link: [click here](#)

CHƯƠNG 5. KẾT LUẬN

5.1 Tổng kết quá trình triển khai

- Tóm tắt các bước xây dựng web App nodejs kết nối với Network Fabric.
- Cài đặt, khởi tạo, cấu hình các components của Fabric Network.
- Phát triển Chaincode để thực hiện các giao dịch với tài sản.
- Xây dựng Backend tương tác với Blockchain thông qua Fabric SDK Node.
- Demo truy vấn và hiển thị thông tin sản phẩm có xác thực trên Blockchain.

5.2 Kết quả đạt được

- Hoàn thành web app Nodejs kết nối và truy xuất dữ liệu từ Fabric Network.
- Minh chứng được tính xác thực về nguồn gốc, quá trình sản xuất của sản phẩm, quá trình chuyển đổi qua các giai đoạn, qua các owner khác nhau.
- Demo hoạt động ổn định với traffic dữ liệu nhất định.

5.3 Video chi tiết kết quả thực hiện

Link: [click here](#)

5.4 Đánh giá ý nghĩa, giá trị khoa học, thực tiễn của hệ thống

- Chứng minh hiệu quả của Fabric áp dụng cho doanh nghiệp, tổ chức.
- Đóng góp cho nghiên cứu phát triển Blockchain trong lĩnh vực CNTT.
- Là cơ sở để nhân rộng, ứng dụng Blockchain Fabric vào thực tế (Ví dụ: chuỗi cung ứng, tài chính - ngân hàng, bảo hiểm, v.v)

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] <http://tinyurl.com/bdf79y8v>

[2] <http://tinyurl.com/47dh23t7>

[3] <http://tinyurl.com/yacb4zr7>

Tiếng Anh

[4] <http://tinyurl.com/5be6p7rh>

[5] <http://tinyurl.com/35nymuvx>

[6] <https://hyperledger-fabric.readthedocs.io/en/latest/>