

**Kiểm thử phần mềm**

**Hệ thống bán cây**

**Sinh viên thực hiện: Nguyễn Nhật Hải - 3122411046**

**Nguyễn Thành Đạt - 3122411039**

**Nguyễn Thành Việt - 3122411245**

**Nguyễn Thành Trung Hiếu - 3122411056**

**Giảng viên hướng dẫn: TS.Đỗ Như Tài**

**Thành phố Hồ Chí Minh, tháng 11, 2023**

# Mục lục

Contents

[Mục lục 2](#_Toc210858980)

[1. Business requirements analysis for ornamental plant sales website 1](#_Toc210858981)

[1.1 Bussiness context 1](#_Toc210858982)

[1.2 Mô tả dự án 3](#_Toc210858983)

[1.3 Yêu cầu nghiệp vụ 3](#_Toc210858984)

[1.3.1 Yêu cầu phi chức năng 3](#_Toc210858985)

[1.3.2 Yêu cầu chức năng 3](#_Toc210858986)

[1.4 Sơ đồ quy trình nghiệp vụ 3](#_Toc210858987)

[2. Usecase view 3](#_Toc210858988)

[**2.1. Sơ đồ tổng quan các chức năng của hệ thống** 3](#_Toc210858989)

[**2.1.1. User Stories** 29](#_Toc210858990)

[**Product Catalog (Danh mục sản phẩm)** 29](#_Toc210858991)

[**Shopping Cart (Giỏ hàng)** 29](#_Toc210858992)

[**Payment Process (Quy trình thanh toán)** 30](#_Toc210858993)

[**Order (Đơn hàng)** 31](#_Toc210858994)

[**Inventory (Kho hàng)** 32](#_Toc210858995)

[**Access Control (Kiểm soát truy cập)** 32](#_Toc210858996)

[2.2. Domain 33](#_Toc210858997)

[2.2.1. Mô tả 33](#_Toc210858998)

[Miền nghiệp vụ được lựa chọn 33](#_Toc210858999)

[Lý do chọn miền nghiệp vụ này 33](#_Toc210859000)

[Các miền con (Subdomain) và thực thể nghiệp vụ chính 34](#_Toc210859001)

[1. Kiểm soát truy cập (Access Control) 34](#_Toc210859002)

[2. Danh mục sản phẩm (Product Catalog) 34](#_Toc210859003)

[3. Giỏ hàng (Shopping Cart) 35](#_Toc210859004)

[4. Quy trình thanh toán (Checkout Process) 35](#_Toc210859005)

[5. Đơn hàng (Orders) 35](#_Toc210859006)

[6. Quản lý kho (Inventory) 36](#_Toc210859007)

[Mối quan hệ giữa các miền 36](#_Toc210859008)

[Kiến trúc của miền nghiệp vụ 36](#_Toc210859009)

[2.2.2 Conceptual Model 37](#_Toc210859010)

[Mô tả sơ đồ 37](#_Toc210859011)

[1️ Access Control 37](#_Toc210859012)

[2️ Product Catalog 37](#_Toc210859013)

[3️ Inventory 38](#_Toc210859014)

[4️ Shopping Cart 38](#_Toc210859015)

[5️ Checkout Process 38](#_Toc210859016)

[6️ Orders 38](#_Toc210859017)

[2.3. Kiến trúc hệ thống 38](#_Toc210859018)

[Decomposition View: C4 39](#_Toc210859019)

[C1 – SystemContext 39](#_Toc210859020)

[C2-Container 41](#_Toc210859021)

[C3-Component(High-Level) 44](#_Toc210859022)

[C3-Component(Model Level) 47](#_Toc210859023)

[C4- Code/Implementation Diagram: 50](#_Toc210859024)

[Deployment view 51](#_Toc210859025)

[Block Diagram 53](#_Toc210859026)

[CommunicationView 57](#_Toc210859027)

[3. Test plan 59](#_Toc210859028)

[3.1. Introduction 59](#_Toc210859029)

[3.1.1. Purpose 59](#_Toc210859030)

[3.1.2. Definitions, Acronyms, and Abbreviations 59](#_Toc210859031)

[3.1.3. References 59](#_Toc210859032)

[3.1.4. Background information 60](#_Toc210859033)

[3.1.5. Scope of testing 60](#_Toc210859034)

[3.1.6. Constraints 60](#_Toc210859035)

[3.1.7. Risk list 60](#_Toc210859036)

[3.1.8. Training needs 61](#_Toc210859037)

[3.2. Requirements for test 61](#_Toc210859038)

[3.2.1. Test Items 61](#_Toc210859039)

[3.2.2. Acceptance Test Criteria 62](#_Toc210859040)

[3.3. Feature to be tested 62](#_Toc210859041)

[3.3.1. Functionality 62](#_Toc210859042)

[3.3.2. Usability 65](#_Toc210859043)

[3.3.3. Design Constraints 66](#_Toc210859044)

[3.3.4. Interfaces 66](#_Toc210859045)

[3.4. Feature not to be tested 66](#_Toc210859046)

[3.4.1. Functional Features 67](#_Toc210859047)

[3.4.2. Non-functional Features 67](#_Toc210859048)

[3.5. Test Strategy 67](#_Toc210859049)

[3.5.1. Test types 67](#_Toc210859050)

[3.6. Test stages 70](#_Toc210859051)

[3.7. Resource 71](#_Toc210859052)

[3.7.1. Human Resource 71](#_Toc210859053)

[3.7.2. Test Management 72](#_Toc210859054)

[3.8. Test Environment 72](#_Toc210859055)

[3.8.1. Hardware 72](#_Toc210859056)

[3.8.2. Software 72](#_Toc210859057)

[3.8.3. Infrastructure 73](#_Toc210859058)

[3.9. Test Milestone 74](#_Toc210859059)

[3.10. Deliverables 74](#_Toc210859060)

[Hình 1 . Sơ đồ tổng quan các chức năng của hệ thống 6](#_Toc210858346)

[Hình 2. Các tác nhân tham gia hệ thống 7](#_Toc210858347)

[Hình 3. Sơ đồ phân rã chức năng kiểm soát người dùng 8](#_Toc210858348)

[Hình 4 . Sơ đồ phân rã chức năng danh mục sản phẩm 9](#_Toc210858349)

[Hình 5. Sơ đồ phân rã chức năng Giỏ hàng 10](#_Toc210858350)

[Hình 6. Sơ đồ phân rã chức năng quy trình thanh toán 10](#_Toc210858351)

[Hình 7. Sơ đồ phân rã chức năng Hóa đơn 11](#_Toc210858352)

[Hình 8. Sơ đồ phân rã chức năng Kho 11](#_Toc210858353)

[Hình 9. Conceptual Model 36](#_Toc210858354)

[Hình 10. C1 – SystemContext 38](#_Toc210858355)

[Hình 11. C2-Container 40](#_Toc210858356)

[Hình 12. C3-Component(High-Level) 43](#_Toc210858357)

[Hình 14. C3-Component(Model Level) 46](#_Toc210858358)

[Hình 15. C4- Code/Implementation Diagram 49](#_Toc210858359)

[Hình 16. Deployment view 50](#_Toc210858360)

[Hình 17. Block Diagram 53](#_Toc210858361)

[Hình 18. CommunicationView 57](#_Toc210858362)

# 1. Business requirements analysis for ornamental plant sales website

## 1.1 Bussiness context

The MOW Garden website has a basic business script that includes Product Catalog, Shopping Cart, Checkout Process, Inventory, and Access Control.

Danh mục sản phẩm (Product Catalog): Người mua có thể xem danh sách sản phẩm với các tùy chọn lọc và sắp xếp theo loại và giá. Người mua có thể xem chi tiết sản phẩm bằng cách nhấp vào sản phẩm trong danh sách. Ở trang chi tiết, họ có thể thấy tên sản phẩm, mô tả, hình ảnh thực tế, giá và số lượng tồn kho. Quản trị hệ thống (Admin) có thể quản lý sản phẩm, thêm sản phẩm mới, cập nhật sản phẩm hiện có, xóa sản phẩm và sau khi thêm sản phẩm thì phân bổ sản phẩm vào kho hiện tại.

Giỏ hàng (Shopping Cart): Người mua có thể mua sản phẩm từ danh sách thông qua nút “Mua” hoặc Mua trực tiếp từ trang chi tiết sản phẩm. Sau khi thêm sản phẩm, người mua có thể xem giỏ hàng với thông tin tóm tắt các sản phẩm đã thêm như số lượng sản phẩm, tên sản phẩm, hình ảnh và giá sản phẩm, phí vận chuyển và tổng giá trị đơn hàng. Bất cứ khi nào người mua thêm hoặc xóa sản phẩm khỏi giỏ hàng, giỏ hàng sẽ được cập nhật tự động. Sau đó, người mua có thể tiến hành thanh toán bằng cách nhấp nút “Thanh toán” trên trang giỏ hàng.

Quy trình thanh toán (Checkout Process): Sau khi người mua nhấp vào nút “Thanh toán”, hệ thống sẽ bắt đầu kiểm tra đơn hàng hiển thị thông tin tổng giỏ hàng và thông tin giao hàng gồm tên,địa chỉ nhận hàng, email, số điện thoại và cung cấp các lựa chọn thanh toán như Thanh toán khi nhận hàng (COD) hoặc chuyển khoản ngân hàng. Khi việc thanh toán được xác nhận, hệ thống sẽ tạo hóa đơn đặt hàng lưu vào đơn hàng của người mua và gửi thông báo xác nhận đến người mua qua tài khoản người dùng đồng thời cập nhật lại số lượng sản phẩm trong kho.

Quy trình đơn hàng(Order): Người mua có thể xem lại đơn hàng của mình sau khi thanh toán với thông tin cơ bản tên hàng, số lượng hàng, giá tiền, tiến trình giao hàng.Người mua có thể xem chi tiết bằng cách nhấp vào Xem chi tiết hệ thống hiển thị Mã chi tiết đơn hàng, thông tin giao hàng,Tổng tiền, Danh sách sản phẩm đã đặt, Trạng thái đơn hàng. Admin có thể theo dõi tất cả đơn hàng của người dùng và có khả năng kiểm tra trạng thái đơn hàng.

Quản lý kho (Inventory): Quản trị viên có thể quản lý kho hàng. Quản trị viên có thể nhập kho sản phẩm từ danh sách cây. Quản trị viên cập nhật số lượng sản phẩm theo từng loại danh mục cây có sẵn. Quản trị viên xóa sản phẩm trong kho.Khi người mua thanh toán thành công sẽ cập nhật số lượng tồn kho luôn nhất quán với tình trạng thực tế.

Kiểm soát truy cập (Access Control): Người mua hoặc Quản trị viên có thể đăng nhập/đăng xuất hệ thống. Nếu là người mua phải đăng kí nếu chưa có tài khoản, đăng kí thành công thì họ sẽ được chuyển hướng đến trang danh mục sản phẩm hoặc đăng nhập thành công nếu đã có tài khoản. Nếu là Quản trị viên, họ sẽ được chuyển hướng đến trang quản trị, Quản trị viên có thể khóa và mở người dùng được quyền đăng nhập.

## 1.2 Mô tả dự án

## 1.3 Yêu cầu nghiệp vụ

### 1.3.1 Yêu cầu phi chức năng

### 1.3.2 Yêu cầu chức năng

## 1.4 Sơ đồ quy trình nghiệp vụ

## 2. Usecase view

## **2.1. Sơ đồ tổng quan các chức năng của hệ thống**

A diagram of a network

Description automatically generated

Hình 1 . Sơ đồ tổng quan các chức năng của hệ thống

**Các tác nhân tham gia hệ thống**

A group of people with text

Description automatically generated

Hình 2. Các tác nhân tham gia hệ thống

- Buyer: Tìm sản phẩm, thêm vào giỏ, thanh toán và theo dõi đơn của **chính mình.**

**- Admin:** Vận hành hệ thống: sản phẩm, kho, đơn, người dùng.

- Payment Provider: Xử lý giao dịch thanh toán không dùng tiền mặt.

**Các chức năng chính**

- Access Control: Chức năng này cho phép người dùng tự đăng ký tài khoản (Buyer). Người dùng (Buyer/Admin) đăng nhập và được chuyển hướng vào đúng khu vực (Catalog hoặc Trang quản trị), người dùng đăng xuất an toàn, Admin có thể khóa/mở tài khoản người dùng.

- Product Catalog: Chức năng này cho phép Buyer duyệt danh mục, lọc theo loại hoặc giá, xem chi tiết sản phẩm (tên, mô tả, ảnh, giá, tồn kho). Admin được quyền Thêm/Cập nhật/Xóa và Phân bổ sản phẩm vào kho.

- Shopping Cart: Chức năng này cho phép Buyer thêm sản phẩm vào giỏ từ Catalog hoặc Trang chi tiết, Xem giỏ với thông tin tóm tắt, cập nhật số lượng sản phẩm hoặc xóa sản phẩm hệ thống tự tính lại tổng tiền (đơn giá, phí vận chuyển, tổng giá trị đơn) và điều hướng tới Checkout.

- Payment Process: Chức năng này cho phép Buyer hoàn tất đơn hàng, nhập và xác nhận đơn hàng, chọn phương thức thanh toán (COD/Chuyển khoản), xác nhận thanh toán. Hệ thống tạo hóa đơn và gắn vào Order, trừ tồn kho theo Item và gửi thông báo xác nhận.

- Order: Chức năng này cho phép (Buyer) xem danh sách đơn của mình, tra cứu chi tiết đơn (mã đơn, thông tin giao hàng, tổng tiền, danh sách sản phẩm, trạng thái). (Admin) xem tất cả đơn của người mua, giám sát và cập nhật trạng thái.

- Inventory: Chức năng này cho phép Admin quản lý số lượng tồn kho như Nhập kho sản phẩm, Cập nhật số lượng theo danh mục hoặc từng sản phẩm, Xóa sản phẩm khỏi kho.

**Sơ đồ phân rã các chức năng cảu hệ thống**

**-** Kiểm soát người dùng

**A diagram of a process

Description automatically generated**

Hình 3. Sơ đồ phân rã chức năng kiểm soát người dùng

- Danh mục sản phẩm

A diagram of a diagram

Description automatically generated

Hình 4 . Sơ đồ phân rã chức năng danh mục sản phẩm

- Giỏ hàng

A diagram of a diagram

Description automatically generated

Hình 5. Sơ đồ phân rã chức năng Giỏ hàng

- Quy trình thanh toán

A diagram of a diagram

Description automatically generated

Hình 6. Sơ đồ phân rã chức năng quy trình thanh toán

- Hóa đơn

A diagram of a flowchart

Description automatically generated

Hình 7. Sơ đồ phân rã chức năng Hóa đơn

- Kho

**A diagram of a diagram

Description automatically generated**

Hình 8. Sơ đồ phân rã chức năng Kho

|  |  |  |  |
| --- | --- | --- | --- |
| UC 1 | | | Access Control |
| Mô tả | | | Chức năng này cho phép người dùng tự đăng ký tài khoản (Buyer). Người dùng (Buyer/Admin) đăng nhập và được chuyển hướng vào đúng khu vực (Catalog hoặc Trang quản trị), người dùng đăng xuất an toàn, Admin có thể khóa/mở tài khoản người dùng. |
| Tác nhân | | Chính | Buyer, Admin |
| Phụ | Hệ thống |
| Tiền điều kiện | | | - Hệ thống hoạt động bình thường.  - Với đăng nhập tài khoản đã tồn tại.  - Với thao tác khóa/mở người thao tác phải là Admin và đã đăng nhập. |
| Hậu điều kiện | Thành công | | - Người dùng được chuyển hướng đúng trang theo vai trò  - Trạng thái tài khoản được cập nhật khi Admin khóa/mở |
| Lỗi | | - Đăng ký: dữ liệu không hợp lệ thông báo lỗi từng trường, giữ lại dữ liệu đã nhập hợp lệ hoặc email đã tồn tại hiển thị email đã dược đăng ký.  - Đăng nhập: tài khoản không tồn tại hiển thị không tìm thấy tài khoản. Sai email/mật khẩu hiển thị email hoặc mật khẩu không đúng, tài khoản bị khóa hiển thị Tài khoản đã bị khóa, liên hệ Admin.  - Khóa/mở không tìm thấy người dùng hiển thị không tìm thấy người dùng. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| Chức năng bắt đầu khi người dùng truy cập hệ thống  - Đăng ký: (Buyer)  + Buyer chọn đăng ký nhập thông tin: Họ tên, Email, Mật Khẩu, Tích chấp nhận điều khoản  + Hệ thống kiểm tra hợp lệ (email đúng định dạng, mật khẩu đạt tiêu chí)  + Hệ thống tạo User mới với role = Buyer, status = Active và lưu  + Hệ thống thông báo thành công, chuyển hướng vào trang catalog  - Đăng nhập: (Buyer/Admin)  + Người dùng nhập Email + Mật khẩu  + Hệ thống tìm tài khoản theo email  + Hệ thống kiểm tra mật khẩu  + Tài khoản tồn tại, xác định vai trò  Nếu Buyer: Kiểm tra status (Lock thì báo tài khoản bị khóa còn Open thì chuyển hướng đến trang Catalog)  Nếu Admin thì chuyển hướng đến trang quản trị  - Đăng xuất:  + Buyer/Admin chọn Đăng xuất  + Hệ thống ghi nhận và chuyển về trang Đăng nhập  - Khóa/Mở tài khoản (Admin)  + Admin vào Trang quản trị → Quản lý người dùng  + Admin chọn người dùng  + Nếu người dùng đang Active → Admin chọn Khóa; nếu đang Locked → Admin chọn Mở  + Hệ thống cập nhập status tương ứng, ghi nhận và hiển thị kết quả | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UC 2 | | | Product Catalog |
| Mô tả | | | Chức năng này cho phép Buyer duyệt danh mục, lọc theo loại hoặc giá, xem chi tiết sản phẩm (tên, mô tả, ảnh, giá, tồn kho). Admin được quyền Thêm/Cập nhật/Xóa và Phân bổ sản phẩm vào kho. |
| Tác nhân | | Chính | Buyer, Admin |
| Phụ | Hệ thống |
| Tiền điều kiện | | | - Hệ thống hoạt động bình thường, dữ liệu danh mục sản phẩm đã tồn tại.  - Admin được quyền thao tác khi đã đăng nhập. |
| Hậu điều kiện | Thành công | | - Buyer xem được danh sách và chi tiết theo bộ lọc.  - Sản phẩm được thêm/sửa/xóa đúng quy tắc, sản phẩm được phân bổ vào kho tương ứng.  - Hệ thống ghi nhận hành động của Admin. |
| Lỗi | | - Buyer không tìm thấy sản phẩm.  - Thêm/Cập nhật dữ liệu không hợp lệ. Thông báo lỗi từng trường, không lưu.  - Xóa không cho phép vì sản phẩm đã có đơn hàng hoặc tồn kho dương.  - Không tìm thấy kho để phân bổ sản phẩm. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| Chức năng này bắt đầu khi (Buyer) muốn tìm hoặc xem sản phẩm (Admin) muốn thêm, sửa, xóa và phân bổ sản phẩm.  - Buyer duyệt danh mục và xem chi tiết  + Buyer chọn bộ lọc (loại sản phẩm, khoảng giá)  + Hệ thống áp dụng bộ lọc → trả về danh sách sản phẩm  + Buyer nhấp vào một sản phẩm  + Hệ thống hiển thị Chi tiết sản phẩm: Tên, Mô tả, Ảnh, Giá, Tồn kho  - Admin Quản lý sản phẩm  + Chọn một trong các hành động  \* Thêm sản phẩm mới  Nhập thông tin: Tên,  Hệ thống kiểm tra hợp lệ → lưu sản phẩm  Admin phân bổ sản phẩm vào kho: chọn kho, số lượng, hệ thống cập nhật bản ghi tồn kho  \* Cập nhật sản phẩm  Chọn sản phẩm → sửa các trường cần thiết → lưu  \* Xóa sản phẩm  Chọn sản phẩm cần xóa, hệ thống kiểm tra ràng buộc (đã có đơn hàng hoặc tồn kho thông báo không thể xóa) nếu đạt ẩn khỏi Catalog. | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UC 3 | | | Shopping Cart |
| Mô tả | | | Chức năng này cho phép Buyer thêm sản phẩm vào giỏ từ Catalog hoặc Trang chi tiết, Xem giỏ với thông tin tóm tắt, cập nhật số lượng sản phẩm hoặc xóa sản phẩm hệ thống tự tính lại tổng tiền (đơn giá, phí vận chuyển, tổng giá trị đơn) và điều hướng tới Checkout. |
| Tác nhân | | Chính | Buyer |
| Phụ | Hệ thống |
| Tiền điều kiện | | | - Hệ thống hoạt động bình thường, sản phẩm sở trạng thái Active.  - Với giỏ của người dùng đăng nhập: tài khoản hợp lệ.  - Định nghĩa công thức tính tổng (đơn giá, ship) |
| Hậu điều kiện | Thành công | | - Giỏ hàng phản ánh chính xác các mục và số lượng người dùng chọn.  - Tổng tiền được tự động tính lại khi Thêm/Xóa/Cập nhật.  - Khi người dùng “Thanh toán” hệ thống chuyển tới Checkout với dữ liệu giỏ đã khóa tạm (để xác thực lại ở bước đặt hàng). |
| Lỗi | | - Hết hàng: Khi thêm hoặc cập nhật số lượng vượt tồn hệ thống hiển thị số lượng vượt quá quy định và tự điều chỉnh về mức tối đa cho phép. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| Chức năng này bắt đầu khi buyer muốn mua hàng:  - Thêm hoặc Mua sản phẩm  + Buyer ở Danh sách hoặc Trang chi tiết.  + Buyer chọn Mua chuyển hướng đến trang giỏ hàng.  + Buyer chọn Số lượng (mặc định là 1) → nhấn Thêm vào giỏ.  + Hệ thống kiểm tra tồn kho tối thiểu và tối đa.  + Hệ thống tự động tính lại tổng và hiển thị thông báo đã thêm.  - Xem giỏ hàng và thông tin tóm tắt  + Buyer mở Giỏ hàng.  + Hệ thống hiển thị danh sách mục (ảnh, tên, giá, số lượng) tạm tính phí vận chuyển ước tính, tổng đơn.  - Cập nhật số lượng hoặc Xóa sản phẩm  + Buyer sửa số lượng hoặc nhấn Xóa sản phẩm.  + Hệ thống kiểm tra min/max, tồn kho.  + Hệ thống cập nhật giỏ và tự tính lại tổng.  - Nhấn Thanh toán (đi tới Checkout)  + Buyer nhấn Thanh toán.  + Hệ thống chụp trạng thái giỏ hiện tại, xác thực lần cuối (giá/tồn), rồi chuyển tới Checkout. | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UC 4 | | | Payment Process |
| Mô tả | | | Chức năng này cho phép Buyer hoàn tất đơn hàng, nhập và xác nhận đơn hàng, chọn phương thức thanh toán (COD/Chuyển khoản), xác nhận thanh toán. Hệ thống tạo hóa đơn và gắn vào Order, trừ tồn kho theo Item và gửi thông báo xác nhận. |
| Tác nhân | | Chính | Buyer |
| Phụ | Hệ thống, Payment Provider |
| Tiền điều kiện | | | - Buyer đi từ Checkout (giỏ đã hợp lệ).  - Phương thức thanh toán được cấu hình hoạt động.  - Chính sách tồn kho đã xác định thời điểm trừ tồn sau khi Order được xác nhận. |
| Hậu điều kiện | Thành công | | - Đơn hàng ở trạng thái phù hợp.  - Inventory được trừ theo từng Item trong Order và có lịch sử biến động.  - Buyer nhận thông báo xác nhận |
| Lỗi | | - Thông tin giao hàng không hợp lệ thông báo lỗi yêu cầu nhập lại.  - Hết hàng hoặc giá bị thay đổi lúc chụp đơn hiển thị khác biệt và yêu cầu Buyer xác nhận lại.  - Thanh toán thất bại cho Buyer chọn lại phương thức.  - Trừ tồn thất bại thông báo lỗi. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| Chức năng bắt đầu khi Buyer muốn Thanh toán:  - Kiểm tra và chụp đơn  + Hệ thống chụp giỏ, tạo Order, tính lại giá/ship/tổng đơn.  + Nếu phát hiện hết hàng hoặc giá thay đổi hiển thị cho Buyer xác nhận lại.  - Thông tin giao hàng  + Buyer nhập và xác nhận tên, địa chỉ, email, SĐT.  + Hệ thống xác nhận dịnh dạng.  - Chọn phương thức  + Buyer chọn COD hoặc Chuyển khoản.  - Xác nhận thanh toán.  + COD  \* Hệ thống tạo hóa đơn (Pending/COD).  \* Trừ tồn kho theo từng Item, ghi nhận biến động.  \* Gửi thông báo xác nhận cho Buyer.  + Chuyển khoản  \* Hệ thống tạo yêu cầu thanh toán và khởi tạo phiên thanh toán. | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UC 5 | | | Order |
| Mô tả | | | Chức năng này cho phép (Buyer) xem danh sách đơn của mình, tra cứu chi tiết đơn (mã đơn, thông tin giao hàng, tổng tiền, danh sách sản phẩm, trạng thái). (Admin) xem tất cả đơn của người mua, giám sát và cập nhật trạng thái. |
| Tác nhân | | Chính | Buyer, Admin |
| Phụ | Hệ thống |
| Tiền điều kiện | | | - Người dùng đăng nhập (Buyer xem đơn của chính mình, Admin có quyền quản trị).  - Đơn hàng đã được tạo từ Checkout. |
| Hậu điều kiện | Thành công | | - Buyer xem được danh sách, chi tiết chính xác.  - Admin cập nhật trạng thái đúng quy tắc, hệ thống ghi nhật ký kiểm tra và gửi thông báo. |
| Lỗi | | - Buyer không thể truy cập đơn hàng thông báo lỗi.  - Chuyển trạng thái không hợp lệ sai tiến trình thông báo từ chối và hiển thị tiến trình hợp lệ. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| - Chức năng này bắt đầu khi Buyer muốn xem lại đơn hàng hoặc Admin muốn xem đơn của người mua và theo dõi tiến trình đơn.  - Buyer xem danh sách hoặc chi tiết đơn hàng  + Buyer mở Đơn hàng của tôi.  + Hệ thống hiển thị danh sách đơn.  + Buyer nhấn Xem chi tiết một đơn.  + Hệ thống hiển thị chi tiết đơn gồm: Mã đơn, Thông tin giao hàng, Tổng tiền, Danh sách sản phẩm, trạng thái hiện tại.  - Admin quản lý và giám sát  + Admin chọn Quản lý đơn hàng.  + Hệ thống hiển thị toàn bộ đơn có thông tin cơ bản Trạng thái, Ngày, Khách.  + Admin chọn đơn để xem chi tiết.  + Admin chọn cập nhật trạng thái theo quy tắc: Processing, Out for delivery, Deveried.  + Hệ thống cập nhật trạng thái và ghi nhận. | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UC 6 | | | Inventory |
| Mô tả | | | Chức năng này cho phép Admin quản lý số lượng tồn kho như Nhập kho sản phẩm, Cập nhật số lượng theo danh mục hoặc từng sản phẩm, Xóa sản phẩm khỏi kho. |
| Tác nhân | | Chính | Admin |
| Phụ | Hệ thống |
| Tiền điều kiện | | | - Người thao tác đã đăng nhập với vai trò Admin.  - Product, Warehouse tồn tại và ở trạng thái Active.  - Quy tắc trừ tồn đã được xác định thống nhất với chức năng Order/Payment. |
| Hậu điều kiện | Thành công | | - Số lượng tồn kho theo Product, warehouse được cập nhật chính xác.  - UI phản ánh đúng tồn hiện tại. |
| Lỗi | | - Số lượng nhập không hợp lệ bé hơn 0 hiển thị lỗi.  - Xóa khỏi kho không hợp lệ vì tồn lớn hơn 0 hoặc có đơn đang treo. |
| Đặc tả chức năng | | | |
| Luồng sự kiện chính | | | |
| - Chức năng bắt đầu khi Admin muốn Quản lý kho.  - Nhập kho sản phẩm  + Admin mở Quản lý kho và chọn Nhập kho.  + Chọn kho và sản phẩm, nhập số lượng.  + Hệ thống kiểm tra hợp lệ (sản phẩm/kho active, số lượng > 0) và cộng dồn vào product, warehouse.  + UI hiển thị tồn mới và thông báo thành công.  - Cập nhật số lượng theo danh mục  + Admin chọn danh mục và sản phẩm cần điều chỉnh.  + Nhập số lượng điều chỉnh.  + Hệ thống kiểm tra không cho tồn âm.  + UI cập nhật tồn mới.  - Xóa sản phẩm khỏi kho  + Admin chọn kho và sản phẩm xong bấm xóa.  + Hệ thống kiểm tra chỉ cho xóa khi số lượng sản phẩm trong kho bằng 0 và không có giao dịch đang treo.  + UI ẩn dòng khỏi danh sách kho. | | | |
| Sơ đồ hành động | | | |
|  | | | |
| Giao diện minh họa | | | |
|  | | | |

**2.1.1. User Stories**

**Product Catalog (Danh mục sản phẩm)**

**Là một Người mua (Buyer)**, tôi muốn xem danh sách sản phẩm trên trang chủ, có thể **lọc theo danh mục** hoặc **mức giá**, danh sách sản phẩm sẽ **thu hẹp lại** và chỉ hiển thị các sản phẩm phù hợp.

**Là một Người mua**, tôi muốn **chuyển đến trang chi tiết sản phẩm** để xem các thông tin cơ bản như: **tên, mô tả, giá, và số lượng tồn kho tại từng cửa hàng/kho**.

**Là một Quản trị viên hệ thống (SysAdmin)**, tôi muốn **quản lý sản phẩm** (thực hiện các thao tác CRUD – Thêm, Xem, Sửa, Xóa) và **gán kho hàng hiện có (warehouse/bin)** cho từng sản phẩm.

**Shopping Cart (Giỏ hàng)**

**Là một Người mua**, tôi muốn **thêm bất kỳ sản phẩm nào từ trang danh mục** vào giỏ hàng (mặc định số lượng = 1).

**Là một Người mua**, tôi muốn **thêm sản phẩm từ trang chi tiết sản phẩm** vào giỏ hàng (mặc định số lượng = 1).

**Là một Người mua**, tôi muốn **xem các sản phẩm vừa thêm vào giỏ hàng** và **một bảng tóm tắt**, bao gồm: **tổng chi phí giỏ hàng, tổng phụ (subtotal), phí vận chuyển (ước tính)** và **tổng giá trị đơn hàng**.

**Là một Người mua**, tôi muốn **cập nhật số lượng** của từng sản phẩm trong giỏ.

Khi số lượng thay đổi, **bảng tóm tắt** cần được **cập nhật lại tương ứng**.

**Là một Người mua**, tôi muốn **xóa bất kỳ sản phẩm nào** khỏi giỏ hàng.

Khi xóa sản phẩm, **bảng tóm tắt** cũng cần được **cập nhật lại**.

**Là một Người mua**, tôi muốn **thanh toán (checkout)** giỏ hàng của mình.

Khi số lượng sản phẩm trong giỏ hàng = 0, **quá trình thanh toán không được phép diễn ra**.

Khi giỏ hàng được thanh toán, **quá trình thanh toán** sẽ bắt đầu.

**Payment Process (Quy trình thanh toán)**

**Bất kỳ Người mua nào** cũng có thể **thanh toán** trong quá trình **Checkout**.

Khi bắt đầu quá trình thanh toán:

Hệ thống **kiểm tra tính hợp lệ của sản phẩm** (giá / tình trạng tồn kho).

Thực hiện **xử lý thanh toán** (bằng **COD** hoặc **cổng thanh toán/PSP / ngân hàng**).

Sau đó **gửi thông báo/xác nhận qua email** cho Người mua.

Nếu có sản phẩm **không hợp lệ** (hết hàng hoặc giá thay đổi), quá trình thanh toán sẽ **bị hủy** và **người mua được thông báo** để xác nhận lại.

Nếu thanh toán **thành công**, hệ thống sẽ:

Đánh dấu giỏ hàng là **đã thanh toán**

**Tạo hóa đơn (invoice)** đính kèm đơn hàng

**Trừ tồn kho** theo từng sản phẩm

**Gửi xác nhận thanh toán** cho Người mua

Nếu thanh toán **thất bại hoặc bị hủy**, hệ thống sẽ:

Đánh dấu thanh toán là **thất bại**

Cho phép Người mua **chọn phương thức khác**.

**Order (Đơn hàng)**

**Là một Người mua**, tôi muốn xem **danh sách đơn hàng của mình**, có thể **lọc theo trạng thái và khoảng thời gian**.

**Là một Người mua**, tôi muốn **xem chi tiết đơn hàng**, bao gồm:

Mã đơn hàng

Thông tin giao hàng

Danh sách sản phẩm

Tổng phụ / phí vận chuyển / thuế / tổng cộng

Trạng thái hiện tại và lịch sử thay đổi trạng thái

**Là một Quản trị viên hệ thống**, tôi muốn xem **tất cả đơn hàng**, có **bộ lọc mạnh mẽ** (theo trạng thái, thời gian, người mua, kênh bán).

**Là một Quản trị viên hệ thống**, tôi muốn **cập nhật trạng thái đơn hàng** theo **luồng hợp lệ**:  
Processing->out for deliveried-> delivery

Khi trạng thái đơn hàng thay đổi, hệ thống phải:

Ghi lại **AuditLog** (lịch sử thao tác)

(Tùy chọn) Gửi **thông báo** cho Người mua.

**Inventory (Kho hàng)**

**Là một Quản trị viên hệ thống**, tôi muốn **nhập sản phẩm (IN)** vào **kho được chọn**, từ **danh mục hoặc danh sách cây**, kèm theo **số lượng** và **lý do nhập**.

**Là một Quản trị viên hệ thống**, tôi muốn **điều chỉnh số lượng tồn (±)** cho sản phẩm, theo **danh mục** hoặc **theo từng mặt hàng**, kèm **lý do** (kiểm kê, hư hỏng, chuyển kho).

**Là một Quản trị viên hệ thống**, tôi muốn **xóa sản phẩm khỏi kho** khi sản phẩm **không còn được lưu trữ** ở đó (**chỉ khi số lượng tồn = 0 và không có đơn giữ hàng**).

Khi tồn kho thay đổi, hệ thống phải **tạo bản ghi InventoryMovement** (biến động kho) với:

Lý do (IN / OUT / ADJUST / REMOVE)

Tham chiếu (ref)

Người thực hiện (actor)  
và cập nhật số lượng tồn một cách **đồng thời, chính xác (atomically)**.

Khi đơn hàng được **xác nhận hoặc thanh toán** theo chính sách, hệ thống sẽ **tạo biến động OUT (xuất kho)** cho từng sản phẩm trong đơn.

**Access Control (Kiểm soát truy cập)**

Mỗi **Buyer / SysAdmin** đều là **một người dùng (User)**.

**Là một Buyer hoặc SysAdmin**, tôi muốn **đăng nhập vào hệ thống**.

Khi **Buyer đăng nhập**, họ được chuyển đến **trang Danh mục sản phẩm**.

Khi **SysAdmin đăng nhập**, họ được chuyển đến **trang Quản trị hệ thống**.

**Là một Buyer hoặc SysAdmin**, tôi muốn **đăng xuất khỏi hệ thống**.

**Là một SysAdmin**, tôi muốn có thể **khóa / mở khóa tài khoản người dùng** theo chính sách.

## 2.2. Domain

## 2.2.1. Mô tả

**Định nghĩa (Definition)**

**Miền nghiệp vụ (Domain)** – Là phạm vi kiến thức, hoạt động hoặc lĩnh vực mà phần mềm được xây dựng để phục vụ.  
(Theo Domain-Driven Design Reference, Eric Evans)

## **Miền nghiệp vụ được lựa chọn**

Miền nghiệp vụ của dự án này là **Thương mại điện tử – Mua bán cây cảnh trực tuyến (Online Plant Shop Domain)**.  
Hệ thống cho phép người mua chọn và đặt cây cảnh trực tuyến, thanh toán, theo dõi đơn hàng,  
trong khi quản trị viên có thể quản lý danh mục cây, sản phẩm, kho hàng và người dùng.

## **Lý do chọn miền nghiệp vụ này**

1. **Gần gũi và dễ hiểu:** Việc mua bán cây cảnh trực tuyến là hoạt động phổ biến, dễ hình dung và áp dụng quy trình thương mại điện tử chuẩn.
2. **Phù hợp học thuật:** Miền này bao gồm các quy tắc nghiệp vụ thật như kiểm tra tồn kho, xác nhận thanh toán, quản lý danh mục, giúp sinh viên hiểu rõ thiết kế hệ thống thương mại điện tử.
3. **Độ phức tạp vừa phải:** Ngoài CRUD còn có logic kiểm soát truy cập, kiểm tra kho, cập nhật đơn hàng — đủ để thể hiện mô hình domain-driven design.
4. **Khả năng mở rộng:** Có thể phát triển thành hệ thống đa miền (micro-domain) hoặc tích hợp dịch vụ thanh toán, vận chuyển trong tương lai.

## **Các miền con (Subdomain) và thực thể nghiệp vụ chính**

### **1. Kiểm soát truy cập (Access Control)**

* **Thực thể:** User, Admin
* **Mô tả:**  
  Quản lý việc đăng ký, đăng nhập, khóa/mở người dùng và phân quyền.
  + Người mua (User) có thể đăng ký, đăng nhập, chọn cây, đặt hàng.
  + Quản trị viên (Admin) có thể quản lý người dùng, sản phẩm, đơn hàng, và kho cây.
  + Người dùng có trạng thái unlock (được phép đăng nhập) hoặc lock (bị cấm đăng nhập).
* **Luật nghiệp vụ:**
  + **A1:** Người dùng bị lock không được phép đăng nhập.
  + **A2:** Quản trị viên có thể khóa hoặc mở người dùng.
  + **A3:** Quản trị viên có toàn quyền quản lý hệ thống.

### **2. Danh mục sản phẩm (Product Catalog)**

* **Thực thể:** Category, Plant (Product)
* **Mô tả:**  
  Danh mục chứa các loại cây cảnh (ví dụ: cây trong nhà, cây thủy sinh, cây phong thủy...).  
  Mỗi cây cảnh có tên, mô tả, hình ảnh, giá và danh mục tương ứng.  
  Người mua có thể xem danh sách cây, lọc và xem chi tiết từng loại cây.  
  Quản trị viên có thể thêm, chỉnh sửa, xóa danh mục và sản phẩm.
* **Luật nghiệp vụ:**
  + **C1:** Tên cây không được trùng trong cùng một danh mục.
  + **C2:** Khi thêm sản phẩm mới, phải phân loại cây vào danh mục có sẵn.

### **3. Giỏ hàng (Shopping Cart)**

* **Thực thể:** Cart, CartItem (lưu trong User.cartData)
* **Mô tả:**  
  Người mua có thể thêm cây vào giỏ, thay đổi số lượng, hoặc xóa cây khỏi giỏ.  
  Hệ thống tự động tính tổng tạm tính và phí vận chuyển.
* **Luật nghiệp vụ:**
  + **S1:** Giỏ hàng luôn lưu đúng thông tin sản phẩm (tên, hình, giá, số lượng).
  + **S2:** Tổng tiền giỏ hàng = tổng giá cây + phí giao hàng.

### **4. Quy trình thanh toán (Checkout Process)**

* **Thực thể:** OrderSummary, Address, PaymentMethod
* **Mô tả:**  
  Khi người mua nhấn “Thanh toán”, hệ thống hiển thị tóm tắt đơn hàng, địa chỉ nhận cây, và phương thức thanh toán (COD hoặc Stripe).  
  Nếu thanh toán thành công, hệ thống tạo đơn hàng mới, trừ kho, và gửi thông báo xác nhận.
* **Luật nghiệp vụ:**
  + **CH1:** Bắt buộc nhập đầy đủ thông tin người nhận (tên, địa chỉ, số điện thoại, email).
  + **CH2:** Với COD, đơn hàng được xem là thanh toán ngay.
  + **CH3:** Với Stripe, đơn hàng chỉ được xác nhận khi thanh toán thành công.

### **5. Đơn hàng (Orders)**

* **Thực thể:** Order, OrderItem, Address (nhúng)
* **Mô tả:**  
  Sau khi thanh toán, đơn hàng được tạo với danh sách cây, giá, số lượng, phí giao hàng, tổng tiền, trạng thái và thông tin người nhận.  
  Người dùng có thể xem danh sách đơn hàng của mình và xem chi tiết từng đơn.  
  Admin có thể theo dõi và thay đổi trạng thái đơn.
* **Luật nghiệp vụ:**
  + **O1:** Tổng tiền = tổng tạm tính + phí giao hàng.
  + **O2:** Mỗi đơn hàng lưu lại thông tin cây (tên, giá, số lượng) tại thời điểm đặt.
  + **O3:** Trạng thái đơn: Processing, Shipping, Delivered, Cancelled.
  + **O4:** Nếu đơn đã thanh toán bị hủy, tồn kho được hoàn lại.

### **6. Quản lý kho (Inventory)**

* **Thực thể:** Stock
* **Mô tả:**  
  Lưu trữ và theo dõi số lượng tồn kho của từng loại cây cảnh.  
  Khi người mua đặt hàng, hệ thống tự động trừ số lượng cây trong kho.  
  Admin có thể nhập cây mới hoặc cập nhật số lượng.
* **Luật nghiệp vụ:**
  + **I1:** Tồn kho không được âm.
  + **I2:** Phải kiểm tra tồn kho trước khi cho phép thanh toán.
  + **I3:** Khi thanh toán thành công, tồn kho bị trừ tương ứng.
  + **I4:** Nếu đơn đã thanh toán bị hủy, tồn kho được hoàn lại.

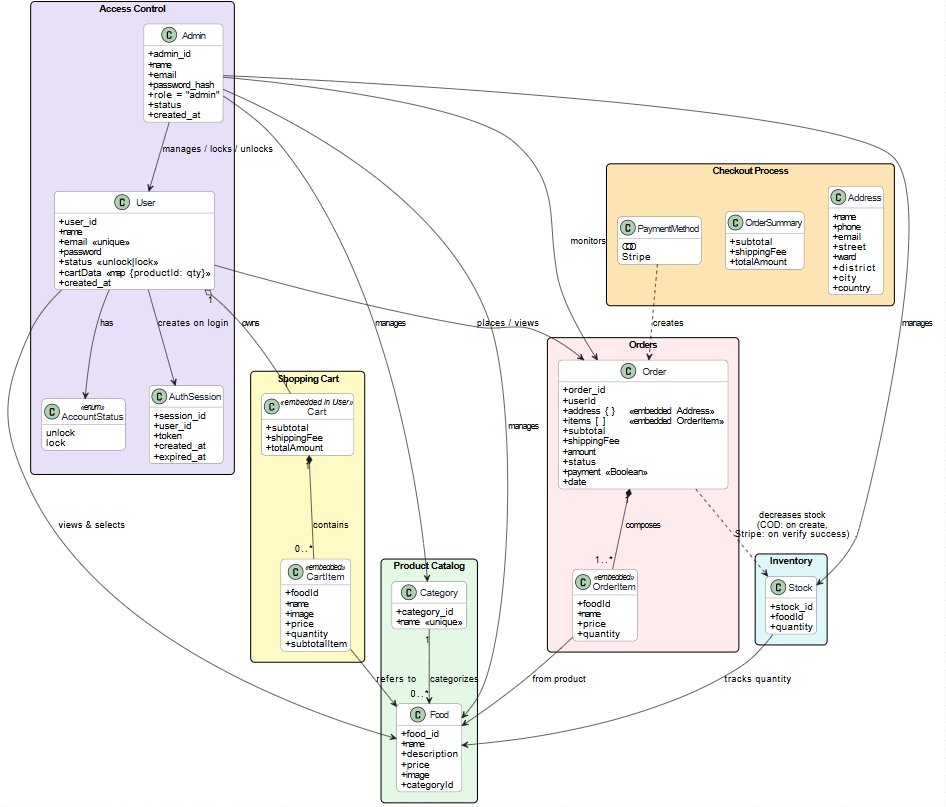
### **Mối quan hệ giữa các miền**

* **Access Control** kiểm soát quyền đăng nhập và hoạt động của người dùng.
* **Product Catalog** cung cấp dữ liệu cây cảnh để người mua duyệt và chọn.
* **Shopping Cart** là bước trung gian giữa danh mục và thanh toán.
* **Checkout Process** tạo **Order**, sau đó cập nhật **Inventory**.
* **Admin** có quyền theo dõi, quản lý, và điều chỉnh toàn bộ quy trình.

### **Kiến trúc của miền nghiệp vụ**

Hệ thống được xây dựng theo **kiến trúc đơn khối (Monolithic)** nhưng tổ chức theo hướng **phân tách miền (Micro-domain)**.  
Mỗi phần (User, Product, Cart, Order, Stock) là một miền nghiệp vụ riêng biệt, có logic độc lập nhưng cùng nằm trong một ứng dụng backend duy nhất.

## 2.2.2 Conceptual Model



Hình 9. Conceptual Model

## Mô tả sơ đồ

### **1️ Access Control**

* **User**: có thể đăng ký, đăng nhập, thêm vào giỏ, thanh toán, xem lịch sử đơn.
* **Admin → User (locks/unlocks)**: Admin có quyền **khóa hoặc mở** tài khoản user (trạng thái status = "lock" | "unlock").  
  Khi **bị khóa**, user **không thể đăng nhập** hay **đặt hàng**.
* **Admin → Category/Product (manages)**: Admin CRUD danh mục & sản phẩm.

### **2️ Product Catalog**

* **Category → Product**: mỗi danh mục có nhiều sản phẩm (1-nhiều).
* **User → Product (views & selects)**: người dùng duyệt, lọc, xem chi tiết, thêm vào giỏ.

### **3️ Inventory**

* **Stock → Product**: mỗi sản phẩm có một bản ghi tồn kho riêng (foodId, quantity).
* **Admin → Stock (manages)**: Admin cập nhật thủ công khi nhập hàng.
* **Order → Stock (reduces stock on payment)**: khi thanh toán thành công thì trừ kho.

### **4️ Shopping Cart**

* **User → Cart (owns)**: mỗi user có **1 giỏ hàng** (lưu trong user.cartData).
* **Cart → CartItem (contains)**: giỏ chứa nhiều dòng sản phẩm.
* **CartItem → Product (refers to)**: mỗi dòng trỏ đến sản phẩm gốc.
* Gồm các giá trị tính toán như subtotal, shippingFee, totalAmount.

### **5️ Checkout Process**

* **Cart → Address (proceeds to checkout)**: người dùng điền thông tin nhận hàng.
* **Address → OrderSummary (summarizes)**: hệ thống tổng hợp subtotal, shippingFee, totalAmount.
* **OrderSummary → PaymentMethod (chooses payment)**: chọn COD hoặc Stripe.

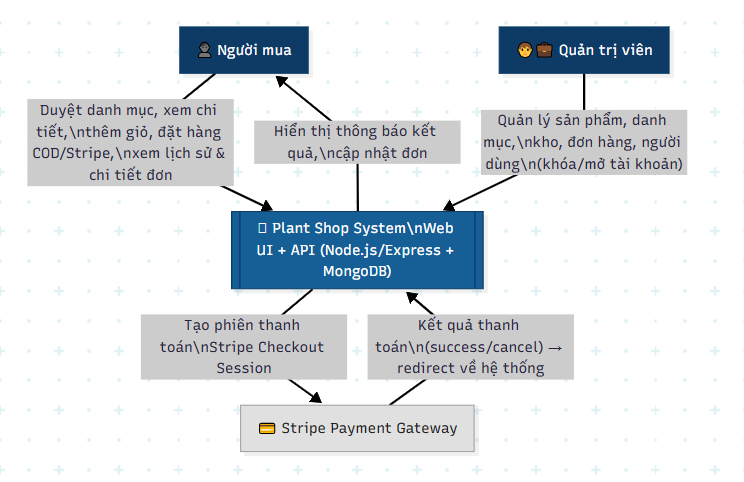
### **6️ Orders**

* **PaymentMethod → Order (creates)**: khi xác nhận thanh toán → tạo order.
* **Order → OrderItem (includes)**: mỗi đơn có nhiều dòng hàng snapshot (name, price, quantity).
* **User → Order (views/orders)**: người dùng xem danh sách và chi tiết đơn.
* **Admin → Order (monitors & updates)**: Admin xem và đổi trạng thái đơn.

# 2.3. Kiến trúc hệ thống

## Decomposition View: C4

### C1 – SystemContext



Hình 10. C1 – SystemContext

**Giải thích chi tiết từng phần**

👤 **Người mua (Buyer)**

* Là người dùng cuối ngoài hệ thống.
* Tương tác thông qua giao diện web (React UI).
* Các hành vi chính:
  + Xem danh mục, lọc sản phẩm, xem chi tiết.
  + Thêm vào giỏ hàng, chọn phương thức thanh toán.
  + Đặt hàng (COD hoặc Stripe).
  + Xem danh sách đơn hàng và chi tiết từng đơn.

🧑‍💼 **Quản trị viên (Admin)**

* Là người dùng có vai trò đặc biệt trong hệ thống.
* Quản lý toàn bộ dữ liệu nghiệp vụ:
  + Thêm/xóa/sửa danh mục, sản phẩm.
  + Cập nhật kho (Stock).
  + Theo dõi và cập nhật trạng thái đơn hàng (updateStatus).
  + Quản lý user: khóa hoặc mở tài khoản đăng nhập.

🪴 **Plant Shop System (Web UI + API + DB)**

* Là **trung tâm của hệ thống**, bạn đang xây dựng với:
  + **Frontend**: React (giao diện người dùng).
  + **Backend**: Node.js/Express.
  + **Database**: MongoDB (quản lý toàn bộ dữ liệu).
* Các nghiệp vụ chính:
  + Quản lý danh mục, sản phẩm, người dùng, kho, đơn hàng.
  + Khi người mua chọn **COD** → tạo đơn, trừ kho, thông báo thành công.
  + Khi người mua chọn **Stripe** → tạo checkout session → chờ Stripe phản hồi → cập nhật đơn và kho.
  + Hiển thị thông báo kết quả thanh toán cho người dùng (toast “Đặt hàng thành công”).
  + Không sử dụng email hay API vận chuyển ngoài.

💳 **Stripe Payment Gateway (Hệ thống ngoài)**

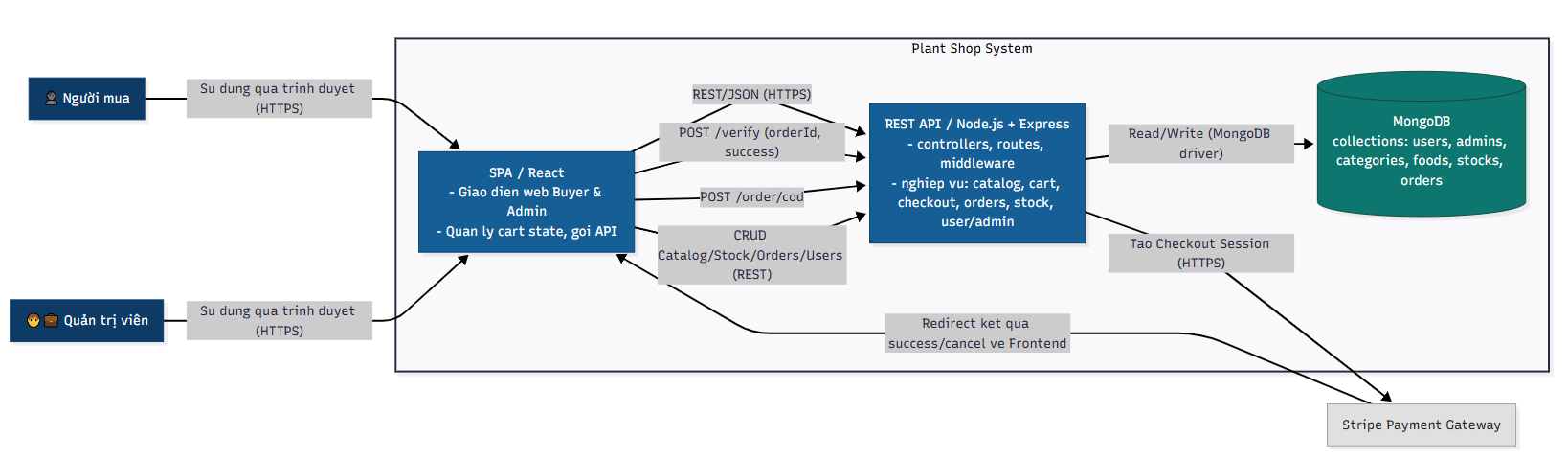
* Hệ thống bên thứ ba xử lý thanh toán trực tuyến.
* **Plant Shop System** tạo Checkout Session gửi lên Stripe, sau đó:
  + Stripe xử lý giao dịch thanh toán thẻ.
  + Gửi kết quả **success/cancel** về hệ thống qua redirect URL.
* Stripe không thay đổi dữ liệu đơn hàng; việc cập nhật (verifyOrder) là do **Plant Shop System** thực hiện.

🎯 **Tóm tắt mối quan hệ chính**

| **Tác nhân** | **Hành động chính** | **Kết nối tới** |
| --- | --- | --- |
| Buyer | Duyệt, thêm giỏ, đặt hàng (COD/Stripe), xem đơn | Plant Shop System |
| Admin | Quản lý toàn bộ dữ liệu nghiệp vụ | Plant Shop System |
| Plant Shop System | Tạo Checkout Session, xử lý thanh toán COD/Stripe, cập nhật kho và đơn hàng | Stripe (khi thanh toán online) |
| Stripe | Gửi kết quả thanh toán về hệ thống | Plant Shop System |

✅ **Kết luận:**  
Sơ đồ của bạn **đã hợp lý**, chỉ cần chỉnh nhỏ phần mô tả tương tác để dễ đọc và đúng phạm vi “Context” (ngữ cảnh).

### C2-Container



Hình 11. C2-Container

Mô tả cấu trúc nội bộ hệ thống, các **container** (UI, API, Database, dịch vụ ngoài) và cách chúng tương tác với nhau.

👤 Actors (Tác nhân bên ngoài)

* **Người mua (Buyer)**: duyệt sản phẩm, thêm giỏ, đặt hàng (COD/Stripe), xem đơn hàng.
* **Quản trị viên (Admin)**: quản lý danh mục, sản phẩm, kho, đơn hàng, người dùng (khóa/mở).

🟦 Containers trong hệ thống

1. **SPA (React)**

* Giao diện người dùng (buyer & admin).
* Quản lý **cart**, điều hướng trang, gọi **REST API**.
* Thanh toán:
  + COD → gọi /order/cod.
  + Stripe → gọi /order → redirect sang Stripe → sau đó /verify.
* Hiển thị thông báo kết quả (“Đặt hàng thành công”).

2. **REST API (Node.js + Express)**

* Chứa **controllers**, **routes**, **middleware**, **models**.
* Thực hiện nghiệp vụ:
  + Quản lý **Catalog**, **Cart**, **Orders**, **Stock**, **User/Admin**.
  + **COD**: kiểm tra kho → trừ tồn → tạo đơn.
  + **Stripe**: tạo session, xác nhận, cập nhật đơn & kho.
  + **Admin**: cập nhật trạng thái đơn, hoàn kho khi hủy.

3. **MongoDB**

* Lưu dữ liệu cho toàn hệ thống: users, admins, categories, foods, stocks, orders.
* Order.items[] nhúng để giữ giá & thông tin sản phẩm tại thời điểm đặt.

⬜ External System

**Stripe Payment Gateway**

* Nhận yêu cầu thanh toán từ API.
* Xử lý giao dịch và redirect về **SPA** (success/cancel).
* Sau đó SPA gọi /verify để API cập nhật đơn hàng & kho.

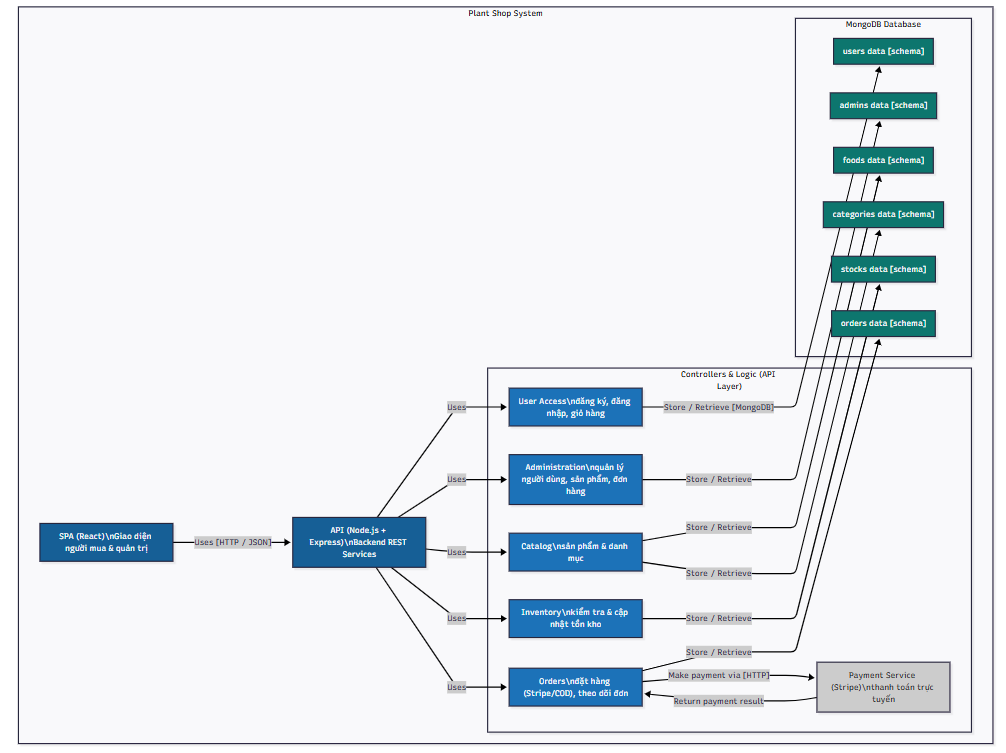
🔄 Luồng tương tác chính

1. Buyer/Admin → **SPA (React)** qua trình duyệt.
2. SPA → **API (Express)** bằng REST/JSON.
3. API ↔ **MongoDB** để ghi/đọc dữ liệu.
4. API ↔ **Stripe** để thanh toán trực tuyến.
5. Stripe → SPA → API → MongoDB (xác nhận đơn).
6. SPA hiển thị kết quả đặt hàng.

✅ Tổng kết

| **Thành phần** | **Công nghệ** | **Vai trò chính** |
| --- | --- | --- |
| **SPA (React)** | ReactJS | Giao diện web, giỏ hàng, gọi API |
| **REST API (Express)** | Node.js | Xử lý nghiệp vụ và kết nối Stripe & DB |
| **MongoDB** | NoSQL | Lưu dữ liệu người dùng, sản phẩm, đơn hàng |
| **Stripe** | External | Cổng thanh toán online |

### C3-Component(High-Level)



Hình 12. C3-Component(High-Level)

**C3 – Component Diagram (High-Level)**

🌿 **Plant Shop System**

Hệ thống bán cây trực tuyến, gồm hai phần chính:

🖥️ **1. SPA (React)**

* Là **giao diện người dùng** (frontend).
* Cho phép **người mua** xem sản phẩm, thêm vào giỏ, thanh toán (COD/Stripe), xem đơn hàng.
* **Admin** cũng đăng nhập từ giao diện này để quản lý hệ thống.
* Gửi/nhận dữ liệu qua **HTTP/JSON** tới API backend.

⚙️ **2. API (Node.js + Express)**

* Là **backend RESTful API**, xử lý toàn bộ nghiệp vụ và kết nối cơ sở dữ liệu.
* Các thành phần chính (Modules):

| **Thành phần** | **Chức năng chính** |
| --- | --- |
| **User Access** | Đăng ký, đăng nhập, cập nhật thông tin và giỏ hàng người dùng. |
| **Catalog** | Hiển thị & quản lý danh mục, sản phẩm (CRUD). |
| **Orders** | Đặt hàng, thanh toán COD hoặc Stripe, xem lịch sử đơn hàng. |
| **Inventory** | Kiểm tra, trừ/tăng tồn kho khi đặt hàng hoặc hủy đơn. |
| **Administration** | Quản trị người dùng, đơn hàng, sản phẩm, tồn kho. |
| **Payment Service (Stripe)** | Tích hợp thanh toán trực tuyến, xử lý tạo phiên checkout và trả kết quả. |

💾 **3. MongoDB Database**

* Lưu trữ toàn bộ dữ liệu của hệ thống qua **Mongoose Models**:

| **Collection** | **Nội dung lưu trữ** |
| --- | --- |
| users | Thông tin người mua, giỏ hàng, tài khoản. |
| admins | Tài khoản quản trị viên. |
| categories | Danh mục sản phẩm (loại cây). |
| foods | Sản phẩm (tên, giá, hình ảnh, mô tả). |
| stocks | Số lượng tồn kho từng sản phẩm. |
| orders | Thông tin đơn hàng, trạng thái, hình thức thanh toán. |

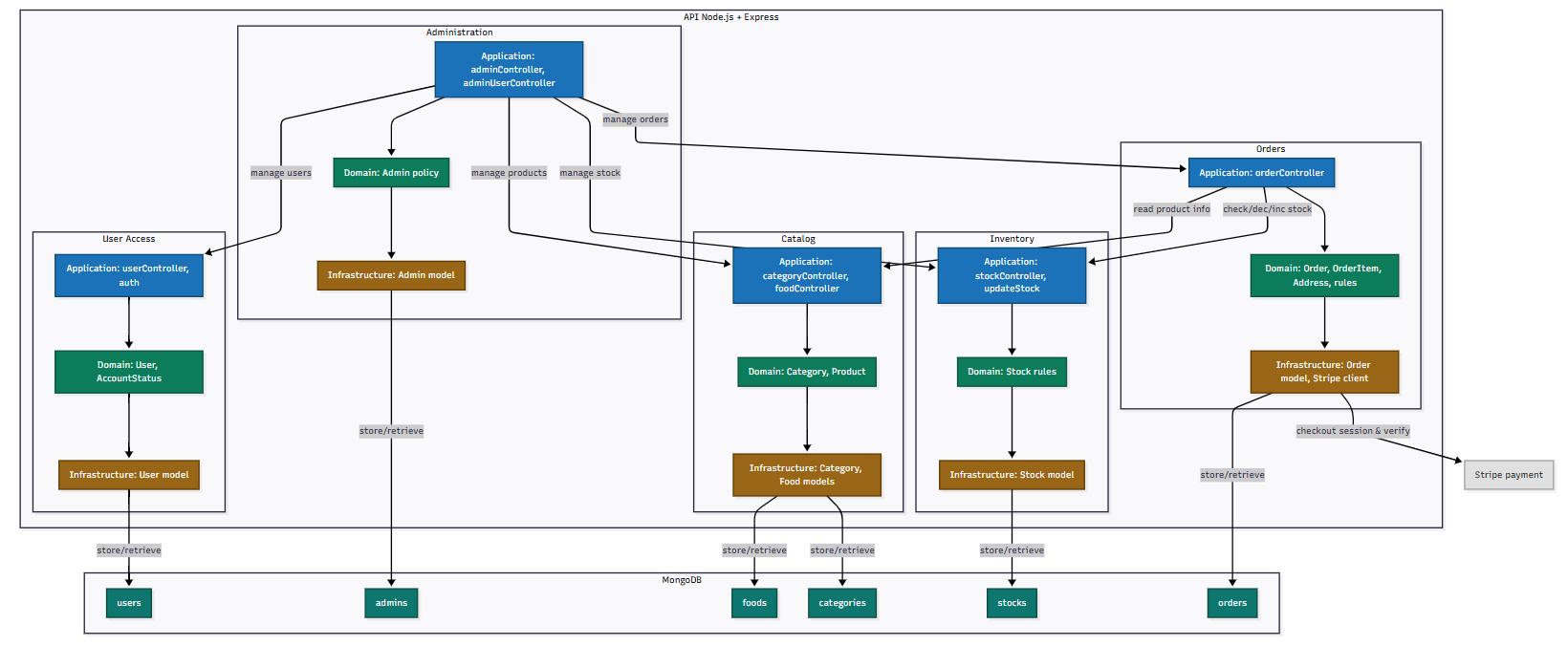
🔁 **Luồng tương tác chính**

1. **SPA → API:** Gửi yêu cầu qua HTTP (REST API).
2. **API → MongoDB:** Ghi/đọc dữ liệu sản phẩm, đơn hàng, user, tồn kho.
3. **Order → Stripe Service:** Khi người mua chọn thanh toán online, API gọi Stripe để tạo phiên thanh toán.
4. **Stripe → Order:** Trả kết quả thanh toán (success/failure).
5. **Order → Inventory:** Nếu thanh toán thành công, cập nhật tồn kho.

✅ **Tóm tắt:**  
Sơ đồ mô tả kiến trúc **đa tầng rõ ràng**:

* **Frontend (React)** giao tiếp với
* **Backend (Node.js + Express)** gồm nhiều module xử lý nghiệp vụ,
* **Database (MongoDB)** làm tầng lưu trữ dữ liệu,
* Và **Stripe** là **hệ thống bên ngoài** hỗ trợ thanh toán.

### C3-Component(Model Level)



Hình 14. C3-Component(Model Level)

Sơ đồ C3 (Module-Level) mô tả cấu trúc nội bộ bên trong container **API (Node.js + Express)** của hệ thống **Plant Shop**.  
Nó thể hiện cách các **module chính** (User Access, Catalog, Inventory, Orders, Administration) được tổ chức theo 3 lớp:

* **Application Layer (Controller / Business logic):** xử lý request từ client (SPA React), gọi domain logic.
* **Domain Layer (Entities / Rules):** định nghĩa quy tắc nghiệp vụ (business rules, validation, stock update, order status...).
* **Infrastructure Layer (Repository / Integration):** kết nối với cơ sở dữ liệu (MongoDB) hoặc dịch vụ ngoài (Stripe).

⚙️ **Cấu trúc và vai trò từng module:**

🔸 1. **User Access Module**

* **Application:**  
  Gồm userController.js, auth.js, adminAuth.js — đảm nhiệm đăng ký, đăng nhập, xác thực JWT, quản lý cartData, và chức năng khóa/mở tài khoản.
* **Domain:**  
  Định nghĩa các quy tắc về người dùng, trạng thái tài khoản (active/locked).
* **Infrastructure:**  
  userModel.js chịu trách nhiệm lưu trữ và truy xuất dữ liệu trong **users collection**.
* **Tương tác:**  
  Admin có thể quản lý người dùng thông qua adminController.

🔸 2. **Catalog Module**

* **Application:**  
  categoryController.js, foodController.js – cho phép Admin CRUD danh mục và sản phẩm; người mua xem danh mục, chi tiết sản phẩm.
* **Domain:**  
  Định nghĩa quy tắc sản phẩm (phân loại, lọc, sắp xếp).
* **Infrastructure:**  
  Category.js, Food.js (hoặc foodModel.js) kết nối với **categories**, **foods** collection.

🔸 3. **Inventory Module**

* **Application:**  
  stockController.js, updateStock.js – quản lý tồn kho, kiểm tra số lượng trước khi đặt hàng, trừ/cộng kho khi thanh toán hoặc hủy.
* **Domain:**  
  Các quy tắc kiểm tra hàng tồn (assertEnoughStock, decStock, incStock).
* **Infrastructure:**  
  Stock.js kết nối với **stocks collection**.
* **Quan hệ:**  
  Module **Orders** gọi Inventory để cập nhật kho mỗi khi có thanh toán hoặc hủy đơn.

🔸 4. **Orders Module**

* **Application:**  
  orderController.js xử lý toàn bộ vòng đời đơn hàng:  
  placeOrder, placeOrderCod, verifyOrder, listOrders, updateStatus, getOrderById.
* **Domain:**  
  Order, OrderItem, Address, OrderStatus, PaymentFlag.  
  Quy tắc nghiệp vụ:
  + Giảm kho khi thanh toán thành công.
  + Hoàn kho nếu đơn thanh toán bị hủy.
* **Infrastructure:**  
  orderModel.js kết nối với **orders collection**; tích hợp **Stripe API** để thanh toán online.
* **Quan hệ:**  
  Gọi **Inventory** để cập nhật kho, và gọi **Stripe** để xử lý thanh toán.

🔸 5. **Administration Module**

* **Application:**  
  adminController.js, adminUserController.js – quản lý toàn hệ thống (người dùng, sản phẩm, tồn kho, đơn hàng).
* **Domain:**  
  Admin, AdminPolicy – xác định quyền thao tác.
* **Infrastructure:**  
  adminModel.js kết nối với **admins collection**.  
  Có thể gọi sang các module khác để thực hiện hành động quản lý.

🔸 6. **Dữ liệu & Dịch vụ ngoài**

* **MongoDB:**  
  Gồm 6 collection: users, admins, categories, foods, stocks, orders.
* **Stripe (External Service):**  
  Được Orders module gọi để tạo session và xác nhận kết quả thanh toán.

🔁 **Luồng hoạt động mẫu:**

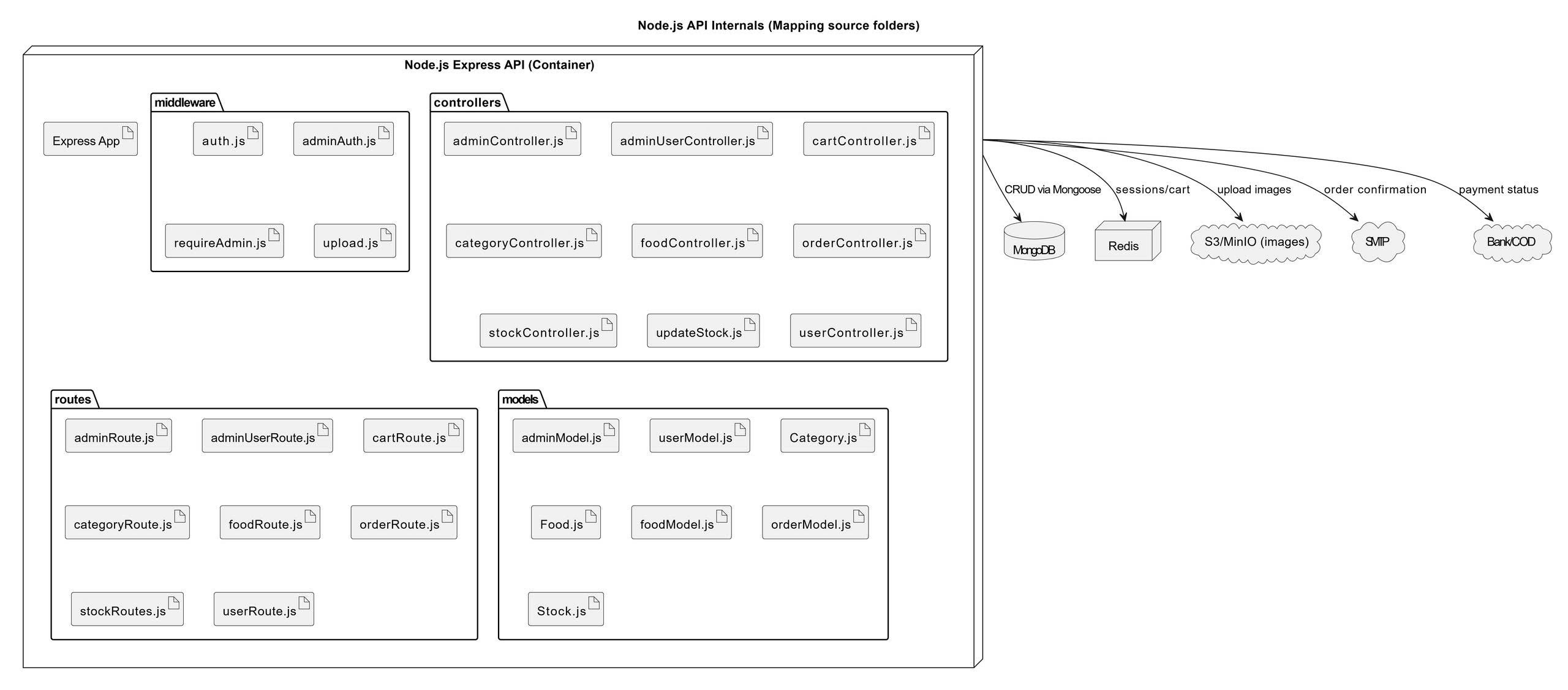
1. Người dùng thêm sản phẩm → giỏ hàng → đặt hàng COD hoặc Stripe.
2. API nhận request → Orders module xử lý → kiểm kho qua Inventory.
3. Nếu thanh toán Stripe → gọi Stripe API, xác nhận kết quả → cập nhật đơn và kho.
4. Nếu COD → lưu đơn hàng, trừ kho ngay.
5. Admin theo dõi hoặc cập nhật trạng thái đơn hàng (hoàn kho nếu hủy).

### C4- Code/Implementation Diagram:



Hình 15. C4- Code/Implementation Diagram

## Deployment view



Hình 16. Deployment view

Sơ đồ này mô tả cấu trúc triển khai nội bộ của backend Node.js API, tương ứng với container “Plant Shop API” trong sơ đồ C2.

Cụ thể, nó liên kết trực tiếp giữa các thư mục mã nguồn (code folders) trong dự án của bạn với các thành phần logic / kỹ thuật khi chạy thật (deployment components, external services).

2. Các phần chính

(A) Node.js Express API (Container)

Đây là ứng dụng chính, được tổ chức thành nhiều thư mục con (module) trong source của bạn:

middleware/

Chứa các logic xử lý ở tầng trung gian — áp dụng cho request trước khi vào controller.

• auth.js – xác thực người dùng (JWT).

• adminAuth.js – xác thực quyền admin.

• requireAdmin.js – chặn truy cập nếu không có quyền quản trị.

• upload.js – xử lý upload hình ảnh (sử dụng Multer hoặc MinIO/S3).

controllers/

Chứa các controller xử lý nghiệp vụ chính.

Mỗi controller tương ứng với một phần module trong hệ thống:

• adminController.js, adminUserController.js: quản lý người dùng, thống kê.

• cartController.js: quản lý giỏ hàng, lưu cartData vào MongoDB hoặc Redis.

• categoryController.js, foodController.js: CRUD danh mục & sản phẩm.

• stockController.js, updateStock.js: kiểm tra, giảm/tăng tồn kho khi đặt hàng.

• orderController.js: đặt hàng, xác thực thanh toán Stripe hoặc COD.

• userController.js: đăng ký, đăng nhập, lấy giỏ hàng, lịch sử đơn.

routes/

Khai báo các endpoint REST API mapping tới controller tương ứng:

• userRoute.js → userController.js

• orderRoute.js → orderController.js

• foodRoute.js, categoryRoute.js → CRUD sản phẩm/danh mục

• stockRoutes.js → stockController.js

• adminRoute.js, adminUserRoute.js → quản lý admin & user

=> Mỗi route import controller và middleware để xử lý request/response đúng pipeline.

models/

Các mô hình (schema) Mongoose tương ứng với bảng trong MongoDB:

• userModel.js, adminModel.js, foodModel.js, categoryModel.js,

orderModel.js, stockModel.js

=> Đây là tầng kết nối dữ liệu (Data Access Layer).

Express App

Là entry point (app.js hoặc server.js), nơi:

• import toàn bộ route, middleware

• khởi tạo Express server

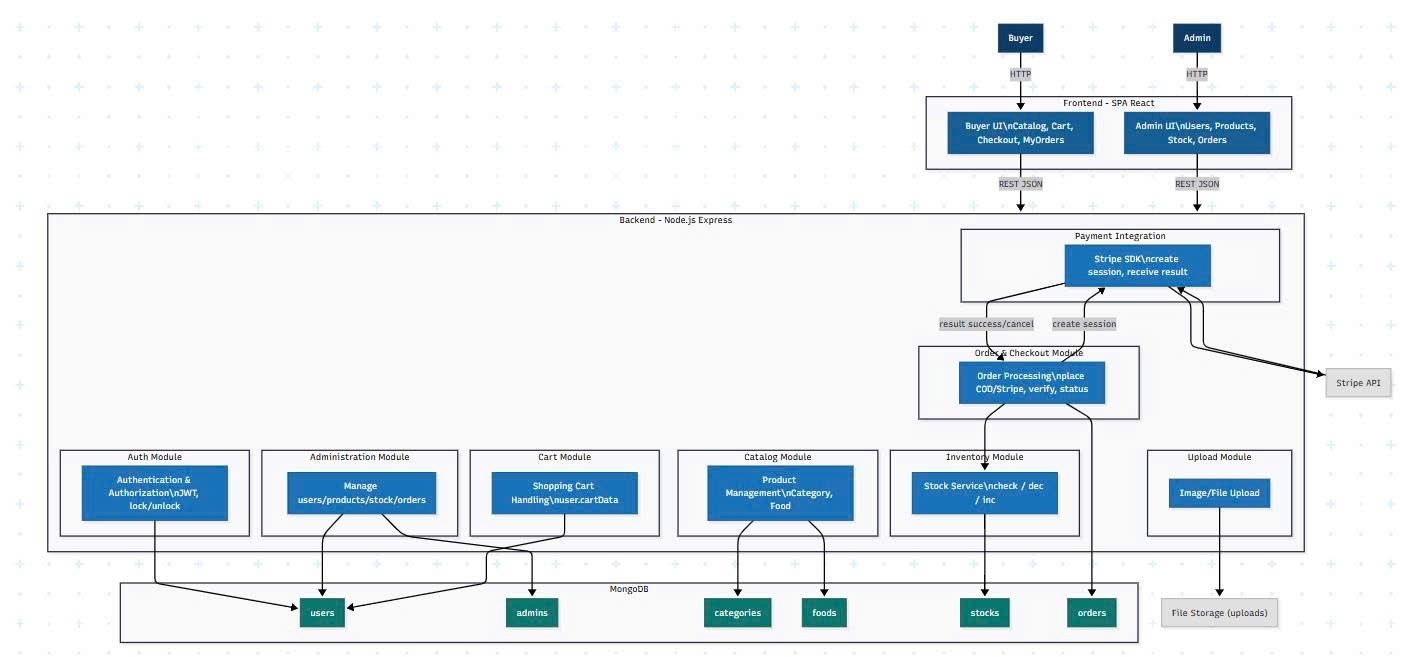
• kết nối MongoDB

• chạy ứng dụng (app.listen(port))

Dòng chảy xử lý tổng quát

* Client (React SPA) gửi request đến API.
* Request đi qua các middleware để xác thực và upload nếu cần.
* Đến controller xử lý nghiệp vụ (đọc/ghi DB, gọi service ngoài).
* Controller dùng model để thao tác với MongoDB.
* Nếu là đơn hàng thanh toán online, controller gọi Stripe API, nhận kết quả success/cancel.
* Trả response JSON về frontend.

## Block Diagram



Hình 17. Block Diagram

**Giải thích sơ đồ kiến trúc tổng thể Plant Shop**

🧍‍♂️ 1. **Tầng người dùng (Actors)**

* **Buyer (Người mua)**:
  + Truy cập hệ thống qua trình duyệt web.
  + Thực hiện các thao tác: xem danh mục sản phẩm, thêm vào giỏ hàng, đặt hàng (COD hoặc Stripe), và theo dõi đơn hàng.
* **Admin (Quản trị viên)**:
  + Cũng truy cập qua giao diện web.
  + Quản lý người dùng, danh mục sản phẩm, kho hàng, đơn hàng, và có thể khóa/mở tài khoản người dùng.

💻 2. **Frontend – SPA React**

Đây là **ứng dụng giao diện một trang (Single Page Application – SPA)** viết bằng **ReactJS**, chia làm 2 phần giao diện:

* **Buyer UI:**  
  Hiển thị cho người dùng mua hàng:
  + Trang chủ, danh mục, chi tiết sản phẩm.
  + Giỏ hàng, thanh toán, theo dõi đơn hàng.
* **Admin UI:**  
  Giao diện dành cho quản trị viên:
  + Quản lý người dùng, sản phẩm, kho, và đơn hàng.

➡️ **Frontend** giao tiếp với **Backend** thông qua **REST API** (dạng JSON qua HTTPS).

⚙️ 3. **Backend – Node.js + Express (Monolithic)**

Toàn bộ nghiệp vụ xử lý tập trung tại backend, gồm nhiều **module con (controllers/services)** tương ứng với từng chức năng trong source code của bạn.

🔹 Auth Module

* **Chức năng:**
  + Đăng ký, đăng nhập người dùng.
  + Xác thực qua **JWT token**.
  + Cho phép **Admin khóa/mở** tài khoản người dùng.
* **Dữ liệu:** bảng users (đăng nhập thường) và admins.

🔹 Administration Module

* **Chức năng:**
  + Quản lý các đối tượng hệ thống: **user**, **product**, **stock**, **order**.
  + Có thể xem thống kê, cập nhật trạng thái đơn hàng, kiểm soát tồn kho.

🔹 Cart Module

* **Chức năng:**
  + Quản lý giỏ hàng của người dùng (cartData trong userModel).
  + Thêm, xóa, cập nhật số lượng, tính tổng tạm thời.

🔹 Catalog Module

* **Chức năng:**
  + Quản lý **danh mục (Category)** và **sản phẩm (Food/Product)**.
  + CRUD cho admin, API đọc cho buyer (xem danh mục, chi tiết sản phẩm).

🔹 Inventory Module

* **Chức năng:**
  + Quản lý **tồn kho (Stock)** – giảm khi đặt hàng, tăng lại khi hủy đơn.
  + Thực thi các hàm nghiệp vụ: assertEnoughStock, decStock, incStock.
  + Kết nối trực tiếp với **Order Module** để cập nhật kho tự động.

🔹 Order/Checkout Module

* **Chức năng:**
  + Xử lý toàn bộ quy trình đặt hàng (COD hoặc Stripe).
  + Tạo order (placeOrder), xác nhận (verifyOrder), cập nhật trạng thái, hoàn kho nếu đơn bị hủy.
  + Lưu đơn vào orders collection.

🔹 Payment Integration

* **Chức năng:**
  + Tích hợp **Stripe API** cho thanh toán online.
  + Tạo **Checkout Session** và nhận **kết quả success/cancel** trả về.
  + Khi thanh toán thành công → trừ kho, cập nhật đơn.
  + Khi hủy → xóa order khỏi database.

🔹 Upload Module

* **Chức năng:**
  + Quản lý tải ảnh sản phẩm lên thư mục uploads/.
  + Dùng cho phần thêm sản phẩm ở giao diện admin.

🗄️ 4. **Database – MongoDB**

Các module backend tương tác với **MongoDB** thông qua **Mongoose models**.  
Các collection tương ứng:

| **Collection** | **Dữ liệu lưu trữ** |
| --- | --- |
| **users** | thông tin người mua (cart, trạng thái, email, password, v.v.) |
| **admins** | thông tin quản trị viên |
| **categories** | danh mục sản phẩm |
| **foods** | sản phẩm chi tiết (tên, giá, hình ảnh, mô tả) |
| **stocks** | tồn kho từng sản phẩm |
| **orders** | đơn hàng của người dùng (items, address, amount, status, payment) |

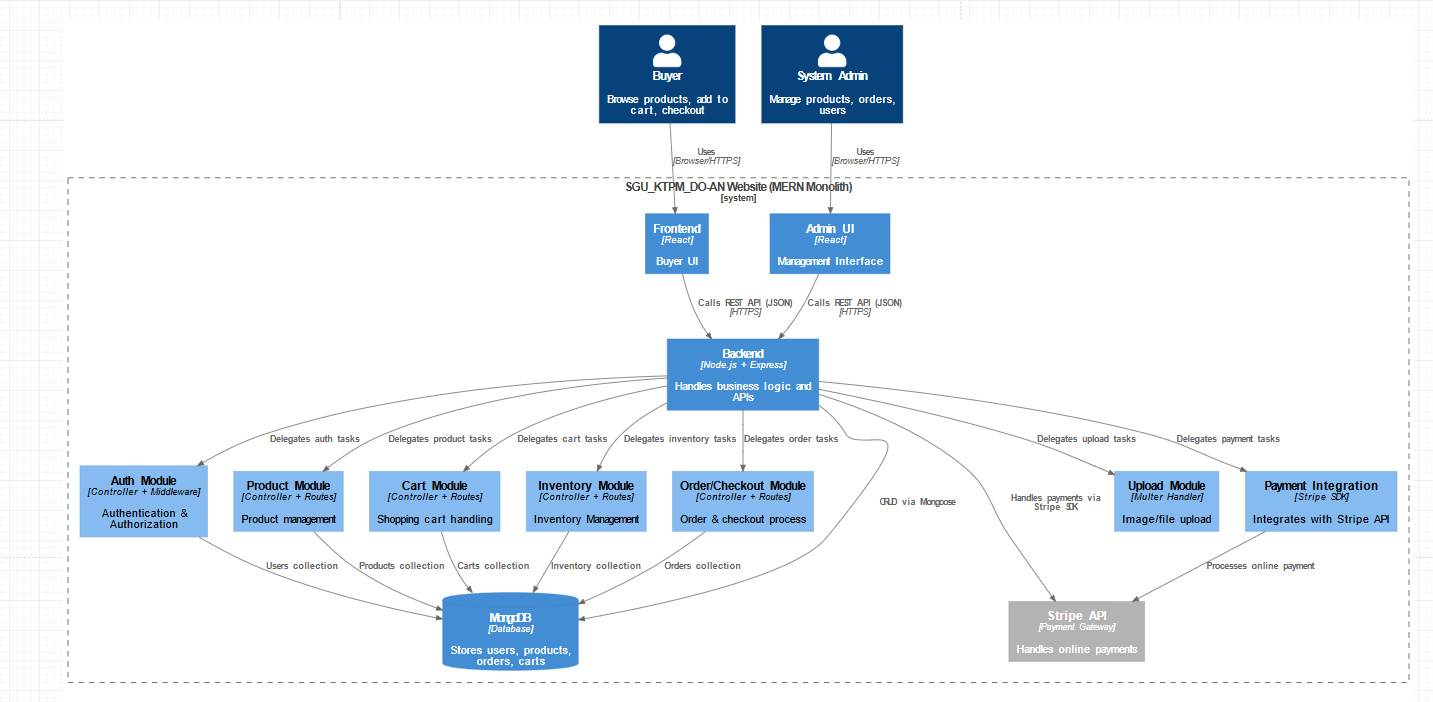
🌐 5. **External Services**

* **Stripe API:**  
  Cổng thanh toán trực tuyến được kết nối qua Stripe SDK.
  + Hệ thống gửi yêu cầu tạo session (checkout).
  + Stripe xử lý và redirect về hệ thống với kết quả.
* **File Storage (uploads):**  
  Lưu trữ ảnh sản phẩm hoặc hình ảnh tải lên từ Admin.

🔁 6. **Luồng hoạt động tóm tắt**

1. Buyer hoặc Admin truy cập website → React SPA.
2. SPA gọi REST API đến backend (Node.js/Express).
3. Backend xử lý nghiệp vụ thông qua các module tương ứng.
4. Dữ liệu được lưu/truy xuất từ **MongoDB**.
5. Nếu thanh toán online, backend tạo session qua **Stripe API**, nhận kết quả → cập nhật đơn hàng & kho.
6. Tất cả dữ liệu hình ảnh được lưu ở **File Storage** nội bộ.

## CommunicationView



Hình 18. CommunicationView

Sơ đồ này tập trung vào **cách người dùng, thành phần nội bộ và hệ thống bên ngoài giao tiếp với nhau**.

* **Tác nhân (Actors)**:
  + **Buyer (Người mua)**: sử dụng Buyer UI trên trình duyệt/HTTPS.
  + **System Admin (Quản trị viên)**: sử dụng Admin UI trên trình duyệt/HTTPS.
* **Hệ thống**:
  + **Buyer UI (React)**: hiển thị danh mục sản phẩm, giỏ hàng, thanh toán cho khách hàng.
  + **Admin UI (React)**: cho quản trị viên quản lý sản phẩm, người dùng, đơn hàng.
  + Cả hai UI đều giao tiếp với backend qua **REST API (HTTPS, JSON)**.
* **Backend (Node.js + Express)**:
  + Là trung tâm nhận API call từ UI, xử lý nghiệp vụ, và phân công cho các module con.
  + **Các module bên trong**:
    - Auth → xác thực/ủy quyền.
    - Product → quản lý sản phẩm.
    - Cart → quản lý giỏ hàng.
    - Order/Checkout → xử lý đơn hàng.
    - Upload → upload hình ảnh/file.
    - Payment → giao tiếp với Stripe.
  + Giao tiếp với **MongoDB** để lưu/truy vấn dữ liệu.
  + Kết nối với **Stripe API** để xử lý thanh toán an toàn.
* **Hệ thống ngoài**:
  + **MongoDB**: chứa dữ liệu Users, Products, Carts, Orders.
  + **Stripe API**: thực hiện giao dịch thanh toán do backend gửi.

Luồng giao tiếp chính:

1. Buyer/Admin → UI (React) → gọi API tới Backend.
2. Backend → Module tương ứng → MongoDB (dữ liệu) hoặc Stripe API (thanh toán).
3. Kết quả trả về UI → hiển thị cho người dùng.

# Test plan

* 1. Introduction
     1. Purpose

Mục đích của tài liệu này là định nghĩa cách thức kiểm thử cho dự án **MOW Garden Website**. Tài liệu giúp đội dự án hiểu rõ phạm vi test, cách tổ chức, tiêu chí chấp nhận, rủi ro và nguồn lực cần thiết. Nội dung chính gồm:

* Giới thiệu chung về dự án và phạm vi kiểm thử.
* Danh sách yêu cầu, tiêu chí chấp nhận.
* Chiến lược và kế hoạch test chi tiết.
* Nguồn lực, môi trường và mốc thời gian.
* Kết quả bàn giao cuối cùng.
  + 1. Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviation | Description | Note |
| AT | Acceptance Test | Kiểm thử chấp nhận |
| IT | Integration Test | Kiểm thử tích hợp |
| SRS | Software Requirement Specification | Đặc tả yêu cầu phần mềm |
| QA | Quality Assurance | Đảm bảo chất lượng |
| DB | Database | Cơ sở dữ liệu |
| UAT | User Acceptance Testing | Kiểm thử chấp nhập của người dùng |
| UI | User Interface | Giao diện người dùng |

* + 1. References

|  |  |  |
| --- | --- | --- |
| Title/File name | Author | Effective Date |
| Business Context – MOW Garden | Nhóm Project | 2025-09-25 |
| SRS – MOW Garden | Nhóm Project | 2025-09-27 |
| Test Guideline Document | Fsoft | 2024-12-01 |

* + 1. Background information

MOW Garden là một website bán cây cảnh, phụ kiện làm vườn. Hệ thống cho phép người dùng:

* Xem danh mục sản phẩm, tìm kiếm và lọc theo loại/giá.
* Xem chi tiết sản phẩm (hình ảnh, mô tả, giá).
* Thêm/xóa sản phẩm trong giỏ hàng.
* Đặt hàng (checkout).
* Nhận thông báo (notification) khi đặt thành công.
* Quản trị viên (Admin) quản lý tồn kho và xác nhận đơn hàng.

Mục tiêu kiểm thử là đảm bảo hệ thống hoạt động đúng, ổn định, bảo mật và mang lại trải nghiệm tốt cho người dùng.

* + 1. Scope of testing

Trong phạm vi:

* Kiểm thử chức năng (catalog, cart, checkout, inventory, notification, phân quyền).
* Kiểm thử giao diện (UI/UX).
* Kiểm thử cơ sở dữ liệu (dữ liệu đồng nhất, không mất mát).
* Kiểm thử hiệu năng cơ bản (tải đồng thời nhỏ và vừa).

Ngoài phạm vi:

* Thanh toán quốc tế.
* Tích hợp với hệ thống logistics ngoài.
* Yêu cầu phi chức năng nâng cao (scalability enterprise-level).
  + 1. Constraints
* Thời gian test chỉ 2 tuần.
* Nhân lực giới hạn (4 người).
* Hệ thống mới chỉ chạy trên môi trường local + demo server, chưa có production đầy đủ.
  + 1. Risk list

|  |  |  |
| --- | --- | --- |
| Risk | Impact | Mitigation |
| Thiếu môi trường Production | Test chưa phản ánh thực tế | Dùng mock/staging server |
| Thời gian ngắn | Có thể sót test case | Ưu tiên test tính năng quan trọng |
| Nhân lực hạn chế | Chậm tiến độ | Chia rõ vai trò, tập trung high priority |
| Thiếu tool tự động | Test thủ công nhiều | Sử dụng Postman, JMeter cơ bản |

* + 1. Training needs

Đội ngũ cần được training nhanh về:

* Cách viết và quản lý test case.
* Sử dụng Postman để test API.
* Dùng JMeter để test hiệu năng cơ bản.
* Cách báo bug và quản lý issue trên GitHub.
  1. Requirements for test
     1. Test Items

Danh sách các tính năng và chức năng chính được chọn làm mục tiêu kiểm thử trong dự án MOW Garden:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Name of function | Outline of functions | Number of Testcase | Note |
| 1 | Product Catalog | Xem danh sách sản phẩm, tìm kiếm, lọc theo loại và giá. Xem chi tiết sản phẩm (tên, mô tả, giá, hình ảnh). | 15 | Bao gồm test UI và chức năng lọc/tìm kiếm |
| 2 | Shopping Cart | Thêm/xóa sản phẩm vào giỏ hàng, cập nhật số lượng, tính tổng giá trị. | 12 | Kiểm thử cả khi chưa đăng nhập |
| 3 | Checkout Process | Nhập thông tin đặt hàng, chọn phương thức thanh toán, xác nhận đơn hàng. | 20 | Bao gồm validation dữ liệu |
| 4 | Inventory Management | Quản lý số lượng tồn kho, cập nhật khi có đơn hàng. | 8 | Dành cho phía Admin |
| 5 | Access Control (User/Admin) | Đăng ký, đăng nhập, phân quyền giữa User và Admin. | 15 | Bao gồm xác thực và bảo mật cơ bản |
| 6 | Order Management | Xem danh sách đơn hàng, xem chi tiết đơn hàng, theo dõi trạng thái giao hàng (User & Admin). | 10 | Bao gồm test dữ liệu hiển thị và cập nhật trạng thái đơn. |
| 7 | Notification Service | Gửi thông báo khi đặt hàng thành công. | 6 | Test push notification |
| 8 | Product Management | CRUD sản phẩm và gán vào Inventory | 12 | Áp dụng quyền Admin |

* + 1. Acceptance Test Criteria

Để được chấp nhận, phần mềm cần đạt các tiêu chí kiểm thử sau:

* **Test coverage** ≥ 90%
* **Successful Test coverage** ≥ 95%
* Số lượng test case (Unit/Integration/System Test) phải được thực thi đầy đủ.
* Số lượng defect/weighted defect ≤ 5 / module chính.
  1. Feature to be tested

Danh sách dưới đây xác định các hạng mục – use cases, yêu cầu chức năng và phi chức năng – được chọn làm đối tượng kiểm thử. Danh sách này đại diện cho **những gì cần được kiểm thử trong hệ thống MOW Garden**.

* + 1. Functionality
       1. Product Catalog
* Xác minh rằng hệ thống có thể hiển thị danh sách sản phẩm đầy đủ từ cơ sở dữ liệu.
* Xác minh rằng người dùng có thể lọc sản phẩm theo loại, giá hoặc trạng thái tồn kho.
* Xác minh rằng người dùng có thể tìm kiếm sản phẩm theo từ khóa.
* Xác minh rằng danh sách sản phẩm được cập nhật tự động khi có thay đổi trong kho hàng.
  + - 1. Product Details
* Xác minh rằng hệ thống hiển thị chính xác thông tin sản phẩm gồm: tên, mô tả, giá, hình ảnh, và số lượng còn lại.
* Xác minh rằng người dùng có thể xem nhiều hình ảnh của cùng một sản phẩm.
* Xác minh rằng sản phẩm hết hàng được hiển thị rõ ràng và không thể thêm vào giỏ hàng.
* Xác minh rằng khi Admin chỉnh sửa thông tin sản phẩm, dữ liệu hiển thị ở trang chi tiết được cập nhật tự động.
  + - 1. Shopping Cart
* Xác minh rằng người dùng có thể thêm sản phẩm vào giỏ hàng từ trang danh mục hoặc chi tiết.
* Xác minh rằng người dùng có thể cập nhật số lượng sản phẩm trong giỏ hàng.
* Xác minh rằng người dùng có thể xóa sản phẩm khỏi giỏ hàng.
* Xác minh rằng tổng tiền giỏ hàng được tính toán chính xác và hiển thị đúng định dạng.
* Xác minh rằng nếu sản phẩm bị xóa hoặc hết hàng trong kho, giỏ hàng sẽ tự động cập nhật trạng thái.
* Xác minh rằng hệ thống xử lý chính xác khi người dùng thêm cùng một sản phẩm nhiều lần.
  + - 1. Checkout Process
* Xác minh rằng người dùng có thể nhập thông tin giao hàng và thanh toán.
* Xác minh rằng hệ thống kiểm tra hợp lệ các thông tin đầu vào (tên, số điện thoại, địa chỉ).
* Xác minh rằng đơn hàng được lưu trong cơ sở dữ liệu và hiển thị trong lịch sử đặt hàng.
* Xác minh rằng người dùng nhận được thông báo sau khi đặt hàng thành công.
* Xác minh tồn kho giảm đúng số lượng.
* Xác minh rằng trạng thái đơn hàng thay đổi sau khi thanh toán (VD: “Đã xác nhận”, “Đang giao”).
  + - 1. Inventory Management
* Xác minh rằng số lượng tồn kho được cập nhật tự động sau mỗi đơn hàng.
* Xác minh rằng sản phẩm hết hàng không thể thêm vào giỏ hàng.
* Xác minh rằng admin có thể cập nhật thông tin sản phẩm (giá, tồn kho, mô tả).
* Xác minh rằng tồn kho cập nhật chính xác sau khi đơn hàng hoàn tất hoặc bị hủy.
* Xác minh rằng thay đổi tồn kho hiển thị tức thời ở trang chi tiết sản phẩm.
  + - 1. Access Control
* Xác minh rằng hệ thống phân quyền đúng cho buyer và admin.
* Xác minh rằng người dùng chỉ có thể thực hiện các thao tác phù hợp với vai trò của họ.
* Xác minh rằng admin có thể thêm, sửa, xóa sản phẩm; còn buyer chỉ được phép mua hàng.
* Xác minh người dùng có thể cập nhật thông tin cá nhân.
* Xác minh tính năng “Quên mật khẩu” (nếu có).
* Xác minh admin có thể khóa/mở tài khoản người dùng.
* Xác minh rằng người dùng có thể cập nhật thông tin cá nhân (họ tên, địa chỉ, số điện thoại).
  + - 1. Notification Service
* Xác minh rằng người dùng nhận được thông báo sau khi đặt hàng thành công.
* Xác minh rằng admin có thể gửi thông báo khuyến mãi hoặc cập nhật đơn hàng đến người dùng.
* Xác minh rằng thông báo được hiển thị chính xác trên màn hình người dùng.
  + - 1. Order Management
* Xác minh rằng người dùng có thể xem danh sách đơn hàng của họ (mã đơn, ngày tạo, tổng tiền, trạng thái).
* Xác minh rằng người dùng có thể xem chi tiết từng đơn hàng (thông tin giao hàng, sản phẩm, hóa đơn, trạng thái).
* Xác minh rằng trạng thái đơn hàng hiển thị đúng theo tiến trình (Chờ xác nhận → Đang xử lý → Đang giao → Hoàn tất).
* Xác minh rằng admin có thể xem tất cả đơn hàng của người dùng.
* Xác minh rằng admin có thể cập nhật trạng thái đơn hàng (VD: từ “Đang xử lý” sang “Hoàn tất”).
* Xác minh rằng khi admin cập nhật trạng thái, người dùng nhận được thông báo.
  + - 1. Product Management
* Xác minh rằng admin có thể thêm sản phẩm mới với đầy đủ thông tin (tên, mô tả, giá, hình ảnh, tồn kho).
* Xác minh rằng admin có thể chỉnh sửa thông tin sản phẩm.
* Xác minh rằng admin có thể xóa sản phẩm khỏi danh sách.
* Xác minh rằng admin có thể gán sản phẩm mới vào kho hàng.
* Xác minh rằng hệ thống không cho phép thêm sản phẩm trùng mã hoặc trùng tên.
* Xác minh rằng khi admin xóa hoặc chỉnh sửa sản phẩm, dữ liệu được cập nhật tương ứng ở **Product Catalog** và **Inventory**.
  + 1. Usability
       1. Session Support
* Xác minh rằng người dùng có thể duy trì trạng thái đăng nhập trong suốt phiên làm việc.
* Xác minh rằng dữ liệu giỏ hàng vẫn được lưu khi người dùng reload trang hoặc tạm thời thoát ra.
* Xác minh rằng hệ thống tự động đăng xuất sau một khoảng thời gian không hoạt động nhất định để đảm bảo bảo mật.
* Xác minh rằng khi người dùng đăng xuất thủ công, mọi dữ liệu tạm thời (session data) đều bị xóa an toàn.
  + - 1. Display Support
* Xác minh rằng giao diện được hiển thị đúng định dạng trên các trình duyệt phổ biến (Chrome, Edge, Safari, Firefox).
* Xác minh rằng bố cục giao diện tự động điều chỉnh tương thích với kích thước màn hình (desktop, tablet, mobile).
* Xác minh rằng font chữ, màu sắc và biểu tượng hiển thị rõ ràng, không bị lỗi khi thay đổi độ phân giải.
* Xác minh rằng hệ thống hỗ trợ chế độ tối (dark mode) hoặc có khả năng tùy chỉnh giao diện (nếu có).
  + 1. Design Constraints
       1. Test Environment
* Xác minh rằng hệ thống hoạt động ổn định trên các môi trường kiểm thử được định nghĩa (Windows 10/11, macOS, Android, iOS).
* Xác minh rằng trình duyệt và phiên bản trình duyệt tương thích được xác định rõ (ví dụ: Chrome 120+, Edge 110+, Safari 15+).
* Xác minh rằng backend (Node.js/Express) và frontend (ReactJS) có thể triển khai và kết nối thành công trên môi trường staging.
* Xác minh rằng cơ sở dữ liệu (MongoDB) hoạt động bình thường, dữ liệu được lưu và truy xuất chính xác.
  + - 1. Characters
* Xác minh rằng hệ thống hỗ trợ ký tự tiếng Việt có dấu, ký tự đặc biệt, ký hiệu tiền tệ và emoji.
* Xác minh rằng dữ liệu có chứa ký tự đặc biệt không gây lỗi hiển thị hoặc lỗi cơ sở dữ liệu.
* Xác minh rằng hệ thống xử lý chính xác khi người dùng nhập dữ liệu Unicode (ví dụ: tên, địa chỉ, mô tả sản phẩm).
  + 1. Interfaces
* Xác minh rằng bố cục các trang (Home, Product List, Product Detail, Cart, Checkout, Admin Dashboard) được sắp xếp hợp lý.
* Xác minh rằng tất cả các nút, liên kết và biểu mẫu đều có phản hồi trực quan khi tương tác (hover, click, focus).
* Xác minh rằng biểu mẫu nhập liệu có thông báo lỗi cụ thể khi nhập sai hoặc bỏ trống trường bắt buộc.
* Xác minh rằng hệ thống hiển thị thông báo (notification, toast message) rõ ràng khi thao tác thành công hoặc thất bại.
* Xác minh rằng giao diện admin có thể quản lý sản phẩm, đơn hàng và tồn kho dễ dàng, trực quan.
* Xác minh API backend cung cấp dữ liệu chính xác cho frontend, phản hồi đúng cấu trúc JSON và mã trạng thái HTTP
  1. Feature not to be tested

Danh sách dưới đây bao gồm các hạng mục, chức năng hoặc yêu cầu hiện **chưa được đưa vào phạm vi kiểm thử** của phiên bản phát hành hiện tại.  
Những tính năng này có thể được xem xét trong các giai đoạn phát triển hoặc bản mở rộng sau.

* + 1. Functional Features
* **Thanh toán trực tuyến qua bên thứ ba (Payment Gateway Integration):**  
  Việc kết nối và xử lý giao dịch với các cổng thanh toán như PayPal, Momo, ZaloPay, hoặc VNPay chưa được triển khai và kiểm thử trong giai đoạn này.
* **Hệ thống gợi ý sản phẩm (Recommendation System):**  
  Chức năng đề xuất sản phẩm dựa trên hành vi người dùng hoặc lịch sử mua hàng chưa nằm trong phạm vi kiểm thử.
* **Chức năng quản lý người dùng nâng cao (User Management Advanced):**  
  Bao gồm khôi phục mật khẩu, xác thực hai lớp (2FA), và hệ thống đăng nhập bằng mạng xã hội (Google, Facebook) chưa được triển khai để kiểm thử.
  + 1. Non-functional Features
* **Kiểm thử hiệu năng nâng cao (Performance under stress):**  
  Kiểm thử hiệu năng với số lượng người dùng cực lớn (trên 10.000 người truy cập đồng thời) sẽ được thực hiện trong giai đoạn sau.
* **Khả năng mở rộng hệ thống (Scalability):**  
  Các bài kiểm thử về khả năng mở rộng theo cụm máy chủ (clustering, load balancing thực tế) chưa nằm trong phạm vi hiện tại.
* **Độ bảo mật cấp cao (Advanced Security):**  
  Các bài test chuyên sâu về an ninh mạng như kiểm thử xâm nhập (penetration testing) hoặc tấn công injection nâng cao chưa được áp dụng ở giai đoạn này.
  1. Test Strategy
     1. Test types
        1. Function Testing

Function Testing (Kiểm thử chức năng) là loại kiểm thử nhằm đảm bảo tất cả các chức năng trong hệ thống hoạt động đúng theo yêu cầu đặc tả. Với MOW Garden, nó tập trung vào các tính năng chính như: xem sản phẩm, tìm kiếm, thêm vào giỏ hàng, đặt hàng, thanh toán, quản lý tài khoản, quản lý kho.

|  |  |
| --- | --- |
| Test Objective | Xác minh rằng các chức năng chính của hệ thống (xem sản phẩm, tìm kiếm, thêm vào giỏ hàng, thanh toán, quản lý tài khoản, quản lý tồn kho) hoạt động đúng với yêu cầu nghiệp vụ. |
| Technique | - Thực hiện test case dựa trên use case và yêu cầu chức năng.  - Sử dụng dữ liệu hợp lệ và không hợp lệ.  - Kiểm tra validation, business rule.  - Công cụ: Postman, Selenium, JUnit. |
| Completion Criteria | - Tất cả test case được thực hiện.  - Các bug nghiêm trọng được fix, không còn blocking bug. |
| Special Considerations | Kiểm tra đặc biệt các luồng quan trọng: thêm vào giỏ khi chưa login, hủy đơn giữa chừng, giỏ hàng lưu tạm. |

* + - 1. Business Cycle Testing

Business Cycle Testing (Kiểm thử vòng đời nghiệp vụ) đảm bảo toàn bộ quy trình nghiệp vụ từ đầu đến cuối hoạt động trơn tru, không gián đoạn. Trong MOW Garden, đó là từ lúc khách chọn sản phẩm → thêm vào giỏ → thanh toán → cập nhật tồn kho → quản lý đơn hàng.

|  |  |
| --- | --- |
| Test Objective | Đảm bảo các quy trình nghiệp vụ chính (mua hàng → thanh toán → cập nhật tồn kho) hoạt động xuyên suốt. |
| Technique | - Thực hiện end-to-end test.  - Kiểm tra luồng dữ liệu qua các module (Product → Cart → Order → Inventory). |
| Completion Criteria | - Các luồng nghiệp vụ chính được thực hiện thành công không lỗi. |
| Special Considerations | - Cần mô phỏng nhiều kịch bản thực tế: người dùng hủy đơn, thanh toán thất bại, sản phẩm hết hàng. |

* + - 1. User Interface Testing

User Interface Testing (Kiểm thử giao diện người dùng) đảm bảo giao diện web hiển thị đúng, dễ sử dụng, và phản hồi chính xác với thao tác người dùng.

|  |  |
| --- | --- |
| Test Objective | Đảm bảo UI hiển thị đúng, responsive và dễ dùng. |
| Technique | Cross-browser (Chrome/Firefox/Edge/Safari), responsive (desktop/tablet/mobile), kết hợp exploratory & Selenium. |
| Completion Criteria | Không còn lỗi UI Major/Blocker; 95% trang mục tiêu đạt kiểm tra responsive. |
| Special Considerations | Kiểm tra mạng chậm, ảnh nặng, fallback font. |

* + - 1. Security and Access Control Testing

Đảm bảo hệ thống an toàn, chỉ người có quyền mới được truy cập và dữ liệu người dùng được bảo mật.

|  |  |
| --- | --- |
| Test Objective | Kiểm tra bảo mật hệ thống, phân quyền, và chống tấn công. |
| Technique | - Test login/logout, session timeout. - SQL Injection, XSS, CSRF. - Kiểm tra phân quyền admin/user. |
| Completion Criteria | - Không bypass được xác thực. - Không tồn tại lỗ hổng nghiêm trọng OWASP Top 10. |
| Special Considerations | Kiểm tra SSL, HTTPS, mã hóa mật khẩu. |

* + - 1. Regression Testing

Regression Testing là kiểm thử hồi quy, đảm bảo chức năng cũ vẫn hoạt động đúng sau khi thêm chức năng mới hoặc fix bug.

|  |  |
| --- | --- |
| Test Objective | Xác minh rằng các chức năng cũ không bị ảnh hưởng sau thay đổi. |
| Technique | - Chạy lại test case đã pass trước đó. - Dùng automation (Selenium, Jest). |
| Completion Criteria | - Tất cả test case cũ pass. - Không phát sinh bug ngoài mong đợi. |
| Special Considerations | Tích hợp regression test vào CI/CD pipeline. |

* + - 1. Performance and Load Testing

Performance and Load Testing nhằm xác định hệ thống có đáp ứng được yêu cầu về tốc độ và độ ổn định khi có nhiều người dùng đồng thời.

|  |  |
| --- | --- |
| Test Objective | Đảm bảo hệ thống hoạt động ổn định dưới tải cao (nhiều người dùng đồng thời, nhiều sản phẩm). |
| Technique | - Sử dụng JMeter/Locust để mô phỏng 1000+ user đồng thời.  - Thực hiện Load test, Stress test, Volume test. |
| Completion Criteria | - Thời gian phản hồi < 3 giây với 500 user đồng thời.  - Hệ thống không crash khi vượt tải. |
| Special Considerations | - Định nghĩa ngưỡng SLA (Service Level Agreement) cụ thể: thời gian phản hồi, số user tối đa. |

* + - 1. Data and Database Integrity Testing

Đây là kiểm thử tính toàn vẹn dữ liệu: đảm bảo dữ liệu không bị mất, sai lệch, trùng lặp khi thao tác.

|  |  |
| --- | --- |
| Test Objective | Đảm bảo dữ liệu sản phẩm, giỏ hàng, đơn hàng, người dùng được lưu chính xác và nhất quán. |
| Technique | - Thực hiện CRUD trên database. - Kiểm tra transaction khi thanh toán. - Kiểm tra ràng buộc khóa chính/ngoại. - Dùng SQL script + automation. |
| Completion Criteria | - Không có lỗi mất/nhân bản dữ liệu. - Toàn vẹn dữ liệu giữa các bảng. |
| Special Considerations | Kiểm tra rollback khi lỗi |

* 1. Test stages

Các loại test sẽ được thực hiện ở các giai đoạn khác nhau của vòng đời kiểm thử:

* **Unit Test**: Kiểm thử mức hàm/module riêng lẻ, thường do dev thực hiện.
* **Integration Test**: Kiểm thử khi ghép nối nhiều module hoặc API với nhau.
* **System Test**: Kiểm thử toàn bộ hệ thống như một sản phẩm hoàn chỉnh.
* **Acceptance Test**: Kiểm thử chấp nhận, xác minh hệ thống đáp ứng yêu cầu nghiệp vụ của khách hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of Test | Unit | Integration | System | Acceptance |
| Function Test | x | x | x | x |
| Business Cycle Test |  | x | x | x |
| User Interface Test |  | x | x | x |
| Data and Database Integrity Test | x | x | x |  |
| Performance Test |  |  | x |  |
| Security and Access Control Test |  | x | x |  |
| Regression Test | x | x | X |  |

* **Function Test**: xuất hiện ở mọi giai đoạn vì từ unit đến acceptance đều cần test chức năng.
* **Business Cycle Test**: chỉ thực sự quan trọng từ Integration trở lên (liên quan đến nhiều module).
* **UI Test**: không có ở unit, mà bắt đầu từ integration (FE + BE), system và acceptance.
* **Database Test**: có thể bắt đầu từ unit (SQL function, procedure), đến integration và system.
* **Performance Test**: chủ yếu chạy ở System Test (cần hệ thống hoàn chỉnh).
* **Security Test**: cần từ integration trở đi (kiểm tra API, phân quyền, login, …).
* **Regression Test**: chạy liên tục từ Unit → Integration → System để đảm bảo thay đổi không phá vỡ chức năng cũ.
  1. Resource
     1. Human Resource

|  |  |  |  |
| --- | --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities | Location |
| Nguyễn Nhật Hải |  |  |  |
| Nguyễn Thành Đạt |  |  |  |
| Nguyễn Thành Việt |  |  |  |
| Nguyễn Thành Trung Hiếu |  |  |  |

* + 1. Test Management

Xác định phương pháp quản lý cho các mục sau:

**Quản lý kiểm thử:**

* Lập kế hoạch kiểm thử, phân công nhân sự, theo dõi tiến độ bằng **Jira / Trello**.
* Họp nhanh hằng ngày (daily meeting) để cập nhật tiến độ.
* Báo cáo tiến độ hàng tuần cho quản lý dự án (PM).

**Quản lý lỗi (Defect Management):**

* Ghi nhận và theo dõi lỗi bằng công cụ **Jira**.
* Phân loại lỗi theo mức độ nghiêm trọng: **Critical, Major, Minor**.
* Quy trình xử lý lỗi: **Ghi nhận → Gán cho Dev → Sửa → Kiểm thử lại → Đóng lỗi**.
  1. Test Environment
     1. Hardware

Các thiết bị phần cứng được sử dụng trong dự án **MOW Garden**:

**Laptop/PC** (dùng cho kiểm thử web):

* CPU: Intel Core i5 hoặc cao hơn
* RAM: 8GB trở lên
* Ổ cứng: SSD 256GB
* Số lượng: 4 máy
* Thời gian sử dụng: Toàn bộ quá trình kiểm thử
* **Thiết bị mạng:**
* Router Wifi chuẩn 802.11ac
* Kết nối Internet tốc độ tối thiểu 50Mbps
* Số lượng: 1 bộ
  + 1. Software

Các phần mềm cần thiết phục vụ kiểm thử:

**Hệ điều hành:**

* Windows 10 Pro (64-bit)
* Ubuntu 20.04 LTS

**Trình duyệt web:**

* Google Chrome (phiên bản mới nhất)
* Mozilla Firefox (phiên bản mới nhất)
* Microsoft Edge (phiên bản mới nhất)

**Công cụ kiểm thử:**

* Postman (API Testing)
* Selenium WebDriver (Automation Testing)
* JMeter (Performance Testing)

**Ứng dụng quản lý mã nguồn:**

* GitHub (repository code)
  + 1. Infrastructure

Danh sách công cụ và hệ thống hỗ trợ cho dự án **MOW Garden**

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | Tool | Vendor/In-house | Version |
| Defect log | Jira | Atlassian | Cloud Version |
| Quản lý Test Case | TestRail | Gurock Software | 7.5 |
| Quản lý công việc | Trello | Atlassian | Cloud Version |
| Quản lý mã nguồn | GitHub | GitHub Inc. | Cloud Version |
| Hiệu năng (Performance test) | JMeter | Apache | 5.6 |
| Ghi nhận thời gian (Test effort) | Timesheet | In-house | 1.2 |

* 1. Test Milestone

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

* 1. Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |