

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



**NIÊN LUẬN CHUYÊN NGÀNH ĐẠI HỌC NGÀNH**  
**CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**PHÁT TRIỂN ỨNG DỤNG**  
**TRUY XUẤT NGUỒN GỐC SẢN PHẨM NÔNG NGHIỆP**

**Cán bộ hướng dẫn**  
**TS. Trương Quốc Định**

**Sinh viên thực hiện:**  
**Họ và tên:**  
**Nguyễn Anh Bằng – B1606777**  
**Nguyễn Chí Linh – B1606906**  
**Khóa: 42**

## TÓM TẮT

Hiện nay, chúng ta đã và đang bước vào cuộc Cách mạng Công Nghiệp 4.0, mà Blockchain là chủ đề rất được quan tâm trên toàn cầu hiện nay bởi các lợi ích cũng như sự thuận tiện của nó vào đời sống, Blockchain được biết đến với các khái niệm như Bitcoin và tiền kỹ thuật số trở thành đề tài bàn luận của tất cả mọi người. Mọi người thường biết đến Blockchain trong các lĩnh vực ngân hàng, tiền tệ. Không dừng ở việc ứng dụng vào tiền kỹ thuật số nó còn được ứng dụng trong nhiều lĩnh vực như giáo dục, kinh tế, y tế ...

Trong khuôn khổ đề tài niên luận, chúng tôi xây dựng hệ thống truy xuất nguồn gốc các loại cây ăn trái có khả năng truy xuất quy trình sản xuất nông sản từ khâu trồng trọt đến khâu đóng gói trước khi đến tay người tiêu dùng. Để thực hiện các chức năng trên chúng tôi đã sử dụng nền tảng Ethereum Blockchain để thiết lập các hợp đồng thông minh (smart contract) nhằm thực hiện các quy trình trong truy xuất các sản phẩm cây ăn trái.

Ở đề tài này hệ thống còn có khả năng truy xuất nguồn gốc sản phẩm với mục đích bảo đảm sự an toàn cho người tiêu dùng thông qua nguồn gốc rõ ràng của nông sản.

# MỤC LỤC

|   |    |
|---|----|
| TÓM TẮT .....   | 2  |
| PHẦN GIỚI THIỆU .....                                     | 1  |
| 1. Đặt vấn đề.....  | 1  |
| 2. Mục tiêu đề tài.....                                   | 1  |
| 3. Đối tượng và phạm vi nghiên cứu .....                  | 2  |
| 4. Nội dung nghiên cứu.....                               | 2  |
| PHẦN NỘI DUNG .....                                       | 3  |
| CHƯƠNG 1. GIỚI THIỆU CÔNG NGHỆ BLOCKCHAIN.....            | 3  |
| 1. Sử dụng Blockchain Ethereum để giải quyết vấn đề ..... | 3  |
| 1.1. Blockchain .....                                     | 3  |
| 1.1.1. Giới thiệu.....                                    | 3  |
| 1.1.2. Nền tảng lý thuyết.....                            | 3  |
| 1.1.3. <i>Hàm băm</i> .....                               | 3  |
| 1.1.4. <i>Tính chất của Blockchain</i> .....              | 3  |
| 1.2. Ethereum .....                                       | 3  |
| 1.2.1. Giới thiệu.....                                    | 3  |
| 1.2.2. Cơ cấu.....  | 3  |
| 1.3. MetaMask .....                                       | 3  |
| 1.4. HTML, CSS, JavaScript và Solidity.....               | 3  |
| 1.4.1. HTML, CSS và JavaScript.....                       | 3  |
| 1.4.2. Solidity.....                                      | 4  |
| 2. Kết luận .....   | 5  |
| CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT.....                        | 6  |
| 1. Thiết kế và phân tích hệ thống .....                   | 6  |
| 2. Cài đặt hệ thống.....                                  | 12 |
| 2.1 Server .....  | 12 |
| 2.2 Client .....  | 14 |
| 3. Hướng dẫn cài đặt hệ thống.....                        | 14 |
| PHẦN KẾT LUẬN .....                                       | 17 |
| TÀI LIỆU THAM KHẢO.....                                   | 19 |

## **PHẦN GIỚI THIỆU**

### **1. Đặt vấn đề**

An toàn thực phẩm, Trá cây tẩm thuốc, và còn rất nhiều tin tức về nguồn gốc thực phẩm gây hại trực tiếp đến sức khỏe của con người. Người sử dụng ngày càng mất niềm tin vào thực phẩm họ sử dụng và ngày càng khủng hoảng với các thông tin về cơ sở sản xuất, thực phẩm ngâm tẩm hóa chất, kinh doanh thực phẩm bẩn, cũng như thực phẩm nhập lậu chưa được kiểm duyệt từ các cơ quan nhà nước có thẩm quyền.

Với thế mạnh của Blockchain là minh bạch thông tin, độ tin cậy cao của nó rất phù hợp để giải quyết vấn đề được nêu trên, nó ghi lại những hoạt động, giao dịch, các giấy tờ công chứng, chứng thực danh tính và chứng minh nguồn gốc của các sản phẩm. Điều này giúp cho các sản phẩm trong hệ thống Blockchain được tin cậy cao với các chất lượng của các sản phẩm. Blockchain còn lưu lại mọi sự tương tác cũng như các giao dịch dù là nhỏ nhất trong suốt các giai đoạn cho đến khi đến tay người tiêu dùng. Với Blockchain, dữ liệu không thể bị thay đổi hoặc đánh cắp khi đã được xác thực và lưu lại. Đó là nhờ vào cơ chế đồng thuận cũng như các cách mã hóa dữ liệu trong Blockchain.

Người dùng sẽ trở nên tin tưởng với những thông tin chính xác và minh bạch từ các thông tin được biểu hiện, với những hình ảnh thông tin cụ thể sản phẩm họ tìm hiểu, từ đó tăng năng suất và sản lượng giao dịch nhờ vào Blockchain.

### **2. Mục tiêu đề tài**

Tạo ra một hệ thống truy xuất nguồn gốc sản phẩm nông nghiệp có thể quản lý quy trình sản xuất nông sản từ giai đoạn trồng trọt đến giai đoạn đóng gói trước khi đến tay người tiêu dùng. Bên cạnh đó, hệ thống còn có khả năng truy xuất nguồn gốc sản phẩm với mục đích bảo đảm sự an toàn, cung cấp thông tin tin cậy cho người tiêu dùng thông qua nguồn gốc rõ ràng của nông sản thông qua mô hình Client – Server:

- Áp dụng công nghệ có khả năng xử lý trực tuyến, dữ liệu phân tán.
- Phần mềm chạy trên nền tảng website và mobile với mô hình Client – Server.
- Hệ thống Server là nền tảng CSDL Ethereum Blockchain.

- Các máy trạm là các thiết bị mạng có thể truy cập Internet và sử dụng các trình duyệt có hỗ trợ plugin MetaMask.
- Các chức năng quản trị.
- Xây dựng phần mềm trên nền tảng mobile android để truy xuất sản phẩm

Tìm hiểu các hình thức truy xuất nguồn gốc hiện nay gồm có: truyền thống bằng cách đóng dấu, điện tử, điện tử và blockchain.

### **3. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của đề tài:

- Blockchain
- Ethereum framework
- Web3.js framework
- Metamask plugin
- Thư viện Webj cho android

Phạm vi nghiên cứu của đề tài giới hạn trong các lĩnh vực nông nghiệp của một công ty, tổ chức cá nhân bất kỳ chuyên sản xuất, xuất khẩu và buôn bán nông sản.

### **4. Nội dung nghiên cứu**

Xây dựng hệ thống quản lý chuỗi cung ứng nông sản:

- Server: Thiết kế mạng Blockchain riêng sử dụng nền tảng Ethereum Blockchain, kết nối với mạng Rinkeby test net, tạo các block cơ bản.
- Client: Thực hiện kết nối vào mạng Blockchain đã được thiết kế, truy xuất thông qua quét mã qr trên thiết bị di động, tạo các giao dịch quản lý người dùng và sản phẩm.

**PHẦN NỘI DUNG**  
**CHƯƠNG 1. GIỚI THIỆU**  
**CÔNG NGHỆ BLOCKCHAIN**

**1. Sử dụng Blockchain Ethereum để giải quyết vấn đề**

**1.1. Blockchain**

**1.2. Ethereum**

**1.2.1. Giới thiệu**

**1.2.2. Cơ cấu**

**1.3. MetaMask**

MetaMask là một plugin, dùng để truy cập vào các mạng phân tán như BlockChain và không cần khởi động toàn bộ node Ethereum dẫn đến trình duyệt trở nên nặng nề và chậm chạp.

MetaMask bao gồm một mã nhận diện mã hóa vô cùng an toàn, cung cấp đến người dùng một giao diện để quản lý các thông tin cá nhân của bạn trên các trang web khác nhau và chấp nhận các giao dịch blockchain trên toàn hệ thống.

- Đầy đủ tính năng của ví Ethereum, Erc20
- Chống ăn cắp key đăng nhập
- Mở khoá MyEtherWallet.com nhanh gọn, bấm vào là mở
- Khi lỡ vào các web giả mạo, nó sẽ hiện thông báo đỏ

**1.4. HTML, CSS, JavaScript và Solidity**

**1.4.1. HTML, CSS và JavaScript**

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu, được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

CSS (Cascading Style Sheets) là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Nếu HTML đóng vai trò định dạng các phần tử trên

website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,...thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,...rất nhiều.

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. Là một ngôn ngữ nhỏ và nhẹ. Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

#### 1.4.2. Solidity

Solidity là ngôn ngữ lập trình trên Blockchain của mạng lưới Ethereum (ETH là đồng coin cơ bản nhất để mạng lưới Ethereum hoạt động và đang đứng 2 trên thị trường hiện nay chỉ xếp sau Bitcoin)

Ethereum và Solidity là Platform và ngôn ngữ lập trình phổ biến.

Solidity là ngôn ngữ lập trình hướng contract, được sử dụng để viết smart contract trong hệ sinh thái Ethereum.

Có 4 mức visibility với phương thức: internal, external, public và private. Mặc định, phương thức sẽ có mức visibility là public. Mô tả khái quát 4 mức này như sau:

- **Internal:** chỉ có thể truy cập từ các phương thức bên trong contract hoặc từ contract con (kế thừa) (internal calling).
- **External:** là 1 phần của giao diện của contract, do đó có thể được truy cập từ contract khác. Bản thân contract chứa phương thức cũng có thể gọi nó bằng cách sử dụng từ khóa this. Ví dụ ta có thể gọi phương thức được khai báo function extFunc() external bằng cách this.extFunc() (external calling).
- **Public:** là 1 phần của giao diện của contract, có thể được gọi từ trong contract (mà không cần từ khóa this) hoặc từ contract khác (internal calling/external calling).
- **Private:** chỉ có thể truy cập từ các phương thức bên trong contract (internal calling).

## **2. Kết luận**

Blockchain là công nghệ lưu trữ thông tin vào các khối và được liên kết với nhau. Mỗi khối chứa đựng các thông tin về thời gian khởi tạo và được liên kết với các khối trước đó.

Blockchain được thiết kế để chống lại sự thay đổi dữ liệu. Thông tin trong Blockchain không thể bị thay đổi và chỉ được bổ sung thêm khi có sự đồng thuận của tất cả các nút trong hệ thống. Khi một nút bị hỏng thì nút khác sẽ vẫn tiếp tục hoạt động để bảo vệ thông tin. Với blockchain, dữ liệu một khi đã được xác thực và lưu lại sẽ không thể thay đổi hay đánh cắp, ngăn chặn được tình trạng sửa đổi dữ liệu vì mục đích cá nhân.

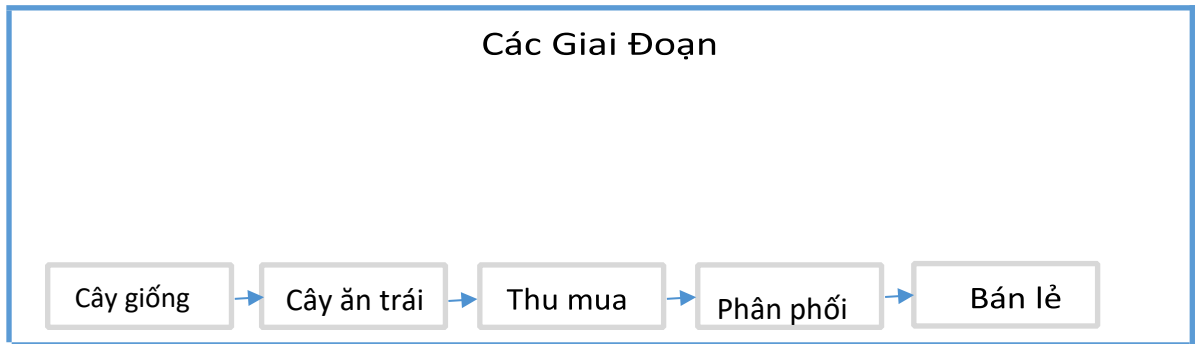
Công nghệ Blockchain được phân cấp, mở rộng quyền truy cập cho tất cả mọi người, mã hóa, có khả năng lưu trữ một lượng lớn thông tin tự động và đồng bộ. Những ưu điểm này khuyến khích sự tin tưởng giữa các đối tác vốn hay nghi ngờ vào chuỗi cung ứng. Càng nhiều thông tin đáng tin cậy được chia sẻ, tính minh mạch trong chuỗi cung ứng sẽ càng cao. Đó là khởi đầu của một chuỗi cung ứng hiệu quả.



## CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT

### 1. Thiết kế và phân tích hệ thống

Hệ thống phục vụ cho việc cập nhập cũng như truy xuất nguồn gốc của sản phẩm cho đơn vị, tổ chức hoặc doanh nghiệp xuất khẩu nông sản. Trong chuỗi cung ứng nông sản thường trải qua 5 giai đoạn chính: cây giống, cây ăn trái, thu mua, phân phối, bán lẻ.



Hình 4. Mô hình hệ thống

Do có nhiều giai đoạn trong chuỗi cung ứng nên việc quản lý sẽ trở nên khó khăn. Để dễ dàng quản lý cũng như cung cấp thông tin đầy đủ và chính xác cho từng giai đoạn ta sẽ phân quyền cho từng giai đoạn. Tức là ở mỗi giai đoạn sẽ có một người đảm nhiệm và chức năng của mỗi giai đoạn là khác nhau. Các chức năng tương ứng:

- Người quản lý giai đoạn cây giống
- Người quản lý giai đoạn cây ăn trái
- Người quản lý giai đoạn thu mua
- Người quản lý giai đoạn phân phối
- Người quản lý giai đoạn thu mua

Ở mỗi công đoạn khi được xác nhận và tạo giao dịch để lưu thông tin vào Block đều sinh ra một Barcode để từ đó truy xuất thông tin từ công đoạn đầu tiên đến công đoạn hiện tại của một lô hàng. Để bắt đầu chuỗi cung ứng, người quản trị cần phải tạo và cung cấp tài khoản cho từng người dùng đảm nhiệm từng giai đoạn trong chuỗi.

#### Quản trị hệ thống

Như đã mô tả, để bắt đầu chuỗi cung ứng thì người quản trị cần phải tạo và cung cấp tài khoản cho từng người dùng đảm nhiệm từng

giai đoạn trong chuỗi (tạo tài khoản cho mỗi Role khác nhau). Tiếp theo, bắt đầu tạo hợp đồng (Smart Contract), Tạo lô hàng (batch) mới và nó chính là lô khởi tạo trong chuỗi cung ứng (init block). Ở đây, ta sẽ khai báo các thông của các User đảm nhiệm ở từng công đoạn để giới hạn phân quyền dựa trên lô hàng.

Người quản trị có thể xem tình trạng của bất cứ lô hàng nào. Truy xuất thông tin nguồn gốc của lô hàng. Và hệ thống có thể xuất báo cáo (Report) về chuỗi hàng đã và đang hoàn tất có đính kèm mã vạch.

### Giai đoạn cây giống

Nguồn gốc sản phẩm cũng như một chuỗi cung ứng luôn bắt đầu từ quá trình trồng trọt của nông dân. Người quản lý giai đoạn cây giống đảm nhiệm ở giai đoạn này có nhiệm vụ ghi nhận và cung cấp đầy đủ thông tin về cây giống cho quản trị. Như trong giai đoạn cây giống, người nông dân đã sử dụng giống như thế nào? Sử dụng phân bón nào? Các thông tin này là cần thiết cho chuỗi cung ứng an toàn. Và đặc biệt là người nông dân cần cung cấp địa chỉ của nông trại của mình để tăng độ tin cậy của hệ thống. Hệ thống cho phép người nông dân có thể thấy được các hợp đồng (Smart Contract) mà họ đảm nhiệm. Có thể nắm được thông tin của lô hàng như lô hàng đang ở giai đoạn nào, các thông tin của từng giai đoạn đã được cập nhật thông tin.

- Use case

|                       |  |
|-----------------------|--|
| <b>Use Case</b>       | Giai đoạn cây giống                    |
| <b>Description</b>    | Cập nhật thông tin giai đoạn cây giống |
| <b>Users</b>          | Người quản lý                          |
| <b>Pre-condition</b>  | Tới giai đoạn cây ăn trái              |
| <b>Post-condition</b> | None                                   |
| <b>Trigger</b>        | None                                   |

*Bảng 1. UC Quản lý giai đoạn cây giống*

- Các bước thực hiện

| Steps | Description  |
|-------|--|
| S1    | Role Người quản lý giai đoạn cây giống login vào hệ thống              |
| S2    | Chọn lô hàng cần cập nhật thông tin                                    |
| S3    | Click vào button Updating để có thể thêm thông tin giai đoạn cây giống |
| S4    | Submit tạo 1 transaction   |

*Bảng 2. Step UC Quản lý giai đoạn cây giống*

### Giai đoạn cây ăn trái

Bộ phận , phân loại, cập nhật thêm thông tin về cây trồng, phân bón hay thuốc trừ sâu, số lượng cây được trồng, thời gian trồng đến khi cây có quả được duy trì trong quá trình này. Và đặc biệt là cần cung cấp địa chỉ của nông trại của mình để tăng độ tin cậy của hệ thống. Hệ thống cho phép người thu hoạch có thể thấy được các hợp đồng (Smart Contract) mà họ đảm nhiệm. Có thể nắm được thông tin của lô hàng như lô hàng đang ở giai đoạn nào, các thông tin của từng giai đoạn đã được cập nhật thông tin.

- Use case

| Use Case       | Giai đoạn cây ăn trái                              |
|----------------|--|
| Description    | Cập nhật thông tin cây trồng, cách bảo quản rau củ |
| Users          | Người trồng cây(chăm sóc cây).                     |
| Pre-condition  | Cây ăn trái và chuyển sang giai đoạn thu hoạch     |
| Post-condition | None   |
| Trigger        | None   |

*Bảng 3. UC Harvester*

- Các bước thực hiện

| Steps | Description  |
|-------|--|
| S1    | Role Người quản lý giai đoạn cây ăn trái login vào hệ thống  |
| S2    | Chọn lô hàng cần cập nhật thông tin                          |
| S3    | Click vào button Updating để có thể thêm thông tin cây trồng |
| S4    | Submit tạo 1 transaction                                     |

*Bảng 4. Step UC Harvester*

### Giai đoạn thu hoạch

Bộ phận , phân loại, cập nhật thêm thông tin về sản lượng , cơ sở thu mua, phương thức thu hoạch ,được duy trì trong quá trình này ,Hệ thống cho phép người xuất có thể thấy được các hợp đồng (Smart Contract) mà họ đảm nhiệm. Có thể nắm được thông tin của lô hàng như lô hàng đang ở giai đoạn nào, các thông tin của từng giai đoạn đã được cập nhật thông tin.

- Use case

|                       |                                       |
|-----------------------|---------------------------------------|
| <b>Use Case</b>       | Giai đoạn thu hoạch                   |
| <b>Description</b>    | Cập nhật thông tin sản phẩm thu hoạch |
| <b>Users</b>          | Người quản lý giai đoạn thu hoạch     |
| <b>Pre-condition</b>  | Giai đoạn phân phối                   |
| <b>Post-condition</b> | None                                  |
| <b>Trigger</b>        | None                                  |

*Bảng 5. UC Exporter*

- Các bước thực hiện

| Steps | Description   |
|-------|---|
| S1    | Role Người quản lý thu hoạch login vào hệ thống                 |
| S2    | Chọn lô hàng cần cập nhật thông tin                             |
| S3    | Click vào button Updating để có thể thêm thông tin lô hàng xuất |
| S4    | Submit tạo 1 transaction  |

*Bảng 6. Step UC Exporter*

### Giai đoạn phân phối

Các Importer là các tổ chức thu mua từ cơ sở thu mua. Cập nhật số lượng, số xe vận chuyển, thông tin vận chuyển, tên nhà kho, địa chỉ nhà kho và địa chỉ của cơ sở phân phối, cách thức bảo quản trong quá trình vận chuyển. Hệ thống cho phép người nhập hàng có thể thấy được các hợp đồng (Smart Contract) mà họ đảm nhiệm. Có thể nắm được thông tin của lô hàng như lô hàng đang ở giai đoạn nào, các thông tin của từng giai đoạn đã được cập nhật thông tin.

#### - Use case

|                       |                                 |
|-----------------------|---------------------------------|
| <b>Use Case</b>       | Giai đoạn phân phối             |
| <b>Description</b>    | Cập nhật thông tin lô hàng nhập |
| <b>Users</b>          | Người quản giai đoạn phân phối  |
| <b>Pre-condition</b>  | Giai đoạn bán lẻ                |
| <b>Post-condition</b> | None                            |
| <b>Trigger</b>        | None                            |

*Bảng 7. UC Importer*

#### - Các bước thực hiện

| Steps | Description |
|-------|-------------|
|-------|-------------|

|    |   |
|----|---|
| S1 | Role Người quản lý giai đoạn phân phối login vào hệ thống       |
| S2 | Chọn lô hàng cần cập nhật thông tin                             |
| S3 | Click vào button Updating để có thể thêm thông tin lô hàng nhập |
| S4 | Submit tạo 1 transaction  |

*Bảng 8. Step UC Importer*

### **Giai đoạn Bán lẻ**

Thêm thông tin như số lượng, số lô nội bộ, thời gian ngày đóng gói, tên và địa chỉ nhà máy sản xuất, các địa điểm bán lẻ đã được phân phối đến. Hệ thống cho phép người phụ trách xử lý có thể thấy được các hợp đồng (Smart Contract) mà họ đảm nhiệm. Có thể nắm được thông tin của lô hàng như lô hàng đang ở giai đoạn nào, các thông tin của từng giai đoạn đã được cập nhật thông tin.

- Use case

|                       |                                       |
|-----------------------|---------------------------------------|
| <b>Use Case</b>       | Giai đoạn bán lẻ                      |
| <b>Description</b>    | Cập nhật thông tin sản phẩm cuối cùng |
| <b>Users</b>          | Người bán lẻ                          |
| <b>Pre-condition</b>  | Quá trình bán lẻ hoàn tất             |
| <b>Post-condition</b> | None                                  |
| <b>Trigger</b>        | None                                  |

*Bảng 9. UC Processor*

- Các bước thực hiện

| <b>Steps</b> | <b>Description</b>                                     |
|--------------|--|
| S1           | Role Người quản lý giai đoạn bán lẻ login vào hệ thống |
| S2           | Chọn lô hàng cần cập nhật thông tin                    |

|    |   |
|----|---|
| S3 | Click vào button Updating để có thể thêm thông tin giai đoạn bán lẻ |
| S4 | Submit tạo 1 transaction  |

*Bảng 10. Step UC Processor*

## 2. Cài đặt hệ thống

### 2.1 Server

Để khởi tạo 1 mạng Blockchain ta cần:

- Nodejs v9.10 trở lên
- Truffle v4.1.8
- Solidity v0.4.23
- Metamask

Để tạo được một smart contract cần có kiến thức về ngôn ngữ lập trình Solidity, để hiểu rõ hơn về Solidity hãy xét qua một ví dụ dưới đây:

```
pragma solidity ^0.4.0;
contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

- Dòng đầu tiên “pragma solidity ^0.4.0;” cho chúng ta biết rằng code này được viết trên Solidity phiên bản 0.4.0, từ khóa pragma để khai báo version đang được sử dụng của Solidity.
- Một hợp đồng (**contract**) là những dòng bao gồm tập hợp các code (là các chức năng của hợp đồng) và dữ liệu (là các trạng thái của hợp đồng) nằm ở cụ thể một địa chỉ trên Ethereum blockchain. Dòng uint storedData; khai báo một biến trạng thái được gọi là storedData của kiểu uint([unsigned integer of 256 bit](#)). Chúng ta có thể hiểu đơn giản là nó là một record trong database thông thường, có thể query được, thay đổi được và

được quản lý bởi database. Trong trường hợp của Ethereum, đây luôn luôn là những hợp đồng sở hữu. Và trong trường hợp này, các function set và get được sử dụng để sửa dữ liệu, hay nhận giá trị từ các biến.

Qua ví dụ trên phần nào giúp ta hiểu được về ngôn ngữ lập trình Solidity và smart contract là gì?

Hệ thống dùng ngôn ngữ Solidity để viết các smart contract, mỗi smart contract có một chức năng riêng như:

- “User”, “Sanphamnongnghiep” contract dùng để kiểm soát các quyền chức năng của hệ thống, điều này giúp đơn giản hóa việc quản lý người dùng.
  - User contract và dùng để:
    - Khởi tạo cấu trúc của user.
    - Các hàm get() và set() thông tin của user.
    - Cập nhật thông tin user đã có trong hệ thống.
  - “Sanphamnongnghiep” contract có chứa các hàm chức năng dùng để:
    - Khởi tạo cấu trúc của sản phẩm.
    - Khởi tạo cấu trúc các giai đoạn :
    - cây giống → cây ăn trái → thu hoạch → phân phối → bán lẻ
      - Hàm get() thông tin sản phẩm ban đầu dùng trong truy xuất nguồn gốc sản phẩm phẩm.
      - Thêm mới sản phẩm vào hệ thống.
      - Cập nhật thông tin quy trình trồng trọt vào hệ thống.
      - Hàm get() thông tin quy trình trồng trọt.
      - Cập nhật thông tin quy trình thu hoạch vào hệ thống.
      - Hàm get() thông tin quy trình thu hoạch.
      - Cập nhật thông tin quy trình thu hoạch vào hệ thống.
      - Hàm get() thông tin quy trình thu hoạch .
      - Cập nhật thông tin giai đoạn phân phối hệ thống.
      - Hàm get() thông tin giai đoạn phân phối .
      - Cập nhật thông tin quy trình bán lẻ vào hệ thống.
      - Hàm get() thông tin quy trình bán lẻ.

Muốn các smart contract hoạt động thì không thể thiếu Migrations contract, Migrations contract được tự động tạo ra khi xây dựng một project, nó được gọi đến đầu tiên khi muốn triển khai server..



## 2.2 Client

Về phía client sử dụng các thư viện và các nền tảng như sau:

- Trên nền tảng website
  - Web3.js: Thư viện này có thể được sử dụng để xây dựng các DApps nền web.
  - Metamask: Đối với nhiều người muốn chạy DApps nhưng không muốn chạy một Ethereum node client, Metamask là một giải pháp hợp lý. Metamask cho phép kết nối với Ethereum network ngay trên trình duyệt thông qua kết nối với các Ethereum node của Metamask. Hiện tại, Metamask được phát triển dưới dạng extension cho Chrome và Firefox.
  - Etherscan.io: Hệ thống sử dụng website có sẵn etherscan.io để truy xuất thông tin một giao dịch để tăng tính xác thực của hệ thống.
- Trên nền tảng android [4]
  - Webj trên nền tảng android là thư viện cho phép bạn làm việc giữa ngôn ngữ Java và Ethereum Blockchain.
  - Infura cho phép chúng ta gửi và giao tiếp các giao dịch giữa các nút nếu như chúng ta có blockchain tải về trên máy tính địa phương của chúng ta.

## 3. Hướng dẫn cài đặt hệ thống

### 1. Website

Để có thể biên dịch được mã lệnh từ Solidity chúng ta cần có trình biên dịch, để cài đặt trình biên dịch sử dụng lệnh như sau:

- Đối với hệ điều hành Windows “npm install -g solc”.
- Đối với hệ điều hành Ubuntu “sudo apt-get install solc”.

Tiếp theo, cần cài đặt Truffle sử dụng lệnh như sau đối với cả hệ điều hành Windows và Ubuntu “npm install -g truffle”.

Nếu như muốn triển khai một mạng Blockchain được công khai triển khai qua các mạng test net như rinkeby, ropsten, kovan ta cần có tài khoản ethereum có số dư khả dụng trong tài khoản, để có thể kiểm tiền cho tài khoản, bạn có thể lên trang “<https://faucet.rinkeby.io/>” và xin ủng hộ, để xin ủng hộ rất đơn giản, bạn chỉ cần đăng một dòng trạng thái trên facebook hoặc twitter của mình có chứa tài khoản ethereum cần xin ủng hộ, sau đó copy và dán đường link chứa

dòng trạng thái đó và làm theo hướng dẫn sẽ nhận được ủng hộ. Tiền trong tài khoản ethereum xin ủng hộ không thể sử dụng ở mạng ethereum chính thức được và chỉ sử dụng trong môi trường test.

- Để sử dụng được các mạng test net, ta cần cài đặt thư viện `trufflehdwallet-provider`, để cài đặt thư viện sử dụng lệnh “`npm install truffle-hdwallet-provider`”.
- Mở tập tin `truffle-config.js` và điều chỉnh các thông tin như sau:

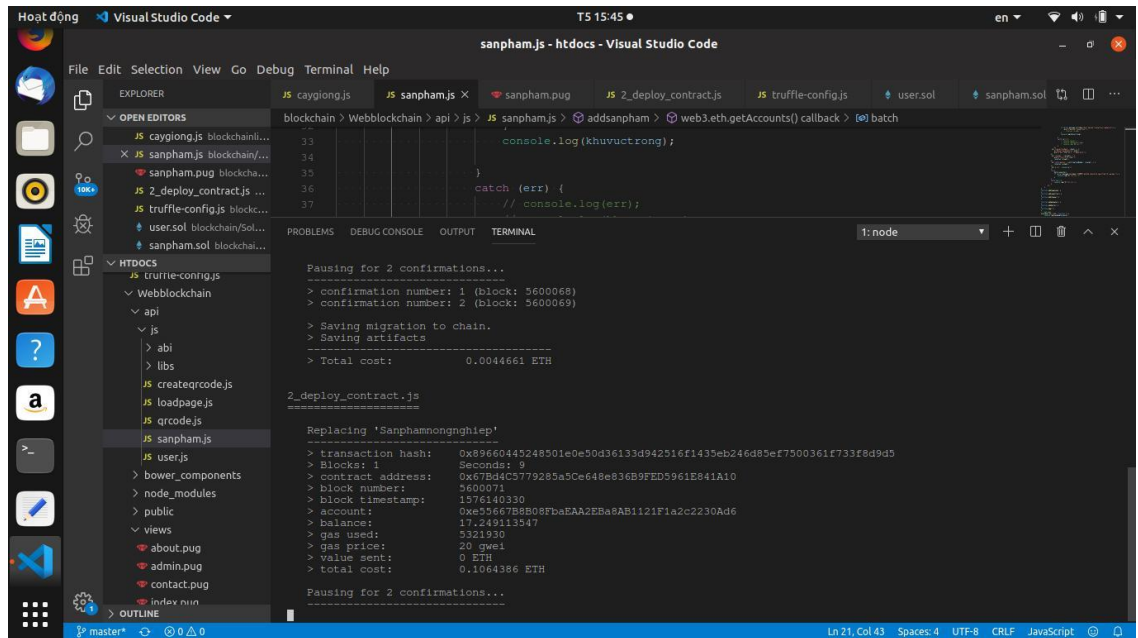
```
const HDWalletProvider = require('truffle-hdwallet-provider');
// const infuraKey = "fj4jll3k...";
//
// const fs = require('fs');
const mnemonic = "mnemonic";

module.exports = {
  /**
```

```
  // Useful for deploying to a public network.
  // NB: It's important to wrap the provider as a function.
  rinkeby: {
    provider: () => new HDWalletProvider(mnemonic, `https://rinkeby.infura.io/v3/6e2f01b49b384897a521b1ecbba415cd`),
    network_id: "4", // Ropsten's id
    gas: 7000000, // Ropsten has a lower block limit than mainnet
    confirmations: 2, // # of confs to wait between deployments. (default: 0)
    timeoutBlocks: 200, // # of blocks before a deployment times out (minimum/default: 50)
    skipDryRun: true // Skip dry run before migrations? (default: false for public nets )
  },
```

Hình 9. Cấu hình mạng Ethereum qua testnet

- Trong đó:
  - ❓ `HDWalletProvider` dùng để khai báo thư viện `truffle-hdwalletprovider` đã được cài đặt.
  - ❓ Mnemonic là khai báo 12 ký tự mã hóa định danh từ tài khoản đã xin ủng hộ ở trên.
  - ❓ `YOUR_KEY` được cấp cấp từ website <https://infura.io/>.
- Cuối cùng mở command line ở thư mục đang chứa file cấu hình mạng Blockchain và gõ lệnh “`truffle migrate`”, khi thành công hệ thống trả về ba địa chỉ của các smart contract đã thiết lập ở trên.



Hình 10. Kết quả khi triển khai mạng Ethereum Blockchain

## 2. Android

Bạn cần kiểm tra các phiên bản

- Solidity: `solc --version`
- Web3j: `web3j --version`

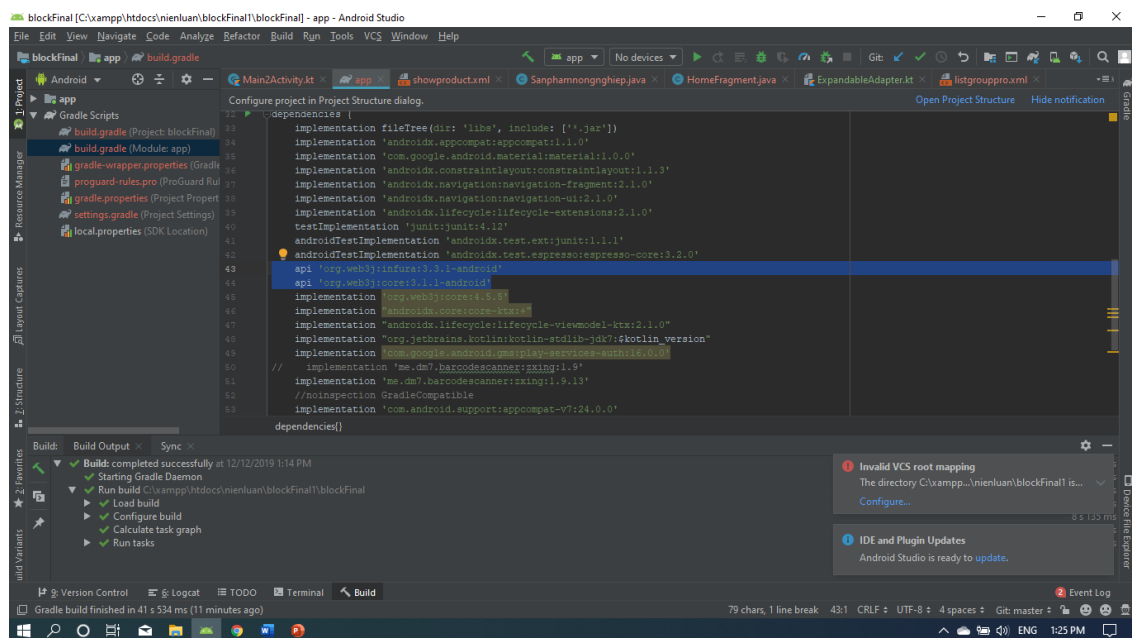
Nếu bạn chưa cài đặt thì hãy thực thi các lệnh sau đây:

- Cài đặt solidity: `npm install -g solc`
- Cài đặt Web3j trên Mac:  
`brew tap eb3j/web3j`  
`brew install web3j`
- Cài đặt Web3j trên Windows:  
<https://github.com/web3j/web3j/releases/tag/v4.1.1>

Bạn chuẩn bị file contract `sanpham.sol`

Bây giờ hãy tạo mới một project trên android và thêm các dependencies vào build.gradle.

```
// dependencies for contract interaction
implementation 'org.web3j:infura:3.3.1-android'
implementation 'org.web3j:core:3.1.1-android'
```



Tiếp theo chuyển đến thư mục có chứa sanpham.sol và thực hiện câu lệnh trên solc sanpham.sol — bin — abi — optimize -o <output-dir>/

Trong thư mục <output-dir>/ nếu có chứa 2 file Sanpham.bin và Sanpham.abi thì các bạn đã thành công việc biên dịch sanpham.sol

Tiếp tục tạo ra mã java bằng cách:

web3j solidity generate -b /path/to/<smart-contract>.bin -a /path/to/<smart-contract>.abi -o /path/to/src/main/java -p com.your.organisation.name

Cuối cùng chúng ta bắt đầu truy xuất dữ liệu bằng file java chúng ta vừa tạo ra

## PHẦN KẾT LUẬN

Qua quá trình nghiên cứu về Blockchain và một số ứng dụng của công nghệ này, cùng với sự giúp đỡ tận tình của thầy hướng dẫn và bạn bè, niên luận đã đạt được một số kết quả như sau:

Về mặt nội dung, niên luận đạt được một số kết quả sau đây:

### 1. Tìm hiểu và nghiên cứu:

- Tiếp cận về công nghệ Blockchain.

- Khái quát về nền tảng Ethereum Blockchain.
- Thiết lập và triển khai một mạng Blockchain trên môi trường cục bộ và môi trường kiểm thử công khai rinkeby.
- Xây dựng ứng dụng quét mã qr trên nền tảng android để truy xuất dữ liệu trên blockchain thông qua thư viện web3j

## 2. Thực nghiệm:

- Hoàn thành hệ thống chuỗi truy xuất nguồn gốc trên nền tảng Ethereum Blockchain trong đó:
  - + Tạo thành công smart contract trên nền tảng web
  - + Truy xuất dữ liệu trên nền tảng android.

### ***Định hướng nghiên cứu tiếp theo***

- Tiếp tục phát triển hệ thống quản lý chuỗi cung ứng, sử dụng chữ ký số để quá trình đăng nhập không cần thông qua plugin Metamask.
- Hoàn thiện ứng dụng đã xây dựng để có thể sử dụng trong thực tế.

## TÀI LIỆU THAM KHẢO

- [1] <https://medium.freecodecamp.org/how-to-write-and-deploy-your-first-smartcontract-341d5e2ffb35>
- [2] <https://kipalog.com/posts/Ethereum--mot-nen-tang-Blockchain-day-trien-vong>
- [3] <http://www.ethdocs.org/en/latest/>
- [4] <https://truffleframework.com/docs>
- [5] <https://infura.io/docs>
- [6] <https://web3js.readthedocs.io/en/1.0/web3-eth.html>
- [7] [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=2&cad=rja&uact=8&ved=0ahUKEwiK1tvCgfreAhXKdH0KHfxdA\\_MQtwlIKjAB&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DWyDrj8jrEjQ&usg=AOvVaw1Yksw7dkSz-4qqYm2SOlgT](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=2&cad=rja&uact=8&ved=0ahUKEwiK1tvCgfreAhXKdH0KHfxdA_MQtwlIKjAB&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DWyDrj8jrEjQ&usg=AOvVaw1Yksw7dkSz-4qqYm2SOlgT)
- [8] [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=1&cad=rja&uact=8&ved=0ahUKEwj1k5bUgfreAhUJT30KHat6CYwQtwlIJzAA&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D\\_rumMSlrmil&usg=AOvVaw3rgf1tWzlzO0oJA38bwZvo](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=1&cad=rja&uact=8&ved=0ahUKEwj1k5bUgfreAhUJT30KHat6CYwQtwlIJzAA&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D_rumMSlrmil&usg=AOvVaw3rgf1tWzlzO0oJA38bwZvo)
- [9] <http://diendan.congdongcviet.com/archive/index.php/t-42687.html>