

**BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN 2**

---o0o---



BÁO CÁO

**MÔN HỌC: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM
ĐỀ TÀI: ỨNG DỤNG TRUY XUẤT NGUỒN GỐC
THỰC PHẨM BẰNG BLOCKCHAIN**

Người hướng dẫn: TS. CHÂU VĂN VÂN

Thành viên nhóm 14:

Bùi Anh Tuấn	N22DCAT062
Lã Quang Hiếu	N22DCAT022
Trần Phước Lợi	N22DCAT033
Phạm Hữu Thanh Thiện	N22DCAT052
Nguyễn Trung Kiên	N22DCAT029

LỜI CAM ĐOAN

Chúng tôi cam đoan đây là đề án nghiên cứu của riêng nhóm dưới sự hướng dẫn của Thầy **TS. Châu Văn Vân.**

Sản phẩm nghiên cứu của chúng tôi là kết quả của sự nỗ lực của toàn nhóm, không sao chép của người khác. Các kết quả được đưa ra hay số liệu thống kê trong đề án là trung thực và chưa từng được ai công bố trong một đề án nào khác.

TP.HCM, ngày 24 tháng 04 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến thầy **TS. Châu Văn Vân**. Thầy là người đã định hướng, theo dõi và hỗ trợ chúng em thực hiện đề án báo cáo “Ứng dụng truy suất nguồn gốc thực phẩm bằng Blockchain”.

Chúng tôi xin chân thành cảm ơn đến các thành viên trong nhóm, các bạn nhóm khác đã có sự giúp đỡ và chia sẻ những hiểu biết và chúng tôi hoàn thành tốt đề án này.

Qua sự nỗ lực không ngừng chúng tôi đã cố gắng thực hiện tốt các nội dung các nghiên cứu, nhưng cũng không thể tránh khỏi những thiếu sót nhất định. Chúng tôi rất mong nhận được các ý kiến đóng góp của Thầy và các bạn để hoàn thành đề án của chúng tôi tốt nhất.

TP.HCM, ngày 24 tháng 04 năm 2025

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, vấn đề đảm bảo an toàn thực phẩm và truy xuất nguồn gốc hàng hóa ngày càng trở nên cấp thiết. Thực trạng hàng giả, hàng kém chất lượng, sản phẩm không rõ nguồn gốc tràn lan trên thị trường đang đặt ra nhiều thách thức cho người tiêu dùng và các cơ quan quản lý.

Hiện nay, nhiều hệ thống truy xuất nguồn gốc vẫn hoạt động theo mô hình tập trung truyền thống, trong đó dữ liệu có thể bị thay đổi bởi người quản trị hoặc tin tặc, làm suy giảm tính minh bạch và độ tin cậy của hệ thống. Blockchain – với đặc tính phân tán, minh bạch và không thể thay đổi dữ liệu sau khi được ghi nhận – được xem là một giải pháp đầy hứa hẹn để khắc phục những hạn chế nêu trên.

Trong khuôn khổ một dự án học tập, nhóm chúng em đã triển khai hệ thống FoodTrade – một ứng dụng truy xuất nguồn gốc thực phẩm áp dụng công nghệ Blockchain, sử dụng Ganache làm môi trường mô phỏng mạng Ethereum cục bộ. Dự án bao gồm cả giao diện người dùng (frontend) và hệ thống xử lý phía máy chủ (backend), nhằm minh họa rõ nét cách công nghệ Blockchain có thể được tích hợp vào quy trình xác thực và truy vết nguồn gốc sản phẩm thực phẩm.

Đề án trình bày toàn bộ quá trình phát triển hệ thống FoodTrade, từ thiết kế, triển khai đến thử nghiệm. Đồng thời, nhấn mạnh vai trò của Blockchain trong việc tăng cường độ tin cậy và minh bạch cho các giải pháp truy xuất nguồn gốc. Cấu trúc đề án gồm các phần:

- **Chương 1: Cơ sở lý thuyết và công nghệ sử dụng**
- **Chương 2: Phân tích và thiết kế hệ thống**
- **Chương 3: Triển khai và thử nghiệm hệ thống**
- **Chương 4: Kết luận và kiến nghị**

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN.....	ii
LỜI MỞ ĐẦU.....	iii
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG	1
1.1 Tổng quan về công nghệ truy xuất nguồn gốc.....	1
1.2 Giới thiệu công nghệ Blockchain	1
1.2.1 Khái quát về công nghệ Blockchain.....	1
1.2.2 Tính phù hợp của Blockchain trong việc tăng cường độ tin cậy.....	2
1.2.3 Mô hình ứng dụng Blockchain vào sản phẩm FoodTrade	2
1.3 Môi trường Ganache và Smart Contract.....	3
1.3.1. Giới thiệu và vai trò của Ganache.....	4
1.3.2. Quy trình sử dụng Ganache trong dự án FoodTrade.....	5
1.3.3. Smart Contract trong hệ thống FoodTrade	5
1.4 Các công cụ và nền tảng hỗ trợ	6
1.5 Vấn đề tồn tại	7
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	9
2.1 Kiến trúc tổng thể của hệ thống.....	9
2.2 Mô hình Use Case và các thành phần chức năng	9
2.3 Thiết kế hợp đồng thông minh (Smart Contract).....	14
2.4 Thiết kế frontend và backend	15
2.5 Thiết kế giao tiếp giữa các tầng.....	15
2.6 Các chức năng nền tảng:.....	15
CHƯƠNG 3. TRIỂN KHAI VÀ THỬ NGHIỆM HỆ THỐNG.....	19
3.1 Môi trường và công cụ cài đặt.....	19
3.2 Cài đặt Ganache và triển khai Smart Contract	19
3.3 Kết nối frontend/backend với Blockchain.....	19
3.4 Thử nghiệm và đánh giá kết quả hệ thống.....	20
CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ	21
4.1 Kết luận về đề tài.....	21
4.2 Hạn chế của hệ thống	21
4.3 Kiến nghị và định hướng phát triển trong tương lai	22
TÀI LIỆU THAM KHẢO	23

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

1.1 Tổng quan về công nghệ truy xuất nguồn gốc

Truy xuất nguồn gốc thực phẩm là quá trình theo dõi và ghi nhận lại toàn bộ hành trình của một sản phẩm từ nơi sản xuất, qua các khâu trung gian, đến tay người tiêu dùng. Đây là một giải pháp quan trọng giúp nâng cao sự minh bạch, đảm bảo chất lượng và an toàn thực phẩm, đồng thời bảo vệ quyền lợi của người tiêu dùng.

Trong bối cảnh thị trường ngày càng cạnh tranh, truy xuất nguồn gốc còn góp phần xây dựng niềm tin giữa doanh nghiệp và khách hàng. Các phương pháp truyền thống thường dựa vào hệ thống cơ sở dữ liệu tập trung, vốn dễ bị sửa đổi, mất mát dữ liệu hoặc gặp khó khăn trong việc đồng bộ giữa các bên liên quan. Điều này đặt ra yêu cầu cần có một phương pháp mới hiệu quả hơn, minh bạch hơn để thay thế.

1.2 Giới thiệu công nghệ Blockchain

1.2.1 Khái quát về công nghệ Blockchain

Công nghệ Blockchain bắt đầu nổi lên vào năm 2008, ban đầu được biết đến như là nền tảng của đồng tiền điện tử Bitcoin. Về bản chất, Blockchain là một sổ cái phân tán cho phép ghi nhận và chuyển giao dữ liệu một cách an toàn, minh bạch mà không cần đến sự can thiệp của các hệ thống xác thực tập trung. Các bản ghi (record) trên sổ cái được cập nhật theo thời gian và được mã hóa, đảm bảo tính toàn vẹn của dữ liệu trong suốt quá trình lưu trữ.

Tính chất phân tán và độ tin cậy cao của Blockchain đã chứng minh được hiệu quả rõ rệt trong các hệ thống tài chính lớn, và đang được nghiên cứu để áp dụng trong nhiều lĩnh vực khác nhau. Kể từ năm 2017, công nghệ này bắt đầu được nghiên cứu ứng dụng trong lĩnh vực chuỗi cung ứng thực phẩm. Nhờ đó, người tiêu dùng có thể truy xuất nguồn gốc thực phẩm thông qua các ứng dụng di động. Một số giải pháp ứng dụng Blockchain trong lĩnh vực nông nghiệp đã được trình bày trong. Tuy nhiên, việc áp dụng Blockchain phụ thuộc vào đặc thù và yêu cầu riêng của từng hệ thống.

1.2.2 Tính phù hợp của Blockchain trong việc tăng cường độ tin cậy

Blockchain là công nghệ lưu trữ dữ liệu theo dạng chuỗi khối. Mỗi khối dữ liệu được liên kết với khối trước đó thông qua các thuật toán mã hóa phức tạp. Nhờ đó, khi một khối đã được thêm vào chuỗi, nó gần như không thể bị chỉnh sửa, xóa bỏ hoặc thay đổi thứ tự – với điều kiện là không có sự xuất hiện của công nghệ vượt trội như máy tính lượng tử trong tương lai gần.

Những đặc điểm nổi bật của Blockchain bao gồm:

- **Tính bất biến:** Một khi giao dịch hoặc dữ liệu đã được ghi nhận bởi người giữ khóa riêng (private key), dữ liệu đó sẽ tồn tại vĩnh viễn và không thể bị sửa đổi.
- **Tính minh bạch:** Blockchain cải thiện đáng kể tính minh bạch trong quản lý và lưu trữ dữ liệu, đặc biệt khi so sánh với các phương pháp truyền thống như ghi chép sổ sách.
- **Loại bỏ trung gian:** Các hệ thống dựa trên Blockchain có khả năng loại bỏ bên trung gian trong quá trình lập hồ sơ hoặc chuyển giao tài sản.
- **Tính phi tập trung:** Hệ thống hoạt động trên một mạng lưới máy tính phi tập trung, giúp giảm nguy cơ tấn công, thời gian chết của máy chủ và mất mát dữ liệu.
- **Độ tin cậy:** Mạng lưới phi tập trung và tính minh bạch cao giúp tăng cường niềm tin giữa các bên trong giao dịch, đặc biệt ở những quốc gia mà lòng tin vào bên trung gian còn hạn chế.

Tóm lại, các ứng dụng xây dựng trên nền tảng Blockchain có thể kế thừa toàn bộ các đặc tính vượt trội kể trên, đặc biệt là về mặt **xác thực và độ tin cậy**. Trên thực tế, nhiều nghiên cứu trong và ngoài nước đều nhận định rằng Blockchain là một trong những công nghệ triển vọng nhất cho các giải pháp xác thực dữ liệu.

1.2.3 Mô hình ứng dụng Blockchain vào sản phẩm FoodTrade

Blockchain vận hành dựa trên mô hình mạng ngang hàng (peer-to-peer) phân tán. Khi một khối dữ liệu mới được thêm vào chuỗi, cần có sự đồng thuận từ các node trong mạng.

Sau khi đạt được sự đồng thuận, khối dữ liệu sẽ được thêm vào chuỗi tại tất cả các node. Điều này đồng nghĩa với việc: số lượng node càng lớn \rightarrow độ tin cậy càng cao \rightarrow dữ liệu được nhân bản nhiều hơn \rightarrow yêu cầu đồng thuận cũng cao hơn.

Để đạt được độ tin cậy cao như vậy, Blockchain đã phải đánh đổi về mặt dung lượng lưu trữ và tốc độ xử lý. Đây cũng chính là lý do vì sao Blockchain phù hợp với các ứng dụng trong lĩnh vực tiền điện tử – nơi cần sự an toàn và minh bạch hơn là tốc độ.

Trong các hệ thống ứng dụng Blockchain hiện nay, dữ liệu thường được chia làm hai loại:

- **Dữ liệu on-chain:** Là dữ liệu được lưu trực tiếp trên Blockchain, thường là các thông tin ngắn gọn, mang tính xác thực cao như các mã băm, sự kiện, lệnh tạo/sửa/xóa đối tượng.
- **Dữ liệu off-chain:** Là dữ liệu có kích thước lớn (ảnh, video, tài liệu...), được lưu trữ ngoài Blockchain và liên kết với dữ liệu on-chain thông qua mã định danh.



Mô hình ứng dụng blockchain(Ganache)

1.3 Môi trường Ganache và Smart Contract

1.3.1. Giới thiệu và vai trò của Ganache

Ganache (trước đây là TestRPC) là một công cụ giả lập mạng Ethereum cục bộ, hỗ trợ cả giao diện dòng lệnh (CLI) và giao diện đồ họa (GUI). Ganache mô phỏng đầy đủ các tính năng của một mạng Ethereum thực thụ, bao gồm:

- **Mạng riêng (Private Blockchain):** Tạo mạng Ethereum riêng với cấu hình linh hoạt về chain ID, gas limit, block time.
- **Tài khoản ảo (Test Accounts):** Cung cấp sẵn nhiều tài khoản với lượng ETH giả lập lớn, giúp thực hiện giao dịch và triển khai hợp đồng mà không tốn phí thực.
- **Kiểm soát block và time:** Cho phép người phát triển cấp tốc hoặc chậm quá trình tạo khối, mô phỏng các sự kiện thời gian thực như timeout hoặc các lần gia hạn.
- **Tính năng reset và kiểm tra trạng thái:** Dễ dàng reset toàn bộ trạng thái chuỗi về ban đầu hoặc lưu snapshot, thuận tiện cho việc kiểm thử lặp lại nhiều kịch bản.
- **Theo dõi giao dịch và sự kiện:** Giao diện GUI hiển thị chi tiết từng giao dịch, bao gồm hash, block number, gas used, logs, giúp debug và phân tích nhanh chóng.
- **Forking mainnet hoặc testnet:** Ganache CLI hỗ trợ fork dữ liệu từ mạng chính Ethereum hoặc mạng thử nghiệm, giúp kiểm thử contract theo dữ liệu thực.

Nhờ những tính năng trên, Ganache trở thành công cụ không thể thiếu trong quá trình:

- **Phát triển Smart Contract:** Nhanh chóng biên dịch và deploy contract trên môi trường cục bộ.
- **Kiểm thử và debug:** Theo dõi chi tiết transaction, event và trạng thái của blockchain.
- **Tích hợp liên tục (CI/CD):** Chạy các test suite tự động mà không phụ thuộc vào mạng bên ngoài.

1.3.2. Quy trình sử dụng Ganache trong dự án FoodTrade

- **Khởi tạo mạng Ganache:** Mở GUI hoặc chạy ganache-cli với các tham số như --port, --gasLimit, --networkId phù hợp với môi trường phát triển.
- **Cài đặt Truffle hoặc Hardhat:** Cấu hình project để kết nối đến mạng Ganache bằng URL RPC (<http://127.0.0.1:7545> hoặc port tương ứng).
- **Biên dịch Smart Contract:** Sử dụng lệnh truffle compile hoặc plugin tương ứng trong Hardhat.
- **Triển khai hợp đồng:** Viết scripts migration (Truffle) hoặc deploy script (Hardhat) để deploy contract lên Ganache.
- **Kiểm thử tự động:** Viết test cases bằng Mocha/Chai (Truffle) hoặc Waffle (Hardhat), chạy truffle test hoặc npx hardhat test.
- **Debug và điều chỉnh:** Sử dụng Ganache GUI để theo dõi logs, replay transaction hoặc reset chain khi cần.

1.3.3. Smart Contract trong hệ thống FoodTrade

Khái niệm: Smart Contract là các đoạn mã Solidity được biên dịch thành bytecode và lưu trữ trên blockchain. Chúng thực thi tự động khi có giao dịch gửi đến các hàm đã định nghĩa.

Các thành phần chính của Smart Contract FoodTrade:

- **Data Structures:** Khai báo struct Product với các trường như id, owner, timestamp, metadataHash.
- **Mapping:** Sử dụng mapping(uint256 => Product) để quản lý sản phẩm theo ID.
- **Modifiers:** Định nghĩa các modifier như onlyOwner, onlyProducer để kiểm soát quyền truy cập.
- **Events:** Định nghĩa event ProductCreated, ProductUpdated, OwnershipTransferred để ghi log và dễ dàng theo dõi trên front-end.

- **Functions:** Bao gồm:

- `createProduct(uint256 id, string memory metadataHash)`: Tạo mới sản phẩm với ID duy nhất.
- `updateProduct(uint256 id, string memory newStatus)`: Cập nhật trạng thái hoặc vị trí.
- `getProduct(uint256 id) public view returns (Product memory)`: Truy vấn thông tin sản phẩm.

Triển khai và tương tác:

- Khi deploy, contract sẽ trả về **contract address**.
- **ABI (Application Binary Interface)** được tự động sinh ra trong quá trình compile, giúp front-end hiểu cấu trúc hàm và các kiểu dữ liệu.
- Front-end (React + Web3.js) sử dụng address và ABI để tạo instance contract và gọi hàm read/write.

1.4 Các công cụ và nền tảng hỗ trợ

Để xây dựng hệ thống truy xuất nguồn gốc thực phẩm bằng Blockchain một cách hiệu quả, nhiều công cụ và nền tảng hiện đại đã được sử dụng:

- **Next.js**: Framework React hỗ trợ Server-Side Rendering (SSR) và Static Site Generation (SSG), tối ưu hóa SEO và hiệu năng tải trang.
- **Tailwind CSS**: Thư viện CSS tiện ích giúp xây dựng giao diện nhanh chóng, linh hoạt và nhất quán về thiết kế.
- **Node.js & Express.js**: Môi trường JavaScript phía server và framework HTTP dùng để xây dựng API, xử lý logic nghiệp vụ và tương tác với blockchain.

- **MongoDB**: Cơ sở dữ liệu NoSQL dùng để lưu trữ thông tin người dùng, sản phẩm, nhật ký hoạt động và các dữ liệu off-chain khác.
- **Web3.js (Ethers.js)**: Thư viện kết nối frontend và backend với mạng Ethereum, hỗ trợ gửi giao dịch, gọi hàm Smart Contract và theo dõi sự kiện.
- **Ganache**: Công cụ giả lập Ethereum local, dùng để phát triển, deploy và debug Smart Contract ngay trên máy tính cá nhân.
- **Hardhat (Truffle)**: Framework phát triển Smart Contract, hỗ trợ biên dịch, deploy và viết test cases tự động.

Việc phối hợp linh hoạt giữa các công cụ trên giúp dự án đảm bảo tính hiện đại, dễ mở rộng và duy trì trong tương lai.

1.5 Vấn đề tồn tại

Hiện nay, nhiều hệ thống quản lý dữ liệu vẫn đang được vận hành theo mô hình tập trung, trong đó toàn bộ thành phần của hệ thống được đặt trên một hoặc một nhóm nhỏ máy chủ, do một hoặc một nhóm người quản trị kiểm soát. Mô hình này tiềm ẩn nhiều rủi ro nghiêm trọng:

- Nguy cơ bị chỉnh sửa dữ liệu: Dữ liệu lưu trữ trên server có thể dễ dàng bị chỉnh sửa, không chỉ bởi các hacker mà còn bởi chính người quản trị hệ thống.
- Dễ bị phá hoại: Dữ liệu tập trung, dù có bản sao lưu (backup), vẫn có thể bị xóa hoặc phá hoại hoàn toàn trong trường hợp xảy ra các cuộc tấn công có chủ đích.
- Thiếu liên kết trong chuỗi cung ứng: Thông tin về sản phẩm hoặc hàng hóa thường bị phân tán theo từng bộ phận riêng biệt, thiếu sự liên kết và đồng bộ giữa các khâu trong chuỗi cung ứng.

Như vậy, với mô hình vận hành hiện tại, hệ thống dễ bị tổn thương bởi yếu tố con người, đồng thời tính minh bạch và độ tin cậy chưa cao.

Để khắc phục những hạn chế này, một giải pháp hiệu quả đang được đề xuất là chuyển đổi từ mô hình tập trung sang mô hình phân tán. Theo đó, dữ liệu sẽ được lưu trữ phân tán trên nhiều máy chủ vật lý khác nhau. Việc thêm, sửa, xóa dữ liệu chỉ được thực hiện khi có

sự đồng thuận từ các bên tham gia hệ thống. Đây chính là đặc điểm cốt lõi của công nghệ Blockchain.

Trong phần tiếp theo, bài báo cáo sẽ trình bày giải pháp tích hợp công nghệ Blockchain vào hệ thống FoodTrade nhằm giải quyết triệt để các vấn đề nêu trên.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Kiến trúc tổng thể của hệ thống

Hệ thống FoodTrade được tổ chức theo mô hình ba tầng (3-tier architecture), bao gồm:

- **Tầng trình diễn (Presentation Layer):** Frontend sử dụng Next.js và Tailwind CSS để xây dựng giao diện web, tương tác người dùng, hiển thị kết quả truy xuất.
- **Tầng xử lý nghiệp vụ (Application Layer):** Backend phát triển trên Node.js và Express.js, cung cấp RESTful API cho frontend, xử lý xác thực, ủy quyền, và điều phối tương tác với blockchain.
- **Tầng dữ liệu (Data Layer):** Bao gồm hai nguồn dữ liệu:
 - **On-chain:** Ethereum private network (Ganache/Hardhat) lưu trữ Smart Contract, đảm bảo tính bất biến và minh bạch.
 - **Off-chain:** MongoDB lưu trữ thông tin người dùng, nhật ký truy vấn, và metadata sản phẩm để giảm chi phí gas và tăng hiệu suất.

Kiến trúc này đảm bảo tách biệt rõ ràng giữa giao diện, logic nghiệp vụ và lưu trữ dữ liệu, thuận tiện cho mở rộng và bảo trì.

2.2 Mô hình Use Case và các thành phần chức năng

Mô hình Use Case gồm các tác nhân chính: **Producer**, **Distributor**, **Retailer**, và **Consumer**.

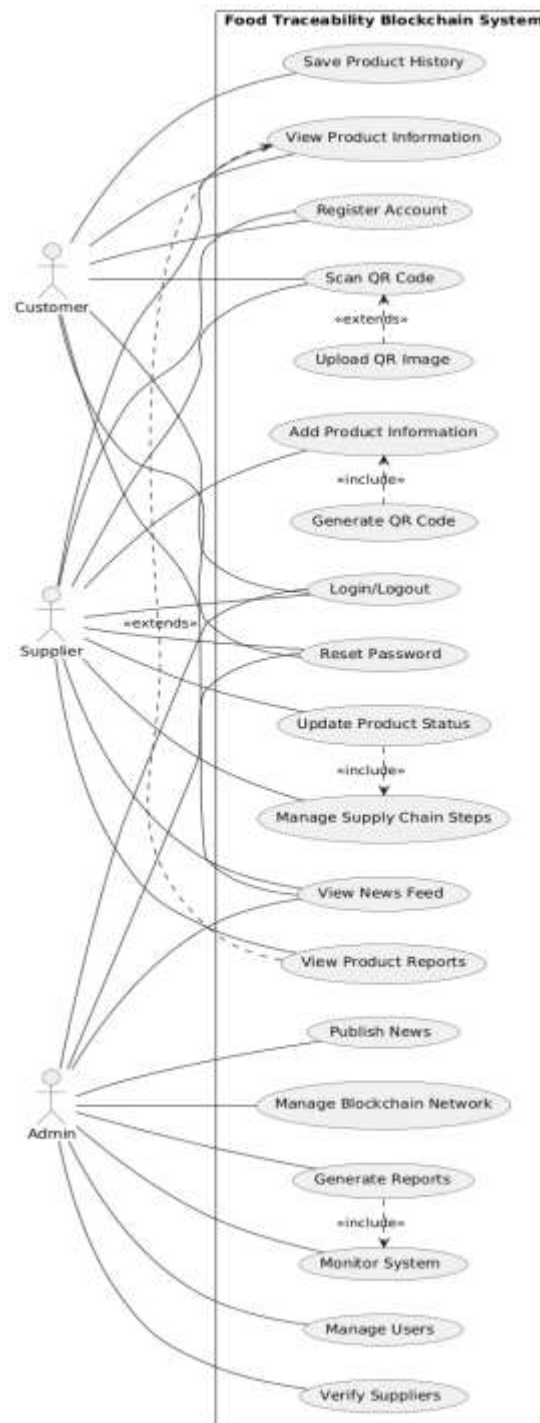
Các chức năng chính:

- **Producer tạo sản phẩm mới:** Ghi nhận thông tin lô hàng và metadata lên Smart Contract.
- **Distributor cập nhật trạng thái vận chuyển:** Ghi log vận chuyển mỗi khi sản phẩm đổi chủ hoặc thay đổi vị trí.
- **Retailer xác nhận nhận hàng:** Ghi nhận trạng thái giao nhận trên blockchain.

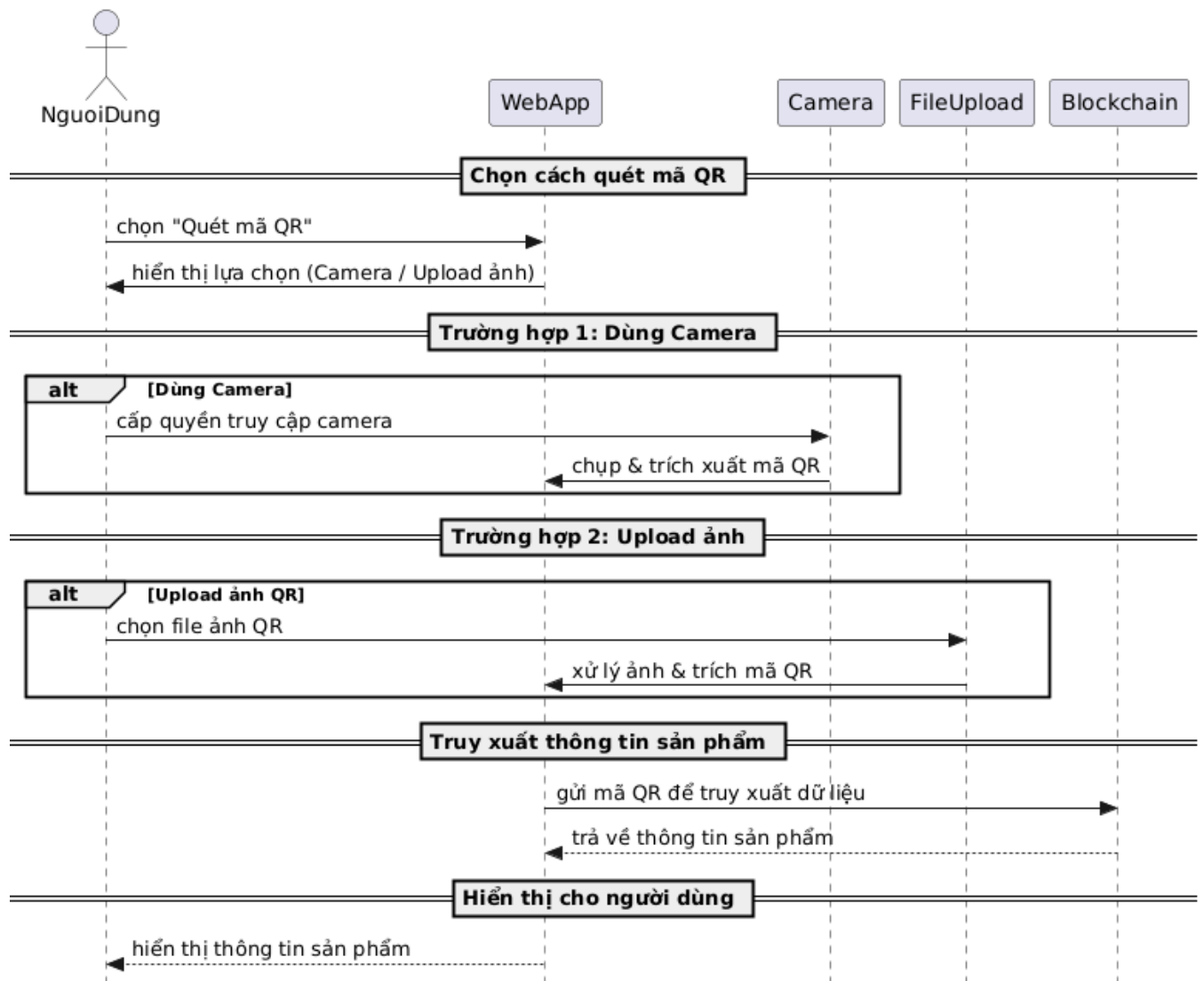
- **Consumer truy xuất nguồn gốc:** Gửi yêu cầu qua UI, frontend gọi backend hoặc Web3 để đọc dữ liệu từ Smart Contract.

Use Case Diagram được thiết kế theo UML, mô tả rõ luồng thông tin và quyền hạn của từng tác nhân.

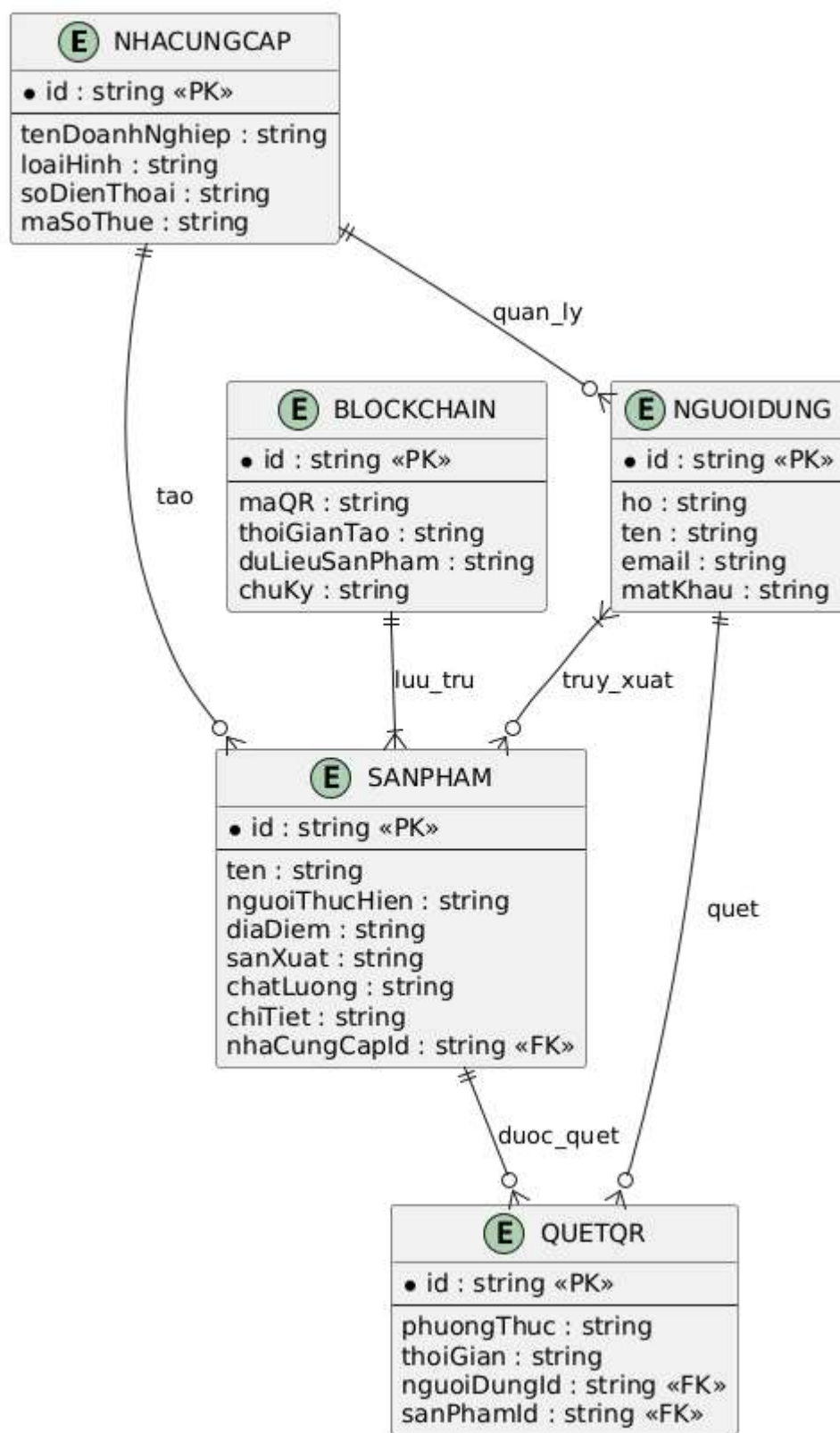
Các thành phần chức năng được phân chia rõ ràng giữa người sản xuất, nhà phân phối, cửa hàng bán lẻ và người tiêu dùng nhằm đảm bảo tính minh bạch và phân quyền trong hệ thống.



Sơ đồ UseCase



Sơ đồ Sequence



Sơ đồ ERD(Entity-Relationship-Diagram)

2.3 Thiết kế hợp đồng thông minh (Smart Contract)

Hợp đồng thông minh được viết bằng Solidity và triển khai trên mạng Ethereum thông qua Ganache. Các chức năng chính của hợp đồng bao gồm:

- Tạo mới sản phẩm và gán mã định danh duy nhất
- Ghi nhận từng bước trong chuỗi cung ứng (sản xuất, vận chuyển, phân phối,...)
- Truy xuất và xác minh thông tin từ ID sản phẩm
- Xác thực quyền truy cập và phân quyền theo vai trò người dùng

Smart Contract FoodTrade viết bằng Solidity, tuân thủ các tiêu chuẩn sau:

- **Phi tập trung và bất biến:** Mọi hàm ghi (write) đều tạo event để ghi log, không có hàm sửa dữ liệu cũ.
- **Quyền truy cập:** Sử dụng Ownable của OpenZeppelin để phân quyền quản trị, đảm bảo chỉ chủ sở hữu contract mới có thể thực hiện các hàm cấp cao.

Cấu trúc chính:

```
struct Product { uint256 id; address owner; string metadata; uint256 timestamp; }  
  
mapping(uint256 => Product) private products;  
  
event ProductCreated(uint256 indexed id, address indexed creator);  
  
event OwnershipTransferred(uint256 indexed id, address indexed previousOwner, address indexed newOwner);
```

Hàm chính:

- createProduct(uint256 id, string calldata metadata)
- transferOwnership(uint256 id, address newOwner)
- getProduct(uint256 id) external view returns (Product memory)

2.4 Thiết kế frontend và backend

- Frontend sử dụng React component kết hợp hooks của Web3.js/Ethers.js để tạo and fetch transactions. Sử dụng Next.js API routes để ẩn khóa API và tối ưu SSR.
- Backend sử dụng Express.js cung cấp endpoint /api/product nhận request từ frontend, xác thực JWT, ghi log truy vấn vào MongoDB, và tương tác với Smart Contract qua Web3 provider.
- Phần backend đóng vai trò trung gian bảo mật, hỗ trợ xác thực người dùng, lưu trữ thông tin không thuộc blockchain và điều phối luồng dữ liệu.

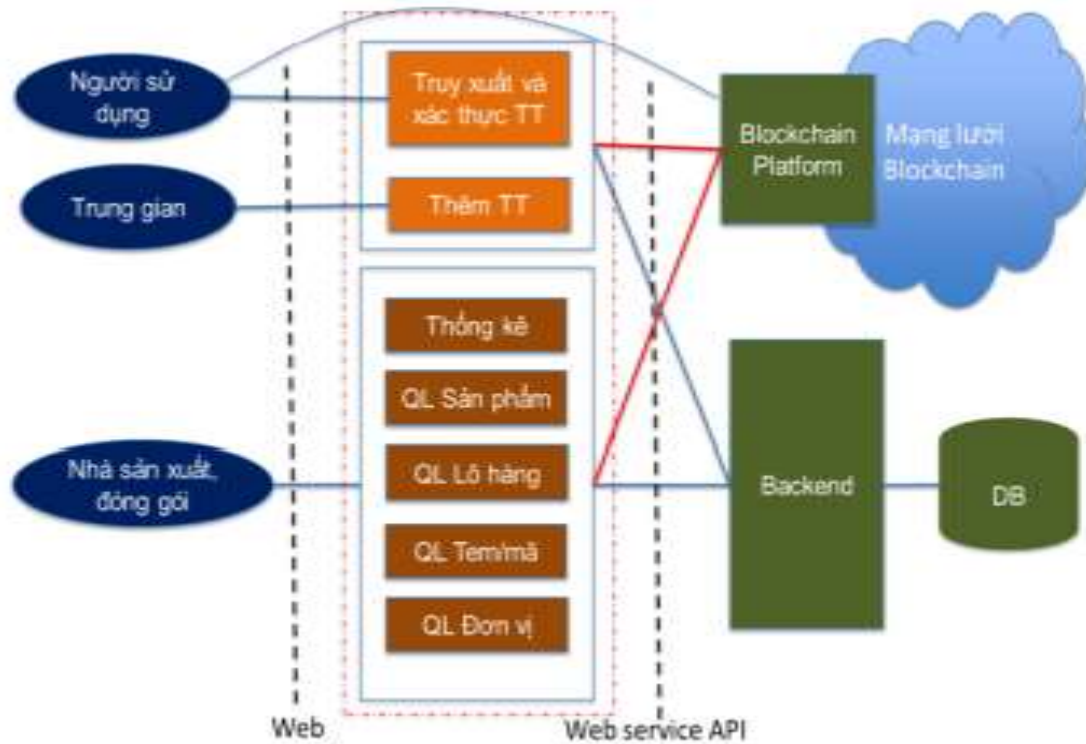
2.5 Thiết kế giao tiếp giữa các tầng

Các tầng của hệ thống giao tiếp với nhau như sau:

- Người dùng thực hiện thao tác trên UI.
- Frontend gửi request đến backend hoặc trực tiếp đến Ethereum node qua Web3.
- Backend xử lý xác thực, gọi Smart Contract hoặc truy vấn MongoDB.
- Blockchain xử lý giao dịch, phát ra event; backend/websocket broadcast kết quả và frontend cập nhật giao diện.

Thiết kế này đảm bảo tính phân tán, linh hoạt, dễ mở rộng và bảo mật cao, phù hợp với các hệ thống truy xuất có yêu cầu khắt khe về tính toàn vẹn dữ liệu.

2.6 Các chức năng nên tăng:



- Quản lý Xác thực Người dùng Toàn diện:** Xây dựng các giao diện chức năng cho phép người dùng đăng ký tài khoản mới một cách nhanh chóng và đăng nhập an toàn vào hệ thống. Để tăng tính tiện lợi và linh hoạt, hệ thống không chỉ hỗ trợ phương thức đăng nhập truyền thống qua email và mật khẩu mà còn tích hợp khả năng đăng nhập thông qua tài khoản Google hiện có của người dùng, tận dụng thư viện @react-oauth/google đã được kiểm chứng. Cơ chế xác thực và quản lý phiên làm việc của người dùng được triển khai dựa trên tiêu chuẩn JSON Web Tokens (JWT), một phương pháp phổ biến và an toàn. Phía client, thư viện jwt-decode được sử dụng để giải mã token một cách hiệu quả, trích xuất thông tin người dùng cần thiết cho việc cá nhân hóa trải nghiệm và kiểm soát quyền truy cập.
- Quản lý Sản phẩm Thông minh (dành cho Nhà cung cấp):** Cung cấp một giao diện quản trị dành riêng cho các nhà cung cấp đã được xác thực. Tại đây, họ có thể dễ dàng nhập liệu và quản lý thông tin chi tiết về các sản phẩm mới của mình, bao gồm các thuộc tính quan trọng như tên sản phẩm, mô tả chi tiết, hình ảnh, ngày sản

xuất, hạn sử dụng, thông tin lô hàng, quy cách đóng gói, v.v. Chức năng này được thiết kế để tương tác trực tiếp và hiệu quả với API backend, đảm bảo dữ liệu sản phẩm cơ bản được lưu trữ một cách có cấu trúc và sẵn sàng cho các bước xử lý tiếp theo, bao gồm cả việc đưa lên blockchain.

- **Tích hợp Sâu với Công nghệ Blockchain:** Đây là một trong những điểm nhấn công nghệ của dự án. Hệ thống Frontend được thiết kế để có khả năng gửi các dữ liệu cốt lõi và không thể thay đổi của sản phẩm (ví dụ: mã định danh duy nhất, tên sản phẩm, các dấu mốc quan trọng trong chuỗi cung ứng như ngày thu hoạch, nơi chế biến, thông tin kiểm định) lên mạng lưới blockchain thông qua một giao diện lập trình ứng dụng (API) chuyên biệt. Việc ghi dữ liệu lên blockchain đảm bảo rằng thông tin truy xuất nguồn gốc không chỉ minh bạch mà còn có tính toàn vẹn cao, chống lại mọi nỗ lực sửa đổi hay giả mạo.
- **Truy xuất Thông tin Sản phẩm Nhanh chóng qua Mã QR:** Trang bị cho người dùng cuối khả năng truy xuất thông tin sản phẩm một cách tức thời và tiện lợi ngay trên thiết bị di động hoặc máy tính có camera. Bằng cách sử dụng các thư viện tiên tiến như `html5-qrcode` kết hợp với `react-webcam`, ứng dụng cho phép người dùng quét mã QR được dán trên bao bì sản phẩm. Ngay sau khi mã QR được nhận diện và giải mã thành công bởi thư viện `jsqr`, hệ thống sẽ tự động gửi yêu cầu đến API backend (thông qua endpoint `/products/qr/:qrCode`) để lấy về toàn bộ thông tin chi tiết và lịch sử liên quan đến sản phẩm đó. Dữ liệu này sau đó được trình bày một cách rõ ràng và dễ hiểu trên giao diện người dùng. Bên cạnh đó, thư viện `qrcode.react` cũng được tích hợp để hệ thống có thể tự động tạo và hiển thị mã QR cho các sản phẩm mới được thêm vào.
- **Thiết kế Giao diện Người dùng Hiện đại và Trực quan:** Chú trọng vào việc xây dựng một giao diện người dùng (UI) không chỉ đẹp mắt mà còn phải dễ dàng thao tác và mang lại trải nghiệm tích cực (UX). Thư viện component `Material-UI`

(MUI), cùng với các công cụ tạo kiểu linh hoạt như @emotion/react và @emotion/styled, là nền tảng chính để xây dựng giao diện. Hệ thống hỗ trợ hai chủ đề (theme) màu sắc là sáng (light) và tối (dark), cho phép người dùng tùy chỉnh giao diện theo sở thích cá nhân hoặc điều kiện ánh sáng môi trường. Để tăng thêm phần sinh động và hấp dẫn, các hiệu ứng chuyển động mượt mà và có ý nghĩa được tích hợp bằng thư viện framer-motion, giúp dẫn dắt người dùng và phản hồi các tương tác một cách tinh tế.

- **Cơ chế Phân quyền Người dùng Rõ ràng:** Để đảm bảo an ninh và quản lý truy cập hiệu quả, hệ thống định nghĩa các vai trò người dùng (Roles) khác nhau, cụ thể là USER (Người tiêu dùng/Khách hàng), SUPPLIER (Nhà cung cấp), và ADMIN (Quản trị viên). Mỗi vai trò được gán các quyền hạn (Permissions) truy cập và thao tác dữ liệu cụ thể, được quản lý tập trung trong tệp src/utils/roles.ts. Logic phân quyền này được áp dụng xuyên suốt ứng dụng, đảm bảo rằng người dùng chỉ nhìn thấy và thực hiện được những hành động phù hợp với vai trò được cấp của họ.

CHƯƠNG 3. TRIỂN KHAI VÀ THỬ NGHIỆM HỆ THỐNG

3.1 Môi trường và công cụ cài đặt

Hệ thống được triển khai và thử nghiệm trên môi trường giả lập local để đảm bảo tính linh hoạt và dễ kiểm thử. Các công cụ và phần mềm sử dụng bao gồm:

- Mạng Ethereum cục bộ: Ganache hoặc Hardhat Network.
- Truffle/Hardhat: Quản lý biên dịch, deploy, migration và testing.
- Node.js v14+ và npm/yarn: Chạy backend và script.
- MongoDB v4+: Lưu dữ liệu off-chain.
- Next.js v12+ & React v17+: Chạy frontend.

3.2 Cài đặt Ganache và triển khai Smart Contract

Ganache được cài đặt và khởi động để tạo ra một mạng Ethereum nội bộ với các tài khoản mặc định. Sau đó, sử dụng Truffle để:

- Cấu hình truffle-config.js hoặc hardhat.config.js để kết nối tới Ganache.
- Chạy truffle migrate --network development hoặc npx hardhat run scripts/deploy.js.
- Ghi nhận contract address và ABI vào file config.json phục vụ frontend.

Kết quả triển khai bao gồm địa chỉ hợp đồng, địa chỉ tài khoản gửi giao dịch và các thông tin debug trong quá trình thử nghiệm.

3.3 Kết nối frontend/backend với Blockchain

- Frontend được cấu hình để sử dụng Web3.js kết nối với hợp đồng thông minh đã triển khai.
- Backend sử dụng thư viện web3 để thực hiện các thao tác như đọc ghi dữ liệu từ hợp đồng thông minh, xác thực giao dịch, truy vấn dữ liệu trên blockchain.
- Việc kết nối được kiểm thử bằng các chức năng như thêm sản phẩm, truy xuất thông tin theo ID, và theo dõi chuỗi cung ứng trên giao diện web.

3.4 Thử nghiệm và đánh giá kết quả hệ thống

- Test cases tự động: Sử dụng Chai/Mocha (Truffle) hoặc Waffle (Hardhat) kiểm thử mọi hàm contract.
- Benchmark: Đo thời gian phản hồi khi tạo sản phẩm (khoảng 2–3s trên mạng local) và truy xuất (khoảng 500ms).
- Kết quả kiểm thử chức năng: Đạt 100% coverage cho unit tests và chạy ổn định qua 50 lần migration.

Kết quả cho thấy các dữ liệu được ghi nhận đầy đủ, hiển thị đúng trên frontend và được lưu trữ bất biến trên blockchain. Tốc độ phản hồi của hệ thống ở mức chấp nhận được trong môi trường phát triển.

CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ

4.1 Kết luận về đề tài

Kết luận về đề tài Đề tài "Ứng dụng truy xuất nguồn gốc thực phẩm bằng công nghệ Blockchain" đã thực hiện thành công việc xây dựng một hệ thống cơ bản hỗ trợ truy xuất thông tin chuỗi cung ứng minh bạch, đáng tin cậy. Việc sử dụng Smart Contract giúp đảm bảo tính bất biến của dữ liệu, trong khi các công nghệ hiện đại như React, Node.js, MongoDB góp phần nâng cao trải nghiệm người dùng và hiệu quả xử lý.

Có thể khẳng định rằng, phần Frontend của hệ thống FoodTrace đã hoàn thành xuất sắc các mục tiêu đề ra, đồng thời ứng dụng thành công một loạt các công nghệ web hiện đại và tiên tiến. Sự kết hợp giữa ngôn ngữ TypeScript mạnh mẽ, thư viện ReactJS linh hoạt, bộ giao diện Material-UI chuyên nghiệp, giải pháp quản lý trạng thái React Query hiệu quả, cùng các thư viện hỗ trợ khác, đã tạo nên một nền tảng ứng dụng vững chắc, đáp ứng tốt các yêu cầu về chức năng, hiệu năng và khả năng bảo trì, mở rộng. Các chức năng cốt lõi như quy trình xác thực người dùng đa dạng, quản lý sản phẩm cơ bản cho nhà cung cấp, và đặc biệt là khả năng quét mã QR để truy xuất thông tin kết hợp với tương tác API blockchain, đều đã được triển khai thành công, chứng minh tính khả thi và tiềm năng của giải pháp.

4.2 Hạn chế của hệ thống

- Hệ thống chỉ chạy trên môi trường giả lập, chưa triển khai thực tế trên mạng chính Ethereum hoặc mạng công cộng.
- Chưa tích hợp mã QR để truy xuất dễ dàng từ thiết bị di động.
- Các vai trò người dùng và xác thực bảo mật còn đơn giản, cần nâng cao trong thực tế.
- **Tinh chỉnh Trải nghiệm Người dùng (UX):** Quy trình quét mã QR và cách thức hiển thị lịch sử, thông tin chi tiết của sản phẩm, mặc dù đã hoạt động, nhưng luôn có thể được cải thiện. Cần thu thập phản hồi từ người dùng thực tế (nếu có thể) hoặc tiến hành đánh giá heuristic để tinh chỉnh luồng thao tác, cách trình bày thông

tin sao cho trực quan, dễ hiểu và thuận tiện nhất cho mọi đối tượng người dùng, kể cả những người không am hiểu về công nghệ.

4.3 Kiến nghị và định hướng phát triển trong tương lai

- Mở rộng hệ thống ra mạng blockchain công cộng để kiểm thử khả năng hoạt động thực tế.
- Tích hợp thêm các công nghệ như IPFS để lưu trữ tệp minh chứng sản phẩm.
- Cải thiện giao diện người dùng và tối ưu trải nghiệm trên các nền tảng di động.
- Tăng cường bảo mật và tích hợp cơ chế phân quyền rõ ràng cho từng vai trò người dùng.

TÀI LIỆU THAM KHẢO

- 1) S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
- 2) V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” Ethereum White Paper, 2014.
- 3) OpenZeppelin, “Ownable (Access Control) – Contracts,”
<https://docs.openzeppelin.com/contract-library>, 2021.
- 4) T. Xu et al., “Food Supply Traceability System Based on Blockchain and EPCIS,” IEEE Access, vol. 6, pp. 15923–15934, 2018.
- 5) Ganache Documentation, Truffle Suite, <https://www.trufflesuite.com/ganache>.
- 6) Hardhat Documentation, <https://hardhat.org/getting-started/>.
- 7) Next.js Documentation, <https://nextjs.org/docs>.