

BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI BÁO CÁO ĐỒ ÁN 1
ĐỀ TÀI: THIẾT KẾ GIAO DIỆN CHO ỨNG DỤNG
TRUY XUẤT NGUỒN GỐC CAM SÀNH

Họ và tên tác giả: NGUYỄN CÔNG HẬU

MSSV:21004130

Lớp: 232_1TH1507_KS3A_06_ngoaigio

Khóa: 46

Người hướng dẫn: Nguyễn Công Kha

Vĩnh Long, 2024

TRƯỜNG ĐẠI HỌC SPKT VĨNH LONG

KHOA CÔNG NGHỆ THÔNG TIN

PHIẾU GIAO ĐỒ ÁN CHUYÊN NGÀNH

Tên đồ án: thiết kế giao diện cho ứng dụng truy xuất nguồn gốc cam sành

Phương pháp đánh giá: ☐ Báo cáo trước hội đồng ☐ Chấm thuyết minh

Ngày giao đồ án: ngày.....tháng.....năm.....

Ngày hoàn thành đồ án: ngày.....tháng.....năm.....

Số lượng sinh viên thực hiện đồ án:

Họ tên sinh viên: NGUYỄN CÔNG HẬU MSSV: 21004130

Vĩnh Long, ngày.....tháng.....năm.....

Người hướng dẫn

NHẬN XÉT & ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN

Ý thức thực hiện:

.....

.....

.....

Nội dung thực hiện:

.....

.....

.....

Hình thức trình bày:

.....

.....

.....

Tổng hợp kết quả:

.....

.....

.....

☐ Tổ chức báo cáo trước hội đồng

☐ Tổ chức chấm thuyết minh

Vĩnh Long, ngày.....tháng.....năm.....

Người hướng dẫn

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng công nghệ vào các lĩnh vực đời sống, kinh doanh ngày càng trở nên phổ biến và cần thiết. Một trong những ứng dụng nổi bật là hệ thống truy xuất nguồn gốc, giúp người tiêu dùng và doanh nghiệp có thể kiểm tra, xác minh nguồn gốc sản phẩm một cách nhanh chóng và chính xác. Đặc biệt, đối với ngành nông sản, việc truy xuất nguồn gốc không chỉ giúp đảm bảo chất lượng sản phẩm mà còn tạo lòng tin cho người tiêu dùng.

Cam Sành, một loại trái cây đặc sản của Việt Nam, đang ngày càng được ưa chuộng bởi hương vị đặc trưng và giá trị dinh dưỡng cao. Tuy nhiên, vấn đề về chất lượng và nguồn gốc sản phẩm vẫn luôn là mối quan tâm lớn đối với người tiêu dùng. Nhằm đáp ứng nhu cầu này, việc phát triển một ứng dụng truy xuất nguồn gốc cho Cam Sành là vô cùng cần thiết.

Mục tiêu của việc thiết kế giao diện cho ứng dụng truy xuất nguồn gốc Cam Sành là mang đến một công cụ hữu ích, dễ sử dụng và thân thiện với người dùng. Ứng dụng không chỉ cung cấp thông tin chi tiết về quá trình trồng trọt, thu hoạch, và vận chuyển mà còn giúp người tiêu dùng an tâm hơn khi lựa chọn sản phẩm.

Trong quá trình thiết kế giao diện, chúng tôi chú trọng đến việc tối ưu hóa trải nghiệm người dùng, đảm bảo mọi thông tin được trình bày rõ ràng, dễ hiểu và dễ tiếp cận. Bên cạnh đó, tính năng bảo mật thông tin cũng được đề cao nhằm bảo vệ quyền lợi của cả người tiêu dùng và nhà sản xuất.

Hy vọng rằng, với ứng dụng truy xuất nguồn gốc Cam Sành này, người tiêu dùng sẽ có thêm một công cụ hữu ích để bảo vệ sức khỏe của mình và gia đình, đồng thời góp phần nâng cao giá trị thương hiệu cho nông sản Việt Nam trên thị trường quốc tế.

Em xin chân thành cảm ơn!

Mục lục

PHIẾU GIAO ĐỒ ÁN CHUYÊN NGÀNH	2
NHẬN XÉT & ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN..	Error! Bookmark not defined.
CHƯƠNG 1: TỔNG QUAN THIẾT KẾ GIAO DIỆN TRUY XUẤT NGỒN GỐC CAM SÀNH..	6
1.1 Giới thiệu về ứng dụng truy xuất nguồn gốc.....	6
1.2 Tầm quan trọng của truy xuất nguồn gốc.....	6
1.3 Lí do chọn đề tài.....	7
1.4 Tổng quan về cam sành.....	7
CHƯƠNG 2: KIẾN TRÚC ỨNG DỤNG VÀ CƠ SỞ DỮ LIỆU	9
2.1. Kiến trúc ứng dụng.....	9
2.1.1. Mô hình MVC.....	9
2.2. Thiết kế cơ sở dữ liệu.....	9
CHƯƠNG 3: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG	11
3.1. Yêu cầu về giao diện.....	11
3.2. Nguyên tắc thiết kế giao diện.....	11
3.3. Giao diện chính	11
CHƯƠNG 4: CƠ SỞ LÝ THUYẾT	13
4.1 LẬP TRÌNH ANDROID LÀ GÌ ?	13
4.2Tìm hiểu một số ngôn ngữ dung để xây dựng app di động.	13
Ngôn ngữ: Java,Kotlin,Swift.Objective-C,JavaScript (với React Native hoặc NativeScript),C#...vv.....	13
4.2.1: Giới thiệu một số ngôn ngữ lập trình java.....	13
4.2.2: Ngôn ngữ lập trình website Kotlin	14
4.2.3: Ngôn ngữ lập trình app C#	15
CHƯƠNG 5: KẾT QUẢ THIẾT KẾ GIAO DIỆN TRUY XUẤT NGỒN GỐC	16
5.1 Actor và use case tương ứng	16
5.2 Sơ đồ use case	16
5.3 Thiết kế sơ đồ lớp (CLASS DIAGRAM).....	18
5.2 Sơ đồ lớp	18
5.4 Thiết kế sơ đồ tuần tự (trình tự)	19
5.5 triển khai và kết quả	20
KẾT LUẬN.....	21
TÀI LIỆU THAM KHẢO.....	22
https://vietjack.com/android/index.jsp	22

CHƯƠNG 1: TỔNG QUAN THIẾT KẾ GIAO DIỆN TRUY XUẤT NGỒN GỐC CAM SÀNH

1.1 Giới thiệu về ứng dụng truy xuất nguồn gốc

Ứng dụng truy xuất nguồn gốc là một công cụ kỹ thuật số cho phép người dùng kiểm tra và xác minh thông tin về nguồn gốc, quá trình sản xuất, và các thông tin liên quan đến sản phẩm. Thông qua việc quét mã QR hoặc sử dụng các phương tiện khác như NFC, người dùng có thể truy cập thông tin chi tiết về sản phẩm một cách nhanh chóng và tiện lợi. Ứng dụng này không chỉ giúp người tiêu dùng có thêm niềm tin vào sản phẩm họ sử dụng mà còn giúp các nhà sản xuất và cung cấp sản phẩm minh bạch hơn trong việc quản lý chuỗi cung ứng.

Mã QR (Quick Response Code) là một dạng mã vạch hai chiều (2D) có khả năng lưu trữ thông tin một cách nhanh chóng và hiệu quả. Mã QR có thể được đọc bởi các thiết bị như smartphone, máy tính bảng hoặc máy quét mã vạch chuyên dụng. Một mã QR tiêu chuẩn có hình dạng vuông, bao gồm các mô-đun đen trên nền trắng, được sắp xếp theo một mẫu nhất định để mã hóa dữ liệu.

Đặc điểm của mã QR:

Khả năng lưu trữ cao: Mã QR có thể chứa một lượng lớn thông tin, bao gồm văn bản, URL, thông tin liên hệ, và nhiều loại dữ liệu khác.

Dễ dàng quét và đọc: Người dùng chỉ cần sử dụng camera của thiết bị di động để quét mã QR, và thông tin sẽ được hiển thị ngay lập tức.

Tính linh hoạt: Mã QR có thể được in trên nhiều bề mặt khác nhau như bao bì sản phẩm, tờ rơi, bảng quảng cáo, và các tài liệu giấy khác.

Các thành phần chính của ứng dụng truy xuất nguồn gốc bao gồm:

Hệ thống quét mã: Cho phép người dùng quét mã QR hoặc mã vạch trên sản phẩm để truy xuất thông tin.

Cơ sở dữ liệu: Chứa đựng thông tin chi tiết về sản phẩm, nhà sản xuất, quá trình vận chuyển và các thông tin liên quan khác.

Giao diện người dùng: Thiết kế thân thiện và dễ sử dụng để người dùng có thể truy cập và hiểu rõ thông tin về sản phẩm.

1.2 Tầm quan trọng của truy xuất nguồn gốc

Truy xuất nguồn gốc đóng vai trò quan trọng trong nhiều khía cạnh của chuỗi cung ứng và tiêu dùng, bao gồm:

Đảm bảo chất lượng và an toàn thực phẩm: Người tiêu dùng có thể xác minh nguồn gốc sản phẩm để đảm bảo chúng đạt tiêu chuẩn chất lượng và an toàn. Điều này đặc biệt quan trọng đối với các sản phẩm tiêu thụ trực tiếp như thực phẩm và đồ uống.

Tăng cường niềm tin của người tiêu dùng: Khi người tiêu dùng có thể truy cập thông tin chi tiết về nguồn gốc sản phẩm, họ sẽ cảm thấy yên tâm hơn khi mua hàng.

Hỗ trợ quản lý chuỗi cung ứng: Các nhà sản xuất và nhà phân phối có thể dễ dàng quản lý và giám sát chuỗi cung ứng của họ, từ đó tối ưu hóa quy trình sản xuất và vận chuyển.

Phát hiện và xử lý vấn đề nhanh chóng: Trong trường hợp phát hiện sản phẩm bị lỗi hoặc không an toàn, hệ thống truy xuất nguồn gốc giúp nhanh chóng xác định và thu hồi sản phẩm, giảm thiểu rủi ro cho người tiêu dùng.

1.3 Lí do chọn đề tài

Việc chọn đề tài thiết kế ứng dụng quét mã QR cho truy xuất nguồn gốc Cam Sành xuất phát từ những lý do sau:

Nhu cầu minh bạch về thông tin sản phẩm

Chất lượng và an toàn thực phẩm: Người tiêu dùng ngày càng quan tâm đến nguồn gốc và chất lượng của các sản phẩm họ sử dụng, đặc biệt là thực phẩm. Ứng dụng quét mã QR giúp cung cấp thông tin chi tiết về quy trình sản xuất, từ đó đảm bảo an toàn thực phẩm.

Minh bạch thông tin: Đảm bảo rằng tất cả thông tin liên quan đến sản phẩm đều minh bạch và dễ dàng truy cập, giúp người tiêu dùng tin tưởng hơn vào sản phẩm họ mua.

Ứng dụng công nghệ hiện đại

Tiện lợi và nhanh chóng: Công nghệ mã QR cho phép truy cập thông tin một cách nhanh chóng và tiện lợi, chỉ cần một lần quét mã bằng thiết bị di động.

Phổ biến và dễ sử dụng: Mã QR đã trở nên phổ biến và quen thuộc với người dùng công nghệ, không đòi hỏi kỹ năng đặc biệt để sử dụng.

Hỗ trợ phát triển nông sản Việt Nam

Nâng cao giá trị sản phẩm: Việc minh bạch thông tin sản phẩm qua mã QR giúp nâng cao giá trị và uy tín của nông sản Việt Nam, cụ thể là Cam Sành.

Chống hàng giả, hàng nhái: Ứng dụng quét mã QR giúp xác minh nguồn gốc, bảo vệ thương hiệu của nhà sản xuất khỏi các sản phẩm giả mạo, kém chất lượng.

Xu hướng tiêu dùng thông minh

Tiết kiệm thời gian: Người tiêu dùng có thể nhanh chóng kiểm tra thông tin sản phẩm ngay tại điểm mua sắm, tiết kiệm thời gian tìm hiểu và ra quyết định.

Tương tác và phản hồi: Ứng dụng cho phép người tiêu dùng đánh giá và phản hồi trực tiếp về chất lượng sản phẩm, từ đó giúp nhà sản xuất cải thiện và phát triển sản phẩm tốt hơn.

1.4 Tổng quan về cam sành

Cam sành (tên khoa học: *Citrus nobilis*) là một loại cây ăn quả phổ biến ở Việt Nam, đặc biệt là ở các tỉnh miền Tây Nam Bộ. Cam sành có vỏ màu xanh đậm, quả to, mọng nước, và vị ngọt thanh. Đây là một loại trái cây được ưa chuộng không chỉ vì hương vị thơm ngon mà còn vì giá trị dinh dưỡng cao.

Một số đặc điểm nổi bật của cam sành:

Hình dáng và kích thước: Quả cam sành có hình cầu dẹt, kích thước lớn, vỏ sần sùi và dày.

Màu sắc: Vỏ cam có màu xanh đậm khi còn non và chuyển sang màu vàng khi chín. Thịt cam có màu vàng cam, mọng nước.

Hương vị: Cam sành có vị ngọt thanh, đôi khi có chút chua nhẹ, hương thơm dịu.

Giá trị dinh dưỡng: Cam sành chứa nhiều vitamin C, vitamin A, và các khoáng chất cần thiết như canxi, kali, và chất xơ. Chúng có tác dụng tăng cường hệ miễn dịch, hỗ trợ tiêu hóa và làm đẹp da.

CHƯƠNG 2: KIẾN TRÚC ỨNG DỤNG VÀ CƠ SỞ DỮ LIỆU

2.1. Kiến trúc ứng dụng

Ứng dụng truy xuất nguồn gốc cam sành được xây dựng trên nền tảng Android và áp dụng mô hình kiến trúc MVC (Model-View-Controller) để đảm bảo tính modular, dễ bảo trì và mở rộng. Mô hình MVC chia ứng dụng thành ba phần chính: Model, View và Controller.

2.1.1. Mô hình MVC

Mô hình MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm phổ biến được sử dụng để tổ chức và cấu trúc mã nguồn của ứng dụng. Mô hình này gồm ba thành phần chính:

Model (Mô hình): Đây là thành phần chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Model bao gồm các lớp và đối tượng tương tác trực tiếp với cơ sở dữ liệu, thực hiện các thao tác như thêm, xóa, sửa và truy vấn dữ liệu. Trong ứng dụng truy xuất nguồn gốc cam sành, Model sẽ chứa các lớp đại diện cho sản phẩm, người dùng và các thao tác truy xuất nguồn gốc.

View (Giao diện): Đây là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng và nhận các tương tác từ người dùng. View bao gồm các hoạt động và giao diện người dùng trong ứng dụng Android. Trong ứng dụng này, View sẽ bao gồm các giao diện như màn hình đăng nhập, màn hình quét mã QR, màn hình hiển thị thông tin sản phẩm và lịch sử quét mã.

Controller (Điều khiển): Đây là thành phần trung gian giữa Model và View, chịu trách nhiệm xử lý các yêu cầu từ người dùng, gọi các phương thức trong Model để thực hiện các thao tác và cập nhật View tương ứng. Controller trong ứng dụng này sẽ bao gồm các lớp điều khiển logic ứng dụng như xử lý đăng nhập, xử lý quét mã và quản lý thông tin người dùng.

2.2. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu của ứng dụng được thiết kế để lưu trữ thông tin về sản phẩm cam sành và thông tin người dùng. Các bảng dữ liệu chính trong cơ sở dữ liệu bao gồm bảng sản phẩm và bảng người dùng.

2.2.1. Bảng sản phẩm

Bảng sản phẩm chứa thông tin về các sản phẩm cam sành, bao gồm các thuộc tính chính như:

product_id (Mã sản phẩm): Khóa chính, định danh duy nhất cho mỗi sản phẩm.

name (Tên sản phẩm): Tên của sản phẩm cam sành.

origin (Nguồn gốc): Thông tin về nguồn gốc xuất xứ của sản phẩm.

harvest_date (Ngày thu hoạch): Ngày thu hoạch sản phẩm.

expiry_date (Ngày hết hạn): Ngày hết hạn sử dụng của sản phẩm.

qr_code (Mã QR): Mã QR dùng để truy xuất thông tin sản phẩm.

Cấu trúc bảng sản phẩm:

product_id	name	origin	harvest_date	expiry_date	qr_code
1	Cam Sành A	Đồng Tháp	01-01-2024	01-03-2024	123456789
2	Cam Sành B	Hà Giang	05-01-2024	05-03-2024	987654321

2.2.2. Bảng người dùng

Bảng người dùng chứa thông tin về các tài khoản người dùng, bao gồm admin và user, với các thuộc tính chính như:

user_id (Mã người dùng): Khóa chính, định danh duy nhất cho mỗi người dùng.

username (Tên đăng nhập): Tên đăng nhập của người dùng.

password (Mật khẩu): Mật khẩu của người dùng, được mã hóa.

email (Email): Địa chỉ email của người dùng.

role (Vai trò): Vai trò của người dùng (admin hoặc user).

Cấu trúc bảng người dùng:

user_id	username	password	email	role
1	admin	encrypted_pass	admin@example.com	admin
2	user1	encrypted_pass	user1@example.com	user

CHƯƠNG 3: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

3.1. Yêu cầu về giao diện

Thân thiện với người dùng: Giao diện cần dễ hiểu, dễ sử dụng và có hướng dẫn rõ ràng cho người dùng.

Tương thích trên nhiều thiết bị: Ứng dụng phải hoạt động mượt mà trên nhiều kích thước màn hình và độ phân giải khác nhau.

Hiệu suất cao: Giao diện phải tải nhanh và phản hồi tốt để cung cấp trải nghiệm mượt mà cho người dùng.

Đảm bảo tính thẩm mỹ: Sử dụng màu sắc, hình ảnh, và bố cục hợp lý để tạo ra một giao diện hấp dẫn và chuyên nghiệp.

Tối ưu hóa trải nghiệm người dùng (UX): Cung cấp các tính năng giúp người dùng dễ dàng truy cập và sử dụng các chức năng của ứng dụng.

3.2. Nguyên tắc thiết kế giao diện

Nhất quán: Đảm bảo sự nhất quán về màu sắc, phông chữ, và phong cách trên toàn bộ ứng dụng.

Đơn giản và trực quan: Giảm thiểu sự phức tạp và đảm bảo rằng người dùng có thể dễ dàng tìm thấy và sử dụng các chức năng mà họ cần.

Tập trung vào người dùng: Thiết kế dựa trên nhu cầu và hành vi của người dùng, đảm bảo rằng mọi yếu tố giao diện đều có mục đích rõ ràng và hữu ích.

Phản hồi tức thì: Cung cấp phản hồi ngay lập tức cho các thao tác của người dùng, giúp họ biết được tình trạng của hệ thống.

Tiêu chuẩn hóa: Tuân thủ các nguyên tắc và hướng dẫn thiết kế giao diện người dùng của Android để đảm bảo tính tương thích và dễ sử dụng.

3.3. Giao diện chính

Giao diện chính của ứng dụng bao gồm ba phần chính: giao diện đăng nhập, giao diện quét mã, và giao diện quản lý. Mỗi phần sẽ được mô tả chi tiết dưới đây.

3.3.1. Giao diện đăng nhập



Mô tả: Giao diện đăng nhập là màn hình đầu tiên mà người dùng thấy khi mở ứng dụng. Nó bao gồm các trường để nhập tên đăng nhập và mật khẩu, cùng với các nút để đăng nhập hoặc đăng ký.

Yêu cầu thiết kế:

Trường nhập liệu: Rõ ràng, dễ nhìn, và đủ lớn để người dùng dễ dàng nhập liệu.

Nút đăng nhập: Nổi bật và dễ nhấn.

Liên kết quên mật khẩu: Cung cấp tùy chọn cho người dùng khôi phục mật khẩu.

Hình ảnh và logo: Hiển thị logo và hình ảnh đại diện của ứng dụng để tạo ấn tượng đầu tiên tốt.

3.3.2. Giao diện quét mã



Mô tả: Giao diện này cho phép người dùng quét mã QR của cam sành để truy xuất nguồn gốc. Nó sử dụng camera của thiết bị để quét mã và hiển thị thông tin sản phẩm. Yêu cầu thiết kế:

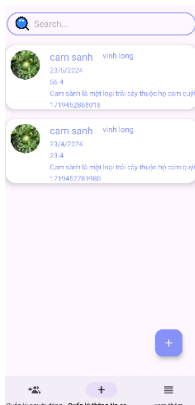
Camera preview: Khu vực hiển thị video từ camera để người dùng có thể căn chỉnh mã QR.

Nút quét: Nổi bật và dễ nhấn để bắt đầu quá trình quét.

Hướng dẫn: Cung cấp chỉ dẫn ngắn gọn cho người dùng về cách sử dụng tính năng quét mã.

Thông tin sản phẩm: Hiển thị thông tin sản phẩm sau khi quét thành công, bao gồm tên, nguồn gốc, và các chi tiết liên quan.

3.3.3. Giao diện quản lý



Mô tả: Giao diện quản lý dành cho admin để quản lý thông tin người dùng và sản phẩm. Nó bao gồm các chức năng thêm, sửa, xóa và xem thông tin.

Yêu cầu thiết kế:

Danh sách người dùng và sản phẩm: Hiển thị dưới dạng bảng hoặc danh sách, dễ dàng tìm kiếm và sắp xếp.

Form nhập liệu: Dễ sử dụng, với các trường bắt buộc được đánh dấu rõ ràng.

Nút thao tác: Rõ ràng và dễ dàng thực hiện các chức năng quản lý (thêm, sửa, xóa).

Bảo mật: Đảm bảo rằng chỉ admin có quyền truy cập và thay đổi thông tin quan trọng.

CHƯƠNG 4: CƠ SỞ LÝ THUYẾT

4.1 LẬP TRÌNH ANDROID LÀ GÌ ?

Lập trình Android là quá trình tạo ra các ứng dụng hoặc game dành cho hệ điều hành Android, được phát triển bởi Google. Ngôn ngữ chính thường được sử dụng để lập trình Android là Java và Kotlin. Các ứng dụng Android có thể được phát triển bằng việc sử dụng Android Studio, môi trường phát triển tích hợp (IDE) chính thức của Google dành cho lập trình Android.

4.2 Tìm hiểu một số ngôn ngữ dùng để xây dựng app di động.

Có các ngôn ngữ thường dùng và thông dụng trong việc xây dựng một app android như:

Ngôn ngữ: Java, Kotlin, Swift, Objective-C, JavaScript (với React Native hoặc NativeScript), C#...vv.



4.2.1: Giới thiệu một số ngôn ngữ lập trình java

JAVA: Ngôn ngữ lập trình JAVA Java là một ngôn ngữ lập trình thông dịch và hướng đối tượng, được phát triển ban đầu bởi Sun Microsystems (nay là Oracle Corporation). Được giới thiệu lần đầu vào năm 1995, Java đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới với nhiều ứng dụng và lĩnh vực khác nhau.

❖ Ưu điểm:

Hỗ trợ mạnh mẽ từ Google: Java được Google lựa chọn làm ngôn ngữ chính thức cho việc phát triển ứng dụng Android, do đó được hỗ trợ mạnh mẽ từ Google và có sự hỗ trợ chặt chẽ với Android SDK.

Cộng đồng lớn và tài liệu phong phú: Java có một cộng đồng lập trình viên rộng lớn và giàu kinh nghiệm. Điều này có nghĩa là có rất nhiều tài liệu học tập, thư viện và framework phát triển ứng dụng Android được viết bằng Java.

Ổn định và đáng tin cậy: Java là một ngôn ngữ ổn định, có một cộng đồng lâu đời và đã được sử dụng rộng rãi trong nhiều ứng dụng lớn trên toàn thế giới. Do đó, nó đã được kiểm chứng và đáng tin cậy trong các dự án phát triển lớn.

❖ Nhược điểm:

Cú pháp phức tạp hơn so với các ngôn ngữ mới hơn: So với một số ngôn ngữ lập trình mới hơn như Kotlin (được Google khuyến khích sử dụng), cú pháp của Java có thể phức tạp hơn và yêu cầu nhiều code để thực hiện cùng một chức năng. Kotlin, mặc

dù tương thích hoàn toàn với Java và có thể sử dụng cùng với Java trong dự án Android, được coi là có cú pháp đơn giản hơn và dễ dàng hơn để làm việc.



4.2.2: Ngôn ngữ lập trình website Kotlin

❖ Ưu điểm:

Tích hợp tốt với Java: Kotlin hoàn toàn tương thích với mã Java tồn tại, cho phép dễ dàng tích hợp vào các dự án Java hiện có mà không cần phải chuyển đổi toàn bộ mã nguồn.

An toàn hơn và dễ bảo trì hơn: Kotlin hỗ trợ kiểm tra kiểu tại thời điểm biên dịch (compile-time type checking), giúp tránh được một số lỗi thường gặp trong quá trình chạy ứng dụng. Nó cũng có cú pháp ngắn gọn hơn so với Java, giúp làm giảm đáng kể lượng mã nguồn.

Hỗ trợ lambda và extension functions: Kotlin hỗ trợ lambda và extension functions một cách rõ rệt, giúp code trở nên ngắn gọn và dễ hiểu hơn. Điều này cũng giúp cho việc viết các đoạn mã xử lý tập trung (functional programming) trở nên dễ dàng hơn.

Hệ sinh thái mở rộng và phong phú: Kotlin có một cộng đồng phát triển mạnh mẽ và hệ sinh thái công cụ (tooling) phát triển, với các thư viện và framework đa dạng để hỗ trợ phát triển ứng dụng.

❖ Nhược điểm:

Khả năng học:

Kotlin là một ngôn ngữ tương đối mới nên có thể cần một thời gian để học và làm quen đối với những lập trình viên đã quen với Java.

Tài liệu và hỗ trợ cộng đồng:

Mặc dù có một cộng đồng người dùng và tài liệu không ngừng mở rộng, nhưng so với Java, Kotlin vẫn có ít tài liệu và hỗ trợ thực hành hơn.

Kích thước file bytecode lớn hơn so với Java:

Kotlin có thể sinh ra bytecode lớn hơn so với Java, dẫn đến các file APK hoặc JAR lớn hơn so với khi sử dụng Java. Tuy nhiên, điều này không phải lúc nào cũng ảnh hưởng nghiêm trọng đến hiệu suất của ứng dụng.



4.2.3: Ngôn ngữ lập trình app C#

❖ **Ưu điểm:**

Phong phú và mạnh mẽ: C# là ngôn ngữ lập trình mạnh mẽ và phát triển nhanh chóng, có nhiều tính năng hiện đại như hỗ trợ lập trình hướng đối tượng (OOP), lập trình hàm, và các cấu trúc dữ liệu phức tạp.

Tích hợp tốt với .NET Framework: C# được thiết kế để hoạt động tốt trên nền tảng .NET, cho phép truy cập vào một loạt các thư viện và công cụ phát triển của Microsoft.

Bảo mật: Có các tính năng bảo mật tích hợp sẵn như kiểm tra loại (type safety), quản lý bộ nhớ tự động và xử lý ngoại lệ (exception handling), giúp giảm thiểu lỗi và tăng tính ổn định của ứng dụng.

Hỗ trợ đa nền tảng: C# không chỉ hỗ trợ Windows mà còn có thể phát triển ứng dụng trên các nền tảng khác như Linux và macOS thông qua .NET Core và .NET 5+.

❖ **Nhược điểm:**

Giới hạn vào hệ sinh thái .NET: Mặc dù C# có thể phát triển ứng dụng đa nền tảng, nhưng hệ sinh thái lớn nhất vẫn tập trung vào Windows và .NET Framework/.NET Core.

Học hàm lượng lớn: C# có một số khái niệm và cú pháp phức tạp hơn so với một số ngôn ngữ lập trình khác, điều này có thể làm cho việc học hỏi ban đầu tốn nhiều thời gian hơn.

Cộng đồng còn nhỏ hơn so với những ngôn ngữ lập trình phổ biến khác như Java hoặc Python.

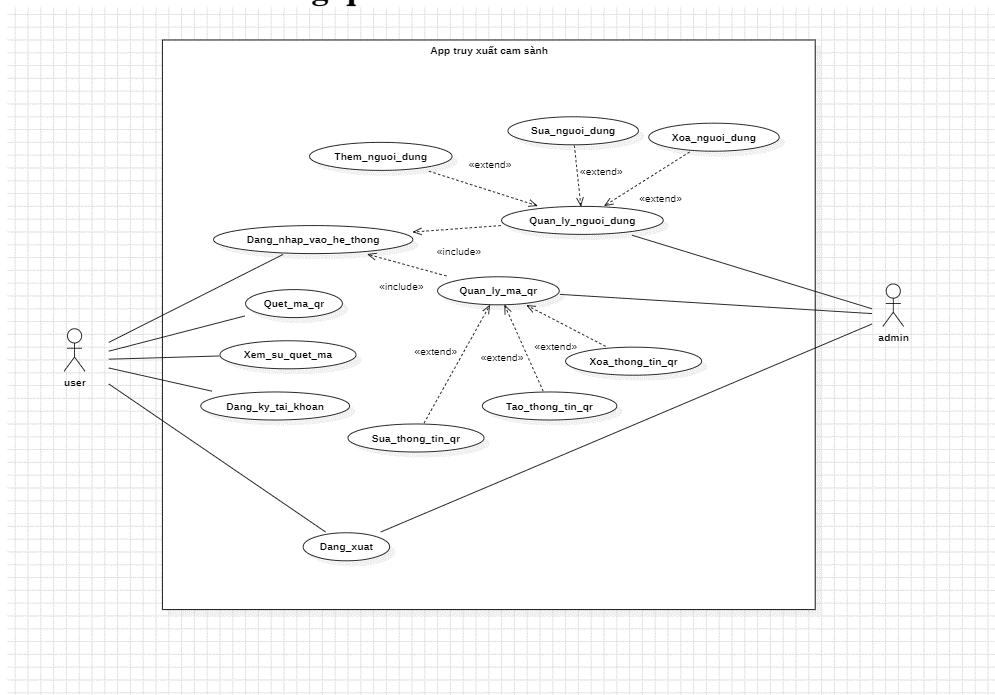
CHƯƠNG 5: KẾT QUẢ THIẾT KẾT GIAO DIỆN TRUY XUẤT NGỒN GỐC

5.1 Actor và use case tương ứng

ACTOR	USE CASE
Quản trị	<ul style="list-style-type: none"> - Tạo thông tin - Xem bài thông tin - Tạo tài khoản - Đăng nhập hệ thống
Người dùng	<ul style="list-style-type: none"> - Quét mã QR - Xem lịch sử quét mã - Tạo tài khoản

5.2 Sơ đồ use case

5.2.1 Sơ đồ use case tổng quát



5.1 Ảnh sơ đồ user case

5.2.2 Mô tả use case

Use case quét mã:

Tác nhân	user
Mục tiêu	Thực hiện quét mã và truy xuất thông tin có trên hệ thống.
Mô tả	Cho phép người dùng có thể truy xuất thông tin cam sành có sẵn trên hệ thống để biết về đầy đủ thông tin của cam sành.

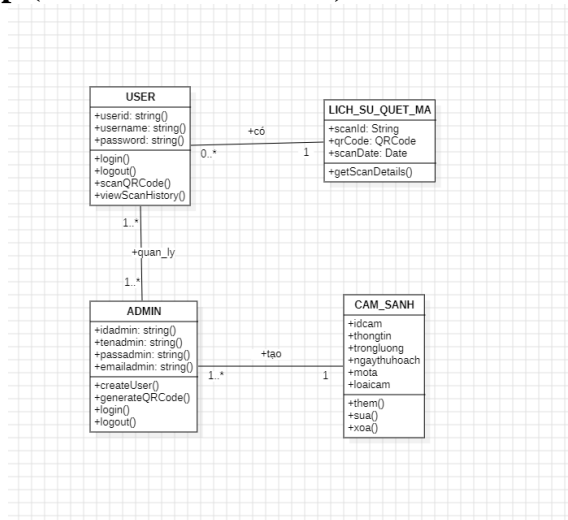
Use case đăng ký:

Tác nhân	User, admin
Mục tiêu	Người dùng có thể đăng ký tài khoản mới.
Mô tả	Người dùng mới có thể tạo tài khoản trong hệ thống bằng cách cung cấp các thông tin cần thiết như tên đăng nhập, mật khẩu và địa chỉ email.

Use case Quản lý người dùng:

Tác nhân	quản trị
Mục tiêu	Admin có thể tạo, cập nhật, và xóa người dùng trong hệ thống.
Mô tả	Use Case này mô tả các hành động mà Admin có thể thực hiện để quản lý thông tin người dùng trong hệ thống.
Điều kiện	Tác nhân phải có đăng nhập vào hệ thống với quyền nhân viên hoặc quản trị

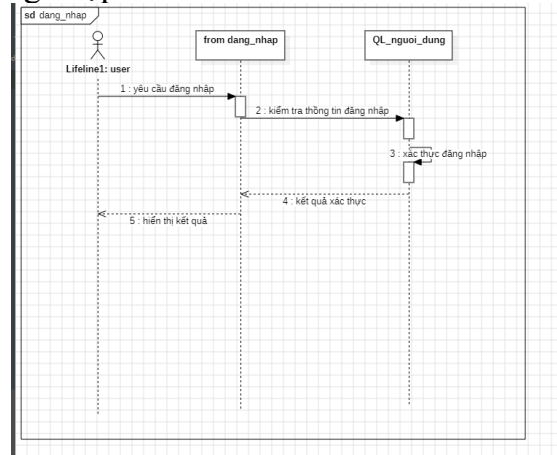
5.3 Thiết kế sơ đồ lớp (CLASS DIAGRAM)



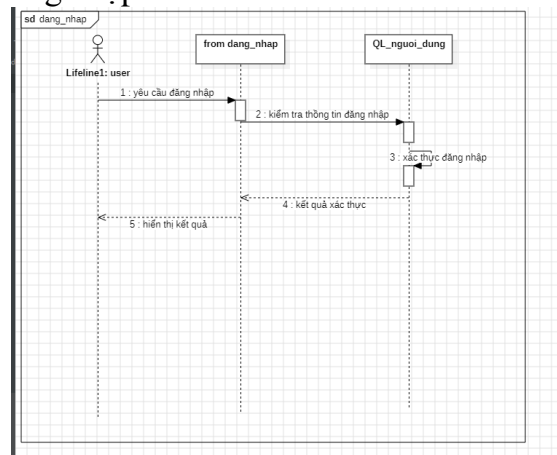
5.2 Sơ đồ lớp

5.4 Thiết kế sơ đồ tuần tự (trình tự)

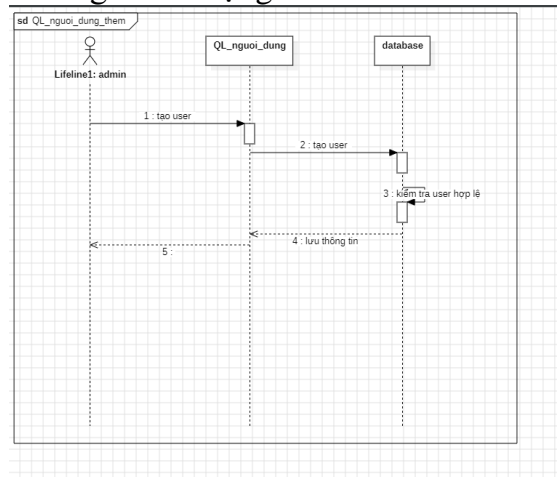
5.4.1 Sơ đồ tuần tự đăng nhập



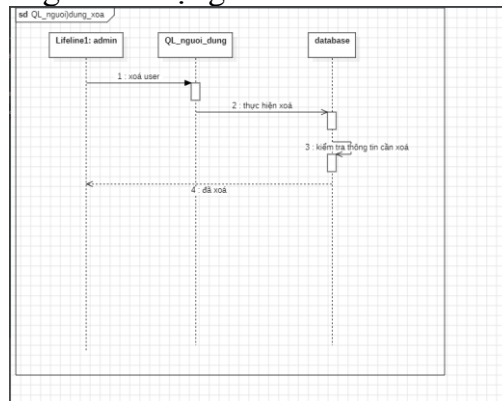
5.4.2 Sơ đồ tuần tự đăng nhập



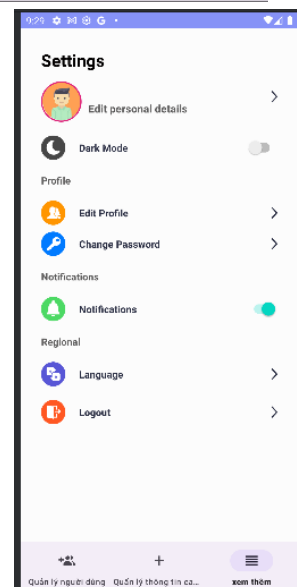
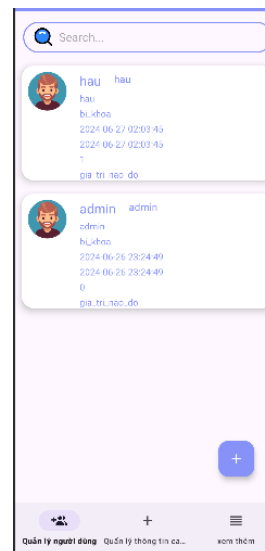
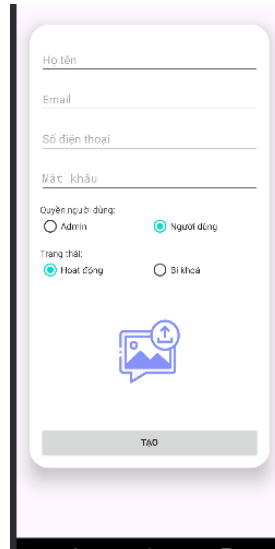
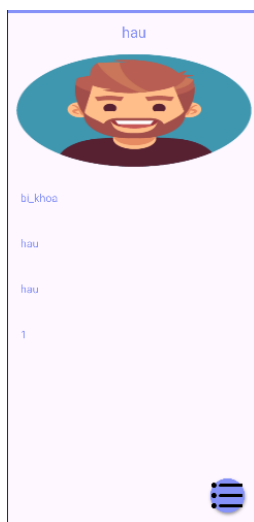
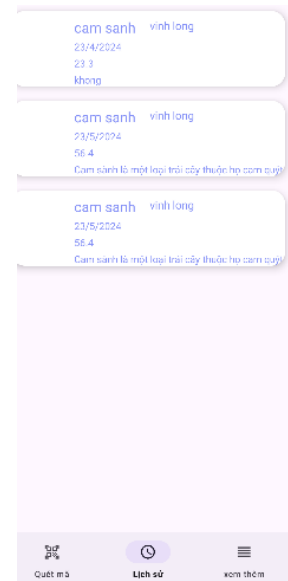
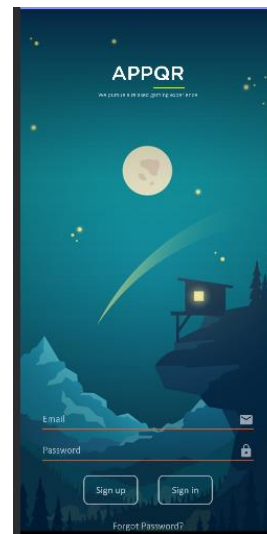
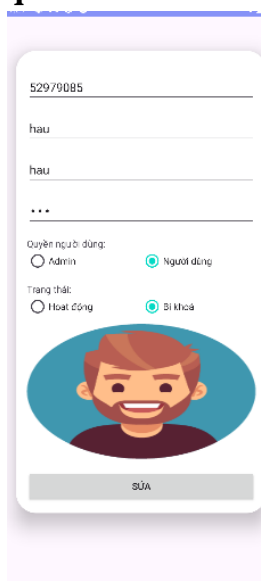
5.4.3 Sơ đồ tuần tự thêm người sử dụng



5.4.4 Sơ đồ tuần tự xoá người sử dụng



5.5 triển khai và kết quả



KẾT LUẬN

Kết luận: Qua quá trình nghiên cứu và phát triển, đề tài "Thiết Kế Giao Diện Cho Ứng Dụng Truy Xuất Nguồn Gốc Cam Sành" đã đạt được các kết quả sau:

Thiết kế giao diện người dùng: Ứng dụng đã được thiết kế với giao diện thân thiện và dễ sử dụng, bao gồm các màn hình chính như đăng nhập, quét mã, quản lý sản phẩm và xem lịch sử quét mã.

Xây dựng cơ sở dữ liệu: Một cơ sở dữ liệu đã được thiết kế và triển khai để lưu trữ thông tin sản phẩm, người dùng và lịch sử quét mã. Cơ sở dữ liệu này đảm bảo việc truy xuất và quản lý thông tin một cách hiệu quả.

Phát triển các chức năng chính: Ứng dụng đã tích hợp các chức năng chính như quét mã QR để truy xuất thông tin sản phẩm, quản lý thông tin sản phẩm bởi admin, và xem lịch sử quét mã của người dùng.

Đảm bảo bảo mật: Các biện pháp bảo mật cơ bản đã được áp dụng để bảo vệ thông tin người dùng và sản phẩm, bao gồm xác thực người dùng và mã hóa dữ liệu.

Hạn chế của đề tài

Mặc dù đề tài đã đạt được nhiều kết quả quan trọng, vẫn còn một số hạn chế cần khắc phục:

Khả năng mở rộng: Ứng dụng hiện tại chưa được tối ưu hóa cho việc mở rộng quy mô lớn, đặc biệt khi số lượng sản phẩm và người dùng tăng lên đáng kể.

Tính năng nâng cao: Một số tính năng nâng cao như thông báo đẩy, tích hợp với các hệ thống quản lý khác, và phân tích dữ liệu chưa được triển khai.

Tương thích thiết bị: Ứng dụng hiện tại mới được thử nghiệm trên một số thiết bị Android phổ biến, cần phải thử nghiệm thêm để đảm bảo tương thích với nhiều loại thiết bị và phiên bản Android khác nhau.

Hiệu suất: Hiệu suất của ứng dụng cần được cải thiện, đặc biệt là khi quét mã QR trong môi trường ánh sáng kém hoặc với các mã có chất lượng in thấp.

Định hướng phát triển

Để khắc phục các hạn chế và phát triển ứng dụng một cách toàn diện hơn, các hướng phát triển tiếp theo bao gồm:

Cải thiện khả năng mở rộng: Nâng cấp kiến trúc cơ sở dữ liệu và tối ưu hóa mã nguồn để ứng dụng có thể xử lý một lượng lớn dữ liệu và người dùng mà không ảnh hưởng đến hiệu suất.

Phát triển các tính năng nâng cao: Bổ sung các tính năng như thông báo đẩy, tích hợp với hệ thống quản lý hàng tồn kho, và phân tích dữ liệu để cung cấp giá trị gia tăng cho người dùng và nhà quản lý.

Tăng cường bảo mật: Nâng cao các biện pháp bảo mật, bao gồm sử dụng giao thức HTTPS cho truyền tải dữ liệu, xác thực hai yếu tố và mã hóa nâng cao cho dữ liệu nhạy cảm.

Tối ưu hóa hiệu suất: Cải thiện hiệu suất của ứng dụng, đặc biệt là tốc độ quét và nhận dạng mã QR, thông qua việc sử dụng các thuật toán xử lý hình ảnh và mã nguồn hiệu quả hơn.

Đảm bảo tương thích thiết bị: Thực hiện nhiều thử nghiệm trên các thiết bị và phiên bản Android khác nhau để đảm bảo tính tương thích và trải nghiệm người dùng tốt nhất trên mọi nền tảng.

TÀI LIỆU THAM KHẢO

[1] app quét mã :

<https://play.google.com/store/apps/details?id=com.gamma.scan&hl=vi>

[2] Giáo trình học phần [2]Lập trình android trường Đại học Sư Phạm Kỹ Thuật Vĩnh Long, biên soạn năm 2023:

[3] Lập trình android từ cơ bản đến nâng cao

<https://vietjack.com/android/index.jsp>