

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

**XÂY DỰNG ỨNG DỤNG MOBILE HỖ TRỢ QUẢN LÝ
TÀI CHÍNH THEO MÙA VỤ CHO NÔNG DÂN**

Người hướng dẫn: TS. TRƯƠNG NGỌC CHÂU

Sinh viên thực hiện: TRƯƠNG TUẤN KIỆT

Số thẻ sinh viên: 102200176

Lớp: 20TCLC_DT4

Đà Nẵng, 01/2026

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

**XÂY DỰNG ỨNG DỤNG MOBILE HỖ TRỢ QUẢN LÝ
TÀI CHÍNH THEO MÙA VỤ CHO NÔNG DÂN**

Người hướng dẫn: TS. TRƯƠNG NGỌC CHÂU

Sinh viên thực hiện: TRƯƠNG TUẤN KIỆT

Số thẻ sinh viên: 102200176

Lớp: 20TCLC_DT4

Đà Nẵng, 01/2026

TÓM TẮT

Tên đề tài: Xây dựng ứng dụng mobile hỗ trợ quản lý tài chính theo mùa vụ cho nông dân

Sinh viên thực hiện: Trương Tuấn Kiệt

Số thẻ SV: 102200176

Lớp: 20TCLC_DT4

Tóm tắt nội dung đồ án:

Hiện nay, quá trình quản lý chi phí và tài chính trong lĩnh vực nông nghiệp tại Việt Nam vẫn còn nhiều bước thao tác thủ công, sử dụng sổ sách giấy tờ để ghi chép các giao dịch thu chi, dẫn đến việc tiêu tốn đáng kể thời gian và công sức, đồng thời dễ phát sinh sai sót trong quá trình ghi chép và tính toán, gây ảnh hưởng đến tính chính xác và hiệu quả của việc quản lý tài chính. Việc này không chỉ làm mất thời gian mà còn khiến cho việc theo dõi và đánh giá hiệu quả kinh doanh của các mùa vụ trở nên khó khăn hơn, đặc biệt là khi cần tra cứu thông tin lịch sử hoặc tạo báo cáo tổng hợp. Hơn nữa, việc quản lý kho hàng và vật liệu một cách thủ công cũng dễ dẫn đến tình trạng thiếu hụt hoặc lãng phí tài nguyên, ảnh hưởng trực tiếp đến lợi nhuận của người nông dân và chủ trang trại. Điều này đòi hỏi cần có giải pháp số hóa để giảm thiểu rủi ro sai sót, tiết kiệm thời gian và giúp người dùng quản lý tài chính nông nghiệp một cách thuận tiện và chính xác hơn.

Xây dựng hệ thống quản lý chi phí nông nghiệp để có thể số hóa quy trình quản lý tài chính, từ việc theo dõi các mùa vụ, ghi nhận giao dịch thu chi, quản lý kho hàng đến tạo báo cáo thống kê, giúp giảm bớt các thao tác ghi chép và tính toán thủ công, tốn nhiều thời gian và tránh được các sai sót trong quá trình quản lý. Ngoài ra, hệ thống sẽ cải thiện việc hỗ trợ quản lý tài chính nông nghiệp hiệu quả thông qua việc tự động hóa nhập liệu bằng AI, cung cấp các công cụ báo cáo và thống kê trực quan, đồng thời đảm bảo được tính chính xác, minh bạch và khả năng theo dõi của các hoạt động tài chính trong nông nghiệp.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Trương Tuấn Kiệt

Số thẻ sinh viên: 102200176

Lớp: 20TCLC_DT4

Khoa: Công nghệ thông tin

Ngành: Công nghệ phần

mềm

1. Tên đề tài đồ án: Xây dựng ứng dụng mobile hỗ trợ quản lý tài chính theo mùa vụ cho nông dân.
2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
3. Các số liệu và dữ liệu ban đầu
4. Nội dung các phần thuyết minh và tính toán:
 - Mở đầu
 - Chương 1: Tổng quan đề tài
 - Chương 2: Cơ sở lý thuyết
 - Chương 3: Phân tích và thiết kế hệ thống
 - Chương 4: Triển khai và đánh giá kết quả
 - Kết luận
5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):
 - Sơ đồ use case
 - Sơ đồ hoạt động
 - Sơ đồ kiến trúc hệ thống
6. Họ tên người hướng dẫn: TS. Trương Ngọc Châu
7. Ngày giao nhiệm vụ đồ án: 08 / 10 / 2025
8. Ngày hoàn thành đồ án: 22 / 01 / 2026

Đà Nẵng, ngày 22 tháng 01 năm 2026

Trưởng Bộ môn.....

Người hướng dẫn

LỜI NÓI ĐẦU

Trong thời gian thực hiện đề án, với sự giúp đỡ và tạo điều kiện của trường Đại học Bách Khoa – Đại học Đà Nẵng, em đã hoàn thiện đề án cùng với bản báo cáo đúng thời gian quy định. Với khả năng và thời gian có hạn nên không thể tránh khỏi những thiếu sót, em rất mong nhận được sự quan tâm, giúp đỡ, tạo điều kiện của quý thầy cô để em có thể hoàn thiện hơn đề tài nghiên cứu trong thời gian tới.

Em xin chân thành cảm ơn đến TS. Trương Ngọc Châu – giảng viên khoa Công nghệ thông tin – Trường Đại học Bách Khoa Đà Nẵng đã tận tình hỗ trợ và theo sát tiến độ của em trong suốt quá trình hoàn thành dự án này.

Mặc dù đã cố gắng hoàn thiện đề án trong phạm vi và khả năng có thể nhưng khó tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp, nhận xét và đánh giá của quý Thầy Cô.

Đà Nẵng, ngày 22 tháng 01 năm 2026

Sinh viên thực hiện

CAM ĐOAN

Em xin cam đoan rằng đề án tốt nghiệp "Xây dựng ứng dụng mobile hỗ trợ quản lý tài chính theo mùa vụ cho nông dân" là kết quả nghiên cứu và làm việc của cá nhân em, dưới sự hướng dẫn của thầy TS. Trương Ngọc Châu. Mọi tham khảo, thông tin, số liệu và kết quả trình bày trong đề án đều trích dẫn, trình bày chính xác và trung thực.

Em xin chịu hoàn toàn trách nhiệm về tính trung thực và chính xác của nội dung đề án này. Nếu có bất kỳ sai sót hoặc vi phạm bản quyền nào được phát hiện, em xin hoàn toàn chịu trách nhiệm trước Hội đồng chấm đề án tốt nghiệp và các quy định của nhà trường.

Đà Nẵng, ngày 22 tháng 01 năm 2026

Sinh viên thực hiện

MỤC LỤC

MỞ ĐẦU.....	1
Chương 1: TỔNG QUAN VỀ HỆ THỐNG.....	2
1.1 Tên đề tài.....	2
1.2 Mục đích đề tài.....	2
1.3 Mục tiêu cụ thể.....	2
Chương 2: CƠ SỞ LÝ THUYẾT	5
2.1 Xây dựng giao diện.....	5
2.1.1 Dart	5
2.1.2 Flutter.....	5
2.2 Xây dựng Back-end	6
2.2.1 NestJs	6
2.3 Cơ sở dữ liệu	7
2.3.1 MongoDB	7
2.4 Giao tiếp Client - Server	7
2.4.1 RESTful API.....	7
2.5 Kết chương.....	8
Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	9
3.1 Phân tích yêu cầu hệ thống.....	9
3.2 Thiết kế hệ thống	10
3.2.1 Sơ đồ ca sử dụng.....	10
3.2.2 Quản lý truy cập.....	10
3.2.3 Quản lý mùa vụ.....	17
3.2.3 Quản lý giao dịch.....	25
3.2.4 Quản lý kho hàng	31
3.2.5 Báo cáo và thống kê.....	36
3.3 Biểu đồ hoạt động	40
3.3.1 Quản lý truy cập.....	40
3.3.2 Quản lý mùa vụ.....	46
3.3.3 Quản lý giao dịch.....	51
3.3.4 Quản lý vật tư	56
3.3.5 Quản lý minh chứng	59
3.4 Kiến trúc hệ thống.....	61
3.4.1 Client.....	63
3.4.2 Server	63
3.5 Thiết kế cơ sở dữ liệu.....	64
3.5.1 Hệ quản trị cơ sở dữ liệu.....	64
3.5.2 Mô hình tổ chức dữ liệu.....	64
3.5.3 Cơ sở dữ liệu.....	64

3.6 Kết chương	67
Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.....	68
4.1 Triển khai và kết quả hệ thống	68
4.2 Kết chương	81
KẾT LUẬN	82
TÀI LIỆU THAM KHẢO.....	84

DANH MỤC BẢNG BIỂU

Bảng 3.1 Đặc tả use case UC – 001.....	11
Bảng 3.2 Đặc tả use case UC – 002.....	12
Bảng 3.3 Đặc tả use case UC – 003.....	13
Bảng 3.4 Đặc tả use case UC – 004.....	14
Bảng 3.5 Đặc tả use case UC – 005.....	14
Bảng 3.6 Đặc tả use case UC – 006.....	15
Bảng 3.7 Đặc tả use case UC – 007.....	16
Bảng 3.8 Đặc tả use case UC – 008.....	17
Bảng 3.9 Đặc tả use case UC – 101.....	19
Bảng 3.10 Đặc tả use case UC – 102.....	19
Bảng 3.11 Đặc tả use case UC – 103.....	20
Bảng 3.12 Đặc tả use case UC – 104.....	21
Bảng 3.13 Đặc tả use case UC – 105.....	21
Bảng 3.14 Đặc tả use case UC – 106.....	22
Bảng 3.15 Đặc tả use case UC – 107.....	23
Bảng 3.16 Đặc tả use case UC – 108.....	24
Bảng 3.17 Đặc tả use case UC – 109.....	24
Bảng 3.18 Đặc tả use case UC – 201.....	26
Bảng 3.19 Đặc tả use case UC – 202.....	27
Bảng 3.20 Đặc tả use case UC – 203.....	27
Bảng 3.21 Đặc tả use case UC – 204.....	28
Bảng 3.22 Đặc tả use case UC – 205.....	29
Bảng 3.23 Đặc tả use case UC – 206.....	29
Bảng 3.24 Đặc tả use case UC – 207.....	30
Bảng 3.25 Đặc tả use case UC – 208.....	31
Bảng 3.26 Đặc tả use case UC – 301.....	32
Bảng 3.27 Đặc tả use case UC – 302.....	33
Bảng 3.28 Đặc tả use case UC – 303.....	33
Bảng 3.29 Đặc tả use case UC – 304.....	34
Bảng 3.30 Đặc tả use case UC – 305.....	35
Bảng 3.31 Đặc tả use case UC – 306.....	35
Bảng 3.32 Đặc tả use case UC – 307.....	36
Bảng 3.33 Đặc tả use case UC – 401.....	37
Bảng 3.34 Đặc tả use case UC – 402.....	37
Bảng 3.35 Đặc tả use case UC – 403.....	38
Bảng 3.36 Đặc tả use case UC – 404.....	39
Bảng 3.37 Đặc tả use case UC – 405.....	39
Bảng 3.38 Mô tả cấu trúc dữ liệu User.....	65
Bảng 3.39 Mô tả cấu trúc dữ liệu Seasons.....	65
Bảng 3.40 Mô tả cấu trúc dữ liệu Transactions.....	66
Bảng 3.41 Mô tả cấu trúc dữ liệu Inventory.....	66
Bảng 3.42 Mô tả cấu trúc dữ liệu InventoryLogs.....	67
Bảng 3.43 Mô tả quan hệ giữa các collection.....	67

DANH MỤC HÌNH ẢNH

Hình 2.1 Logo Dart.....	5
Hình 2.2 Logo Flutter	6
Hình 2.3 Logo NestJS.....	6
Hình 2.4 Logo MongoDB.....	7
Hình 2.5 RESTful API.....	8
Hình 3.1 Sơ đồ Usecase tổng quát	10
Hình 3.2 Sơ đồ Usecase quản lý truy cập.....	10
Hình 3.3 Sơ đồ Usecase quản lý mùa vụ.....	17
Hình 3.4 Sơ đồ Usecase quản lý giao dịch.....	25
Hình 3.5 Sơ đồ Usecase quản lý kho hàng	31
Hình 3.6 Sơ đồ Usecase báo cáo và thống kê.....	36
Hình 3.7 Sơ đồ hoạt động đăng ký tài khoản	40
Hình 3.8 Sơ đồ hoạt động Đăng nhập.....	41
Hình 3.9 Sơ đồ hoạt động Quên mật khẩu.....	42
Hình 3.10 Sơ đồ hoạt động Xác minh OTP.....	42
Hình 3.11 Sơ đồ hoạt động Đặt lại mật khẩu	43
Hình 3.12 Sơ đồ hoạt động Đổi mật khẩu	44
Hình 3.13 Sơ đồ hoạt động Xem thông tin profile	44
Hình 3.14 Sơ đồ hoạt động Chỉnh sửa thông tin profile.....	45
Hình 3.15 Sơ đồ hoạt động Tạo mùa vụ mới.....	46
Hình 3.16 Sơ đồ hoạt động Xem danh sách mùa vụ	46
Hình 3.17 Sơ đồ hoạt động Xem chi tiết mùa vụ	47
Hình 3.18 Sơ đồ hoạt động Chỉnh sửa mùa vụ.....	47
Hình 3.19 Sơ đồ hoạt động Xoá mùa vụ	48
Hình 3.20 Sơ đồ hoạt động Xem thống kê mùa vụ	48
Hình 3.21 Sơ đồ hoạt động Thêm thành viên vào mùa vụ	49
Hình 3.22 Sơ đồ hoạt động Xoá thành viên khỏi mùa vụ.....	50
Hình 3.23 Sơ đồ hoạt động Quản lý vật liệu của mùa vụ.....	50
Hình 3.24 Sơ đồ hoạt động Tạo giao dịch mới.....	51
Hình 3.25 Sơ đồ hoạt động Xem danh sách giao dịch.....	51
Hình 3.26 Sơ đồ hoạt động Xem chi tiết giao dịch	52
Hình 3.27 Sơ đồ hoạt động Chỉnh sửa giao dịch.....	52
Hình 3.28 Sơ đồ hoạt động Xoá giao dịch.....	53
Hình 3.29 Sơ đồ hoạt động Lọc giao dịch.....	53
Hình 3.30 Sơ đồ hoạt động Phê duyệt giao dịch	54
Hình 3.31 Sơ đồ hoạt động Nhập giao dịch từ hình ảnh	55
Hình 3.32 Sơ đồ hoạt động Thêm vật tư mới	56
Hình 3.33 Sơ đồ hoạt động Xem danh sách kho hàng.....	56
Hình 3.34 Sơ đồ hoạt động Xem chi tiết vật liệu	57
Hình 3.35 Sơ đồ hoạt động Chỉnh sửa vật liệu.....	57
Hình 3.36 Sơ đồ hoạt động Xoá vật liệu	58
Hình 3.37 Sơ đồ hoạt động Xem lịch sử sử dụng vật liệu.....	58
Hình 3.38 Sơ đồ hoạt động Xem lịch sử sử dụng vật liệu theo mùa vụ.....	59
Hình 3.39 Sơ đồ hoạt động Xem báo cáo tổng quan.....	59
Hình 3.40 Sơ đồ hoạt động Xuất báo cáo PDF.....	60
Hình 3.41 Sơ đồ hoạt động Xuất báo cáo CSV	60

Hình 3.42 Sơ đồ hoạt động Chia sẻ báo cáo.....	61
Hình 3.43 Kiến trúc hệ thống	62
Hình 3.44 Cơ sở dữ liệu hệ thống.....	64
Hình 4.1 Màn hình Splash	69
Hình 4.2 Màn hình Đăng nhập	70
Hình 4.3 Màn hình Đăng ký	71
Hình 4.4 Màn hình Nhập mã OTP.....	72
Hình 4.5 Màn hình Trang chủ	73
Hình 4.6 Màn hình Danh sách mùa vụ	74
Hình 4.7 Màn hình Chi tiết mùa vụ	75
Hình 4.8 Màn hình Thống kê.....	76
Hình 4.9 Màn hình Danh sách giao dịch	77
Hình 4.10 Màn hình Tạo giao dịch.....	78
Hình 4.11 Màn hình Danh sách vật liệu	79
Hình 4.12 Màn hình Báo cáo tổng quan.....	80
Hình 4.13 Màn hình Xem profile	81

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Chữ viết tắt	Mô tả
1	API	Application Programming Interface
2	HTTP	Hypertext Transfer Protocol
3	HTTPS	Hypertext Transfer Protocol Secure
4	JWT	JSON Web Token
5	OTP	One-Time Password
6	AI	Artificial Intelligence

MỞ ĐẦU

1. Lí do chọn đề tài

Hiện nay, nông dân gặp rất nhiều khó khăn trong việc quản lý tài chính, chi tiêu, tính toán lợi nhuận trong các hoạt động về nông nghiệp. Việc ghi chép thủ công dễ dẫn đến sai sót, thất thoát ngân sách, khó theo dõi hiệu quả kinh tế. Cộng với việc nhà nước và các tổ chức đang thúc đẩy nông nghiệp thông minh, điện thoại thông minh ngày càng phổ biến, tạo điều kiện cho nông dân tiếp cận các giải pháp công nghệ. Đặc biệt, ứng dụng di động là công cụ tiện lợi, dễ sử dụng, phù hợp với đặc thù công việc đồng áng của nông dân. Việc phát triển ứng dụng di động hỗ trợ góp phần nâng cao đời sống cũng như giảm thiểu rủi ro về mặt tài chính cho nông dân.

2. Mục đích thực hiện đề tài

Xây dựng ứng dụng mobile giúp nông dân dễ dàng quản lý tài chính. Hỗ trợ ghi chép thu nhập, chi phí, lợi nhuận giúp nông dân theo dõi tài chính một cách hiệu quả. Cung cấp thêm công cụ tích hợp AI giúp nông dân phân tích hình ảnh các giao dịch.

3. Phạm vi và đối tượng nghiên cứu

a. Phạm vi

Ứng dụng được thiết kế dành cho nông dân và các đối tượng, tổ chức tại các khu vực nông thôn, hướng tới việc cung cấp công cụ hỗ trợ để cải thiện quy trình quản lý tài chính, tối ưu hoá thời gian và công sức.

b. Đối tượng nghiên cứu

- Nông dân làm việc trong lĩnh vực nông nghiệp cần một công cụ để quản lý thu nhập, chi phí và các hoạt động sản xuất.
- Các đối tượng, tổ chức khác có nhu cầu về việc quản lý tài chính và các hoạt động sản xuất như hợp tác xã nông nghiệp,...

4. Phương pháp nghiên cứu

- Tiến hành thu thập thông tin về cách quản lý tài chính hiện tại của nông dân, xác định những khó khăn và nhu cầu thực tế.
- Phân tích mức độ tiếp cận công nghệ của nông dân để thiết kế ứng dụng đơn giản và thân thiện.
- Phân tích yêu cầu, xác định các tính năng chính của ứng dụng về quản lý tài chính.
- Xác định các yêu cầu về kỹ thuật và các ràng buộc về bảo mật đảm bảo dữ liệu và thông tin được truyền tải và lưu trữ chính xác.
- Từ các dữ liệu thu nhập được, tiến hành xây dựng cơ sở dữ liệu và phát triển ứng dụng.

5. Cấu trúc của đề án tốt nghiệp

- Chương 1: Tổng quan về hệ thống
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phân tích và thiết kế hệ thống
- Chương 4: Triển khai và đánh giá kết quả
- Kết luận

Chương 1: TỔNG QUAN VỀ HỆ THỐNG

1.1 Tên đề tài

Xây dựng ứng dụng mobile hỗ trợ quản lý tài chính theo mùa vụ cho nông dân.

1.2 Mục đích đề tài

Trong bối cảnh nông nghiệp Việt Nam đang phát triển mạnh mẽ và hướng tới hiện đại hóa, việc quản lý chi phí và tài chính một cách hiệu quả đóng vai trò quan trọng trong việc nâng cao năng suất và lợi nhuận cho người nông dân và các chủ trang trại. Tuy nhiên, hiện tại phần lớn các hoạt động quản lý tài chính nông nghiệp vẫn được thực hiện theo phương thức truyền thống, sử dụng sổ sách giấy tờ, dẫn đến nhiều hạn chế như dễ mất mát dữ liệu, khó khăn trong việc tra cứu thông tin, thiếu tính thống kê và phân tích, cũng như tốn nhiều thời gian và công sức. Nhận thức được những thách thức này, đề tài "Hệ thống Quản lý Chi phí Nông nghiệp" được nghiên cứu và phát triển nhằm mục đích xây dựng một giải pháp công nghệ toàn diện, hỗ trợ người dùng quản lý các hoạt động tài chính nông nghiệp một cách khoa học và hiệu quả. Hệ thống được thiết kế với mục tiêu chính là số hóa toàn bộ quy trình quản lý chi phí, từ việc theo dõi các mùa vụ, ghi nhận các giao dịch thu chi, quản lý kho hàng và vật liệu, đến việc tự động hóa quá trình nhập liệu thông qua AI, đồng thời cung cấp các công cụ báo cáo và thống kê trực quan để hỗ trợ việc ra quyết định kinh doanh. Bên cạnh đó, đề tài còn hướng tới việc ứng dụng các công nghệ hiện đại như phát triển ứng dụng di động đa nền tảng, xây dựng hệ thống backend RESTful API có khả năng mở rộng, và tích hợp trí tuệ nhân tạo để nâng cao trải nghiệm người dùng. Thông qua việc giải quyết các vấn đề thực tiễn trong quản lý tài chính nông nghiệp, đề tài không chỉ góp phần nâng cao hiệu quả quản lý cho người nông dân mà còn đóng góp vào quá trình chuyển đổi số trong lĩnh vực nông nghiệp Việt Nam, hướng tới một nền nông nghiệp thông minh và bền vững.

1.3 Mục tiêu cụ thể

Quản lý người dùng và xác thực :

- Người dùng có thể đăng ký tài khoản mới với thông tin username, email và mật khẩu, hệ thống sẽ tự động kiểm tra tính duy nhất của thông tin đăng ký.
- Người dùng có thể đăng nhập vào hệ thống bằng username và mật khẩu, sau khi xác thực thành công sẽ nhận được token JWT để truy cập các chức năng.
- Hệ thống hỗ trợ chức năng quên mật khẩu thông qua email, gửi mã OTP để xác minh và cho phép người dùng đặt lại mật khẩu mới một cách an toàn.
- Người dùng có thể xem và chỉnh sửa thông tin cá nhân trong hồ sơ của mình.

Quản lý mùa vụ:

- Người dùng có thể tạo mới các mùa vụ với thông tin tên mùa vụ, ngày bắt đầu và ngày kết thúc, người tạo sẽ tự động trở thành chủ sở hữu (owner) của mùa vụ.
- Chủ sở hữu có thể thêm thành viên vào mùa vụ với các vai trò khác nhau (VIEWER chỉ xem, EDITOR có thể tạo/chỉnh sửa giao dịch nhưng cần phê

duyet).

- Người dùng có thể xem danh sách các mùa vụ mà mình tham gia, bao gồm cả mùa vụ mình sở hữu và mùa vụ mình là thành viên.
- Hệ thống cung cấp thông kê chi tiết cho từng mùa vụ bao gồm tổng thu, tổng chi, lợi nhuận và biểu đồ thu chi theo tháng.

Quản lý giao dịch tài chính:

- Người dùng có thể tạo các giao dịch thu nhập hoặc chi phí với thông tin đầy đủ bao gồm loại giao dịch, danh mục, số tiền, mô tả, ngày thực hiện và hình ảnh minh chứng.
- Nếu người tạo là chủ sở hữu mùa vụ, giao dịch sẽ tự động được phê duyệt, nếu là thành viên EDITOR thì cần chủ sở hữu phê duyệt trước khi tính vào thống kê.
- Chủ sở hữu có thể xem danh sách tất cả giao dịch trong mùa vụ, lọc theo loại, danh mục, khoảng thời gian và trạng thái phê duyệt.
- Người dùng có thể chỉnh sửa hoặc xóa các giao dịch đã tạo (nếu chưa được phê duyệt hoặc là chủ sở hữu)

Tích hợp công nghệ AI để tự động hoá nhập liệu:

- Người dùng có thể chụp hoặc chọn hình ảnh hóa đơn/biên lai từ thư viện ảnh hoặc camera của thiết bị.
- Hệ thống sử dụng AI để tự động trích xuất thông tin từ hình ảnh bao gồm số tiền, ngày tháng, danh mục và mô tả.
- Người dùng có thể xem xét và chỉnh sửa thông tin được trích xuất trước khi tạo giao dịch, đảm bảo độ chính xác của dữ liệu.

Quản lý kho hàng & vật liệu:

- Người dùng có thể thêm các vật liệu vào kho hàng với thông tin tên, số lượng, đơn vị tính và ngưỡng cảnh báo khi số lượng thấp.
- Hệ thống tự động cảnh báo khi số lượng vật liệu xuống dưới ngưỡng đã thiết lập hoặc sắp hết hạn sử dụng.
- Người dùng có thể cập nhật số lượng vật liệu khi nhập hoặc xuất kho, hệ thống sẽ tự động ghi lại lịch sử thay đổi với thông tin người thực hiện, lý do và mùa vụ liên quan.
- Người dùng có thể xem lịch sử nhập/xuất kho của từng vật liệu để theo dõi việc sử dụng trong các mùa vụ.

Báo cáo và thống kê:

- Hệ thống cung cấp báo cáo tổng quan về tài chính bao gồm tổng thu, tổng chi, lợi nhuận của tất cả các mùa vụ với khả năng lọc theo khoảng thời gian.

- Người dùng có thể xem các biểu đồ trực quan về thu chi theo thời gian và phân bổ chi phí theo danh mục để phân tích xu hướng tài chính.
- Hệ thống hỗ trợ xuất báo cáo ra định dạng PDF và CSV để người dùng có thể lưu trữ, chia sẻ hoặc phân tích thêm bằng các công cụ khác.
- Báo cáo được cung cấp ở cả cấp độ mùa vụ (chi tiết từng mùa vụ) và cấp độ tổng quan (tổng hợp tất cả mùa vụ).

Chương 2: CƠ SỞ LÝ THUYẾT

2.1 Xây dựng giao diện

2.1.1 Dart

Dart là một ngôn ngữ lập trình đa mục đích mã nguồn mở, đã được đặt nền móng bởi Google. Đây là một khía cạnh của ngôn ngữ lập trình hướng đối tượng, đặc trưng bởi cú pháp kiểu C. Tinh thần lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp, mở ra khả năng sáng tạo không giới hạn khi đặt lên bàn cân với những ngôn ngữ khác. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn.



Hình 2.1 Logo Dart

2.1.2 Flutter

Flutter là một bộ công cụ giao diện người dùng (UI) mã nguồn mở của Google, cho phép phát triển ứng dụng đa nền tảng (iOS, Android, web, desktop) từ một codebase duy nhất, sử dụng ngôn ngữ lập trình Dart và có hiệu suất cao, gần giống ứng dụng gốc (native) nhờ biên dịch trực tiếp ra mã máy.

Các đặc điểm nổi bật:

- Đa nền tảng (Cross-platform): Viết một lần, chạy trên nhiều hệ điều hành (iOS, Android, Web, Windows, macOS, Linux).
- Hiệu suất cao (High Performance): Biên dịch trực tiếp ra mã máy, cho tốc độ và trải nghiệm mượt mà như ứng dụng native.
- Giao diện đẹp (Beautiful UI): Cung cấp các widget có sẵn phong phú, dễ dàng tùy chỉnh để tạo giao diện đẹp, bắt mắt.
- Phát triển nhanh (Fast Development): Tính năng Hot Reload giúp xem kết quả thay đổi ngay lập tức, tiết kiệm thời gian.
- Mã nguồn mở & Miễn phí: Hoàn toàn miễn phí và có cộng đồng hỗ trợ lớn.
- Sử dụng ngôn ngữ Dart: Ngôn ngữ lập trình hiện đại, dễ học, được Google phát triển.

Thành phần chính: SDK:

- Bộ công cụ phát triển, bao gồm trình biên dịch.
- Framework: Thư viện widget (nút, text, slider,...) dùng để xây dựng giao diện.
- Engine: Lõi của Flutter, viết bằng C++, chịu trách nhiệm render giao diện.



Hình 2.2 Logo Flutter

2.2 Xây dựng Back-end

2.2.1 NestJs

NestJS là một framework mã nguồn mở để phát triển ứng dụng server-side (backend applications) bằng ngôn ngữ TypeScript hoặc JavaScript. Nó được xây dựng trên cơ sở của Node.js và sử dụng các khái niệm từ TypeScript để tạo ra một môi trường phát triển hiện đại và mạnh mẽ cho việc xây dựng các ứng dụng web và API.

Mục tiêu chính của NestJS là cung cấp một cấu trúc ứng dụng rõ ràng và dễ quản lý, giúp tăng tính bảo trì và sự tổ chức trong mã nguồn. Để đạt được điều này, NestJS triển khai mô hình kiến trúc lõi (core architecture) dựa trên các nguyên tắc của Angular, đặc biệt là sử dụng Dependency Injection (DI) và Modules (Các module).



Hình 2.3 Logo NestJS

2.3 Cơ sở dữ liệu

2.3.1 MongoDB

MongoDB là một chương trình 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.

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.



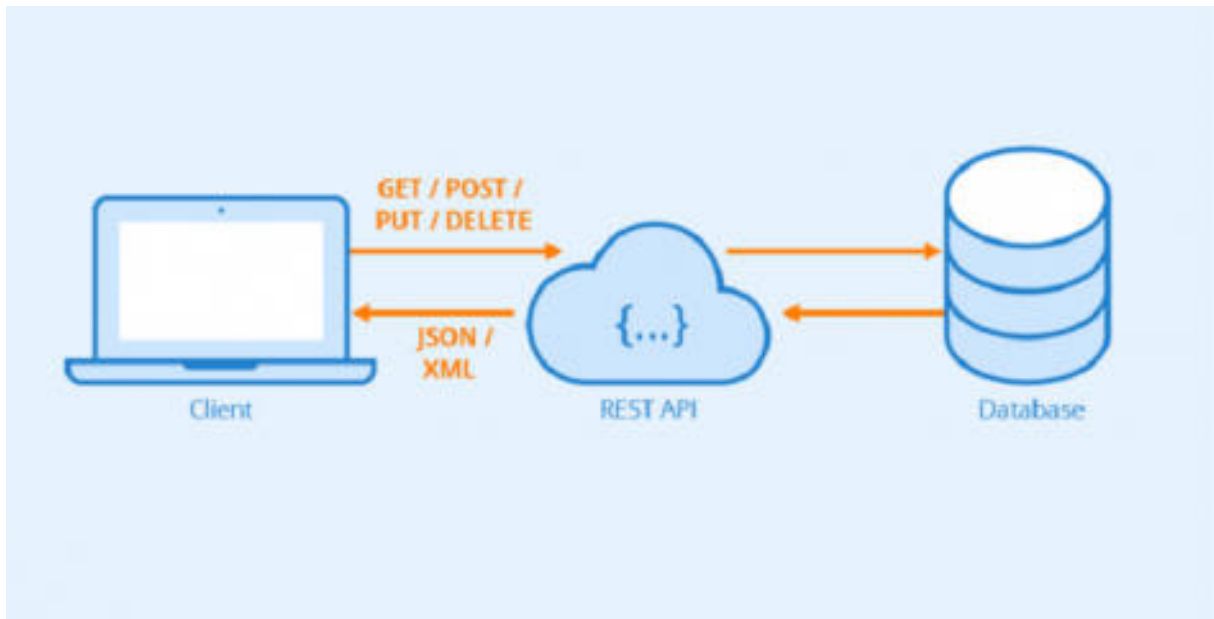
Hình 2.4 Logo MongoDB

2.4 Giao tiếp Client - Server

2.4.1 RESTful API

RESTful API (Representational State Transfer API) là một kiểu kiến trúc cho các API (Application Programming Interface) được sử dụng để truyền tải và trao đổi dữ liệu giữa các ứng dụng web. RESTful API sử dụng giao thức HTTP để truyền tải dữ liệu giữa máy chủ và máy khách, và sử dụng các phương thức HTTP như GET, POST, PUT và DELETE để thực hiện các thao tác trên tài nguyên.

RESTful API sử dụng các URL dễ đọc và dễ hiểu, và sử dụng các định dạng dữ liệu như JSON hoặc XML để trao đổi thông tin giữa máy chủ và máy khách. RESTful API cũng có tính khả di động cao, cho phép các ứng dụng khác nhau có thể truy cập và sử dụng các tài nguyên một cách dễ dàng.



Hình 2.5 RESTful API

2.5 Kết chương

Chương này cung cấp cái nhìn tổng quan về các lý thuyết được áp dụng để xây dựng ứng dụng:

- Đối với mobile: Sử dụng Flutter.
- Đối với back-end: Áp dụng framework như: NestJS.
- Giao tiếp giữa client-server chủ yếu sử dụng giao thức HTTP theo chuẩn RESTful API.

Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Phân tích yêu cầu hệ thống

Hệ thống tập trung chủ yếu vào việc quản lý chi phí và tài chính nông nghiệp, hỗ trợ người dùng theo dõi các mùa vụ, ghi nhận giao dịch thu chi, quản lý kho hàng và tạo báo cáo thống kê. Các đối tượng dùng chính trong hệ thống bao gồm 2 vai trò:

- Chủ sở hữu: quản lý toàn bộ mùa vụ, phê duyệt giao dịch và quản lý thành viên
- Thành viên: tham gia quản lý mùa vụ với các quyền hạn khác nhau (xem hoặc chỉnh sửa)

Hệ thống được thiết kế để hỗ trợ chủ trang trại quản lý hiệu quả các mùa vụ và tài chính, giúp thành viên theo dõi và ghi nhận các giao dịch, đồng thời tự động hóa quá trình nhập liệu thông qua AI và cung cấp các công cụ báo cáo để hỗ trợ ra quyết định kinh doanh.

Các chức năng chính được chia theo từng vai trò như sau:

- Chủ sở hữu:

- Quản lý mùa vụ: Tạo mới, chỉnh sửa, xóa mùa vụ và quản lý trạng thái mùa vụ (ACTIVE/COMPLETED).
- Quản lý thành viên: Thêm hoặc xóa thành viên vào mùa vụ, phân quyền cho thành viên (VIEWER hoặc EDITOR).
- Phê duyệt giao dịch: Xem xét và phê duyệt các giao dịch do thành viên EDITOR tạo ra trước khi tính vào thống kê.
- Quản lý kho hàng: Tạo, cập nhật và quản lý các vật liệu trong kho hàng, theo dõi lịch sử nhập/xuất kho.
- Xem báo cáo và thống kê: Xem thống kê chi tiết của từng mùa vụ và báo cáo tổng quan, xuất báo cáo ra PDF/CSV.

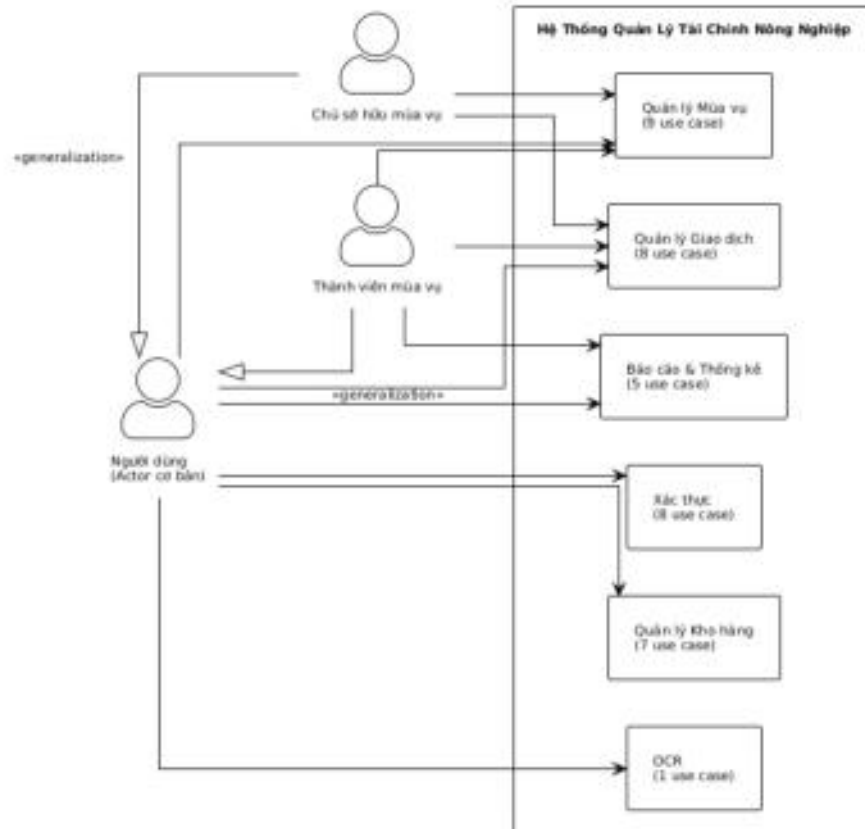
- Thành viên:

- Xem thông tin mùa vụ: Xem danh sách các mùa vụ mà mình tham gia, xem thông tin chi tiết và thống kê mùa vụ..
- Quản lý giao dịch (EDITOR): Tạo mới, chỉnh sửa các giao dịch thu nhập hoặc chi phí, upload hình ảnh minh chứng
- Sử dụng AI: Chụp hoặc chọn hình ảnh hóa đơn/biên lai để tự động trích xuất thông tin và tạo giao dịch.
- Quản lý kho hàng (nếu được phân quyền): Cập nhật số lượng vật liệu khi nhập/xuất kho, xem lịch sử kho hàng.
- Xem báo cáo: Xem báo cáo và thống kê của các mùa vụ mà mình tham gia.

3.2 Thiết kế hệ thống

3.2.1 Sơ đồ ca sử dụng

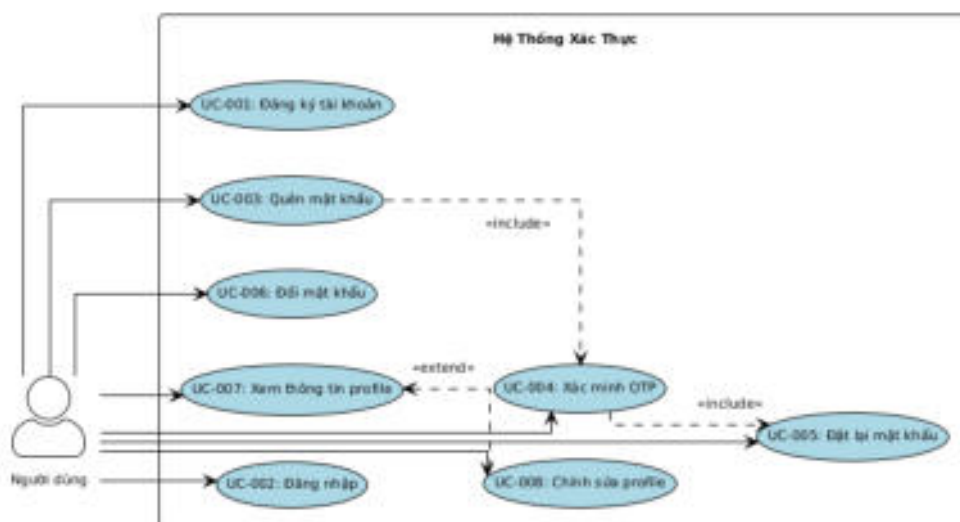
Sơ đồ sau đây cung cấp một bức tranh tổng thể về các nhóm chức năng, các tác nhân của hệ thống.



Hình 3.1 Sơ đồ Usecase tổng quát

3.2.2 Quản lý truy cập

Module Xác Thực và Quản Lý Người Dùng



Hình 3.2 Sơ đồ Usecase quản lý truy cập

Use case ID	UC - 001
Mô tả	Người dùng mới đăng ký tài khoản để sử dụng hệ thống
Tác nhân	Người dùng (chưa đăng nhập)
Tiền điều kiện	- Người dùng chưa có tài khoản trong hệ thống - Ứng dụng đang chạy và kết nối với back-end
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình đăng ký 2. Hệ thống hiển thị form đăng ký 3. Người dùng nhập thông tin: <ul style="list-style-type: none"> - Username (bắt buộc, duy nhất) - Email (bắt buộc, duy nhất, định dạng email hợp lệ) - Password (bắt buộc, tối thiểu 6 ký tự) - Full Name (bắt buộc) - Phone (tùy chọn) 4. Người dùng nhấn nút "Đăng ký" 5. Hệ thống validate dữ liệu đầu vào 6. Hệ thống kiểm tra username và email chưa tồn tại 7. Hệ thống mã hóa password bằng bcrypt 8. Hệ thống tạo user mới trong database 9. Hệ thống trả về thông báo thành công 10. Hệ thống tự động đăng nhập người dùng
Ngoại lệ	<ul style="list-style-type: none"> - Username đã tồn tại: Hệ thống hiển thị lỗi "Username đã được sử dụng" - Email đã tồn tại: Hệ thống hiển thị lỗi "Email đã được sử dụng" - Password không đủ mạnh: Hệ thống hiển thị lỗi "Mật khẩu phải có ít nhất 6 ký tự" - Email không hợp lệ: Hệ thống hiển thị lỗi "Email không hợp lệ"
Hậu điều kiện	<ul style="list-style-type: none"> - Tài khoản mới được tạo thành công - Người dùng đã đăng nhập vào hệ thống
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - Username và email phải duy nhất trong hệ thống - Password được mã hóa trước khi lưu vào database - Mặc định role là USER

Bảng 3.1 Đặc tả use case UC – 001

Use case ID	UC - 002
Mô tả	Người dùng đăng nhập vào hệ thống bằng username và password
Tác nhân	Người dùng
Tiền điều kiện	- Người dùng đã có tài khoản trong hệ thống - Ứng dụng đang chạy và kết nối với back-end
Luồng sự kiện chính	1. Người dùng mở màn hình đăng nhập 2. Hệ thống hiển thị form đăng nhập 3. Người dùng nhập username và password 4. Người dùng nhấn nút "Đăng nhập" 5. Hệ thống gửi request đến backend với username và password 6. Backend validate thông tin đăng nhập 7. Backend tạo JWT token 8. Hệ thống lưu token vào local storage 9. Hệ thống chuyển đến màn hình Home
Ngoại lệ	- Username không tồn tại: Hệ thống hiển thị lỗi "Thông tin đăng nhập không chính xác" - Password sai: Hệ thống hiển thị lỗi "Thông tin đăng nhập không chính xác" - Kết nối mạng lỗi: Hệ thống hiển thị lỗi "Không thể kết nối đến server"
Hậu điều kiện	- Người dùng đã đăng nhập thành công - JWT token được lưu và sử dụng cho các request tiếp theo
Quy tắc nghiệp vụ	- Token có thời hạn sử dụng - Token được gửi kèm trong header Authorization cho các API yêu cầu xác thực

Bảng 3.2 Đặc tả use case UC – 002

Use case ID	UC - 003
Mô tả	Người dùng yêu cầu reset mật khẩu khi quên mật khẩu
Tác nhân	Người dùng
Tiền điều kiện	- Người dùng đã có tài khoản trong hệ thống - Người dùng biết email đã đăng ký
Luồng sự kiện chính	1. Người dùng mở màn hình "Quên mật khẩu"

	<ol style="list-style-type: none"> 2. Hệ thống hiển thị form nhập email 3. Người dùng nhập email 4. Người dùng nhấn nút "Gửi OTP" 5. Hệ thống kiểm tra email có tồn tại trong hệ thống 6. Hệ thống tạo mã OTP ngẫu nhiên (6 chữ số) 7. Hệ thống lưu OTP và thời gian hết hạn (15 phút) vào database 8. Hệ thống gửi email chứa OTP đến địa chỉ email của người dùng 9. Hệ thống chuyển đến màn hình nhập OTP
Ngoại lệ	<ul style="list-style-type: none"> - Email không tồn tại: Hệ thống hiển thị lỗi "Email không tồn tại trong hệ thống" - Gửi email thất bại: Hệ thống hiển thị lỗi "Không thể gửi email, vui lòng thử lại"
Hậu điều kiện	<ul style="list-style-type: none"> - OTP đã được gửi đến email của người dùng - Người dùng ở màn hình nhập OTP
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - OTP có thời hạn 15 phút - OTP chỉ có thể sử dụng một lần - Sau khi hết hạn, người dùng phải yêu cầu OTP mới

Bảng 3.3 Đặc tả use case UC – 003

Use case ID	UC - 004
Mô tả	Người dùng nhập mã OTP để xác minh và tiếp tục quá trình reset mật khẩu.
Tác nhân	Người dùng
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã yêu cầu reset mật khẩu (UC-003) - OTP đã được gửi đến email
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng ở màn hình nhập OTP 2. Người dùng nhập mã OTP (6 chữ số) 3. Người dùng nhấn nút "Xác minh" 4. Hệ thống kiểm tra OTP có hợp lệ và chưa hết hạn 5. Hệ thống tạo reset token 6. Hệ thống lưu reset token vào database 7. Hệ thống chuyển đến màn hình đặt mật khẩu mới
Ngoại lệ	<ul style="list-style-type: none"> - OTP sai: Hệ thống hiển thị lỗi "Mã OTP không chính xác" - OTP đã hết hạn: Hệ thống hiển thị lỗi "Mã OTP đã hết hạn, vui lòng yêu cầu mã mới"

	- OTP đã được sử dụng: Hệ thống hiển thị lỗi "Mã OTP đã được sử dụng"
Hậu điều kiện	- OTP đã được gửi đến email của người dùng - Người dùng ở màn hình nhập OTP
Quy tắc nghiệp vụ	- OTP đã được xác minh thành công - Reset token đã được tạo - Người dùng ở màn hình đặt mật khẩu mới

Bảng 3.4 Đặc tả use case UC – 004

Use case ID	UC - 005
Mô tả	Người dùng đặt mật khẩu mới sau khi xác minh OTP thành công.
Tác nhân	Người dùng
Tiền điều kiện	- Người dùng đã xác minh OTP thành công (UC-004) - Reset token đã được tạo
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng ở màn hình đặt mật khẩu mới 2. Người dùng nhập mật khẩu mới và xác nhận mật khẩu 3. Người dùng nhấn nút "Đặt lại mật khẩu" 4. Hệ thống validate mật khẩu mới (tối thiểu 6 ký tự) 5. Hệ thống kiểm tra mật khẩu và xác nhận mật khẩu khớp nhau 6. Hệ thống kiểm tra reset token có hợp lệ và chưa hết hạn 7. Hệ thống mã hóa mật khẩu mới bằng bcrypt 8. Hệ thống cập nhật mật khẩu mới vào database 9. Hệ thống xóa reset token 10. Hệ thống hiển thị thông báo thành công 11. Hệ thống chuyển đến màn hình đăng nhập
Ngoại lệ	<ul style="list-style-type: none"> - Mật khẩu không đủ mạnh: Hệ thống hiển thị lỗi "Mật khẩu phải có ít nhất 6 ký tự" - Mật khẩu và xác nhận không khớp: Hệ thống hiển thị lỗi "Mật khẩu xác nhận không khớp" - Reset token không hợp lệ hoặc hết hạn: Hệ thống hiển thị lỗi "Token không hợp lệ hoặc đã hết hạn"
Hậu điều kiện	- Mật khẩu mới đã được cập nhật - Người dùng có thể đăng nhập bằng mật khẩu mới

Bảng 3.5 Đặc tả use case UC – 005

Use case ID	UC - 006
Mô tả	Người dùng đã đăng nhập đổi mật khẩu của mình.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Đổi mật khẩu" từ profile 2. Hệ thống hiển thị form đổi mật khẩu 3. Người dùng nhập: - Mật khẩu cũ - Mật khẩu mới - Xác nhận mật khẩu mới 4. Người dùng nhấn nút "Đổi mật khẩu" 5. Hệ thống validate dữ liệu đầu vào 6. Hệ thống kiểm tra mật khẩu cũ có chính xác 7. Hệ thống kiểm tra mật khẩu mới và xác nhận khớp nhau 8. Hệ thống mã hóa mật khẩu mới 9. Hệ thống cập nhật mật khẩu mới vào database 10. Hệ thống hiển thị thông báo thành công
Ngoại lệ	<ul style="list-style-type: none"> - Mật khẩu cũ sai: Hệ thống hiển thị lỗi "Mật khẩu cũ không chính xác" - Mật khẩu mới và xác nhận không khớp: Hệ thống hiển thị lỗi "Mật khẩu xác nhận không khớp" - Mật khẩu mới không đủ mạnh: Hệ thống hiển thị lỗi "Mật khẩu phải có ít nhất 6 ký tự"
Hậu điều kiện	- Mật khẩu đã được đổi thành công - Người dùng phải đăng nhập lại bằng mật khẩu mới

Bảng 3.6 Đặc tả use case UC – 006

Use case ID	UC - 007
Mô tả	Người dùng xem thông tin profile của mình
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình Profile 2. Hệ thống gửi request đến API /auth/profile với JWT token

	<p>3. Backend xác thực token và lấy userId</p> <p>4. Backend lấy thông tin user từ database</p> <p>5. Backend trả về thông tin user (không bao gồm password)</p> <p>6. Hệ thống hiển thị thông tin:</p> <ul style="list-style-type: none"> - Username - Email - Full Name - Phone
Ngoại lệ	<ul style="list-style-type: none"> - Token không hợp lệ: Hệ thống yêu cầu đăng nhập lại - User không tồn tại: Hệ thống hiển thị lỗi "Không tìm thấy thông tin người dùng"
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin profile được hiển thị

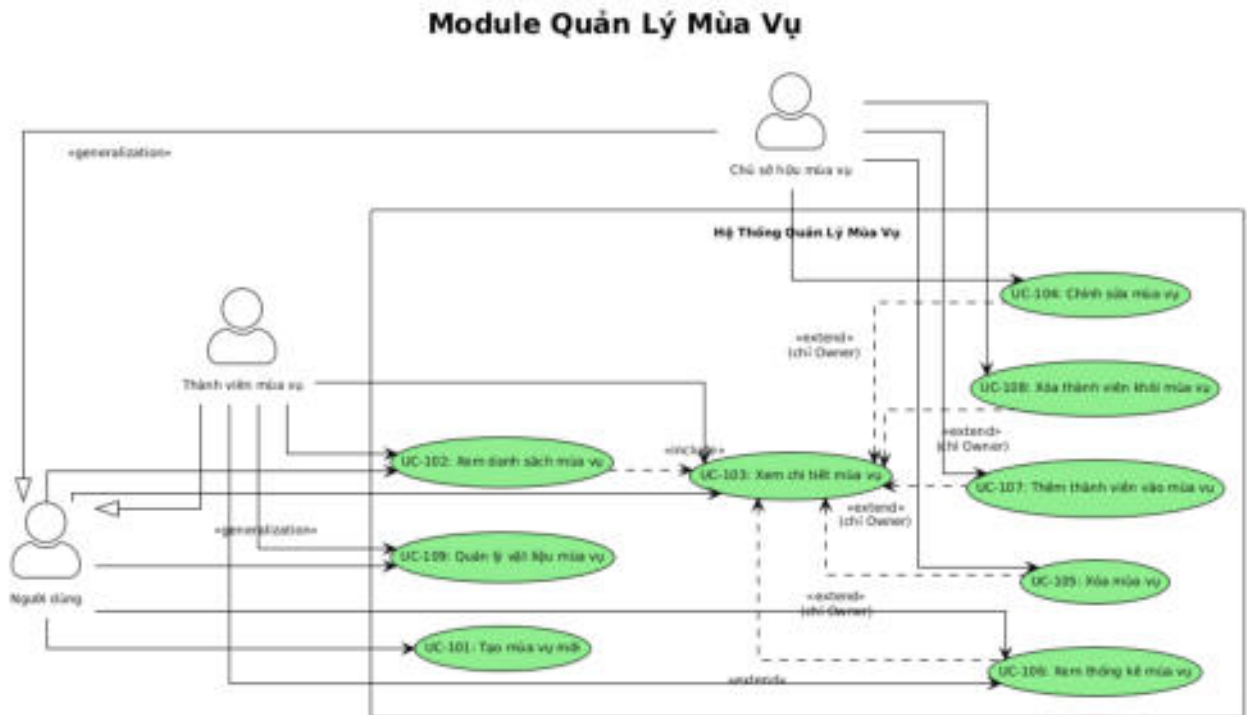
Bảng 3.7 Đặc tả use case UC – 007

Use case ID	UC - 008
Mô tả	Người dùng chỉnh sửa thông tin profile của mình.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Chỉnh sửa Profile" 2. Hệ thống hiển thị form với thông tin hiện tại 3. Người dùng chỉnh sửa các trường: <ul style="list-style-type: none"> - Full Name - Phone - Email 4. Người dùng nhấn nút "Lưu" 5. Hệ thống validate dữ liệu 6. Hệ thống gửi request cập nhật đến API 7. Backend cập nhật thông tin vào database 8. Hệ thống hiển thị thông báo thành công 9. Hệ thống cập nhật thông tin trên màn hình Profile
Ngoại lệ	<ul style="list-style-type: none"> - Email không hợp lệ: Hệ thống hiển thị lỗi "Email không hợp lệ" - Email đã được sử dụng bởi user khác: Hệ thống hiển thị lỗi "Email đã được sử dụng"

Hậu điều kiện	- Thông tin profile đã được cập nhật
Quy tắc nghiệp vụ	- Username không thể thay đổi - Password phải thay đổi thông qua UC-006

Bảng 3.8 Đặc tả use case UC – 008

3.2.3 Quản lý mùa vụ



Hình 3.3 Sơ đồ Usecase quản lý mùa vụ

Use case ID	UC - 101
Mô tả	Người dùng tạo một mùa vụ mới để quản lý các hoạt động nông nghiệp trong một khoảng thời gian.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Tạo mùa vụ" 2. Hệ thống hiển thị form tạo mùa vụ 3. Người dùng nhập thông tin: <ul style="list-style-type: none"> - Tên mùa vụ (bắt buộc) - Ngày bắt đầu (bắt buộc) - Ngày kết thúc (tùy chọn) 4. Người dùng nhấn nút "Tạo" 5. Hệ thống validate dữ liệu 6. Hệ thống gửi request đến API /seasons với JWT token 7. Backend tạo season mới với: <ul style="list-style-type: none"> - ownerId = userId hiện tại - status = ACTIVE - members = [] (rỗng ban đầu) 8. Backend lưu vào database 9. Hệ thống hiển thị thông báo thành công 10. Hệ thống chuyển đến màn hình chi tiết mùa vụ
Ngoại lệ	- Tên mùa vụ trống: Hệ thống hiển thị lỗi "Tên mùa vụ không được để trống" - Ngày kết thúc trước ngày bắt đầu: Hệ thống hiển thị lỗi "Ngày kết thúc phải sau ngày bắt đầu"
Hậu điều kiện	- Mùa vụ mới đã được tạo - Người dùng là owner của mùa vụ này
Quy tắc nghiệp vụ	- Mỗi mùa vụ có một owner duy nhất - Owner có quyền cao nhất đối với mùa vụ

Bảng 3.9 Đặc tả use case UC – 101

Use case ID	UC - 102
Mô tả	Người dùng xem danh sách tất cả các mùa vụ mà họ là owner hoặc member.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Danh sách mùa vụ" 2. Hệ thống gửi request đến API /seasons với JWT token 3. Backend lấy tất cả seasons mà user là owner hoặc member 4. Backend trả về danh sách seasons 5. Hệ thống hiển thị danh sách với thông tin: <ul style="list-style-type: none"> - Tên mùa vụ - Ngày bắt đầu - Ngày kết thúc - Trạng thái (ACTIVE/COMPLETED) - Vai trò của user (Owner/Member)
Ngoại lệ	- Không có mùa vụ nào: Hệ thống hiển thị thông báo "Chưa có mùa vụ nào"
Hậu điều kiện	- Danh sách mùa vụ được hiển thị

Bảng 3.10 Đặc tả use case UC – 102

Use case ID	UC - 103
Mô tả	Người dùng xem thông tin chi tiết của một mùa vụ.
Tác nhân	Người dùng (đã đăng nhập, là owner hoặc member)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn một mùa vụ từ danh sách 2. Hệ thống gửi request đến API /seasons/:id với JWT token 3. Backend kiểm tra user có quyền truy cập mùa vụ 4. Backend lấy thông tin chi tiết mùa vụ 5. Backend trả về thông tin: <ul style="list-style-type: none"> - Thông tin cơ bản (tên, ngày bắt đầu, ngày kết thúc, trạng thái)

	<ul style="list-style-type: none"> - Danh sách thành viên - Thống kê (tổng thu, tổng chi, lợi nhuận) <p>6. Hệ thống hiển thị thông tin chi tiết</p>
Ngoại lệ	<ul style="list-style-type: none"> - User không có quyền truy cập: Hệ thống hiển thị lỗi "Bạn không có quyền truy cập mùa vụ này" - Mùa vụ không tồn tại: Hệ thống hiển thị lỗi "Mùa vụ không tồn tại"
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin chi tiết mùa vụ được hiển thị

Bảng 3.11 Đặc tả use case UC – 103

Use case ID	UC - 104
Mô tả	Chủ sở hữu chỉnh sửa thông tin của mùa vụ.
Tác nhân	Chủ sở hữu của mùa vụ
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu của mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Chỉnh sửa mùa vụ" từ chi tiết mùa vụ 2. Hệ thống hiển thị form với thông tin hiện tại 3. Người dùng chỉnh sửa: <ul style="list-style-type: none"> - Tên mùa vụ - Ngày bắt đầu - Ngày kết thúc - Trạng thái 4. Người dùng nhấn nút "Lưu" 5. Hệ thống validate dữ liệu 6. Hệ thống gửi request đến API /seasons/:id với JWT token 7. Backend kiểm tra user là owner 8. Backend cập nhật thông tin vào database 9. Hệ thống hiển thị thông báo thành công 10. Hệ thống cập nhật thông tin trên màn hình chi tiết
Ngoại lệ	<ul style="list-style-type: none"> - Người dùng không phải chủ sở hữu: Hệ thống hiển thị lỗi "Chỉ chủ sở hữu mới có quyền chỉnh sửa"
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin mùa vụ đã được cập nhật
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - Chỉ chủ sở hữu mới có quyền chỉnh sửa mùa vụ - Không thể thay đổi ownerId

Bảng 3.12 Đặc tả use case UC – 104

Use case ID	UC - 105
Mô tả	Chủ sở hữu xóa một mùa vụ khỏi hệ thống.
Tác nhân	Chủ sở hữu của mùa vụ
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu của mùa vụ
Luồng sự kiện chính	1. Người dùng mở màn hình chi tiết mùa vụ 2. Người dùng nhấn nút "Xóa mùa vụ" 3. Hệ thống hiển thị dialog xác nhận 4. Người dùng xác nhận xóa 5. Hệ thống gửi request DELETE đến API /seasons/:id với JWT token 6. Backend kiểm tra user là owner 7. Backend xóa mùa vụ và tất cả dữ liệu liên quan (transactions, inventory logs) 8. Hệ thống hiển thị thông báo thành công 9. Hệ thống chuyển về màn hình danh sách mùa vụ
Ngoại lệ	- Người dùng hủy: Hệ thống đóng dialog, không xóa - User không phải owner: Hệ thống hiển thị lỗi "Chỉ owner mới có quyền xóa"
Hậu điều kiện	- Mùa vụ và tất cả dữ liệu liên quan đã được xóa
Quy tắc nghiệp vụ	- Xóa mùa vụ sẽ xóa tất cả ghi chú giao dịch và vật tư liên quan - Hành động này không thể hoàn tác

Bảng 3.13 Đặc tả use case UC – 105

Use case ID	UC - 106
Mô tả	Người dùng xem các thống kê tài chính của mùa vụ.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu hoặc thành viên)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu hoặc thành viên của mùa vụ
Luồng sự kiện chính	1. Người dùng mở màn hình chi tiết mùa vụ 2. Người dùng chọn tab "Thống kê" 3. Hệ thống gửi request đến API /seasons/:id/stats với JWT token

	<p>4. Backend tính toán thống kê:</p> <ul style="list-style-type: none"> - Tổng thu nhập - Tổng chi phí - Lợi nhuận (Thu - Chi) - Số lượng giao dịch - Biểu đồ theo thời gian <p>5. Backend trả về dữ liệu thống kê</p> <p>6. Hệ thống hiển thị:</p> <ul style="list-style-type: none"> - Các số liệu tổng quan - Biểu đồ
Hậu điều kiện	- Thống kê mùa vụ được hiển thị

Bảng 3.14 Đặc tả use case UC – 106

Use case ID	UC - 107
Mô tả	Chủ sở hữu thêm một thành viên mới vào mùa vụ với vai trò xem hoặc chỉnh sửa.
Tác nhân	Chủ sở hữu của mùa vụ
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu của mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Quản lý thành viên" từ chi tiết mùa vụ 2. Người dùng nhấn nút "Thêm thành viên" 3. Hệ thống hiển thị form tìm kiếm user 4. Người dùng tìm kiếm user bằng: <ul style="list-style-type: none"> - Tên - Số điện thoại - Email 5. Hệ thống gửi request đến API /users/search 6. Backend trả về danh sách user khớp 7. Người dùng chọn một user 8. Người dùng chọn vai trò (xem hoặc chỉnh sửa) 9. Người dùng nhấn nút "Thêm" 10. Hệ thống gửi request đến API /seasons/:id/members với JWT token 11. Backend kiểm tra user là owner

	<p>12. Backend kiểm tra user chưa là thành viên</p> <p>13. Backend thêm user vào danh sách members</p> <p>14. Backend cập nhật vào database</p> <p>15. Hệ thống hiển thị thông báo thành công và cập nhật danh sách</p>
Ngoại lệ	<p>- Không tìm thấy user: Hệ thống hiển thị "Không tìm thấy user"</p> <p>- User đã là thành viên: Hệ thống hiển thị lỗi "User đã là thành viên của mùa vụ này"</p> <p>- User không phải owner: Hệ thống hiển thị lỗi "Chỉ owner mới có quyền thêm thành viên"</p>
Hậu điều kiện	- Thành viên mới đã được thêm vào mùa vụ
Quy tắc nghiệp vụ	<p>- Người xem: Chỉ xem, không thể chỉnh sửa</p> <p>- Người chỉnh sửa: Có thể tạo và chỉnh sửa giao dịch, nhưng không thể phê duyệt</p>

Bảng 3.15 Đặc tả use case UC – 107

Use case ID	UC - 108
Mô tả	Chủ sở hữu xóa một thành viên khỏi mùa vụ.
Tác nhân	Chủ sở hữu của mùa vụ
Tiền điều kiện	<p>- Người dùng đã đăng nhập vào hệ thống</p> <p>- Người dùng là owner của mùa vụ</p> <p>- Mùa vụ có ít nhất một thành viên</p>
Luồng sự kiện chính	<p>1. Người dùng mở màn hình "Quản lý thành viên"</p> <p>2. Người dùng chọn một thành viên</p> <p>3. Người dùng nhấn nút "Xóa"</p> <p>4. Hệ thống hiển thị dialog xác nhận</p> <p>5. Người dùng xác nhận xóa</p> <p>6. Hệ thống gửi request DELETE đến API /seasons/:id/members/:userId với JWT token</p> <p>7. Backend kiểm tra user là owner</p> <p>8. Backend xóa thành viên khỏi danh sách members</p> <p>9. Backend cập nhật vào database</p> <p>10. Hệ thống hiển thị thông báo thành công</p> <p>11. Hệ thống cập nhật danh sách thành viên</p>
Ngoại lệ	<p>- Người dùng hủy: Hệ thống đóng dialog, không xóa</p> <p>- User không phải chủ sở hữu: Hệ thống hiển thị lỗi "Chỉ chủ sở hữu"</p>

	mới có quyền xóa thành viên"
Hậu điều kiện	- Thành viên đã được xóa khỏi mùa vụ - Thành viên không còn quyền truy cập mùa vụ

Bảng 3.16 Đặc tả use case UC – 108

Use case ID	UC - 109
Mô tả	Người dùng xem và quản lý các vật liệu được sử dụng trong mùa vụ.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu hoặc thành viên)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu hoặc thành viên của mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết mùa vụ 2. Người dùng chọn tab "Vật liệu" 3. Hệ thống gửi request đến API /inventory/season/:seasonId/logs với JWT token 4. Backend lấy tất cả inventory logs liên quan đến mùa vụ 5. Backend trả về danh sách logs 6. Hệ thống hiển thị danh sách vật liệu đã sử dụng: <ul style="list-style-type: none"> - Tên vật liệu - Số lượng - Đơn vị - Ngày sử dụng - Người sử dụng
Hậu điều kiện	- Danh sách vật liệu mùa vụ được hiển thị

Bảng 3.17 Đặc tả use case UC – 109

	<p>7. Hệ thống upload hình ảnh (nếu có) lên server</p> <p>8. Hệ thống gửi request đến API /transactions với JWT token</p> <p>9. Backend tạo transaction mới với: - creatorId = userId hiện tại - isVerified = false (chưa được phê duyệt)</p> <p>10. Backend lưu vào database</p> <p>11. Hệ thống hiển thị thông báo thành công</p> <p>12. Hệ thống chuyển đến màn hình chi tiết giao dịch</p>
Ngoại lệ	<ul style="list-style-type: none"> - Số tiền ≤ 0: Hệ thống hiển thị lỗi "Số tiền phải lớn hơn 0" - Danh mục trống: Hệ thống hiển thị lỗi "Danh mục không được để trống" - Upload hình ảnh thất bại: Hệ thống hiển thị cảnh báo nhưng vẫn cho phép tạo giao dịch
Hậu điều kiện	<ul style="list-style-type: none"> - Giao dịch mới đã được tạo - Giao dịch ở trạng thái chưa được phê duyệt (nếu người dùng là người chỉnh sửa)
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - Người chỉnh sửa tạo giao dịch cần được người chủ sở hữu phê duyệt - Người chỉnh sửa tạo giao dịch có thể tự phê duyệt hoặc để chưa phê duyệt - Hình ảnh được lưu trong thư mục uploads

Bảng 3.18 Đặc tả use case UC – 201

Use case ID	UC - 202
Mô tả	Người dùng xem danh sách các giao dịch.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Giao dịch" 2. Hệ thống gửi request đến API /transactions với JWT token 3. Backend lấy tất cả transactions của user 4. Backend trả về danh sách transactions 5. Hệ thống hiển thị danh sách với thông tin: <ul style="list-style-type: none"> - Loại giao dịch (Thu/Chi) - Danh mục - Số tiền

	<ul style="list-style-type: none"> - Ngày - Trạng thái phê duyệt - Mùa vụ
Ngoại lệ	- Không có giao dịch nào: Hệ thống hiển thị "Chưa có giao dịch nào"
Hậu điều kiện	- Danh sách giao dịch được hiển thị

Bảng 3.19 Đặc tả use case UC – 202

Use case ID	UC - 203
Mô tả	Người dùng xem thông tin chi tiết của một giao dịch.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Giao dịch tồn tại
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn một giao dịch từ danh sách 2. Hệ thống gửi request đến API /transactions/:id với JWT token 3. Backend lấy thông tin chi tiết transaction 4. Backend trả về thông tin: <ul style="list-style-type: none"> - Tất cả thông tin cơ bản - Hình ảnh (nếu có) - Người tạo - Người phê duyệt (nếu đã phê duyệt) - Ngày phê duyệt 5. Hệ thống hiển thị thông tin chi tiết
Ngoại lệ	- Giao dịch không tồn tại: Hệ thống hiển thị lỗi "Giao dịch không tồn tại"
Hậu điều kiện	- Thông tin chi tiết giao dịch được hiển thị

Bảng 3.20 Đặc tả use case UC – 203

Use case ID	UC - 204
Mô tả	Người dùng chỉnh sửa thông tin của một giao dịch.
Tác nhân	Người dùng (đã đăng nhập, là người tạo giao dịch hoặc là chủ sở hữu của mùa vụ)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là người tạo của giao dịch hoặc chủ sở hữu của của

	mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết giao dịch 2. Người dùng nhấn nút "Chỉnh sửa" 3. Hệ thống hiển thị form với thông tin hiện tại 4. Người dùng chỉnh sửa các trường (trừ creatorId và seasonId) 5. Người dùng nhấn nút "Luu" 6. Hệ thống validate dữ liệu 7. Hệ thống gửi request PATCH đến API /transactions/:id với JWT token 8. Backend kiểm tra quyền chỉnh sửa 9. Backend cập nhật thông tin vào database 10. Hệ thống hiển thị thông báo thành công 11. Hệ thống cập nhật thông tin trên màn hình chi tiết
Ngoại lệ	- User không có quyền chỉnh sửa: Hệ thống hiển thị lỗi "Bạn không có quyền chỉnh sửa giao dịch này"
Hậu điều kiện	- Thông tin giao dịch đã được cập nhật
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - Creator hoặc owner của mùa vụ mới có quyền chỉnh sửa - Không thể thay đổi creatorId và seasonId

Bảng 3.21 Đặc tả use case UC – 204

Use case ID	UC - 205
Mô tả	Người dùng xóa một giao dịch.
Tác nhân	User (đã đăng nhập, là người tạo giao dịch hoặc chủ sở hữu của mùa vụ)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là người tạo của giao dịch hoặc chủ sở hữu của mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết giao dịch 2. Người dùng nhấn nút "Xóa" 3. Hệ thống hiển thị dialog xác nhận 4. Người dùng xác nhận xóa 5. Hệ thống gửi request DELETE đến API /transactions/:id với JWT token 6. Backend kiểm tra quyền xóa 7. Backend xóa transaction khỏi database 8. Hệ thống hiển thị thông báo thành công

	9. Hệ thống chuyển về màn hình danh sách giao dịch
Ngoại lệ	- Người dùng hủy: Hệ thống đóng dialog, không xóa - User không có quyền xóa: Hệ thống hiển thị lỗi "Bạn không có quyền xóa giao dịch này"
Hậu điều kiện	- Giao dịch đã được xóa

Bảng 3.22 Đặc tả use case UC – 205

Use case ID	UC - 206
Mô tả	Người dùng lọc danh sách giao dịch theo các tiêu chí khác nhau.
Tác nhân	User (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Có ít nhất một giao dịch
Luồng sự kiện chính	1. Người dùng mở màn hình "Giao dịch" 2. Người dùng nhấn nút "Lọc" 3. Hệ thống hiển thị form lọc với các tùy chọn: - Mùa vụ (seasonId) - Khoảng thời gian (startDate, endDate) - Loại giao dịch (INCOME/EXPENSE) - Danh mục (category) 4. Người dùng chọn các tiêu chí lọc 5. Người dùng nhấn nút "Áp dụng" 6. Hệ thống gửi request đến API /transactions với query parameters 7. Backend lọc transactions theo các tiêu chí 8. Backend trả về danh sách đã lọc 9. Hệ thống hiển thị danh sách giao dịch đã lọc
Ngoại lệ	- Người dùng nhấn "Xóa bộ lọc": Hệ thống hiển thị tất cả giao dịch
Hậu điều kiện	- Danh sách giao dịch đã được lọc theo tiêu chí

Bảng 3.23 Đặc tả use case UC – 206

Use case ID	UC - 207
Mô tả	Chủ sở hữu phê duyệt một giao dịch do người chỉnh sửa tạo.
Tác nhân	Chủ sở hữu mùa vụ
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống

	<ul style="list-style-type: none"> - Người dùng là chủ sở hữu của mùa vụ - Giao dịch chưa được phê duyệt
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết giao dịch 2. Người dùng nhấn nút "Phê duyệt" 3. Hệ thống hiển thị dialog xác nhận 4. Người dùng xác nhận phê duyệt 5. Hệ thống gửi request PATCH đến API /transactions/:id/verify với JWT token 6. Backend kiểm tra user là owner của mùa vụ 7. Backend cập nhật: - isVerified = true - verifiedBy = userId hiện tại 8. Backend cập nhật vào database 9. Hệ thống hiển thị thông báo thành công 10. Hệ thống cập nhật trạng thái trên màn hình chi tiết
Ngoại lệ	<ul style="list-style-type: none"> - Người dùng hủy: Hệ thống đóng dialog, không phê duyệt - User không phải chủ sở hữu: Hệ thống hiển thị lỗi "Chỉ owner mới có quyền phê duyệt giao dịch"
Hậu điều kiện	<ul style="list-style-type: none"> - Giao dịch đã được phê duyệt - Trạng thái isVerified = true
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - Chỉ chủ sở hữu của mùa vụ mới có quyền phê duyệt - Một giao dịch chỉ có thể được phê duyệt một lần

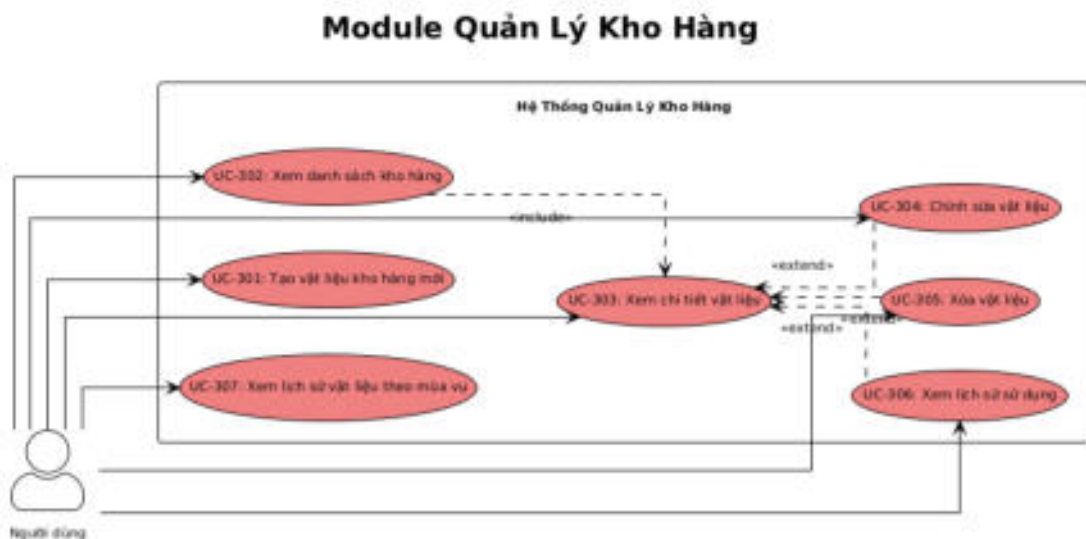
Bảng 3.24 Đặc tả use case UC – 207

Use case ID	UC - 208
Mô tả	Người dùng sử dụng AI để tự động nhập thông tin giao dịch từ hình ảnh hóa đơn/biên lai.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Ứng dụng có quyền truy cập camera/thư viện ảnh
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Tạo giao dịch" 2. Người dùng nhấn nút "Quét AI" 3. Hệ thống mở camera hoặc thư viện ảnh 4. Người dùng chụp hoặc chọn hình ảnh hóa đơn/biên lai 5. Hệ thống hiển thị hình ảnh đã chọn 6. Người dùng nhấn nút "Xử lý AI" 7. Hệ thống upload hình ảnh lên server

	<p>8. Hệ thống gửi request đến API /ocr/upload</p> <p>9. Backend xử lý hình ảnh bằng AI</p> <p>10. Backend trích xuất thông tin:</p> <ul style="list-style-type: none"> - Số tiền - Ngày - Danh mục (nếu có thể nhận diện) - Mô tả <p>11. Backend trả về dữ liệu đã trích xuất</p> <p>12. Hệ thống tự động điền vào form tạo giao dịch</p> <p>13. Người dùng kiểm tra và chỉnh sửa thông tin (nếu cần)</p> <p>14. Người dùng nhấn nút "Tạo" để hoàn tất</p>
Ngoại lệ	<ul style="list-style-type: none"> - Người dùng hủy: Hệ thống quay lại màn hình tạo giao dịch - AI không thể đọc được: Hệ thống hiển thị cảnh báo "Không thể đọc thông tin từ hình ảnh, vui lòng nhập thủ công" - AI đọc sai: Người dùng chỉnh sửa thông tin trước khi tạo
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin giao dịch đã được điền tự động từ hình ảnh - Người dùng có thể tạo giao dịch với thông tin đã điền
Quy tắc nghiệp vụ	<ul style="list-style-type: none"> - AI không phải lúc nào cũng chính xác 100%, người dùng nên kiểm tra lại - Hình ảnh được lưu tạm thời và xóa sau khi xử lý

Bảng 3.25 Đặc tả use case UC – 208

3.2.4 Quản lý kho hàng



Hình 3.5 Sơ đồ Usecase quản lý kho hàng

Use case ID	UC - 301
Mô tả	Người dùng thêm một vật liệu mới vào kho hàng.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Kho hàng" 2. Người dùng nhấn nút "Thêm vật liệu" 3. Hệ thống hiển thị form tạo vật liệu 4. Người dùng nhập thông tin: <ul style="list-style-type: none"> - Tên vật liệu (bắt buộc) - Số lượng (bắt buộc) - Đơn vị (bắt buộc: kg, lít, bao, thùng, ...) - Ngưỡng cảnh báo tối thiểu - Hình ảnh (tùy chọn) 5. Người dùng nhấn nút "Tạo" 6. Hệ thống validate dữ liệu 7. Hệ thống gửi request đến API /inventory với JWT token 9. Backend tạo inventory item mới với ownerId = userId hiện tại 10. Backend lưu vào database 11. Hệ thống hiển thị thông báo thành công 12. Hệ thống chuyển đến màn hình chi tiết vật liệu
Ngoại lệ	- Số lượng < 0: Hệ thống hiển thị lỗi "Số lượng phải lớn hơn hoặc bằng 0" - Tên vật liệu trống: Hệ thống hiển thị lỗi "Tên vật liệu không được để trống"
Hậu điều kiện	- Vật liệu mới đã được thêm vào kho hàng

Bảng 3.26 Đặc tả use case UC – 301

Use case ID	UC - 302
Mô tả	Người dùng xem danh sách tất cả vật liệu trong kho hàng của mình.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - JWT token hợp lệ

Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Kho hàng" 2. Hệ thống gửi request đến API /inventory với JWT token 3. Backend lấy tất cả inventory items của user 4. Backend trả về danh sách inventory items 5. Hệ thống hiển thị danh sách với thông tin: <ul style="list-style-type: none"> - Tên vật liệu - Số lượng và đơn vị - Trạng thái cảnh báo (nếu số lượng < ngưỡng cảnh báo)
Ngoại lệ	- Không có vật liệu nào: Hệ thống hiển thị "Kho hàng trống"
Hậu điều kiện	- Danh sách kho hàng được hiển thị

Bảng 3.27 Đặc tả use case UC – 302

Use case ID	UC - 303
Mô tả	Người dùng xem thông tin chi tiết của một vật liệu.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Vật liệu tồn tại
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn một vật liệu từ danh sách 2. Hệ thống gửi request đến API /inventory/:id với JWT token 3. Backend lấy thông tin chi tiết inventory item 4. Backend trả về thông tin: <ul style="list-style-type: none"> - Tất cả thông tin cơ bản - Lịch sử sử dụng (logs) 5. Hệ thống hiển thị thông tin chi tiết
Ngoại lệ	- Vật liệu không tồn tại: Hệ thống hiển thị lỗi "Vật liệu không tồn tại"
Hậu điều kiện	- Thông tin chi tiết vật liệu được hiển thị

Bảng 3.28 Đặc tả use case UC – 303

Use case ID	UC - 304
Mô tả	Người dùng chỉnh sửa thông tin của một vật liệu.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu của vật liệu)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống

	- Người dùng là chủ sở hữu của vật liệu
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết vật liệu 2. Người dùng nhấn nút "Chỉnh sửa" 3. Hệ thống hiển thị form với thông tin hiện tại 4. Người dùng chỉnh sửa các trường 5. Người dùng nhấn nút "Lưu" 6. Hệ thống validate dữ liệu 7. Hệ thống gửi request PATCH đến API /inventory/:id với JWT token 8. Backend kiểm tra user là owner 9. Backend cập nhật thông tin vào database 10. Hệ thống hiển thị thông báo thành công 11. Hệ thống cập nhật thông tin trên màn hình chi tiết
Ngoại lệ	- Người dùng không phải chủ sở hữu: Hệ thống hiển thị lỗi "Bạn không có quyền chỉnh sửa vật liệu này"
Hậu điều kiện	- Thông tin vật liệu đã được cập nhật

Bảng 3.29 Đặc tả use case UC – 304

Use case ID	UC - 305
Mô tả	Người dùng xóa một vật liệu khỏi kho hàng.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu của vật liệu)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu của vật liệu
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết vật liệu 2. Người dùng nhấn nút "Xóa" 3. Hệ thống hiển thị dialog xác nhận 4. Người dùng xác nhận xóa 5. Hệ thống gửi request DELETE đến API /inventory/:id với JWT token 6. Backend kiểm tra user là owner 7. Backend xóa inventory item khỏi database 8. Hệ thống hiển thị thông báo thành công 9. Hệ thống chuyển về màn hình danh sách kho hàng
Ngoại lệ	- Người dùng hủy: Hệ thống đóng dialog, không xóa

	- User không phải owner: Hệ thống hiển thị lỗi "Bạn không có quyền xóa vật liệu này"
Hậu điều kiện	- Vật liệu đã được xóa khỏi kho hàng

Bảng 3.30 Đặc tả use case UC – 305

Use case ID	UC - 306
Mô tả	Người dùng xem lịch sử sử dụng của một vật liệu.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu của vật liệu)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Vật liệu tồn tại
Luồng sự kiện chính	1. Người dùng mở màn hình chi tiết vật liệu 2. Người dùng chọn tab "Lịch sử" 3. Hệ thống gửi request đến API /inventory/:id/logs với JWT token 4. Backend lấy tất cả inventory logs của vật liệu 5. Backend trả về danh sách logs 6. Hệ thống hiển thị danh sách với thông tin: - Ngày sử dụng - Số lượng sử dụng - Mùa vụ sử dụng - Người sử dụng - Ghi chú
Ngoại lệ	- Không có lịch sử: Hệ thống hiển thị "Chưa có lịch sử sử dụng"
Hậu điều kiện	- Lịch sử sử dụng vật liệu được hiển thị

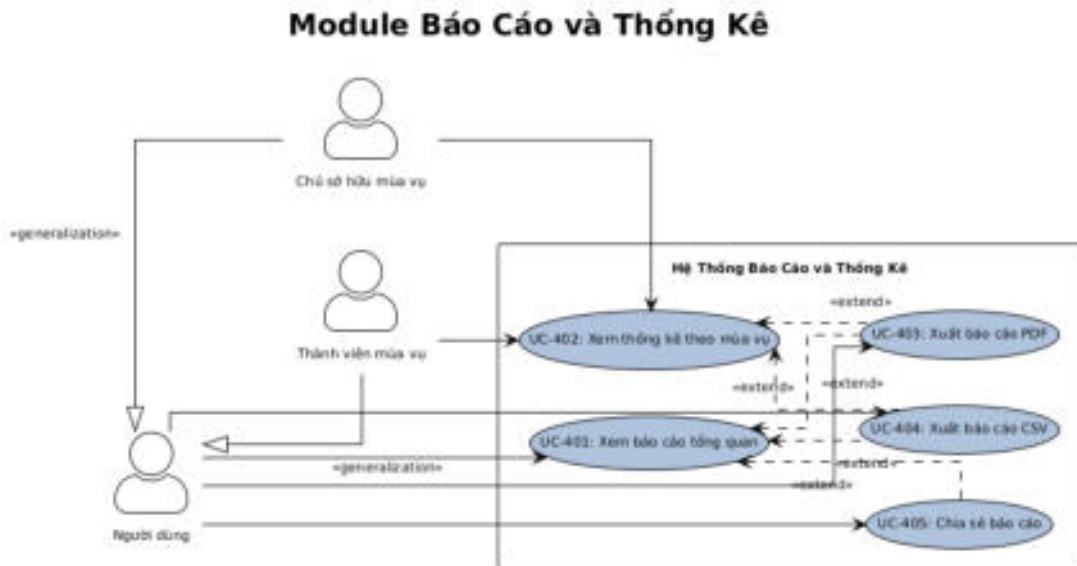
Bảng 3.31 Đặc tả use case UC – 306

Use case ID	UC - 307
Mô tả	Người dùng xem tất cả vật liệu đã được sử dụng trong một mùa vụ.
Tác nhân	Người dùng (đã đăng nhập, là chủ sở hữu hoặc thành viên của mùa vụ)
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu hoặc thành viên của mùa vụ
Luồng sự kiện chính	1. Người dùng mở màn hình chi tiết mùa vụ 2. Người dùng chọn tab "Vật liệu"

	<p>3. Hệ thống gửi request đến API /inventory/season/:seasonId/logs với JWT token</p> <p>4. Backend lấy tất cả inventory logs của mùa vụ</p> <p>5. Backend trả về danh sách logs</p> <p>6. Hệ thống hiển thị danh sách vật liệu đã sử dụng trong mùa vụ</p>
Hậu điều kiện	- Lịch sử vật liệu theo mùa vụ được hiển thị

Bảng 3.32 Đặc tả use case UC – 307

3.2.5 Báo cáo và thống kê



Hình 3.6 Sơ đồ Usecase báo cáo và thống kê

Use case ID	UC - 401
Mô tả	Người dùng xem báo cáo tổng quan về tài chính và hoạt động.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Có ít nhất một giao dịch hoặc mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Báo cáo" 2. Hệ thống tính toán và hiển thị: <ul style="list-style-type: none"> - Tổng thu nhập (tất cả mùa vụ) - Tổng chi phí (tất cả mùa vụ) - Lợi nhuận tổng - Số lượng mùa vụ - Số lượng giao dịch

	<ul style="list-style-type: none"> - Biểu đồ thu chi theo thời gian - Top danh mục chi phí 3. Người dùng có thể chọn khoảng thời gian để xem báo cáo
Hậu điều kiện	- Báo cáo tổng quan được hiển thị

Bảng 3.33 Đặc tả use case UC – 401

Use case ID	UC - 402
Mô tả	Người dùng xem báo cáo tổng quan về tài chính và hoạt động.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Người dùng là chủ sở hữu hoặc thành viên của mùa vụ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình chi tiết mùa vụ 2. Người dùng chọn tab "Thống kê" 3. Hệ thống gửi request đến API /seasons/:id/stats 4. Backend tính toán thống kê: <ul style="list-style-type: none"> - Tổng thu nhập - Tổng chi phí - Lợi nhuận - Số lượng giao dịch - Biểu đồ thu chi theo tháng/tuần - Phân bổ chi phí theo danh mục 5. Backend trả về dữ liệu thống kê 6. Hệ thống hiển thị thống kê với biểu đồ
Hậu điều kiện	- Thống kê mùa vụ được hiển thị

Bảng 3.34 Đặc tả use case UC – 402

Use case ID	UC - 403
Mô tả	Người dùng xuất báo cáo ra file PDF.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Có dữ liệu để xuất báo cáo
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Báo cáo"

	<ol style="list-style-type: none"> 2. Người dùng chọn loại báo cáo (tổng quan hoặc theo mùa vụ) 3. Người dùng nhấn nút "Xuất PDF" 4. Hệ thống tạo file PDF với nội dung: <ul style="list-style-type: none"> - Thông tin tổng quan - Bảng dữ liệu chi tiết - Ngày xuất báo cáo 5. Hệ thống lưu file PDF vào thiết bị 6. Hệ thống hiển thị thông báo thành công 7. Hệ thống cho phép mở file hoặc chia sẻ
Ngoại lệ	- Không có quyền lưu file: Hệ thống yêu cầu cấp quyền
Hậu điều kiện	- File PDF đã được tạo và lưu

Bảng 3.35 Đặc tả use case UC – 403

Use case ID	UC - 404
Mô tả	Người dùng xuất dữ liệu giao dịch ra file CSV.
Tác nhân	Người dùng (đã đăng nhập)
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống - Có dữ liệu để xuất báo cáo
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở màn hình "Giao dịch" hoặc "Báo cáo" 2. Người dùng nhấn nút "Xuất CSV" 3. Hệ thống lấy dữ liệu giao dịch 4. Hệ thống tạo file CSV với các cột: <ul style="list-style-type: none"> - Ngày - Loại (Thu/Chi) - Danh mục - Số tiền - Mô tả - Mùa vụ - Trạng thái phê duyệt 5. Hệ thống lưu file CSV vào thiết bị 6. Hệ thống hiển thị thông báo thành công 7. Hệ thống cho phép mở file hoặc chia sẻ
Ngoại lệ	- Không có quyền lưu file: Hệ thống yêu cầu cấp quyền
Hậu điều kiện	- File CSV đã được tạo và lưu

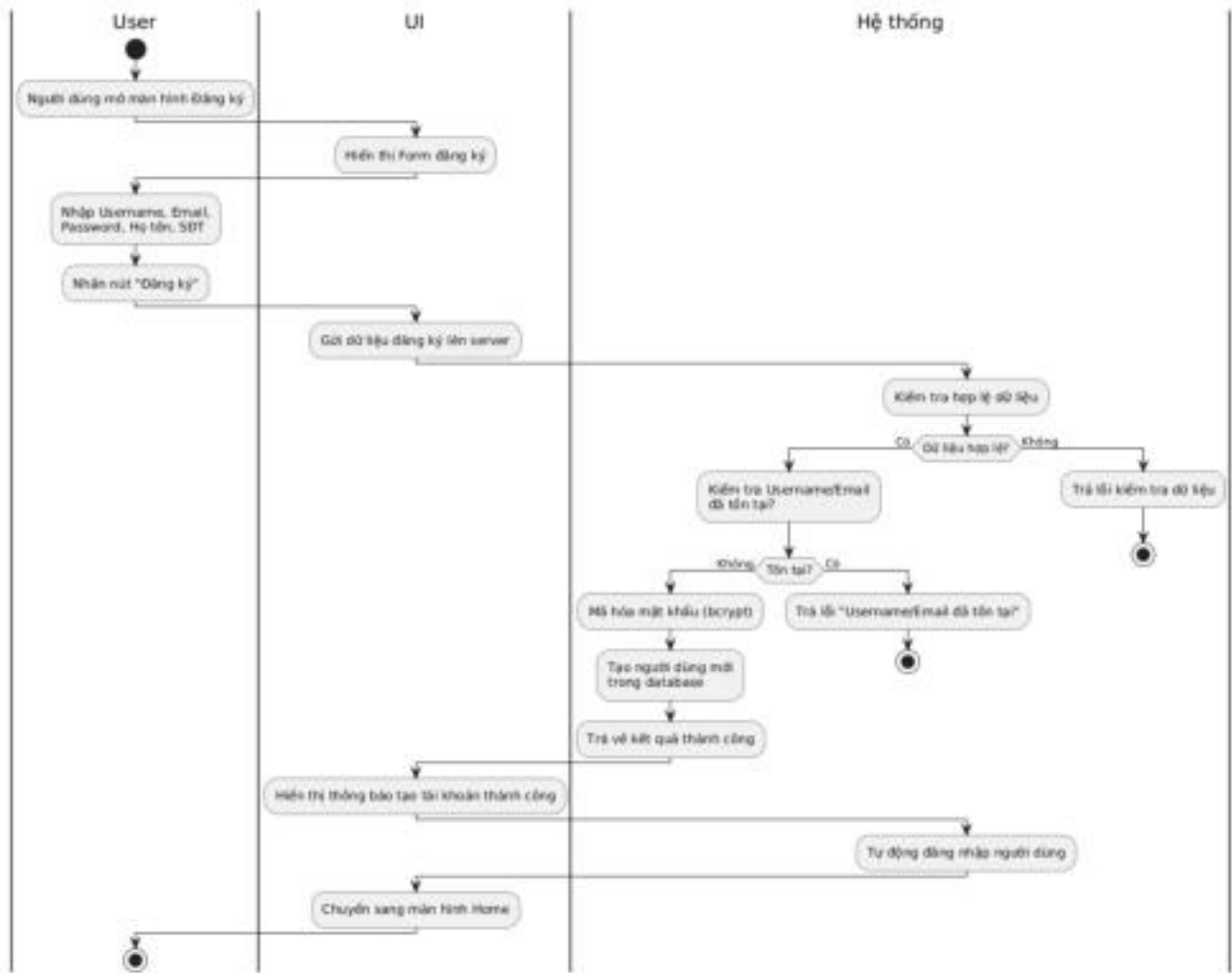
Bảng 3.36 Đặc tả use case UC – 404

Use case ID	UC - 405
Mô tả	Người dùng chia sẻ báo cáo (PDF hoặc CSV) với người khác.
Tác nhân	Người dùng
Tiền điều kiện	- File báo cáo đã được tạo và lưu về máy
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng mở file báo cáo (PDF hoặc CSV) 2. Người dùng nhấn nút "Chia sẻ" 3. Hệ thống hiển thị các tùy chọn chia sẻ: <ul style="list-style-type: none"> - Email - Tin nhắn - Ứng dụng khác 4. Người dùng chọn phương thức chia sẻ 5. Hệ thống mở ứng dụng chia sẻ tương ứng với file đính kèm
Hậu điều kiện	- File báo cáo đã được chia sẻ

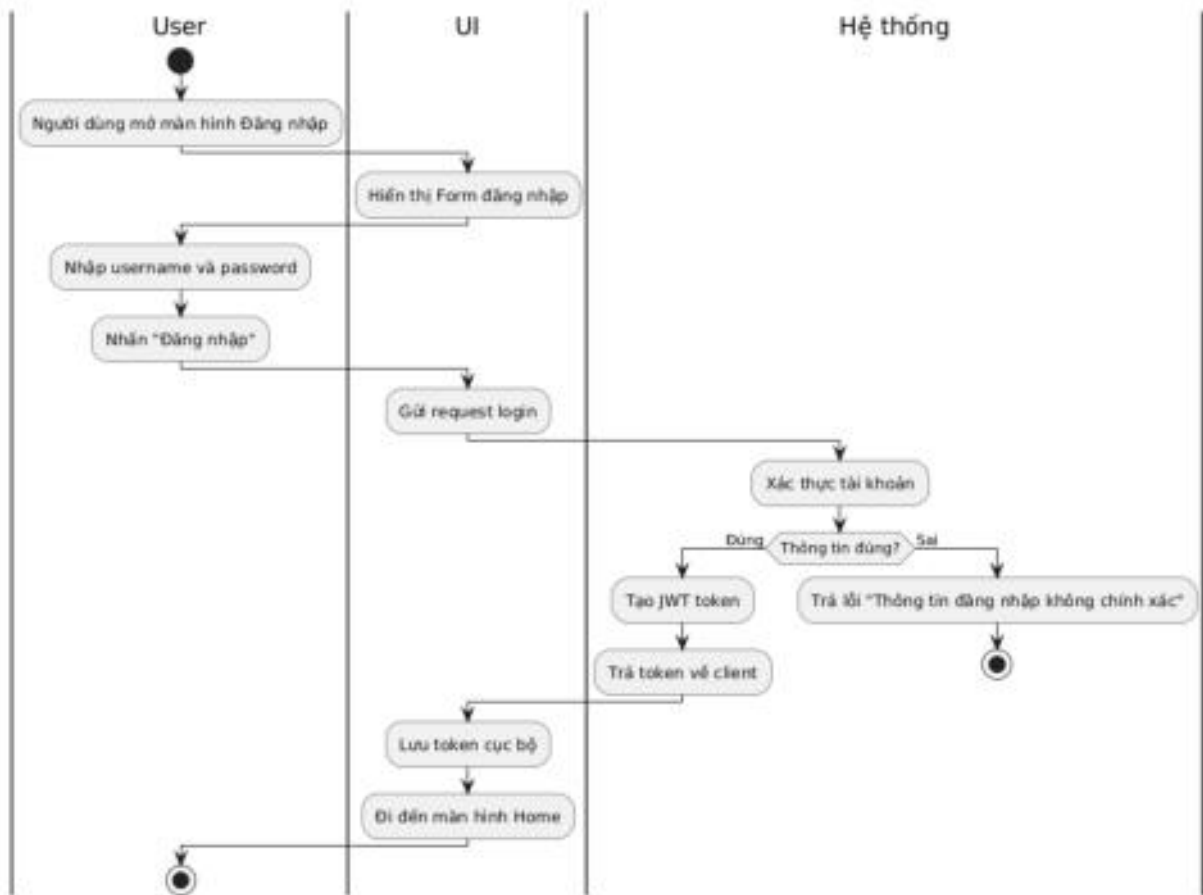
Bảng 3.37 Đặc tả use case UC – 405

3.3 Biểu đồ hoạt động

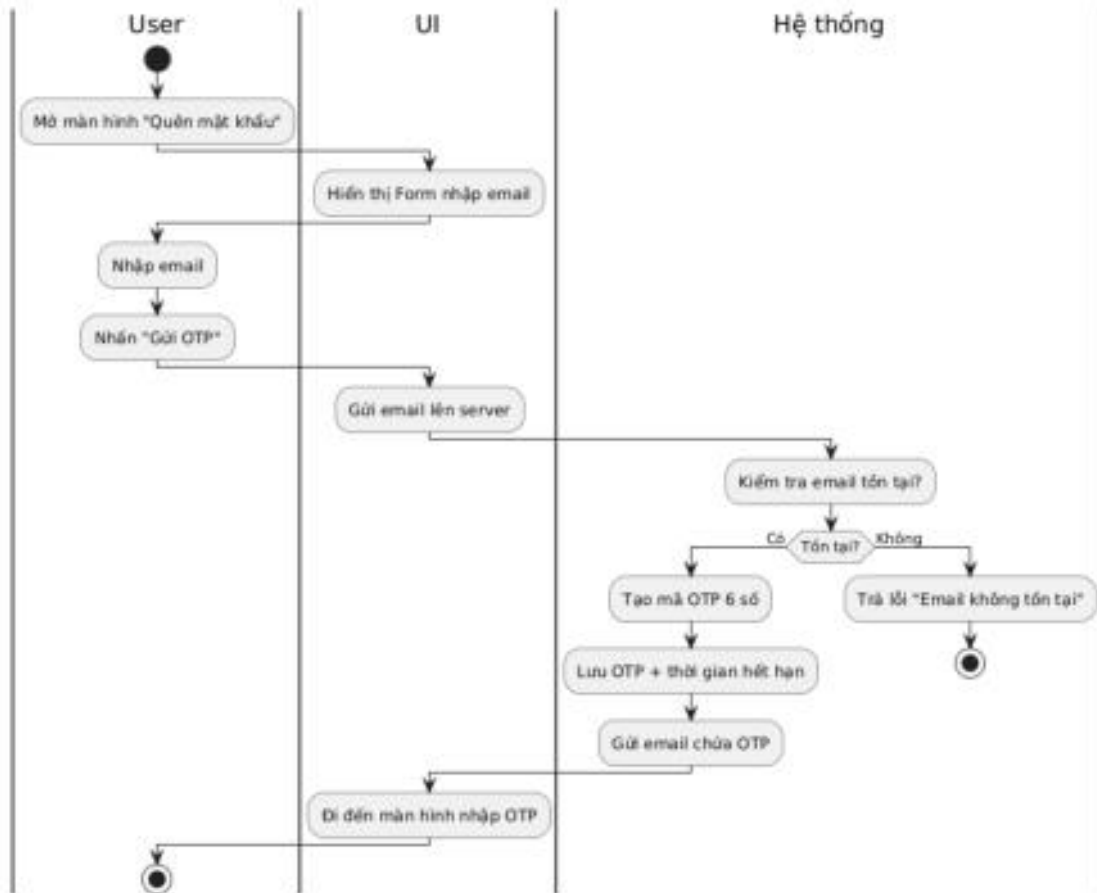
3.3.1 Quản lý truy cập



Hình 3.7 Sơ đồ hoạt động đăng ký tài khoản



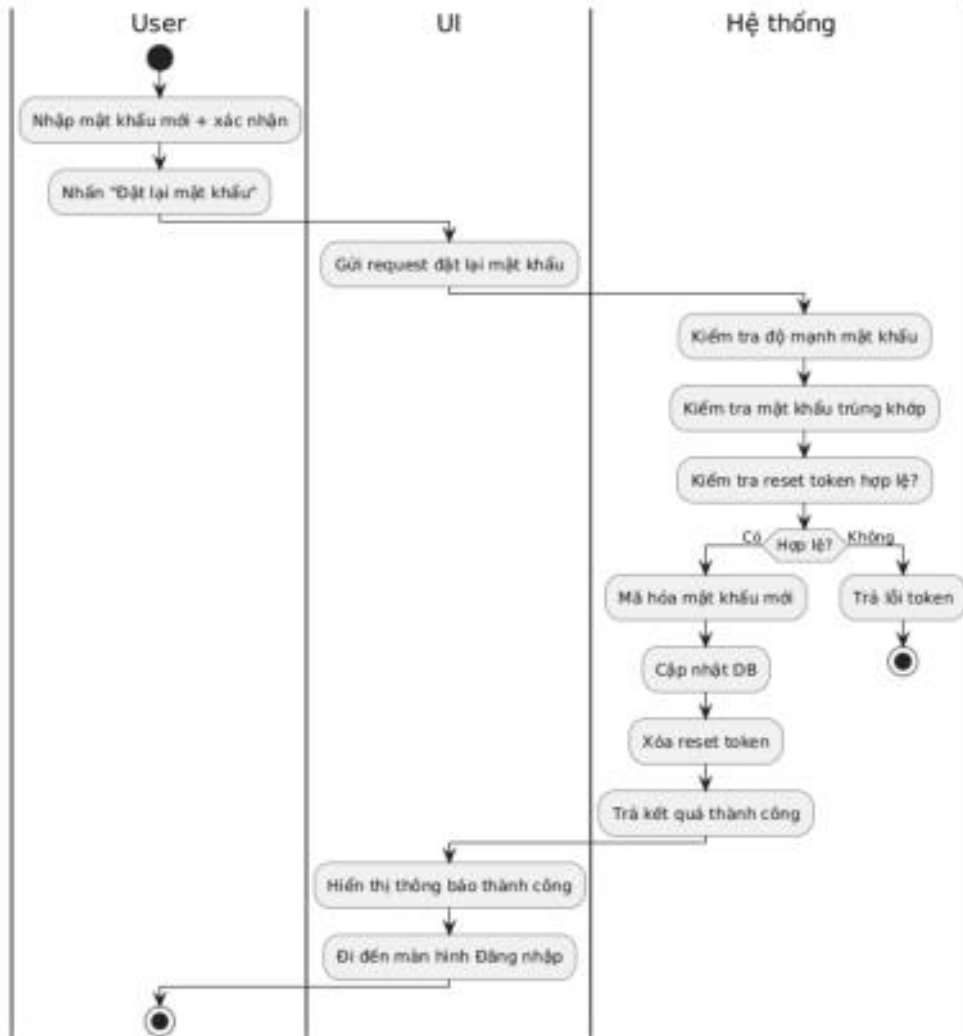
Hình 3.8 Sơ đồ hoạt động Đăng nhập



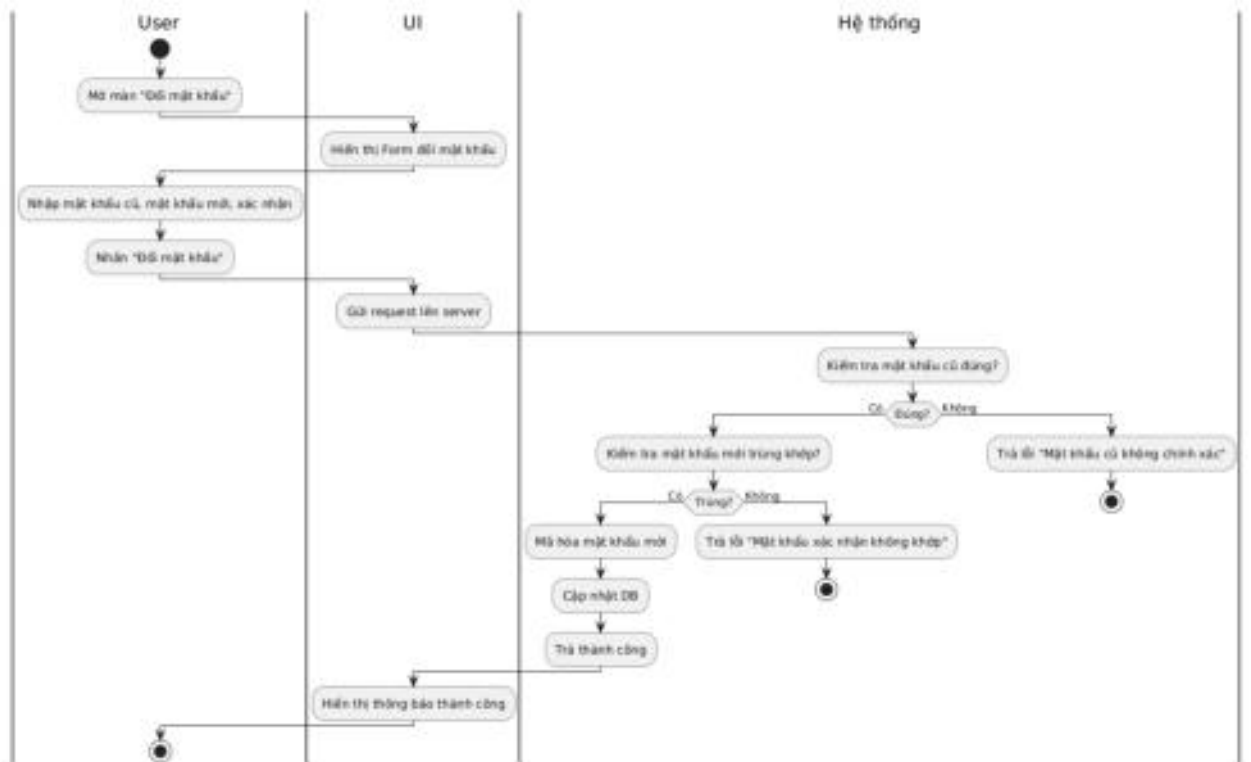
Hình 3.9 Sơ đồ hoạt động Quên mật khẩu



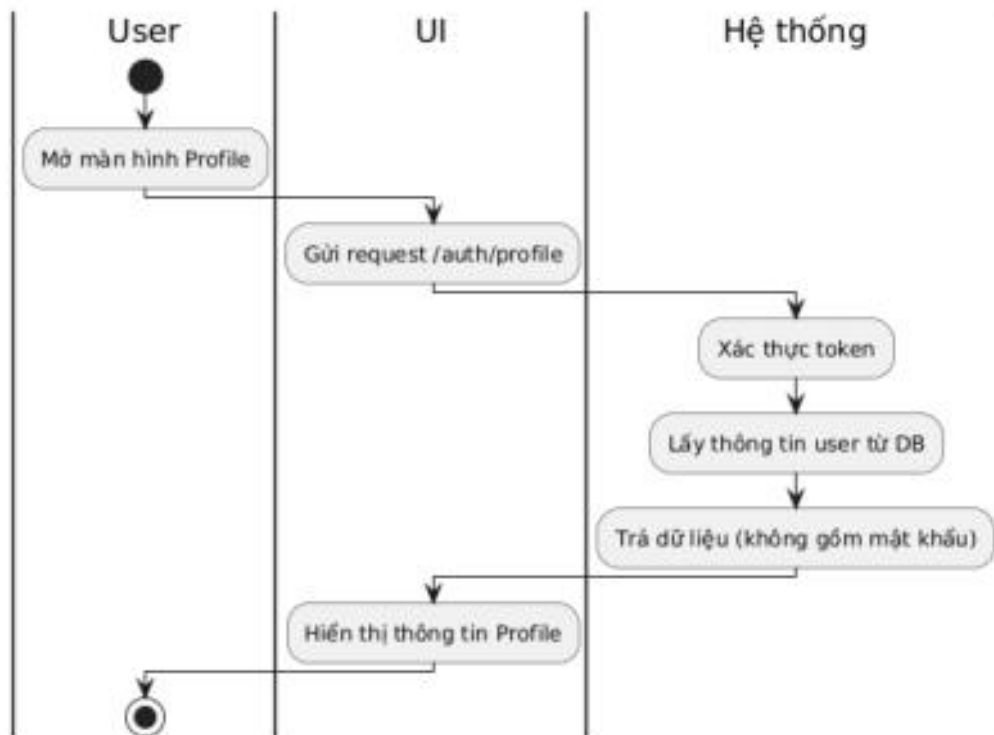
Hình 3.10 Sơ đồ hoạt động Xác minh OTP



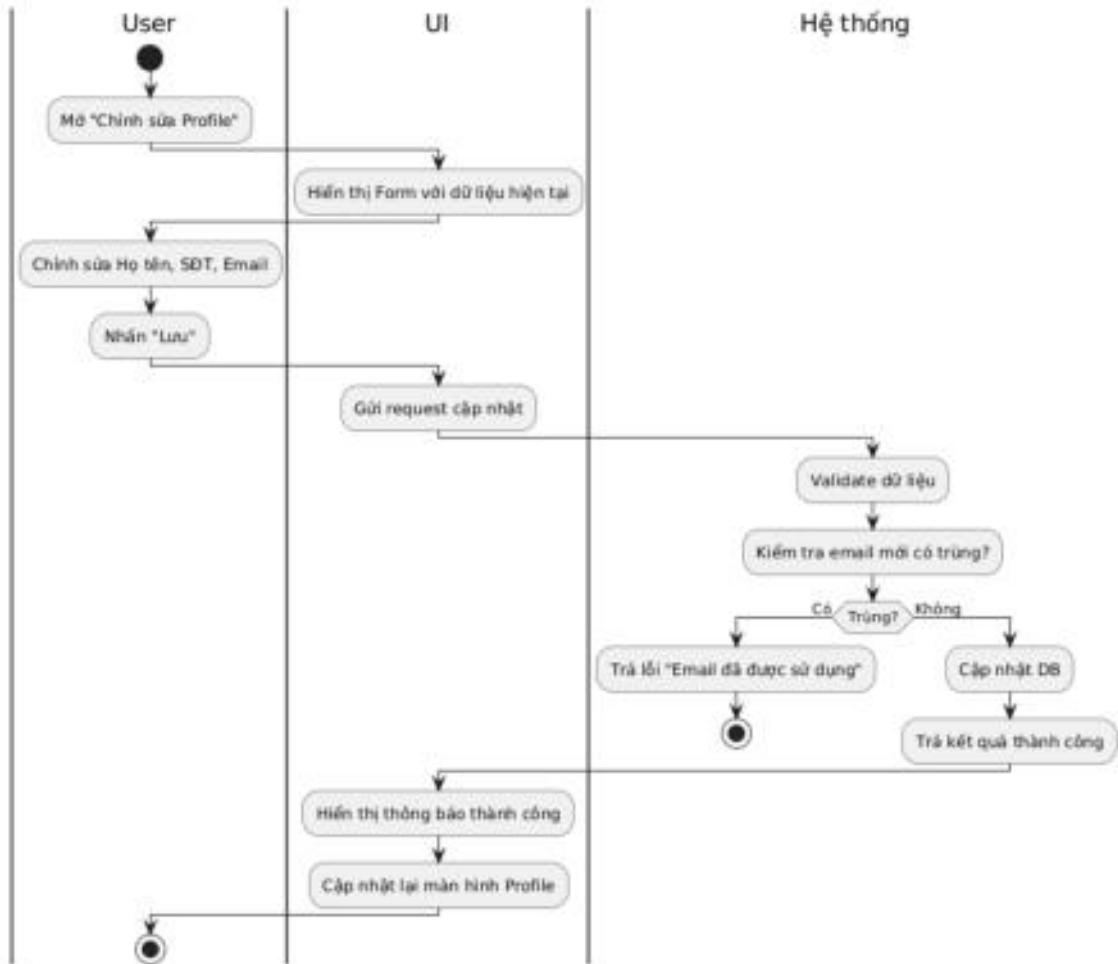
Hình 3.11 Sơ đồ hoạt động Đặt lại mật khẩu



Hình 3.12 Sơ đồ hoạt động Đổi mật khẩu

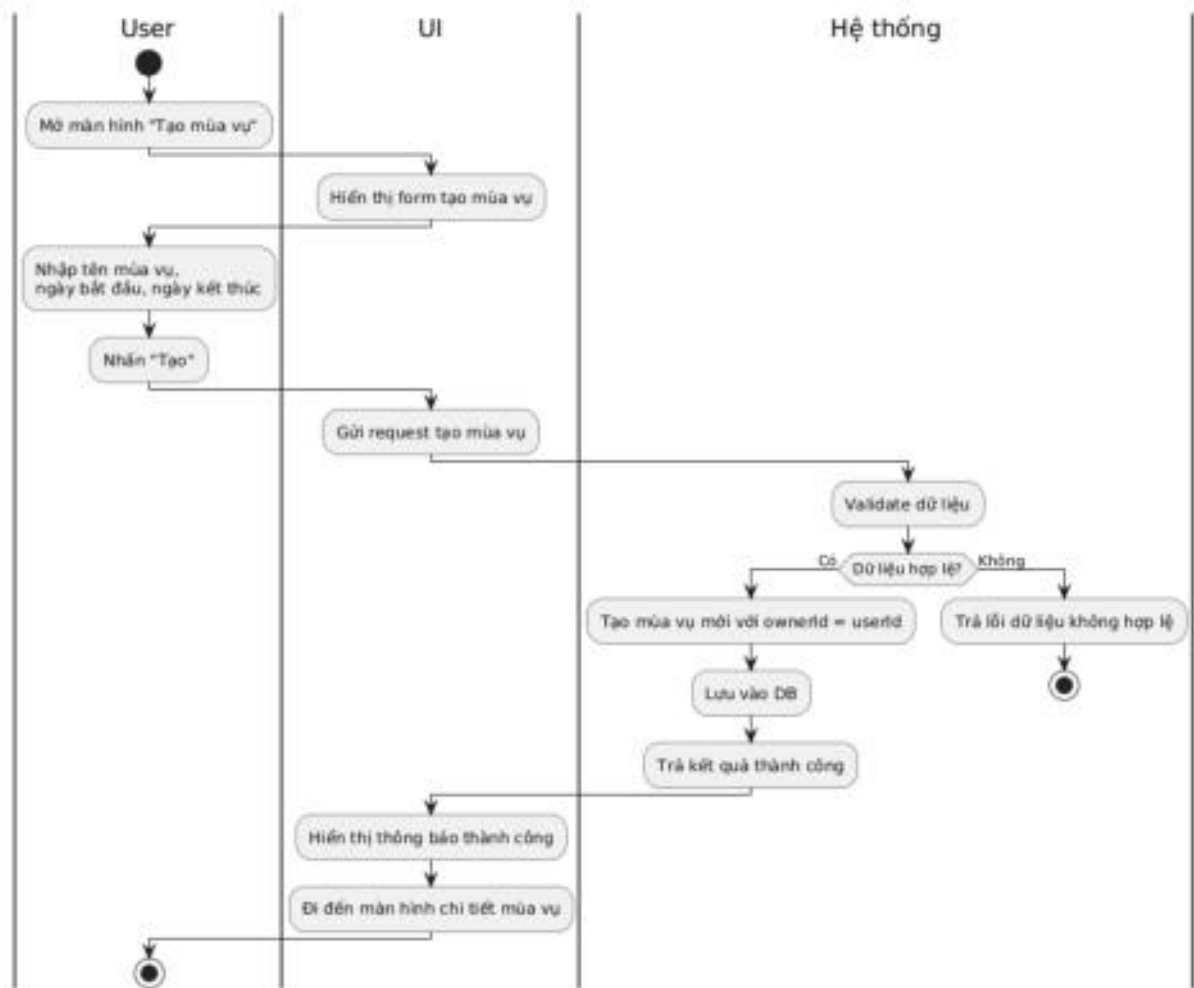


Hình 3.13 Sơ đồ hoạt động Xem thông tin profile



Hình 3.14 Sơ đồ hoạt động Chỉnh sửa thông tin profile

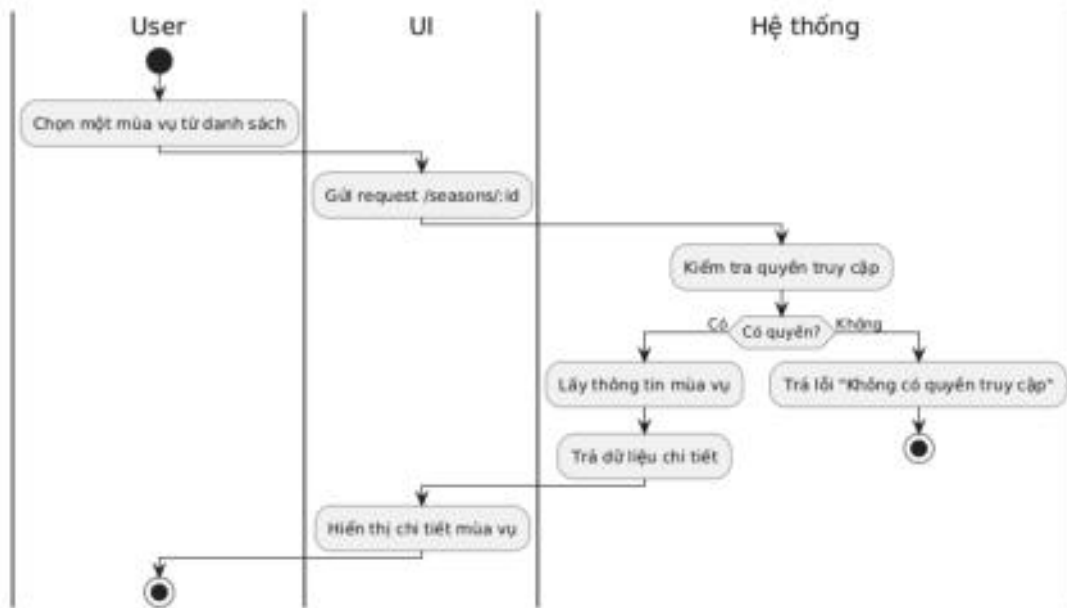
3.3.2 Quản lý mùa vụ



Hình 3. 15 Sơ đồ hoạt động Tạo mùa vụ mới



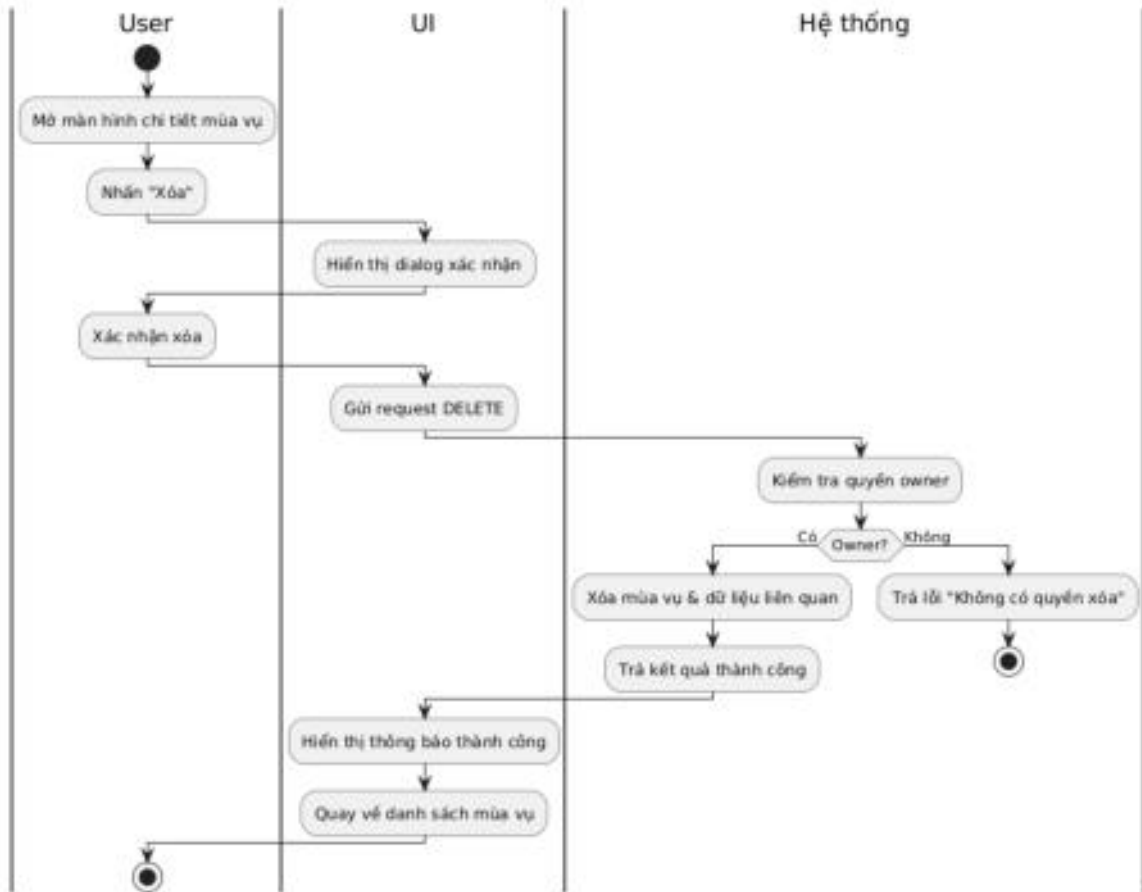
Hình 3.16 Sơ đồ hoạt động Xem danh sách mùa vụ



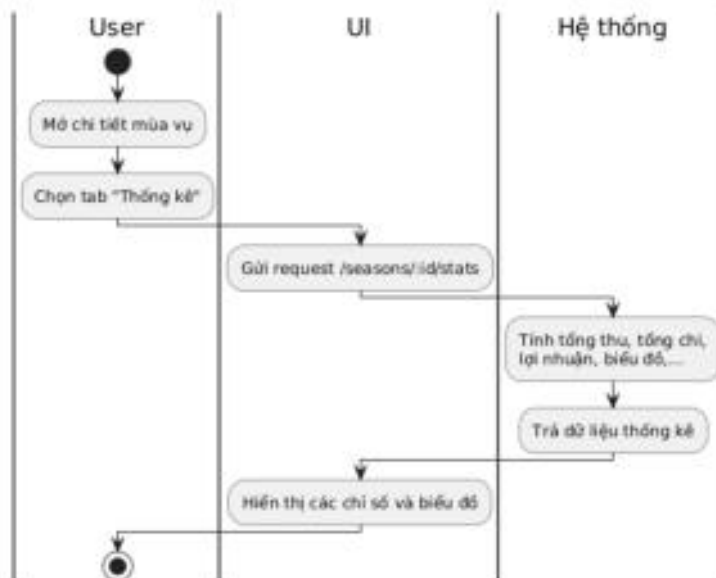
Hình 3.17 Sơ đồ hoạt động Xem chi tiết mùa vụ



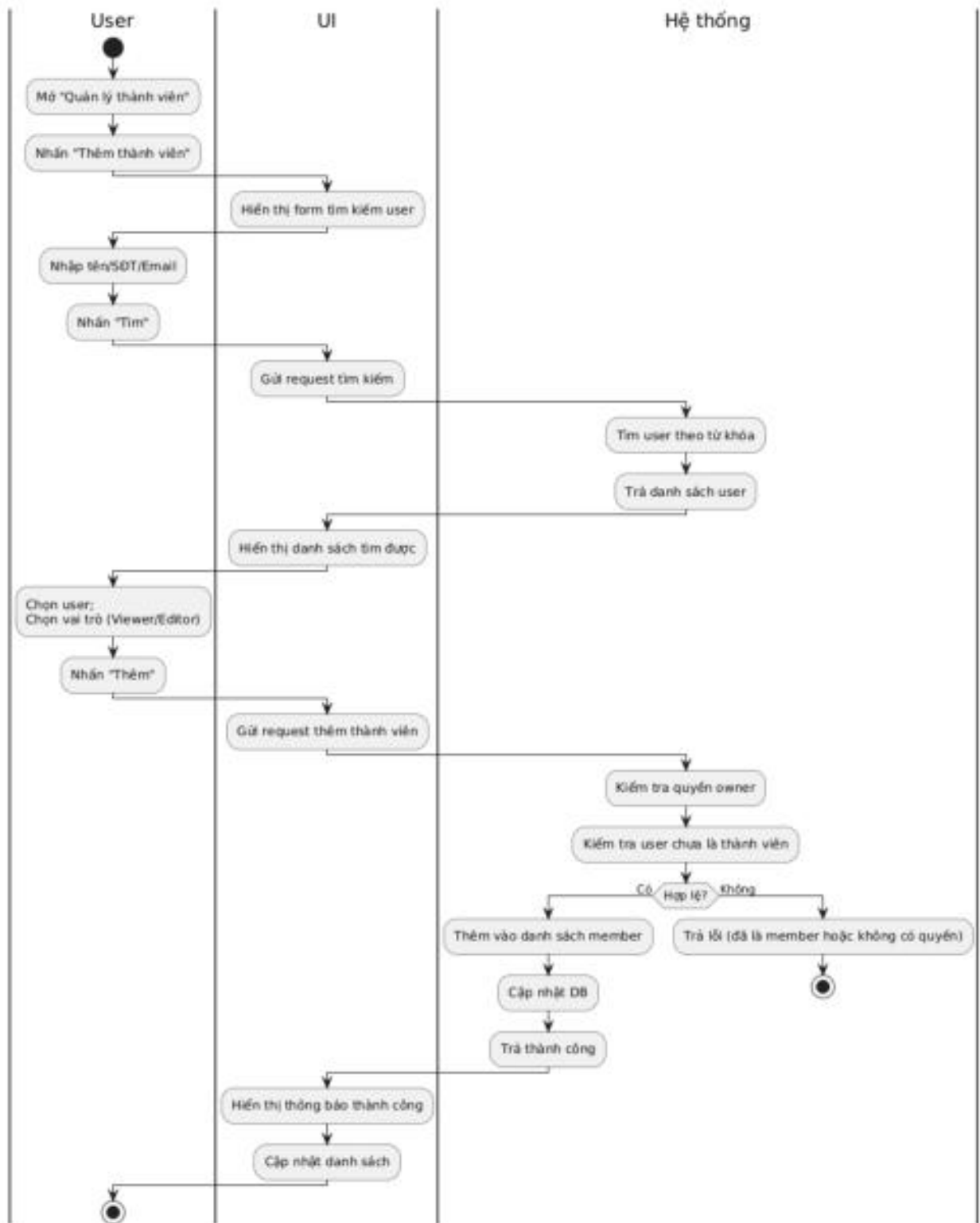
Hình 3.18 Sơ đồ hoạt động Chỉnh sửa mùa vụ



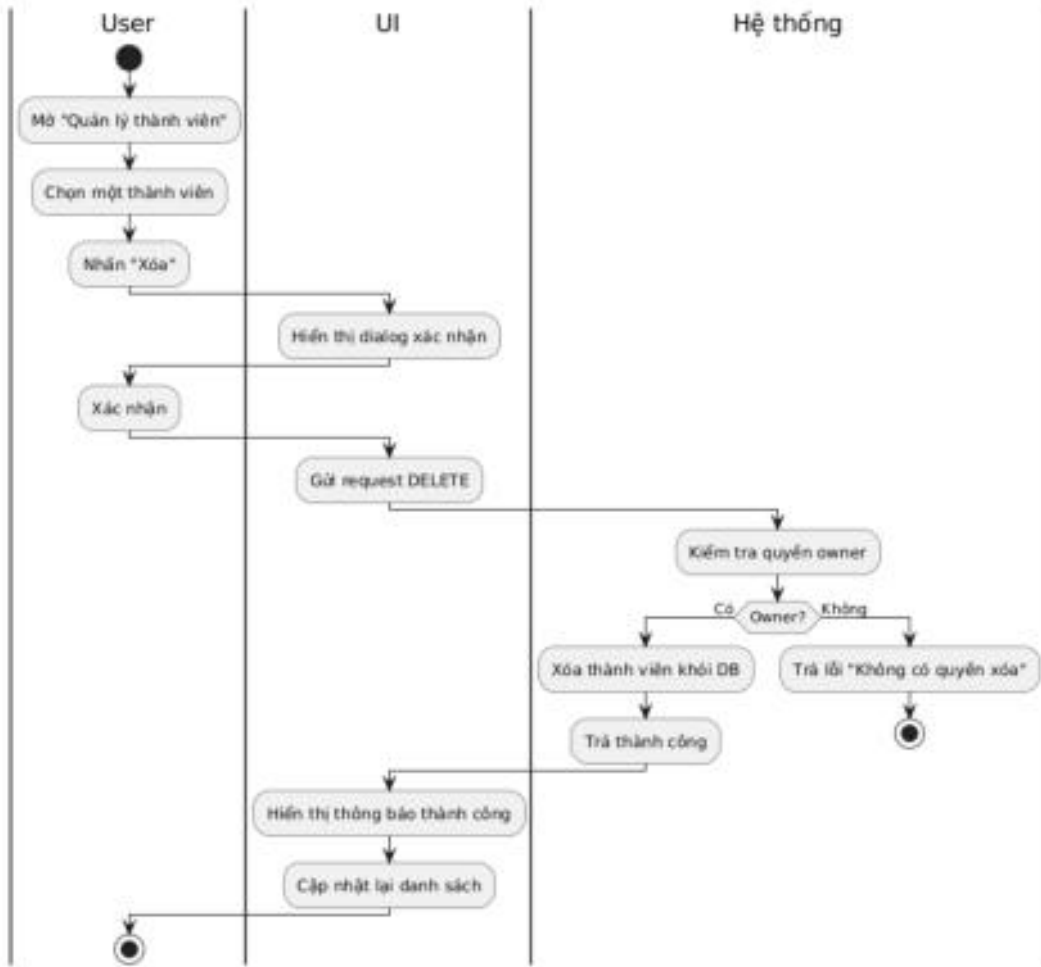
Hình 3.19 Sơ đồ hoạt động Xóa mùa vụ



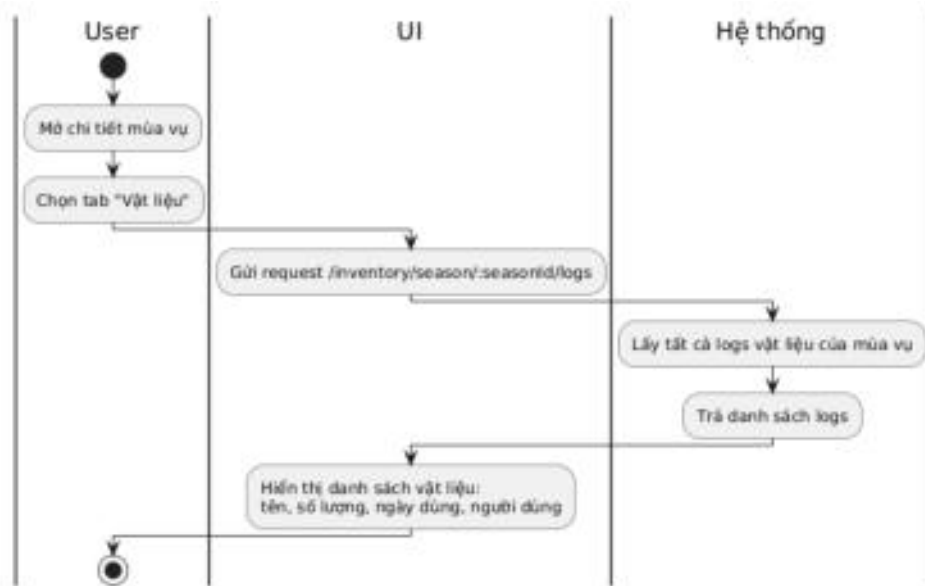
Hình 3.20 Sơ đồ hoạt động Xem thống kê mùa vụ



Hình 3.21 Sơ đồ hoạt động Thêm thành viên vào mùa vụ

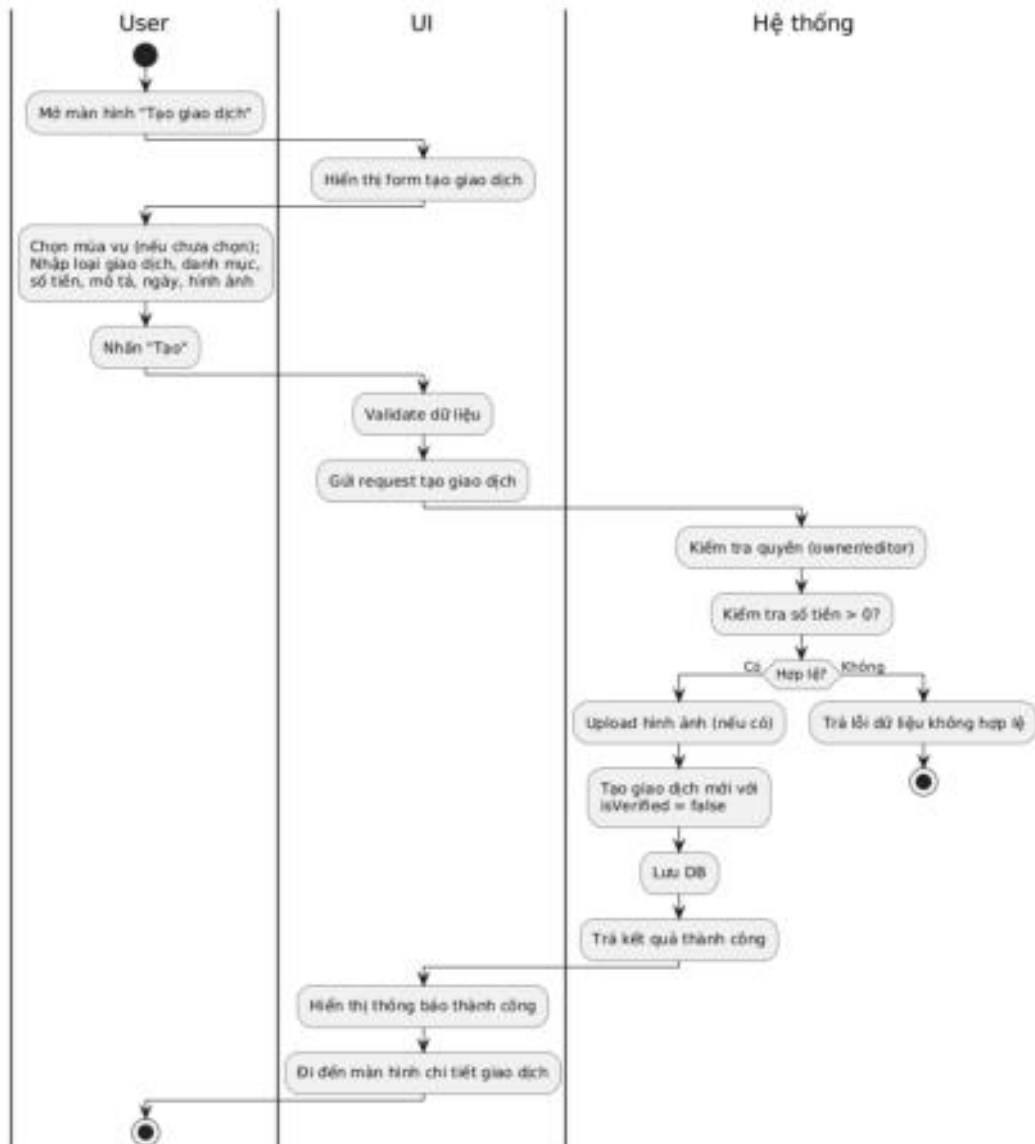


Hình 3.22 Sơ đồ hoạt động Xóa thành viên khỏi mùa vụ

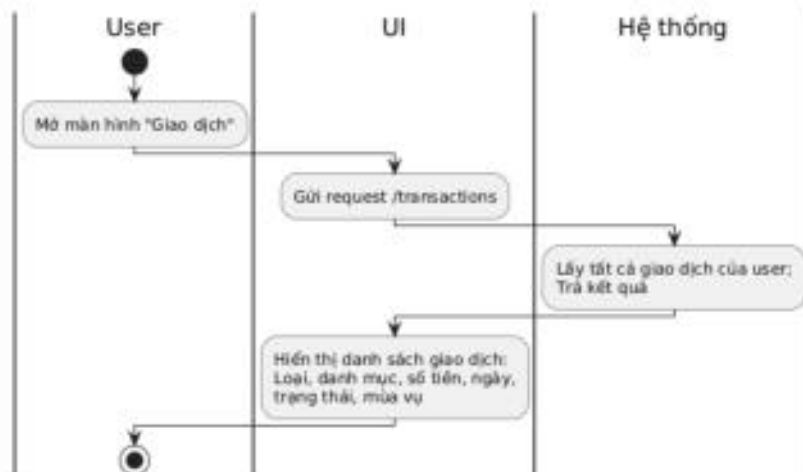


Hình 3.23 Sơ đồ hoạt động Quản lý vật liệu của mùa vụ

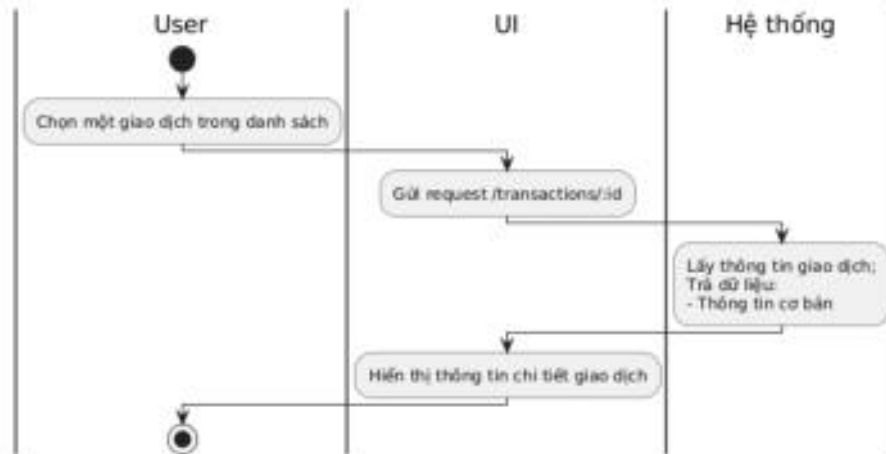
3.3.3 Quản lý giao dịch



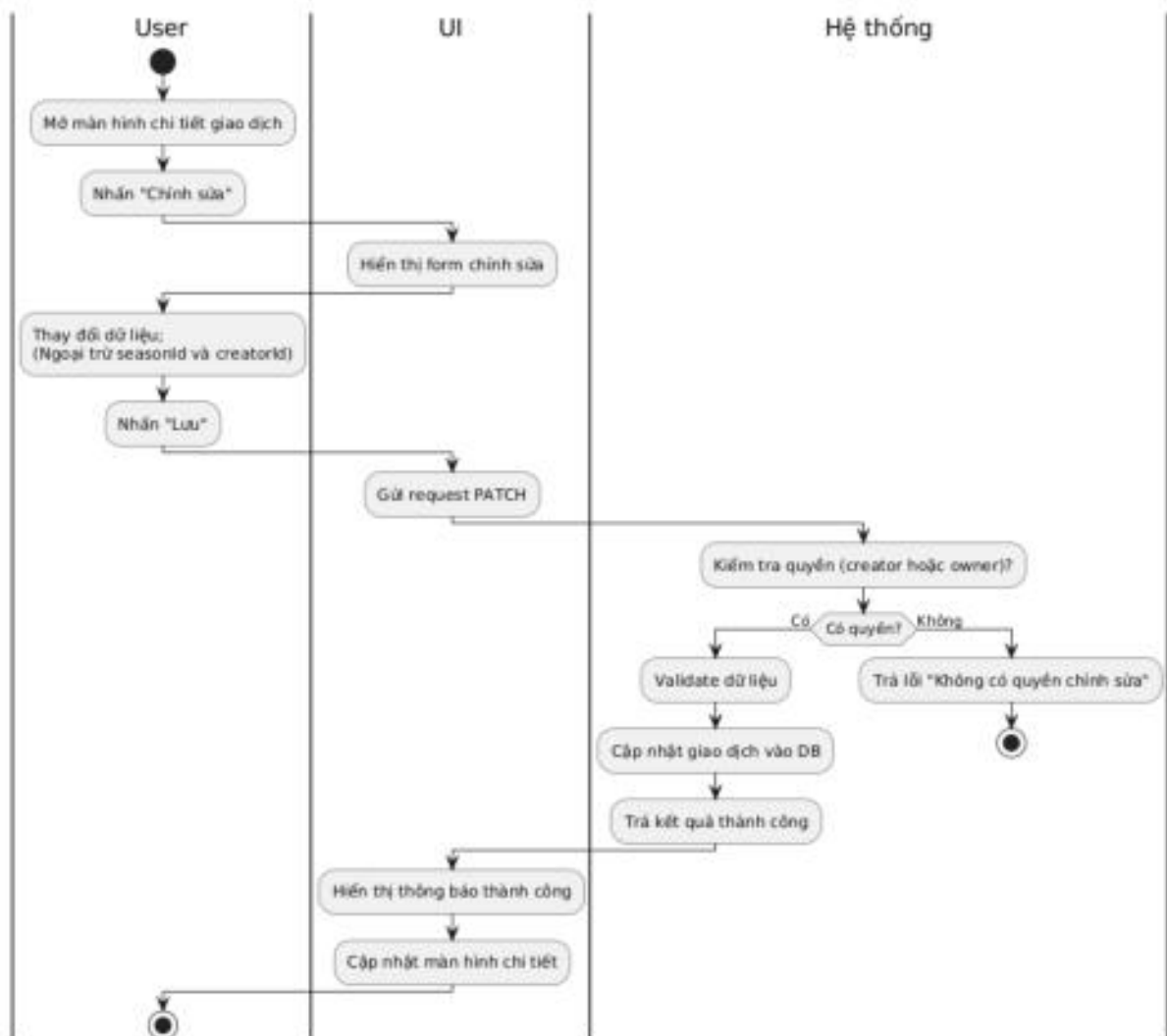
Hình 3.24 Sơ đồ hoạt động Tạo giao dịch mới



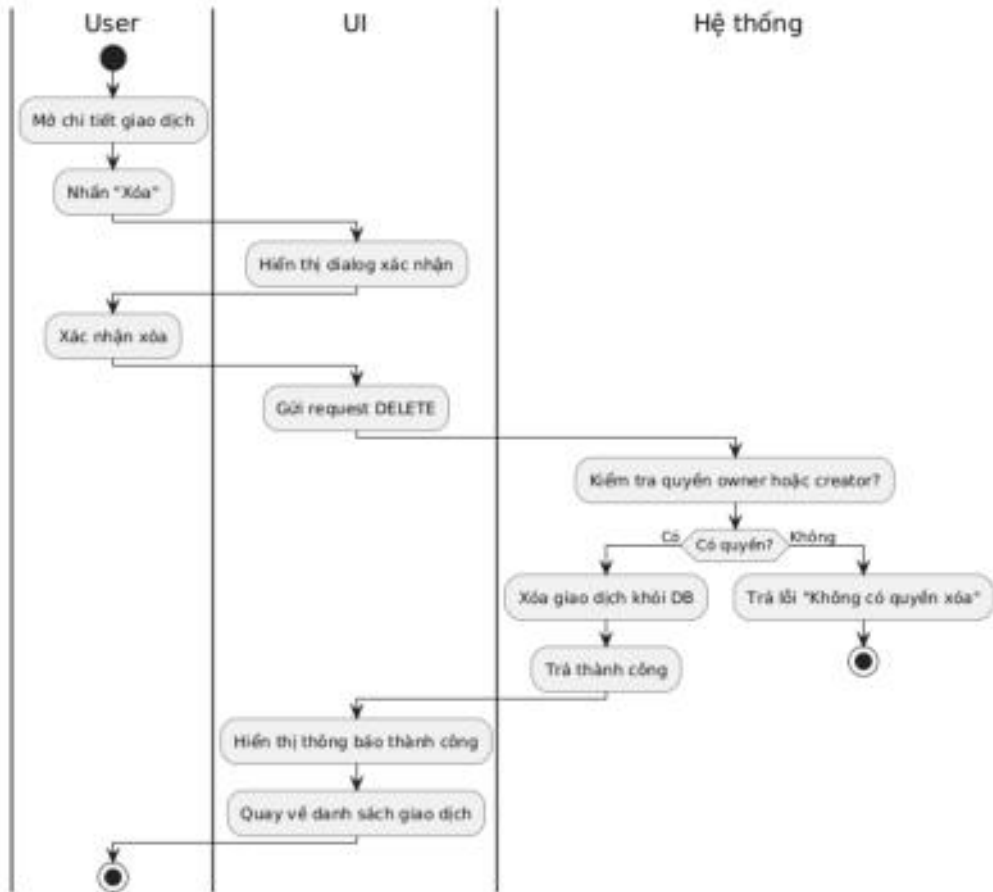
Hình 3.25 Sơ đồ hoạt động Xem danh sách giao dịch



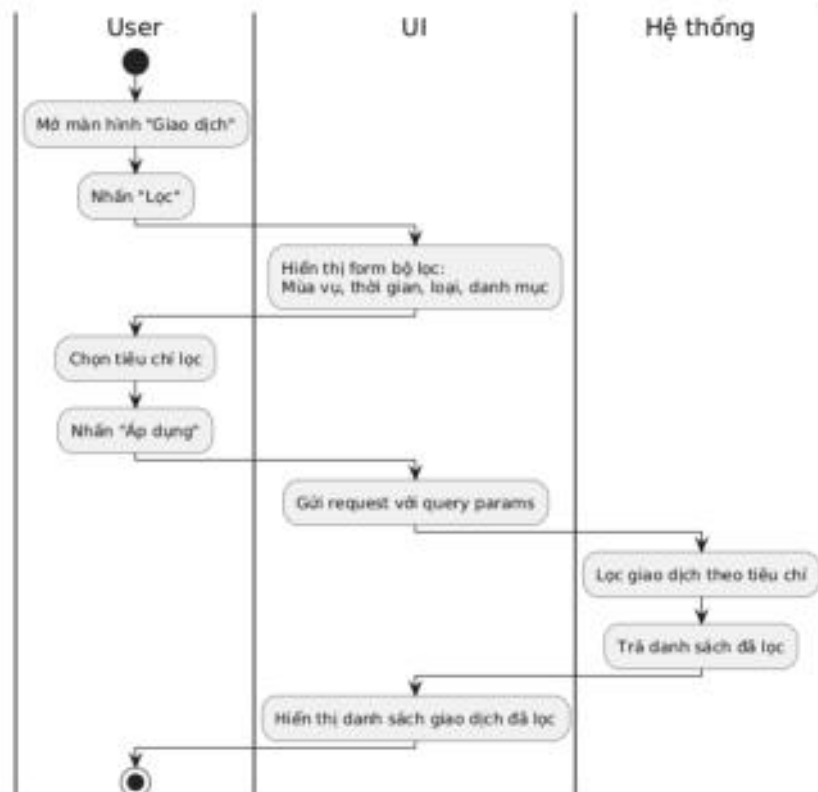
Hình 3.26 Sơ đồ hoạt động Xem chi tiết giao dịch



Hình 3.27 Sơ đồ hoạt động Chỉnh sửa giao dịch



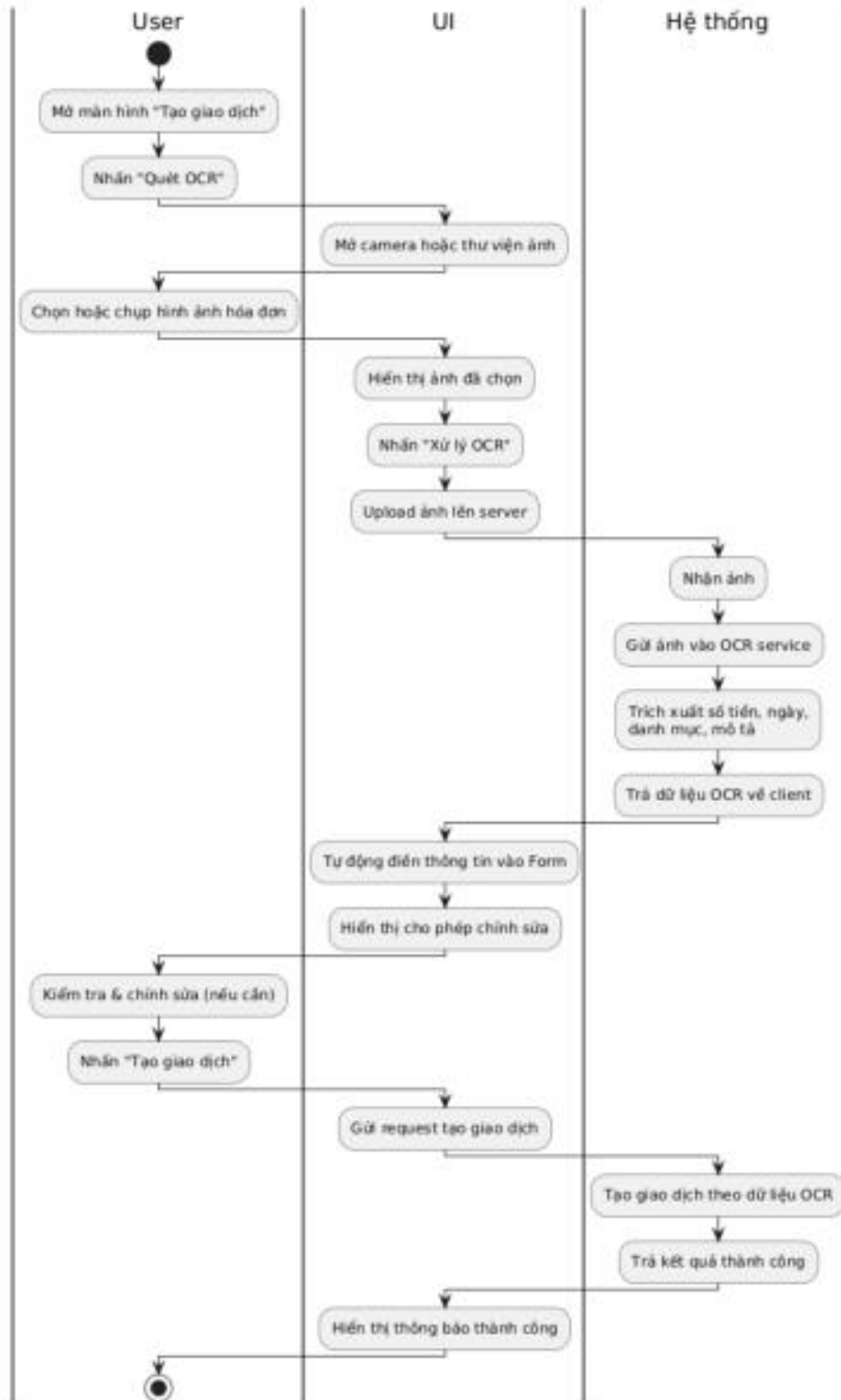
Hình 3.28 Sơ đồ hoạt động Xóa giao dịch



Hình 3.29 Sơ đồ hoạt động Lọc giao dịch

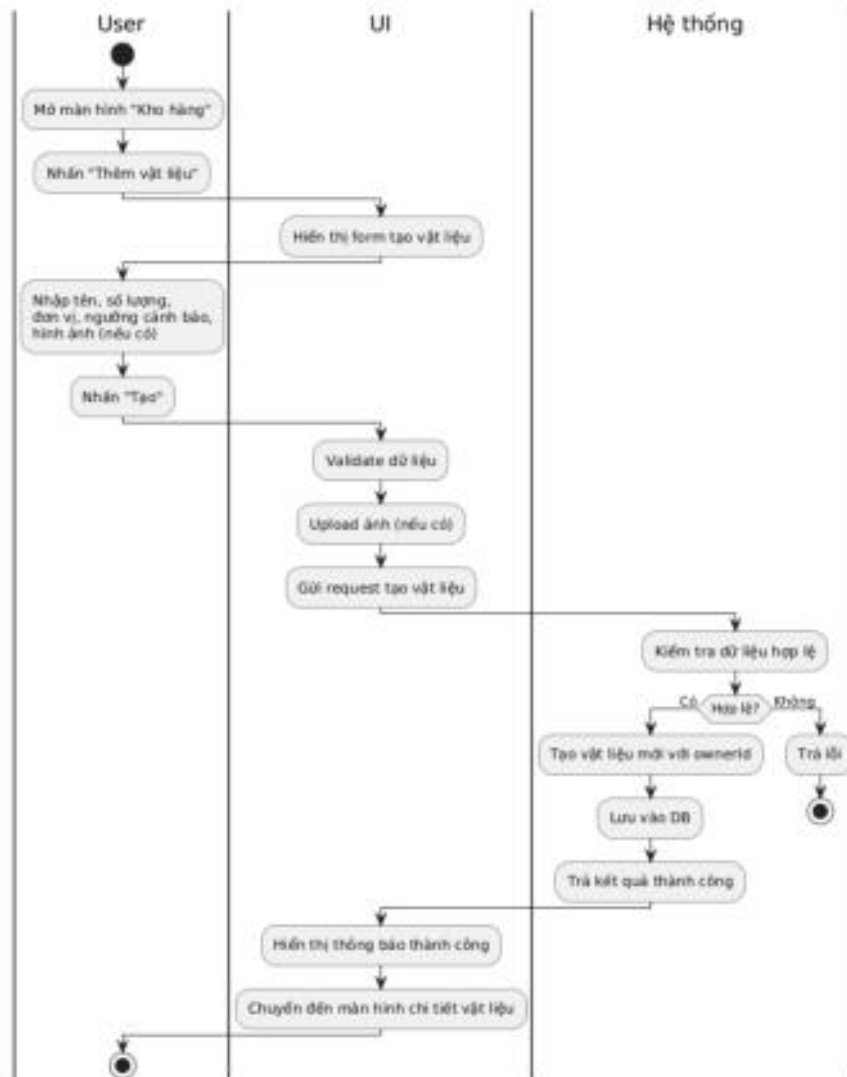


Hình 3.30 Sơ đồ hoạt động Phê duyệt giao dịch

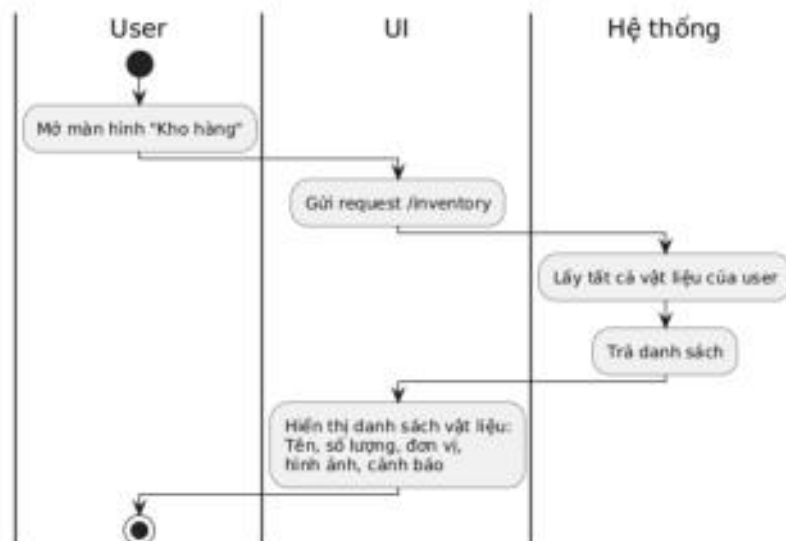


Hình 3.31 Sơ đồ hoạt động Nhập giao dịch từ hình ảnh

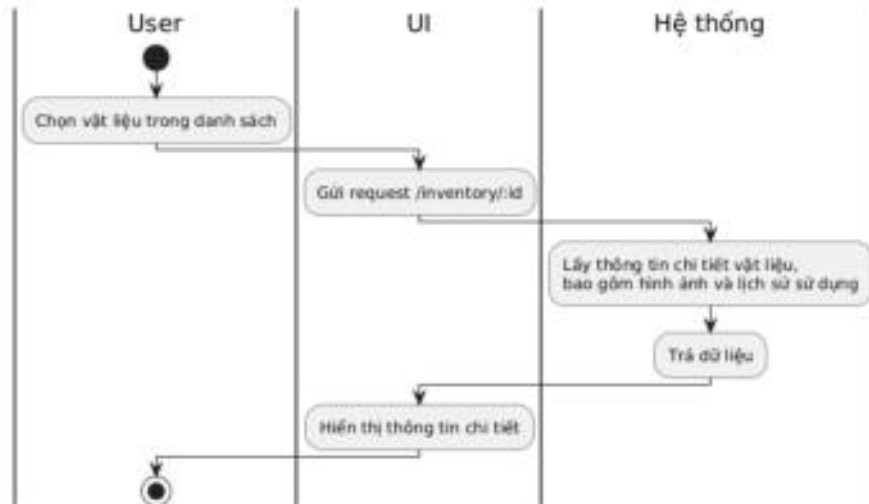
3.3.4 Quản lý vật tư



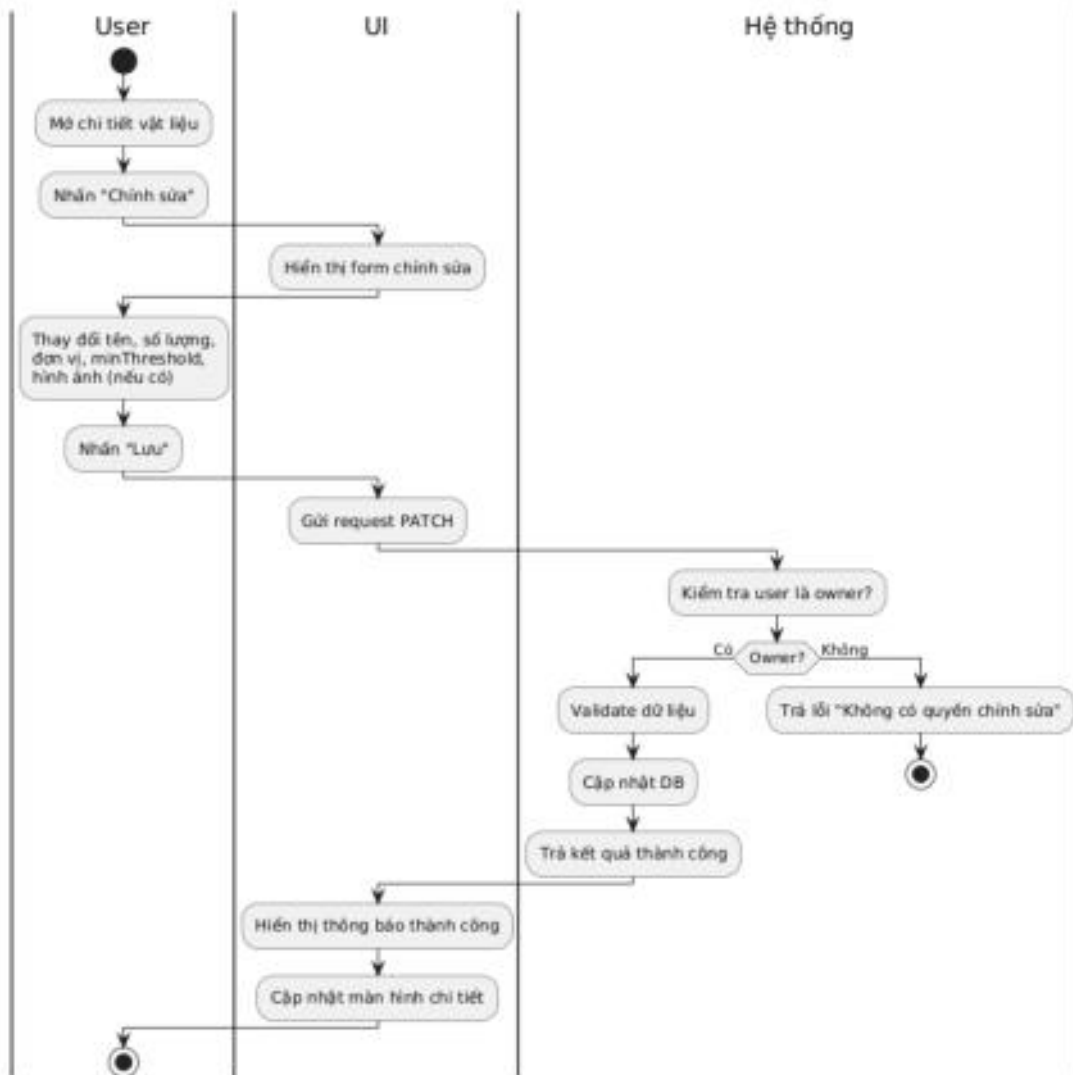
Hình 3.32 Sơ đồ hoạt động Thêm vật tư mới



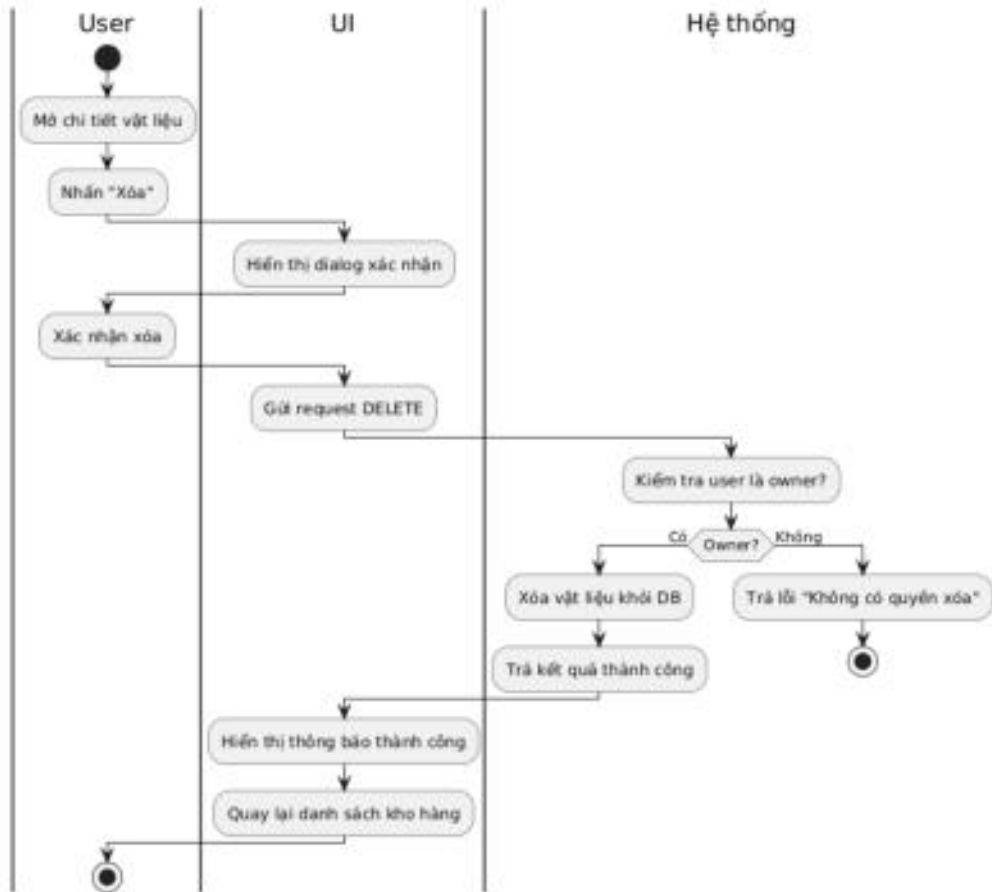
Hình 3.33 Sơ đồ hoạt động Xem danh sách kho hàng



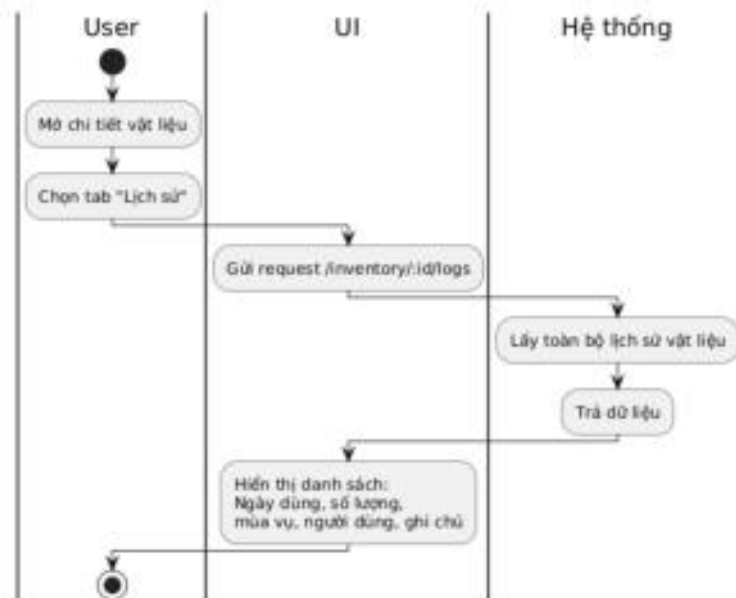
Hình 3.34 Sơ đồ hoạt động Xem chi tiết vật liệu



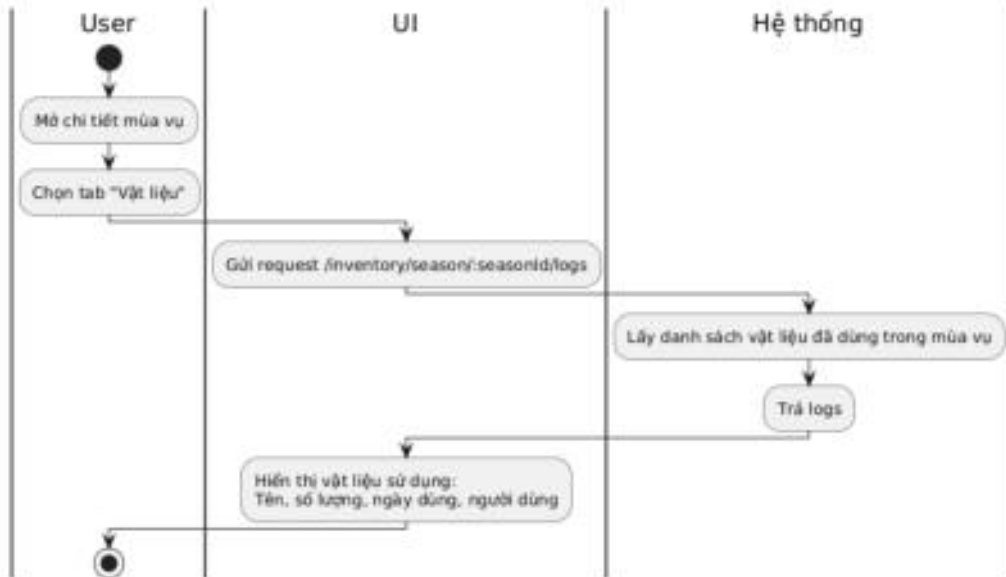
Hình 3.35 Sơ đồ hoạt động Chỉnh sửa vật liệu



Hình 3.36 Sơ đồ hoạt động Xóa vật liệu

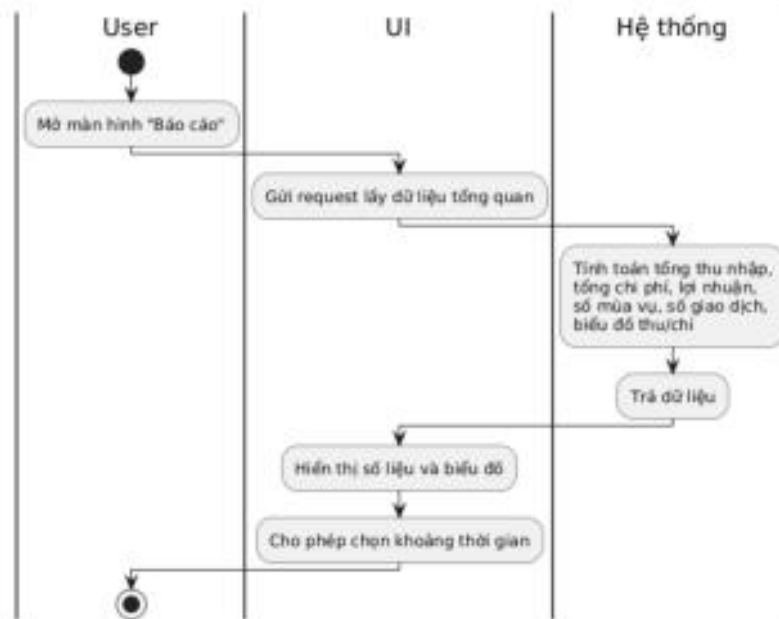


Hình 3.37 Sơ đồ hoạt động Xem lịch sử sử dụng vật liệu

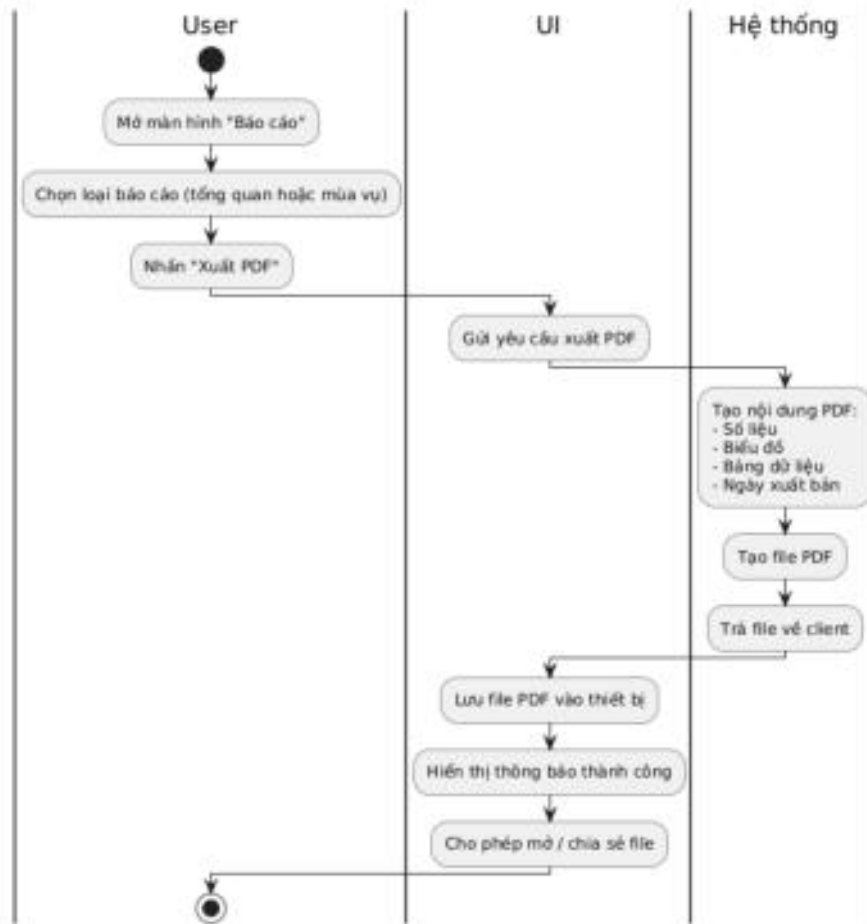


Hình 3.38 Sơ đồ hoạt động Xem lịch sử sử dụng vật liệu theo mùa vụ

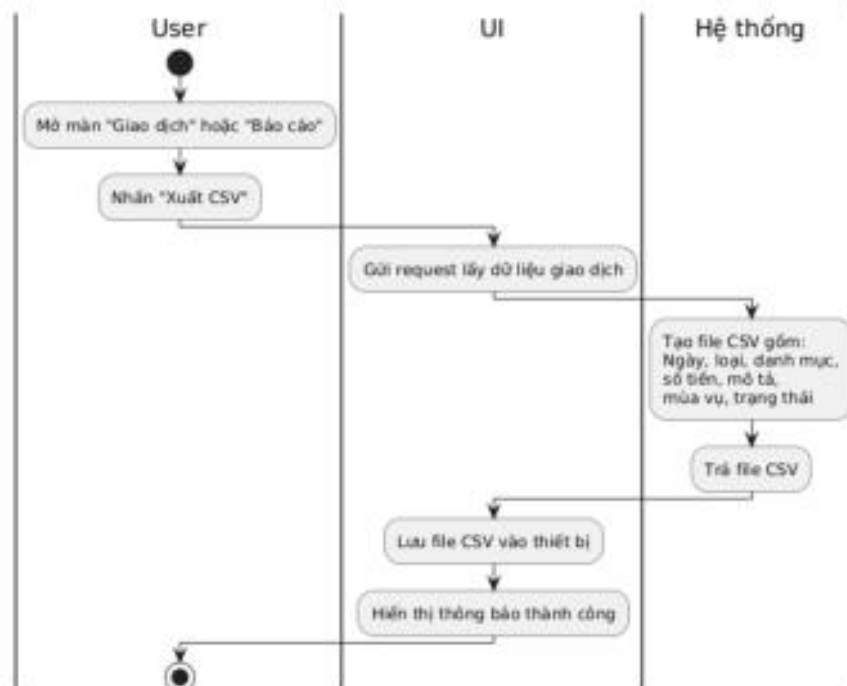
3.3.5 Quản lý minh chứng



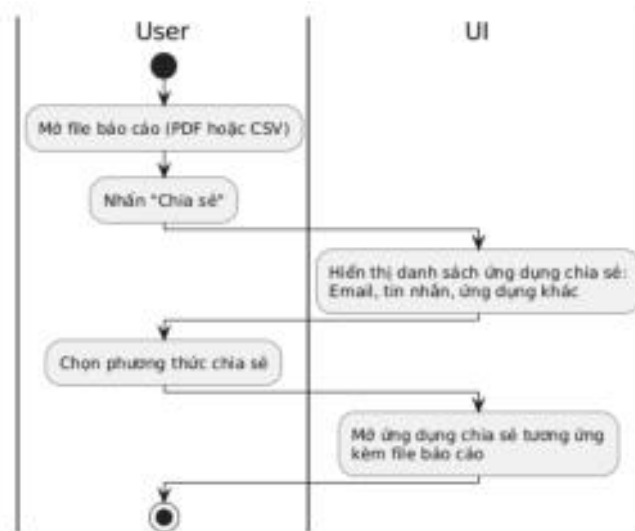
Hình 3.39 Sơ đồ hoạt động Xem báo cáo tổng quan



Hình 3.40 Sơ đồ hoạt động Xuất báo cáo PDF

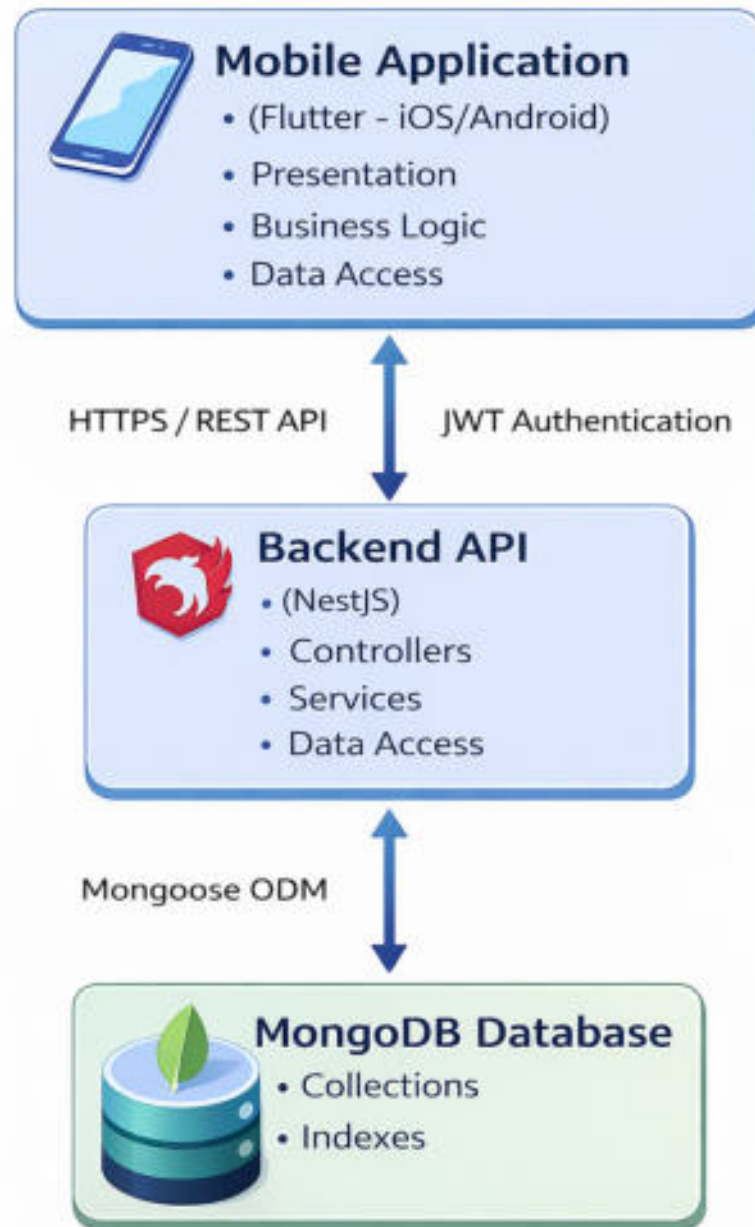


Hình 3.41 Sơ đồ hoạt động Xuất báo cáo CSV



Hình 3.42 Sơ đồ hoạt động Chia sẻ báo cáo

3.4 Kiến trúc hệ thống



Hình 3.43 Kiến trúc hệ thống

3.4.1 Client

Ứng dụng mobile được tổ chức theo Clean Architecture với 3 lớp:

- Presentation Layer:

- + Pages: Các màn hình UI (đăng nhập, đăng ký, mùa vụ, ..)
- + Widgets: Các component tái sử dụng
- + Themes: Cấu hình giao diện và màu sắc

- Data Layer:

- + Repositories: Xử lý truy cập dữ liệu từ API
- + Models: Định nghĩa cấu trúc dữ liệu
- + DTOs: Data Transfer Objects

- Core Layer:

- + ApiClients: Xử lý truy cập dữ liệu từ API
- + Models: Định nghĩa cấu trúc dữ liệu
- + DTOs: Data Transfer Objects

3.4.2 Server

Backend được xây dựng theo Layered Architecture với NestJS:

- Controller Layer:

- + Xử lý HTTP request/responses
- + Kiểm tra dữ liệu đầu vào (DTOs)
- + Bảo vệ xác thực (Authentication guards)
- + Tài liệu Swagger

- Service Layer:

- + Logic nghiệp vụ
- + Thao tác với database
- + Tích hợp dịch vụ bên ngoài
- + Chuyển đổi dữ liệu

- Data Access Layer:

- + Mongoose Schemes
- + Database Models
- + Quan hệ và Indexs

				hash
5	fullName	String	Required	Họ và tên
6	role	Enum (ADMIN/USER)	Default: USER	Vai trò người dùng
7	phone	String	Optional	Số điện thoại
8	otp	String	Optional	Mã OTP
9	otpExpires	Date	Optional	Hạn OTP
10	createdAt	Date	Auto	Ngày tạo
11	updatedAt	Date	Auto	Ngày cập nhật

Bảng 3.38 Mô tả cấu trúc dữ liệu User

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	id	ObjectId	Primary Key	Khóa chính
2	name	String	Required	Tên mùa vụ
3	startDate	Date	Required	Ngày bắt đầu
4	endDate	Date	Optional	Ngày kết thúc
5	ownerId	ObjectId	Ref: users, Required	Chủ sở hữu
6	members	Array	Optional	Danh sách thành viên
7	status	Enum (ACTIVE/COMPLETED)	Default: ACTIVE	Trạng thái
8	createdAt	Date	Auto	Ngày tạo
9	updatedAt	Date	Auto	Ngày cập nhật

Bảng 3.39 Mô tả cấu trúc dữ liệu Seasons

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	id	ObjectId	Primary Key	Khóa chính
2	seasonId	ObjectId	Ref: seasons, Required	Thuộc mùa vụ
3	creatorId	ObjectId	Ref: users, Required	Người tạo giao dịch
4	type	Enum (INCOME/EXPENSE)	Required	Loại giao dịch
5	category	String	Required	Danh mục giao dịch
6	amount	Number (>0)	Required	Số tiền
7	description	String	Optional	Mô tả
8	images	Array[String]	Optional	Hình ảnh

9	isVerified	Boolean	Default: false	Trạng thái phê duyệt
10	verifiedBy	ObjectId	Ref: users	Người phê duyệt
11	date	Date	Default: now	Ngày giao dịch
12	createdAt	Date	Auto	Ngày tạo
13	updatedAt	Date	Auto	Ngày cập nhật

Bảng 3.40 Mô tả cấu trúc dữ liệu Transactions

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	_id	ObjectId	Primary Key	Khóa chính
2	ownerId	ObjectId	Ref: users, Required	Chủ sở hữu
3	name	String	Required	Tên vật tư
4	quantity	Number (≥ 0)	Required	Số lượng
5	unit	String	Required	Đơn vị
6	expiryDate	Date	Optional	Hạn sử dụng
7	minThreshold	Number	Default: 0	Ngưỡng cảnh báo
8	image	String	Optional	Hình ảnh
9	createdAt	Date	Auto	Ngày tạo
10	updatedAt	Date	Auto	Ngày cập nhật

Bảng 3.41 Mô tả cấu trúc dữ liệu Inventory

STT	Trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	_id	ObjectId	Primary Key	Khóa chính
2	itemId	ObjectId	Ref: inventory, Required	Mục kho
3	type	Enum (IMPORT/EXPORT/ADJUSTMENT)	Required	Loại thao tác
4	amount	Number (> 0)	Required	Số lượng thay đổi
5	previousQuantity	Number	Required	Số lượng trước đó
6	newQuantity	Number	Required	Số lượng sau đó
7	reason	String	Optional	Lý do
8	seasonId	ObjectId	Ref: seasons	Mùa vụ liên

				quan
9	performedBy	ObjectId	Ref: users	Người thực hiện
10	createdAt	Date	Auto	Ngày tạo
11	updatedAt	Date	Auto	Ngày cập nhật

Bảng 3.42 Mô tả cấu trúc dữ liệu InventoryLogs

STT	Collection A	Collection B	Loại quan hệ	Trường tham chiếu	Mô tả
1	users	seasons	1 – N	seasons.ownerId → users._id	Một user sở hữu nhiều season
2	users	seasons.members	N – N	members.userId → users._id	User có thể là thành viên nhiều season
3	seasons	transactions	1 – N	transactions.seasonId → seasons._id	Một season có nhiều giao dịch
4	users	transactions (creator)	1 – N	transactions.creatorId → users._id	User tạo nhiều giao dịch
5	users	transactions (verifiedBy)	1 – N	transactions.verifiedBy → users._id	User phê duyệt nhiều giao dịch
6	users	inventory	1 – N	inventory.ownerId → users._id	User sở hữu nhiều vật tư
7	inventory	inventorylogs	1 – N	inventorylogs.itemId → inventory._id	Item kho có nhiều log
8	seasons	inventorylogs	1 – N	inventorylogs.seasonId → seasons._id	Season liên quan nhiều log
9	users	inventorylogs	1 – N	inventorylogs.performedBy → users._id	User thực hiện nhiều thao tác kho

Bảng 3.43 Mô tả quan hệ giữa các collection

3.6 Kết chương

Qua chương này, chúng ta đã tiến hành phân tích và thiết kế hệ thống, chỉ ra được các tác nhân và ca sử dụng trong hệ thống, và trình bày một số sơ đồ để chỉ ra luồng hoạt động và cách các thực thể tương tác với nhau trong hệ thống.

Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Triển khai và kết quả hệ thống

4.1.1 Triển khai hệ thống

Kiến trúc tổng thể: Hệ thống được xây dựng theo mô hình Client-Server với kiến trúc phân tầng rõ ràng

- Frontend (Mobile Application): Ứng dụng di động được phát triển bằng Flutter/Dart, áp dụng Clean Architecture với pattern BLoC (Business Logic Component) để quản lý state. Ứng dụng hỗ trợ đa nền tảng IOS và Android từ một codebase duy nhất.
- Backend API: Hệ thống backend được xây dựng bằng NestJS/TypeScript, áp dụng Layered Architecture với các module độc lập (Auth, Seasons, Transactions, Inventory, Reports). API được thiết kế theo chuẩn RESTful với Swagger documentation đầy đủ.
- Database: Sử dụng MongoDB 4.4+ làm cơ sở dữ liệu NoSQL, phù hợp với cấu trúc dữ liệu linh hoạt của hệ thống. Dữ liệu được tổ chức thành các collection: users, seasons, transactions, inventory, inventorylogs.
- Authentication: Hệ thống xác thực sử dụng JWT (JSON Web Token).

4.1.2 Triển khai Backend API

- Công nghệ và framework:
 - NestJS 11.0+ với TypeScript 5.7+
 - Mongoose ODM để tương tác với MongoDB
 - Passport JWT cho xác thực
 - Class-validator và Class-transformer cho validation
 - Logger của NestJS cho logging (HTTP request/response)
 - Nodemailer cho gửi email OTP
 - Swagger/OpenAPI cho API documentation
- Cấu trúc module:
 - Users Module: Quản lý thông tin người dùng
 - Auth Module: Xử lý đăng ký, đăng nhập, quên mật khẩu với OTP email (thời hạn 5 phút), quản lý JWT token
 - Seasons Module: Quản lý mùa vụ, thành viên, phân quyền (OWNER, EDITOR, VIEWER)
 - Transactions Module: Quản lý giao dịch thu/chi, phê duyệt giao dịch, lọc và tìm kiếm
 - Inventory Module: Quản lý kho hàng, vật liệu, lịch sử nhập/xuất kho, cảnh báo
 - Reports Module: Tạo báo cáo tổng quan, thống kê, xuất PDF/CSV
 - Mail Module: Gửi email OTP và thông báo

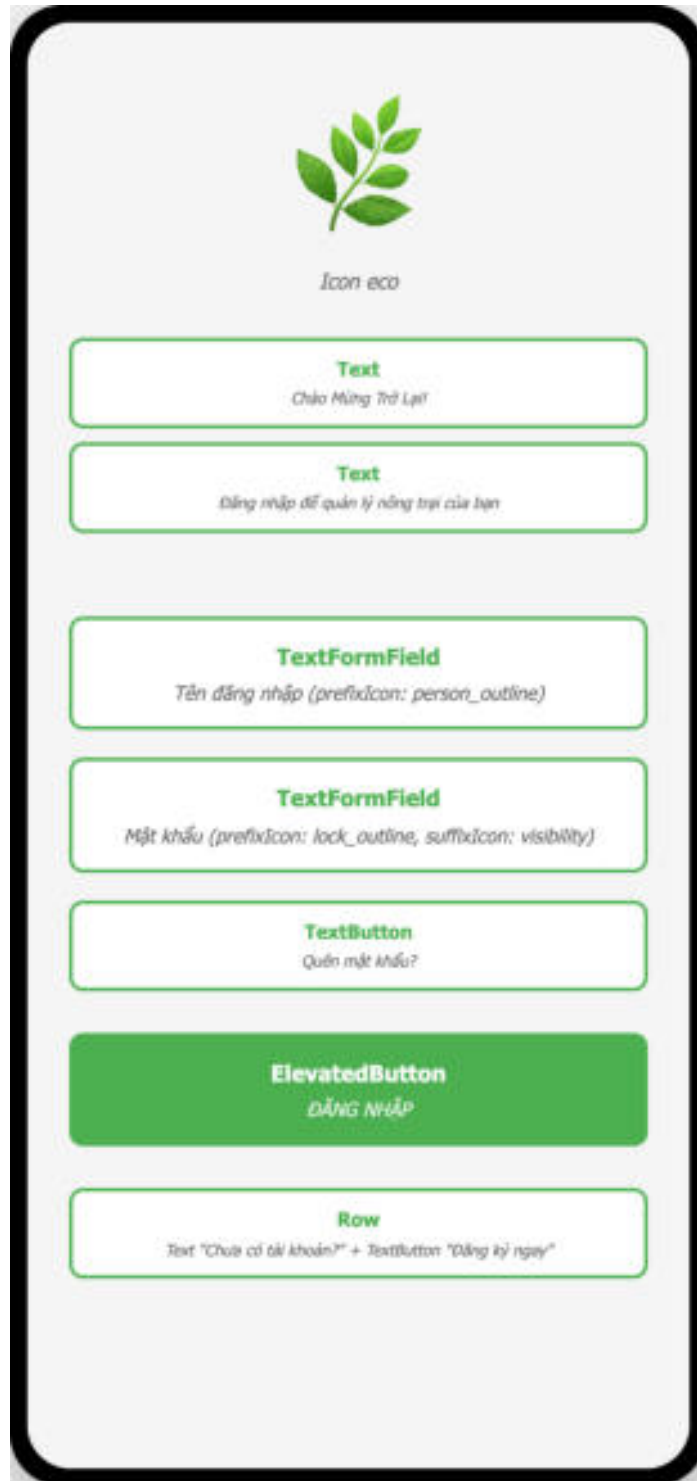
4.1.3 Triển khai ứng dụng mobile

- Công nghệ và dependences:
 - Flutter SDK 3.0+ với Dart - flutter_bloc 8.1+ cho state management
 - dio 5.0+ cho HTTP client với interceptors
 - go_router 10.0+ cho navigation
 - get_it 7.6+ cho dependency injection
 - fl_chart 0.63+ cho biểu đồ thống kê –
 - camera và image_picker cho chụp/chọn ảnh

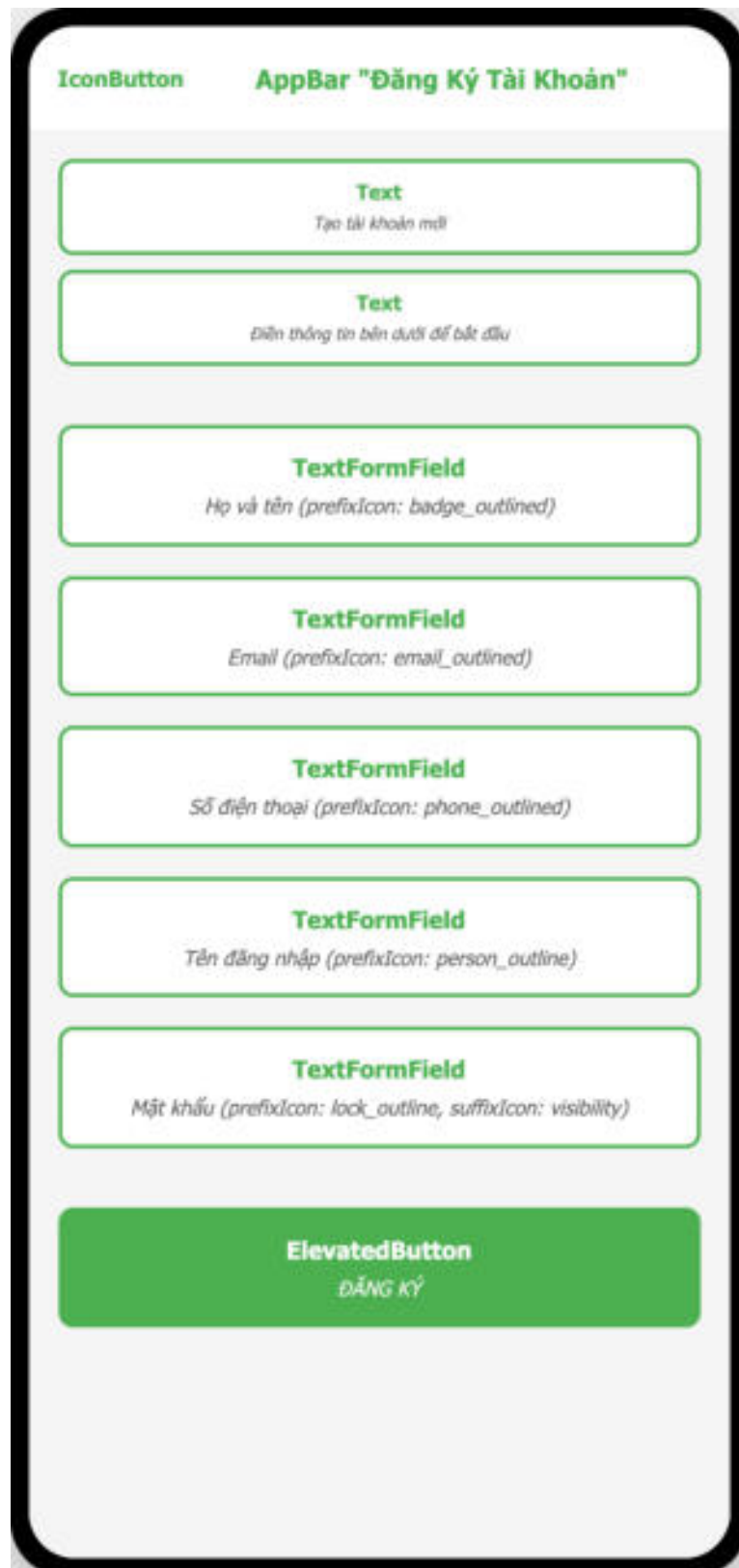
- pdf và printing cho xuất báo cáo PDF
 - csv và share_plus cho xuất và chia sẻ file
- Phác thảo giao diện:



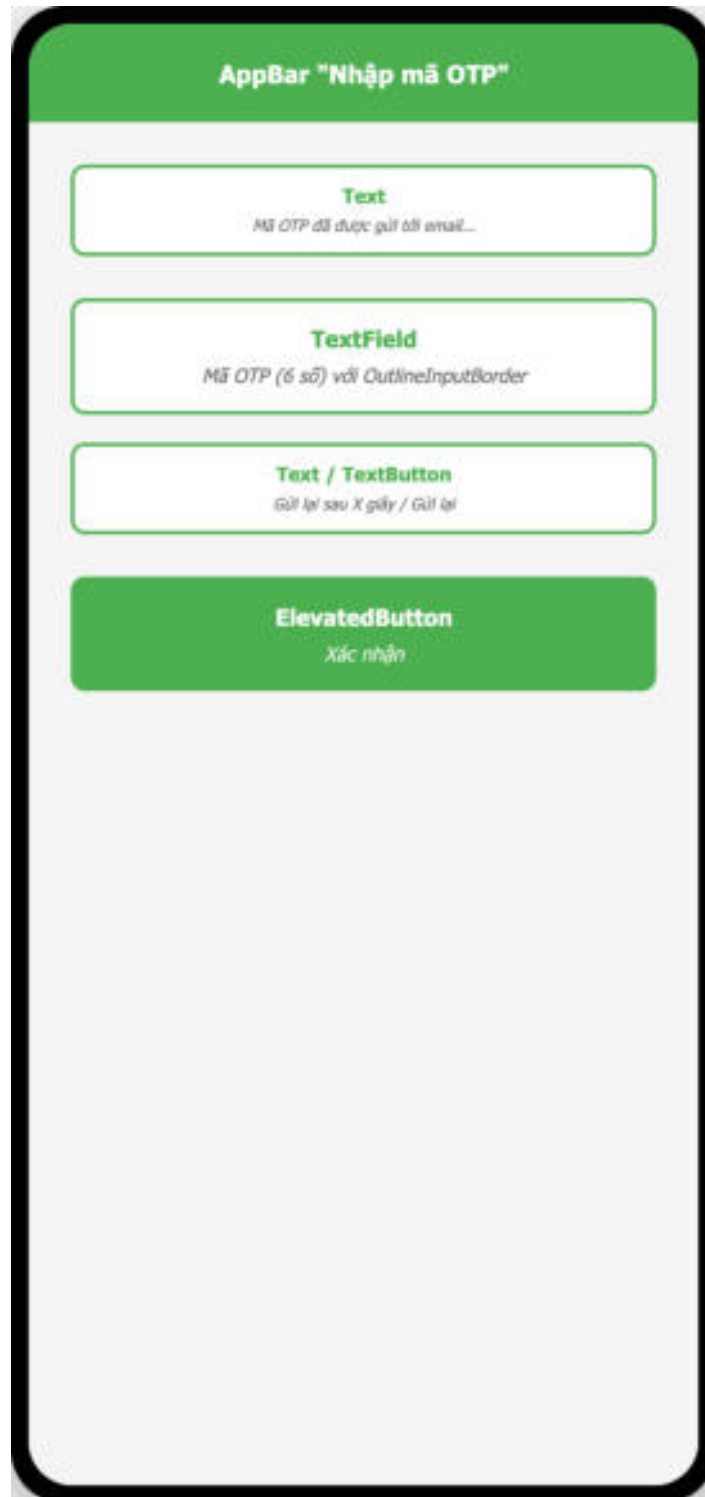
Hình 4.1 Màn hình Splash



Hình 4.2 Màn hình Đăng nhập



Hình 4.3 Màn hình Đăng ký



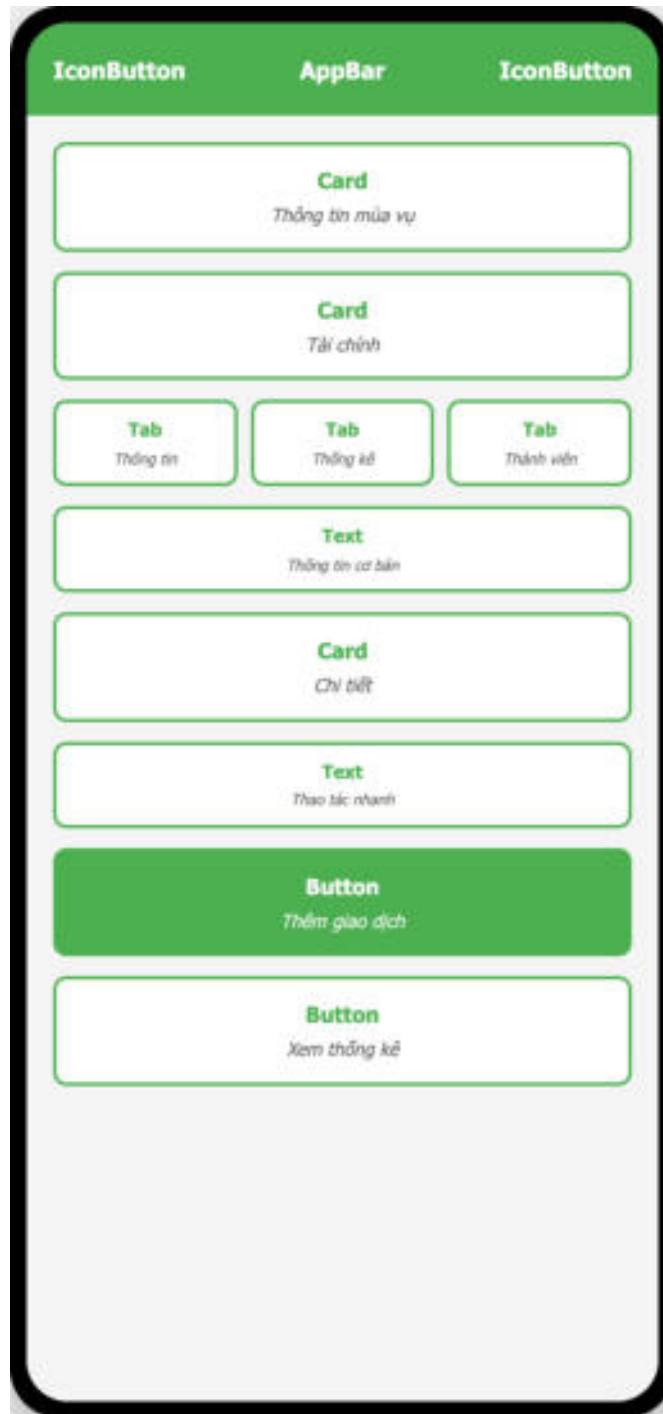
Hình 4.4 Màn hình Nhập mã OTP



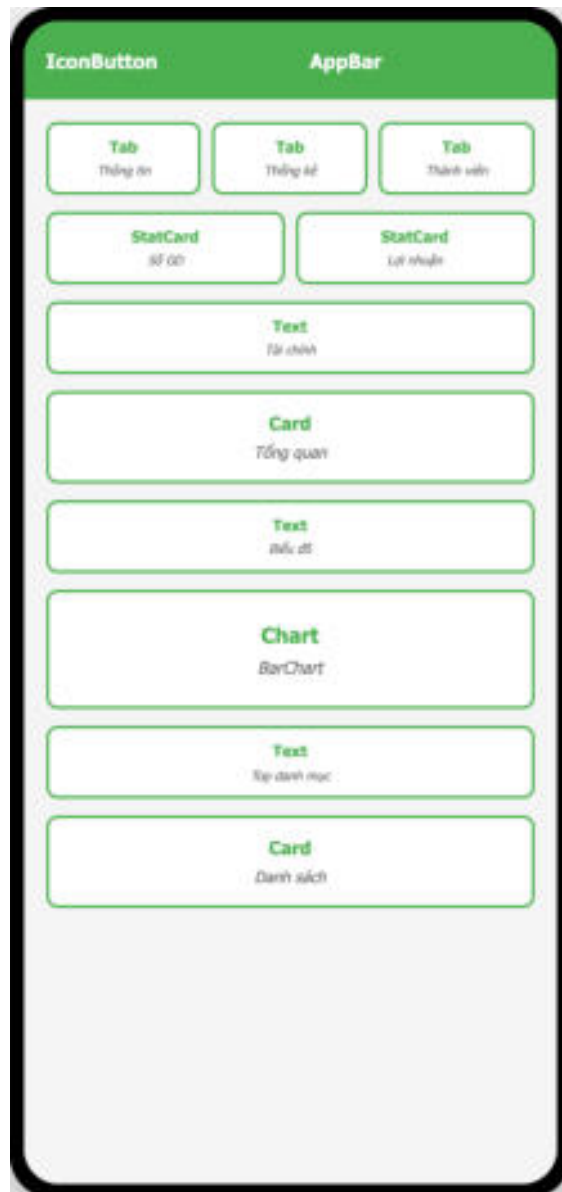
Hình 4.5 Màn hình Trang chủ



Hình 4.6 Màn hình Danh sách mùa vụ



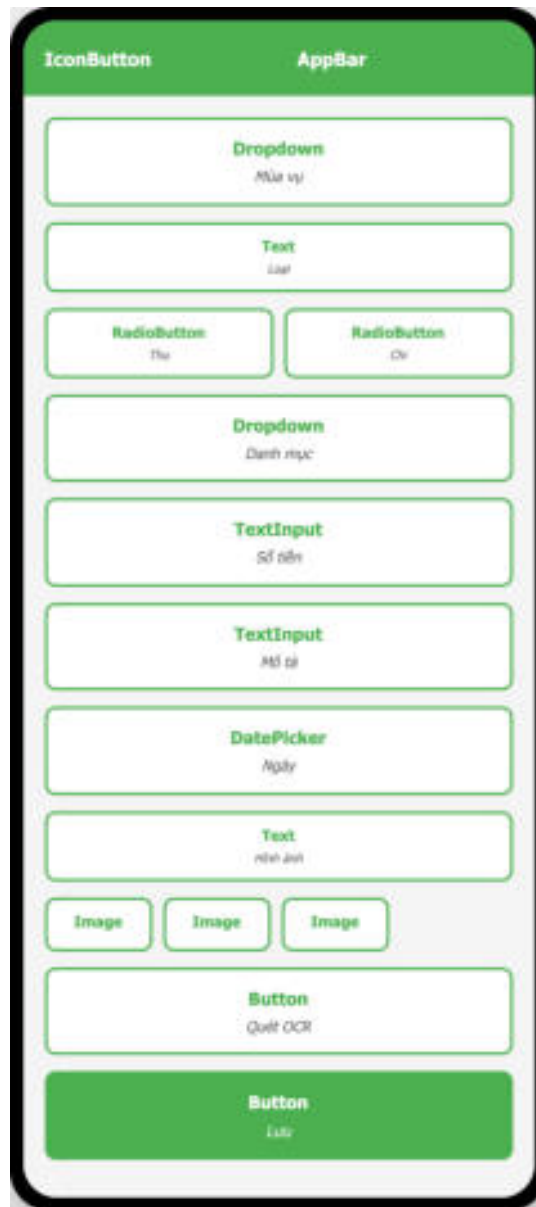
Hình 4.7 Màn hình Chi tiết mùa vụ



Hình 4.8 Màn hình Thống kê



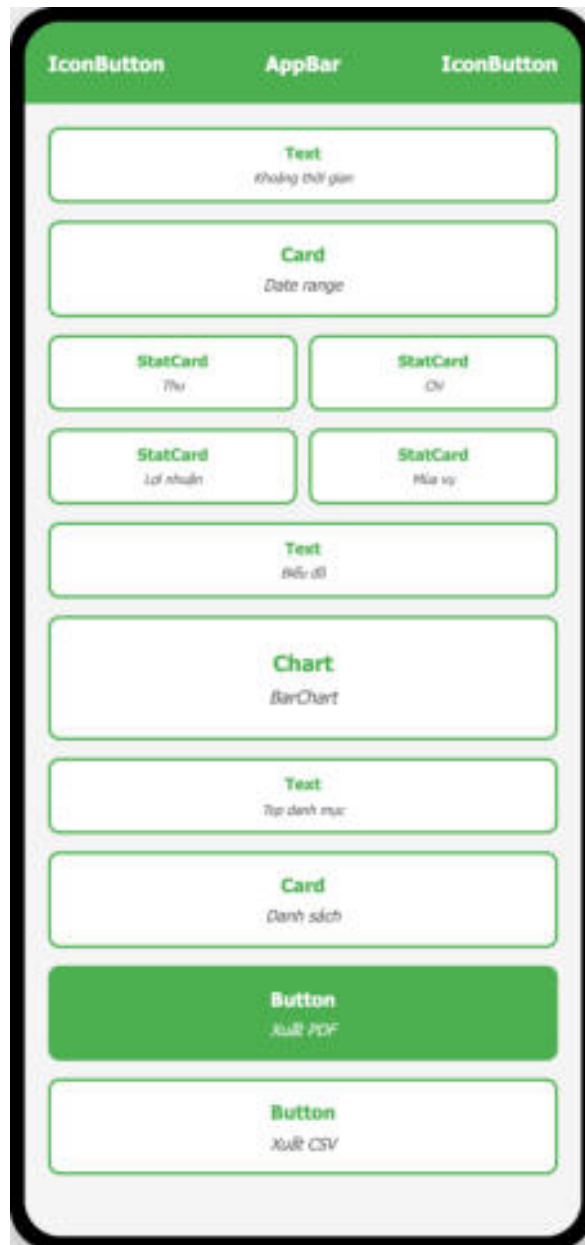
Hình 4.9 Màn hình Danh sách giao dịch



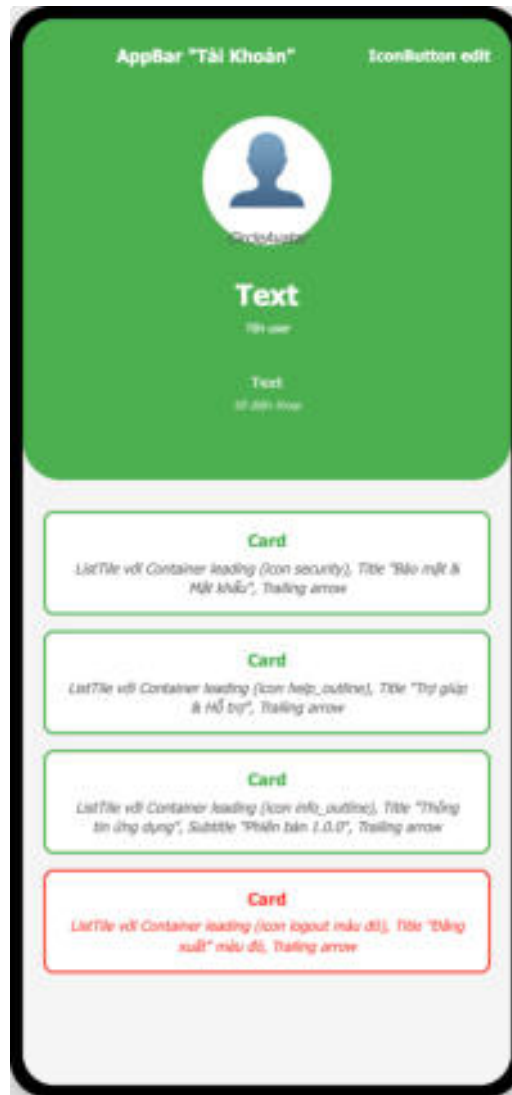
Hình 4.10 Màn hình Tạo giao dịch



Hình 4.11 Màn hình Danh sách vật liệu



Hình 4.12 Màn hình Báo cáo tổng quan



Hình 4.13 Màn hình Xem profile

4.2 Kết chương

Chương này đã trình bày rõ về việc triển khai ứng dụng trên di động và kết quả triển khai.

KẾT LUẬN

1. Kết luận

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề án đã đạt được những kết quả sau:

Về mặt lý thuyết, đề án đã áp dụng các công nghệ hiện đại như Flutter cho phát triển ứng dụng di động đa nền tảng, NestJS cho xây dựng RESTful API, MongoDB cho quản lý dữ liệu NoSQL, và AI để tự động hóa việc nhập liệu. Đề án cũng áp dụng các kiến trúc phần mềm hiện đại như Clean Architecture cho mobile app và Layered Architecture cho backend, giúp code dễ bảo trì và mở rộng.

Về mặt triển khai thực tế, hệ thống đã được triển khai thành công với ứng dụng di động có giao diện thân thiện và dễ sử dụng, đáp ứng đầy đủ các chức năng quản lý mùa vụ, quản lý giao dịch tài chính, quản lý kho hàng, báo cáo thống kê và tích hợp AI. Hệ thống đã được kiểm thử và hoạt động ổn định trên nền tảng IOS.

Tuy nhiên, đề án vẫn còn tồn tại một số hạn chế như sau:

- Hệ thống chưa hỗ trợ chế độ offline, người dùng cần có kết nối internet để sử dụng các chức năng chính.
- Chưa có tính năng thông báo đẩy (push notification) để nhắc nhở người dùng về các giao dịch cần phê duyệt hoặc cảnh báo kho hàng.
- Hệ thống chưa có ứng dụng web, chỉ hỗ trợ trên nền tảng di động, gây khó khăn khi người dùng muốn quản lý từ máy tính.
- Chưa có tính năng đồng bộ dữ liệu giữa nhiều thiết bị, người dùng chỉ có thể truy cập từ một thiết bị tại một thời điểm.

2. Hướng phát triển

- Phát triển ứng dụng web

- Xây dựng ứng dụng web để người dùng có thể quản lý từ máy tính, tạo điều kiện thuận tiện hơn cho việc nhập liệu và xem báo cáo trên màn hình lớn.
- Phát triển dashboard web với nhiều biểu đồ và báo cáo chi tiết hơn, hỗ trợ in ấn và xuất file dễ dàng.

- Hỗ trợ chế độ offline

- Tích hợp local database (SQLite) cho ứng dụng di động để người dùng có thể sử dụng một số chức năng cơ bản khi không có kết nối internet.
- Tự động đồng bộ dữ liệu khi có kết nối internet trở lại, đảm bảo tính nhất quán của dữ liệu.

- Thêm các tính năng thông báo

- Tích hợp push notification để thông báo khi có giao dịch mới cần phê duyệt, cảnh báo khi số lượng kho hàng thấp, hoặc nhắc nhở các công việc quan trọng.
- Gửi email thông báo định kỳ về báo cáo tài chính và tình trạng mùa vụ.

- Thêm các tính năng cộng tác nâng cao

- Thêm tính năng chat giữa các thành viên trong mùa vụ để trao đổi thông tin.
- Cho phép comment trên các giao dịch để giải thích hoặc thảo luận.
- Tạo activity feed để theo dõi các hoạt động gần đây trong mùa vụ.

TÀI LIỆU THAM KHẢO

- [1] TOPDev, [RESTful API là gì? Cách thiết kế RESTful API, 2021.](#)
- [2] NestJS, Documentation, [Documentation | NestJS - A progressive Node.js framework](#)
- [3] [NestJS Documentation – Using MongoDB with NestJS](#)
- [4] [NestJS Recipes – MongoDB \(Mongoose\) Guide](#)
- [5] [Flutter Official Documentation – Flutter Developer Docs](#)
- [6] [Dart Official Documentation – Language & SDK Docs](#)
- [7] [Pub.dev – Flutter/Dart Package Repository](#)