

**Thành phố Hồ Chí Minh - 2022**

**KHÓA LUẬN TỐT NGHIỆP**

*Ngành*

**KHOA HỌC MÁY TÍNH**

*Đề tài*

**Nghiên cứu React.JS, Mendix, Node.JS xây dựng ứng dụng truy xuất nguồn gốc nông sản**

*Giảng viên hướng dẫn*

**Th.S Nguyễn Đặng Thế Vinh**

*Sinh viên*

**Phạm Nguyễn Hữu Phương**

Mã sinh viên: 81011801420

****

# LỜI CAM ĐOAN

Em xin cam đoan khóa luận “Nghiên cứu React.JS, Mendix, Node.JS xây dựng ứng dụng truy xuất nguồn gốc nông sản” là một công trình nghiên cứu độc lập dưới sự hướng dẫn của giảng viên Th.S Nguyễn Đặng Thế Vinh.

Ngoài ra không có bất cứ sự sao chép nào của người khác. Nội dung khóa luận tốt nghiệp là sản phẩm mà em đã nỗ lực nghiên cứu trong quá trình học tập tại trường. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, em xin chịu hoàn toàn trách nhiệm, kỷ luật của Khoa và nhà trường nếu như có vấn đề xảy ra.

Tp. Hồ Chí Minh, ngày 7 tháng 7 năm 2022

Sinh viên

**Phạm Nguyễn Hữu Phương**

# LỜI CẢM ƠN

Để hoàn thành khóa luận tốt nghiệp “Nghiên cứu React.JS, Mendix, Node.JS xây dựng ứng dụng truy xuất nguồn gốc nông sản”, trước tiên cho phép em xin gửi lời cảm ơn tới thầy Nguyễn Đặng Thế Vinh đã giúp đỡ rất nhiệt tình trong suốt thời gian qua. Hơn nữa, khóa luận của em sẽ không thể hoàn thành tốt nếu không có sự giúp đỡ của quý thầy/cô giảng viên khoa Kỹ thuật & Khoa học máy tính - Trường Đại học Quốc tế Sài Gòn.

Thời gian thực hiện khóa luận tuy ngắn, nhưng nhờ sự hướng dẫn của thầy Nguyễn Đặng Thế Vinh đã tạo cơ hội cho em áp dụng nền tảng React.JS, Node.JS và Mendix vào công tác nghiên cứu. Trong suốt thời gian thực hiện khóa luận, em đã có cơ hội rèn luyện được các kỹ năng và nâng cao hiểu biết của mình trong việc thực hiện viết báo cáo và xây dựng chương trình, từ đó nhận thức rõ hơn về tầm quan trọng của kiến thức cũng như kỹ năng thực hiện một ứng dụng thực tiễn.

Vì vốn kiến thức và kinh nghiệm còn hạn chế nên bài báo cáo và chương trình thực nghiệm không thể tránh khỏi những hạn chế, thiếu sót. Em rất mong muốn nhận được sự góp ý của quý thầy/cô để giúp em hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn thầy Nguyễn Đặng Thế Vinh và quý thầy/cô giảng viên khoa Kỹ thuật & Khoa học máy tính - Trường Đại học Quốc tế Sài Gòn đã tạo điều kiện cho em có cơ hội được phát triển ý tưởng sáng tạo, tìm tòi, học hỏi, biết cách áp dụng những kiến thức đã học để xây dựng ứng dụng thực tế và giúp em hoàn thành khóa luận tốt nghiệp của mình.

Tp. Hồ Chí Minh, ngày 7 tháng 7 năm 2022

Sinh viên

**Phạm Nguyễn Hữu Phương**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Tp. Hồ Chí Minh, ngày 7 tháng 7 năm 2022

Giảng viên hướng dẫn

**ThS Nguyễn Đặng Thế Vinh**

# MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc107954492)

[LỜI CẢM ƠN ii](#_Toc107954493)

[MỤC LỤC iv](#_Toc107954494)

[DANH MỤC TỪ VIẾT TẮT vii](#_Toc107954495)

[DANH MỤC CÁC BẢNG viii](#_Toc107954496)

[DANH MỤC HÌNH ẢNH ix](#_Toc107954497)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc107954498)

[1.1. Thực trạng hiện nay 1](#_Toc107954499)

[1.2. Nhiệm vụ khóa luận 1](#_Toc107954500)

[1.3. Phạm vi ứng dụng 2](#_Toc107954501)

[1.4. Đối tượng sử dụng 2](#_Toc107954502)

[1.5. Mục tiêu của ứng dụng 2](#_Toc107954503)

[1.6. Các bước xây dựng khóa luận 2](#_Toc107954504)

[1.7. Phương án lưu trữ 2](#_Toc107954505)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 4](#_Toc107954506)

[2.1. Môi trường lập trình: Visual Studio Code 4](#_Toc107954507)

[2.1.1. Giới thiệu về Visual Studio Code 4](#_Toc107954508)

[2.1.2. Tính năng Visual Studio Code 4](#_Toc107954509)

[2.2. Tổng quan về React.JS 5](#_Toc107954510)

[2.2.1. Giới thiệu về React.JS 5](#_Toc107954511)

[2.2.2. React là gì? 5](#_Toc107954512)

[2.2.3. Giới thiệu về JSX 6](#_Toc107954513)

[2.2.4. Giới thiệu về Components 6](#_Toc107954514)

[2.2.5. Props và State 7](#_Toc107954515)

[2.3. Tổng quan về Tailwind CSS 7](#_Toc107954516)

[2.3.1. Tailwind CSS là gì? 7](#_Toc107954517)

[2.3.2. Tại sao chúng ta nên dùng Tailwind? 7](#_Toc107954518)

[2.4. Tổng quan về Socket 7](#_Toc107954519)

[2.4.1. Socket là gì? 7](#_Toc107954520)

[2.4.2. Tại sao cần sử dụng Socket? 8](#_Toc107954521)

[2.4.3. Phân loại Socket 8](#_Toc107954522)

[2.5. Tổng quan về redux 8](#_Toc107954523)

[2.5.1. Giới thiệu 8](#_Toc107954524)

[2.5.2. Nguyên lý hoạt động 8](#_Toc107954525)

[2.6. Tổng quan về NodeJS 9](#_Toc107954526)

[2.6.1. NodeJS là gì? 9](#_Toc107954527)

[2.6.2. Các tính năng của NodeJS 10](#_Toc107954528)

[2.7. Tổng quan về MongoDB 11](#_Toc107954529)

[2.7.1. Khái niệm 11](#_Toc107954530)

[2.7.2. Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ (RDBMS) 11](#_Toc107954531)

[2.7.3. Một số đặc điểm của MongoDB 11](#_Toc107954532)

[2.7.4. Lưu ý khi sử dụng MongoDB 12](#_Toc107954533)

[2.8. Tổng quan về Mendix 12](#_Toc107954534)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ 14](#_Toc107954535)

[3.1. Xác định yêu cầu 14](#_Toc107954536)

[3.1.1. Yêu cầu hệ thống 14](#_Toc107954537)

[3.1.2. Yêu cầu chức năng 14](#_Toc107954538)

[3.1.3. Yêu cầu phi chức năng 14](#_Toc107954539)

[3.2. Sơ đồ chức năng quản lý 15](#_Toc107954540)

[3.3. Mô hình hóa yêu cầu 15](#_Toc107954541)

[3.3.1. Đăng ký người dùng 15](#_Toc107954542)

[3.3.2. Đăng nhập 16](#_Toc107954543)

[3.3.3. Tìm kiếm nông sản trên trang chủ 18](#_Toc107954544)

[3.3.4. Tìm kiếm nông sản 19](#_Toc107954545)

[3.3.5. Thêm nông sản 20](#_Toc107954546)

[3.3.6. Tùy chỉnh nông sản 21](#_Toc107954547)

[3.3.7. Xóa nông sản 22](#_Toc107954548)

[3.3.8. Tìm kiếm quy trình 5 bước 24](#_Toc107954549)

[3.3.9. Thêm quy trình 5 bước 25](#_Toc107954550)

[3.3.10. Tùy chỉnh quy trình 5 bước 26](#_Toc107954551)

[3.3.11. Xóa quy trình 5 bước 27](#_Toc107954552)

[3.3.12. Duyệt nông sản 28](#_Toc107954553)

[3.3.13. Quản lý tài khoản 30](#_Toc107954554)

[3.3.14. Tìm kiếm tài khoản 31](#_Toc107954555)

[3.3.15. Tìm kiếm nông sản theo mã QR 32](#_Toc107954556)

[3.4. Kết quả thực nghiệm 33](#_Toc107954557)

[3.4.1. Mô hình trình tự hoạt động 33](#_Toc107954558)

[3.4.2. Giao diện Landing Page 34](#_Toc107954559)

[3.4.3. Giao diện đăng nhập 36](#_Toc107954560)

[3.4.4. Giao diện chính Dashboard 37](#_Toc107954561)

[3.4.5. Giao diện quản lý nông sản 38](#_Toc107954562)

[3.4.6. Giao diện quản lý quy trình 5 bước 41](#_Toc107954563)

[3.4.7. Giao diện duyệt nông sản 43](#_Toc107954564)

[3.4.8. Giao diện chi tiết nông sản 44](#_Toc107954565)

[3.4.9. Giao diện chat 45](#_Toc107954566)

[3.4.10. Giao diện tìm kiếm nông sản theo mã QR trên điện thoại 46](#_Toc107954567)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc107954568)

[**4.1.** **Kết luận** 49](#_Toc107954569)

[**4.2.** **Hướng phát triển** 49](#_Toc107954570)

[TÀI LIỆU THAM KHẢO 50](#_Toc107954571)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu chữ viết tắt** | **Chữ viết đầy đủ** |
| 1 | Pop-up | Cửa sổ bật lên |
| 2 | Landing Page | Trang đích |
| 3 | QR | Mã phản hồi nhanh |
| 4 | Chk | Check |
| 5 | Txt | Text |
| 6 | Btn | Button |
| 7 | User | Người dùng |
| 8 | Admin | Quản trị viên |

# DANH MỤC CÁC BẢNG

[Bảng 1. Bảng giao diện trang liên hệ 36](#_Toc107954729)

[Bảng 2. Bảng giao diện trang đăng nhập 37](#_Toc107954730)

[Bảng 3. Bảng giao diện trang giao diện chính dashboard 38](#_Toc107954731)

[Bảng 4. Bảng giao diện trang quản lý nông sản 40](#_Toc107954732)

[Bảng 5. Bảng giao diện trang quản lý quy trình 5 bước 43](#_Toc107954733)

[Bảng 6. Bảng giao diện trang duyệt nông sản 44](#_Toc107954734)

# DANH MỤC HÌNH ẢNH

[Hình 1. Vòng đời của ứng dụng (Mendix là gì? Cách Mendix-Low-Code-Platform Hoạt động, 2021) 13](#_Toc107954578)

[Hình 2. Sơ đồ chức năng quản lý 15](#_Toc107954579)

[Hình 3. Mô hình đăng ký người dùng 15](#_Toc107954580)

[Hình 4. Mô hình đăng nhập 17](#_Toc107954581)

[Hình 5. Mô hình tìm kiếm nông sản trang chủ 18](#_Toc107954582)

[Hình 6. Mô hình tìm kiếm nông sản 19](#_Toc107954583)

[Hình 7. Mô hình thêm nông sản 20](#_Toc107954584)

[Hình 8. Mô hình tùy chỉnh nông sản 21](#_Toc107954585)

[Hình 9. Mô hình xóa nông sản 23](#_Toc107954586)

[Hình 10. Mô hình tìm kiếm quy trình 5 bước 24](#_Toc107954587)

[Hình 11. Mô hình thêm quy trình 5 bước 25](#_Toc107954588)

[Hình 12. Mô hình tùy chỉnh quy trình 5 bước 26](#_Toc107954589)

[Hình 13. Mô hình xóa quy trình 5 bước 27](#_Toc107954590)

[Hình 14. Mô hình duyệt nông sản 29](#_Toc107954591)

[Hình 15. Mô hình quản lý tài khoản 30](#_Toc107954592)

[Hình 16. Mô hình tìm kiếm tài khoản 31](#_Toc107954593)

[Hình 17. Mô hình tìm kiếm nông sản theo mã QR 32](#_Toc107954594)

[Hình 18. Mô hình trình tự hoạt động 33](#_Toc107954595)

[Hình 19. Giao diện Landing Page 1 34](#_Toc107954596)

[Hình 20. Giao diện Landing Page 2 34](#_Toc107954597)

[Hình 21. Giao diện Landing Page 3 35](#_Toc107954598)

[Hình 22. Giao diện Landing Page 4 35](#_Toc107954599)

[Hình 23. Giao diện Landing Page 5 36](#_Toc107954600)

[Hình 24. Trang đăng nhập 37](#_Toc107954601)

[Hình 25. Trang giao diện chính dashboard 38](#_Toc107954602)

[Hình 26. Trang quản lý nông sản khi chấp thuận 39](#_Toc107954603)

[Hình 27. Pop-Up tạo mới sản phẩm 39](#_Toc107954604)

[Hình 28. Pop-Up tùy chỉnh sản phẩm 40](#_Toc107954605)

[Hình 29. Trang quản lý quy trình 5 bước 41](#_Toc107954606)

[Hình 30. Pop-up tạo mới quy trình 41](#_Toc107954607)

[Hình 31. Pop-up tùy chỉnh quy trình 42](#_Toc107954608)

[Hình 32. Pop-up xóa quy trình 42](#_Toc107954609)

[Hình 33. Trang duyệt nông sản 43](#_Toc107954610)

[Hình 34. Hình ảnh chi tiết review sản phầm 44](#_Toc107954611)

[Hình 35. Trang chi tiết nông sản 45](#_Toc107954612)

[Hình 36. Trang giao diện chat 45](#_Toc107954613)

[Hình 37. Giao diện chính sản phẩm 46](#_Toc107954614)

[Hình 38. Giao diện quét mã QR 47](#_Toc107954615)

[Hình 39. Giao diện chi tiết sản phẩm 48](#_Toc107954616)

# TỔNG QUAN

## Thực trạng hiện nay

* Hiện nay, trên thị trường các loại thực phẩm không rõ nguồn gốc được bày bán khá nhiều, người tiêu dùng khó có thể tránh khỏi những mối họa mà thực trạng này gây ra. Trên thực tế, có không ít người phải đối mặt với sự đe dọa từ chính những thực phẩm mà họ đang sử dụng hằng ngày. Nhiều thực phẩm có vẻ trông tươi ngon, bắt mắt, thế nhưng, sự thật về nguồn gốc, xuất xứ của chúng thì ít ai biết được. Công nghệ QR Code dù mới được ứng dụng phổ biến trong vài năm trở lại đây nhưng đã chứng minh là bước đột phá lớn thay thế cho mã vạch truyền thống. Đó là nhờ khả năng lưu giữ lượng thông tin lớn gấp hàng trăm lần với nhiều định dạng ký tự khác nhau.
* Điểm thuận lợi là người dùng có thể tự do quét QR Code từ bất cứ nơi nào, tại bất kỳ khoảng cách và vị trí nào. Với khả năng có thể phục hồi từ 30 đến 35% dữ liệu trong trường hợp bị vấy bẩn, bị hỏng, công nghệ QR Code giúp khắc phục được hoàn toàn những hạn chế của mã vạch truyền thống. Khả năng bảo mật thông tin tuyệt đối cũng là một trong những lý do giúp thanh toán qua QR Code ngày càng trở nên phổ biến. Theo đó, khi lựa chọn hình thức thanh toán này, khách hàng không cần nhập các thông tin cá nhân như số tài khoản, số thẻ… vốn được coi là những thông tin rất dễ bị lấy cắp.
* Việc sử dụng công nghệ truy xuất nguồn gốc sản phẩm, công nghệ QR Code sẽ dễ dàng xử lý các lô hàng kém chất lượng hoặc hàng giả và giúp cho việc kiểm soát hàng hóa đưa vào thị trường diễn ra minh bạch, rõ ràng, qua đó hạn chế lượng hàng kém chất lượng, hàng giả lưu thông trên thị trường.

## Nhiệm vụ khóa luận

* Từ những vấn đề trên, em quyết định xây dựng một website phục vụ nhu cầu truy xuất nguồn gốc của nông sản để người dùng có thể sử dụng để tạo mã QR và hiển thị nguồn gốc của sản phẩm đó.
* Ngoài ra, em sử dụng Mendix để xây dựng thêm 1 ứng dụng nhỏ trên nền tảng mobile để quét mã QR và hiển thị sản phẩm dựa trên mã QR đó.

## Phạm vi ứng dụng

* Đơn vị sử dụng: Những doanh nghiệp thuộc lĩnh vực nông sản.
* Tên dự án: Nghiên cứu React.JS, Mendix, Node.JS xây dựng ứng dụng truy xuất nguồn gốc nông sản.

## Đối tượng sử dụng

* Các giám đốc công ty nông sản.
* Các nhân viên quản lý nhà xưởng nông sản.
* Người làm nông nghiệp nhỏ, hộ gia đình.

## Mục tiêu của ứng dụng

* Không phải quản lý thủ công (giấy, tờ, sổ sách).
* Giúp quản lý nông sản (những loại sắp hoặc đã quá hạn sử dụng).
* Bảo mật thông tin của khách hàng.
* Tiết kiệm được thời gian trong quản lý thông tin, thao tác nhanh chóng với mã QR.
* Truy xuất được nguồn gốc nông sản theo thời gian thực thông qua quy trình 5 bước.
* Giúp người làm nông nghiệp nhỏ, lẻ tiếp cận và ứng dụng công nghệ để quản lý, kinh doanh sản phẩm và có thể đưa sản phẩm của mình đến với nhiều đối tượng khách hàng hơn.

## Các bước xây dựng khóa luận

* Lập kế hoạch phát triển hệ thống
* Phân tích hệ thống.
* Thiết kế
* Cài đặt
* Kiểm tra
* Biên soạn tài liệu và hướng dẫn

## Phương án lưu trữ

Sử dụng hệ quản trị cơ sở dữ liệu MongoDB với những lí do sau:

* Các truy vấn đa đạng hơn.
* Cập nhật nhanh hơn.
* Sự hỗ trợ chuyên nghiệp bởi MongoDB.
* Dữ liệu được lưu trong các tài liệu kiểu JSON.
* Sử dụng tốt với React.JS, được hỗ trợ với những packages đa dạng thuận tiện cho việc truy xuất dữ liệu.

# CƠ SỞ LÝ THUYẾT

## Môi trường lập trình: Visual Studio Code

### Giới thiệu về Visual Studio Code

* Visual studio Code là một trong những trình soạn thảo mã nguồn mở phổ biến nhất hiện nay được sử dụng bởi các lập trình viên. Với tốc độ nhanh và hỗ trợ được cho các nền tảng như Windows, Linux và macOS. Với những lý do trên khiến Visual Studio Code được sử dụng rộng rãi cho đến ngày hôm nay.
* Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, hỗ trợ syntax highlight và hoàn thành mã thông minh, tích hợp AI vào việc ghi nhớ hành vi của người dùng. Nó cũng cho phép tùy chỉnh giao diện, phím tắt một cách tiện lợi nhất để lập trình viên có thể sử dụng và phát triển các ứng dụng. Visual Studio Code là hoàn toàn miễn phí và nó là phần mềm mã nguồn mở theo giấy phép MIT, mặc dù bản phát hành của Microsoft là theo giấy phép phần mềm miễn phí (Wikipedia, “Visual Studio Code”, 2017).
* Visual Studio Code cung cấp kho tiện ích mở rộng cho phép người dùng tải các tiện ích tương tự như các tiện ích trên Chrome. Ngoài ra người dùng có thể tự tạo tiện ích và đăng nó lên để người khác cùng tải. Điều này sẽ không làm giảm hiệu năng của phần mềm vì nó hoạt động như một chương trình độc lập.
* Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là “Monaco”).
* Trong cuộc khảo sát vào năm 2018 trên Stack Overflow, Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, với 34.9% của 75398 người trả lời tuyên bố sử dụng nó (Wikipedia, “Visual Studio Code”, 2017).

### Tính năng Visual Studio Code

* Visual Studio Code hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json. Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.
* Visual Studio Code có thể được mở rộng qua plugin. Điều này giúp bổ sung thêm chức năng cho trình biên tập và hỗ trợ thêm ngôn ngữ. Một tính năng đáng chú ý là khả năng tạo phần mở rộng để phân tích mã, như là các linter và công cụ phân tích, sử dụng Language Server Protocol.
* Quản lý phiên bản (version control) là một tính năng tích hợp sẵn nổi bật của Visual Studio Code. Chức năng này có một tab chuyên dụng riêng bên trong thanh menu, cho phép người dùng có thể truy cập cài đặt kiểm soát phiên bản và xem các thay đổi được thực hiện đối với dự án hiện tại. Để sử dụng tính năng này, người sử dụng phải liên kết Visual Studio Code với bất kỳ hệ thống quản lý phiên bản được hỗ trợ nào (Git, SVN, Apache Subversion, Perforce, v.v.). Điều này cho phép bạn tạo kho lưu trữ cũng như thực hiện các yêu cầu đẩy (push) và kéo (pull) trực tiếp từ chương trình Visual Studio Code.

## Tổng quan về React.JS

### Giới thiệu về React.JS

* React.JS là một thư viện Javascript theo xu hướng Single Page Application. Trong khi những framework khác như Angular.JS, Vue.JS đang cố gắng để hướng đến mô hình Model View Controller thì React.JS lại đi theo khuynh hướng phối hợp với những thư viện Javascript khác. React là một thư viện cho phép nhúng code html trong code javascript nhờ vào JSX, do đó có thể dễ dàng lồng các đoạn HTML vào trong Javascript. Tích hợp giữa HTML và Javascript vào trong JSX làm cho các đoạn component (là những thành phần giao diện được định nghĩa độc lập, có thể tái sử dụng và hoàn toàn tách biệt nhau. Chúng ta có thể hiểu component là một hàm trong Javascript) dễ hiểu hơn.

### React là gì?

* React là một thư viện giao diện được phát triển bởi Meta để hỗ trợ trong việc xây dựng các components mà người dùng có thể tương tác, có các đầu vào và có thể sử dụng lại được. Hai nền tảng mạng xã hội nổi tiếng được chính Meta phát triển là Facebook và Instagram cũng được phát triển trên React.
* Điểm hấp dẫn của React là thư viện không chỉ hoạt động trên phía client, nó còn được render trên server và cả 2 có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của các lần render với giá trị của lần render trước, sau đó cập nhật ít thay đổi nhất trên DOM. Sau đây là một số khái niệm thường được nhắc đến khi nói tới React:

### Giới thiệu về JSX

* JSX là một phần mở rộng của React đối với cú pháp sử dụng ngôn ngữ Javascript, cung cấp phương thức để cấu trúc kết xuất thành phần bằng cách sử dụng cú pháp quen thuộc với nhiều nhà phát triển. Nó giúp viết các mã HTML trong Javascript. Nó có đặc điểm là nhanh hơn khi nó đã tối ưu hóa trong khi biên dịch các đoạn mã sang Javascript với thời gian thực hiện nhanh hơn nhiều so với một đoạn mã tương tự được viết bằng Javascript thuần. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch (Đỗ Khánh Toàn, 2017).
* Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

### Giới thiệu về Components

* React được xây dựng xung quanh các component, chứ không dùng template như các framework khác (Đỗ Khánh Toàn, 2017).
* Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác (Đỗ Khánh Toàn, 2017).
* Các component này có thể tái sử dụng, hoặc kết hợp với các Component khác để tạo ra một giao diện hoàn chỉnh (Đỗ Khánh Toàn, 2017).
* Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái (Đỗ Khánh Toàn, 2017).
* Mọi thứ React tạo ra đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo (Đỗ Khánh Toàn, 2017).

### Props và State

* Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến (Đỗ Khánh Toàn, 2017).
* State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI (Đỗ Khánh Toàn, 2017).

## Tổng quan về Tailwind CSS

### Tailwind CSS là gì?

Tailwind CSS là một utility-first framework khá tương đồng với bootstrap, nó có các class built-in mà chúng ta có thể dùng. Tuy nhiên, Tailwind CSS có nhiều các class bao gồm các thuộc tính khác nhau và chúng rất quan trọng trong việc xây dựng giao diện người dùng, nhờ đó chúng ta có thể dễ dàng mở rộng và tạo ra những class sử dụng chính những class được built-in đó, khiến công việc tùy chỉnh giao diện dễ dàng hơn rất nhiều trong việc đọc hiểu code cũng như làm gọn lại khối lượng code phải viết.

### Tại sao chúng ta nên dùng Tailwind?

Dù mang nhiều đặc tính tương đồng bootstrap nhưng điều tiện lợi của Tailwind CSS so với bootstrap là chúng ta có nhiều class hơn và sử dụng được 1 cách linh hoạt hơn nếu so sánh với Bootstrap. Chúng ta cũng phải nói đến khả năng tùy biến và mở rộng cao, đem đến cho ta sự linh hoạt không giới hạn.

## Tổng quan về Socket

### Socket là gì?

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket (Thân Hoàng, 2020).

### Tại sao cần sử dụng Socket?

* Chức năng chính của Socket là để “duy trì kết nối” giữa client với server để ứng dụng có thể hoạt động realtime. Chẳng hạn như một ứng dụng nhắn tin, Socket sẽ được sử dụng để “lắng nghe” mỗi khi có tin nhắn mới, reaction... để cập nhật lên giao diện. Việc cập nhật dữ liệu thời gian thực như thế đem lại một trải nghiệm tuyệt vời cho người dùng (Thân Hoàng, 2020).
* Nếu không sử dụng socket, ta vẫn có một cách khác để cập nhật dữ liệu đó là liên tục gửi/nhận các request/response trong một khoảng thời gian nhất định, ví dụ cứ sau mỗi 2s ta lại cho ứng dụng check api xem dữ liệu có gì thay đổi không, nếu có thì cập nhật vào ứng dụng. Tuy nhiên các xử lí này khiến hệ thống làm việc liên tục (Thân Hoàng, 2020).

### Phân loại Socket

* Stream Socket: Dựa trên giao thức TCP (Tranmission Control Protocol), việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Do đó, hình thức này được gọi là socket hướng kết nối (Thân Hoàng, 2020).
* Datagram Socket: Dựa trên giao thức UDP (User Datagram Protocol) việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 quá trình. Do đó, hình thức này được gọi là socket không hướng kết nối (Thân Hoàng, 2020).

## Tổng quan về redux

### Giới thiệu

* Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Được dựa trên nền tảng tư tưởng của kiến trúc Flux do Facebook giới thiệu, do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React (React Js và React Native) (Đỗ Khánh Toàn, 2017).

### Nguyên lý hoạt động

* Tất cả mọi thứ thay đổi trong ứng dụng, bao gồm dữ liệu và trạng thái giao diện sẽ được lưu trong một đối tượng gọi là state. Một state sẽ phụ thuộc vào khá nhiều yếu tố như dữ liệu từ máy chủ ban đầu, thao tác người dùng, dữ liệu từ máy chủ. Những dữ liệu này đến từ nhiều nguồn khác nhau dẫn đến việc chúng ta khó kiểm soát luồng dữ liệu và sẽ làm hao tổn bộ nhớ máy tính nếu chúng ta cứ gọi lại 1 nguồn dữ liệu dù chúng ta đã gọi trước đó. Chính vì lẽ đó, Redux ra đời để giải quyết vấn đề này, các nguồn dữ liệu sau khi được phân loại và quản lý bởi Redux sẽ tạo thành 1 nguồn duy nhất và đáng tin cậy.
* State chỉ được phép đọc, cách duy nhất để thay đổi State của ứng dụng là phát một Action (là 1 object mô tả những gì xảy ra). Trạng thái của ứng dụng không được phép thay đổi “trực tiếp”, trạng thái cũng chỉ là một đối tượng mà thôi, nên việc thay đổi là được. Tuy nhiên, với Redux hay Flux thì trạng thái chỉ thay đổi khi và chỉ khi có một sự kiện xảy ra, giống như ra trận thì chỉ được phép nghe lời từ chỉ huy, mọi tin tức tình báo đều được gởi tới chỉ huy, nếu không có lệnh từ chỉ huy thì tất cả không được phép manh động (Đỗ Khánh Toàn, 2017).
* Sử dụng pure function nhận tham số là state trước đó và action và trả về state tiếp theo. Hàm này gọi là reducer (pure function là hàm trả về giá trị phụ thuộc duy nhất vào giá trị của tham số, pure function có tập các tham số thì giá trị trả về cũng là một tập tương tự như thế). Việc thay đổi trạng thái của ứng dụng, được thực hiện thông qua các hàm thuần tuý. Đưa vào giá trị sự kiện, trạng thái hiện tại và hàm trả về trạng thái tiếp theo. Dù tương lai ứng dụng của bạn có thể rất lớn, nhưng các hàm reducer này thì chỉ cần nhỏ gọn thay đổi trên từng lá của cây trạng thái, và chúng hoàn toàn có thể kết hợp với nhau tạo thành chuỗi sự kiện. Ví dụ: người click vào menu (một sự kiện → thay đổi trạng thái), sau đó router cũng cần thay đổi để phù hợp với ngữ cảnh (Đỗ Khánh Toàn, 2017).

## Tổng quan về NodeJS

### NodeJS là gì?

* NodeJS là một mã nguồn mở, đa nền tảng, chạy trên môi trường JavaSript, 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 (Đoàn Văn Phú, 2020).
* Ở chế độ đồng bộ khi thực thi từng dòng sẽ tiến hành thực thi dòng kế tiếp sau khi dòng trước đã thực thi xong, còn khi sử dụng ở chế độ bất đồng bộ thì thực thi tất cả các dòng code cùng 1 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 (Đoàn Văn Phú, 2020).
* 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 (Đoàn Văn Phú, 2020).
* 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 (Đoàn Văn Phú, 2020).

### Các tính năng của NodeJS

* Lập trình hướng sự kiện và không đồng bộ: Toàn bộ API trong thư viện NodeJS đều không đồng bộ, hay không bị chặn. Về cơ bản điều này có nghĩa là một server sử dụng NodeJS sẽ không bao giờ chờ một API trả về dữ liệu. Server sẽ chuyển sang API kế tiếp sau khi gọi API đó và cơ chế thông báo của Events trong NodeJS giúp server nhận được phản hồi từ lần gọi API trước (Đoàn Văn Phú, 2020).
* Cực kỳ nhanh chóng: Được xây dựng trên Công cụ JavaScript V8 của Google Chrome, thư viện NodeJS có khả năng xử lý mã vô cùng nhanh.
* Đơn luồng/Single thread nhưng có khả năng mở rộng cao: NodeJS sử dụng một mô hình luồng đơn với vòng lặp sự kiện/event. Cơ chế event cho phép máy chủ phản hồi non-blocking và cũng cho phép khả năng mở rộng cao hơn so với các server truyền thống hỗ trợ giới hạn các thread để xử lý yêu cầu. NodeJS sử dụng một chương trình đơn luồng, cùng một chương trình có thể cung cấp dịch vụ cho một số lượng yêu cầu lớn hơn so với các máy chủ truyền thống như Apache HTTP Server (Đoàn Văn Phú, 2020).
* Không có buffer - Các ứng dụng NodeJS không có vùng nhớ tạm thời (buffer) cho bất kỳ dữ liệu nào. Các ứng dụng này chỉ đơn giản xuất dữ liệu theo khối.
* License - NodeJS được phát hành theo giấy phép MIT.

## Tổng quan về MongoDB

### Khái niệm

* MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data). Nó lưu trữ dưới dạng dữ liệu văn bản, là một cấu trúc dữ liệu bao gồm các cặp giá trị và trường tương tự như các đối tượng JSON.

### Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ (RDBMS)

* Sử dụng ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
* Cấu trúc của một đối tượng là rõ ràng.
* Không có các Join phức tạp.
* Khả năng truy vấn sâu hơn. MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.
* MongoDB dễ dàng để mở rộng.
* Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
* Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

### Một số đặc điểm của MongoDB

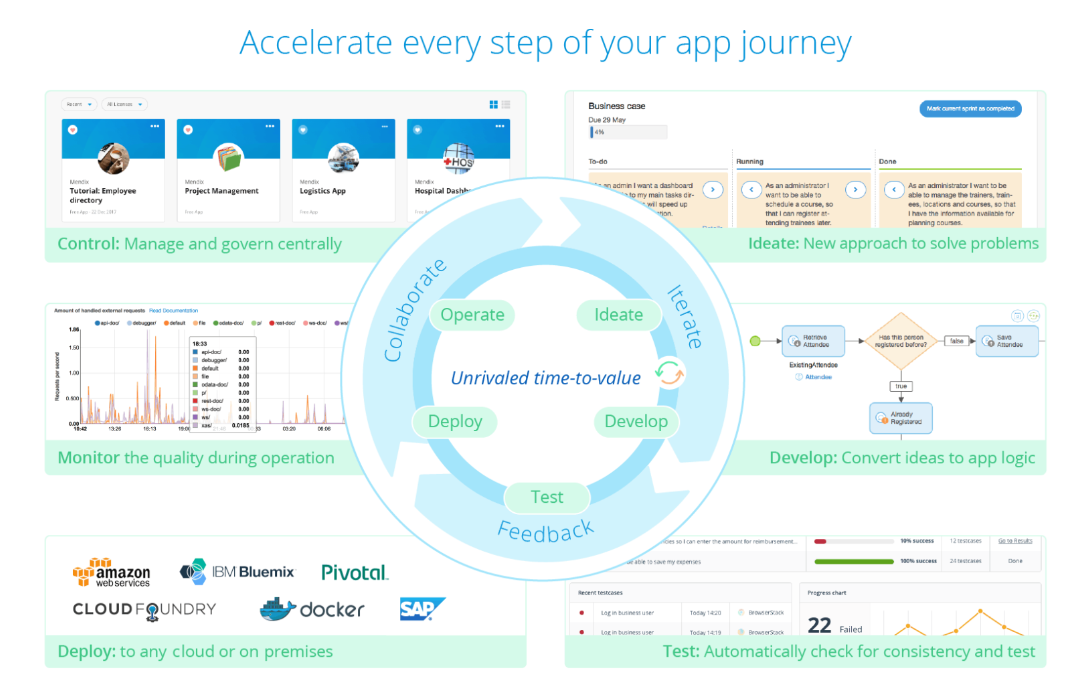
* Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON.
* Lập chỉ mục trên bất kỳ thuộc tính nào.
* Các truy vấn đa dạng.
* Cập nhật nhanh hơn.

### Lưu ý khi sử dụng MongoDB

* MongoDB sinh ra để tăng tốc độ truy xuất dữ liệu, phù hợp cho các ứng dụng cần tốc độ phản hồi nhanh (realtime như facebook chẳng hạn). Còn các tác nghiệp cần tính toàn vẹn dữ liệu (trong banking) thì NoSQL sẽ không bao giờ là 1 giải pháp cả mà người ta sẽ dùng MySQL.
* MongoDB không có tính ràng buộc, một điều tồi tệ trong Database vì vậy sẽ rất cần sự cẩn thận khi thao tác trên các collection có quan hệ dữ liệu với nhau.
* MongoDB đẩy trách nghiệm thao tác Database cho tầng ứng dụng nên sẽ tốn tài nguyên (tài nguyên bây giờ không còn là vấn đề quá lớn nữa).
* MongoDB có thể mở trộng theo chiều ngang (scale out) phương pháp tăng cường khả năng lưu trữ và xử lý là dùng nhiều máy tính phân tán. MongoDB còn có thể mở rộng theo chiều dọc (scale up) tăng cấu hình server.
* MongoDB có thể dùng nhiều máy tính phân tán để lưu trữ dữ liệu nên chi phí sẽ rẻ hơn MySQL. MySQL sử dụng những máy chủ hàng khủng, độc quyền nên sẽ đắt đỏ hơn.

## Tổng quan về Mendix

* Mendix là một nền tảng low-code dùng để phát triển ứng dụng hàng đầu trong ngành giúp các tổ chức xây dựng và phát triển ứng dụng cấp doanh nghiệp trên quy mô lớn.
* Nền tảng được thiết kế để tăng tốc vòng đời phát triển, từ ý tưởng đến triển khai và vận hành, cho phép cộng tác ở mỗi bước.
* Mendix cung cấp công cụ bao gồm Mendix Studio và Mendix Studio Pro trong một nền tảng tích hợp đầy đủ và duy nhất:
* Mendix Studio cung cấp cho người dùng một giao diện được dùng để tạo ứng dụng một cách trực quan mà không cần sử dụng đến bất kỳ một dòng lệnh nào và có thể được sử dụng ngay trên website mà Mendix cung cấp.
* Mendix Studio Pro là một studio giúp người dùng có toàn quyền để phát triển ứng dụng một cách đầy đủ nhất, giúp người dùng có thể tạo ra ứng dụng trên nền tảng web và mobile một cách dễ dàng, nhanh chóng.
* Nhờ sự kết hợp này, các chuyên gia kinh doanh (nhà phân tích và phát triển) có thể làm việc cùng với các nhà phát triển ứng dụng để đạt được sự phối hợp và phát triển ứng dụng nhanh chóng, hiệu quả hơn. Hơn nữa, các công cụ tự động hóa và kiến trúc cloud-native hỗ trợ triển khai, quản lý và giám sát ứng dụng doanh nghiệp có tính khả dụng cao.



1. Vòng đời của ứng dụng (Mendix là gì? Cách Mendix-Low-Code-Platform Hoạt động, 2021)

* Bước đầu trong vòng đời của ứng dụng là lập ý tưởng. Mendix cung cấp một cách tiếp cận mới nhằm giải quyết các vấn đề, tạo điều kiện cho việc quản lý yêu cầu của dự án phát triển ứng dụng. Mendix hỗ trợ các stories cũng như sprint nhằm quản lý các công việc hằng ngày của các thành viên trong nhóm theo mô hình SCRUM. Đồng thời tăng tốc độ xử lý các vấn đề.
* Mendix hỗ trợ phát triển ứng dụng và chuyển đổi các ý tưởng thành ứng dụng. Nhà phân tích kinh doanh và nhà phát triển kinh doanh có thể sử dụng Mendix Studio để xây dựng các ứng dụng mẫu, khai thác ý tưởng cho các nhà phát triển ứng dụng, đồng thời cộng tác với các nhà phát triển để phát triển ứng dụng (Kyanon Digital, (2021).
* Product Owner và Scrum Master có thể sử dụng các cổng thông tin của nhà phát triển Mendix để quản lý các yêu cầu từ người dùng, phản hồi người dùng và quản lý các công việc có trong Sprint (Kyanon Digital, (2021)).

# PHÂN TÍCH THIẾT KẾ

## Xác định yêu cầu

### Yêu cầu hệ thống

* Hệ điều hành Windows 7 trở lên.
* Thiết bị android hoặc iOS.
* Cài đặt Browser (Chrome, Cốc Cốc).
* Có kết nối internet.

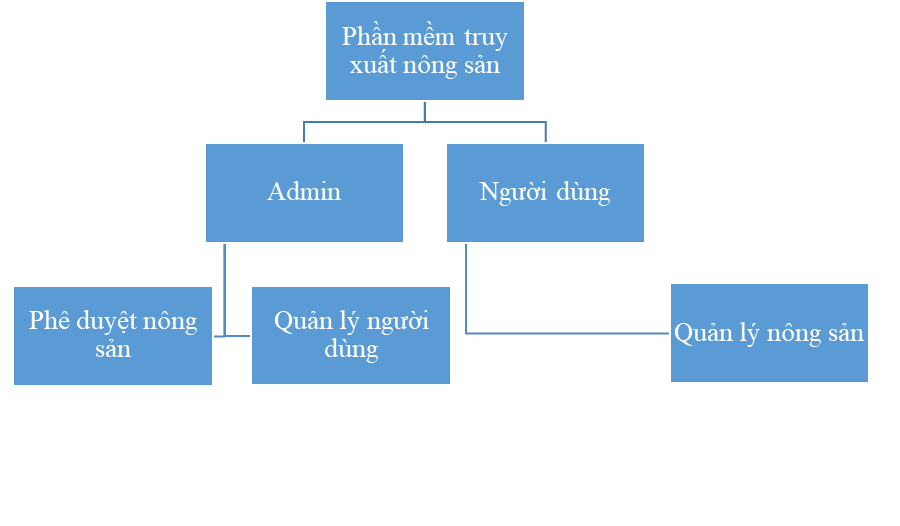
### Yêu cầu chức năng

* Chức năng đăng ký và đăng nhập.
* Chức năng phân quyền.
* Chức năng quản lý nông sản cho phép thêm, xóa, sửa và lưu trữ các loại nông sản, tự động thêm mã QR mỗi khi thêm dữ liệu.
* Chức năng quản lý quy trình 5 bước cho từng loại nông sản.
* Chức năng tìm kiếm tùy theo mục đích sử dụng.
* Chức năng duyệt mã QR để xem các loại nông sản.

### Yêu cầu phi chức năng

* Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với mọi người dùng.
* Cho phép truy cập dữ liệu đa người dùng.
* Tốc độ xử lí của hệ thống phải nhanh chóng và chính xác.
* Người sử dụng phần mềm có thể sẽ không biết nhiều về tin học nhưng vẫn sử dụng một cách dễ dàng nhờ vào sự trợ giúp của hệ thống.
* Phần hướng dẫn sử dụng phần mềm phải ngắn gọn, dễ hiểu và sinh động.

## Sơ đồ chức năng quản lý



1. Sơ đồ chức năng quản lý

## Mô hình hóa yêu cầu

### Đăng ký người dùng

* Người dùng đăng ký tài khoản dùng để đăng nhập vào ứng dụng. Sử dụng tài khoản google để đăng ký, sẽ tự động kiểm tra hệ thống nếu như tài khoản không tồn tại, nếu có sẽ tự động đăng nhập.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình đăng ký người dùng

* Các ký hiệu:
* D1: Các thông đăng ký (Tài khoản google).
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến đăng ký người dùng.
* D4: Lưu người dùng mới vào cơ sở dữ liệu.
* D5: Không có.
* D6: Trả về danh sách người dùng sau khi thêm thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Kiểm tra thông tin nhập vào đã đúng form và có trong cơ sở dữ liệu chưa.
* Bước 5. Nếu chưa có thông tin thêm D1 vào cơ sở dữ liệu.
* Bước 6. Trả danh sách nhân viên cho người dùng tại D6.
* Bước 7. Đóng kết nối cơ sở dữ liệu.
* Bước 8. Kết thúc.

### Đăng nhập

* Người dùng đăng nhập vào tài khoản để thực hiện những chức năng chính trong ứng dụng. Sử dụng tài khoản Google để đăng nhập. Khi đăng nhập sẽ tạo ra 1 token, 1 vài dữ liệu quan trọng như thông tin nông sản và quy trình 5 bước sẽ cần token để truy cập. Nếu token hết hạn, khi người dùng chuyển trang để truy cập 1 chức năng nào đó, trang sẽ tự động chuyển về trang chủ.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình đăng nhập

* Các ký hiệu:
* D1: Các thông tin đăng nhập.
* D2: Không có.
* D3: Kiểm tra thông tin đăng nhập.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin người dùng đã đăng nhập.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nếu D1 có trong cơ sở dữ liệu.
* Bước 5. Đưa đến phiên làm việc dựa trên quyền hạn của tài khoản đã đăng nhập.
* Bước 6. Trả về thông tin tài khoản người dùng vừa đăng nhập.
* Bước 7. Đóng kết nối cơ sở dữ liệu.
* Bước 8. Kết thúc.

### Tìm kiếm nông sản trên trang chủ

* Nhằm tìm kiếm nông sản theo mong muốn người dùng. Người dùng có thể tìm kiếm theo tên, theo danh mục và sắp xếp theo độ phổ biến, mới nhất.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tìm kiếm nông sản trang chủ

* Các ký hiệu:
* D1: Các thông tin nông sản người dùng nhập trên trang chủ.
* D2: Không có.
* D3: Kiểm tra thông tin nông sản.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin sau khi tìm kiếm nông sản.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nếu D1 có trong cơ sở dữ liệu.
* Bước 5. Đưa ra các loại nông sản tương ứng với dữ liệu được nhập.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc.

### Tìm kiếm nông sản

* Nhằm tìm kiếm nông sản theo mong muốn người dùng. Chức năng này được sử dụng trong quản lý nông sản và chỉ tìm được các nông sản của người dùng đó.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tìm kiếm nông sản

* Các ký hiệu:
* D1: Các thông tin nông sản người dùng nhập.
* D2: Không có.
* D3: Kiểm tra thông tin nông sản.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin sau khi tìm kiếm nông sản.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nếu D1 có trong cơ sở dữ liệu.
* Bước 5. Đưa ra các loại nông sản tương ứng với dữ liệu được nhập.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc.

### Thêm nông sản

* Nhằm tạo nông sản theo mong muốn của người dùng, có thể thêm các thông tin của loại nông sản đó, sau khi tạo xong, nông sản sẽ vào trạng thái chờ admin phê duyệt trước khi chính thức được đưa vào hoạt động.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình thêm nông sản

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra thông tin nông sản.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin sau khi thêm nông sản.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nếu D1 có trong cơ sở dữ liệu.
* Bước 5. Đưa thông báo nông sản đã tồn tại trong cơ sở dữ liệu.
* Bước 6. Nếu D1 không có trong cơ sở dữ liệu, tiến hành tạo nông sản.
* Bước 7. Đóng kết nối cơ sở dữ liệu.
* Bước 8. Kết thúc.

### Tùy chỉnh nông sản

* Tùy chỉnh thông tin chi tiết của 1 loại nông sản, không bao gồm các bước trong quy trình 5 bước. Chỉ áp dụng với loại nông sản đang trong trạng thái chờ chấp thuận hoặc từ chối, nếu ở trạng thái chấp thuận sẽ không thể tùy chỉnh được nữa.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tùy chỉnh nông sản

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành cập nhật nông sản.
* D4: Cập nhật nông sản vào cơ sở dữ liệu.
* D5: Không có.
* D6: Trả về danh sách nông sản sau khi cập nhật thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu. Kiểm tra dữ liệu trong cơ sở dữ liệu có trùng với D1.
* Bước 4. Nếu thỏa bước 3 và thông tin có thay đổi thì cập nhật thông tin nông sản vào cơ sở dữ liệu.
* Bước 5. Trả danh sách nông sản cho người dùng tại D6.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc

### Xóa nông sản

* Dùng để xóa 1 loại nông sản, khi sử dụng chức năng này, sẽ hiện thông báo để xác nhận cho người dùng, nếu sản phẩm đã được chấp thuận, không thể xóa nông sản được nữa. Đối với trường hợp bài viết vẫn ở trạng thái chờ chấp thuận hoặc từ chối, bài viết sẽ được xóa ngay lập tức.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình xóa nông sản

* Các ký hiệu:
* D1: Thông tin nông sản cần xóa.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành xóa nông sản.
* D4: Xóa nông sản ra khỏi cơ sở dữ liệu.
* D5: Không có.
* D6: Hiển thị thông báo xóa nông sản thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Kiểm tra thông tin nông sản đã đúng form và có trong cơ sở dữ liệu chưa.
* Bước 5. Nếu đã có thông tin thì sẽ tiến hành xóa nông sản.
* Bước 6. Trả danh sách nông sản cho người dùng tại D6.
* Bước 7. Đóng kết nối cơ sở dữ liệu.
* Bước 8. Kết thúc.

### Tìm kiếm quy trình 5 bước

* Nhằm tìm kiếm quy trình 5 bước theo mong muốn người dùng.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tìm kiếm quy trình 5 bước

* Các ký hiệu:
* D1: Các thông tin quy trình 5 bước người dùng nhập.
* D2: Không có.
* D3: Kiểm tra thông tin nông sản.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin sau khi tìm kiếm nông sản.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nếu D1 có trong cơ sở dữ liệu.
* Bước 5. Đưa ra các loại nông sản tương ứng với dữ liệu được nhập.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc.

### Thêm quy trình 5 bước

* Nhằm thêm 1 quy trình cho 1 loại nông sản cụ thể, có thể tạo nông sản ở 1 trong 5 bước kèm theo ghi chú.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình thêm quy trình 5 bước

* Các ký hiệu:
* D1: Các thông tin quy trình 5 bước.
* D2: Không có.
* D3: Kiểm tra thông tin quy trình 5 bước.
* D4: không có.
* D5: Không có.
* D6: Trả về thông tin sau khi thêm quy trình 5 bước.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Kiểm tra D1 trong cơ sở dữ liệu.
* Bước 4. Nhập đầy đủ thông tin, báo lỗi nếu nhập thiếu.
* Bước 5. Tiến hành tạo quy trình 5 bước.
* Bước 6. Hiện thông báo tạo quy trình 5 bước thành công.
* Bước 7. Đóng kết nối cơ sở dữ liệu.
* Bước 8. Kết thúc.

### Tùy chỉnh quy trình 5 bước

* Được sử dụng khi người dùng muốn tùy chỉnh quy trình 5 bước, trong vòng 1 tiếng tính từ lúc tạo bước, nếu không sửa đổi, chức năng này sẽ bị khóa nhằm tạo tính minh bạch cho nông sản.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tùy chỉnh quy trình 5 bước

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành cập nhật quy trình 5 bước.
* D4: Cập nhật quy trình 5 bước sản vào cơ sở dữ liệu.
* D5: Không có.
* D6: Trả về danh sách quy trình 5 bước sau khi cập nhật thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu. Kiểm tra xem thời gian chỉnh sửa có quá 1 tiếng so với thời gian tạo hay không (tránh việc sửa đổi quá khung giờ quy định là 1 giờ, sẽ được so sánh với thời gian thực tế), đồng thời kiểm tra xem nội dung có thay đổi với nội dung gốc hay không (Sử dụng thư viện sentence-similarity để kiểm tra nội dung, nội dung có thể giống 50% so với nội dung gốc).
* Bước 4. Nếu thỏa bước 3 và thông tin có thay đổi thì cập nhật thông tin quy trình 5 bước vào cơ sở dữ liệu.
* Bước 5. Trả danh sách quy trình 5 bước cho người dùng tại D6.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc

### Xóa quy trình 5 bước

* Dùng để xóa quy trình không mong muốn, có sai sót hoặc vì lí do khác. Trong vòng 15 phút, nếu không xóa, quy trình sẽ không thể xóa được.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình xóa quy trình 5 bước

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành xóa quy trình 5 bước.
* D4: Xóa quy trình 5 bước sản trong cơ sở dữ liệu.
* D5: Không có.
* D6: Sau khi xóa sẽ trả thông báo dựa trên hoàn cảnh, có thể thất bại nếu gặp sự cố và thành công nếu không. Trả về danh sách quy trình 5 bước sau khi cập nhật thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Nếu thỏa bước 3 và thông tin có thay đổi thì xóa thông tin quy trình 5 bước vào cơ sở dữ liệu.
* Bước 5. Sau khi xóa sẽ trả thông báo dựa trên hoàn cảnh, có thể thất bại nếu gặp sự cố và thành công nếu không. Trả danh sách quy trình 5 bước cho người dùng tại D6.
* Bước 6. Đóng kết nối cơ sở dữ liệu.
* Bước 7. Kết thúc

### Duyệt nông sản

* Sau khi người dùng tạo 1 loại nông sản, admin có nhiệm vụ duyệt nông sản, nếu chấp thuận, bài viết sẽ được cập nhật thành chấp thuận bên phía người dùng và admin. Với tính năng này, admin phải accept thủ công.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình duyệt nông sản

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành duyệt nông sản.
* D4: Duyệt quy trình 5 bước sản vào cơ sở dữ liệu.
* D5: Không có.
* D6: Trả về danh sách nông sản khi duyệt thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu. Kiểm tra xem thông tin nông sản có phù hợp hay không, nếu có nhấn chấp thuận, nếu không nhấn từ chối.
* Bước 4. Hiển thị thông báo đã chấp thuận hay từ chối.
* Bước 5. Đóng kết nối cơ sở dữ liệu.
* Bước 6. Kết thúc

### Quản lý tài khoản

* Giúp người dùng phân quyền admin được phép thay đổi phân quyền của các tài khoản có trong cơ sở dữ liệu. Có 2 quyền là admin và user. Với phân quyền user chỉ được truy cập đến chức năng quản lý nông sản, với phân quyền admin sẽ được truy cập đến duyệt nông sản và quản lý tài khoản. Trang chủ hiển thị chung cho cả 2 phân quyền.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình quản lý tài khoản

* Các ký hiệu:
* D1: Các thông tin nông sản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành phân quyền người dùng.
* D4: Phân quyền người dùng và lưu vào cơ sở dữ liệu.
* D5: Không có.
* D6: Trả về danh sách tài khoản khi cập nhật thành công.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Hiển thị thông báo phân quyền thành công.
* Bước 5. Đóng kết nối cơ sở dữ liệu.
* Bước 6. Kết thúc

### Tìm kiếm tài khoản

* Giúp người dùng tìm kiếm tài khoản theo Email, Quyền, Ngày tạo trong mục duyệt tài khoản dành cho admin.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tìm kiếm tài khoản

* Các ký hiệu:
* D1: Các thông tin tài khoản.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành tìm kiếm tài khoản.
* D4: Cơ sở dữ liệu tìm kiếm thông tin từ người dùng.
* D5: Không có.
* D6: Trả về danh sách người dùng sau khi tìm kiếm.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Hiển thị danh sách người dùng sau khi tìm kiếm.
* Bước 5. Đóng kết nối cơ sở dữ liệu.
* Bước 6. Kết thúc

### Tìm kiếm nông sản theo mã QR

* Giúp người dùng tìm kiếm nông sản dựa trên mã QR quét được.

Người dùng

D6

D1

Thiết bị xuất

D5

Cơ sở dữ liệu

D4

D3

D2

Thiết bị nhập

1. Mô hình tìm kiếm nông sản theo mã QR

* Các ký hiệu:
* D1: Mã trích xuất được sau khi scan QR.
* D2: Không có.
* D3: Kiểm tra D1 nếu đủ thông tin thì tiến hành tìm kiếm nông sản.
* D4: Cơ sở dữ liệu tìm kiếm thông tin từ mã QR được cung cấp.
* D5: Không có.
* D6: Trả về nông sản cần tìm.
* Thuật toán:
* Bước 1. Nhận D1 từ người dùng.
* Bước 2. Kết nối cơ sở dữ liệu.
* Bước 3. Đọc D1 từ cơ sở dữ liệu.
* Bước 4. Hiển thị nông sản sau khi tìm kiếm.
* Bước 5. Đóng kết nối cơ sở dữ liệu.
* Bước 6. Kết thúc

## Kết quả thực nghiệm

### Mô hình trình tự hoạt động

Khởi động phần mềm

Người dùng nhập thông tin tài khoản

Hệ thống kiểm tra

Giao diện phần mềm chính

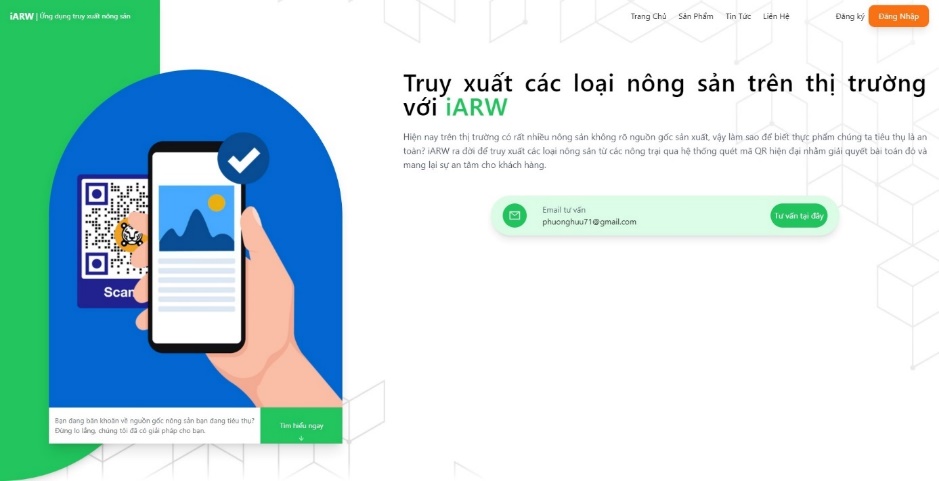
Hiển thị thông báo không thành công

Thực hiện quản lý

1. Mô hình trình tự hoạt động

### Giao diện Landing Page

* Giới thiệu cho người dùng các thông báo mới nhất của sản phẩm.



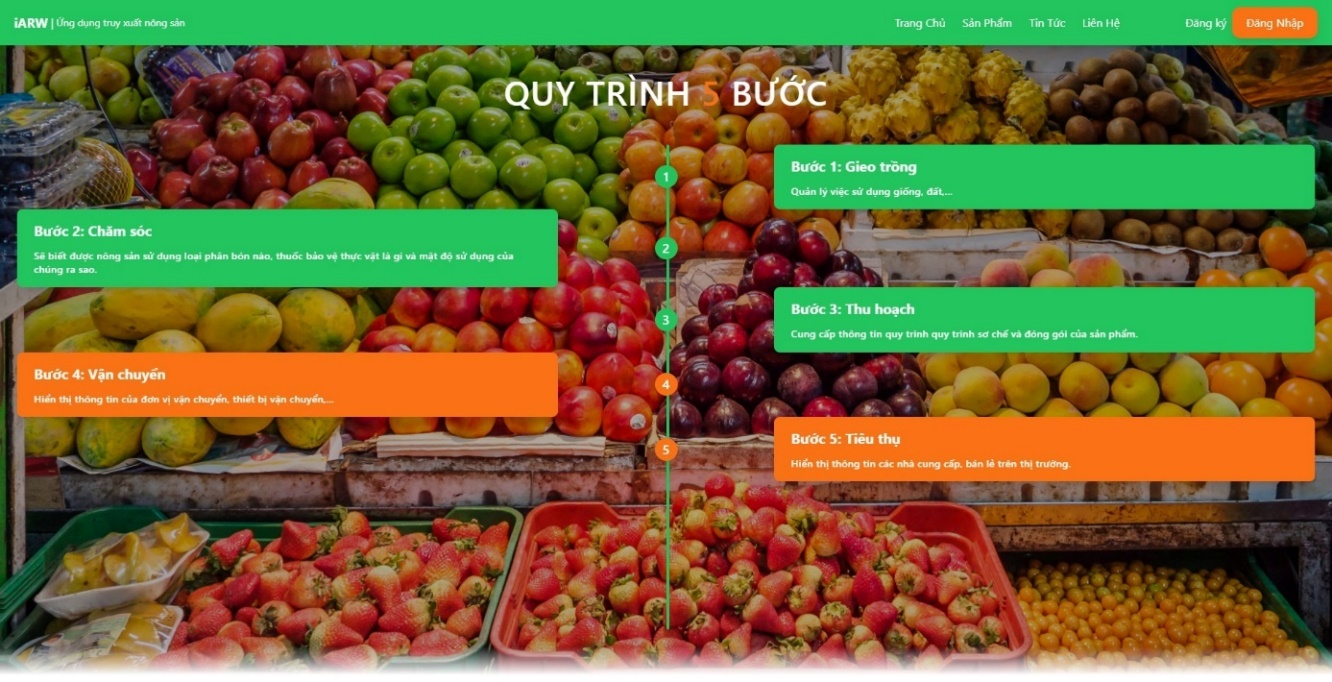
1. Giao diện Landing Page 1

* Giới thiệu về tạo mã truy xuất nguồn gốc nông sản gồm 4 bước như trong mô tả.



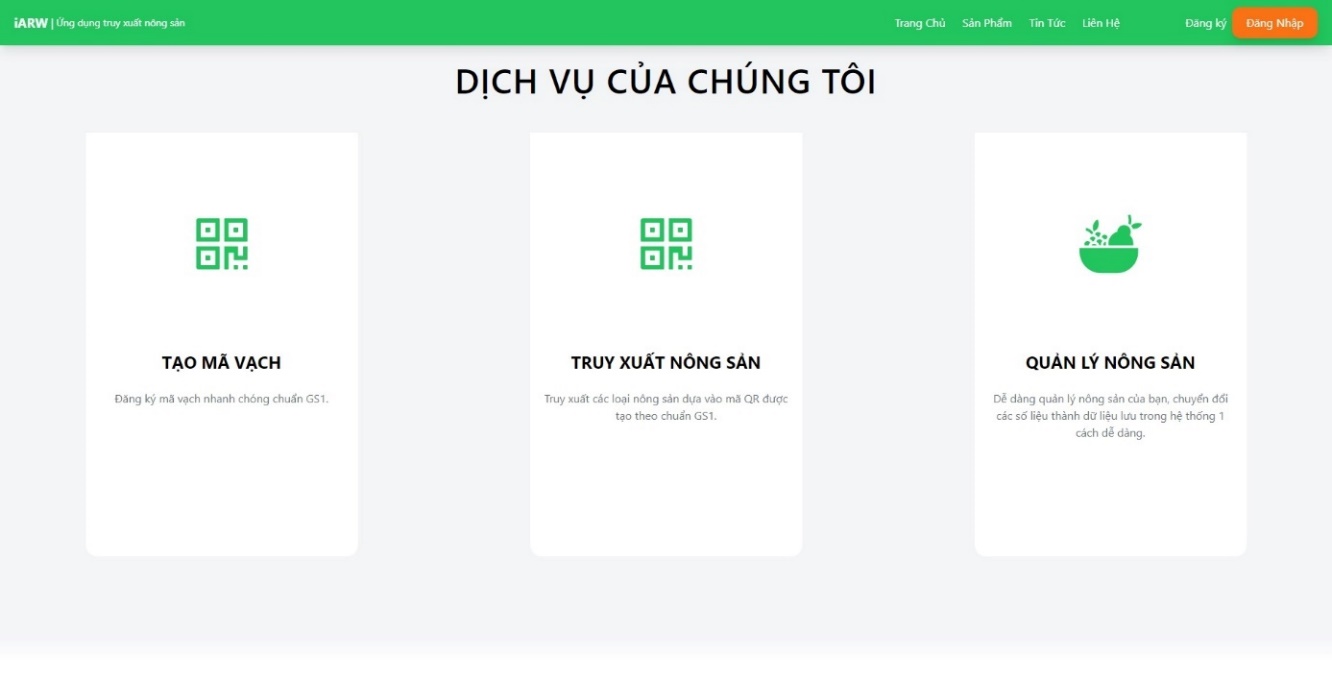
1. Giao diện Landing Page 2

* Giới thiệu sơ lược về quy trình 5 bước để khách hàng hiểu thêm về các thức hoạt động của nó.



1. Giao diện Landing Page 3

* Giới thiệu về các dịch vụ mà ứng dụng cung cấp.



1. Giao diện Landing Page 4

* Nếu người dùng có câu hỏi về ứng dụng sẽ điền đầy đủ thông tin và nhấn Liên hệ.

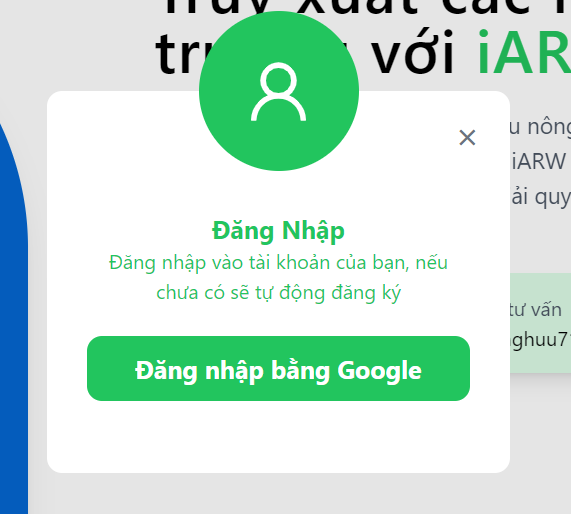


1. Giao diện Landing Page 5
2. Bảng giao diện trang liên hệ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | txtEmail | TextField | Tên email |
| 2 | txtName | TextField | Họ Tên |
| 3 | txtQuestion | TextField | Thắc mắc |
| 4 | btnContact | Button | Nút Liên hệ |

### Giao diện đăng nhập

* Khi mở, hệ thống sẽ hiện lên giao diện đăng nhập.
* Sử dụng tài khoản Google để đăng nhập và đăng ký.
* Sau khi đăng nhập thành công, tùy vào loại tài khoản là người dùng hay Admin, người dùng có thể thực hiện các tác vụ trong quyền hạn cho phép.

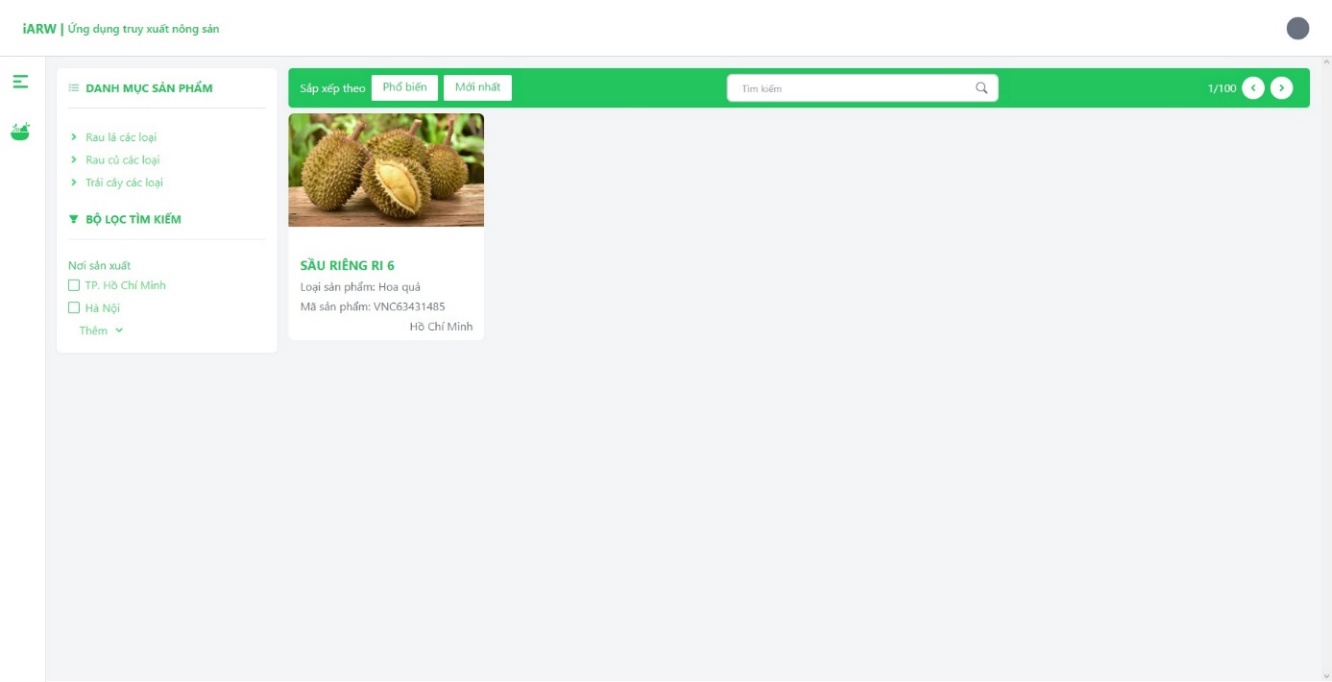


1. Trang đăng nhập
2. Bảng giao diện trang đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | btnLogin | Button | Đăng nhập |

### Giao diện chính Dashboard

* Trang chủ của giao diện Dashboard, cho phép người dùng xem các nông sản của người dùng khác.
* Có các chức năng Filter giúp người dùng dễ tìm kiếm sản phẩm.
* Có các chức năng sắp xếp theo độ phổ biến và mới nhất
* Khi bấm vào sản phẩm sẽ đưa đến trang thông tin chi tiết của sản phẩm đó.

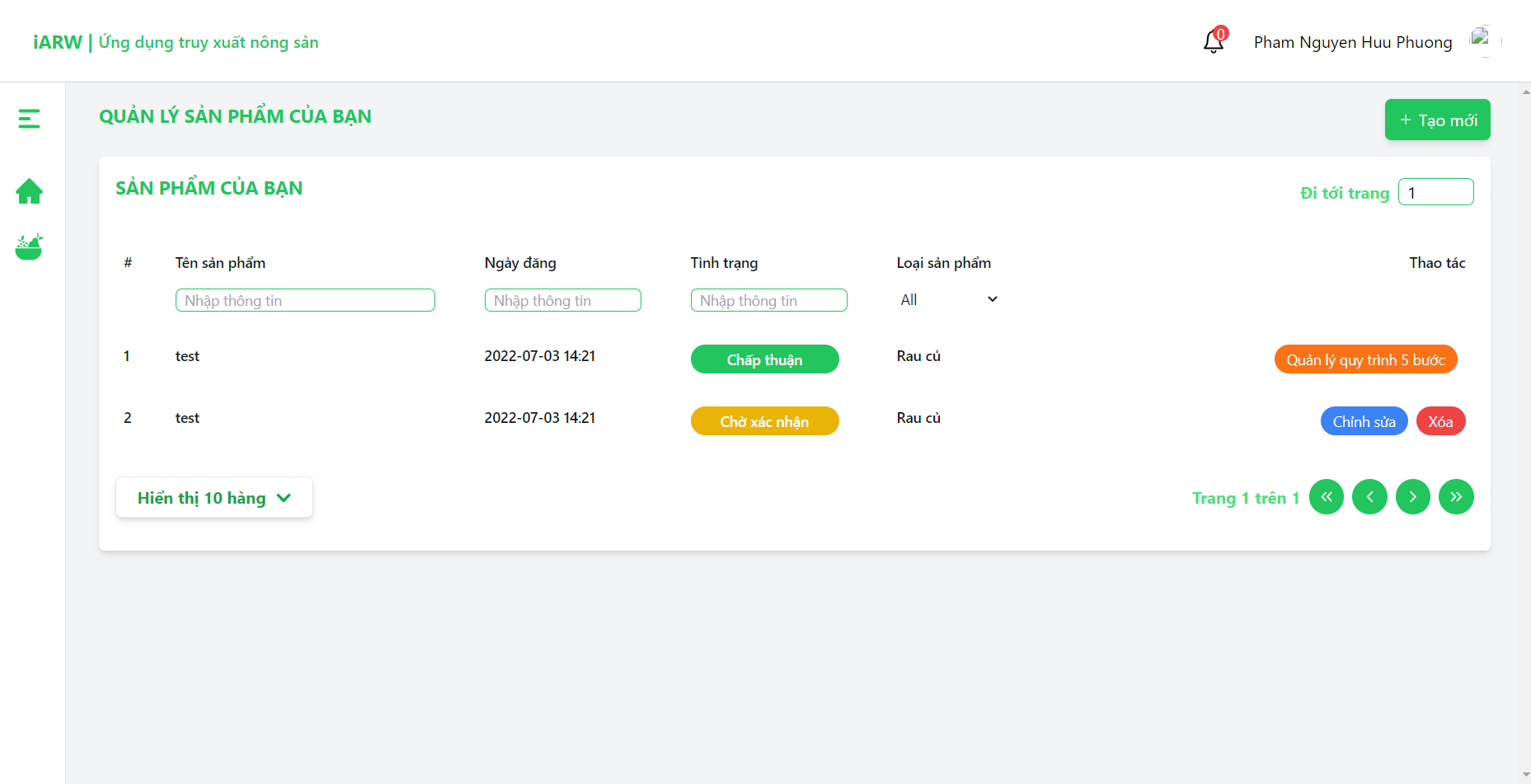


1. Trang giao diện chính dashboard
2. Bảng giao diện trang giao diện chính dashboard

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | btnLatest | Button | Nút tìm kiếm mới nhất |
| 2 | btnMostFamous | Button | Nút tìm kiếm phổ biến |
| 3 | lbVeg1 | Label | Tìm kiếm theo loại rau lá |
| 4 | lbVeg2 | Label | Tìm kiếm theo loại rau củ |
| 5 | lbFruits | Label | Tìm kiếm theo trái cây |
| 6 | chkProducedPlace | Checkbox | Checkbox tìm nơi sản xuất |
| 7 | txtSearch | TextField | Thanh tìm kiếm |

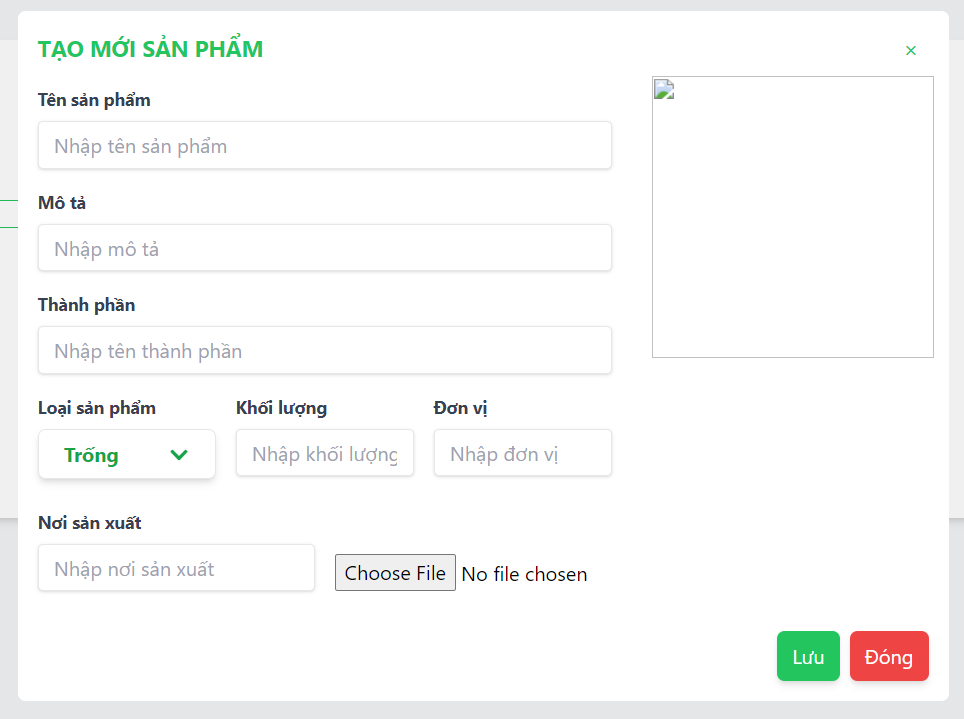
### Giao diện quản lý nông sản

* Người dùng có thể tạo, chỉnh sửa, tìm kiếm, xóa nông sản.
* Người dùng có thể tìm kiếm theo từng field của nông sản.
* Khi ở trạng thái chấp thuận, chỉ hiển thị nút quy trình 5 bước.
* Khi ở trạng thái từ chối hoặc chờ xác nhận chỉ hiển thị nút Tùy chỉnh và xóa.
* Người dùng có thể quản lý quy trình 5 bước của nông sản.
* Mỗi loại nông sản có 1 quy trình 5 bước riêng.



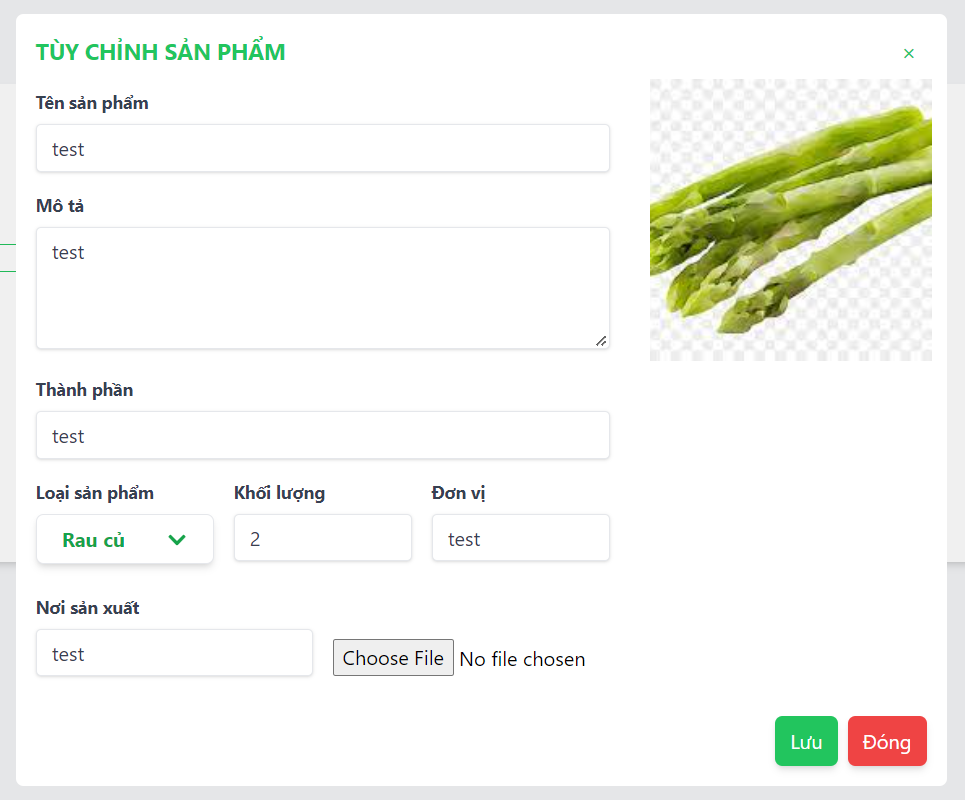
1. Trang quản lý nông sản khi chấp thuận

* Được sử dụng khi người dùng muốn tạo mới một loại nông sản, có các trường để nhập và có thể upload hình ảnh.



1. Pop-Up tạo mới sản phẩm

* Được sử dụng khi người dùng muốn tạo mới một loại nông sản, có các trường để nhập và có thể upload hình ảnh.

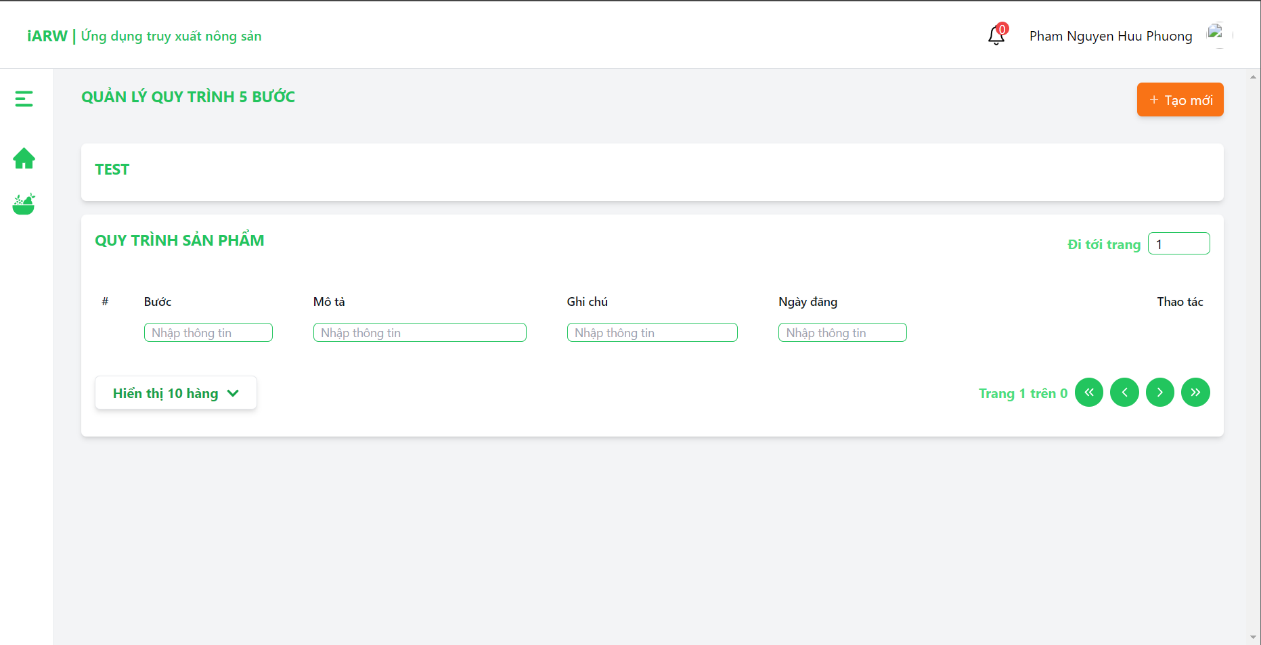


1. Pop-Up tùy chỉnh sản phẩm
2. Bảng giao diện trang quản lý nông sản

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | btnCreate | Button | Nút tạo nông sản |
| 2 | txtSearch | TextField | Thanh tìm kiếm |
| 3 | btnSearch | Button | Nút tìm kiếm |
| 4 | btnProcess | Button | Nút quản lý quy trình 5 bước |
| 5 | btnDelete | Button | Nút xóa nông sản |
| 6 | btnEdit | Button | Nút tùy chỉnh nông sản |

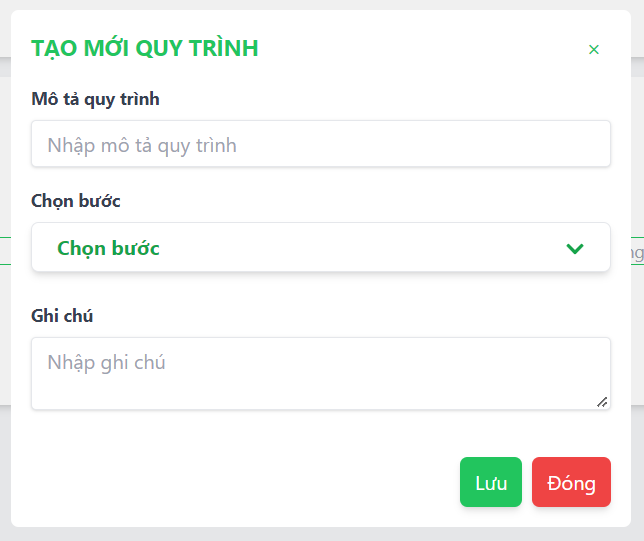
### Giao diện quản lý quy trình 5 bước

* Hiển thị các quy trình 5 bước của 1 sản phẩm bao gồm từ giao đoạn gieo trồng đến giai đoạn tiêu thụ.
* Hiển thị thêm các thông tin chi tiết như ngày đăng, mốc thời gian đăng.
* Thêm, xóa, sửa sẽ hiện pop-up riêng tương ứng với từng quá trình thao tác.
* Có thể thêm ghi chú tương ứng với từng quy trình.



1. Trang quản lý quy trình 5 bước

* Được sử dụng để tạo mới 1 quy trình của 1 loại nông sản riêng biệt.



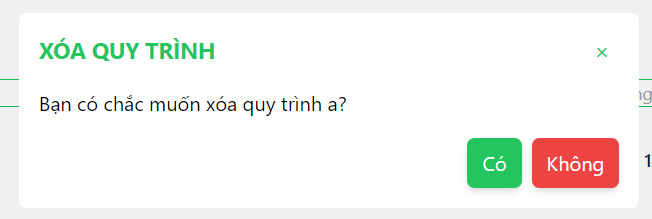
1. Pop-up tạo mới quy trình

* Được sử dụng để tùy chỉnh 1 quy trình của 1 loại nông sản riêng biệt.



1. Pop-up tùy chỉnh quy trình

* Được sử dụng để xóa 1 quy trình của 1 loại nông sản riêng biệt.

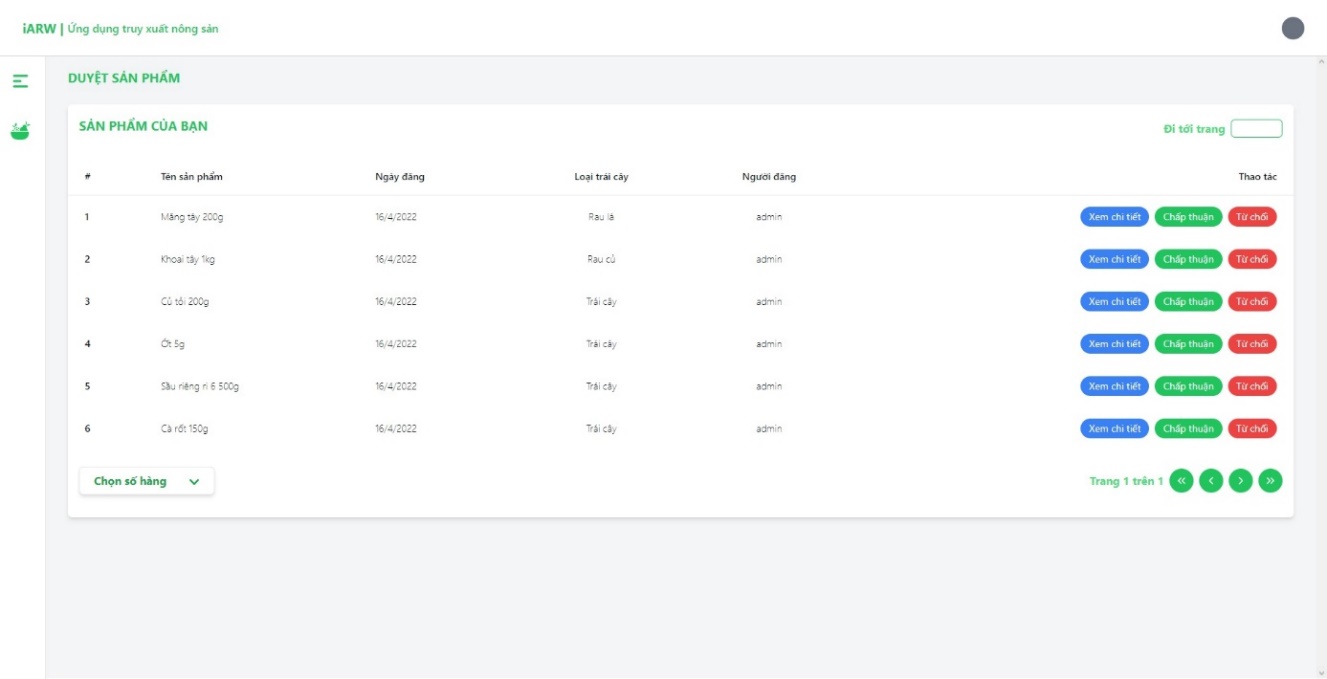


1. Pop-up xóa quy trình
2. Bảng giao diện trang quản lý quy trình 5 bước

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | btnCreate | Button | Nút tạo quy trình 5 bước |
| 2 | txtSearch | TextField | Thanh tìm kiếm |
| 3 | btnSearch | Button | Nút tìm kiếm |
| 5 | btnDelete | Button | Nút xóa quy trình 5 bước |
| 6 | btnEdit | Button | Nút tùy chỉnh quy trình 5 bước |

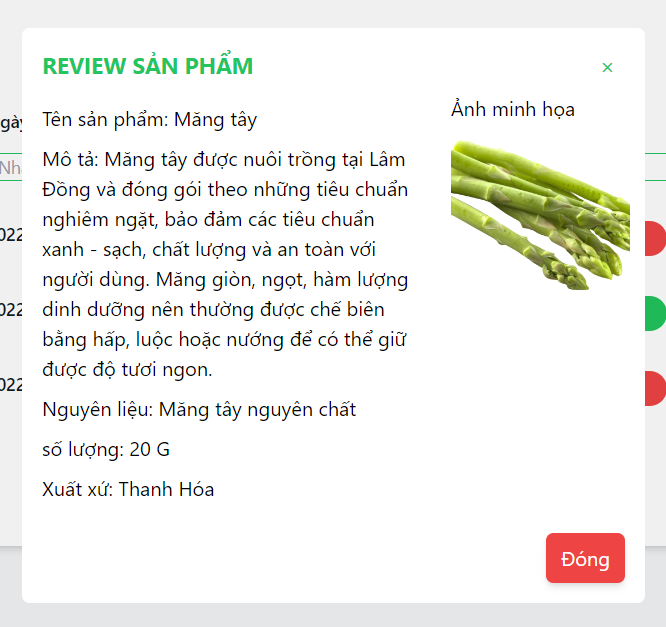
### Giao diện duyệt nông sản

* Sau khi người dùng tạo nông sản thành công, nông sản sẽ được đưa vào hàng chờ.
* Chỉ sau khi Admin chấp thuận thì nó mới được hiển thị trên trang chủ, đồng thời sẽ chuyển trạng thái của nông sản tương ứng với việc Admin chấp thuận hay từ chối.



1. Trang duyệt nông sản

* Khi nhấn xem chi tiết nông sản sẽ hiển thị Pop-Up để xem thông tin của loại nông sản đó.

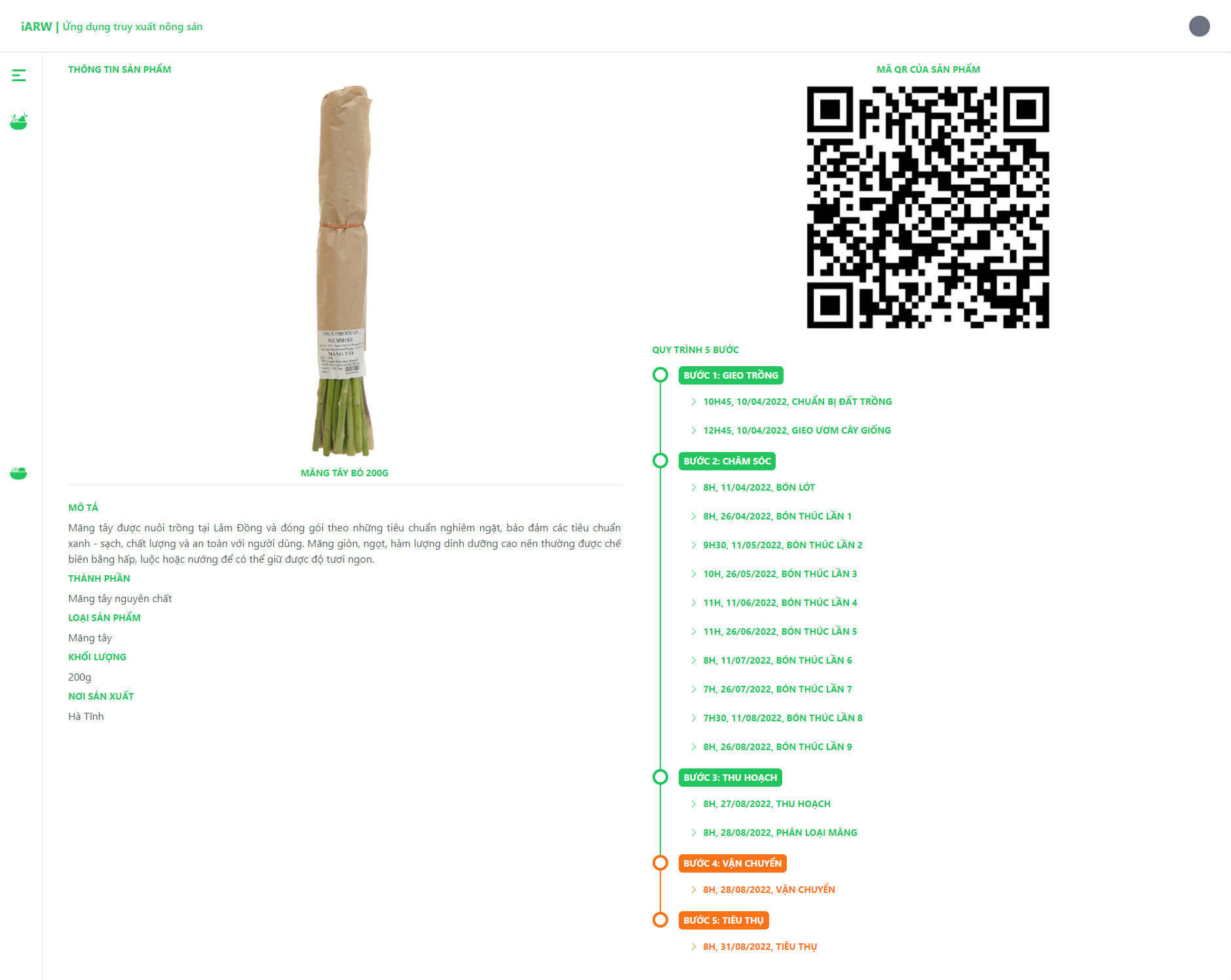


1. Hình ảnh chi tiết review sản phầm
2. Bảng giao diện trang duyệt nông sản

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Control** | **Loại** | **Ý nghĩa** |
| 1 | btnDetails | Button | Nút xem chi tiết nông sản |
| 2 | btnAcp | Button | Nút chấp thuận |
| 3 | btnDec | Button | Nút từ chối |

### Giao diện chi tiết nông sản

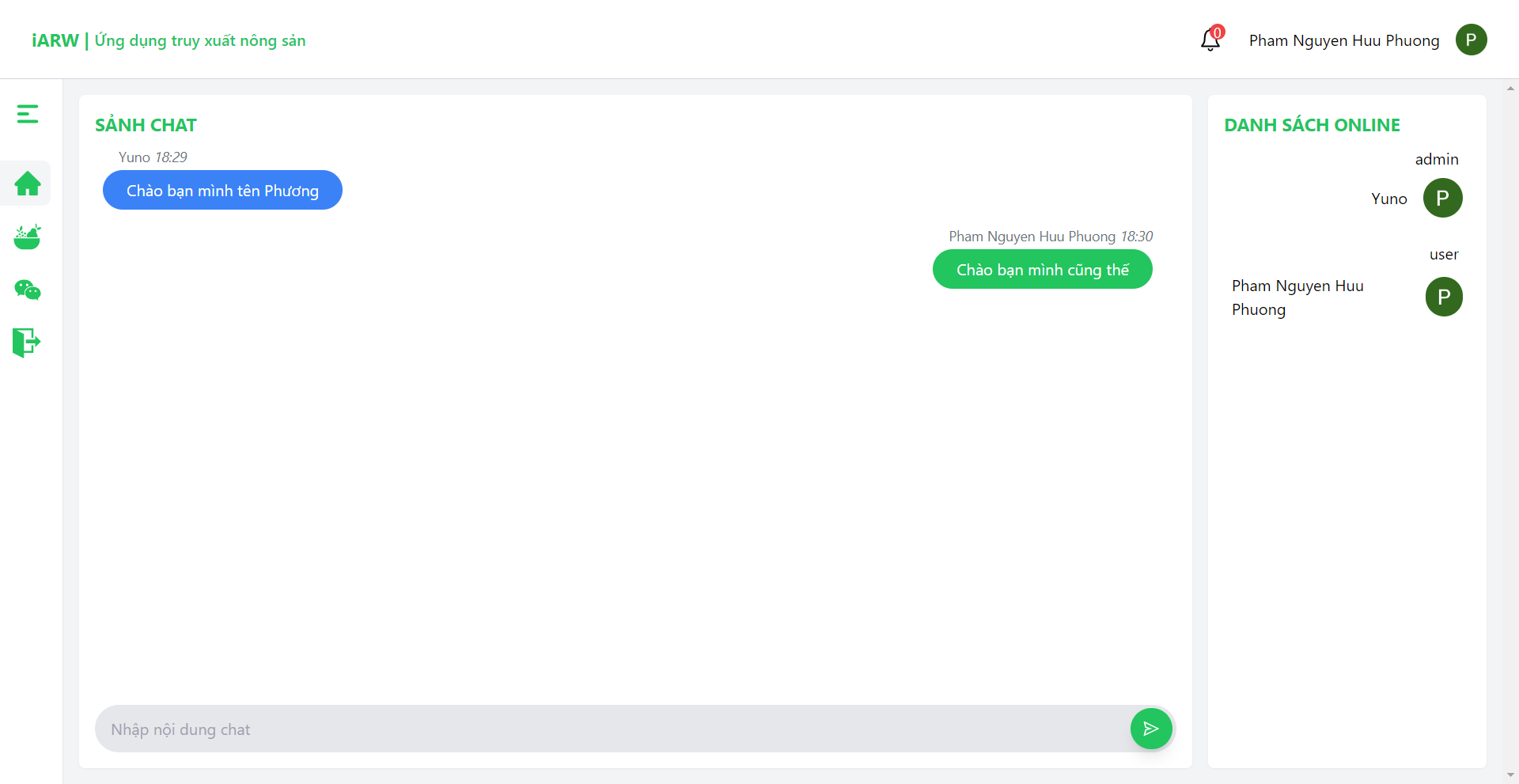
* Giao diện dùng để xem chi tiết của 1 loại nông sản.
* Bao gồm các thông tin từ mô tả đến quy trình 5 bước của loại nông sản đó.
* Ngoài ra khi người dùng tạo sản phẩm, 1 mã QR sẽ được tạo tự động theo dạng GS1.
* Thông tin của mã QR có thể check tại đây.



1. Trang chi tiết nông sản

### Giao diện chat

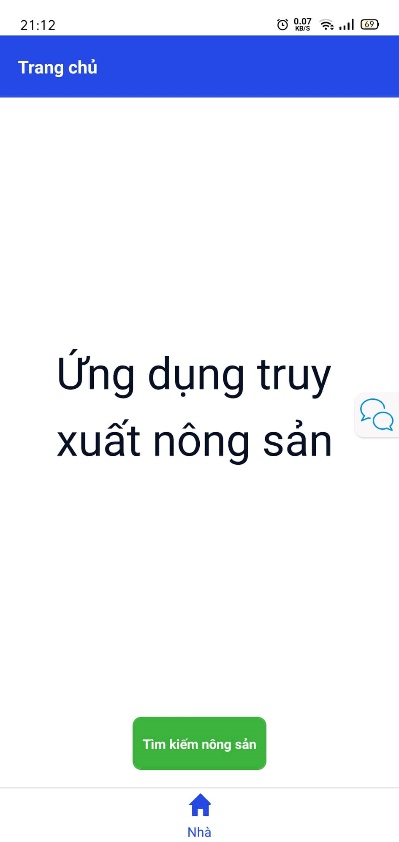
* Giao diện sử dụng socket.io để chat giữa các người dùng với nhau, ở bên trái là khung chat, ở bên phải là danh sách những thành viên online và có ghi phân quyền cũng như thông tin cơ bản của người dùng đó.



1. Trang giao diện chat

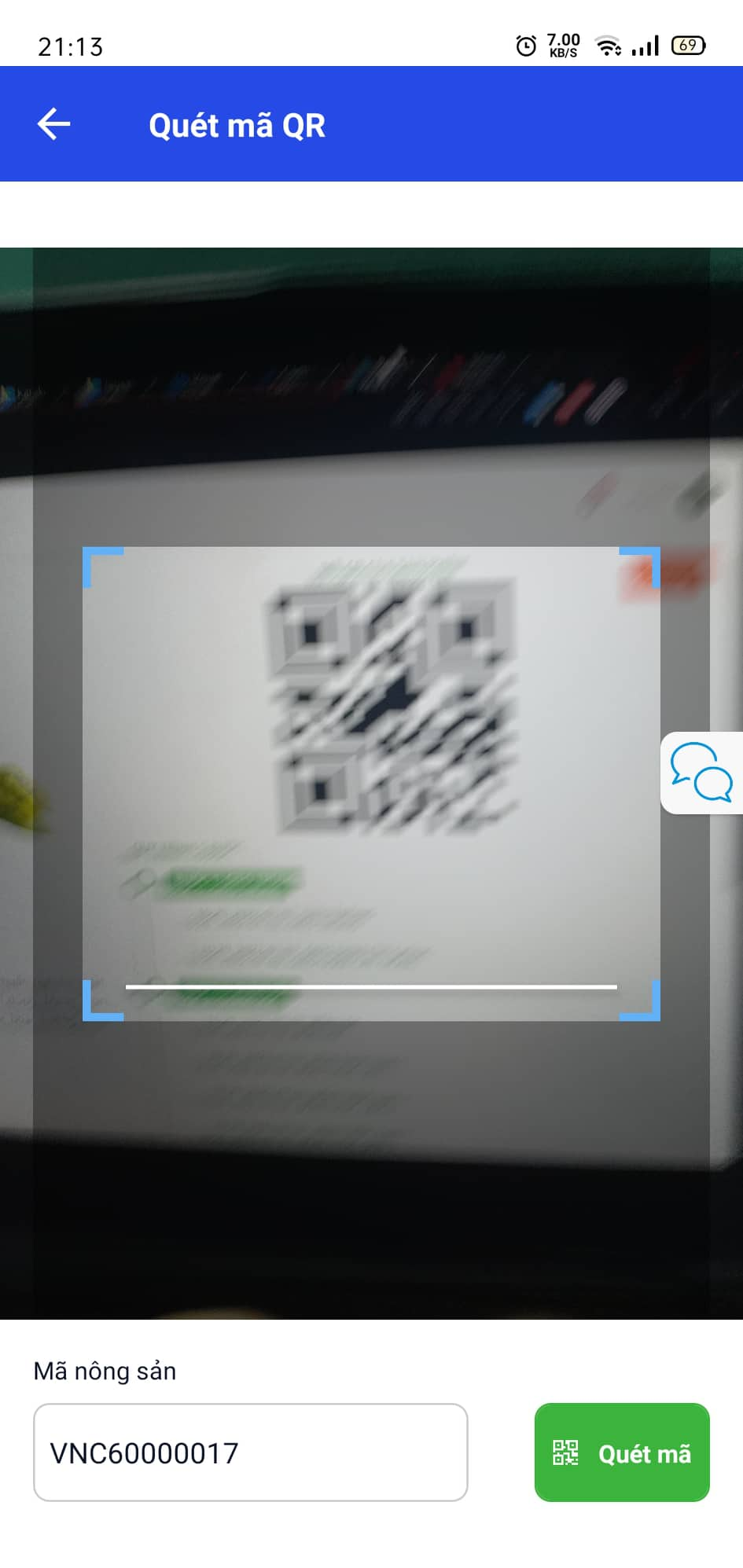
### Giao diện tìm kiếm nông sản theo mã QR trên điện thoại

* Giao diện dùng để tìm kiếm nông sản theo mã QR của sản phẩm đó.



1. Giao diện chính sản phẩm

* Được sử dụng để quét mã QR nhằm tìm loại nông sản dựa trên mã QR đó.



1. Giao diện quét mã QR

* Nhằm hiển thị thông tin loại nông sản sau khi quét mã QR.



1. Giao diện chi tiết sản phẩm

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **Kết luận**

* Truy xuất được nguồn gốc nông sản theo thời gian thực thông qua quy trình 5 bước.
* Sử dụng công nghệ truy xuất nguồn gốc sản phẩm, công nghệ QR Code giúp cho việc kiểm soát hàng hóa minh bạch, rõ ràng hơn.
* Sản phẩm ứng dụng công nghệ vào công tác quản lý, giúp nâng cao hiệu quả và năng suất làm việc so với công tác tổ chức truyền thống.
* Giao diện thân thiện, gần gũi với người dùng.
* Sử dụng dễ dàng, nhanh chóng, ít tốn thời gian.
* Tuy nhiên, khóa luận vẫn đang trong quá trình phát triển, các chức năng đang được hoàn thiện, nên vẫn còn xuất hiện lỗi.

1. **Hướng phát triển**

* Hoàn thiện những chức năng đang trong quá trình phát triển.
* Bổ sung thêm tính năng mua bán sản phẩm, so sánh và đề xuất các sản phẩm cùng loại.
* Ứng dụng và phát triển các tính năng sử dụng socket.io, nhằm đưa ứng dụng phát triển theo hướng real-time.
* Bổ sung thêm hướng blockchain nhằm nâng cao bảo mật hệ thống.

# TÀI LIỆU THAM KHẢO

[1] Kyanon Digital. (2021). Mendix là gì? Cách Mendix-Low-Code-Platform Hoạt động. https://kyanon.digital/mendix-la-gi-cach-mendix-low-code-platform-hoat-dong/

[2] Đoàn Văn Phú. (2020). Tổng quan về NodeJS - Một số điểm khác biệt với PHP. https://viblo.asia/p/tong-quan-ve-nodejs-mot-so-diem-khac-biet-voi-php-1Je5E9945nL

[3] Thân Hoàng. (2020). [Networking] Socket hoạt động như thế nào?. https://viblo.asia/p/networking-socket-hoat-dong-nhu-the-nao-aWj53LxYK6m

[4] Hoàng Nguyễn. (2019). Tìm hiểu về Tailwind CSS. https://viblo.asia/p/tim-hieu-ve-tailwind-css-924lJp6WKPM

[5] Đỗ Khánh Toàn. (2017). Giới thiệu về redux. https://viblo.asia/p/gioi-thieu-ve-redux-63vKj67VK2R

[6] Đỗ Khánh Toàn. (2017). Giới thiệu về ReactJS – Phần I (Các khái niệm cơ bản). https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7

[7] Wikipedia. (2017). Visual Studio Code. https://vi.wikipedia.org/wiki/Visual\_Studio\_Code

[8] Nguyễn Văn Hiển. (2016). Tổng quan về MongoDB. https://viblo.asia/p/tong-quan-ve-mongodb-EoDkQoxqGbV

[9] Wikipedia. (2013). React. https://en.wikipedia.org/wiki/React\_(JavaScript\_library)

[10] Wikipedia. (2009). NodeJS. https://en.wikipedia.org/wiki/Node.js



Khóa luận tốt nghiệp