

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐÀ LẠT**



**BÁO CÁO TỔNG KẾT
ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN NĂM 2024**

**XÂY DỰNG ỨNG DỤNG TRUY XUẤT NGUỒN GỐC
DÊ/CỪU NINH THUẬN**

Thuộc nhóm ngành khoa học: Công nghệ Thông tin

Lâm Đồng, tháng 5/2024

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN NĂM 2024

**XÂY DỰNG ỨNG DỤNG TRUY XUẤT NGUỒN GỐC
DÊ/CỪU NINH THUẬN**

Thuộc nhóm ngành khoa học: Công nghệ thông tin

Sinh viên thực hiện: Phan Thanh Thảo Quyên

Giới tính: Nữ

Dân tộc: Kinh

Lớp, khoa: Lớp CTK45B, Khoa CNTT

Năm thứ: 3 /Số năm đào tạo: 4

Ngành học: Công nghệ Thông tin

Người hướng dẫn: KS. La Quốc Thắng

Lâm Đồng, tháng 5/2024

MỤC LỤC

MỤC LỤC	ii
DANH MỤC HÌNH ẢNH.....	v
DANH MỤC BẢNG BIỂU.....	vii
DANH MỤC TỪ VIẾT TẮT	viii
THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI	x
THÔNG TIN VỀ SINH VIÊN CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI	xiii
MỞ ĐẦU	1
1. Tổng quan tình hình nghiên cứu	1
2. Lý do chọn đề tài	1
3. Mục tiêu đề tài	2
4. Phương pháp nghiên cứu	3
5. Đối tượng nghiên cứu	3
6. Phạm vi nghiên cứu	3
Chương 1. GIỚI THIỆU TỔNG QUAN	4
1.1. Tổng quan về hệ thống truy xuất nguồn gốc	4
1.1.1. Truy xuất nguồn gốc sản phẩm là gì	4
1.1.2. Quy trình truy xuất	4
1.1.3. Chuỗi cung ứng sản phẩm được sử dụng	5
1.1.4. Các yếu tố cần đảm bảo	6
1.2. Tổng quan về Flutter.....	7
1.2.1. Giới thiệu	7
1.2.2. Layout trong Flutter	8
1.2.3. Widget trong Flutter.....	8
1.3. Tổng quan về ReactJS	9
1.3.1. Giới thiệu sơ lược về ReactJS	9
1.3.2. Rendering Elements	10
1.3.3. Stateful components và props	10
1.3.4. Một số hook thông dụng	12
1.4. Tổng quan về NodeJS	14

1.4.1. NodeJS là gì ?	14
1.4.2. Các tính năng của NodeJS	15
1.4.3. Ứng dụng của NodeJS.....	15
1.5. Các thư viện hỗ trợ	16
1.5.1. Axios	16
1.5.2. React-router-dom	17
Chương 2. PHÂN TÍCH VÀ THIẾT KẾ	18
2.1. Phân tích và mô tả chức năng hệ thống	18
2.1.1. Về sơ đồ use case	18
2.1.2. Danh sách actor.....	19
2.1.3. Danh sách use case.....	20
2.1.4. Lược đồ use case.....	22
2.2. Cơ sở dữ liệu.....	22
2.2.1 Bảng dữ liệu	22
2.2.1. Các quan hệ.....	23
Chương 3. TRIỂN KHAI ỨNG DỤNG.....	28
3.1. Kiến trúc hệ thống.....	28
3.2. Xác thực API với Json Web Token (JWT).....	28
3.2.1. Json Web Token là gì?	28
3.2.2. Sơ đồ hoạt động	29
3.3. Xử lý upload ảnh lên Cloudinary.....	30
3.4. Triển khai hệ thống.....	31
3.4.1. API	32
3.4.2. Ứng dụng di động	33
3.4.3. Ứng dụng web.....	33
3.5. Kết quả.....	36
3.5.1. Ứng dụng di động	36
3.5.2. Ứng dụng web.....	41
Chương 4. KẾT LUẬN VÀ KIẾN NGHỊ.....	51
4.1. Kết luận.....	51
4.2. Hướng phát triển	51
TÀI LIỆU THAM KHẢO	52

1. Sách.....	52
2. Link tham khảo	52

DANH MỤC HÌNH ẢNH

Hình 1. Quy trình truy xuất nguồn gốc sản phẩm	6
Hình 2. Logo Flutter	7
Hình 3. Multiple Child Widgets (minh họa)	8
Hình 4. Hook useState trong class component.	13
Hình 5. Hook useState trong functional component.	13
Hình 6. Hook useEffect trong class component.	14
Hình 7. Sơ đồ Use Case	22
Hình 8. Cơ sở dữ liệu.....	22
Hình 9. Cơ sở dữ liệu chi tiết 1	24
Hình 10. Cơ sở dữ liệu chi tiết 2	25
Hình 11. Cơ sở dữ liệu chi tiết 3	26
Hình 12. Cơ sở dữ liệu chi tiết 4	27
Hình 13. Mô hình kiến trúc hệ thống	28
Hình 14. Cấu trúc của Json Web Token (JWT)	29
Hình 15. Ví dụ về một JWT	29
Hình 16. Quá trình hoạt động của json web token	30
Hình 17. Minh họa hoạt động khi upload ảnh lên clouddinary	31
Hình 18. Hàm xử lý lấy nội dung API.....	34
Hình 19. Hàm xử lý khi tạo	34
Hình 20. Hàm xử lý khi sửa	35
Hình 21. Hàm xử lý khi xóa	35
Hình 22. Hiện dữ liệu lên bảng (bảng danh sách đàn)	36
Hình 23. Đăng nhập/Đăng kí trên app di động.....	37
Hình 24. Giao diện quét mã QR	37
Hình 25. Giao diện thông tin truy xuất 1.....	38
Hình 26. Giao diện thông tin truy xuất 2.....	39
Hình 27. Giao diện thông tin tài khoản	40
Hình 28. Giao diện cây trồng/vùng trồng di động.....	40
Hình 29. Giao diện khi quét mã lỗi	41
Hình 30. Giao diện danh sách cơ sở (trang trại/nhà máy).....	42
Hình 31. Giao diện chi tiết trang trại	42

Hình 32. Trang web thông tin lúa đàn.....	43
Hình 33. Trang web thông tin chi tiết một lúa đàn và những con trong đàn	43
Hình 34. Trang web thông tin chi tiết một lúa đàn 1	43
Hình 35. Trang web thông tin chi tiết một lúa đàn 2	44
Hình 36. Trang web thông tin chi tiết một lúa đàn 3	44
Hình 37. Giao diện thông tin chi tiết một lúa đàn 4.....	45
Hình 38. Giao diện trang quản lý thuốc	45
Hình 39. Giao diện trang bệnh (thông tin)	46
Hình 40. Giao diện trang bệnh (hình ảnh).....	46
Hình 41. Giao diện trang xử lý đóng gói sản phẩm	47
Hình 42. Giao diện xử lý đóng gói.....	48
Hình 43. Giao diện thông tin sản phẩm.....	48
Hình 44. Giao diện trang phân phối	49
Hình 45. Trang giao diện danh sách người dùng	49
Hình 46. Trang web thông tin admin.....	50

DANH MỤC BẢNG BIỂU

Bảng 1. Danh sách actor	19
Bảng 2. Danh sách API	32

DANH MỤC TỪ VIẾT TẮT

STT	Chữ viết tắt	Tên đầy đủ	Chú giải
1	QR	Quick Response	Mã phản hồi nhanh, mã QR, mã vạch hai chiều
2	API	Application Programming Interface	Giao diện lập trình ứng dụng
3	REST	Representational State Transfer	Một kiến trúc phần mềm cho phép các hệ thống phân tán giao tiếp và truy cập vào dữ liệu thông qua giao thức HTTP.
4	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
5	JWT	Json web token	Truyền thông tin xác thực giữa các bên dưới dạng JSON
6	NPM	Node Package Manager	Trình quản lý gói cho ngôn ngữ lập trình JavaScript, cho phép người phát triển quản lý và cài đặt các thư viện và công cụ.
7	MVC	Model - View - Controller	Mô hình kiến trúc phần mềm
8	DOM	Document Object Model	Giao diện lập trình ứng dụng
9	ExpressJS, NestJS, MeteorJS, NodeJS		Các framework hoặc nền tảng phát triển ứng dụng web bằng JavaScript.
10	CLI	Command Line Interface	Giao diện dòng lệnh, tương tác với máy tính bằng các lệnh trực tiếp.
11	AJAX	Asynchronous JavaScript and XML	Kỹ thuật cho phép các ứng dụng web tải và gửi dữ liệu một cách bất đồng bộ, không cần tải lại trang hoàn toàn.
12	JSON	JavaScript Object Notation	Định dạng trao đổi dữ liệu con người và máy tính
13	XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng, thường được sử dụng để lưu trữ và truyền tải dữ liệu cấu trúc.
14	HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu siêu văn bản, dùng để tạo cấu trúc của các trang web.

15	CSS	Cascading Style Sheets	Ngôn ngữ mô tả kiểu, dùng để định dạng và trang trí cho các trang web.
16	UI	User Interface	Giao diện người dùng
17	GlobalGAP, VietGAP	Global Good Agricultural Practice	Các tiêu chuẩn và hướng dẫn cho sản xuất nông nghiệp an toàn và bền vững, tương ứng trong phạm vi Việt Nam và toàn cầu.
18	ASC	Aquaculture Stewardship Council	Tổ chức đặt ra các tiêu chuẩn cho việc quản lý nuôi trồng thủy sản.
19	HACCP	Hazard Analysis and Critical Control Points	Phương pháp tiếp cận dự phòng hệ thống đối với an toàn thực phẩm, nhằm xác định, đánh giá và kiểm soát các nguy cơ trong quá trình sản xuất thực phẩm.

THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung

- Tên đề tài: Xây dựng ứng dụng truy xuất nguồn gốc dê/cừu Ninh Thuận
- Sinh viên thực hiện:

STT	Họ và tên	MSSV	Lớp	Email
1	Phan Thanh Thảo Quyên	2115260	CTK45B	2115260@dlu.edu.vn
2	Đậu Thị Tiểu Diệp	2115196	CTK45B	2115196@dlu.edu.vn
3	Lê Viết Đăng Quang	2111875	CTK45B	2111875@dlu.edu.vn
4	Nguyễn Hoàng Nam Khánh	2212391	CTK46B	2212391@dlu.edu.vn
5	Nhâm Vĩnh Thủy	2213743	CTK46B	2213743@dlu.edu.vn

- Người hướng dẫn: KS. La Quốc Thắng

2. Mục tiêu đề tài

Xây dựng ứng dụng truy xuất nguồn gốc các sản phẩm của dê/cừu ở Ninh Thuận qua hình ảnh QR nhằm hỗ trợ người dùng xác minh nguồn gốc của sản phẩm.

- Thu thập dữ liệu và xây dựng cơ sở dữ liệu.
- Phát triển ứng dụng di động.
- Xây dựng hệ thống mã QR.
- Phát triển trang quản trị trên web.
- Đánh giá kết quả và đề xuất phương hướng phát triển.

3. Tính mới và sáng tạo

Trong bối cảnh ngày càng nghiêm trọng về vấn đề hàng giả và không rõ nguồn gốc của sản phẩm, việc thiết lập hệ thống truy xuất nguồn gốc trở nên cấp bách trên toàn cầu, đặc biệt trong ngành sản xuất nông nghiệp. Điều này không chỉ đảm bảo an toàn thực phẩm mà còn nâng cao giá trị sản phẩm, bảo vệ quyền lợi của người tiêu dùng và đóng góp vào sự phát triển bền vững của ngành nông nghiệp.

Việc ứng dụng công nghệ thông tin vào quá trình truy xuất nguồn gốc sản phẩm đã mang lại nhiều lợi ích to lớn. Một trong những phương pháp phổ biến là sử dụng mã QR (Quick Response) để đối chiếu thông tin sản phẩm với cơ sở dữ liệu trung tâm. Điều này cho phép người tiêu dùng dễ dàng kiểm tra thông tin về nguồn gốc, quy trình sản xuất và các thông tin liên quan về sản phẩm bằng cách quét mã QR trên sản phẩm bằng điện thoại di động của họ.

Việc thực nghiệm trên mô hình dê/cừu ở Ninh Thuận là một bước quan trọng trong quá trình nghiên cứu này. Điều này cho phép nhóm nghiên cứu áp dụng các kỹ thuật và kinh nghiệm thu được vào thực tế, đánh giá tính khả thi và hiệu quả của hệ thống truy xuất nguồn gốc trên một tình huống thực tế.

Tóm lại, việc nghiên cứu và xây dựng ứng dụng truy xuất nguồn gốc sản phẩm là một bước quan trọng trong việc đảm bảo an toàn thực phẩm và nâng cao chất lượng sản phẩm trong ngành nông nghiệp.

4. Kết quả nghiên cứu

Đề tài này chủ yếu tập trung vào xây dựng ba hệ thống bao gồm:

- **Ứng dụng web:** Người quản trị viên: quản lý các thông tin về tài khoản, bệnh, thuốc, trang trại, danh mục vật nuôi. Người dùng khi quản lý quy trình chăm sóc đàn vật nuôi, các thông tin như nhật ký chăm sóc, phương pháp điều trị, thông tin thu hoạch, xử lý đóng gói, lô sản phẩm, sản phẩm, phân phối.

- **Ứng dụng di động:** người dùng dùng ứng dụng di động để thực hiện quét mã QR đã được in trên bao bì sản phẩm, ứng dụng sẽ hiển thị thông tin truy xuất cho người tiêu dùng xem. Các thông tin sẽ được hiển thị bao gồm: Trang trại, lứa/đàn vật nuôi, nhật ký chăm sóc, các điều trị đã thực hiện (vắc-xin, bệnh, phối giống,...), thu hoạch, xử lý đóng gói, phân phối sản phẩm.

- **API:** được hiểu là các phương thức, giao thức dùng để kết nối các thư viện và ứng dụng này tới khác thư viện và ứng dụng khác. Nhóm đã xây dựng một loạt các API cung cấp cho ứng dụng web quản lý các thông tin truy xuất để ứng dụng di động có thể truy xuất để lấy thông tin hiển thị.

5. Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài

5.1. Hiệu quả xã hội

- Ứng dụng giúp truy xuất thông tin các sản phẩm dê/cừu cung cấp cho người tiêu dùng thông tin về quy trình sản xuất sản phẩm. Giúp tăng sự minh bạch và độ tin cậy trong ngành chăn nuôi, giúp người tiêu dùng yên tâm hơn về nguồn gốc và chất lượng sản phẩm.

- Tạo ra cơ hội việc làm và tăng thu nhập cho người dân địa phương, đặc biệt là những người tham gia vào quy trình sản xuất và quản lý dữ liệu.

5.2. Hiệu quả an ninh

- Giúp kiểm soát và quản lý hiệu quả hơn các vấn đề an ninh thực phẩm, như dịch bệnh và kiểm soát chất lượng sản phẩm.

- Xây dựng một hệ thống quản lý dữ liệu an toàn và bảo mật, ngăn chặn các hoạt động giả mạo hoặc gian lận, đảm bảo an ninh và vệ sinh thực phẩm.

5.3. Khả năng áp dụng của đề tài

- Ứng dụng này có thể mở rộng và áp dụng cho các ngành sản xuất khác, không chỉ trong lĩnh vực chăn nuôi dê/cừu mà còn trong nhiều lĩnh vực nông nghiệp và thực phẩm khác.

- Các kinh nghiệm và hệ thống được xây dựng từ việc triển khai ứng dụng này có thể chia sẻ và áp dụng trong các khu vực khác, cả trong nước và quốc tế, mở ra tiềm năng phát triển lớn.

6. Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài (*ghi rõ tên tạp chí nếu có*) hoặc nhận xét, đánh giá của cơ sở đã áp dụng các kết quả nghiên cứu (*nếu có*):

Ngày ... tháng ... năm

Sinh viên chịu trách nhiệm chính
thực hiện đề tài
(*ký, họ và tên*)

Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài (*phần này do người hướng dẫn ghi*):

Ngày ... tháng ... năm

Xác nhận của trường đại học
(*ký tên và đóng dấu*)

Người hướng dẫn
(*ký, họ và tên*)

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐÀ LẠT

THÔNG TIN VỀ SINH VIÊN
CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI

I. SƠ LƯỢC VỀ SINH VIÊN

Họ và tên: Phan Thanh Thảo Quyên

Sinh ngày: 23 tháng 01 năm 2003

Nơi sinh: Lâm Đồng

Lớp: CTK45B

Khóa: 45

Khoa: Công nghệ thông tin

Địa chỉ liên hệ: Lớp CTK45B, Khoa Công nghệ thông tin, Trường đại học Đà Lạt

Điện thoại: 0838830573

Email: 2115260@dlu.edu.vn

II. QUÁ TRÌNH HỌC TẬP

*** Năm thứ 1:**

Ngành học:

Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: ĐTBTL: 3.94/4.0 – Xuất sắc

*** Năm thứ 2:**

Ngành học:

Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: ĐTBTL: 3.76/4.0 – Xuất sắc

*** Năm thứ 3 – Học kỳ I:**

Chuyên ngành: Kỹ thuật Phần mềm

Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: ĐTBTL: 3.88/4.0 – Xuất sắc

Ngày ... tháng ... năm

Xác nhận của trường đại học

(ký tên và đóng dấu)

Sinh viên chịu trách nhiệm chính

thực hiện đề tài

(ký, họ và tên)

MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu

Trước bài toán ngày càng nghiêm trọng về hàng giả, hàng nhái, và sự không rõ nguồn gốc xuất xứ của các sản phẩm, việc thiết lập hệ thống truy xuất nguồn gốc đã trở thành một ưu tiên quan trọng trên phạm vi toàn cầu. Đối với ngành sản xuất nông nghiệp, việc này càng trở nên quan trọng hơn bao giờ hết, truy xuất nguồn gốc sản phẩm nông nghiệp không chỉ giúp người tiêu dùng có được thông tin chi tiết và chính xác về sản phẩm mà họ sử dụng, mà còn đóng góp vào việc nâng cao giá trị của các sản phẩm nông nghiệp. Chúng ta đã chứng kiến sự phát triển đáng kể của công nghệ trong thời gian gần đây, đặc biệt là công nghệ thông tin. Công nghệ này đã được ứng dụng để xây dựng các hệ thống truy xuất nguồn gốc điện tử tiên tiến. Bằng cách sử dụng các công nghệ thu thập dữ liệu trên mã QR (Quick Response), việc truy xuất nguồn gốc của sản phẩm đã trở nên nhanh chóng và hiệu quả hơn bao giờ hết. Những cải tiến này không chỉ giúp cải thiện quản lý nguồn gốc của sản phẩm mà còn đảm bảo tính minh bạch trong chuỗi cung ứng, từ người sản xuất đến người tiêu dùng. Việc ứng dụng thành tựu của khoa học và công nghệ, đặc biệt là công nghệ thông tin, vào việc xây dựng hệ thống truy xuất nguồn gốc sản phẩm đã góp phần quan trọng vào việc bảo vệ quyền lợi của người tiêu dùng và cải thiện chất lượng của sản phẩm nông nghiệp. Đây là một bước quan trọng trong việc đảm bảo an toàn thực phẩm và thúc đẩy sự phát triển bền vững của ngành nông nghiệp.

2. Lý do chọn đề tài

Lý do, truy xuất nguồn gốc dê cừu Ninh Thuận là do yếu tố quan trọng và bắt buộc đối với nhiều loại sản phẩm, đặc biệt là mặt hàng dinh dưỡng. Tuy nhiên ở nước ta, vấn đề này chưa thực sự được chú ý bởi người dân thường hay mua sản phẩm ở những khu chợ quán không có người quản lý hay kiểm tra đúng quy trình.

Dẫn đến các loại hàng giả, hàng kém chất lượng có cơ hội hoành hành. Điều này khiến người tiêu dùng mua nhầm sản phẩm, mất niềm tin vào thương hiệu và các doanh nghiệp sản xuất chính hãng thì mất uy tín và khả năng tiêu thụ hàng hóa sụt giảm. Vì vậy ứng dụng truy xuất nguồn gốc giúp người tiêu dùng mua sản phẩm an toàn và đảm bảo chất lượng nói chung và truy xuất nguồn gốc dê cừu nói riêng. Tại Ninh Thuận có nhiều trang trại nuôi dê cừu nhóm em chọn đề tài này giúp các trang trại dễ quản lý và theo dõi quá trình chăm sóc dê cừu từ khi sinh đến lúc thành phẩm đến tay người tiêu dùng.

Thông qua app truy xuất dễ dàng biết nguồn gốc , quy trình sản xuất, vận chuyển của mỗi khâu chăm sóc hình thành sản phẩm. Đảm bảo được niềm tin của người dùng trước khi mua sản phẩm khi bản thân họ đã hiểu rõ nguồn gốc của chúng. Ngoài ra, giúp các nhà cung cấp và phân phối tránh và dễ quản lý được các sản phẩm không uy tín, không an toàn trong xuất hiện ngoài thị trường.

3. Mục tiêu đề tài

Mục tiêu của đề tài nghiên cứu là xây dựng một ứng dụng truy xuất nguồn gốc cho ngành chăn nuôi dê/cừu tại Ninh Thuận. Việc này sẽ giúp cải thiện quản lý nguồn lực, nâng cao chất lượng sản phẩm, đồng thời tạo điều kiện thuận lợi cho việc tiêu thụ và quảng bá sản phẩm địa phương ra thị trường rộng lớn hơn.

Trong thời đại công nghệ thông tin hiện nay, việc áp dụng công nghệ để quản lý và truy xuất nguồn gốc sản phẩm ngày càng trở nên cần thiết. Đặc biệt đối với ngành chăn nuôi, việc có thể theo dõi và kiểm soát từng giai đoạn của quá trình sản xuất từ trại nuôi đến tay người tiêu dùng sẽ giúp tăng cường tính minh bạch và đảm bảo chất lượng sản phẩm.

Một trong những vấn đề lớn nhất mà ngành chăn nuôi dê/cừu ở Ninh Thuận đang phải đối mặt là việc thiếu thông tin về nguồn gốc của sản phẩm. Điều này gây khó khăn cho việc xác định chất lượng, đồng thời ảnh hưởng đến uy tín của sản phẩm trên thị trường. Do đó, việc xây dựng một hệ thống truy xuất nguồn gốc sẽ đóng vai trò quan trọng trong việc giải quyết vấn đề này.

Ứng dụng sẽ được phát triển với các tính năng chính như sau:

1. Ghi chép thông tin từ trại nuôi: hệ thống sẽ cho phép các trang trại ghi chép thông tin về quá trình nuôi dưỡng của dê/cừu, bao gồm thức ăn, điều kiện sống, thông tin về sức khỏe và vắc xin.

2. Quản lý hồ sơ cá nhân của từng con vật: mỗi con vật sẽ được gán một mã số riêng, qua đó hồ sơ của chúng sẽ được quản lý một cách tổ chức và chi tiết.

3. Theo dõi quá trình vận chuyển và xử lý sản phẩm: hệ thống sẽ ghi nhận thông tin về quá trình vận chuyển từ trại nuôi đến điểm bán hàng, cũng như các quy trình xử lý sản phẩm như giết mổ và đóng gói.

4. Xác thực nguồn gốc: người tiêu dùng có thể sử dụng ứng dụng để kiểm tra nguồn gốc của sản phẩm thông qua mã vạch hoặc mã QR được in trên bao bì.

5. Cung cấp thông tin chi tiết cho người tiêu dùng: ứng dụng cung cấp thông tin chi tiết về nguồn gốc, quy trình sản xuất và kiểm định chất lượng của sản phẩm, giúp người tiêu dùng có quyết định mua hàng được thông tin và tự tin hơn.

Việc triển khai một ứng dụng truy xuất nguồn gốc cho ngành chăn nuôi dê/cừu tại Ninh Thuận không chỉ mang lại lợi ích cho các nhà sản xuất và người tiêu dùng mà còn đóng vai trò quan trọng trong việc thúc đẩy sự phát triển bền vững của ngành này.

4. Phương pháp nghiên cứu

Tìm hiểu các tài liệu về thông tin vật nuôi, quá trình chăm sóc. Từ đó đưa ra các chức năng cần có của ứng dụng. Điều tra và tìm hiểu về các ứng dụng và thách thức trong việc triển khai ứng dụng quét mã truy xuất nguồn gốc.

5. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này là chuỗi cung ứng sản phẩm, quy trình phát triển của vật nuôi. Hệ thống quét mã truy xuất nguồn gốc sản phẩm từ dê/cừu.

6. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài này ứng dụng truy xuất nguồn gốc sản phẩm từ dê/cừu trong khu vực tỉnh Ninh Thuận, tập trung vào các doanh nghiệp nuôi và xử lý chế biến dê/cừu theo cách truyền thống, chưa ứng dụng công nghệ thông tin trong việc quản lý quá trình sản xuất, vận chuyển và cung cấp các sản phẩm đến người tiêu dùng.

Về chức năng, đề tài này chủ yếu tập trung thử nghiệm quy trình truy xuất nguồn gốc sản phẩm dê/cừu thông qua quét mã QR với các thông tin được lưu trữ trong cơ sở dữ liệu.

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

1.1. Tổng quan về hệ thống truy xuất nguồn gốc

1.1.1. Truy xuất nguồn gốc sản phẩm là gì

Truy xuất nguồn gốc sản phẩm là một quá trình giúp người tiêu dùng có khả năng truy xuất và tìm hiểu về nguồn gốc, xuất xứ của sản phẩm mà họ đã mua. Nó cho phép người tiêu dùng “truy ngược” từ sản phẩm đang được bày bán trên kệ hàng về nơi sản xuất ban đầu và rà soát từng công đoạn trong quá trình chế biến và phân phối của sản phẩm đó.

Quá trình truy xuất nguồn gốc sản phẩm bắt đầu từ việc ghi nhận thông tin chi tiết về sản phẩm tại mỗi giai đoạn của chuỗi cung ứng, từ nguyên liệu đến sản phẩm hoàn chỉnh. Điều này bao gồm thông tin về nguồn nguyên liệu, quá trình sản xuất, điều kiện lưu trữ và vận chuyển, cũng như thông tin về các bước kiểm định và xác thực.

Thông tin này thường được lưu trữ trong hệ thống quản lý chuỗi cung ứng của các doanh nghiệp và có thể được truy cập thông qua mã vạch, mã QR, hoặc các công nghệ khác. Khi sản phẩm đến tay người tiêu dùng, họ có thể sử dụng các công cụ như ứng dụng di động, trang web, hoặc máy quét mã vạch để truy xuất thông tin về nguồn gốc và lịch sử của sản phẩm.

Việc truy xuất nguồn gốc sản phẩm mang lại nhiều lợi ích cho người tiêu dùng: Đầu tiên, nó cung cấp thông tin chi tiết về chất lượng và an toàn của sản phẩm, giúp họ đưa ra quyết định mua hàng thông minh và có ý thức. Thứ hai, nó tạo điều kiện cho sự minh bạch và trung thực trong quản lý chuỗi cung ứng, đồng thời đảm bảo tuân thủ các quy định về an toàn thực phẩm và quy định pháp lý.

Ngoài ra, việc truy xuất nguồn gốc sản phẩm cũng giúp giảm thiểu rủi ro về sản phẩm giả mạo và hàng giả, bảo vệ quyền lợi của người tiêu dùng và tạo ra một môi trường cạnh tranh lành mạnh trên thị trường. Đồng thời, nó cũng tạo điều kiện cho việc theo dõi và giám sát hiệu quả các chiến dịch bảo vệ người tiêu dùng và đảm bảo chất lượng sản phẩm.

Tóm lại, truy xuất nguồn gốc sản phẩm là một công cụ quan trọng trong việc xây dựng một hệ thống chuỗi cung ứng an toàn, minh bạch và đáng tin cậy, đồng thời giúp người tiêu dùng có được thông tin đầy đủ và chính xác về sản phẩm mà họ tiêu thụ.

1.1.2. Quy trình truy xuất

Quy trình truy xuất nguồn gốc sản phẩm nông sản gồm có 6 bước sau:

- **Bước 1: tiến hành khảo sát quy mô chăn nuôi**

Các chuyên gia từ nhà cung cấp sẽ tiến hành khảo sát tổng thể về quy mô sản xuất của sản phẩm, bao gồm từ trại giống, trang trại nuôi trồng, cơ sở chế biến và quá trình vận chuyển, cho đến khi sản phẩm được đưa ra thị trường.

Nhà cung cấp giải pháp sẽ theo dõi mỗi quy trình và giai đoạn trong quá trình hình thành sản phẩm, nhằm đảm bảo rằng thông tin được cung cấp cho khách hàng là chính xác và tường minh nhất.

- Bước 2: xây dựng quy trình truy xuất nguồn gốc thực phẩm phù hợp

Sau khi hoàn thành khảo sát, các chuyên gia sẽ xác lập quy trình truy xuất nguồn gốc, tuân thủ chặt chẽ các quy trình hoạt động và tiêu chuẩn quốc gia đang áp dụng như VietGAP, GlobalGAP, ASC, HACCP...

Mục tiêu của chúng là đảm bảo khi truy xuất, người tiêu dùng có thể biết rõ từng giai đoạn và thời điểm của quá trình hình thành, chế biến và phân phối sản phẩm một cách chính xác.

- Bước 3: tiến hành thiết kế biểu mẫu truy xuất nguồn gốc thực phẩm

Bước quan trọng này liên quan đến việc xây dựng các biểu mẫu để thu thập thông tin cần thiết cho quá trình nhập liệu. Trong giai đoạn này, cần thu thập thông tin về quá trình sản xuất cũng như các nguyên liệu hỗ trợ trong quá trình chăn nuôi như thức ăn, các loại thuốc điều trị bệnh, vắc-xin tiêm chủng...

Cần tạo biểu mẫu dựa trên quy trình truy xuất, đảm bảo rằng tất cả thông tin cần thiết về sản phẩm có sẵn khi thực hiện truy xuất.

- Bước 4: xây dựng hệ thống truy xuất nguồn gốc sản phẩm chăn nuôi

Tiến hành triển khai hệ thống phần mềm truy xuất nguồn gốc dê/cừu Ninh Thuận. Mục tiêu là đảm bảo người dùng có thể dễ dàng truy xuất nguồn gốc sản phẩm và cung cấp đầy đủ thông tin truy xuất của sản phẩm.

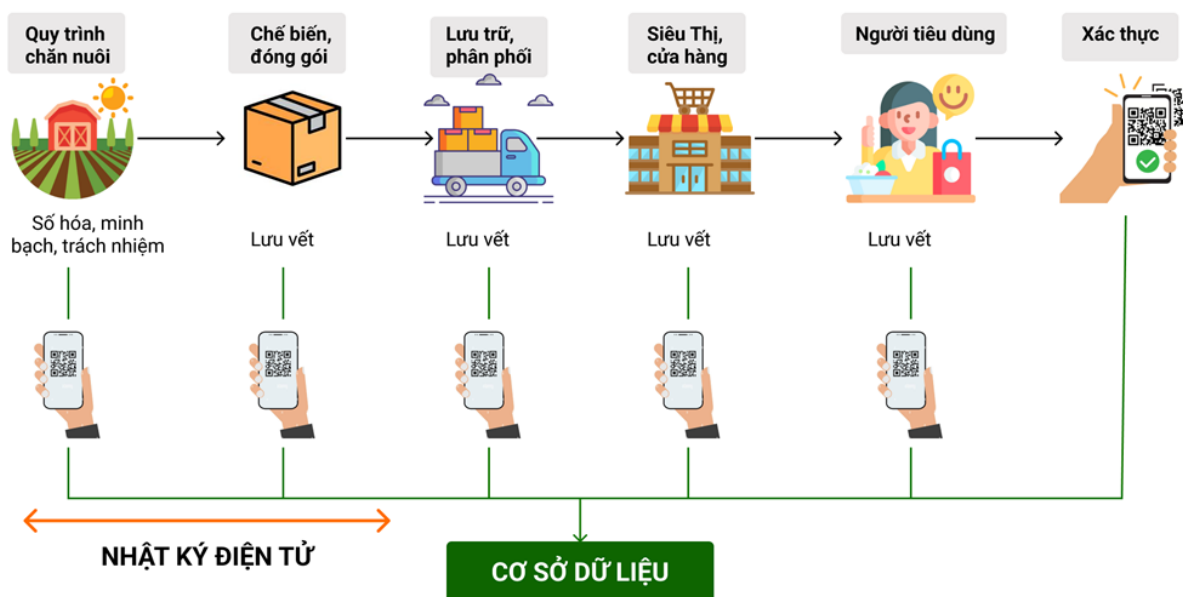
- Bước 5: hướng dẫn dùng hệ thống phần mềm truy xuất nguồn gốc thực phẩm

Khi sử dụng phần mềm truy xuất nguồn gốc hàng hóa, người dùng sẽ được hướng dẫn, đào tạo sử dụng hệ thống phần mềm để có thể dễ dàng tiếp cận cũng như sử dụng.

- Bước 6: triển khai quy trình truy xuất nguồn gốc sản phẩm chăn nuôi

Khách hàng tiến hành truy xuất nguồn gốc thông qua hệ thống phần mềm đã xây dựng và phải đảm bảo các yếu tố: 1 bước trước 1 bước sau – thông tin dữ liệu chính – minh bạch – và có sự tham gia đầy đủ của các bên cung ứng.

1.1.3. Chuỗi cung ứng sản phẩm được sử dụng



Hình 1. Quy trình truy xuất nguồn gốc sản phẩm

Ở giai đoạn đầu của chuỗi cung ứng, đối tượng chính là các loại vật nuôi, cụ thể ở đây là dê/cừu, chưa có khả năng thu hoạch ngay tại thời điểm đó. Trải qua thời gian chăm sóc, nuôi dưỡng, chúng trở thành nông sản bán thành phẩm. Để đến với tay người tiêu dùng, nông sản cần phải trải qua các giai đoạn như thu hoạch, chế biến đóng gói, lưu trữ, phân phối và buôn bán mới chính thức trở thành sản phẩm tiêu dùng. Toàn bộ quá trình trong chuỗi cung ứng được ghi lại và tổ chức theo từng giai đoạn, do đó, khách hàng, chính là người tiêu dùng có thể nắm bắt được thông tin đó bằng việc quét mã được dán trên các sản phẩm đó.

1.1.4. Các yếu tố cần đảm bảo

Truy xuất nguồn gốc chuẩn phải bảo đảm các yếu tố:

- **Thứ nhất**, có thông tin về chuỗi liên kết giá trị sản phẩm. Khi truy xuất nguồn gốc phải truy được sản phẩm đó có quá trình hình thành, vận chuyển, đồng thời thấy được toàn bộ chuỗi quá trình mà tất cả các đơn vị tham gia vào việc tạo ra sản phẩm.

- **Thứ hai**, thông tin của sản phẩm khi truy xuất phải thể hiện giống như đã công bố lưu hành.

- **Thứ ba**, Có sự chuẩn hóa, kiểm tra, giám sát của bên thứ ba trong toàn bộ chuỗi quá trình tạo ra sản phẩm đó.

- **Thứ tư**, cần có nhật ký điện tử về toàn bộ quá trình hình thành ra sản phẩm. Những tài liệu này để chứng minh quá trình tạo ra cũng như luân chuyển sản phẩm có bảo đảm an toàn hay không.

1.2. Tổng quan về Flutter

1.2.1. Giới thiệu

a. Khái niệm

Flutter là một framework phát triển ứng dụng di động và web nhanh chóng và đa nền tảng, được phát triển bởi Google. Flutter sử dụng ngôn ngữ lập trình Dart để phát triển các ứng dụng với thiết kế đẹp, trải nghiệm người dùng tốt và khả năng tùy chỉnh cao. Dart là một ngôn ngữ lập trình được phát triển bởi Google, được sử dụng chủ yếu để phát triển các ứng dụng web và di động.



Hình 2. Logo Flutter

Dart là một ngôn ngữ hiện đại, hướng đối tượng và có tính năng phát triển ứng dụng nhanh chóng và đáp ứng được yêu cầu của các ứng dụng web và di động phức tạp. Điều đặc biệt về Dart là tính tương thích ngược với JavaScript, giúp các nhà phát triển web và di động dễ dàng chuyển đổi sang Dart.

Ngôn ngữ này cũng được sử dụng trong framework phát triển ứng dụng Flutter của Google, cho phép các nhà phát triển tạo các ứng dụng di động đa nền tảng với khả năng tùy chỉnh cao và hiệu suất tốt. Với các tính năng tiên tiến và khả năng tích hợp đa dạng, Dart đang trở thành một trong những ngôn ngữ lập trình được ưa chuộng trong cộng đồng phát triển ứng dụng web và di động.

b. Tính năng

Flutter sử dụng ngôn ngữ lập trình Dart rất đơn giản và dễ dàng sử dụng. Hiện Flutter đã và đang là một react framework. Tính năng hot reload tiện lợi, dễ trải nghiệm, dễ dàng xây dựng giao diện, sửa lỗi nhanh chóng.

Các widget built-in đẹp mắt giúp giao diện người dùng đẹp, hoạt động phong phú, scroll mượt mà, linh hoạt, tự nhiên và khả năng tự nhận thức nền tảng.

Framework giải quyết những thách thức khó khăn trong giao diện người dùng bằng việc tập hợp các layout, platform và widget phong phú. Flutter là ứng dụng có hiệu năng cao. Thể hiện cùng một UI trên nhiều nền tảng.

1.2.2. Layout trong Flutter

a. Single Child Widgets

Các widget layout loại này chỉ có duy nhất một widget con và thường có chức năng bố trí nhất định. Center widget chỉ căn giữa widget con so với widget cha của nó và Container widget cung cấp khả năng linh hoạt trong việc đặt widget con bên trong nó thông qua các tùy chọn như padding, đường viền, nền,...

Single child widgets thích hợp cho việc tạo ra các widget có tính ứng dụng cao và chỉ có một chức năng duy nhất như button, label....

Một số single child layout widgets quan trọng trong Flutter:

- **Padding:** Được sử dụng để padding child widget. Ở đây, padding có thể sử dụng EdgeInsets class.

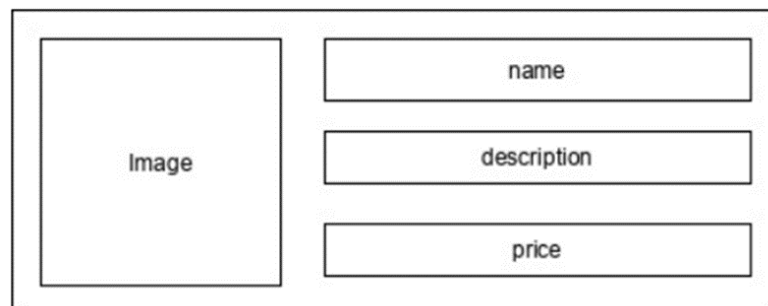
- **Align:** Căn lề child widget sử dụng thuộc tính alignment. Giá trị của alignment có thể được cung cấp bởi FractionalOffset class. FractionalOffset class xác định vị trí của phần tử từ vị trí điểm trên cùng bên trái

- Một số single child layout khác: Fitted, Aspect, Ratio, ConstrainedBox, ...

b. Multiple Child Widgets

Loại widget layout này sẽ cho phép có nhiều hơn một widget con. Ví dụ Row widget cho phép bố trí các widget con theo chiều ngang thành một hàng trong khi Column widget cho phép bố trí các widget con theo chiều dọc thành một cột.

Một số widget layout dạng này được sử dụng phổ biến: Row, Column, ListView,



Hình 3. Multiple Child Widgets (minh họa)

1.2.3. Widget trong Flutter

a. Text

Text là một widget được sử dụng để hiển thị văn bản. Trong đó text có một số những thuộc tính hay sử dụng sau :

- **textAlign:** Căn chỉnh text theo chiều ngang

- **maxLines**: Số lượng dòng tối đa
- **style**: TextStyle() bao gồm các thuộc tính để set style như là color,fontSize...

b. RichText

Là một widget mở rộng của Text giúp hiển thị text với nhiều style khác nhau . Văn bản khi cần set nhiều thuộc tính ta sử dụng TextSpan để set các thuộc tính cần thiết cho từng đoạn văn bản đó.

c. Button

Widget này thì thường bao gồm một image hoặc text hoặc có thể bao gồm cả hai . Khi người dùng chạm vào nó, nó sẽ kích hoạt sự kiện nhấp chuột và thực hiện hành động thích hợp. Flutter có nhiều loại nút khác nhau. Các loại nút này được sử dụng cho các mục đích khác nhau. Ở đây có một số loại button cơ bản như Flat Button, Raised Button, Floating Button, Drop Down Button, Icon Button, Popup Button)

1.3. Tổng quan về ReactJS

1.3.1. Giới thiệu sơ lược về ReactJS

React là một thư viện JavaScript mã nguồn mở dùng để xây dựng giao diện người dùng và ứng dụng web một cách nhanh chóng, hiệu quả.

Mục đích chính của React trong một ứng dụng web là để quản lý lớp giao diện của ứng dụng, tương tự như phần V (View) trong mô hình MVC (Model - View - Controller) vốn đã rất phổ biến trước đây trong website. Thay vì phải xử lý toàn bộ giao diện người dùng như một thành phần duy nhất, React có thể chia nhỏ những giao diện phức tạp thành các thành phần độc lập nhỏ hơn gọi là các component, có thể tái sử dụng lại cho toàn bộ website. Nhờ đó, React sẽ tận dụng được tốc độ và hiệu năng của ngôn ngữ lập trình JavaScript để tương tác với DOM (Document Object Model) trong website giúp việc render trang web sẽ trở nên nhanh hơn rất nhiều, đồng thời làm cho trang web trở nên linh hoạt, có tính tương tác cao và có thể tương thích với nhiều thiết bị có kích thước màn hình khác nhau.

React là một trong những thư viện để xây dựng giao diện người dùng phổ biến nhất hiện nay bởi những ưu điểm:

- **Hiệu năng tốt**: React sử dụng DOM ảo. Trong ứng dụng React, khi một đối tượng thay đổi thì DOM ảo sẽ được cập nhật. Sau đó nó sẽ so sánh với trạng thái trước đó và chỉ cập nhật những đối tượng có sự thay đổi ở trong DOM thật thay vì cập nhật toàn bộ các đối tượng. Điều này sẽ giúp website trở nên nhanh hơn rất nhiều.

- **Tạo ra các component tái sử dụng:** Component là các thành phần để xây dựng nên một ứng dụng React. Trong mỗi component sẽ chứa nội dung hiển thị lên giao diện người dùng và các logic xử lý riêng của component đó. Các component này có thể hoạt động độc lập và được tái sử dụng trong toàn bộ ứng dụng, điều này sẽ giúp giảm thiểu thời gian phát triển ứng dụng, tránh việc lặp lại các đoạn mã giống nhau nhiều lần.

- **Đơn giản hóa mã nguồn:** React cung cấp thêm các cú pháp mở rộng cho ngôn ngữ JavaScript gọi là JSX. Tính năng này sẽ cho phép viết các mã HTML bên trong JavaScript. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã JavaScript. Các mã này sẽ có thời gian thực hiện nhanh hơn nhiều so với mã viết trực tiếp bằng JavaScript. Đồng thời nó cũng an toàn hơn vì mã JSX sẽ được biên dịch trước khi chạy giống như các ngôn ngữ lập trình khác như Java, C++,... vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch.

1.3.2. Rendering Elements

React element là thành phần nhỏ nhất có thể render trên ứng dụng. Component trong React sẽ được xây dựng từ nhiều element khác nhau. React element không phải là một DOM element và nó sẽ được React DOM ánh xạ thành DOM element của trình duyệt.

Các ứng dụng được xây dựng bằng React sẽ có một node gốc gọi là root. Toàn bộ nội dung chứa trong node root này sẽ được quản lý bởi React DOM. Node root này sẽ là node ngoài cùng của React, thông thường thì sẽ là một thẻ `<div>` có thuộc tính `id="root"`.

Một React element là một đối tượng bất biến (immutable), nghĩa là từ khi sinh ra thì element đó sẽ không thể bị thay đổi thuộc tính hoặc các nội dung con bên trong nó. Nói cách khác, một element sẽ tương tự như một khung hình trong một bộ phim, nó thể hiện giao diện tại một thời điểm nhất định. Để cập nhật lại giao diện mới, React sẽ sử dụng cơ chế tạo ra một element mới, sau đó đưa nó vào hàm `ReactDOM.render()` để render lại giao diện.

Mặc dù quá trình React tạo ra element mới và render lại giao diện sẽ khiến nhiều người hoài nghi về tính hiệu quả của cơ chế này. Tuy nhiên trên thực tế, trước khi React xử lý việc render lại giao diện thì nó còn thực hiện một bước kiểm tra toàn bộ các node ở trong trang hiện tại và chỉ cập nhật lại các node có sự thay đổi. Vì vậy tốc độ của cơ chế này sẽ nhanh hơn rất nhiều so với các cơ chế render lại toàn bộ trang giao diện như trước đây.

1.3.3. Stateful components và props

a. Stateful components:

Components là các khối xây dựng cơ bản của React. Chúng là các đơn vị độc lập có thể tái sử dụng, chứa trong chúng mã HTML, CSS và logic JavaScript. Components giúp chia nhỏ giao diện thành các phần nhỏ hơn, dễ quản lý và tái sử dụng. Dưới đây là các lợi ích khi sử dụng components:

- **Tái sử dụng và quản lý độc lập:** Components trong ReactJS là các đơn vị độc lập mà bạn có thể tái sử dụng trong ứng dụng của mình. Bằng cách chia nhỏ giao diện thành các thành phần nhỏ hơn, bạn có thể quản lý chúng một cách độc lập và tập trung vào từng phần riêng biệt. Điều này giúp tăng tính modular và linh hoạt trong việc phát triển ứng dụng.

- **Mã HTML, CSS và logic JavaScript:** Mỗi component có thể chứa mã HTML, CSS và logic JavaScript riêng của nó. Điều này cho phép bạn tạo giao diện người dùng phức tạp và tùy chỉnh theo ý muốn. Bạn có thể viết mã HTML bên trong component để xác định cấu trúc và các phần tử giao diện, sử dụng CSS để tạo kiểu cho component, và triển khai logic JavaScript để xử lý các sự kiện và tương tác với dữ liệu.

- **Tương tác thông qua Props:** Components có thể truyền dữ liệu và tương tác thông qua Props (thuộc tính). Props là các đối số được truyền từ component cha đến component con. Điều này cho phép chia sẻ dữ liệu giữa các component và truyền thông tin từ component gốc xuống các component con trong cây component. Props giúp tạo giao tiếp linh hoạt và giữ cho các component độc lập và tái sử dụng được.

- **Tổ chức cây thành phần:** Các components có thể được tổ chức thành cây thành phần (component tree) theo cấu trúc phân cấp. Mỗi component có thể chứa các component con, tạo thành một cây phân cấp. Điều này giúp tổ chức và quản lý giao diện người dùng một cách có tổ chức và dễ dàng hiểu.

- **Lifecycle Methods:** React cung cấp các lifecycle methods (phương thức vòng đời) cho components, cho phép bạn thực hiện các hành động trong quá trình khởi tạo, cập nhật và hủy bỏ component. Điều này cho phép bạn thực hiện các tác vụ như khởi tạo dữ liệu, tải dữ liệu từ API, cập nhật giao diện, và giải phóng tài nguyên khi cần thiết.

b. Props

Props là các đối số được truyền từ component cha đến component con. Props cho phép truyền dữ liệu từ một component gốc xuống các component con trong cây component. Điều này giúp chia sẻ dữ liệu và truyền thông tin giữa các component.

- **Truyền dữ liệu giữa các component:** Props cho phép truyền dữ liệu từ component cha xuống các component con trong cây component. Bằng cách truyền các giá trị thông qua props, ta có thể chia sẻ thông tin giữa các component khác nhau và tái sử dụng chúng một cách dễ dàng. Khả năng truyền dữ liệu tùy chỉnh: cho phép truyền các đối tượng, mảng, hàm, hoặc các thành phần React khác.

- **Props là không thay đổi (immutable):** Props được coi là không thay đổi, tức là sau khi props được truyền vào một component, nó không thể bị thay đổi từ bên trong component đó. Điều này đảm bảo tính ổn định và dễ dàng kiểm tra sự thay đổi trong ứng dụng.

- **Sử dụng trong việc tạo các component tái sử dụng:** Props là công cụ quan trọng trong việc tạo các component tái sử dụng. Bằng cách truyền các giá trị và dữ liệu thông qua props, ta có thể xây dựng các component đa dạng và linh hoạt, có khả năng tái sử dụng trong nhiều phần của ứng dụng.

- **Xác định và sử dụng props trong component:** Để sử dụng props trong một component, ta chỉ cần truy cập vào các thuộc tính được định nghĩa trong đối tượng props của component đó. Props được truyền vào component thông qua việc khai báo các thuộc tính của component khi sử dụng nó.

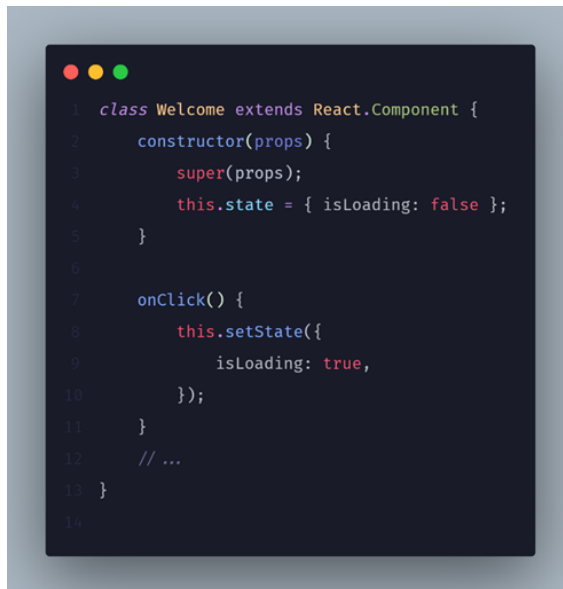
1.3.4. Một số hook thông dụng

Hooks chính thức được giới thiệu trong phiên bản React 16.8, nó cho phép functional component có thể quản lý các state bên trong nó và sử dụng các tính năng khác của React mà không cần dùng đến class component như các phiên bản trước đó. Đây được xem là cột mốc đánh dấu sự thay đổi lớn của thư viện React khi nó giúp cho những người mới tiếp cận dễ dàng sử dụng hơn, đồng thời không làm thay đổi cách thức hoạt động của React đối với những người đã có kinh nghiệm làm việc với class component.

Các hook xây dựng sẵn trong React được chia thành ba nhóm chính là hooks cơ bản, hooks bổ sung và hooks thư viện.

Trong nhóm hooks cơ bản sẽ có hai hook chính là useState và useEffect.

- **useState:** cho phép thêm state vào function component. Trước đây để quản lý state trong một component, ta sẽ phải sử dụng class component và định nghĩa nó trong phương thức constructor().



```

1  class Welcome extends React.Component {
2    constructor(props) {
3      super(props);
4      this.state = { isLoading: false };
5    }
6
7    onClick() {
8      this.setState({
9        isLoading: true,
10      });
11    }
12    // ...
13  }
14

```

Hình 4. Hook useState trong class component.

Với sự hỗ trợ của hook useState, functional component đã có thể quản lý state bên trong nó:



```

1  function Welcome(props) {
2    const [isLoading, setIsLoading] = useState(false);
3
4    function onClick() {
5      setIsLoading(true);
6    }
7    // ...
8  }

```

Hình 5. Hook useState trong functional component.

Ta có thể thấy functional component và useState hook làm cho mã nguồn trở nên ngắn gọn hơn, dễ hiểu và dễ quản lý hơn rất nhiều so với class component. Hook useState sẽ nhận tham số đầu vào là giá trị khởi tạo cho một state. Giá trị này có thể là kiểu dữ liệu bất kỳ và chỉ được sử dụng một lần duy nhất để gán cho state khi component được khởi tạo render. Hook useState sẽ trả về một mảng gồm hai phần tử. Phần tử đầu tiên là state hiện tại, trong lần render đầu tiên, giá trị này sẽ bằng với giá trị khởi tạo được truyền vào. Phần tử thứ hai là một hàm để cập nhật lại giá trị khác cho state và đồng thời thực hiện re-render lại giao diện.

- **useEffect**: giúp xử lý các side effects, hook này sẽ tương đương với các hàm `componentDidMount()`, `componentDidUpdate()` và `componentWillUnmount()` trong vòng đời của component.



```
1 class Example extends React.Component {
2   constructor(props) {
3     super(props);
4     this.state = {
5       count: 0,
6     };
7   }
8
9   componentDidMount() {
10    document.title = `You clicked ${this.state.count} times`;
11  }
12  componentDidUpdate() {
13    document.title = `You clicked ${this.state.count} times`;
14  }
15  // ...
16 }
17
```

Hình 6. Hook `useEffect` trong class component.

1.4. Tổng quan về NodeJS

1.4.1. NodeJS là gì ?

NodeJS là một mã nguồn mở, đa nền tảng, chạy trên môi trường JavaScript, được xây dựng trên V8 JavaScript engine của Chrome - V8 thực thi mã JavaScript bên ngoài trình duyệt. Nó được tạo ra vào năm 2009 đi kèm với một lợi thế chính - NodeJS cho phép thực hiện lập trình bất đồng bộ.

Ở chế độ đồng bộ thực thi từng dòng và tiến hành thực thi dòng tiếp theo khi dòng hiện tại đã thực thi xong. Khi bất đồng bộ thực thi tất cả dòng code cùng một lúc.

NodeJS là một nền tảng được xây dựng trên JavaScript runtime của Chrome với mục đích xây dựng các ứng dụng mạng nhanh chóng và có thể mở rộng được một cách dễ dàng hơn. NodeJS sử dụng mô hình I/O lập trình theo sự kiện, non-blocking, do đó nodeJS khá gọn nhẹ và hiệu quả - công cụ hoàn hảo cho các ứng dụng chuyên sâu về dữ liệu theo thời gian thực chạy trên các thiết bị phân tán.

NodeJS là môi trường runtime mã nguồn mở đa nền tảng, được sử dụng để phát triển các ứng dụng mạng và ứng dụng server-side. Các ứng dụng NodeJS được viết bằng JavaScript và có thể chạy trong NodeJS runtime trên OS X, Microsoft Windows và Linux.

NodeJS cũng cung cấp một thư viện bao gồm rất nhiều các module JavaScript khác nhau nhằm đơn giản hóa việc phát triển các ứng dụng web, qua đó giảm thiểu tình trạng sử dụng quá nhiều Node.js.

1.4.2. Các tính năng của NodeJS

Một số tính năng chính của NodeJS như:

- ***Non-blocking I/O***: NodeJS cho phép xử lý các yêu cầu I/O một cách không đồng bộ, cho phép ứng dụng tiếp tục hoạt động trong khi đợi các yêu cầu I/O phức tạp kết thúc.

- ***Single-threaded event loop***: NodeJS sử dụng mô hình xử lý sự kiện đơn luồng giúp cho việc xử lý các yêu cầu trở nên hiệu quả hơn.

- ***Các module có sẵn***: NodeJS đi kèm với rất nhiều module hữu ích có sẵn trong thư viện standard library, từ các module cơ bản như HTTP, HTTPS, File System đến các module phức tạp hơn như Socket.io, Express, Mongoose.

- ***NPM (Node Package Manager)***: NPM là công cụ quản lý gói phổ biến nhất trong cộng đồng NodeJS, cho phép bạn dễ dàng cài đặt và quản lý các gói phần mềm của NodeJS.

- ***Phù hợp với các ứng dụng real-time***: NodeJS được xây dựng để xử lý các ứng dụng real-time có tính tương tác cao như chat, trò chơi trực tuyến, và các ứng dụng web thời gian thực khác.

- ***Dễ dàng bảo trì***: Do cùng sử dụng ngôn ngữ JavaScript cho cả frontend và backend, điều này giúp việc bảo trì và phát triển ứng dụng trở nên dễ dàng hơn.

1.4.3. Ứng dụng của NodeJS

NodeJS có một số ứng dụng chính như:

- ***Phát triển ứng dụng web***: Với ExpressJS, NestJS hoặc MeteorJS, NodeJS được sử dụng để xây dựng phần server-side của các ứng dụng web.

- ***Xây dựng API***: NodeJS cung cấp một thư viện HTTP để xây dựng các API RESTful và giao tiếp với các hệ thống khác thông qua các giao thức như JSON hay XML.

- ***Các công cụ dòng lệnh (CLI)***: NodeJS có thể được sử dụng để xây dựng các công cụ dòng lệnh như gulp, grunt hay webpack.

- **Các ứng dụng desktop:** ElectronJS là một framework cho phép xây dựng các ứng dụng desktop bằng HTML, CSS và JavaScript, NodeJS là một phần quan trọng của ElectronJS.

- **Xử lý ảnh:** SharpJS là một thư viện xử lý ảnh sử dụng NodeJS, nó hỗ trợ nhiều định dạng ảnh khác nhau.

- **Xử lý dữ liệu lớn:** NodeJS có khả năng xử lý dữ liệu lớn và thực hiện các tính toán phức tạp trên server-side.

1.5. Các thư viện hỗ trợ

Để hỗ trợ trong việc nghiên cứu và phát triển hệ thống truy xuất nguồn gốc vật nuôi, chúng tôi sẽ sử dụng một số thư viện liên quan đến từng công nghệ để hỗ trợ phát triển như sau:

1.5.1. Axios

Axios là một thư viện HTTP client được sử dụng phổ biến trong Javascript. Nó được sử dụng để tương tác với các API bên ngoài và lấy dữ liệu từ server về để sử dụng trong ứng dụng web.

Một số tính năng của Axios:

- **Promise-based:** Axios sử dụng các promise để xử lý các yêu cầu HTTP, cho phép bạn xử lý các kết quả thành công hoặc thất bại của yêu cầu.

- **Interceptors:** Axios cung cấp interceptors để giúp bạn can thiệp vào các yêu cầu trước khi chúng được gửi, và các kết quả sau khi nhận được từ server.

- **Cancellation:** Axios hỗ trợ việc hủy bỏ một yêu cầu HTTP, cho phép bạn ngừng một yêu cầu đang chạy nếu không cần thiết.

- **Automatic JSON data transformation:** Axios tự động chuyển đổi dữ liệu từ JSON sang JavaScript object và ngược lại, giúp tiết kiệm thời gian và công sức trong việc xử lý các định dạng dữ liệu khác nhau.

- **Support for common browsers and platforms:** Axios hỗ trợ các trình duyệt phổ biến và các nền tảng khác nhau bao gồm Node.js, React Native và Electron.

- **Easy to use:** Axios rất dễ sử dụng và có thể được sử dụng cho các yêu cầu HTTP đơn giản hoặc phức tạp.

Thư viện Axios rất hữu ích trong việc tương tác với các API bên ngoài và lấy dữ liệu từ server về để sử dụng trong ứng dụng web. Nó được coi là một trong những thư viện HTTP client phổ biến nhất trong Javascript.

1.5.2. React-router-dom

React Router Dom là một thư viện ReactJS được sử dụng để quản lý định tuyến trong ứng dụng web. Thư viện này cung cấp cho bạn các thành phần để tạo và quản lý các route, nhằm giúp người dùng điều hướng trang web một cách dễ dàng và hiệu quả.

Một số tính năng của React-router-dom:

- **Declarative routing:** React Router Dom cho phép bạn khai báo các route trong JSX một cách rõ ràng và đơn giản, giúp cho việc quản lý đường dẫn trở nên dễ dàng hơn.

- **Nested routing:** React Router Dom hỗ trợ các route lồng nhau, cho phép bạn xây dựng các ứng dụng web phức tạp với nhiều trang con.

- **Dynamic routing:** React Router Dom có thể tạo ra các route dựa trên các thông tin được truyền vào từ URL, cho phép bạn xây dựng các ứng dụng web linh hoạt và tùy chỉnh hơn.

- **History management:** React Router Dom quản lý lịch sử duyệt web của người dùng, cho phép bạn điều hướng trang web một cách nhanh chóng và hiệu quả.

- **Server-side rendering:** React Router Dom hỗ trợ tính năng server-side rendering, cho phép ứng dụng của bạn tải nhanh hơn và tối ưu hóa cho công cụ tìm kiếm.

- **Large community support:** React Router Dom có một cộng đồng phát triển lớn, cung cấp nhiều tài liệu hướng dẫn và các ví dụ thực tế để giúp bạn phát triển các ứng dụng web nhanh chóng và dễ dàng.

React Router Dom là một thư viện quan trọng trong việc phát triển các ứng dụng web với ReactJS. Nó giúp cho việc quản lý định tuyến trong ứng dụng web trở nên dễ dàng hơn và cung cấp nhiều tính năng tiện ích giúp tối ưu hóa trải nghiệm người dùng.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ

Yêu cầu của ứng dụng khi được thiết kế, phải đáp ứng được những tiêu chí như: Tính tiện dụng, app hiển thị thông tin cần dễ đọc, dễ hiểu, dễ dùng. Thiết kế phải mới, tuy nhiên không quá phức tạp gây khó chịu cho người dùng.

Ứng dụng mang đến những thông tin về các sản phẩm dê cừu khi được quét mã QR vì vậy trang thông tin cần hiển thị được những hình rõ ràng hợp lý, tiện ích khi nhìn vào người dùng nhanh chóng tìm được nội dung mong muốn.

Hiển thị thông tin chi tiết quá trình chăm sóc từ khi bắt đầu nuôi đến khi thu hoạch, quá trình ra sao, được chăm sóc như thế nào (chi tiết từ ăn uống, chăn thả, điều trị bằng vắc-xin, thuốc), các thông tin thu hoạch, xử lý đóng gói sản phẩm, phân phối.

Ngoài ra còn có trang web quản trị để admin thực hiện quản lý chăm sóc, quản lý vật nuôi trong quá trình phát triển nhằm lưu trữ dữ liệu cho hệ thống truy xuất.

2.1. Phân tích và mô tả chức năng hệ thống

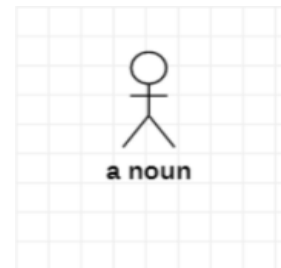
2.1.1. Về sơ đồ use case

- Sơ đồ Use Case (UC) là sơ đồ tính năng của sản phẩm cung cấp cho người dùng. Giúp người dùng hiểu được các chức năng của sản phẩm.

- Sơ đồ UC bao gồm:

- **Tác nhân (Actor)**

- Actor là một thực thể trong yêu cầu phần mềm.
- Một sơ đồ UC có thể có nhiều Actor.
- Một người dùng có thể thực hiện được nhiều chức năng của các Actor khác nhau.
- Thường là danh từ.



- **Hành động (Use Case)**

- Use Case là các chức năng của phần mềm được Actor sử dụng.
- Thường là động từ.



- Các quan hệ trong sơ đồ use case

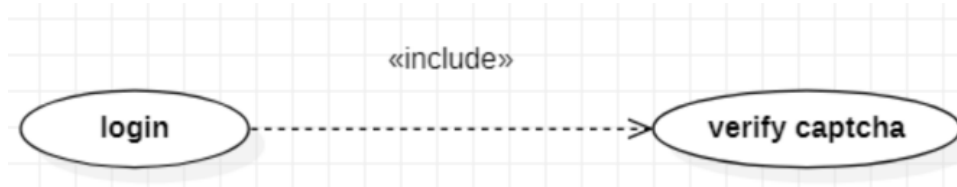
- **Liên kết (Association):** Thường dùng để chỉ mối quan hệ giữa Actor với Use Case hoặc giữa các Use Case với nhau.



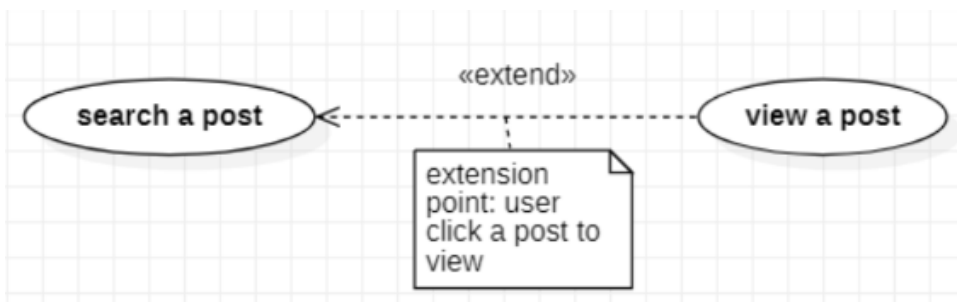
- **Tính khái quát hóa (Generalization):** Là quan hệ kế thừa, chỉ quan hệ giữa đối tượng con với đối tượng cha (Actor).



- **Bao gồm (Include):** Thường dùng giữa các Use Case, mô tả một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa).



- **Mở rộng (Extend):** Extend dùng để mô tả quan hệ giữa 2 Use Case, được sử dụng khi có một Use Case được tạo ra để bổ sung chức năng cho một Use Case có sẵn.



- **Ranh giới hệ thống (System Boundary):** Là đường biên xác định phạm vi của thiết kế. Các đối tượng nằm ngoài phạm vi này tương tác với phần mềm là Actor.

2.1.2. Danh sách actor

Bảng 1. Danh sách actor

STT	Tên actor	Mô tả
1	Không phải thành viên (NonMember)	Người sử dụng ứng dụng di động quét mã QR, truy xuất ra thông tin

2	Thành viên (Member)	• Người được lưu giữ thông tin cá nhân (họ tên, địa chỉ, số điện thoại, email,...), thành viên có mật khẩu (password) để truy cập vào hệ thống để quản lý, chăm sóc các vật nuôi của họ. • Quản lý các thông tin thông tin thu hoạch, xử lý/đóng gói sản phẩm, phân phối. • Kế thừa các chức năng từ không phải thành viên (NonMember).
3	Quản trị viên (Admin)	• Người quản trị viên quản lý danh sách các tài khoản tài khoản, danh mục vật nuôi, danh sách thuốc, trang trại, bệnh

2.1.3. Danh sách use case

- U1: Duyệt qua chỉ mục (Browse Index): Một người dùng duyệt các mã QR. (Chi tiết hóa bởi U46, bao gồm U2)

- U2: Xem thông tin sản phẩm (View): Một người dùng sẽ xem được thông tin sản phẩm đã được quét mã. (Bao gồm U1 và U4, mở rộng U3)

- U3: Xem chi tiết sản phẩm (View Product Details): Một người dùng sẽ xem chi tiết về nhật ký chăm sóc, vùng trồng sản phẩm, các phương pháp điều trị, đóng gói. (Mở rộng bởi U2, mở rộng U12)

- U4: Tìm kiếm (Search): Tìm kiếm các sản phẩm từ người dùng thông qua tên, loại, vùng trồng sản phẩm và mã QR code. (Chi tiết hóa bởi U46, mở rộng bởi U5, U6)

- U5: Tìm kiếm các sản phẩm: Thông tin nổi bật của sản phẩm như tên, loại, vùng trồng để người dùng tìm kiếm. (Mở rộng U4)

- U6: Mã QR (QR Code): mã QR của các sản phẩm dê/cừu Ninh Thuận. (Mở rộng U4)

- U7: Khôi phục mật khẩu (Recover password).

- U8: Đăng nhập (Login)

- U9: Thay đổi mật khẩu (Change password). (Mở rộng U8)

- U10: Chỉnh sửa thông tin cá nhân (Update personal information). (Mở rộng U8)

- U11: Thông báo (Notifications). (Mở rộng U8)

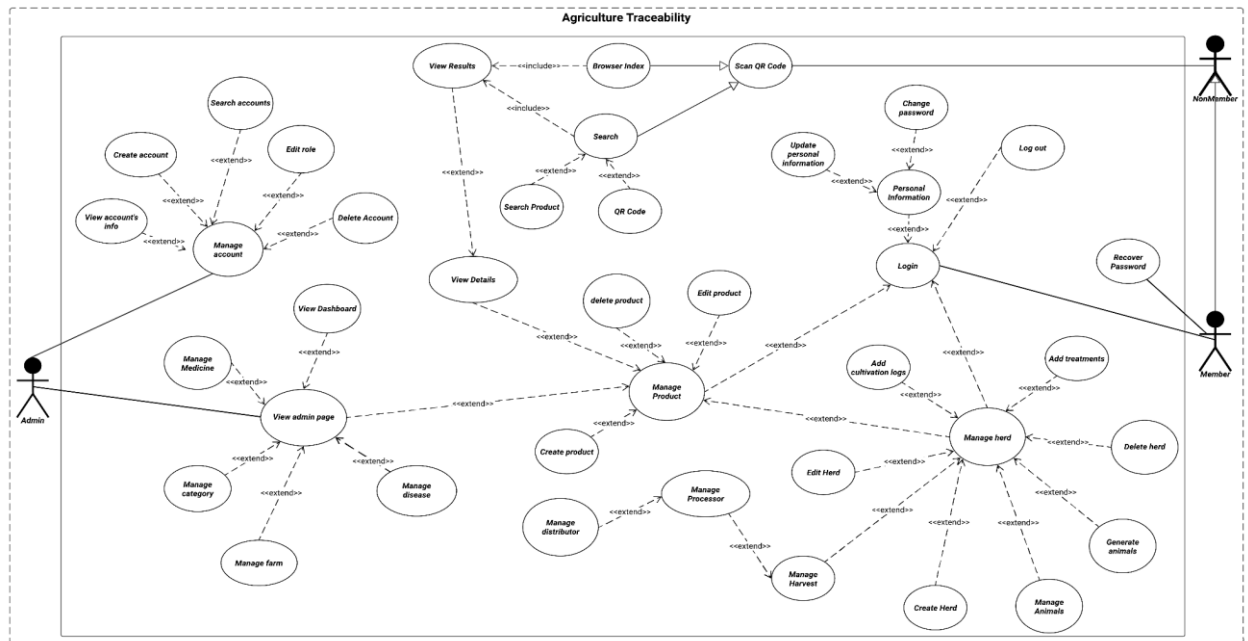
- U12: Quản lý các sản phẩm đàn vật nuôi (Herd Management): Người dùng quản lý thông tin, chăm sóc của các sản phẩm. (Mở rộng bởi U3, U14, U15, U16, U17, U18, Bao gồm U19)

- U13: Đăng xuất (Log out). (Mở rộng U8)

- U14: Nhật ký chăm sóc (Cultivation Logs): Toàn bộ hoạt động của các sản phẩm từ trước đến hiện tại hoặc đã kết thúc. (Mở rộng U12)
- U15, U16, U17: Lần lượt là thêm, xóa, chỉnh sửa sản phẩm đàn vật nuôi (Create, Delete, Edit Herd). (Mở rộng U12)
- U18: Xem trang quản trị (View admin page): Quản lý bảng điều khiển, các loại thuốc, nhóm, bệnh. (Mở rộng bởi U20, U21, U22, U23)
- U19: Xem bảng điều khiển (View Dashboard): Các thông tin chung mà người dùng quản lý. (Mở rộng U19)
- U20: Quản lý thuốc (Medicine Management). (Mở rộng bởi U24, U25, U26, U27)
- U21: Quản lý nhóm vật nuôi (Category Management). (Mở rộng bởi U28, U29, U30, U31)
- U22: Quản lý trang trại (Farm Management). (Mở rộng bởi U32, U33, U34, U35)
- U23: Quản lý bệnh (Diseases Management). (Mở rộng bởi U37, U38, U39, U40)
- U24, U25, U26, U27: Lần lượt là thêm, xóa, sửa, xem chi tiết thuốc (Create, Delete, Edit Medicine và View Medicine details).
- U28, U29, U30, U31: Lần lượt là thêm, xóa, sửa, xem chi tiết nhóm (Create, Delete, Edit Category và View Category details).
- U32, U33, U34, U35: Lần lượt là thêm, xóa, sửa, xem chi tiết cơ sở (Create, Delete, Edit Farm và View Farm details).
- U36, U37, U38, U39: Lần lượt là thêm, xóa, sửa, xem chi tiết bệnh (Create, Delete, Edit Diseases và View Diseases details).
- U40, U41, U42, U43: Lần lượt là thêm, xóa, sửa, xem chi tiết thông tin thu hoạch (Create, Delete, Edit Harvest và View Harvest details)
- U44, U45, U46, U47: Lần lượt là thêm, xóa, sửa, xem chi tiết thông tin xử lý, đóng gói (Create, Delete, Edit Processor và View Processor details)
- U48, U49, U50, U51: Lần lượt là thêm, xóa, sửa, xem chi tiết thông tin sản phẩm (Create, Delete, Edit Product Info và View Product Info details)
- U52: Quản lý tài khoản (Account Management): Người quản trị quản lý thêm, xóa và vai trò của các tài khoản. (Mở rộng bởi U55, U56)
- U53: Chỉnh sửa vai trò (Edit role): Chỉnh sửa vai trò của tài khoản gồm quản trị (admin) hoặc thành viên (user). (Mở rộng U52)

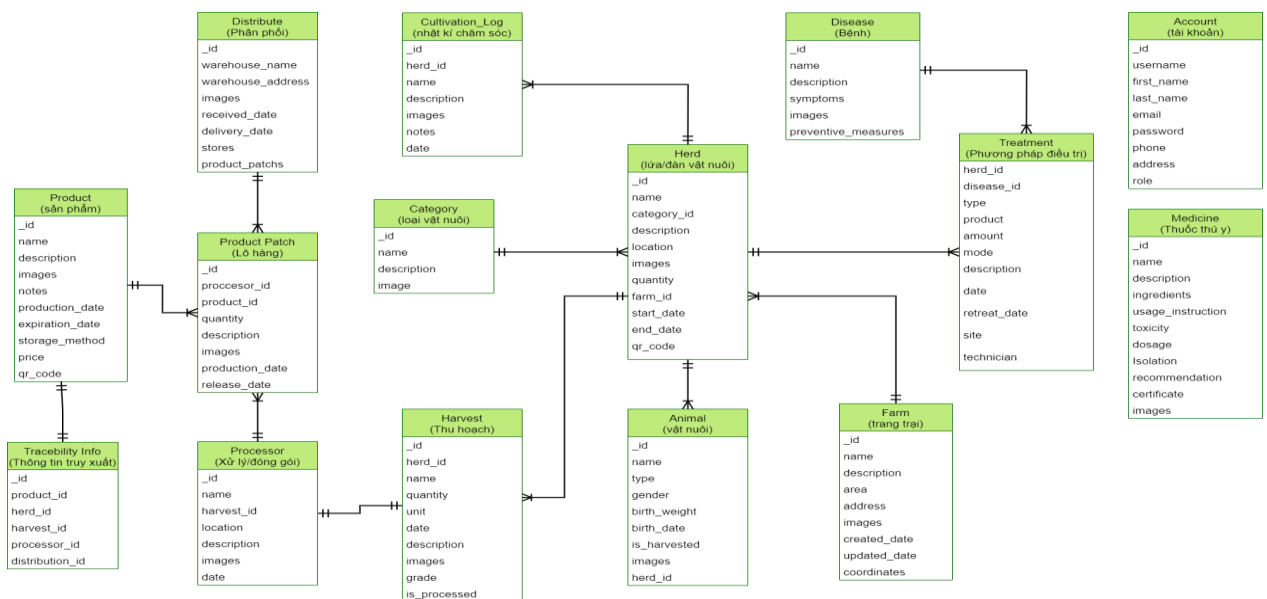
- U54: Tìm kiếm tài khoản (Search account): Tìm kiếm tài khoản thực hiện các chức năng khác. (Mở rộng U52)
- U55, U56: Lần lượt là thêm, xóa các tài khoản (Create, Delete Account). (Mở rộng U52)
- U57: Quét mã QR (Scan QR code): Người dùng quét mã Qr từ các sản phẩm vật nuôi. (Khái quát hóa bởi U1 và U4)

2.1.4. Lược đồ use case



Hình 7. Sơ đồ Use Case

2.2. Cơ sở dữ liệu



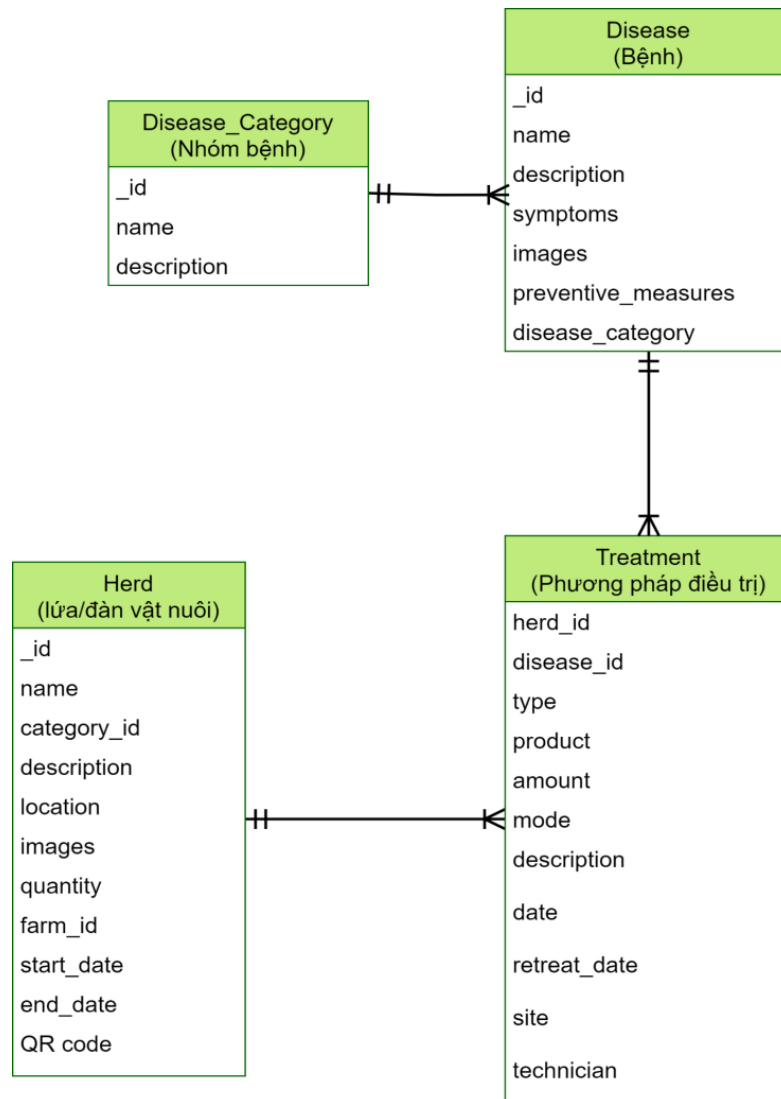
Hình 8. Cơ sở dữ liệu

2.2.1 Bảng dữ liệu

- **Product**(id, name, herd, description, images, notes, production_date, expiration_date, qrcode)
- **Traceability_Info**(_id, product_id, herd_id, harvest_id, processor_id, distributor_id)
- **Farm**(_id, name, description, image, area, address, coordinates, created_date, updated_date)
- **Category**(_id, name, description, image)
- **Herd**(_id, name, category, description, location, images, farm, quantity, start_date, end_date, qrcode)
- **Animal**(_id, name, type, gender, birth_weight, birth_date, is_harvested, images, herd)
- **Cultivation_Log**(_id, herd_id, date, images, notes)
- **Disease**(_id, name, description, images, symptoms, prevent_measures)
- **Medicine**(_id, name, description, ingredients, usage_instruction, toxicity, dosage, isolation, recommendation, certificates, images)
- **Treatment**(herd_id, disease_id, type, product, amount, mode, description, date, retreat_date, site, technician)
- **Harvest**(_id, herd_id, name, quantity, unit, date, description, grade, isProcessed, images)
- **Processor**(_id, name, harvest_id, location, description, date, images)
- **Product_Patch**(_id, product_id, processor_id, quantity, description, production_date, release_date, images)
- **Distributor**(_id, warehouse_name, warehouse_address, received_date, delivery_date, stores, images)
- **Account**(_id, username, first_name, last_name, email, password, phone, address, role)

2.2.1. Các quan hệ

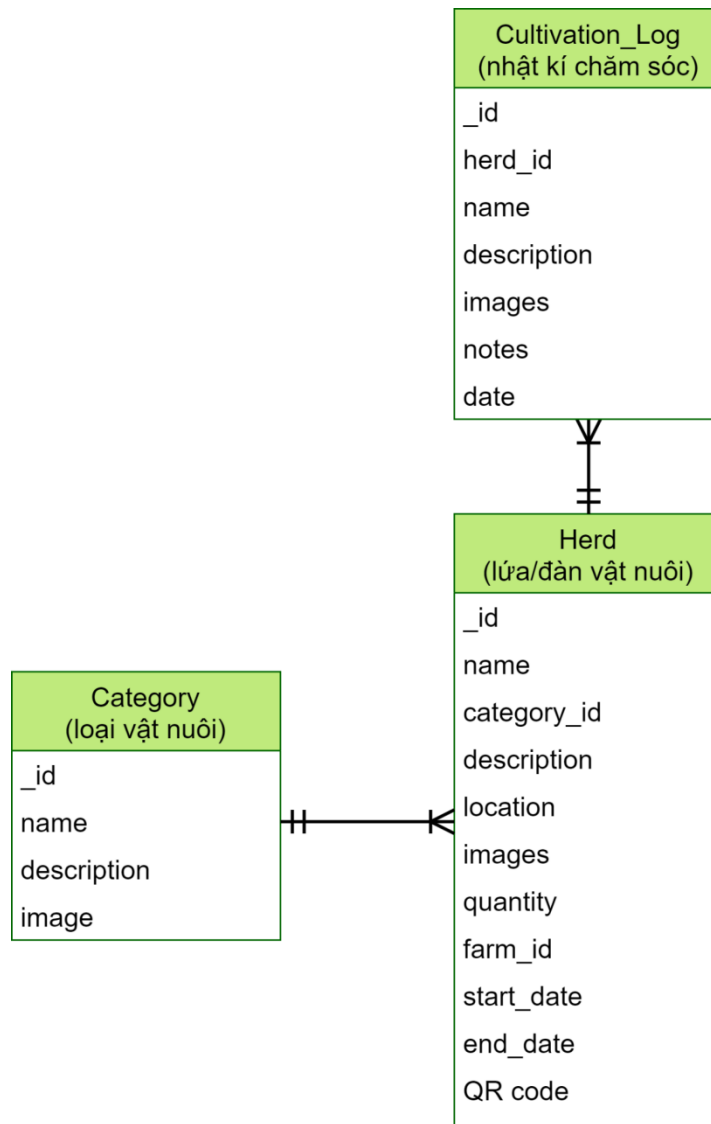
- **Quan hệ nhóm bệnh và bệnh**: Một nhóm bệnh có một hay nhiều bệnh.
- **Quan hệ giữa bệnh và phương pháp điều trị**: Một bệnh có 1 hay nhiều phương pháp điều trị; Một phương pháp điều trị thì chỉ có nhiều bệnh.
- **Quan hệ giữa vật nuôi và phương pháp điều trị**: Một vật nuôi thì có nhiều phương pháp điều trị; Một phương pháp điều trị chỉ cho nhiều vật nuôi.



Hình 9. Cơ sở dữ liệu chi tiết 1

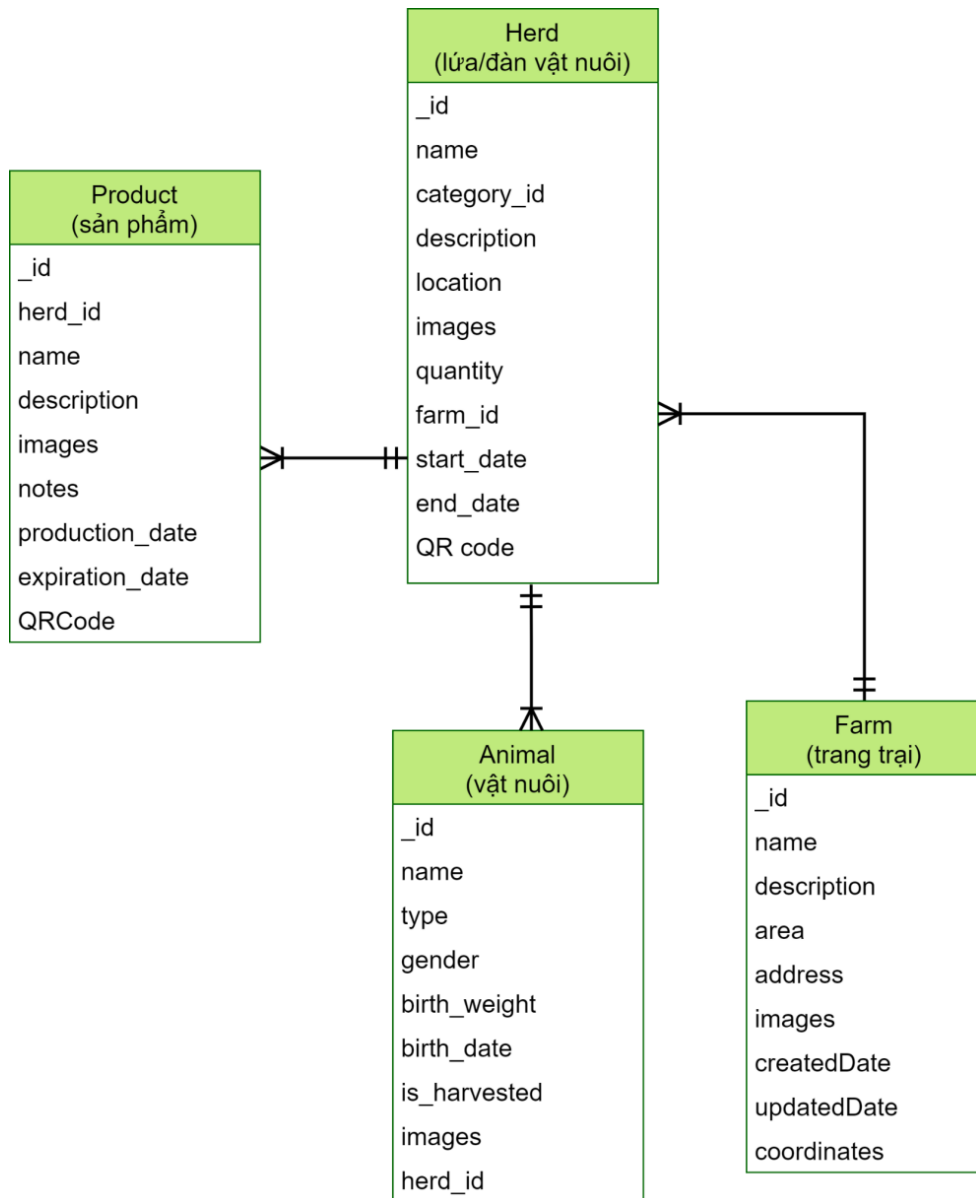
- **Quan hệ giữa nhật ký chăm sóc và lúa đàn vật nuôi:** Một lúa đàn vật nuôi có nhiều nhật ký chăm sóc; Một nhật ký chăm sóc thì chỉ cho một lúa đàn vật nuôi (tại một thời điểm chăm sóc khác nhau).

- **Quan hệ loại vật nuôi và lúa đàn vật nuôi:** Một loại vật nuôi có một hay nhiều lúa đàn vật nuôi; Một lúa đàn vật nuôi thì thuộc duy nhất một loại vật nuôi.



Hình 10. Cơ sở dữ liệu chi tiết 2

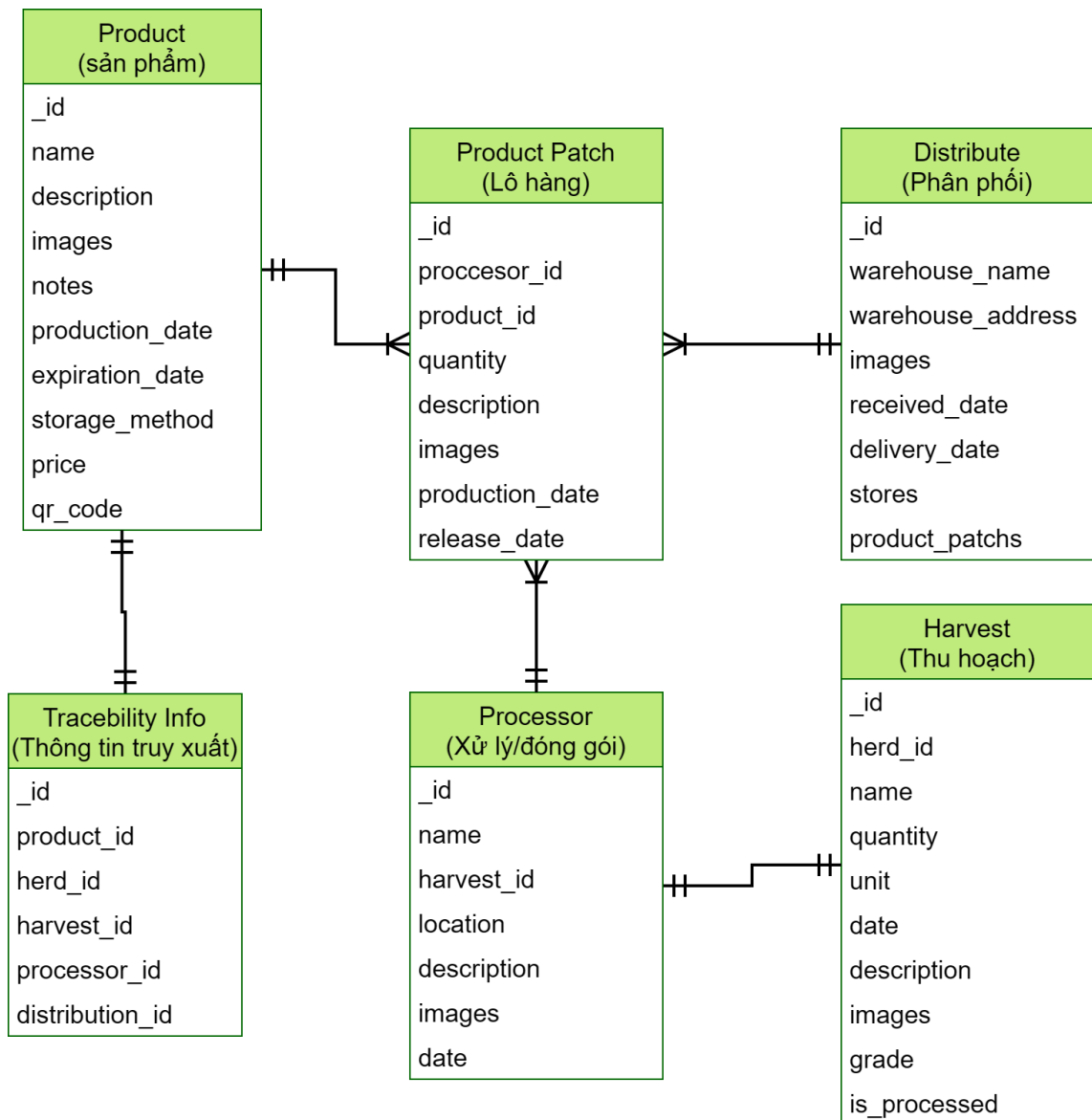
- **Quan hệ giữa sản phẩm và lứa đàn vật nuôi:** Một lứa đàn vật nuôi tạo ra nhiều sản phẩm (lông, sữa, thịt,...); Một sản phẩm chỉ được tạo ra từ một lứa đàn vật nuôi.
- **Quan hệ giữa vật nuôi và lứa đàn vật nuôi:** Một lứa đàn vật nuôi có nhiều vật nuôi; vật nuôi ở đây lưu trữ các thông tin như tên, loại, giới tính, cân nặng,...; Một vật nuôi chỉ thuộc một lứa đàn vật nuôi.
- **Quan hệ giữa trang trại và lứa đàn vật nuôi:** Một trang trại có nhiều lứa đàn vật nuôi; vật nuôi ở đây lưu trữ các thông tin như tên, loại, giới tính, cân nặng,...; Một lứa đàn vật nuôi chỉ thuộc một trang trại.



Hình 11. Cơ sở dữ liệu chi tiết 3

- Quan hệ thu hoạch và xử lý đóng gói là một một một; thu hoạch những gì thì đóng gói những thứ đó.
- Mỗi quan hệ giữa xử lý đóng gói và lô hàng (lô sản phẩm) là một xử lý đóng gói có nhiều lô hàng (tương đương 1 sản phẩm thu hoạch).
- Mỗi quan hệ giữa phân phối và lô hàng là một phân phối có nhiều lô hàng.
- Ngược lại một lô hàng chỉ có một phân phối. Một kho gồm lô hàng chứa các sản phẩm thu hoạch trước đó thì sẽ được phân phối tới một cửa hàng bán trên thị trường.
- Lô hàng khi được tạo sẽ lấy xử lý đóng gói của sản phẩm thu hoạch, sau khi nhận lô hàng cùng lúc mã QR được tạo gán cho sản phẩm và infor (thông tin) sản phẩm (nhật ký chăm sóc, điều trị,...) cũng được tạo và gán cho sản phẩm.

- Mỗi quan hệ giữa lô hàng và sản phẩm là một sản phẩm có trong nhiều lô hàng với số lượng khác nhau.
- Mỗi quan hệ giữa sản phẩm và thông tin truy xuất là mối quan hệ 1-1. Một sản phẩm chỉ duy nhất một thông tin.



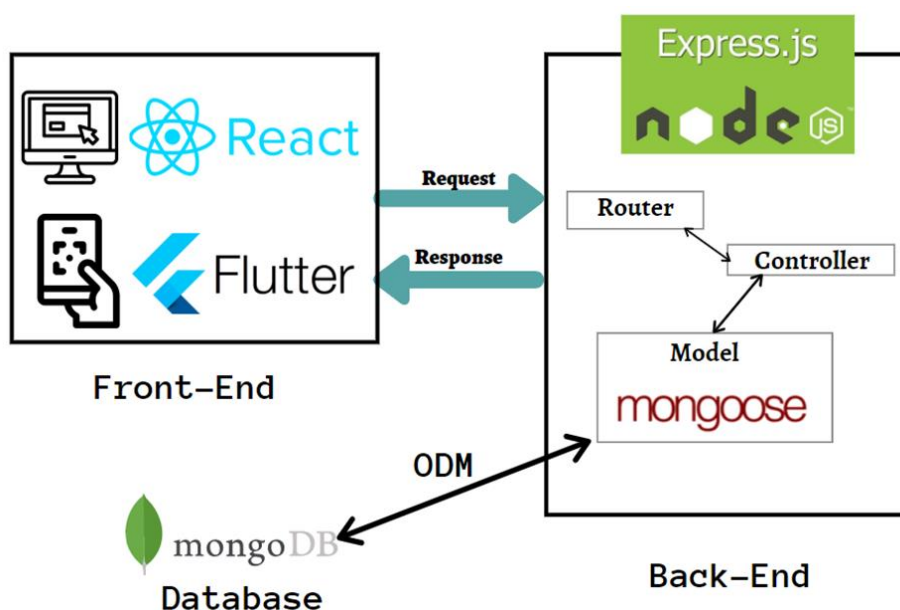
Hình 12. Cơ sở dữ liệu chi tiết 4

CHƯƠNG 3. TRIỂN KHAI ỨNG DỤNG

3.1. Kiến trúc hệ thống

Kiến trúc hệ thống được chia thành 3 phần:

- **Phần Front-End:** sử dụng React đối với web và Flutter đối với ứng dụng di động.
- **Phần Back-End:** sử dụng framework Express chạy trên môi trường Node.js và thư viện Mongoose sẽ định nghĩa các document và truy vấn tới hệ quản trị cơ sở dữ liệu MongoDB.
- **Hệ quản trị cơ sở dữ liệu:** sử dụng MongoDB



Hình 13. Mô hình kiến trúc hệ thống

Trong quá trình phân tích, chúng tôi nhận thấy mô hình trên đáp ứng được yêu cầu đề ra của dự án. Luồng xử lý của hệ thống, đầu tiên bên phía Client (React) sẽ gửi một yêu cầu (request) bất đồng bộ (AJAX). (GET, POST, PUT, DELETE, ..) kèm theo một số thông tin kèm theo (XML, Json) đến Server.

Server (Express Web Framework) sẽ nhận yêu cầu và xử lý, truy vấn đến hệ quản trị cơ sở dữ liệu

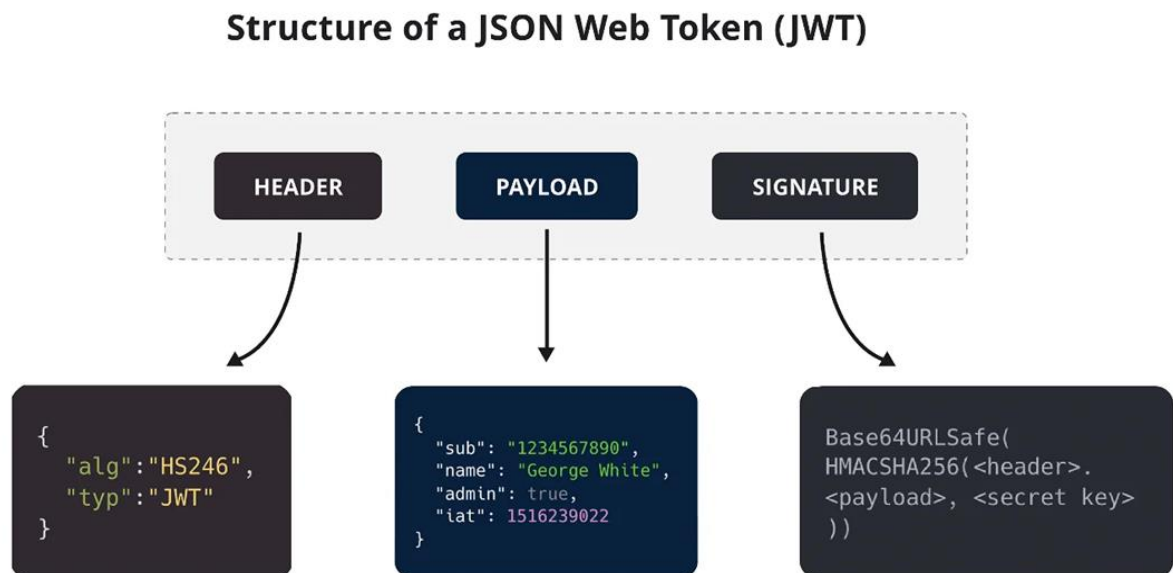
MongoDB (sử dụng Mongoose) và phản hồi (response) mã trạng thái thành công hay thất bại và kèm theo dữ liệu trả về (XML, Json) đến Client.

3.2. Xác thực API với Json Web Token (JWT)

3.2.1. Json Web Token là gì?

JWT (JSON Web Token) là một tiêu chuẩn mã nguồn mở dùng để truyền tải thông tin an toàn, gọn nhẹ và khép kín giữa các bên tham gia dưới định dạng JSON. JWT được sử dụng cho xác thực và chia sẻ thông tin. Thông tin được chia sẻ trong JWT được xác thực và tin cậy thông qua chữ ký số (Digital signature).

JWT gồm 3 phần chính, và phần tách nhau bằng một dấu chấm (.): Header.Payload.Signature.



Hình 14. Cấu trúc của Json Web Token (JWT)

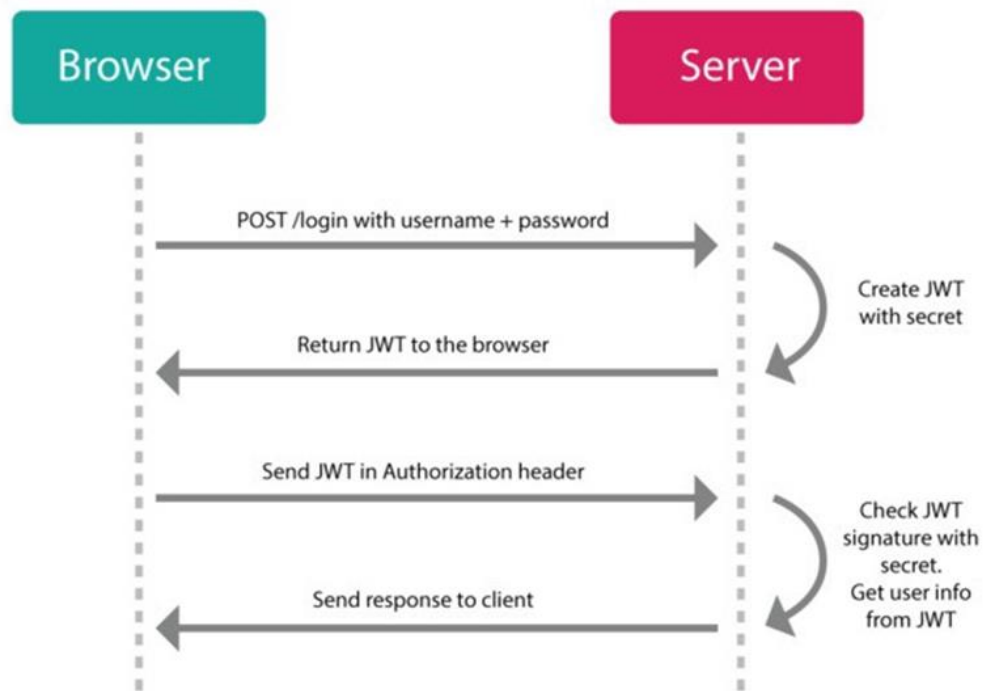
- Header thông thường sẽ bao gồm 2 thành phần, gồm: Kiểu token, mà với trường hợp này luôn luôn là JWT, và thuật toán mã hoá được sử dụng, ở đây là HS256.
- Payload trong JWT chứa các claims. Claims đề cập đến sự tuyên bố quyền truy cập hoặc quyền sử dụng tài nguyên.(ví dụ: user_id) và claims sẽ có dạng Key - Value.
- Signature là phần cuối của JWT, có chức năng xác thực danh tính người gửi. Để tạo ra một signature chính xác, ta cần encode header, payload, chọn mật mã khóa thích hợp đã được xác định ở header kèm secret key và thực hiện sign.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaXNjb2NpYWwiOiJhbnRydWV9.4pcPyMD09o1PSyXnrXCjTwXyr4BsezDI1AVTmud2fU4
```

Hình 15. Ví dụ về một JWT

3.2.2. Sơ đồ hoạt động

Sơ đồ sau cho thấy cách sử dụng JWT để truy cập API và tài nguyên:



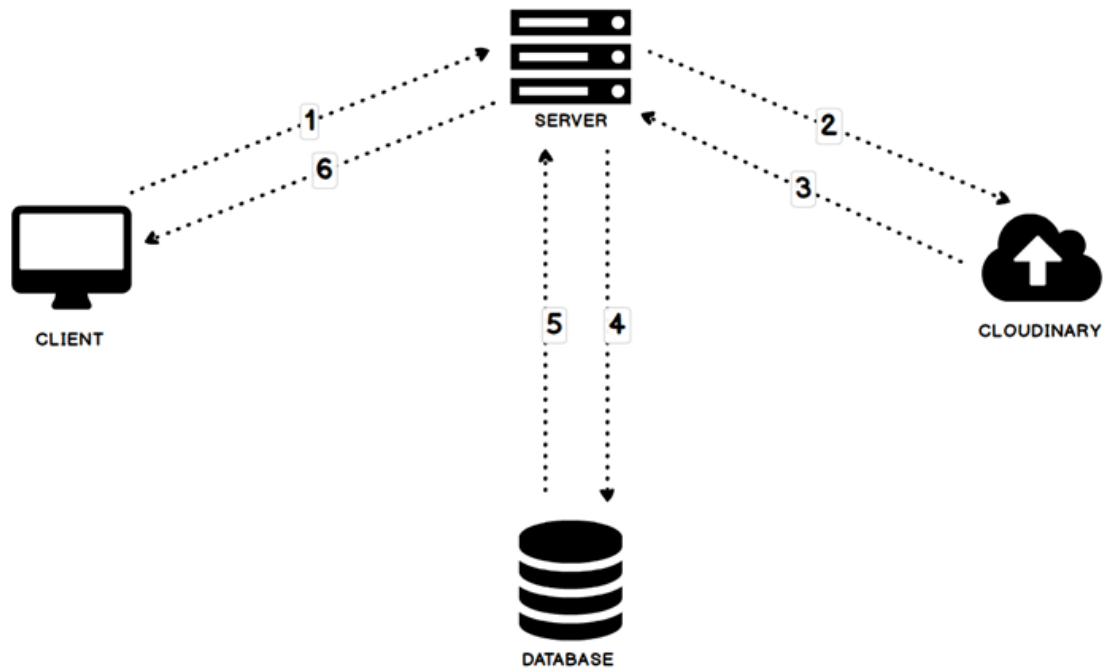
Hình 16. Quá trình hoạt động của json web token

Hình trên mô tả quá trình hoạt động của JWT, khi user truy cập `/login` từ browser và gửi một HTTP request gồm 2 params là username và password thì phía server sẽ tạo JWT gửi lại cho browser. Sau đó browser sẽ set JWT vào Authorization header (lúc này JWT cũng được lưu vào location storage hay là session storage).

Sau khi JWT đã được gán vào Authorization header thì server sẽ dựa vào đó mà xác thực token user bằng những thông tin từ Signature. Nếu token (user) là hợp lệ thì sẽ có quyền truy cập vào những private API.

3.3. Xử lý upload ảnh lên Cloudinary

Cloudinary là dịch vụ đám mây cung cấp các giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác và tối ưu hoá.



Hình 17. Minh họa hoạt động khi upload ảnh lên cloudinary

Quá trình khi thực hiện upload ảnh lên Cloudinary:

1. Người dùng gửi yêu cầu để tải ảnh lên máy chủ
2. Máy chủ sẽ thực hiện tải ảnh lên Cloudinary
3. Cloudinary sẽ lưu ảnh và trả về cho máy chủ đường dẫn hình ảnh
4. Máy chủ chèn đường dẫn hình ảnh vào cơ sở dữ liệu
5. Hình ảnh đã được lưu trong cơ sở dữ liệu và được cấp một id
6. Khi thành công server sẽ phản hồi cho người dùng rằng đã tải ảnh thành công

3.4. Triển khai hệ thống

Về việc vận hành hệ thống, ứng dụng sẽ sử dụng REST API giúp thao tác với dữ liệu hệ thống đã được triển khai (<https://agriculture-traceability.vercel.app/>) để thực hiện. Quá trình diễn ra như sau:

- Về ứng dụng web:

- Người quản trị viên: quản lý các thông tin về tài khoản, bệnh, thuốc, trang trại, danh mục vật nuôi.
- Người dùng khi quản lý quy trình chăm sóc đàn vật nuôi, các thông tin như nhật ký chăm sóc, phương pháp điều trị, thông tin thu hoạch, xử lý đóng gói, lô sản phẩm, sản phẩm, phân phối.

- Về ứng dụng di động: người dùng dùng ứng dụng di động để thực hiện quét mã QR đã được in trên bao bì sản phẩm, ứng dụng sẽ hiển thị thông tin truy xuất cho người tiêu

dùng xem. Các thông tin sẽ được hiển thị bao gồm: Trang trại, lứa/đàn vật nuôi, nhật ký chăm sóc, các điều trị đã thực hiện (vắc-xin, bệnh, phối giống,...), thu hoạch, xử lý đóng gói, phân phối sản phẩm.

3.4.1. API

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource.

Qua quá trình phân tích, hệ thống truy xuất yêu cầu phải là một API có tốc độ xử lý nhanh, hiệu suất cao, để đáp ứng yêu cầu này, chúng tôi chọn triển khai một REST API hỗ trợ việc truy xuất, quản lý dữ liệu nông sản giúp việc biểu diễn kết quả nhận diện được trực quan hơn. REST API sử dụng cơ sở dữ liệu là MongoDB để lưu trữ các dữ liệu nông sản. Về công nghệ, chúng tôi sử dụng NodeJs, ExpressJS để triển khai REST API.

Bảng 2. Danh sách API

API	Phương thức	Đường dẫn
Quét mã QR	GET	https://agriculture-traceability.vercel.app/api/v1/qrcode/:id
Sản phẩm nông sản	GET	https://agriculture-traceability.vercel.app/api/v1/processors/products
Thông tin sản phẩm	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/product-infos
Cơ sở	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/farms
Loại vật nuôi	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/categories
Lứa/đàn	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/herds
Vật nuôi	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/animals

Nhật ký chăm sóc	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/cultivation-logs
Điều trị	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/treatments
Thu hoạch	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/harvests
Xử lý/đóng gói	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/processors
Bệnh	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/diseases
Thuốc	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/medicines
Người dùng (user, admin)	GET, POST, PATCH, DELETE	https://agriculture-traceability.vercel.app/api/v1/users
Đăng nhập Đăng kí	GET, POST	https://agriculture-traceability.vercel.app/api/v1/login(register, logout)

3.4.2. Ứng dụng di động

Đầu tiên, import các gói cần thiết, bao gồm Flutter framework, mobile_scanner (để quét mã QR), và qr_scanner_overlay (để tạo giao diện trình quét QR)... Sau đó sử dụng thư viện đã thêm vào để viết hàm quét mã QR.

Sau khi quét mã QR, dữ liệu mã QR là mã ID của sản phẩm sẽ được lưu vào trong biến code và được thực hiện API để lấy dữ liệu.

Nếu quét mã QR không thành công thì sẽ chuyển sang trang Error Screen để thông báo với người dùng rằng đã gặp lỗi như chưa kết nối mạng, quét sai mã QR. Nếu mã quét thành công thì trang sẽ lấy dữ liệu và hiển thị lên giao diện người dùng để người dùng có thể xem các thông tin của sản phẩm bao gồm quy trình sản xuất, vùng trồng và các thông tin chi tiết bên trong đó.

3.4.3. Ứng dụng web

a. Xử lý lấy dữ liệu từ API

```
const fetchData = async (value = "") => {
  try {
    const response = await fetch(
      `/herds?limit=${currentLimit}&page=${currentPage}&searchQuery=${encodeURIComponent( value)}`);
    const data = await response.json();
    data.herds.forEach((element) => {
      element.date = <DateConverter originalDate={element.start_date} />;
    });
    setProducts(data.herds);
    setTotalPages(data.totalPages);
  } catch (error) {
    console.error("Error", error);
  }
};
```

Hình 18. Hàm xử lý lấy nội dung API

Hàm fetchData là một hàm bất đồng bộ (asynchronous) trong JavaScript, được sử dụng để gửi yêu cầu HTTP đến một địa chỉ /herds trên máy chủ để lấy dữ liệu. Đối số value được truyền vào hàm để tìm kiếm dữ liệu cụ thể.

Một yêu cầu fetch được gửi đến địa chỉ /herds, với các tham số như currentLimit, currentPage, và value để tìm kiếm. Sau khi nhận được phản hồi từ máy chủ, dữ liệu được chuyển đổi sang định dạng JSON bằng cách sử dụng phương thức response.json(). Mỗi phần tử trong mảng data.herds được duyệt qua, và một thuộc tính date được thêm vào mỗi phần tử. Giá trị của thuộc tính này là kết quả của việc gọi một component DateConverter với tham số originalDate là element.start_date. DateConverter là một component React được thiết kế để chuyển đổi ngày thành một định dạng cụ thể.

Dữ liệu được gán vào state sử dụng setProducts, và số trang tổng cộng được gán vào state totalPages. Nếu có lỗi xảy ra trong quá trình fetch dữ liệu, lỗi sẽ được in ra console để debug.

```
const handleCreate = async () => {
  if (!validate()) {return;}
  try {
    await axios.post(`/treatments`, product);
    toast.current.show({severity: "success", summary: "Thêm hoàn thành",life: 3000, });
    reloadData();
    setProduct(emptyProduct);
  } catch (error) {
    console.log("Error :", error);
  }
};
```

Hình 19. Hàm xử lý khi tạo

Hàm handleCreate sử dụng axios.post để gửi một yêu cầu POST đến đường dẫn /treatments trên máy chủ, cùng với dữ liệu được truyền vào hàm. Nếu yêu cầu thành công,

một thông báo “Thêm hoàn thành” sẽ được hiển thị. Trong trường hợp có lỗi xảy ra, một thông báo lỗi sẽ được hiển thị trong console.

```
const handleCreate = async () => {
  if (!validate()) {return; }

  try {
    const response = await axios.patch(`/processors/${data._id}`, product);
    toast.current.show({severity: "success", summary: "Sửa hoàn thành",life: 3000, });
    reloadData();
    setProduct(response.data);
  } catch (error) {
    console.log("Error update:", error);
  }
};
```

Hình 20. Hàm xử lý khi sửa

Trước khi thực hiện yêu cầu sửa đổi, hàm kiểm tra điều kiện bằng cách gọi hàm validate(). Nếu kết quả trả về là false, có nghĩa là dữ liệu không hợp lệ và hàm sẽ dừng lại tại đây và không thực hiện bất kỳ hành động nào tiếp theo.

Nếu dữ liệu được xác thực thành công, hàm tiếp tục gửi một yêu cầu PATCH đến đường dẫn /product-patches/\${data._id} trên máy chủ, với dữ liệu sản phẩm mới được truyền vào (biến product).

Sau khi yêu cầu được gửi đi, hàm đợi phản hồi từ máy chủ bằng cách sử dụng từ khóa await. Phản hồi được gán cho biến response. Yêu cầu thành công, một thông báo “Sửa hoàn thành” sẽ được hiển thị bằng cách sử dụng thư viện toast (là thư viện PrimeReact). Tiếp theo reloadData() để tải lại dữ liệu sản phẩm, giả sử là để cập nhật danh sách sản phẩm sau khi sửa đổi.

Cuối cùng, dữ liệu sản phẩm trong trạng thái của thành phần được cập nhật với dữ liệu mới nhận được từ phản hồi (setProduct(response.data)). Trong trường hợp có lỗi xảy ra trong quá trình gửi yêu cầu PATCH, thông báo lỗi sẽ được hiển thị trong console để gỡ lỗi.

```
const handleDelete = async (product) => {
  try {
    await axios.delete(`/herds/${product._id}`, product);
    alert("xóa thành công");
  } catch (error) {
    console.log("Error:", error);
  }
};
```

Hình 21. Hàm xử lý khi xóa

Hàm `handleDelete` sử dụng `axios.delete` để gửi một yêu cầu DELETE đến đường dẫn tương ứng với sản phẩm được chọn (`/herds/${product._id}`) trên máy chủ. Nếu yêu cầu thành công, một thông báo “Xóa thành công” sẽ được hiển thị. Trong trường hợp có lỗi xảy ra, một thông báo lỗi sẽ được hiển thị trong console.

b. Xử lý hiển thị giao diện

```
<DataTable value={products} selectionMode="row" selection={selectedProducts} onSelectionChange={(e) => setSelectedProducts(e.value)}
editMode="row" dataKey="_id" tableStyle={{ minWidth: "68rem" }} header={header} >
  <Column selectionMode="multiple" exportable={true}></Column>
  <Column field="name" header="Tên đàn" sortable value={product.name} style={{ minWidth: "10rem" }}></Column>
  <Column field="member_count" header="Số lượng" sortable value={product.member_count} style={{ minWidth: "10rem" }}></Column>
  <Column field="date" sortable header="Ngày tạo" value={product.start_date} style={{ width: "20%" }}></Column>
  <Column header="Nhóm" sortable sortField="category.name" filterField="category" style={{ minWidth: "14rem" }} body={representativeBodyTemplate}>
  <Column body={actionBodyTemplate} headerStyle={{ width: "10%", minWidth: "4rem" }} bodyStyle={{ left: "0" }}></Column>
</DataTable>
<Paginator
  first={{(currentPage - 1) * currentLimit}} totalRecords={totalPages * currentLimit}
  rows={currentLimit} rowsPerPageOptions={[5, 10, 20]}
  onPageChange={onPageChange}
/>
```

Hình 22. Hiện dữ liệu lên bảng (bảng danh sách đàn)

`DataTable` được tạo để hiển thị dữ liệu từ `products`. Các hàng có thể được chọn bằng cách chỉ định `selectionMode="row"` và `selection={selectedProducts}`. Sự kiện `onSelectionChange` được gán để cập nhật `selectedProducts` khi người dùng thay đổi lựa chọn. `dataKey="_id"` xác định trường dữ liệu duy nhất của mỗi hàng là `"_id"`.

Các cột (`Columns`) được định nghĩa trong `DataTable`, mỗi cột đại diện cho một trường dữ liệu trong `products`. (`field`: tên trường; `header`: Tên cột; `value`: lưu giá trị cột, `sortable`: sắp xếp). Cột đầu tiên được sử dụng để chọn nhiều hàng, được thiết lập với `selectionMode="multiple"` và có thể được xuất ra với `exportable={true}`. Các cột khác đều chứa thông tin từ các trường của `products`, như `name`, `member_count`, `date`, và `category`.

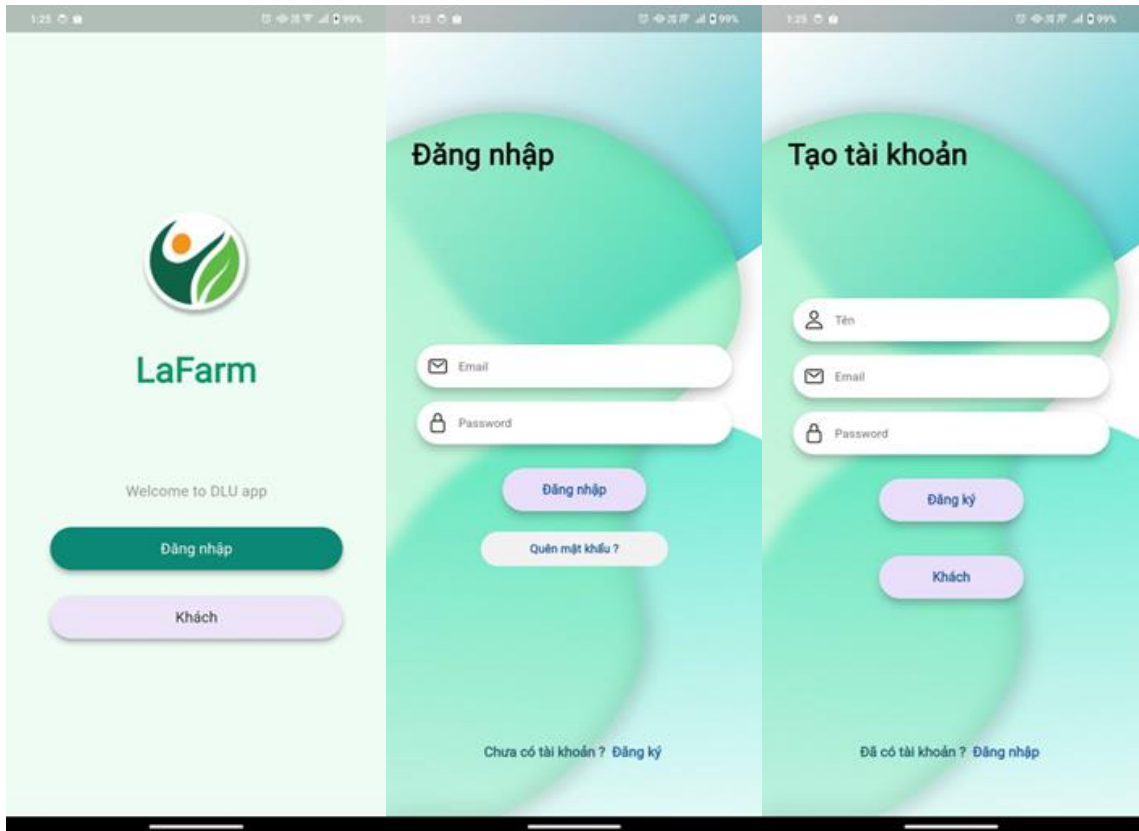
`Paginator`: phân trang kết quả hiển thị.

3.5. Kết quả

3.5.1. Ứng dụng di động

Mô tả: Chức năng chính dùng quét mã truy xuất thông tin của các sản phẩm vật nuôi. Cung cấp đầy đủ thông tin của các sản phẩm đó...

a. Giao diện đăng nhập và quét mã QR



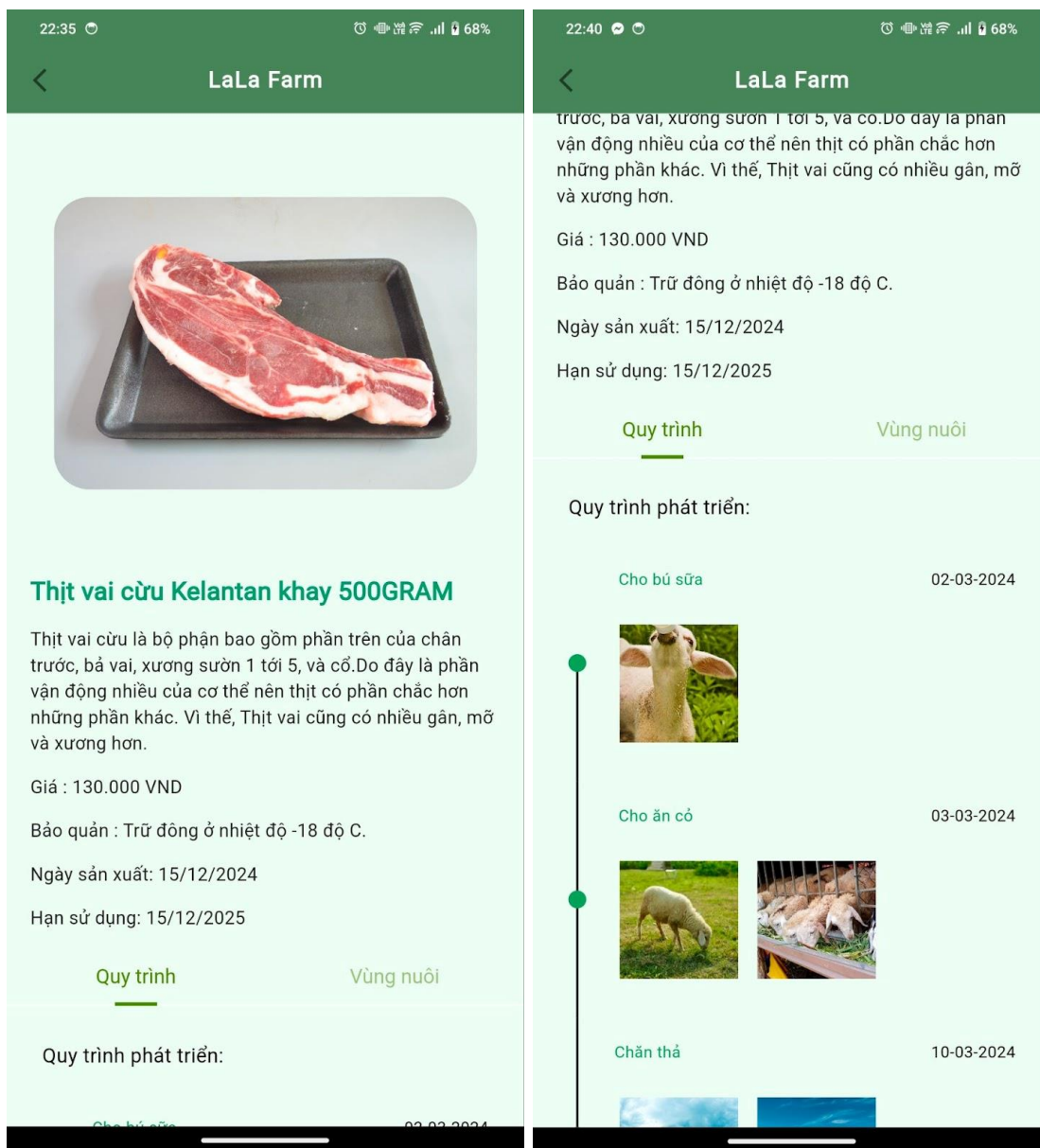
Hình 23. Đăng nhập/Đăng kí trên app di động



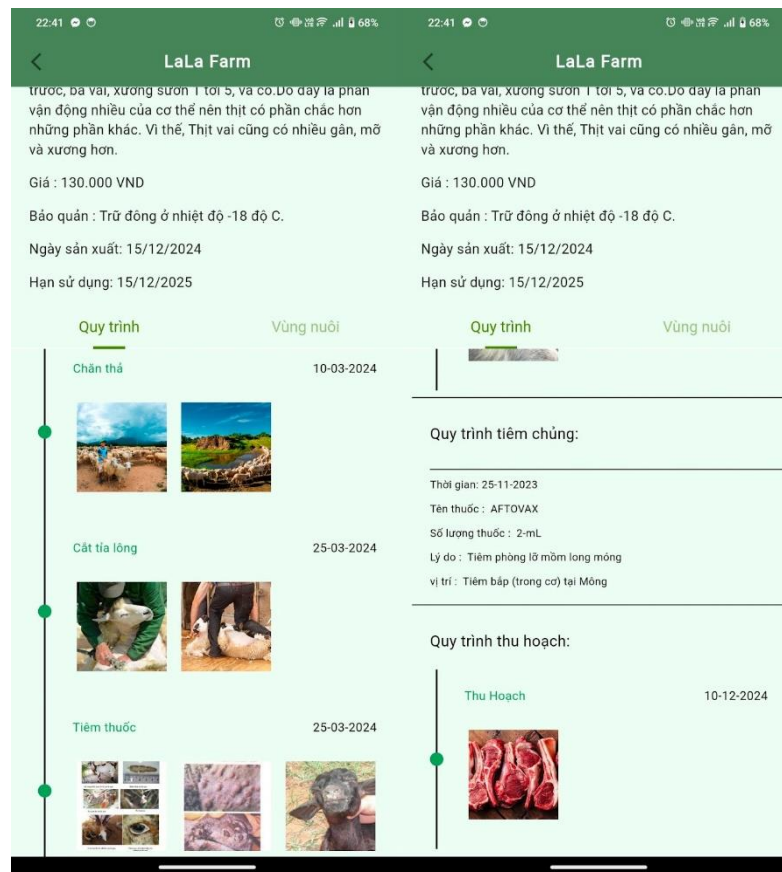
Hình 24. Giao diện quét mã QR

b. Giao diện thông tin vật nuôi, trang trại và quy trình nuôi trồng

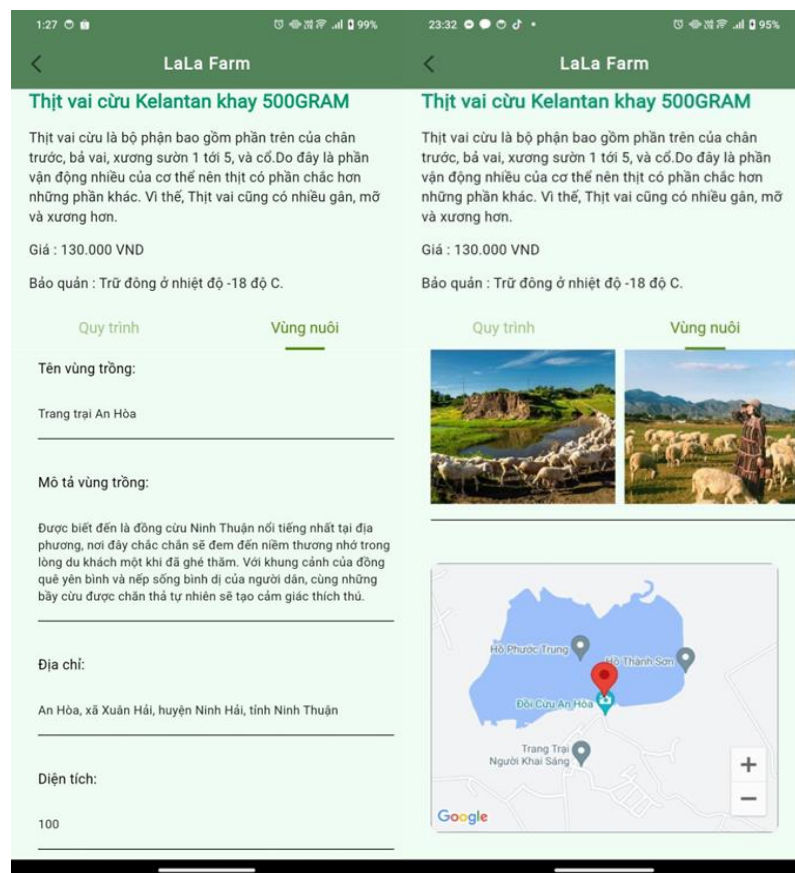
Mô tả: Gồm hình ảnh của sản phẩm , thông tin. Ngoài ra còn có thông tin vùng trồng (bản đồ, địa chỉ , diện tích), thông tin quy trình gồm có các quy trình từng ngày và hoạt động gì. Khi nhấp chuột vào từng quy trình sẽ hiện ra chi tiết hơn, có mô tả chi tiết hơn về hoạt động đó.



Hình 25. Giao diện thông tin truy xuất 1

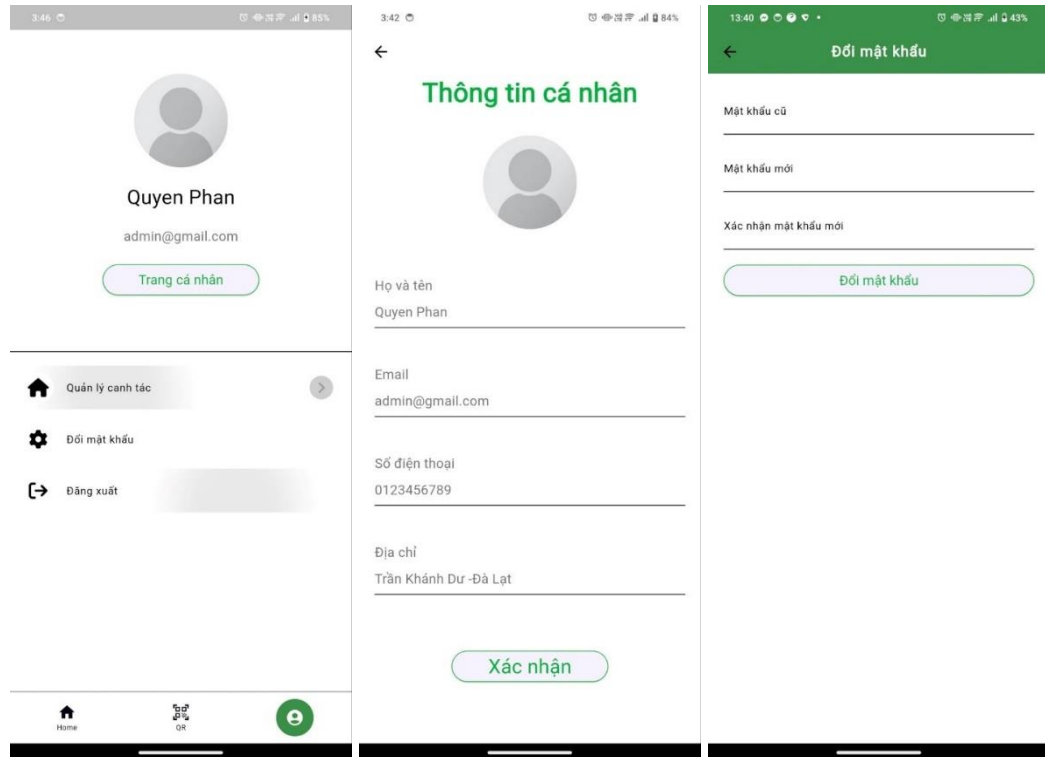


Hình 26. Giao diện thông tin truy xuất 2



Hình 27. Giao diện thông tin truy xuất 3

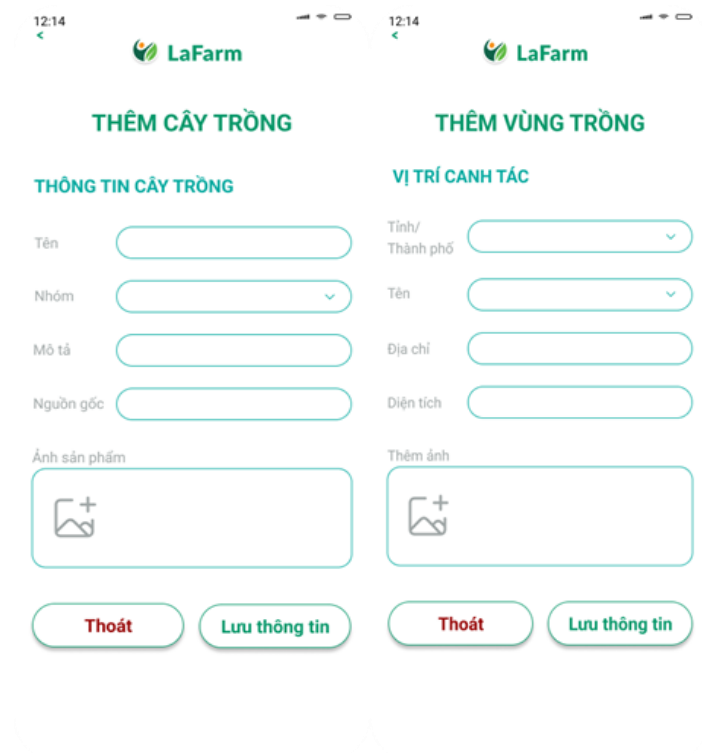
c. Giao diện cài đặt và thay đổi mật khẩu



Hình 27. Giao diện thông tin tài khoản

d. Giao diện quản lý bằng ứng dụng di động

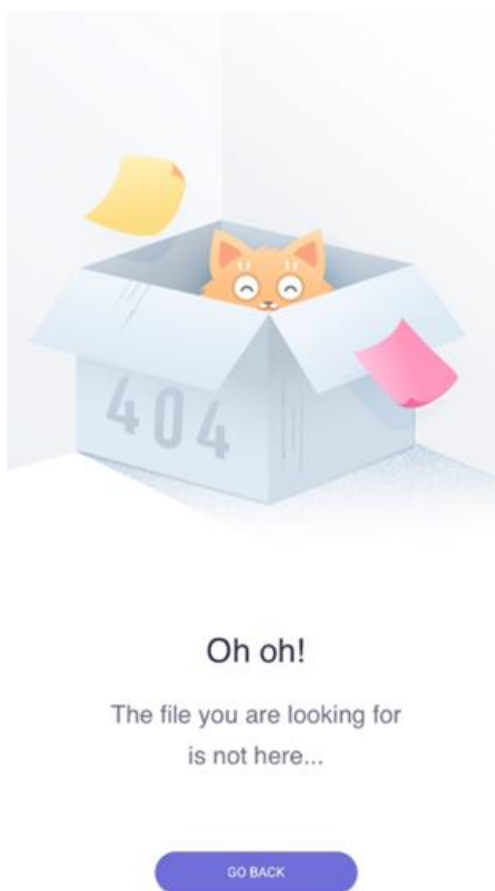
Mô tả: Có thể thực hiện chỉnh sửa thêm xóa sửa thông tin vật nuôi, trang trại, nhật ký chăm sóc, phương pháp điều trị...



Hình 28. Giao diện cây trồng/vùng trồng di động

e. Giao diện khi quét mã lỗi

Mô tả: Trang lỗi hiện ra khi mã không tồn tại trong cơ sở dữ liệu



Hình 29. Giao diện khi quét mã lỗi

3.5.2. Ứng dụng web

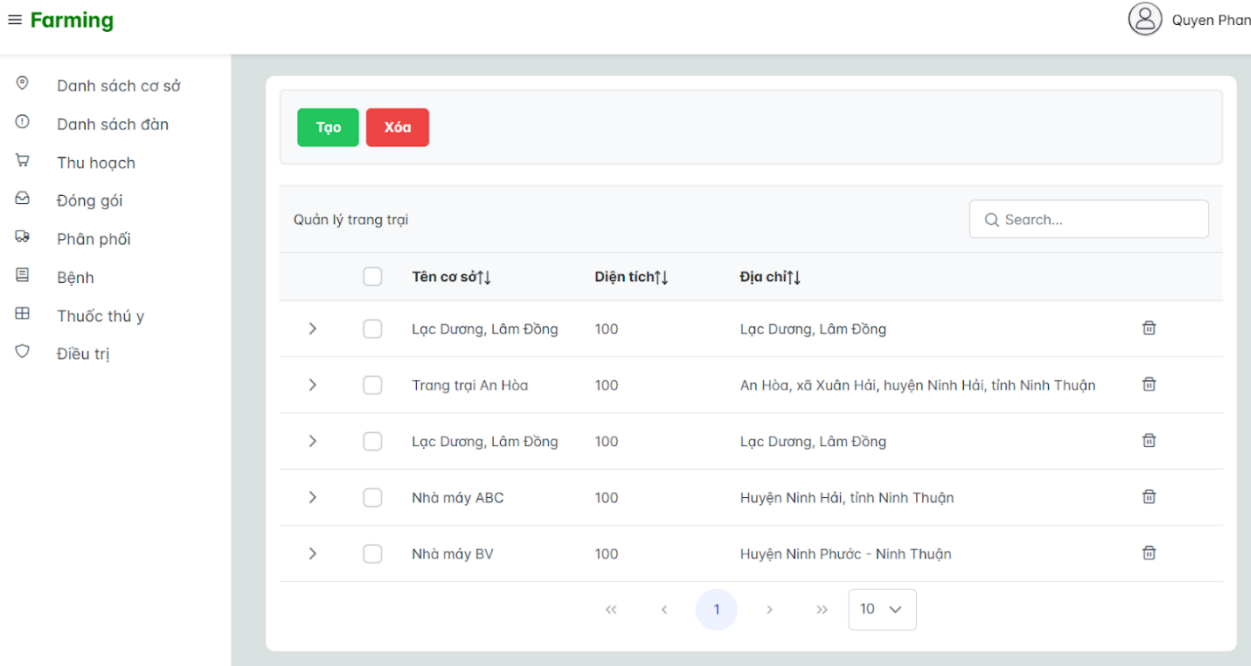
Mô tả: Các trang web được xây dựng nhằm hỗ trợ quản lý thao tác thêm xóa sửa thông tin sản phẩm dê cừu Ninh Thuận, theo dõi quá trình sinh trưởng vật nuôi trước khi thành sản phẩm bán ra thị trường. Thêm các quy trình chăm sóc từng ngày và từng hoạt động.

a. Giao diện trang danh sách cơ sở

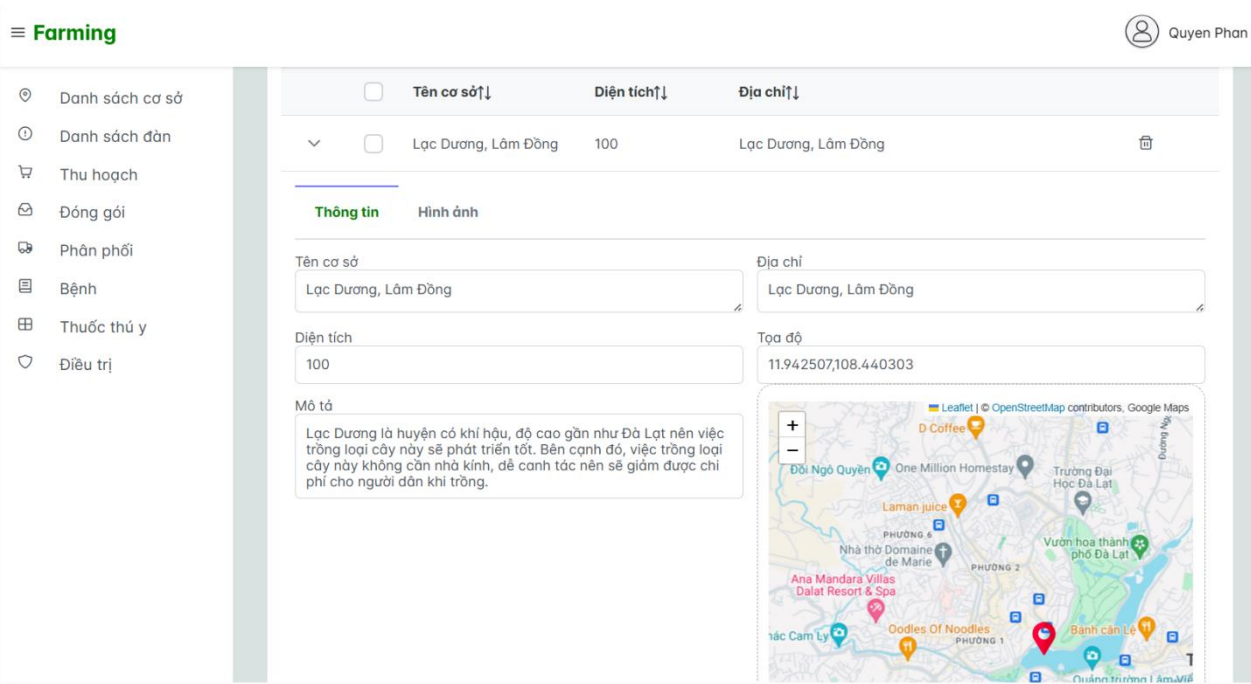
Mô tả: Trang web danh sách cơ sở gồm các trang trại, nhà máy.

Trang trại là thể hiện một khu vực nuôi dê cừu (như chuồng, nơi chăn thả,..). Cung cấp thông tin trang trại mà những lứa dê/cừu đang ở gồm có những thông tin tên cơ sở (vùng chăn nuôi), địa điểm cụ thể trên bản đồ, hình ảnh.

Nhà máy là thể hiện một khu vực xử lý đóng gói các sản phẩm dê cừu. Cung cấp thông tin nơi đóng gói gồm có những thông tin tên cơ sở (nhà máy), địa điểm cụ thể trên bản đồ, hình ảnh



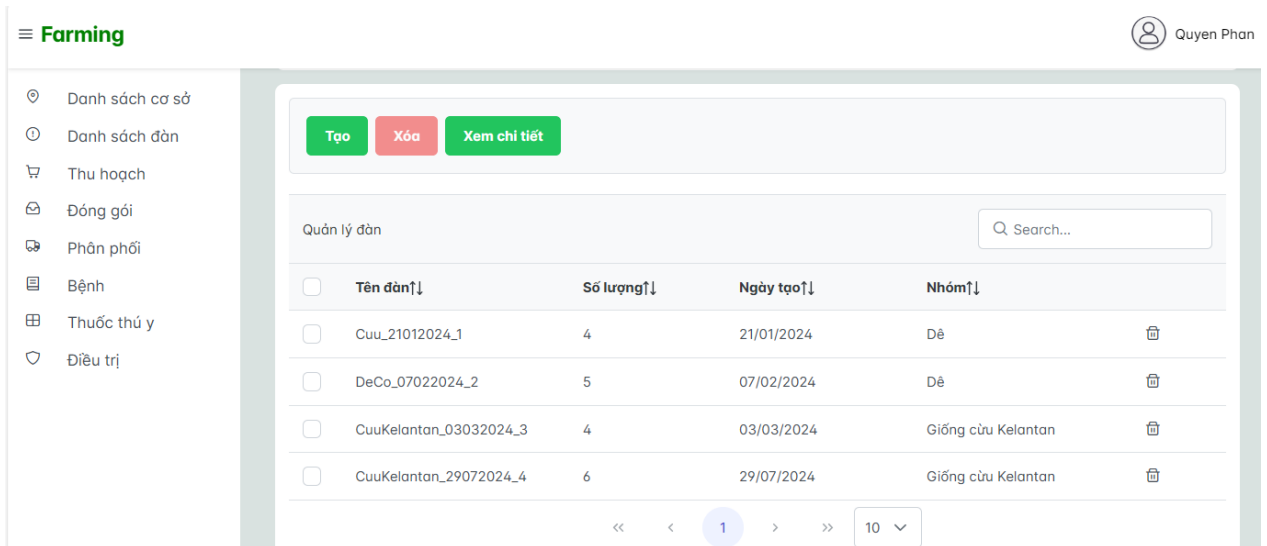
Hình 30. Giao diện danh sách cơ sở (trang trại/nhà máy)



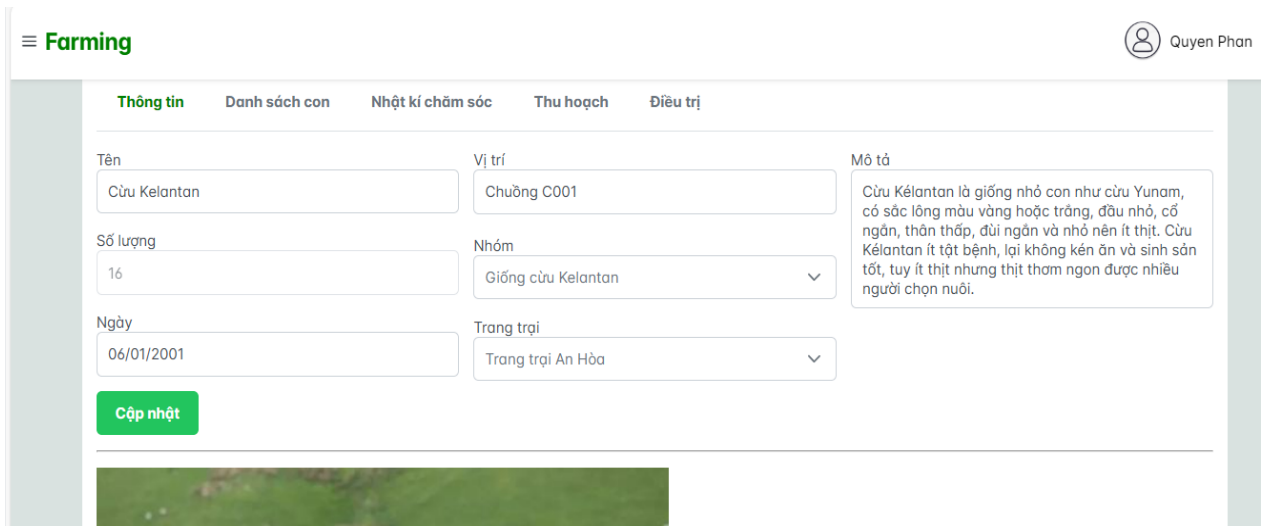
Hình 31. Giao diện chi tiết trang trại

b. Giao diện thông tin vật nuôi

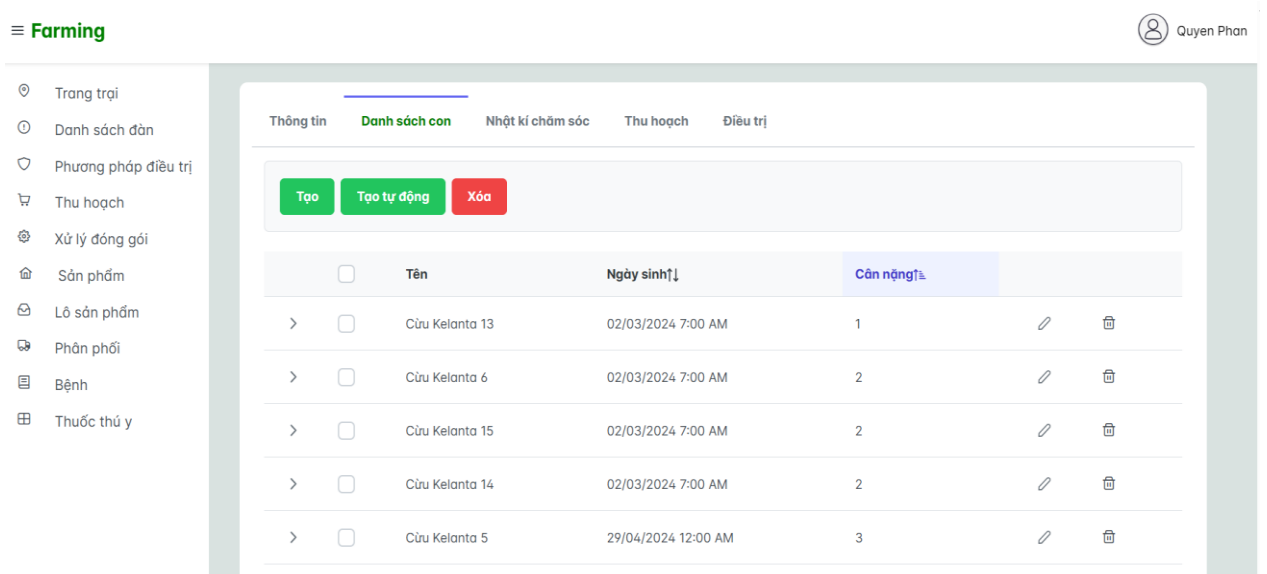
Mô tả: Quản lý thông tin vật nuôi chung của một lứa, đàn hoặc một con vật.



Hình 32. Trang web thông tin lứa đàn



Hình 33. Trang web thông tin chi tiết một lứa đàn và những con trong đàn



Hình 34. Trang web thông tin chi tiết một lứa đàn 1

- Trang trại
- Danh sách đàn
- Phương pháp điều trị
- Thu hoạch
- Xử lý đóng gói
- Sản phẩm
- Lô sản phẩm
- Phân phối
- Bệnh
- Thuốc thú y

Thông tin Danh sách con Nhật kí chăm sóc Thu hoạch Điều trị

Tạo Xóa



Tên hoạt động

Ngày



Cho bú sữa

02/03/2024 7:00 AM



Thông tin

Tên

Cho bú sữa

Mô tả

Đưa con đến cho bú

Ngày

03/02/2024

Lưu

Hình ảnh



Hình 35. Trang web thông tin chi tiết một lứa đàn 2

- Danh sách cơ sở
- Danh sách đàn
- Thu hoạch
- Đóng gói
- Phân phối
- Bệnh
- Thuốc thú y
- Điều trị

Thông tin Danh sách con Nhật kí chăm sóc Thu hoạch Điều trị

Tạo Xóa

Quản lý thu hoạch

Search...



Tên sản phẩm

Số lượng

ĐVT

Tên đàn

Ngày thu hoạch

Trạng thái



Thịt cừu

100

kg

CuuKelantan_03032024_3

10/12/2024

false



Thông tin

Hình ảnh

Đàn

CuuKelantan_03032024_3

Số lượng

100

Đơn vị

Cân

Tên sản phẩm

Thịt cừu

Trạng thái

False

Mô tả

Thịt cừu thường có màu hồng nhạt đến hồng đậm

Ngày

12/10/2024

Lưu



Sữa Cừu Kelantan A

2

Cân

CuuKelantan_03032024_3

25/04/2024

true



Hình 36. Trang web thông tin chi tiết một lứa đàn 3

Farming

Quyên Phan

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

Tạo

Xóa

Quản lý điều trị

Q Search...

	Đàn	Loại	Thuốc sử dụng	Liều lượng	Hình thức điều trị	Ngày
▼	CuuKelantan_03032024_3	Thiến (Castration)	AFTOVAX	2-mL	Tiêm bắp (trong cơ)	26/11/2023

Thông tin

Đàn

CuuKelantan_03032024_3

Mô tả

Tiêm phòng lở mồm long móng

Loại

Thiến (Castration)

Vị trí

Mông

Thuốc sử dụng

AFTOVAX

KT thuật

12345678

Liều lượng

2-mL

Ngày tiến hành

11/26/2023

Hình 37. Giao diện thông tin chi tiết một lứa đàn 4

c. Giao diện thuốc thú y, bệnh

Hỗ trợ người nông dân dùng các loại thuốc áp dụng phương pháp điều trị tương ứng với những căn bệnh của vật nuôi.

- **Giao diện thuốc thú y:** Trang web đưa ra các loại thuốc, thông tin, cách sử dụng, phạm vi sử dụng.

Farming

Quyên Phan

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

	Tên thuốc
>	Enrofloxacin 10% Oral solution
▼	Hipralona Enro-I

Thông tin

Hình ảnh

Tên

Hipralona Enro-I

Mô tả

Hipralona Enro-I Thuốc trị tiêu chảy ở dê cừu hiệu quả

Thành phần

Thuốc có thành phần chính là Enrofloxacin - một loại thuốc kháng sinh diệt khuẩn, có hoạt phổ kháng khuẩn rộng với vi khuẩn Gram âm và Gram dương. Thuốc có dạng bột, kết tinh màu trắng, ít tan trong nước, nhưng tan tốt trong dung môi hữu cơ.

Hướng dẫn sử dụng

Điều trị các bệnh về tiêu chảy ở dê cừu và gia súc. Ngoài ra thuốc còn được dùng chữa bệnh viêm mũi truyền nhiễm CRD ở gia cầm hiệu quả.

Độ độc

Enrofloxacin hoạt động bởi ức chế sự sao chép DNA, bằng cách ức chế vi khuẩn DNAgyrase và topoisomerase IV. Nó diệt khuẩn ở nồng độ rất thấp. Đó là hoạt động chống lại vi khuẩn gram âm (*Escherichia coli*, *Salmonella* sp, *Pasteurella* sp, *Klebsiella* sp,...)

Liều lượng

Enrofloxacin 50 mg / ml dung dịch tiêm cho gia súc và lợn/ cừu

Cách ly

Bà con dùng thuốc để tiêm vào bắp hoặc dưới da theo lượng chỉ dẫn của nhà sản xuất Sau khi sử dụng cần bảo quản thuốc ở nơi khô ráo, thoáng mát và tránh ánh nắng trực tiếp từ mặt trời Không được sử dụng thuốc vào mục đích khác vì các thuốc thú y hầu hết đều độc hại và không áp dụng với con người.

Khuyến nghị

xem kĩ hướng dẫn sử dụng thuốc

Hình 38. Giao diện trang quản lý thuốc

45

- **Giao diện trang bệnh:** Trang web cung cấp thông tin của các bệnh thường gặp của vật nuôi, giúp hỗ trợ nông dân quan sát các triệu chứng bệnh và từ đó đưa ra cách phòng tránh hoặc chữa trị kịp thời.

Farming Quyên Phan

Quản lý bệnh

☐ Bệnh↓

▼ ☐ Bệnh loét miệng truyền nhiễm.

Thông tin **Hình ảnh**

Tên bệnh: Bệnh loét miệng truyền nhiễm.

Mô tả: Nguyên nhân: do siêu vi trùng hoặc ăn thức ăn già, cứng gây xây sát niêm mạc.

Triệu chứng: Xung quanh môi, trong miệng đều có mụn to, loét ra; nặng thì tại mũi bầu vú cũng bị viêm loét, con vật khó nhai, khó nuốt, nước dãi thối.

Biện pháp phòng ngừa: Hằng ngày rửa vết loét bằng nước muối loãng, hay nước oxy già rồi bôi thuốc mỡ kháng sinh. Kinh nghiệm dùng chanh, khế sát vào vết loét nhiều lần cũng khỏi bệnh.

Lưu

Hình 39. Giao diện trang bệnh (thông tin)

Farming Quyên Phan

Quản lý bệnh

☐ Bệnh↓

▼ ☐ Bệnh loét miệng truyền nhiễm.

Thông tin **Hình ảnh**

No file chosen

Hình 40. Giao diện trang bệnh (hình ảnh)

d. Giao diện thu hoạch

Mô tả: Trang giao diện thu hoạch chứa tất cả các sản phẩm từ trước đến hiện tại. Mỗi sản phẩm thu hoạch cho biết tên sản phẩm, số lượng, đơn vị tính, tên đàn cung cấp sản phẩm và ngày thu hoạch. Mỗi sản phẩm được tạo ra từ nhiều đàn, vì một đàn có thể thu hoạch nhiều lần như dê có các sản phẩm sữa theo nhiều đợt, thịt dê là đợt sản phẩm thu

hoạch cuối; cừu tương tự có các sản phẩm lông cừu theo đợt, thịt cũng là đợt sản phẩm cuối.

Farming

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

Tạo

Xóa

Quản lý thu hoạch

Q Tìm kiếm...

☐	Tên sản phẩm↑↓	Số lượng↑↓	ĐVT↑↓	Tên đàn↑↓	Ngày thu hoạch↑↓	Trạng thái↑↓	
▼	☐ Sữa dê nguyên chất	7	Cân	Cuu_21012024_1	29/04/2024	✓	

Thông tin

Hình ảnh

Tên sản phẩm

Số lượng

Đơn vị

Ngày

Trạng thái

Sữa dê nguyên chất

7

Cân

04/29/2024

True

Hình 43 Giao diện trang thu hoạch

e. Giao diện đóng gói

Mô tả: Khi đã thu hoạch sản phẩm thành công, chúng ta xử lý đóng gói các sản phẩm lại. Mỗi việc xử lý đóng gói có tên, ngày đóng gói gói hàng, sản phẩm gói hàng từ đó biết số lượng, đơn vị tính của sản phẩm.

Farming

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

Quản lý đóng gói

Q Tìm kiếm...

☐	Qrcode	Sản phẩm↑↓	Số lượng↑↓	ĐVT	Ngày sản xuất	
▼		Thịt vai cừu	30	kg	29/04/2024	

Thông tin

Thu hoạch sản phẩm

Thông tin sản phẩm

Hình ảnh

Giá

Số lượng

Khối lượng tịnh

Hạn sử dụng

ĐVT

Vị trí

436000

29

1

2 năm kể từ ngày sản xuất

VND

206 đường Huỳnh Phước , huyện Ninh Phước, tỉnh Ninh Thuận

Lưu

<<

<

1

>

>>

10

Hình 41. Giao diện trang xử lý đóng gói sản phẩm

Thông tin trong đóng gói sản phẩm: Giá, khối lượng, ĐVT, số lượng, hạn sử dụng và vị trí sản xuất của gói hàng.

Khi bấm vào các thông tin xử lý đóng gói cung cấp thông tin sản phẩm (tên_loại, mô tả, cách bảo quản,...); Sản phẩm được thu hoạch: chi tiết sản phẩm được đóng gói.

Farming

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

TạoXóa

Quản lý đóng gói

Qrcode

Sản phẩm

Số lượng

ĐVT

Ngày sản xuất

Thịt vai cừu

30

kg

29/04/2024

Thông tin

Thu hoạch sản phẩm

Thông tin sản phẩm

Hình ảnh

Tên sản phẩm

Thịt vai cừu

Số lượng

30

Đơn vị

Kg

Ngày

04/29/2024

Trạng thái

True

Lưu

Hình 42. Giao diện xử lý đóng gói

Farming

Danh sách cơ sở

Danh sách đàn

Thu hoạch

Đóng gói

Phân phối

Bệnh

Thuốc thú y

Điều trị

TạoXóa

Quản lý đóng gói

Qrcode

Sản phẩm

Số lượng

ĐVT

Ngày sản xuất

Thịt vai cừu

30

kg

29/04/2024

Thông tin

Thu hoạch sản phẩm

Thông tin sản phẩm

Hình ảnh

Tên

Thịt vai cừu

Mô tả

Thịt vai cừu là bộ phận bao gồm phần trên của chân trước, bả vai, xương sườn 1 tới 5, và cổ. Do đây là phần vận động nhiều của cơ thể nên thịt có phần chắc hơn những phần khác. Ví thể, Thịt vai cũng có nhiều gân, mỡ và xương hơn.

Phương pháp bảo quản

Trữ đông ở nhiệt độ -18 độ C.

Lưu

Hình 43. Giao diện thông tin sản phẩm

f. Giao diện trang phân phối

Mô tả: Trang phân phối sản phẩm cung cấp một cái nhìn tổng quan về quá trình vận chuyển hàng hóa từ nhà kho đến các cửa hàng. Mỗi dòng trong bảng đại diện cho một lô sản phẩm cụ thể và cung cấp các thông tin quan trọng như tên và địa chỉ của nhà kho xuất phát, cửa hàng nhận hàng, cũng như ngày giao và ngày nhận lô hàng.

The screenshot displays the 'Phân phối' (Distribution) interface. It includes a sidebar with navigation links and a main content area. The main area features a table for managing distribution with columns for selection, warehouse name, address, store name, and lot number. Below the table, there are tabs for 'Thông tin' (Information) and 'Hình ảnh' (Image). The 'Thông tin' tab is active, showing fields for warehouse name, address, store name, and delivery date.

Hình 44. Giao diện trang phân phối

g. Giao diện quản lý tài khoản

Mô tả: Gồm danh sách các tài khoản người dùng tham gia quản lý trang trại. Mỗi tài khoản gồm họ, tên, email, vai trò. Người quản trị viên có thể chỉnh sửa vai trò của người dùng và xem chi tiết thông tin của họ.

The screenshot displays the 'Quản lý tài khoản' (Manage Account) interface. It includes a sidebar with navigation links and a main content area. The main area features a table for managing accounts with columns for selection, last name, first name, email, and role. Each row has edit and delete icons.

Hình 45. Trang giao diện danh sách người dùng

h. Giao diện thông tin tài khoản

Mô tả: Trang giao diện các thông tin của tài khoản, mỗi tài khoản gồm họ, tên, email, số điện thoại, địa chỉ, vai trò, hình đại diện. Có thể thực hiện chỉnh sửa các thông tin cá nhân. Người quản trị viên có thể chỉnh sửa vai trò của tài khoản.

Tài khoản



quyen
admin

Thông tin tài khoản

Tên
quyen

Email
admin@gmail.com

Đổi mật khẩu

Chỉnh sửa

Họ
quyen

Tên
Phan

Email
admin@gmail.com

Số điện thoại
0123456789

Địa chỉ
Trần Khánh Dư - Đà Lạt

Vai trò
admin

Chỉnh sửa



Hình 46. Trang web thông tin admin

CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ

4.1. Kết luận

Dự án đã hoàn thiện chức năng cơ bản về việc quét mã QR và hiển thị giao diện truy xuất nguồn gốc sản phẩm thông qua mã QR. Dự án có tiềm năng để tiếp tục phát triển và mở rộng để đáp ứng nhu cầu của ngành vật nuôi nói riêng và các mặt hàng sản phẩm khác nói chung. Nhóm chúng tôi đã học được rất nhiều trong quá trình phát triển dự án này, từ quá trình xây dựng một ứng dụng đến kỹ năng xây dựng ứng dụng, và hy vọng rằng nó có thể là một khởi đầu cho các ứng dụng truy xuất nguồn gốc phức tạp hơn trong tương lai sử dụng Flutter.

4.2. Hướng phát triển

Tối ưu hóa tốc độ truy xuất thông tin giúp người dùng dễ sử dụng đáp ứng đủ được nhu cầu. Cải thiện về mặt giao diện, tiếp tục tích hợp tính năng tìm kiếm nâng cao, cho phép người dùng lọc kết quả theo nhiều tiêu chí khác nhau. Phát triển tính năng đề xuất thông minh dựa trên lịch sử tìm kiếm và tương tác trước đó của người dùng. Tích hợp tính năng gợi ý nội dung để tăng cường trải nghiệm người quản lý dễ dàng nhanh chóng hơn và khuyến khích khám phá thêm thông tin mới. Tìm hiểu quy trình chăn nuôi đúng với thực tế. Cải thiện các biện pháp bảo mật để đảm bảo an toàn thông tin cá nhân của người dùng. Đảm bảo tính tương thích trên nhiều nền tảng khác nhau bao gồm máy tính cá nhân, điện thoại di động và máy tính bảng.

TÀI LIỆU THAM KHẢO

1. Sách

- [1] Netzel, M., & Sultanbawa, Y. (Eds.). Foods of Plant Origin. MDPI (2020).
- [2] Stephan Thesmann (auth.) - Interface Design Usability, User Experience und Accessibility im Web gestalten - Springer Vieweg (2016).
- [3] (r40) Anthony Accomazzo, Nate Murray, Ari Lerner, Clay Allsopp, David Gutman, and Tyler McGinnis - Fullstack React _ The Complete Guide to ReactJS and Friends-Newline.co (2020).
- [4] Marco L. Napoli - Beginning Flutter _ A Hands On Guide To App Development.
- [5] Ethan Brown - Web Development with Node and Express [2 ed.] _ Leveraging the JavaScript Stack.

2. Link tham khảo

- [1] Flutter, [Online]. <https://flutter.dev/>
- [2] React, [Online]. <https://react.dev/>
- [3] Express, [Online]. <https://expressjs.com/>
- [4] “Cloudinary Image & Video Management - Documentation Home | Cloudinary.” Cloudinary, cloudinary.com/documentation/
- [5] “JSON Web Token (JWT) Documentation.” JWT, self-issued.info/docs/draft-ietf-oauth-json-web-token.html
- [6] “iCheck,” Công ty Cổ phần iCheck, [Online]. Available: <https://icheck.com.vn/>