## CHÚ Ý:

1. Mỗi nhóm soạn thảo báo cáo trực tiếp vào tệp văn bản (docx, edit online bằng Google Docs) theo cấu trúc dưới đây, đặt trong thư mục online của nhóm và đó.
2. Quy tắc trình bày quyển báo cáo phải tuân thủ [Quy định của khoa](http://fithou.edu.vn/userfiles/files/bieumau/QuidinhTrinhbayBaocao_v4_5_2_1.pdf)
3. Báo cáo môn này được phép in 2 mặt (do số trang rất nhiều)
4. **Yêu cầu về Bài toán**

* Đa người dùng.
* Đa thiết bị/phương tiện (máy tính, điện thoại, máy in...).
* Có nghiệp vụ (yêu cầu, quy trình, mẫu biểu) thực tế.
* Sử dụng UML để mô hình hoá các kết quả khảo sát, phân tích, thiết kế
* Sử dụng Git (hoặc công cụ tương đương khác) để quản lý mã nguồn của dự án

1. Mọi nhận định trong báo cáo này phải được chứng minh hoặc dẫn nguồn nếu trích từ tài liệu khác (tài liệu đó phải được liệt kê trong phần TÀI LIỆU THAM KHẢO)
2. Mọi bảng biểu (Table) phải được thiết lập để tiêu đề in lặp lại ở trang thứ 2 trở đi (nếu bảng trải dài nhiều trang)
3. Tài liệu tham khảo được chấp nhận gồm:
   1. Sách về CNPM
   2. Bài báo khoa học về CNPM
   3. Các chuẩn đã được công bố về CNPM
4. Mẫu báo cáo này được cập nhật lần cuối ngày 16/09/2024

## 

**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------

## 

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**PHÁT TRIỂN PHẦN MỀM THEO LỚP BÀI TOÁN “QUẢN LÝ HOẠT ĐỘNG CỦA MỘT NHÀ THUỐC ĐẠT CHUẨN GPP” CHO NHÀ THUỐC HỒNG VÂN**

Giảng viên hướng dẫn : ThS Lê Hữu Dũng

Nhóm sinh viên thực hiện : Nhóm G12

Nguyễn Hồng Ánh – 2210A04 Nguyễn Văn Mạnh – 2210A0

Nguyễn Hải Đăng – 2210A0 Nguyễn Minh Sơn – 2210A0

Đỗ Đan Trường– 2210A0 Trần Thị Mai Thi – 2210A03

Nguyễn Mai Hương – 2210A05 Nguyễn Nguyên Vũ – 2210A0

Trần Tuấn Huy – 2110A0

**Hà Nội – 2025**

## LỜI NÓI ĐẦU

Nhập môn Công nghệ Phần mềm là môn học cung cấp những kiến thức cơ bản về quy trình phát triển phần mềm, giúp sinh viên hiểu và áp dụng các phương pháp, kỹ thuật trong việc thu thập yêu cầu, phân tích, thiết kế, cài đặt, cũng như xây dựng tài liệu phần mềm. Thông qua môn học này, sinh viên có thể nắm vững các giai đoạn trong quy trình phát triển phần mềm và sẵn sàng tham gia vào các dự án thực tế.

Dưới sự hướng dẫn của giảng viên ThS. Lê Hữu Dũng, nhóm chúng em đã thực hiện đề tài “Phát triển phần mềm theo lớp bài toán Quản lý hoạt động của một nhà thuốc đạt chuẩn GPP cho nhà thuốc Hồng Vân”. Đề tài này nhằm xây dựng một hệ thống hỗ trợ công tác quản lý nhà thuốc theo tiêu chuẩn GPP, giúp tối ưu hóa quy trình hoạt động và nâng cao hiệu quả quản lý.

Trong quá trình thực hiện, nhóm chúng em đã vận dụng những kiến thức được học để thiết kế và triển khai phần mềm theo đúng quy trình của Công nghệ phần mềm. Mặc dù đã cố gắng hoàn thành tốt nhất có thể, nhưng chắc chắn vẫn còn những hạn chế và thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp từ thầy và các bạn để có thể hoàn thiện sản phẩm hơn nữa.

Nhóm chúng em xin chân thành cảm ơn thầy Lê Hữu Dũng đã tận tình hướng dẫn, tạo điều kiện để nhóm có cơ hội thực hành và nâng cao kiến thức trong suốt học kỳ.

## 

## MỤC LỤC

## CÁC DANH MỤC

Hình ảnh:

[Hình 1.1 Các đề tài/hệ thống liên quan](#bookmark=id.ya8kjo3wka8w)

[Hình 1.2 Phương pháp phát triển](#bookmark=id.fdfbq0a8gfn3)

[Hình 4.1 Kiến trúc phân lớp - Layered pattern](#bookmark=id.mbg4o0k96hqu)

[Hình 4.2 Kiến trúc tổng thể của hệ thống](#bookmark=id.n9ea2ilfm3pe)

[Hình 4.3: Lược đồ cơ sở dữ liệu](#bookmark=id.i9b2scqobcsz)

Bảng biểu

[Bảng 1.1 Vai trò thành viên dự án](#bookmark=id.f56xxfzetuhx)

[Bảng 1.2 Phân công nhiệm vụ](#bookmark=kix.utxfjtsumfoq)

[Bảng 1.3 Kế hoạch triển khai chi tiết](#bookmark=id.ynttmhlbnvq7)

[Bảng 2.1 Danh mục tài liệu, sản phẩm](#bookmark=id.hzpoy47mdni4)

[Bảng 2.2 Quy định về Naming convention](#bookmark=id.q1bqtdkqyvfa)

[Bảng 2.3 Bảng định về Database](#bookmark=id.pqpbnrqqb7s7)

[Bảng 2.4 Mục tiêu chất lượng của dự án](#bookmark=id.9xi8qkppqu44)

[Bảng 2.5 Chi tiết hóa các tiêu chí của đặc tính kiểm thử được](#bookmark=id.bgylqantn06h)

[Bảng 3.1 Đánh giá chất lượng yêu cầu chức năng](#bookmark=id.gjxjad1olt2k)

[Bảng 3.2 Gom nhóm chức năng](#bookmark=id.69nt5bo67fno)

[Bảng 3.3 Đánh giá chất lượng Yêu cầu phi chức năng](#bookmark=id.q0oovq87yp69)

[Bảng 4.1 Chuyển đổi từ tên thực thể sang tên bảng](#bookmark=id.2jnmxh81zfgq)

[Bảng 4.2 Xác định các mối quan hệ](#bookmark=id.wuq3lkgdmajl)

[Bảng 4.3 Đặc tả dữ liệu hệ thống bảng nhân viên](#bookmark=id.uwdgs759ggx7)

[Bảng 4.4 Đặc tả dữ liệu hệ thống bảng nhà cung cấp](#bookmark=id.ppgclpw320iz)

[Bảng 4.5 Đặc tả dữ liệu hệ thống bảng khách hàng](#bookmark=kix.sx7rdxlrdcr4)

[Bảng 4.6 Đặc tả dữ liệu hệ thống bảng loại sản phẩm](#bookmark=kix.437tulu0i7wy)

[Bảng 4.7 Đặc tả dữ liệu hệ thống bảng sản phẩm](#bookmark=kix.s5znqxb955rp)

[Bảng 4.8 Đặc tả dữ liệu hệ thống lô sản phẩm](#bookmark=kix.dsw2nv49he5l)

[Bảng 4.9 Đặc tả dữ liệu hệ thống thuốc kê đơn](#bookmark=kix.jl8x80sfufos)

[Bảng 4.10 Đặc tả dữ liệu hệ thống chi tiết thuốc kê đơn](#bookmark=kix.xaddxdovuroq)

[Bảng 4.11 Đặc tả dữ liệu hệ thống hóa đơn](#bookmark=kix.91x04iq00amg)

[Bảng 4.12 Đặc tả dữ liệu hệ thống chi tiết hóa đơn](#bookmark=kix.nmuz8vkw7gec)

[Bảng 4.13 Đặc tả dữ liệu hệ thống khiếu nại](#bookmark=kix.itrdras5jh0a)

[Bảng 4.14 Đặc tả dữ liệu hệ thống nhập hàng](#bookmark=id.8bgjshfwe51f)

[Bảng 4.15 Đặc tả dữ liệu hệ thống lô sản phẩm](#bookmark=kix.pf8hpq7dtup3)

## CHƯƠNG 1. GIỚI THIỆU

### Tổng quan về đề tài

Phát triển phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GPP (Good Pharmacy Practices) theo công nghệ phần mềm là quá trình xây dựng hệ thống phần mềm nhằm tối ưu hóa hoạt động quản lý nhà thuốc, đảm bảo tuân thủ các tiêu chuẩn nghiêm ngặt về thực hành tốt nhà thuốc do Bộ Y tế quy định. Theo công nghệ phần mềm, việc phát triển này không chỉ dừng lại ở việc lập trình mà còn bao gồm nhiều khía cạnh khác nhau. Đầu tiên là nghiên cứu các phương pháp, quy trình, công cụ và kỹ thuật để phát triển, vận hành và bảo trì phần mềm một cách hiệu quả và chất lượng. Nó bao gồm các giai đoạn từ phân tích yêu cầu, thiết kế hệ thống, lập trình, kiểm thử đến triển khai và bảo trì phần mềm. Tiếp theo là đo lường các yếu tố như hiệu suất, độ tin cậy, khả năng bảo trì và tính bảo mật của hệ thống. Đối với phần mềm quản lý nhà thuốc, các chỉ số này đặc biệt quan trọng để đảm bảo dữ liệu thuốc, đơn thuốc và thông tin khách hàng được xử lý nhanh chóng, chính xác và an toàn. Từ đó, đặt ra yêu cầu phải tuân thủ chặt chẽ các quy trình, tiêu chuẩn và nguyên tắc kỹ thuật. Tính kỷ luật giúp đảm bảo phần mềm được phát triển có hệ thống, giảm thiểu lỗi và dễ dàng bảo trì, nâng cấp. Đặc biệt với phần mềm quản lý nhà thuốc đạt chuẩn GPP, tính kỷ luật còn đảm bảo tuân thủ các quy định pháp lý và quy trình kiểm soát chất lượng dược phẩm.

### Giới thiệu bài toán

Việc quản lý bán thuốc đạt chuẩn GPP (Good Pharmacy Practices) đặt ra những yêu cầu khắt khe về tính chính xác, an toàn và minh bạch trong quy trình hoạt động của nhà thuốc. Tiêu chuẩn GPP là một trong những tiêu chuẩn thực hành tốt trong ngành dược, được Bộ Y tế quy định nhằm đảm bảo chất lượng dịch vụ cung ứng thuốc đến người tiêu dùng. Tiêu chuẩn GPP yêu cầu các cơ sở bán lẻ thuốc phải tuân thủ các nguyên tắc về bảo quản và phân phối thuốc (Bộ Y tế, 2018).Thông tư này áp dụng cho các nhà thuốc, quầy thuốc và tủ thuốc trạm y tế xã, yêu cầu các cơ sở này phải tuân thủ các tiêu chuẩn về cơ sở vật chất, trang thiết bị, nhân sự, quy trình bảo quản và phân phối thuốc, cũng như tư vấn và hướng dẫn sử dụng thuốc cho người bệnh.

Dù nhiều nhà thuốc đã áp dụng các phần mềm quản lý cơ bản, nhưng khi đối mặt với các tiêu chuẩn nghiêm ngặt của GPP thì luôn phải đáp ứng các quy định:

1. **Quản lý thuốc theo quy định của bộ y tế**

Theo I - 1a.II.4.b của **Thông tư 02/2018/TT-BYT**

*“Phải có sổ sách hoặc máy tính để quản lý việc nhập, xuất, tồn trữ, theo dõi số lô, hạn dùng, nguồn gốc của thuốc và các thông tin khác có liên quan, bao gồm:*

*- Thông tin thuốc: Tên thuốc, số Giấy phép lưu hành/Số Giấy phép nhập khẩu, số lô, hạn dùng, nhà sản xuất, nhà nhập khẩu, điều kiện bảo quản.*

*- Nguồn gốc thuốc: Cơ sở cung cấp, ngày tháng mua, số lượng;*

*- Cơ sở vận chuyển, điều kiện bảo quản trong quá trình vận chuyển;*

*- Số lượng nhập, bán, còn tồn của từng loại thuốc;*

*- Người mua/bệnh nhân, ngày tháng, số lượng (đối với thuốc gây nghiện, thuốc tiền chất, thuốc hướng thần, thuốc dạng phối hợp có chứa dược chất gây nghiện, dược chất hướng thần, tiền chất);*

*- Đối với thuốc kê đơn phải thêm người kê đơn và cơ sở hành nghề.”*

Theo I - 1a.II.4.c của **Thông tư 02/2018/TT-BYT**

*“[...] bảo đảm kiểm soát xuất xứ, giá cả, nguồn gốc thuốc mua vào, bán ra. Có cơ chế chuyển thông tin về việc mua bán thuốc, chất lượng thuốc giữa nhà cung cấp với khách hàng cũng như việc chuyển giao thông tin cho cơ quan quản lý liên quan khi được yêu cầu.”*

**Yêu cầu:** Phần mềm cần hỗ trợ quản lý về thuốc, Hệ thống nhà thuốc phải có sổ sách hoặc hệ thống máy tính để quản lý việc nhập, xuất, tồn kho và theo dõi đầy đủ thông tin thuốc, bao gồm: tên thuốc, số Giấy phép lưu hành/Số Giấy phép nhập khẩu, số lô, hạn dùng, nhà sản xuất, nhà nhập khẩu, điều kiện bảo quản, cơ sở cung cấp, ngày tháng mua và số lượng nhập. Ngoài ra, hệ thống cần ghi nhận thông tin vận chuyển, tình trạng kho thuốc, cũng như thông tin người mua và đơn thuốc kê đơn đối với thuốc kiểm soát đặc biệt. Để đáp ứng quy định, phần mềm quản lý nhà thuốc phải hỗ trợ lưu trữ, đảm bảo tính toàn vẹn dữ liệu, đồng bộ trên nhiều thiết bị và tích hợp với hệ thống thanh tra nhằm phục vụ công tác kiểm tra và giám sát. Thách thức nằm ở việc đảm bảo tính toàn vẹn dữ liệu, đồng bộ trên nhiều thiết bị và tích hợp với các hệ thống bên ngoài.

1. **Quản lý kho thuốc và kiểm soát tồn kho**

Theo I - 1a.III.1 của **Thông tư 02/2018/TT-BYT**

*a) Nguồn thuốc được mua tại các cơ sở kinh doanh thuốc hợp pháp.*

*b) Có hồ sơ theo dõi, lựa chọn các nhà cung cấp có uy tín, đảm bảo chất lượng thuốc trong quá trình kinh doanh;*

*c) Chỉ mua các thuốc được phép lưu hành (thuốc có số đăng ký hoặc thuốc chưa có số đăng ký được phép nhập khẩu). Thuốc mua còn nguyên vẹn và có đầy đủ bao gói của nhà sản xuất, nhãn đúng quy định theo quy chế hiện hành. Có đủ hóa đơn, chứng từ hợp lệ của thuốc mua về;*

*d) Khi nhập thuốc, người bán lẻ kiểm tra hạn dùng, kiểm tra các thông tin trên nhãn thuốc theo quy chế ghi nhãn, kiểm tra chất lượng (bằng cảm quan, nhất là với các thuốc dễ có biến đổi chất lượng) và có kiểm soát trong suốt quá trình bảo quản;*

**Yêu cầu**: đảm bảo các yêu cầu về nguồn gốc, chất lượng và lưu hành của thuốc. Phần mềm cần theo dõi nguồn gốc thuốc, hồ sơ nhà cung cấp, chỉ mua thuốc được phép lưu hành, còn nguyên vẹn, nhãn đúng quy định và có hóa đơn, chứng từ hợp lệ. Khi nhập thuốc, phần mềm cần hỗ trợ kiểm tra hạn dùng, nhãn, chất lượng và kiểm soát quá trình bảo quản. Về quản lý kho, phần mềm cần theo dõi tình trạng tồn kho, kiểm soát thuốc hết hạn, hỗ trợ nhập xuất kho tránh thiếu hoặc dư thuốc, đặc biệt là thuốc có hạn sử dụng ngắn. Ngoài ra, phần mềm cần có các chức năng cảnh báo, báo cáo, tìm kiếm và phân quyền truy cập.

1. **Quản lý thuốc khiếu nại, thuốc không được phép lưu hành, thuốc phải thu hồi**

Theo I - 1a.III.4.c của **Thông tư 02/2018/TT-BYT**

*- Phải có hệ thống lưu giữ các thông tin, thông báo về thuốc khiếu nại, thuốc không được phép lưu hành, thuốc phải thu hồi;*

*- Có thông báo thu hồi cho khách hàng đối với các thuốc thuộc danh mục thuốc kê đơn. Biệt trữ các thuốc thu hồi để chờ xử lý;*

*- Có hồ sơ ghi rõ về việc khiếu nại và biện pháp giải quyết cho người mua về khiếu nại hoặc thu hồi thuốc;*

*- Đối với thuốc cần hủy phải chuyển cho cơ sở có chức năng xử lý chất thải để hủy theo quy định*

*- Có báo cáo các cấp theo quy định*

Yêu cầu: Phần mềm cần có hệ thống lưu giữ thông tin chi tiết về các loại thuốc này, bao gồm cả thông báo thu hồi cho khách hàng đối với thuốc kê đơn và biệt trữ thuốc thu hồi để chờ xử lý. Hồ sơ ghi rõ về việc khiếu nại và biện pháp giải quyết cho người mua cũng cần được lưu trữ đầy đủ. Đối với thuốc cần hủy, phần mềm cần đảm bảo việc chuyển giao cho cơ sở có chức năng xử lý chất thải để hủy theo quy định. Cuối cùng, phần mềm cần hỗ trợ tạo báo cáo các cấp theo quy định. Ngoài ra, phần mềm nên có các chức năng bổ sung như cảnh báo khi có thuốc thuộc diện cần quản lý đặc biệt, truy xuất nguồn gốc thuốc, quản lý kho thuốc biệt trữ và tạo báo cáo thống kê. Giao diện phần mềm cần thân thiện, dễ sử dụng, tuân thủ quy định về bảo mật và an toàn dữ liệu, đồng thời có khả năng tích hợp với các hệ thống khác nếu cần.*.*

1. **Bảo mật và an toàn thông tin bệnh nhân**

Theo I - 1a.III.4.a của **Thông tư 02/2018/TT-BYT**

*“[...]- Giữ bí mật các thông tin của người bệnh trong quá trình hành nghề như bệnh tật, các thông tin người bệnh yêu cầu; [...]”*

**Yêu cầu:** Phần mềm cần bảo vệ thông tin cá nhân và dữ liệu liên quan đến bệnh nhân một cách an toàn. Thông tin cá nhân của khách hàng và bệnh nhân có thể nhằm phục vụ nhu cầu phân tích, dự đoán giúp dược sĩ có thể theo dõi kê đơn phù hợp tăng hiệu quả làm việc song đồng thời chúng cũng dễ bị lợi dụng hoặc lợi dụng sai mục đích. Cần phân quyền rõ ràng để chỉ các dược sĩ hợp pháp mới có quyền truy cập và cấp thuốc.

### Mục tiêu của đề tài

Các yêu cầu chính bao gồm: quản lý thuốc theo quy định của Bộ Y tế; quản lý kho thuốc và kiểm soát tồn kho; quản lý thuố‌c khiếu nại, thuốc không được phép lưu hành, thuốc phải thu hồi; bảo mật và an toàn thông tin bệnh nhân. Dựa trên các yêu cầu trên, đề tài sẽ tập trung phát triển chức năng quản lý kho thuố‌c và kiểm soát tồn kho, cụ thể:

Mục tiêu 1: Xây dựng các chức năng kiểm soát nguồn gốc, chất lượng và lưu hành thuốc, đảm bảo tuân thủ quy định về nhập kho, xuất kho {Trước ngày 05/03/2025};

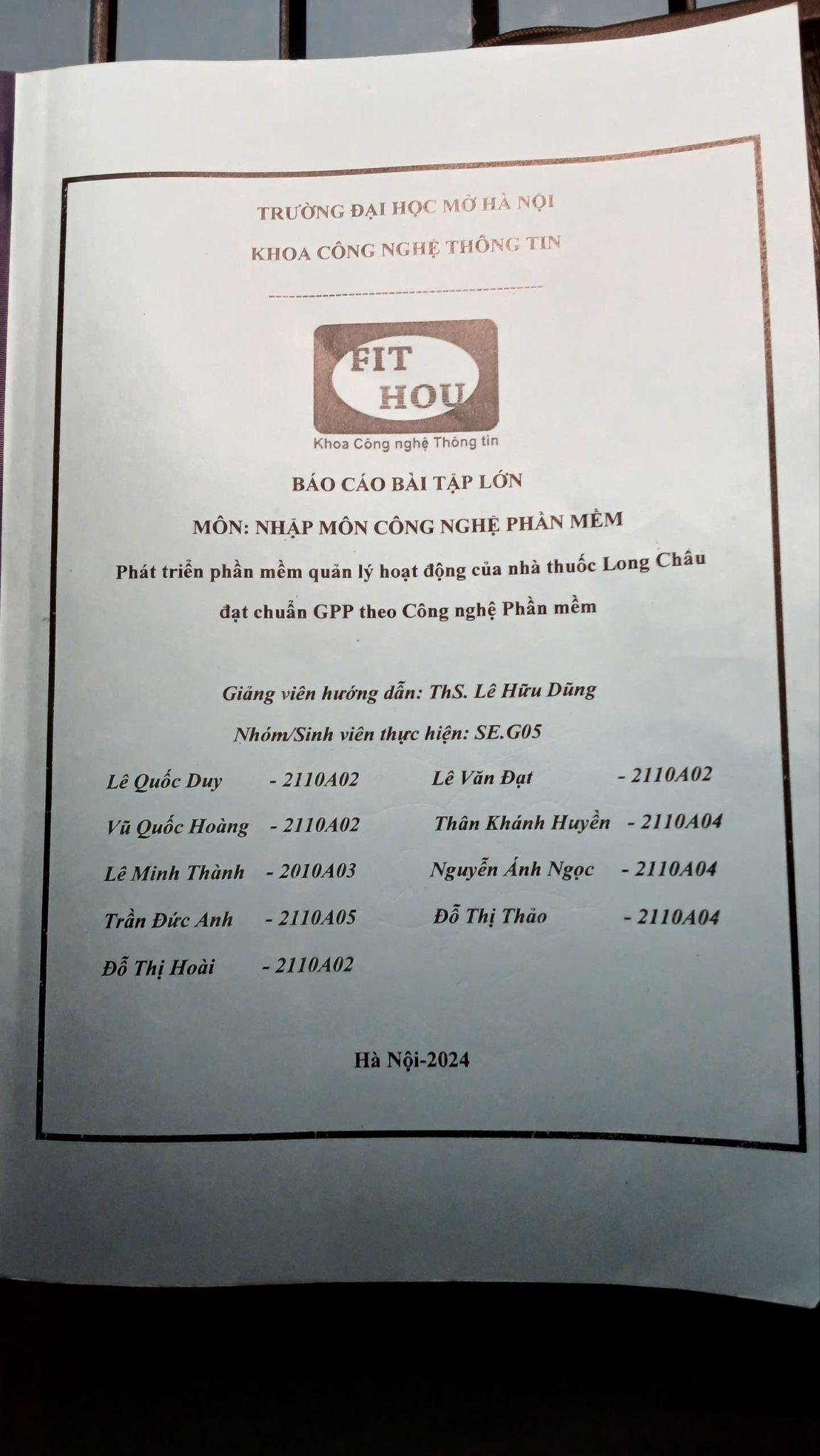
Mục tiêu 2: Thiết kế hệ thống quản lý tồn kho với các tính năng giám sát và cảnh báo về thuốc hết hạn, tồn kho quá mức hoặc thiếu hụt {Trước ngày 05/03/2025};

Mục tiêu 3: Phát triển các chức năng báo cáo, tìm kiếm và phân quyền truy cập để nâng cao hiệu quả quản lý {Trước ngày 05/03/2025};

Mục tiêu 4: Xây dựng và thực hiện quy trình kiểm thử phần mềm, đáp ứng đầy đủ yêu cầu về quản lý kho thuốc và kiểm soát tồn kho {Trước ngày 05/03/2025}

### Các đề tài/hệ thống liên quan

Minh chứng:

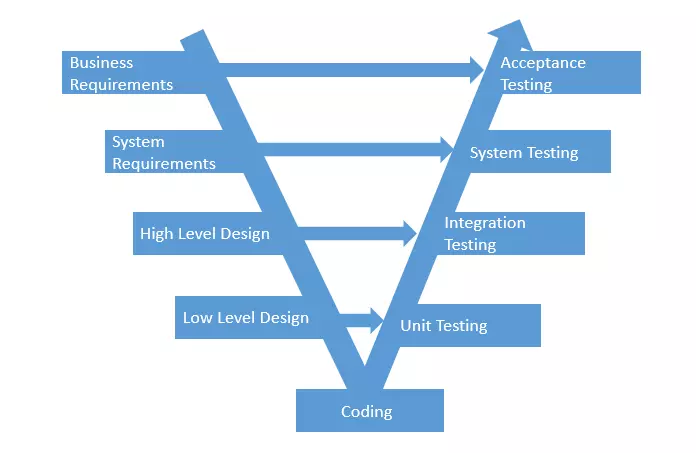


Hình 1.1 Các đề tài/hệ thống liên quan

**Ưu điểm:** Hệ thống đề xuất giúp định hướng rõ ràng các mục tiêu và phạm vi cần triển khai, đảm bảo tập trung vào những yếu tố cốt lõi của đề tài. Trong quá trình xây dựng, hệ thống xác định và xử lý các nghiệp vụ quan trọng như quản lý chính xác thông tin về thuốc, khách hàng và nhà cung cấp, giúp dữ liệu được tổ chức một cách khoa học và dễ truy xuất. Ngoài ra, cơ chế phân quyền được thiết kế hợp lý, đảm bảo mỗi cá nhân có quyền hạn phù hợp theo vai trò và chức vụ của mình. Bên cạnh đó, hệ thống hỗ trợ các chức năng quan trọng như kiểm kê, báo cáo và trích xuất dữ liệu nhanh chóng, giúp tối ưu hóa quá trình quản lý. Nhờ tự động hóa nhiều công đoạn, hệ thống góp phần giảm bớt công việc thủ công, tiết kiệm thời gian, đồng thời mang lại trải nghiệm thân thiện, dễ sử dụng và có khả năng tích hợp với các hệ thống khác khi cần thiết.

**Nhược điểm:** Mặc dù hệ thống đã xác định được các chức năng quan trọng, nhưng vẫn chưa có sự phân chia rõ ràng về thứ tự thực hiện, các mục tiêu cần hoàn thành cũng như thời gian dự kiến để hoàn tất từng phần công việc. Điều này có thể ảnh hưởng đến tiến độ và khả năng quản lý dự án một cách hiệu quả. Bên cạnh đó, một số nghiệp vụ vẫn chưa được hoàn thiện đầy đủ, đặc biệt là các vấn đề liên quan đến bảo mật thông tin. Việc đảm bảo an toàn dữ liệu, bảo vệ quyền riêng tư của người dùng và ngăn chặn các rủi ro tiềm ẩn vẫn cần được nghiên cứu và cải thiện thêm trong quá trình triển khai.

### Phương pháp phát triển



Hình 1.2 Phương pháp phát triển

#### Tổng quan Mô hình - V Model:

Mô hình chữ V là một phương pháp phát triển phần mềm được phát triển dựa trên mô hình thác nước, nhưng với cấu trúc được chia thành hai nhánh chính: Phát triển và Kiểm thử. Đặc điểm nổi bật của mô hình này là mỗi bước trong quá trình phát triển đều đi kèm với một hoạt động kiểm thử tương ứng được thực hiện song song. Nguyên tắc cốt lõi của V-model là tích hợp kiểm thử ngay từ những giai đoạn đầu của quy trình, ví dụ như việc lập kế hoạch kiểm thử toàn hệ thống được thực hiện đồng thời với quá trình phân tích và thiết kế. Điều này giúp phát hiện và xử lý sớm các vấn đề tiềm ẩn, từ đó nâng cao chất lượng sản phẩm cuối cùng.

#### Ưu nhược điểm của mô hình V-model

**Ưu điểm:** Mô hình chữ V nổi bật nhờ vào cấu trúc tổ chức và hệ thống trong quy trình phát triển phần mềm. Nhờ việc kiểm thử được thực hiện song song với các giai đoạn phát triển, mô hình giúp tiết kiệm thời gian và sớm phát hiện những điểm mơ hồ, từ đó giảm thiểu sai sót về sau. Phương pháp này rất thích hợp cho các dự án có quy mô vừa và nhỏ, khi yêu cầu và phạm vi của dự án đã được xác định rõ ràng từ đầu. Thêm vào đó, việc chia nhỏ dự án thành các giai đoạn với mục tiêu cụ thể tạo điều kiện thuận lợi cho việc quản lý và định hướng công việc cho nhóm.

**Nhược điểm:** Tuy có nhiều ưu điểm, mô hình chữ V lại không phù hợp với các dự án có quy mô lớn, phức tạp hoặc có yêu cầu thay đổi liên tục, vì bất kỳ sự điều chỉnh nào cũng đòi hỏi phải quay lại các bước trước đó và cập nhật lại toàn bộ tài liệu. Hơn nữa, trong quá trình phát triển, mô hình không tạo ra sản phẩm phần mềm hoàn chỉnh ở các giai đoạn trung gian, điều này gây khó khăn trong việc tiến hành kiểm thử sớm. Đồng thời, mô hình không hỗ trợ tốt việc phát triển song song với triển khai và bán sản phẩm, và thiếu các biện pháp phân tích rủi ro cụ thể, khiến cho dự án dễ bị ảnh hưởng bởi các yếu tố bất ngờ, đặc biệt khi có những thay đổi về kỹ thuật trong quá trình thực hiện.

#### Ứng dụng phương pháp phát triển V Model trong đề tài

Quy trình phát triển hệ thống quản lý hoạt động cho hiệu thuốc Hồng Vân theo mô hình chữ V được khởi đầu bằng giai đoạn thu thập yêu cầu. Trong bước này, các nhu cầu và tiêu chí của hệ thống được thu thập một cách tỉ mỉ từ nhiều bên liên quan như khách hàng, nhân viên và các tiêu chuẩn GPP theo thông tư của Bộ Y Tế. Việc làm rõ các yêu cầu từ ban đầu giúp định hướng đúng đắn cho toàn bộ quá trình phát triển sau này.

Dựa trên các yêu cầu đã xác định, hệ thống chuyển sang giai đoạn thiết kế. Ở đây, các tính năng và chức năng cần có của hệ thống được lên kế hoạch chi tiết, bao gồm việc xây dựng kiến trúc hệ thống, thiết kế các module riêng biệt, cơ sở dữ liệu và giao diện người dùng, nhằm đảm bảo hệ thống có khả năng đáp ứng đúng nhu cầu thực tế. Cùng lúc đó, kế hoạch kiểm thử cũng được soạn thảo song song, nhằm chuẩn bị các hoạt động kiểm thử từ đơn vị, tích hợp, đến kiểm thử tổng thể và kiểm thử chấp nhận của người dùng (UAT) nhằm đảm bảo chất lượng ngay từ các giai đoạn đầu.

Sau khi hoàn thiện giai đoạn thiết kế, quy trình phát triển hệ thống được triển khai với sự phối hợp chặt chẽ giữa các đội phát triển và kiểm thử. Trước tiên, kiểm thử đơn vị (Unit Testing) được tiến hành để đảm bảo từng module hoạt động chính xác riêng lẻ. Sau đó, kiểm thử tích hợp (Integration Testing) được thực hiện nhằm kiểm tra sự tương tác và liên kết giữa các module khác nhau. Khi toàn bộ hệ thống đã được xây dựng hoàn chỉnh, kiểm thử hệ thống (System Testing) được thực hiện để đánh giá toàn diện hoạt động của hệ thống theo đúng yêu cầu đã đề ra. Cuối cùng, kiểm thử chấp nhận người dùng (UAT) cho phép người dùng cuối trải nghiệm và đánh giá, đảm bảo hệ thống đáp ứng được yêu cầu kinh doanh và vận hành một cách ổn định.

Khi hệ thống vượt qua tất cả các giai đoạn kiểm thử và nhận được sự chấp thuận, nó sẽ được triển khai chính thức vào môi trường sản xuất. Quá trình duy trì và bảo trì sau triển khai được thực hiện liên tục để đảm bảo hệ thống luôn hoạt động ổn định, cũng như cập nhật, nâng cấp theo các yêu cầu mới hoặc điều chỉnh từ thực tế vận hành.

### Vai trò của thành viên dự án

| **STT** | **Họ và tên** | **rAI** | **DT** |
| --- | --- | --- | --- |
| 1 | Nguyễn Hồng Ánh | x |  |
| 2 | Nguyễn Nguyên Vũ | x |  |
| 3 | Nguyễn Mai Hương |  | x |
| 4 | Đỗ Đan Trường | x |  |
| 5 | Nguyễn Hải Đăng | x |  |
| 6 | Trần Thị Mai Thi |  | x |
| 7 | Nguyễn Văn Mạnh | x |  |
| 8 | Trần Tuấn Huy | x |  |
| 9 | Nguyễn Minh Sơn | x |  |

Bảng 1.1 Vai trò thành viên dự án

### Phân công nhiệm vụ

| **STT** | **Công việc** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Tổng quan đề tài |  |  |  |  |  |  |  |  |  |
| 2 | Giới Thiệu Bài Toán | x |  |  |  |  |  |  |  |  |
| 3 | Mục tiêu đề tài | x |  |  |  |  |  |  |  |  |
| 4 | Đề tài hệ thống liên quan |  | x |  |  |  | x |  |  |  |
| 5 | Thu thập yêu cầu | x |  |  | x |  | x |  |  |  |
| 6 | Xác định cơ cấu tổ chức | x |  |  |  |  |  |  |  |  |
| 7 | Khảo sát hệ thống |  |  |  | x |  | x |  |  |  |
| 8 | Mô tả các bên liên quan | x | x |  | x |  |  | x |  | x |
| 9 | Quy trình nghiệp vụ |  |  |  |  |  |  |  | x |  |
| 10 | Xác định yêu cầu chức năng |  | x | x |  |  | x |  |  |  |
| 11 | Xác định yêu cầu phi chức năng |  |  |  | x |  | x |  |  |  |
| 12 | Xác định các quy tắc chung | x |  |  |  |  |  |  |  |  |
| 13 | Xác định mục tiêu chất lượng dự án | x |  |  |  |  |  |  |  |  |
| 14 | Xác định các tiêu chuẩn kiểm thử | x |  | x |  |  | x |  |  |  |
| 15 | Lập danh sách yêu cầu hệ thống |  | x | x | x |  | x | x |  | x |
| 16 | Đánh giá chất lượng yêu cầu |  | x | x | x |  | x | x |  | x |
| 17 | Xây dựng cơ sở dữ liệu mức vật lý |  |  |  |  |  |  | x |  | x |
| 18 | Thiết kế một số chi tiết |  | x |  |  | x |  |  | x |  |
| 19 | Đặc tả chức năng mức lá |  |  |  |  |  |  |  |  |  |
| 20 | Phân tích hệ thống về dữ liệu |  |  |  |  |  |  | x |  | x |
| 21 | Kiến trúc hệ thống | x |  |  |  |  |  |  |  |  |
| 22 | Thiết kế cơ sở dữ liệu |  |  |  |  |  |  | x |  | x |
| 23 | Giao diện các chức năng | x | x |  | x |  | x | x | x | x |
| 24 | Phát triển các chức năng |  | x |  | x |  | x | x | x | x |
| 25 | Kiểm thử các chức năng |  |  | x |  |  | x |  |  |  |

Bảng 1.2 Phân công nhiệm vụ

### Kế hoạch triển khai chi tiết

| STT | Công việc | Kết quả như nào là đạt | Ngày bắt đầu | Ngày kết thúc | Cách thực hiện |
| --- | --- | --- | --- | --- | --- |
| 1 | Khảo sát thông tin về tổ chức, phỏng vấn trực tiếp, xin biểu mẫu | Hoàn thành tài liệu khảo sát và thêm vào mục phụ lục của SRS: Biết được thông tin tổ chức, biểu mẫu cần thiết | 13/2/25 | 15/02/25 | Cần tiến hành khảo sát thông tin từ các nguồn đáng tin cậy, yêu cầu cung cấp các biểu mẫu, tài liệu có sẵn để thu thập thông tin chi tiết và chính xác. |
| 2 | Quy trình nghiệp vụ | Xác định rõ các bước, vai trò, dữ liệu đầu vào/đầu ra, quy tắc nghiệp vụ và các yêu cầu để đảm bảo hệ thống đáp ứng đúng nhu cầu | 13/2/25 | 16/02/25 | Cần xác định rõ các bước, vai trò, nguồn lực và các kết quả đầu ra của từng hoạt động trong quy trình. |
| 3 | Xác định các yêu cầu chức năng và phi chức năng | Yêu cầu chức năng và phi chức năng đạt chuẩn phải rõ ràng, cụ thể, có thể đo lường, kiểm tra, truy vết, không mơ hồ, không mâu thuẫn và khả thi trong thực tế. | 13/2/25 | 16/02/25 | Tài liệu hóa yêu cầu dưới dạng đặc tả, sử dụng mô hình như Use Case, SRS, |
| 4 | Biểu đồ và đặc tả use case | Thể hiện rõ các tác nhân, luồng tương tác, mô tả chi tiết hành vi hệ thống, kịch bản chính/phụ, và các ràng buộc để đảm bảo tính đầy đủ, chính xác và dễ hiểu. | 13/2/25 | 16/02/25 | xác định tác nhân, phân tích chức năng hệ thống, vẽ biểu đồ Use Case minh họa quan hệ, sau đó viết đặc tả chi tiết cho từng Use Case theo kịch bản chính, kịch bản phụ và điều kiện ràng buộc. |
| 5 | Xác định tiêu chuẩn, phương pháp đánh giá chất lượng yêu cầu | Tiêu chí khi xác định tiêu chuẩn, phương pháp đánh giá chất lượng yêu cầu bao gồm tính đầy đủ, rõ ràng, khả thi , đo lường được , khả năng. kiểm thử và phù hợp với mục tiêu và yêu cầu người dùng . | 16/2/25 | 17/2/25 | Để thực hiện tiêu chí này, cần xác định các yêu cầu cụ thể, đảm bảo chúng có thể kiểm tra, đo lường được và phù hợp với mục tiêu, đồng thời sử dụng các công cụ và phương pháp kiểm thử để đánh giá tính đầy đủ, rõ ràng và khả thi của yêu cầu. |
| 6 | Xác định các quy tắc chung và kiểm thử của dự án | rõ ràng, nhất quán, dễ tuân thủ và phù hợp với mục tiêu của dự án | 17/2/25 | 20/2/25 | Xây dựng, tài liệu hóa quy tắc chung, phổ biến cho đội ngũ và giám sát việc tuân thủ trong quá trình phát triển dự án. |
| 7 | Lập danh sách yêu cầu hệ thống | danh sách yêu cầu rõ ràng, đầy đủ, không mâu thuẫn và được tất cả các bên liên quan phê duyệt. | 19/2/25 | 22/2/25 | Thu thập yêu cầu từ khách hàng, phân tích, viết tài liệu đặc tả (SRS) và xác nhận với các bên liên quan. |
| 8 | Đánh giá chất lượng yêu cầu | yêu cầu không mâu thuẫn, có thể thực hiện, đo lường và đáp ứng được mục tiêu hệ thống. | 20/2/25 | 22/2/25 | So sánh dựa theo tiêu chuẩn đề ra |
| 9 | Lựa chọn công nghệ và kiến trúc phát triển phần mềm | công nghệ và kiến trúc được lựa chọn đáp ứng tốt hiệu suất, bảo mật, khả năng mở rộng và dễ bảo trì. | 23/2/25 | 23/2/25 | Phân tích yêu cầu, so sánh các công nghệ, đánh giá ưu nhược điểm và thống nhất lựa chọn với nhóm. |
| 10 | Xây dựng cơ sở dữ liệu mức vật lý | cơ sở dữ liệu được triển khai đúng thiết kế, tối ưu truy vấn, đảm bảo tính toàn vẹn dữ liệu và hoạt động ổn định. | 18/2/25 | 22/2/25 | Chuyển mô hình dữ liệu logic thành lược đồ vật lý, tạo bảng, chỉ mục, thiết lập quan hệ và tối ưu hiệu suất lưu trữ. |
| 11 | Thiết kế 1 số chức năng | thiết kế đảm bảo tính chính xác, khả thi, dễ hiểu và đáp ứng yêu cầu nghiệp vụ. | 19/2/25 | 22/2/25 | Phân tích yêu cầu, xây dựng luồng xử lý |
| 12 | Kiến trúc phần mềm | Kiến trúc đảm bảo tính chính xác khả thi | 20/2/25 | 25/2/25 | Dựa vào yêu cầu, lựa chọn kiến trúc |
| 13 | Lập trình các chức năng đã thiết kế | Đảm bảo 80% yêu cầu, thiết kế đề ra | 23/2/25 | 1/3/25 | Dựa theo công nghệ đã lựa chọn |
| 14 | Lập trình kiểm thử | Soure code được đảm bảo theo kiến trúc phần mềm, đáp ứng đầy đủ các chức năng, phi chức năng | 25/2/25 | 7/3/25 | Code các chức năng theo phân công |

Bảng 1.3 Kế hoạch triển khai chi tiết

## CHƯƠNG 2. SẢN PHẨM CHUYỂN GIAO GIỮA CÁC PHA VÀ TIÊU CHUẨN ĐÁNH GIÁ

### Danh mục tài liệu, sản phẩm

| **STT** | **Loại tài liệu/Sản phẩm** | **Biểu mẫu** | **Tiêu chuẩn đánh giá** | **R** | **A** | **D** | **I** | **T** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Bản khảo sát về quy trình nghiệp vụ | [Artifact: Analysis Model](https://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/workproducts/rup_analysis_model_76C3B70C.html) | [**IEEE 29148:2018**](https://standards.ieee.org/ieee/29148/6937/) đưa ra các yêu cầu: *Tính đầy đủ (Completeness)* **-** Đảm bảo nội dung khảo sát bao quát mọi khía cạnh cần thiết của dự án/sản phẩm và thu thập đầy đủ thông tin từ người dùng, khách hàng và các bên liên quan. *Tính chính xác (Accuracy)* - phản ánh đúng nhu cầu. *Tính rõ ràng (Clarity)* **-** dễ hiểu, có biểu mẫu hỗ trợ. *Tính nhất quán (Consistency)* **-** không mâu thuẫn, khớp với tài liệu khác, *Tính phù hợp (Relevance)* - phù hợp mục tiêu dự án và *Tính minh bạch (Transparency)* - rõ ràng về phương pháp, quy trình. | Out | In |  |  |  |
| 2 | Xác định cơ cấu tổ chức | [Business Architecture Document](https://www.edrawmax.com/templates/1006061/) | Cơ cấu tổ chức gồm: Tên bộ phận, quy trình nghiệp vụ của từng bộ phận. Có minh chứng qua khảo sát thực tế. | Out | In |  |  |  |
| 3 | Danh sách các chức năng và phi chức năng | [On Non-Functional Requirements](https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/) | Yêu cầu chức năng được đề cập đầy đủ dựa vào tài liệu khảo sát, phi chức năng phải đầy đủ, rõ ràng yêu cầu phi chức năng | Out | In | In | In |  |
| 4 | Tài liệu đặc tả quy trình nghiệp vụ | [SRS Document](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document) | Tài liệu phải đầy đủ, rõ ràng, nhất quán, khả thi, kiểm thử được, theo dõi được và tuân thủ quy định. Mô tả chi tiết quy trình với đầu vào, đầu ra, bước trung gian, tiêu chí đánh giá và liên kết với yêu cầu nghiệp vụ, đảm bảo áp dụng thực tế và không mâu thuẫn. | Out | In |  |  | In |
| 5 | Tài liệu đặc tả yêu cầu | Tài liệu ghi rõ ràng chi tiết từng yêu cầu: lý do, mục tiêu, các định nghĩa, tóm tắt, tài liệu tham khảo của từng yêu cầu, luồng sự kiện | Out |  |  |  | In |
| 6 | Đặc tả tiến trình nghiệp vụ | [Business Use-Case Realization Specification](https://files.defcon.no/RUP/webtmpl/templates/bm/rup_bucr.htm) | Các tiến trình nghiệp vụ phải khớp với cơ cấu tổ chức được khảo sát thực tế (tài liệu khảo sát) | Out | In |  |  |  |
| 7 | Sơ đồ phân rã chức năng | [Functional Decomposition Diagram](https://www.geeksforgeeks.org/functions-decomposition-in-software-engineering/) | Xác định phạm vi của hệ thống cần phân tích, biểu diễn chức năng trực tiếp để tìm ra các chức năng thiếu hoặc trùng lặp. |  | Out | In | In |  |
| 8 | Đặc tả các chức năng mức lá | [Leaf-Level Functional Specification](https://www.techtarget.com/searchsoftwarequality/definition/functional-specification) | Đặc tả chi tiết từng chức năng mức lá được đề cập ở sơ đồ phân rã chức năng |  | Out | In | In | In |
| 9 | Thiết kế cơ sở dữ liệu | [Database Design](https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5) | CSDL có đầy đủ các trường thông tin, các bảng được đặt tên rõ ràng theo quy tắc đặt tên, đạt chuẩn BCNF, được chuẩn hóa ERD sang RDB |  |  | Out | In |  |
| 10 | Đặc tả CSDL | [Database Specification](https://radekp.github.io/qtmoko/api/database-specification.html) | Đặc tả toàn bộ CSDL đã đưa ra từ bước thiết kế CSDL |  |  | Out | In |  |
| 11 | Bản thiết kế tổng thể | [Detailed Design Document](https://senior.ceng.metu.edu.tr/2012/dbug/documents/DDR.pdf) | Mô hình thiết kế tổng quan; đưa ra được khảo sát thiết kế mô hình; có đánh giá thông qua của khách hàng và PM dự án. |  |  | Out | In | In |
| 12 | Bản thiết kế chi tiết | Chính xác, đầy đủ, nhất quán, khả thi, dễ bảo trì và đáp ứng yêu cầu hệ thống. |  |  | Out | In |  |
| 13 | Mã nguồn |  | Source code đáp ứng được các độ đo về mật độ lỗi, độ đo tính nghiêm trọng của lỗi |  |  |  | Out | In |
| 14 | Lập kế hoạch kiểm thử | [Artifact: Test Results](http://files.defcon.no/RUP/process/artifact/ar_tstrs.htm) | đầy đủ phạm vi kiểm thử, rõ ràng về mục tiêu, phương pháp, tiêu chí chấp nhận, lịch trình và nguồn lực thực hiện |  |  | Out |  | In |
| 15 | Lập kịch bản kiểm thử cho các chức năng | rõ ràng, đầy đủ các trường hợp kiểm thử, có dữ liệu đầu vào, bước thực hiện, kết quả mong đợi và bao quát các tình huống quan trọng. |  |  | Out |  | In |
| 16 | Báo cáo kiểm thử cho các chức năng | đầy đủ thông tin về phạm vi kiểm thử, kết quả thực tế, số lượng lỗi, mức độ nghiêm trọng, và đề xuất xử lý, đảm bảo tính chính xác và minh bạch. |  |  |  | In | Out |

Bảng 2.1 Danh mục tài liệu, sản phẩm

### Quy tắc đặt tên (naming convention)

#### Quy tắc chung

Tất cả các tên trong hệ thống, bao gồm tên class, tên biến, tên hằng số, tên hàm và tên file, phải được đặt bằng tiếng Anh hoặc tiếng Việt không dấu, đảm bảo đúng chính tả và có ý nghĩa. Tên phải phản ánh đúng đặc điểm hoặc chức năng của đối tượng tương ứng, ví dụ: "username" để thể hiện tên người dùng, "image" để thể hiện ảnh. Các biến chạy trong vòng lặp (for, while) không bắt buộc phải có nghĩa cụ thể, vì các biến như "i", "j" đã trở thành tiêu chuẩn chung.

Tên class và interface nên được đặt bằng tiếng Anh, viết hoa chữ cái đầu tiên của mỗi từ và không sử dụng dấu gạch dưới. Đối với tên biến, nên sử dụng kiểu camel case, trong đó chữ cái đầu tiên của từ đầu tiên viết thường, và chữ cái đầu tiên của các từ tiếp theo viết hoa. Trong khi đó, tên hằng số phải được viết hoa hoàn toàn và sử dụng dấu gạch dưới để phân tách các từ nhằm đảm bảo tính dễ đọc và tuân thủ quy ước đặt tên.

#### Quy định về Naming convention

| **Khai báo** | **Quy định** | **Ví dụ** |
| --- | --- | --- |
| Class | Sử dụng PascalCase cho tên lớp | CustomerController |
| Interface | Đối với interface, thêm tiền tố “I” | IOrderRepository, ICustomerService |
| Phương thức, Hàm | Sử dụng PascalCase | GetMedicineById, SaveOrder |
| Tên biến, Tham số | Sử dụng camelCase cho tên biến và tham số | customerId, orderDate |
| Hằng số | Viết hoa tất cả ký tự, các từ cách nhau bằng dấu gạch dưới | MAX\_USER\_COUNT, DEFAULT\_TIMEOUT, MIN\_PASSWORD\_LENGTH |
| Namespaces | Cấu trúc theo mô-đun hoặc tầng của ứng dụng | CompanyName.ProjectName.ModuleName |

Bảng 2.2 Quy định về Naming convention

#### Quy định về Database Convention

| **STT** | **Type** | **Quy tắc đặt tên** |
| --- | --- | --- |
| 1 | Database | db\_ |
| 2 | Table | tbl |
| 3 | Int | i |
| 4 | Float | f |
| 5 | Nvarchar | s |
| 6 | Varchar | s |
| 7 | Nchar | s |
| 8 | Char | s |
| 9 | Text | t |
| 10 | Date | d |
| 11 | Datetime | d |
| 12 | bit | b |
| 13 | decimal | m |
| 14 | Primary Key | PK\_ |
| 15 | Foreign Key | FK\_ |
| 16 | Store Procedure | sp\_ |
| 17 | View | vw\_ |

Bảng 2.3 Bảng định về Database

### Xác định Mục tiêu chất lượng của dự án

| **STT** | **Tên độ đo** | **Ý nghĩa** | **Công thức/cách đo** |
| --- | --- | --- | --- |
| 1 | Độ bao phủ kiểm thử (Test coverage) | Đo lường phần trăm mã nguồn được kiểm thử và báo cáo về độ phủ sóng của các ca kiểm thử đối với mã nguồn. | Test coverage (%) = (Số lượng ca kiểm thử đã thực hiện / Tổng số ca kiểm thử) x 100% |
| 2 | Tốc độ phát triển (Development speed) | Đo lường thời gian và nỗ lực cần thiết để hoàn thành các tác vụ phát triển, ví dụ như số lượng tính năng được hoàn thành trong một khoảng thời gian nhất định. | Development Speed = Số dòng code đã viết / Tổng số giờ lao động của tất cả các nhân viên tham gia vào quá trình phát triển |
| 3 | Số lỗi phát hiện sau khi triển khai (Defects found after deployment) | Đo lường số lỗi phát hiện sau khi phần mềm đã được triển khai và sử dụng. | Số lỗi phát hiện sau khi triển khai = (Số lỗi phát hiện trong giai đoạn kiểm thử / Tổng số lỗi phát hiện) x 100% |
| 4 | Thời gian hoạt động liên tục (Mean Time Between Failures - MTBF) | Đo lường thời gian hoạt động trung bình giữa các lỗi, tức là thời gian trung bình giữa các lần phần mềm gặp sự cố hoặc ngừng hoạt động. | MTBF = Tổng thời gian hoạt động / Số lần hỏng  [công thức tính MTBF](https://7pace.com/blog/software-development-quality-metrics) |
| 5 | Tỷ lệ lỗi (Defect Density) | Đo lường số lỗi trong một đơn vị mã nguồn hoặc số lỗi trên một đơn vị dòng mã. | Defect Density = số lỗi phát hiện trong quá trình kiểm thử / kích thước bản phát hành  [công thức tính Defect Density](https://viblo.asia/p/software-testing-metrics-kpis-4dbZNqQkKYM) |
| 6 | Thời gian giải quyết lỗi (Mean Time to Repair - MTTR) | Đo lường thời gian trung bình để sửa lỗi sau khi phát hiện được. | MTTR = tổng thời gian sửa chữa lỗi / số lỗi đã sửa chữa  [công thức tính MTTR](https://7pace.com/blog/software-development-quality-metrics) |
| 7 | Độ tin cậy (Reliability) | Đo lường khả năng của phần mềm hoạt động đúng và đáng tin cậy trong một khoảng thời gian nhất định. | Reliability = MTBF / (MTBF + MTTR) |
| 8 | Khả năng bảo trì (Maintainability) | Đo lường khả năng của phần mềm để thay đổi, bảo trì và nâng cấp. | Maintainability = (Tổng số giờ để bảo trì) / (Tổng số giờ để phát triển) |
| 9 | Thời gian đáp ứng yêu cầu (Response time) | Đo lường thời gian mà phần mềm trả về kết quả hoặc thực hiện hành động phản hồi khi có yêu cầu từ người dùng. | Response Time = Thời gian chờ tài nguyên + Thời gian thực hiện yêu cầu |

Bảng 2.4 Mục tiêu chất lượng của dự án

Tổng thời gian hoạt động được tính bằng tổng thời gian giữa các lần hỏng. Số lần hỏng là số lần phần mềm không hoạt động như mong đợi hoặc gặp lỗi trong suốt thời gian vận hành. Tổng thời gian sửa chữa lỗi bao gồm toàn bộ thời gian mà các nhân viên bảo trì phải dành để khắc phục các lỗi đã được phát hiện, bao gồm các giai đoạn như phát ao, xác nhận, đề xuất phương án khắc phục và thực hiện sửa chữa. Số lỗi đã sửa chữa là tổng số lỗi đã được xác định, báo cáo và khắc phục trong khoảng thời gian đánh giá. MTBF (Mean Time Between Failures) thể hiện thời gian hoạt động liên tục giữa các lần hỏng, trong khi MTTR (Mean Time to Repair) là thời gian trung bình để giải quyết một lỗi.

### Xác định tiêu chuẩn Kiểm thử được (Testability) cho dự án

#### Khái niệm, ý nghĩa của đặc tính Kiểm thử được

**Khái niệm:** Testability (khả năng kiểm thử) là khả năng của một hệ thống hoặc phần mềm để dễ dàng kiểm tra và xác nhận tính đúng đắn của nó thông qua việc thử nghiệm. Điều này bao gồm việc thiết kế các tính năng và chức năng của hệ thống sao cho chúng có thể được kiểm tra một cách hiệu quả, và cung cấp các công cụ và quy trình cho các nhà phát triển và nhà kiểm thử để tiến hành các bài kiểm tra một cách dễ dàng và hiệu quả.

Điều này giúp các nhà phát triển và kiểm thử viên có thể dễ dàng tìm ra các lỗi và xác định được hiệu suất của hệ thống một cách hiệu quả. Giups tạo ra một sản phẩm phần mềm ổn định và chất lượng, giảm thiểu các lỗi và sự cố trong quá trình sử dụng và tăng cường khả năng bảo trì và nâng cấp sản phẩm .

#### Chi tiết hóa các tiêu chí của đặc tính Kiểm thử được

| **STT** | **Tiêu chí** | **Yêu cầu cụ thể đối với dự án** |
| --- | --- | --- |
| 1 | Khả năng Vận hành (Operability) "Càng hoạt động tốt, càng dễ kiểm tra." | - Hệ thống ít lỗi, mỗi chức năng cần được kiểm thử ngay để dễ dàng sửa chữa  - Không có lỗi nào làm chặn việc thực hiện các bài test, tức không có lỗi nào khiến cả hệ thống gặp lỗi  - Sản phẩm phát triển theo từng giai đoạn chức năng (cho phép kiểm tra đồng thời khi phát triển), mỗi giai đoạn phát triển có thể cho đại diện khách hàng là chị Đào Thùy Dung tham gia kiểm thử |
| 2 | Khả năng Quan sát (Observability): "Những gì bạn thấy là những gì bạn kiểm tra." | - Đầu ra riêng biệt được tạo ra cho mỗi đầu vào.  - Trạng thái và biến của hệ thống có thể nhìn thấy hoặc truy vấn trong quá trình thực thi. Trạng thái và biến của hệ thống trong quá khứ có thể nhìn thấy.  - Tất cả các yếu tố ảnh hưởng đến đầu ra có thể nhìn thấy.  - Đầu ra không đúng dễ dàng nhận biết.  - Lỗi nội bộ được phát hiện tự động thông qua các cơ chế tự kiểm tra.  - Mã nguồn có thể truy cập. |
| 3 | Khả năng Kiểm soát (Controllability): "Càng có thể kiểm soát phần mềm, càng có thể tự động hóa và tối ưu hóa kiểm tra." | - Kiểm soát mã mã nguồn của hệ thống thống qua git, từ đó có thể dễ dàng thay đổi và nâng cấp hệ thống  - Kiểm soát quy tắc đặt tên biến (naming convention) đã được quy định ở đầu báo cáo khi nhóm có nhiều thành viên cùng tham gia phát triển  - Kiểm soát rủi ro để lộ thông tin người dùng bằng cách đội triển khai sử dụng các phương pháp xác thực |
| 4 | Khả năng Phân giải (Decomposability): "Phân rã càng tốt thì càng dễ kiểm thử." | - Phân chia dự án thành các thành phần nhỏ hơn để kiểm thử dễ dàng: module quản lý vị trí công việc, module quản lý ứng viên, module quản lý quy trình tuyển dụng…. Mỗi module không quá 10 cấu trúc rẽ nhánh.  - Cung cấp khả năng kiểm thử riêng lẻ cho từng module và tính năng của hệ thống. |
| 5 | Sự Đơn giản (Simplicity): "Càng ít cần kiểm tra, càng có thể kiểm tra nhanh chóng." | - Sự đơn giản về chức năng: loại bỏ những chức năng không cần thiết, đơn giản các tính năng cần thiết tối thiểu để đáp ứng yêu cầu.  - Sự đơn giản về cấu trúc: phân chia cấu trúc theo các module nhỏ để giới hạn sự lan truyền của lỗi.  - Sự đơn giản trong mã nguồn: áp dụng tiêu chuẩn mã hóa để dễ kiểm tra và bảo trì.Sử dụng các quy ước đặt tên nhất quán ,tránh sử dụng các câu lệnh lồng nhau, các toán tử 3 ngôi, … |
| 6 | Sự Ổn định (Stability): "Càng ít thay đổi, càng ít gây gián đoạn cho quá trình kiểm tra." | - Thay đổi trong phần mềm xảy ra ít.  - Thay đổi trong phần mềm được kiểm soát.  - Thay đổi trong phần mềm không làm mất giá trị các bài kiểm tra hiện có.  - Phần mềm phục hồi tốt sau khi xảy ra lỗi. |
| 7 | Khả năng Hiểu rõ (Understandability): "Càng nhiều thông tin chúng ta có, chúng ta càng kiểm tra thông minh." | - Thiết kế được hiểu rõ: các yêu cầu chức năng, phi chức năng của hệ thống.  -Nắm bắt được các cách thức hoạt động hạn chế của hệ thống.Xác định các đối tượng và nhu cầu của người dùng.  - Có khả năng phân tích các yêu cầu, thiết kế và chức năng phần mềm để xác định các rủi ro và vấn đề tiềm ẩn.  - Tài liệu kỹ thuật được tổ chức cẩn thận, cụ thể và chi tiết, chính xác. |
| 8 | Khả năng Truy cập (Availability): "Càng nhiều đối tượng hoặc thực thể có sẵn, thì càng dễ thiết kế các trường hợp kiểm tra." Điều này liên quan đến tính sẵn sàng của các đối tượng hoặc thực thể để thực hiện kiểm tra, bao gồm lỗi, mã nguồn, v.v. | - Các thực thể đã được xác định trong pha design như thông tin, tài khoản, nhân viên, … đều có thể được sử dụng để kiểm tra  - Các chức năng liên quan tới các thực thể trên cũng đều có thế tạo test case để kiểm tra |
| 9 | Công Cụ Kiểm tra (Testing Tools): "Các công cụ kiểm tra dễ sử dụng sẽ giảm thiểu số lượng nhân viên và giảm thiểu sự cần thiết về đào tạo." | - Sử dụng các công cụ kiểm thử như: Selenium, Appium, TestComplete…  -Chọn các công cụ có khả năng tích hợp dễ dàng với quy trình phát triển phần mềm hiện tại giúp giảm tối thiểu thời gian và công sức cho việc cấu hình và triển khai. |
| 10 | Tài liệu (Documentation): "Tài liệu về đặc tả và yêu cầu phải tuân theo nhu cầu của khách hàng và đầy đủ tính năng." | - Sử dụng tài liệu khảo sát từ khách hàng để phát triển các tính năng của sản phẩm nhằm xác định chính xác nhu cầu của khách hàng, tránh dư thừa và thiếu chức năng |

Bảng 2.5 Chi tiết hóa các tiêu chí của đặc tính kiểm thử được

## CHƯƠNG 3. XÁC ĐỊNH YÊU CẦU

### Vai trò của việc Xác định yêu cầu đối với dự án phần mềm

Với mỗi một dự án phát triển phần mềm thì việc xác định yêu cầu là việc quan trọng và cần làm đầu tiên sau đó mới tới các pha tiếp theo. Vai trò của người phân tích thiết kế hệ thống là phải hiểu rõ các yêu cầu chức năng và phi chức năng của phía khách hàng và các bên liên quan. Từ đó phân tích và thiết kế các chức năng chương trình phù hợp, đáp ứng đúng các yêu cầu của khách hàng và các bên liên quan.

Việc xác định các yêu cầu là giúp cho một dự án phát triển phần mềm biết được lộ trình phát triển của phần mềm bằng cách chia ra các giai đoạn phát triển tương ứng với các chức năng và yêu cầu của các bên đối tác và khách hàng. Giúp cho đội phát triển làm đúng việc mà mình cần làm (đúng theo yêu cầu của khách hàng).

Bằng việc xác định yêu yêu cầu đối với dự án phần mềm sẽ giúp cho dự án phần mềm:

+ Giảm thiểu chi phí cần thiết để mở rộng

+ Giảm thiểu số lần làm lại trong quá trình thiết kế

+ Giảm thiểu khả năng sai sót trong quá trình xây dựng thiết kế

+ Hạn chế dữ liệu hỗn độn trong dự án

+ Gia tăng khả năng thỏa mãn khách hàng

+ Tránh dư thừa và sai sót dữ liệu

### Phương pháp đánh giá chất lượng của yêu cầu

Trong dự án, nhóm đã sử dụng 12 tiêu chí theo “Requirements Management Using IBM® Rational® RequisitePro” để đánh giá chất lượng phần mềm:

TC01: Không mập mờ

TC02: Kiểm thử được

TC03: Rõ ràng (Ngắn gọn đơn giản chính xác)

TC04: Đúng đắn với thực tế

TC05: Dễ hiểu

TC06: Khả thi

TC07: Độc lập với yêu cầu khác

TC08: Không thể chia nhỏ

TC09: Cần thiết

TC10: Tách rời với việc triển khai

TC11: Không trùng lặp

TC12: Đầy đủ (thông tin)

### Danh sách các yêu cầu của hệ thống

3.3.1. Xác định các yêu cầu chức năng

1. Xác thực người dùng

* R1.1: Xác thực người dùng: Hệ thống yêu cầu người dùng nhập thông tin đăng nhập (email/tên đăng nhập và mật khẩu), sau đó xác thực để cấp quyền truy cập vào hệ thống.
* R1.2: Quản lý phiên làm việc: Duy trì phiên đăng nhập, tự động đăng xuất sau thời gian không hoạt động
* R1.3: Hệ thống hỗ trợ khôi phục mật khẩu qua email/số điện thoại với mã xác nhận
* R1.4:Ghi lại lịch sử đăng nhập (thời gian, IP, thiết bị) để phát hiện truy cập bất thường

1. Khởi tạo tài khoản

* R2.1: Tạo tài khoản mới: Cho phép đăng ký tài khoản với thông tin cá nhân.
* R2.2: Xác thực email/số điện thoại trước khi kích hoạt.
* R2.3: Chính sách mật khẩu an toàn: Yêu cầu mật khẩu mạnh để bảo mật tài khoản.

1. Quản trị người dùng

* R3.1: Phân quyền truy cập: hệ thống cho phép quản trị viên gán, thay đổi vai trò người dùng và kiểm soát quyền truy cập.
* R3.2: Quản lý tài khoản người dùng: cho phép tạo, sửa, xóa tài khoản người dùng, cho phép người dùng cập nhật thông tin cá nhân.
* R3.3: Ghi nhận hoạt động của người dùng: ghi nhận và lưu trữ lịch sử hoạt động của người dùng, hỗ trợ truy vấn và xuất báo cáo.
* R3.4: Bảo vệ dữ liệu và ngăn chặn truy cập trái phép: mã hóa mật khẩu, hạn chế quyền truy cập và tự động đăng xuất sau một khoảng thời gian.

1. Quản lý vật tư y tế

* R4.1: Quản lý danh mục vật tư y tế: ghi thông tin thuốc (tên, đơn vị tính, nhà cung cấp, giá trị, số lượng tồn kho, hạn sử dụng
* R4.2: Quản lý nhập - xuất vật tư: cho phép nhập mới, cập nhật vật tư
* R4.3: Cập nhật và theo dõi trạng thái vật tư: hỗ trợ chỉnh sửa thông tin, ghi nhận lịch sử thay đổi, cảnh báo hết hạn.
* R4.4: Kiểm soát tồn kho và báo cáo: hệ thống theo dõi số lượng tồn kho theo thời gian thực, xuất báo cáo tổng hợp.
* R4.5: Đảm bảo tính chính xác và bảo mật: đảm bảo tính chính xác.

1. Quản lý kho thuốc

* R5.1: Ghi nhận thông tin nhập thuốc (số lượng, nhà cung cấp, hạn dùng...).
* R5.2: Quản lý xuất thuốc (loại, số lượng, thời gian, người thực hiện...)
* R5.3: Theo dõi tồn kho, cảnh báo khi thuốc sắp hết hoặc quá mức.
* R5.4: Cảnh báo thuốc sắp hết hạn, bảo quản sai điều kiện. Hỗ trợ thu hồi, hủy thuốc, ghi nhận lý do và xác nhận

1. Quản lý thu hồi và khiếu lại thuốc.

* R6.1: Ghi nhận khiếu nại hoặc thu hồi thuốc từ người dùng hoặc cơ quan sản xuất.
* R6.2: Ghi chi tiết thông tin thuốc bị thu hồi/khiếu lại ( tên thuốc, lô sản xuất, hạn sử dụng, lý do bị thu hồi/khiếu lại..,)
* R6.3: Tạo thông báo thuốc bị thu hồi cho bệnh nhân/ người mua.
* R6.4: Theo dõi và xác nhận thu hồi thuốc/thuốc khiếu nại.
* R6.5: Quản lý biệt trữ thuốc( đối với thuốc bị thu hồi)
* R6.6: Báo cáo tổng hợp về quá trình xử lý thuốc bị thu hồi/ khiếu nại bao gồm kết quả và biện pháp khắc phục
* R6.7: Tạo báo cáo thống kê thuốc thu hồi/ bị khiếu nại.

1. Quản lý thuốc kê đơn

* R7.1: Tạo và lưu trữ thông tin thuốc kê đơn(tên thuốc, liều lượng, cách sử dụng..,)
* R7.2: Tìm kiếm và truy xuất đơn thuốc theo đầu danh mục ( Khách hàng, bác sĩ, đơn thuốc)
* R7.3: Theo dõi số lượng thuốc kê đơn đã cấp và số lượng thuốc còn lại trong kho.
* R7.4: Gửi và hiển thị cảnh báo về tình trạng thuốc trong kho (hết hạn, tồn kho ít, thu hồi).
* R7.5: Báo cáo chi tiết về thuốc kê đơn.

1. Báo cáo tồn kho và thống kê

* R8.1: Xem chi tiết tiết về số lượng thuốc( thuốc tồn kho, đã nhập, đã xuất , sắp hết hạn)
* R8.2: Lọc và tìm kiếm theo từng loại thuốc, số lượng thuốc, hạn sử dụng.
* R8.3: Cảnh báo về các số lượng thuốc trong kho, thuốc hết hạn và đã hết hạn.
* R8.4: Báo cáo tồn kho theo từng danh mục.
* R8.5: Thống kê và phân tích xu hướng sử dụng thuốc trong từng giai đoạn,hỗ trợ quản lý tồn kho.
* R8.6: Thống thuốc bị thu hồi/ khiếu lại.

1. Đặt hàng thuốc không kê đơn

* R9.1: Hệ thống hiển thị thuốc mà khách hàng có thể mua mà không cần kê đơn
* R9.2: Thêm vào giỏ hàng: Khách hàng có thể chọn số lượng thuốc cần mua và thêm vào giỏ hàng. Hệ thống sẽ cập nhật giỏ hàng theo thời gian thực.
* R9.3: Xác nhận đơn hàng: Hệ thống hiển thị giỏ hàng để khách hàng kiểm tra danh sách thuốc đã chọn, số lượng, đơn giá và tổng giá trị đơn hàng.
* R9.4: Thanh toán: Khách hàng có thể chọn phương thức thanh toán phù hợp (tiền mặt, thẻ tín dụng, ví điện tử). Hệ thống xử lý giao dịch và xác nhận đơn hàng.
* R9.5: Hủy hoặc chỉnh sửa đơn hàng: Khách hàng có thể chỉnh sửa hoặc hủy đơn hàng trước khi đơn hàng được xử lý.
* R9.6: Quyền truy cập: Chỉ khách hàng đã đăng ký tài khoản mới có thể đặt hàng.

1. Xem lịch sử mua thuốc của bản thân

* R10.1: Xem danh sách đơn hàng: Khách hàng có thể xem lại tất cả các đơn hàng đã mua trước đó.
* R10.2: Xem chi tiết đơn hàng: Khách hàng có thể truy cập vào từng đơn hàng để xem thông tin như ngày mua, danh sách thuốc, tổng số tiền và trạng thái đơn hàng.
* R10.3: Tìm kiếm đơn hàng: Hệ thống hỗ trợ tìm kiếm đơn hàng theo khoảng thời gian hoặc theo tên thuốc để dễ dàng tra cứu.
* R10.4: Quyền truy cập: Chỉ khách hàng đăng nhập mới có thể xem lịch sử mua hàng của mình.

### Xác định các yêu cầu phi chức năng

1. Yêu cầu giao diện

* R11.1: Thích ứng đa thiết bị.
* R11.2: Lưu trữ và hiển thị các thông tin trong hệ thống một cách đầy đủ theo chuẩn GPP.
* R11.3: Người dùng đăng nhập vào hệ thống sẽ có quyền sử dụng chức năng tương ứng với vai trò của mình.
* R11.4: Hỗ trợ 2 ngôn ngữ chính là Tiếng Anh và Tiếng Việt.
* R11.5: Có bố cục hài hòa, không làm rối sự chú ý của người dùng khỏi những hạng mục hay chức năng chính.
* R11.6: Màu sắc đơn giản, phù hợp với môi trường nhà thuốc.

1. Yêu cầu Tính bảo mật

* R12.1: Phải bao gồm biện pháp xác thực người dùng an toàn, đặc biệt là đối với tài khoản có quyền hạn cao.
* R12.2: Mật khẩu người dùng phải mạnh.
* R12.3: Người dùng chỉ có thể truy cập dữ liệu phù hợp với vai trò
* R12.4: Có khả năng phục hồi và sao lưu dữ liệu để đảm bảo tính an toàn dữ liệu cho hệ thống.

1. Yêu cầu Ràng buộc

* R13.1: Hệ thống dễ dàng cài đặt, chuyển đổi, tích hợp được với nhiều hệ thống
* R13.2: Hoạt động tốt trên nhiều hệ điều hành, nền tảng phổ biến.
* R13.3: Các chức năng, mô-đun có tính độc lập để đảm bảo cho việc dễ dàng bảo trì, nâng cấp.
* R13.4: Có tài liệu hướng dẫn sử dụng phần mềm đầy đủ.
* R13.5: Phần mềm cần tuân thủ quy định của Bộ Y Tế về quản lý, bảo quản, hành nghề thuốc, đáp ứng đủ tiêu chuẩn của GPP.

1. Yêu cầu độ tin cậy

* R14.1: Mọi chức năng và quy trình phải có tính độc lập để đảm bảo hoạt động ổn định ngay cả khi gặp sự cố.
* R14.2: Bao gồm các cơ chế cho trường hợp có sự cố về dữ liệu cũng như hệ thống.
* R14.3: Các thủ tục thanh toán thông tin trong hệ thống phải được bảo mật tuyệt đối an toàn.

1. Yêu cầu hiệu suất

* R15.1: Tối ưu hiệu suất để hệ thống có thể hoạt động mượt mà và liên tục.
* R15.2: Có khả năng mở rộng linh hoạt để duy trì độ ổn định của hệ thống khi có lượng truy cập và yêu cầu cao.

### Kết quả đánh giá chất lượng các yêu cầu

* **Đánh giá chất lượng yêu cầu chức năng**

|  | TC01 | TC02 | TC03 | TC04 | TC05 | TC06 | TC07 | TC08 | TC09 | TC10 | TC11 | TC12 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R1.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R1.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R1.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R2.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ |
| R2.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R2.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R3.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R3.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R3.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R3.4 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ | Đ | Đ | Đ |
| R4.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R4.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R4.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R4.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R4.5 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ | Đ | KĐ | KĐ |
| R5.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R5.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R5.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R5.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ |
| R6.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R6.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  (có thể chia nhỏ  thuốc thu hồi và khiếu nại) | Đ | Đ | Đ | KĐ  (chưa liệt kê đầy đủ thông tin thuốc thu hồi/ khiếu nại) |
| R6.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R6.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R6.5 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R6.6 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  ( R8.6 đều thống kê đề cập thuốc thu hồi/ khiếu nại) | KĐ  (có thể chia nhỏ  thuốc thu hồi và khiếu nại) | Đ | Đ | Đ | Đ |
| R6.7 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  (trùng lặp với R8.6) | Đ |
| R7.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R7.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R7.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R7.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R7.5 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.5 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R8.6 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ | Đ | KĐ | Đ |
| R9.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R9.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R9.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R9.4 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  (có liên quan đến xác nhận đơn hàng) | Đ | Đ | Đ | Đ | Đ |
| R9.5 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R9.6 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  (có liên quan chặt chẽ đến yêu cầu xác thực hệ thống) | Đ | Đ | Đ | Đ | Đ |
| R10.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R10.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R10.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R10.4 | Đ | Đ | Đ | Đ | Đ | Đ | KĐ  (có liên quan chặt chẽ đến yêu cầu xác thực hệ thống) | Đ | Đ | Đ | Đ | Đ |

Bảng 3.1 **Đánh giá chất lượng yêu cầu chức năng**

Sau khi đánh giá các yêu cầu theo 12 tiêu chí của IBM Rational RequisitePro, chúng ta có được danh sách chức năng:

F1: Đăng Nhập (email/tên đăng nhập và mật khẩu). Có quản lý phiên làm việc, hỗ trợ khôi phục mật khẩu và ghi lại lịch sử đăng nhập.

F2: Khởi tạo tài khoản.

F3: Phân quyền truy cập

F4: Quản lý tài khoản người dùng dành cho quản trị.

F5: Ghi nhận hoạt động của người dùng.

F6: Quản lý danh mục vật tư y tế

F7: Quản lý nhập - xuất vật tư y tế

F8: Cập nhật và theo dõi trạng thái vật tư y tế

F9: Kiểm soát tồn kho và báo cáo.

F10: Ghi nhận thông tin nhập thuốc

F11: Quản lý xuất thuốc

F12: Theo dõi tồn kho, cảnh báo thuốc sắp hết hoặc quá mức

F13: Ghi nhận khiếu nại và thu hồi thuốc

F14: Tạo thông báo thuốc bị thu hồi cho bệnh nhân/người mua

F15: Theo dõi và xác nhận thu hồi thuốc/thuốc khiếu nại

F16: Quản lý biệt trữ thuốc đối với thuốc bị thu hồi

F17: Tạo và lưu trữ thông tin thuốc kê đơn

F18: Tìm kiếm và truy xuất đơn thuốc theo danh mục

F19: Theo dõi số lượng thuốc kê đơn đã cấp và số lượng thuốc còn lại trong kho

F20: Gửi và hiển thị cảnh báo về tình trạng thuốc trong kho

F21: Báo cáo chi tiết về thuốc kê đơn

F22: Xem chi tiết số lượng thuốc

F23: Lọc và tìm kiếm theo từng loại thuốc, số lượng thuốc, hạn sử dụng

F24: Cảnh báo về số lượng thuốc trong kho, thuốc hết hạn và đã hết hạn

F25: Báo cáo tồn kho theo từng danh mục

F26: Thống kê và phân tích xu hướng sử dụng thuốc

F27: Hiển thị danh sách thuốc không kê đơn

F28: Thêm vào giỏ hàng

F29: Xác nhận đơn hàng

F30: Hủy hoặc chỉnh sửa đơn hàng

F31: Xem danh sách đơn hàng

F32: Xem chi tiết đơn hàng

F33: Tìm kiếm đơn hàng

**Gom nhóm các chức năng**

| **Chức năng mức 2** | | **Chức năng mức 1** |
| --- | --- | --- |
| **STT** | **Tên chức năng** |
| 1 | Đăng nhập (quản lý phiên, khôi phục mật khẩu, ghi lịch sử đăng nhập). | Quản lý người dùng & phân quyền |
| 2 | Đăng ký |
| 3 | Phân quyền truy cập. |
| 4 | Quản lý tài khoản người dùng dành cho quản trị. |
| 5 | Ghi nhận hoạt động của người dùng. |
| 6 | Quản lý danh mục vật tư y tế. | Quản lý vật tư y tế |
| 7 | Quản lý nhập - xuất vật tư y tế. |
| 8 | Cập nhật và theo dõi trạng thái vật tư y tế. |
| 9 | Kiểm soát tồn kho và báo cáo. |
| 10 | Ghi nhận thông tin nhập thuốc. | Quản lý thuốc thông thường (nhập/xuất – thu hồi) |
| 11 | Quản lý xuất thuốc. |
| 12 | Theo dõi tồn kho, cảnh báo thuốc sắp hết hoặc quá mức. |
| 13 | Ghi nhận khiếu nại và thu hồi thuốc. |
| 14 | Tạo thông báo thuốc bị thu hồi cho bệnh nhân/người mua. |
| 15 | Theo dõi và xác nhận thu hồi thuốc/thuốc khiếu nại. |
| 16 | Quản lý biệt trữ thuốc đối với thuốc bị thu hồi. |
| 17 | Tạo và lưu trữ thông tin thuốc kê đơn. | Quản lý thuốc kê đơn |
| 18 | Tìm kiếm và truy xuất đơn thuốc theo danh mục. |
| 19 | Theo dõi số lượng thuốc kê đơn đã cấp và số lượng thuốc còn lại trong kho. |
| 20 | Gửi và hiển thị cảnh báo về tình trạng thuốc trong kho. |
| 21 | Báo cáo chi tiết về thuốc kê đơn. |
| 22 | Xem chi tiết số lượng thuốc. | Quản lý Số Lượng Thuốc & Tồn Kho |
| 23 | Lọc và tìm kiếm theo từng loại thuốc, số lượng thuốc, hạn sử dụng. |
| 24 | Cảnh báo về số lượng thuốc trong kho, thuốc hết hạn và đã hết hạn. |
| 25 | Báo cáo tồn kho theo từng danh mục. |
| 26 | Thống kê và phân tích xu hướng sử dụng thuốc. |
| 27 | Hiển thị danh sách thuốc không kê đơn. |
| 28 | Thêm vào giỏ hàng. | Quản lý đơn hàng |
| 29 | Xác nhận đơn hàng. |
| 30 | Hủy hoặc chỉnh sửa đơn hàng. |
| 31 | Xem danh sách đơn hàng. |
| 32 | Xem chi tiết đơn hàng. |
| 33 | Tìm kiếm đơn hàng. |

Bảng 3.2 Gom nhóm chức năng

* **Đánh giá chất lượng Yêu cầu phi chức năng**

|  | TC01 | TC02 | TC03 | TC04 | TC05 | TC06 | TC07 | TC08 | TC09 | TC10 | TC11 | TC12 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R11.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R11.2 | Đ | KĐ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R11.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R11.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R11.5 | Đ | KĐ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R11.6 | Đ | KĐ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R12.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R12.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R12.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R12.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R13.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R13.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R13.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R13.4 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R13.5 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |
| R14.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R14.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R14.3 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R15.1 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | KĐ | Đ | Đ |
| R15.2 | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ | Đ |

Bảng 3.3 Đánh giá chất lượng Yêu cầu phi chức năng

* Nguyên nhân các yêu cầu không đạt một số tính chất
* R11.1 - TC10: Việc thích ứng đa thiết được thực hiện quá trình triển khai => Không đáp ứng TC10.
* R11.2 - TC02: Việc lưu trữ thông tin chính xác ngoài việc kiểm soát các thông tin nhập vào thì việc nhập sai thông tin từ bên thứ ba là điều không thể kiểm thử được => Không đáp ứng TC02.
* R11.3 - TC10: Liên quan đến kỹ thuật khi triển khai => Không đáp ứng TC10.
* R11.4 - TC10: Liên quan đến kỹ thuật khi triển khai => Không đáp ứng TC10.
* R11.5 - TC02: Sắp xếp bố cục thuộc về diện thẩm mỹ, không phải kỹ thuật nên không thể kiểm thử => Không đáp ứng TC02.
* R11.6 - TC02: Màu sắc liên quan đến thẩm mỹ, không phải kĩ thuật nên không thể kiểm thử => Không đáp ứng TC02.
* R13.1 - TC10: Việc dễ dàng cài đặt, chuyển đổi, tích hợp sẽ do ảnh hưởng khi thiết kế hệ thống => Không đáp ứng TC10
* R13.2 - TC10: Việc thích ứng đa hệ điều hành, nền tảng phải được tiến hành và kiểm thử trong quá trình triển khai => Không đáp ứng TC10.
* R13.3 - TC10: Việc độc lập mô đun, chức năng sẽ chịu ảnh hưởng của việc hệ thống có được tiến hành một cách hợp lý, thiết kế tốt => Không đáp ứng TC10.
* R14.1 - TC10: Cơ chế kỹ thuật phải gắn liền với việc triển khai => Không đáp ứng TC10.
* R14.2 - TC10: Cơ chế kỹ thuật phải gắn liền với việc triển khai => Không đáp ứng TC10.
* R14.3 - TC10: Bảo mật thuộc mặt kỹ thuật khi triển khai => Không đáp ứng TC10.
* R15.1 - TC10: Hiệu suất tốt sẽ xuất phát từ việc hệ thống được triển khai tốt => Không đáp ứng TC10.

## CHƯƠNG 4. KIẾN TRÚC PHẦN MỀM

### Vai trò của kiến trúc phần mềm với quá trình phát triển dự án phần mềm

Kiến trúc phần mềm có thể được xem như là bản thiết kế tổng thể cho một hệ thống, giúp giảm bớt độ phức tạp bằng cách cung cấp các mức độ trừu tượng và thiết lập cơ chế giao tiếp, phối hợp giữa các thành phần. Nó không chỉ định hình cấu trúc của các thành phần mà còn đảm bảo rằng hệ thống đáp ứng đủ các yêu cầu kỹ thuật và vận hành, đồng thời tối ưu hóa các thuộc tính chất lượng như hiệu suất và bảo mật.

Theo định nghĩa, “Kiến trúc phần mềm là tập hợp các cấu trúc cần thiết để lập trình nên hệ thống phần mềm, bao gồm các phần tử, mối quan hệ giữa chúng và các thuộc tính của chúng” [Software Architecture in Practice (2013)]. Nhờ vậy, kiến trúc phần mềm không những tạo điều kiện thuận lợi cho việc ra quyết định trong thiết kế mà còn giúp tái sử dụng các thành phần và mẫu thiết kế đã được chứng minh qua các dự án trước đó.

Các vai trò chính của kiến trúc phần mềm bao gồm:

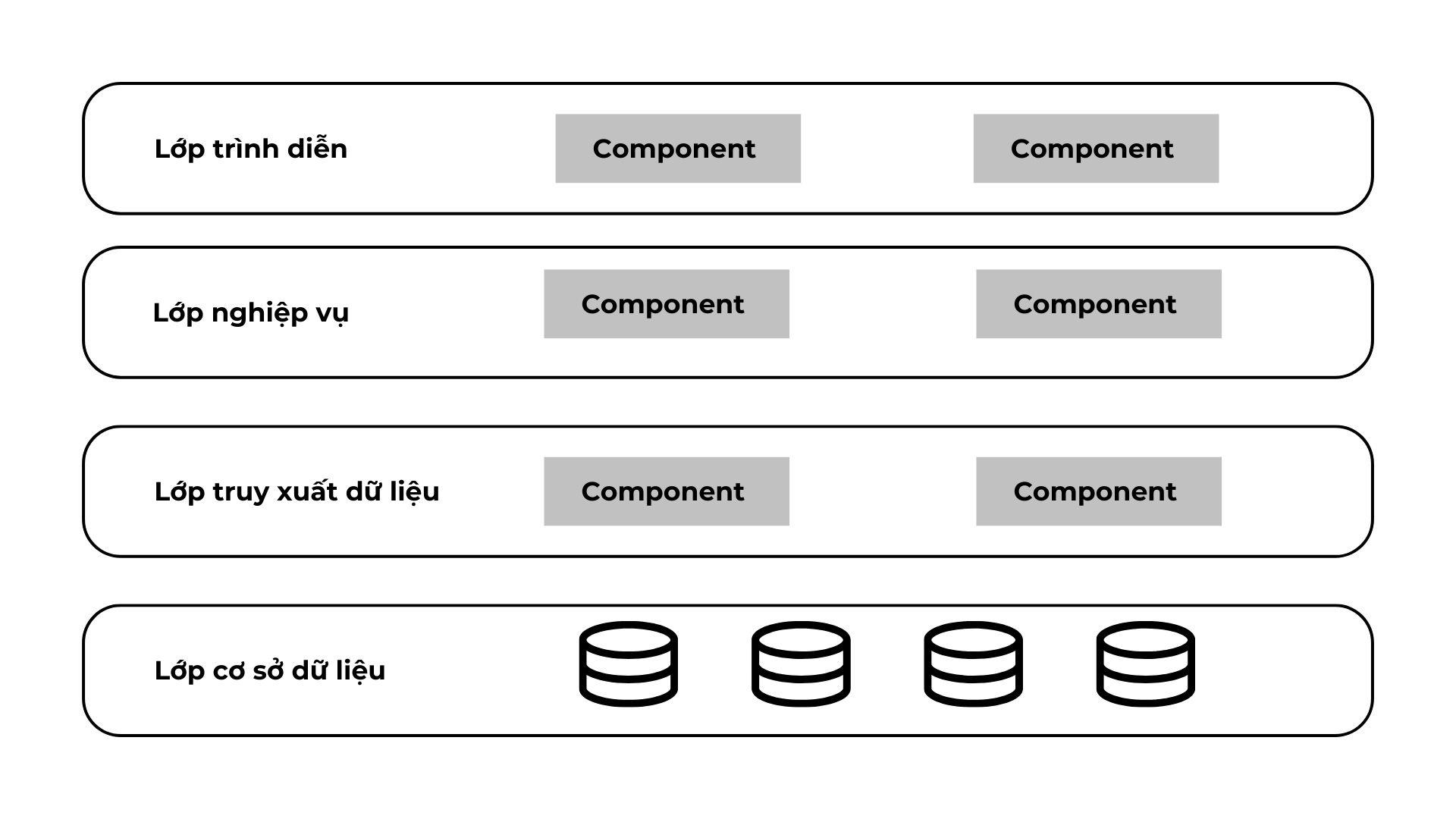
* **Hỗ trợ giao tiếp và phân tích:** Bằng cách trừu tượng hóa hệ thống qua những khái niệm dễ hiểu, kiến trúc phần mềm giúp các bên liên quan và thành viên dự án hình dung, hiểu rõ và phân tích hệ thống một cách hiệu quả, từ đó tạo điều kiện cho việc phối hợp và trao đổi thông tin.
* **Phác họa cấu trúc hệ thống:** Nó định nghĩa một giải pháp có cấu trúc, đảm bảo các yêu cầu kỹ thuật và vận hành được đáp ứng đồng thời cải thiện các thuộc tính chất lượng như hiệu suất và bảo mật.
* **Nền tảng cho các quyết định quan trọng:** Khi phần mềm vận hành thực tế có những sai lệch so với kế hoạch ban đầu, việc đối chiếu với kiến trúc thiết kế giúp nhanh chóng nhận diện và khắc phục vấn đề.
* **Tài liệu tham khảo cho dự án sau:** Kiến trúc phần mềm tạo ra một mô hình gắn kết với hệ thống hiện tại, từ đó giúp việc chuyển giao hoặc áp dụng vào các hệ thống tương tự được thực hiện một cách liền mạch, đảm bảo các thuộc tính chất lượng và yêu cầu chức năng không bị thay đổi quá nhiều.

Như vậy, kiến trúc phần mềm không chỉ đơn thuần là bản vẽ thiết kế mà còn là nền tảng để quản lý và phát triển hệ thống một cách hiệu quả, từ khâu giao tiếp đến việc đảm bảo chất lượng và hỗ trợ phát triển lâu dài.

### Một số kiến trúc phần mềm phổ biến

#### **Kiến trúc phân lớp - Layered pattern**

Kiến trúc phân lớp là một mô hình thiết kế phần mềm được sử dụng rộng rãi nhằm chia hệ thống thành các tầng với các trách nhiệm rõ ràng. Mô hình này giúp tổ chức mã nguồn một cách có trật tự, tăng tính bảo trì và khả năng mở rộng của hệ thống. Mỗi tầng trong kiến trúc phân lớp chỉ đảm nhiệm một phần công việc nhất định, từ giao diện người dùng cho đến xử lý nghiệp vụ và truy xuất dữ liệu, qua đó tạo ra một sự phân chia trách nhiệm rõ ràng và hợp lý.



Hình 4.1 **Kiến trúc phân lớp - Layered pattern**

Các lớp chính trong kiến trúc phân lớp:

Lớp trình diễn (Presentation Layer) là tầng gần gũi nhất với người dùng, nơi các yêu cầu từ người dùng được nhận và các dữ liệu cần hiển thị được xử lý. Tầng này chịu trách nhiệm về giao diện, tương tác và trình bày thông tin, giúp người dùng dễ dàng tiếp cận và sử dụng các chức năng của hệ thống. Việc tách riêng lớp trình diễn cho phép thay đổi giao diện mà không ảnh hưởng đến các tầng bên dưới.

Tiếp theo, lớp nghiệp vụ (Business Logic Layer) chứa các quy tắc và xử lý nghiệp vụ cốt lõi của ứng dụng. Đây là nơi các logic, quy trình xử lý dữ liệu được triển khai nhằm đáp ứng các yêu cầu cụ thể của doanh nghiệp. Bằng cách tách riêng logic nghiệp vụ khỏi giao diện và dữ liệu, ta có thể kiểm soát và bảo trì các quy trình một cách độc lập, đồng thời dễ dàng thực hiện kiểm thử tự động cho các chức năng của hệ thống.

Lớp truy xuất dữ liệu (Data Access Layer) đảm nhiệm việc giao tiếp với cơ sở dữ liệu hay các nguồn dữ liệu khác. Lớp này giúp trừu tượng hóa các thao tác truy vấn, lưu trữ và cập nhật dữ liệu, đảm bảo rằng các tầng trên không cần quan tâm đến chi tiết kỹ thuật của việc xử lý dữ liệu. Việc tách biệt tầng truy xuất dữ liệu không chỉ giúp cải thiện khả năng bảo trì mà còn tạo điều kiện thuận lợi cho việc thay đổi công nghệ lưu trữ mà không ảnh hưởng đến các phần khác của ứng dụng.

Lớp cơ sở dữ liệu (Database Layer) là nơi chứa dữ liệu thực tế, có thể là cơ sở dữ liệu quan hệ hoặc NoSQL. Tầng này thường được truy xuất thông qua tầng truy xuất dữ liệu.

Một trong những ưu điểm nổi bật của kiến trúc phân lớp là tính module cao, cho phép các thành phần của hệ thống được phát triển, kiểm thử và bảo trì độc lập. Nhờ đó, khi có yêu cầu thay đổi hoặc mở rộng, chỉ cần điều chỉnh tầng liên quan mà không làm gián đoạn toàn bộ hệ thống. Tuy nhiên, cần lưu ý rằng việc truyền dữ liệu qua lại giữa các tầng có thể gây ra một số vấn đề về hiệu suất nếu không được tối ưu hóa hợp lý.

Nhìn chung, kiến trúc phân lớp mang lại sự rõ ràng trong việc quản lý và phát triển phần mềm, giúp tách biệt các chức năng và tối ưu hóa quá trình phát triển. Đây là một lựa chọn lý tưởng cho nhiều ứng dụng doanh nghiệp và các dự án phát triển web, đặc biệt khi yêu cầu tính mở rộng, bảo trì và kiểm thử là ưu tiên hàng đầu.

#### **Kiến trúc nguyên khối (Monolithic Architecture):**

Kiến trúc nguyên khối (Monolithic Architecture) là mô hình xây dựng phần mềm truyền thống, trong đó toàn bộ các chức năng, module và thành phần của ứng dụng được xây dựng, triển khai và chạy như một đơn vị thống nhất. Điều này có nghĩa là giao diện người dùng, logic nghiệp vụ, truy xuất dữ liệu và các thành phần khác đều được gói gọn trong một khối duy nhất, không tách biệt ra thành các dịch vụ hoặc module độc lập. Một trong những ưu điểm chính của kiến trúc nguyên khối là sự đơn giản trong phát triển và triển khai. Vì toàn bộ hệ thống được tích hợp thành một ứng dụng duy nhất, việc xây dựng, kiểm thử và triển khai thường ít phức tạp hơn so với các kiến trúc phân tán. Điều này giúp cho việc khởi tạo dự án ban đầu trở nên nhanh chóng và dễ dàng, đặc biệt với các ứng dụng có quy mô nhỏ hoặc vừa. Ngoài ra, việc giao tiếp giữa các thành phần bên trong hệ thống cũng diễn ra nhanh chóng do không cần phải xử lý các giao thức mạng phức tạp.

Kiến trúc nguyên khối cũng tồn tại một số hạn chế. Khi hệ thống phát triển và mở rộng, khối lượng mã nguồn tăng lên có thể dẫn đến việc khó bảo trì, khó mở rộng và khó triển khai các thay đổi mà không ảnh hưởng đến toàn bộ hệ thống. Mọi thay đổi nhỏ cũng có thể gây ra ảnh hưởng lan tỏa vì sự phụ thuộc chặt chẽ giữa các thành phần. Hơn nữa, kiến trúc nguyên khối có thể gặp khó khăn trong việc tối ưu hóa hiệu suất khi ứng dụng phải chịu tải lớn, do toàn bộ hệ thống đều chạy trên cùng một quá trình hoặc máy chủ.

Vì vậy, kiến trúc nguyên khối thường được lựa chọn cho các dự án khởi đầu hoặc các ứng dụng có quy mô nhỏ, nơi mà tính đơn giản và tốc độ triển khai là ưu tiên hàng đầu. Khi hệ thống dần phát triển và yêu cầu về khả năng mở rộng, bảo trì, hoặc phân chia công việc trở nên phức tạp hơn, các mô hình kiến trúc phân tán như Microservices có thể được cân nhắc thay thế hoặc kết hợp để giải quyết các vấn đề này.

#### **Microservices Architecture (Kiến trúc Microservices)**

Kiến trúc Microservices là mô hình phát triển phần mềm trong đó hệ thống được chia thành nhiều dịch vụ nhỏ, độc lập, mỗi dịch vụ đảm nhiệm một chức năng nghiệp vụ cụ thể và giao tiếp với nhau thông qua các API. Mỗi dịch vụ có thể được phát triển, kiểm thử, triển khai và mở rộng riêng biệt, giúp giảm bớt sự phụ thuộc giữa các phần của hệ thống. Một ưu điểm nổi bật của kiến trúc Microservices là khả năng mở rộng linh hoạt. Khi một dịch vụ gặp tải cao, chỉ cần mở rộng dịch vụ đó mà không ảnh hưởng đến toàn bộ hệ thống. Điều này không chỉ giúp tối ưu hóa tài nguyên mà còn tăng tính ổn định và khả năng chịu lỗi của hệ thống, vì sự cố tại một dịch vụ không làm sập toàn bộ ứng dụng. Ngoài ra, mô hình này cũng hỗ trợ việc áp dụng công nghệ phù hợp với từng dịch vụ cụ thể, giúp nhóm phát triển có thể lựa chọn ngôn ngữ lập trình, cơ sở dữ liệu hoặc framework phù hợp nhất cho từng trường hợp.

Tuy nhiên, kiến trúc Microservices cũng đi kèm với những thách thức nhất định. Việc quản lý và giám sát một hệ thống phân tán đòi hỏi các công cụ phức tạp để theo dõi, ghi log và đảm bảo an toàn dữ liệu. Sự giao tiếp qua lại giữa các dịch vụ, thường dựa trên giao thức HTTP hoặc message queue, cần được thiết kế cẩn thận để tránh các vấn đề về độ trễ và đảm bảo tính nhất quán của dữ liệu. Hơn nữa, việc triển khai và duy trì nhiều dịch vụ độc lập cũng đòi hỏi kiến thức sâu về DevOps và các công cụ tự động hóa.

Kiến trúc Microservices phù hợp với các hệ thống có quy mô lớn, yêu cầu khả năng mở rộng cao và cần đáp ứng tốc độ thay đổi của thị trường. Khi được triển khai đúng cách, mô hình này giúp cải thiện hiệu suất, tăng tính linh hoạt và giảm thiểu rủi ro hệ thống, nhưng cũng đòi hỏi sự đầu tư về mặt kỹ thuật và quản lý phù hợp để vượt qua các thách thức kỹ thuật đi kèm.

#### **Event-Driven Architecture (Kiến trúc hướng sự kiện)**

Kiến trúc hướng sự kiện (Event-Driven Architecture) là một mô hình phát triển phần mềm tập trung vào việc xử lý các sự kiện khi chúng xảy ra trong hệ thống. Thay vì thực hiện các lệnh gọi hàm trực tiếp giữa các thành phần, mô hình này dựa vào việc phát sinh, truyền tải và xử lý các sự kiện bất đồng bộ. Mỗi khi có một hành động hoặc thay đổi trạng thái diễn ra, hệ thống sẽ tạo ra một sự kiện, sau đó các thành phần quan tâm (listeners hoặc subscribers) sẽ nhận và xử lý sự kiện đó theo cách riêng của mình. Ưu điểm nổi bật của kiến trúc hướng sự kiện là khả năng giảm sự phụ thuộc lẫn nhau giữa các thành phần. Các thành phần không cần phải biết thông tin chi tiết về nhau mà chỉ cần đăng ký nhận các sự kiện mà chúng quan tâm. Điều này không chỉ giúp hệ thống trở nên linh hoạt hơn mà còn dễ dàng mở rộng khi có thêm thành phần mới cần tham gia xử lý sự kiện mà không ảnh hưởng đến các phần còn lại của hệ thống. Hơn nữa, kiến trúc hướng sự kiện hỗ trợ xử lý bất đồng bộ, cho phép hệ thống tiếp nhận và xử lý nhiều yêu cầu cùng lúc mà không gây tắc nghẽn. Các công cụ trung gian như message broker, message queue (ví dụ: RabbitMQ, Apache Kafka) thường được sử dụng để đảm bảo các sự kiện được truyền tải một cách hiệu quả và an toàn giữa các thành phần, từ đó cải thiện hiệu năng và khả năng chịu tải của hệ thống.

Mô hình này cũng đặt ra một số thách thức nhất định. Việc đảm bảo tính nhất quán của dữ liệu và quản lý lỗi trong một hệ thống phân tán dựa trên sự kiện đòi hỏi phải có chiến lược thiết kế tỉ mỉ. Đặc biệt, khi số lượng sự kiện tăng cao, việc giám sát, ghi log và xử lý các sự kiện thất bại trở nên phức tạp và cần có các công cụ quản lý chuyên dụng.

Kiến trúc hướng sự kiện mang lại sự linh hoạt, khả năng mở rộng và hiệu suất cao cho các hệ thống cần xử lý nhiều tác vụ bất đồng bộ. Đó là lựa chọn phù hợp cho các ứng dụng thời gian thực, xử lý dữ liệu lớn và các hệ thống cần phản hồi nhanh trước các thay đổi trong môi trường hoạt động.

#### **Client-Server Architecture (Kiến trúc Client-Server)**

Kiến trúc Client-Server là mô hình thiết kế phần mềm trong đó các chức năng được phân chia rõ ràng giữa máy khách (Client) và máy chủ (Server). Trong mô hình này, máy khách chịu trách nhiệm thu thập yêu cầu từ người dùng và gửi chúng đến máy chủ, trong khi máy chủ đảm nhiệm việc xử lý các yêu cầu đó, truy xuất dữ liệu và trả về kết quả cho máy khách. Sự phân chia này giúp tối ưu hóa hiệu năng, bảo mật và khả năng quản lý hệ thống.

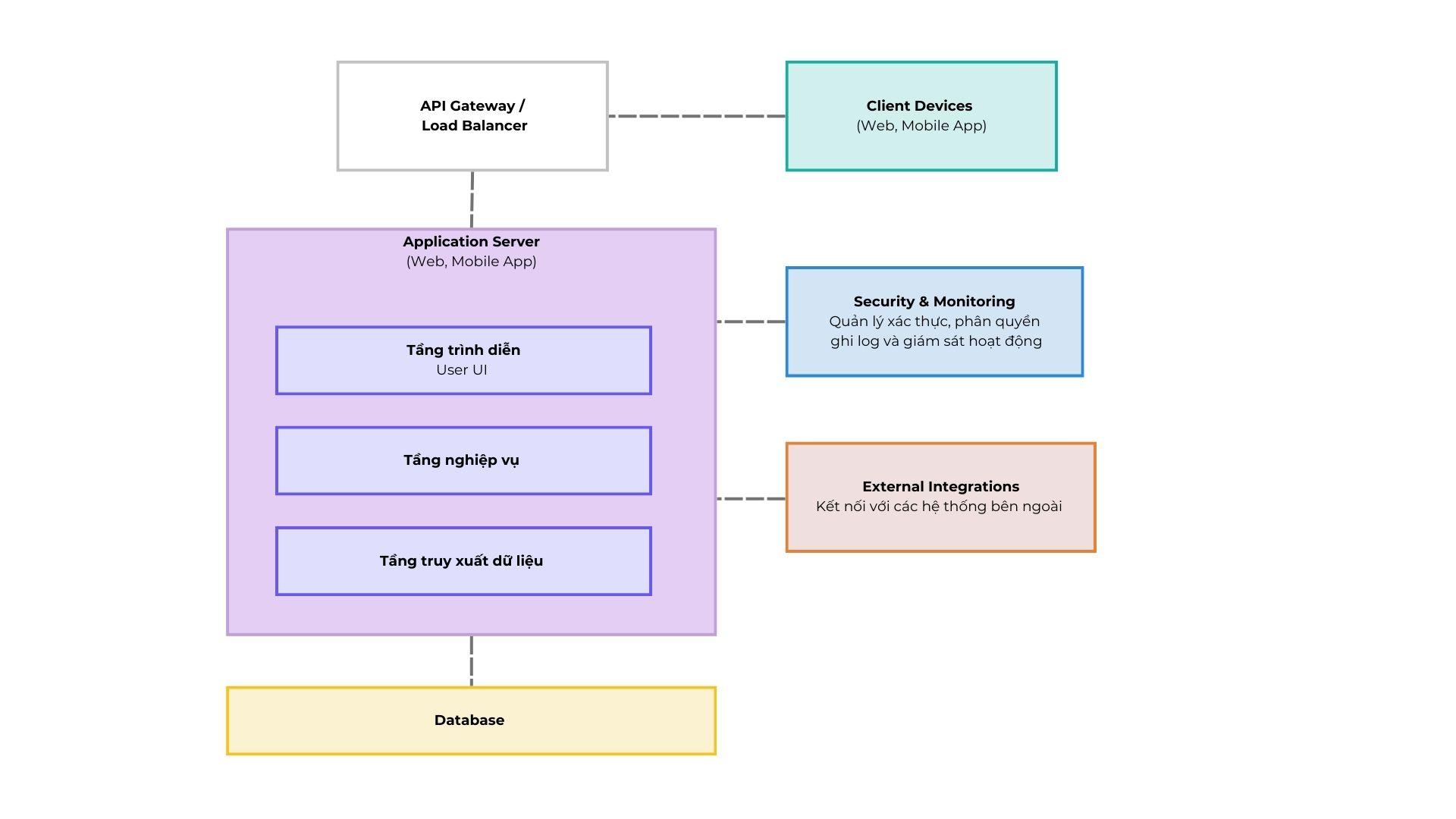
Máy khách thường là các ứng dụng giao diện người dùng như trình duyệt web, ứng dụng di động hoặc phần mềm desktop, nơi người dùng tương tác trực tiếp với hệ thống. Ngược lại, máy chủ là trung tâm xử lý, nơi thực hiện các logic nghiệp vụ phức tạp, lưu trữ dữ liệu và quản lý phiên làm việc. Việc tập trung xử lý trên máy chủ giúp các biện pháp bảo mật, kiểm soát truy cập và quản lý dữ liệu được triển khai một cách hiệu quả.

Ưu điểm của kiến trúc Client-Server nằm ở khả năng bảo trì và mở rộng hệ thống. Khi các thành phần được tách biệt, việc cập nhật hay nâng cấp máy chủ không làm gián đoạn trải nghiệm của người dùng tại máy khách. Hơn nữa, hệ thống có thể áp dụng các giải pháp cân bằng tải, dự phòng và tối ưu hóa hiệu năng để đảm bảo hoạt động liên tục và ổn định.

Tuy nhiên, nhược điểm của mô hình này là sự phụ thuộc lớn vào máy chủ. Nếu máy chủ gặp sự cố hoặc bị quá tải, toàn bộ hệ thống có thể bị ảnh hưởng. Do đó, việc triển khai các biện pháp phòng ngừa như dự phòng máy chủ, sử dụng kiến trúc phân tán hoặc tích hợp thêm các công nghệ hỗ trợ là rất cần thiết.

Nhìn chung, kiến trúc Client-Server là nền tảng của nhiều ứng dụng hiện đại, từ website, email đến các dịch vụ trực tuyến, nhờ vào khả năng tách biệt rõ ràng giữa giao diện người dùng và xử lý dữ liệu, giúp hệ thống trở nên linh hoạt, dễ quản lý và an toàn hơn.

### Kiến trúc tổng thể của hệ thống



Hình 4.2 Kiến trúc tổng thể của hệ thống

**Client Devices (Thiết bị người dùng):** Điểm giao tiếp đầu tiên giữa người dùng và hệ thống. Người dùng, bao gồm nhân viên bán thuốc, quản trị viên hoặc khách hàng, sẽ truy cập vào hệ thống thông qua các trình duyệt web hoặc ứng dụng di động. Giao diện này cho phép người dùng nhập dữ liệu, tìm kiếm thông tin thuốc, tạo đơn hàng và theo dõi tình trạng giao dịch, tạo nên trải nghiệm trực quan và thân thiện.

**API Gateway / Load Balancer:** Thành phần này đóng vai trò là cổng trung gian giữa các thiết bị người dùng và máy chủ ứng dụng. API Gateway chịu trách nhiệm tiếp nhận tất cả các yêu cầu từ phía client, chuyển đổi hoặc định tuyến chúng đến các dịch vụ phù hợp. Bên cạnh đó, Load Balancer đảm bảo phân phối đều lưu lượng truy cập đến nhiều máy chủ, giúp tránh tình trạng quá tải, tăng cường tính sẵn sàng và đảm bảo hiệu suất của hệ thống khi có lượng truy cập lớn.

**Application Server (Máy chủ ứng dụng):** Đây là nơi triển khai mã nguồn và thực thi các quy trình nghiệp vụ của hệ thống. Sử dụng nền tảng ASP.NET Core MVC, máy chủ ứng dụng được tổ chức thành nhiều tầng chức năng. Tầng trình diễn (Presentation Layer) xử lý giao diện và tương tác với người dùng, tầng nghiệp vụ (Business Logic Layer) chứa các quy tắc và quy trình xử lý giao dịch bán thuốc, đơn thuốc, kiểm soát tồn kho… còn tầng truy xuất dữ liệu (Data Access Layer) đảm bảo việc giao tiếp với cơ sở dữ liệu diễn ra an toàn và hiệu quả.

**Database Server (Máy chủ cơ sở dữ liệu):** Đây là thành phần lưu trữ toàn bộ thông tin của hệ thống, từ danh mục thuốc, thông tin đơn hàng, dữ liệu người dùng cho đến lịch sử giao dịch. Việc sử dụng hệ quản trị cơ sở dữ liệu như SQL Server giúp đảm bảo dữ liệu được lưu trữ một cách có cấu trúc, an toàn và có khả năng mở rộng khi hệ thống phát triển.

**Security & Monitoring (Bảo mật và giám sát):** Để đảm bảo hệ thống hoạt động an toàn và tuân thủ các tiêu chuẩn nghiêm ngặt, thành phần này thực hiện các biện pháp bảo mật như xác thực, phân quyền, mã hóa dữ liệu và ghi log hoạt động. Đồng thời, hệ thống giám sát sẽ theo dõi hiệu suất, ghi nhận lỗi và phát hiện sớm các sự cố, từ đó hỗ trợ quá trình xử lý và bảo trì hệ thống một cách kịp thời.

**External Integrations (Tích hợp bên ngoài):** Hệ thống quản lý bán thuốc không hoạt động độc lập mà cần tương tác với các dịch vụ bên ngoài như hệ thống báo cáo, kiểm soát chất lượng thuốc hay các dịch vụ thanh toán. Thành phần tích hợp này thường sử dụng các API để trao đổi dữ liệu một cách bất đồng bộ, đảm bảo thông tin được chuyển giao chính xác và nhanh chóng giữa các hệ thống.

Những thành phần trên kết hợp với nhau tạo nên một kiến trúc tổng thể chặt chẽ, đảm bảo rằng hệ thống quản lý bán thuốc không chỉ đáp ứng được các yêu cầu nghiệp vụ và chuẩn GPP mà còn có khả năng mở rộng, bảo trì và duy trì hiệu suất ổn định trong quá trình vận hành.

### Lựa chọn công nghệ phát triển cho hệ thống

* **Front-end:** Sử dụng ASP.NET Core MVC với Razor Pages để xây dựng giao diện server-rendered, tận dụng các tính năng tích hợp sẵn của .NET cho bảo mật và hiệu năng.
* **Back-end:** ASP.NET Core là nền tảng phù hợp cho việc phát triển backend với ngôn ngữ C#. Với khả năng hỗ trợ đa luồng, tối ưu hiệu năng và bảo mật, ASP.NET Core giúp xử lý các nghiệp vụ liên quan đến quản lý bán thuốc, kiểm soát tồn kho, đơn hàng và các quy trình nghiệp vụ khác một cách hiệu quả.
* **Cơ sở dữ liệu:** Microsoft SQL Server – một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, ổn định và dễ tích hợp với .NET. Cho phép truy cập dữ liệu trong hệ thống cơ sở dữ liệu quan hệ, cho phép xác định dữ liệu và thao tác với dữ liệu. Lưu được lượng dữ liệu lớn và tốc độ truy vấn nhanh.
* **Công cụ được sử dụng để phát triển:** Visual Studio Code, Microsoft SQL Server 2019

### Kiến trúc phần mềm của hệ thống

Hệ thống phần mềm quản lý nhà thuốc **Hồng Vân GPP** được thiết kế theo **kiến trúc phân lớp (Layered Architecture)** kết hợp với mô hình **MVC (Model-View-Controller)**. Sự kết hợp này giúp hệ thống đạt được **tính linh hoạt, dễ bảo trì, hiệu suất cao và bảo mật**, đồng thời đảm bảo khả năng mở rộng trong tương lai.

Hệ thống được tổ chức thành **các tầng chức năng (layers)** để đảm bảo mỗi phần có nhiệm vụ riêng biệt, giúp quản lý mã nguồn dễ dàng và giảm thiểu sự phụ thuộc giữa các thành phần. Cấu trúc gồm các tầng chính:

1. **Presentation Layer (Giao diện người dùng):** Hiển thị thông tin, nhận tương tác từ người dùng và gửi yêu cầu đến tầng xử lý.
2. **Application Layer (Xử lý nghiệp vụ):** Chứa các quy tắc nghiệp vụ, xác thực dữ liệu và thực hiện các thao tác cần thiết trước khi truy cập dữ liệu.
3. **Domain Layer (Quản lý dữ liệu & logic cốt lõi):** Định nghĩa các đối tượng dữ liệu và quy tắc nghiệp vụ như quản lý thuốc, đơn hàng, khách hàng, v.v.
4. **Data Access Layer (Tầng truy xuất dữ liệu):** Giao tiếp với cơ sở dữ liệu, cung cấp các phương thức đọc/ghi dữ liệu một cách hiệu quả.

Việc phân chia theo Layered Architecture giúp **dễ dàng thay đổi giao diện mà không ảnh hưởng đến logic nghiệp vụ**, hoặc thay đổi cơ sở dữ liệu mà không cần chỉnh sửa toàn bộ hệ thống. Trong tầng Presentation và Application, hệ thống áp dụng mô hình **MVC**, giúp tách biệt giữa giao diện, xử lý và dữ liệu.

* **Model (M):** Đại diện cho dữ liệu và các quy tắc nghiệp vụ, đảm bảo toàn vẹn dữ liệu và tính nhất quán.
* **View (V):** Giao diện hiển thị thông tin, nhận dữ liệu từ Model để hiển thị cho người dùng.
* **Controller (C):** Tiếp nhận yêu cầu từ người dùng, xử lý logic và gọi Model để lấy dữ liệu phù hợp, sau đó trả kết quả về View.

Mô hình **MVC giúp tổ chức mã nguồn chặt chẽ, dễ mở rộng và bảo trì**. Khi cần thay đổi giao diện hoặc logic nghiệp vụ, các thành phần có thể được cập nhật độc lập mà không làm ảnh hưởng đến các phần khác của hệ thống.

**Lý do lựa chọn:** Hệ thống quản lý nhà thuốc Hồng Vân GPP yêu cầu một kiến trúc phần mềm có khả năng đáp ứng các tiêu chí quan trọng như hiệu suất cao, độ tin cậy, bảo mật, khả năng mở rộng và dễ bảo trì. Việc áp dụng Layered Architecture kết hợp với MVC là lựa chọn phù hợp vì nó cung cấp một mô hình tổ chức chặt chẽ, giúp hệ thống hoạt động ổn định, dễ dàng mở rộng trong tương lai và đảm bảo chất lượng phần mềm theo tiêu chuẩn.

### **1. Đáp ứng Yêu cầu Độ Tin Cậy:** Kiến trúc **Layered Architecture** giúp hệ thống vận hành ổn định nhờ việc tách biệt các thành phần thành nhiều lớp độc lập. **Tầng xử lý nghiệp vụ (Application Layer)** đảm bảo các quy tắc kinh doanh được thực thi chặt chẽ, giảm thiểu lỗi khi thao tác dữ liệu. Bên cạnh đó, nhờ kiến trúc phân lớp, hệ thống có thể triển khai các cơ chế kiểm tra, sao lưu định kỳ và phục hồi dữ liệu tự động, đảm bảo quá trình vận hành không bị gián đoạn. Nếu một tầng gặp sự cố, các tầng khác vẫn có thể tiếp tục hoạt động, giúp hệ thống duy trì tính liên tục ngay cả trong trường hợp có lỗi xảy ra.

### **2. Đáp ứng Yêu Cầu Hiệu Suất:** Với yêu cầu xử lý 95% giao dịch dưới 1 giây và hỗ trợ tối thiểu 50 người dùng, hệ thống áp dụng **cơ chế tối ưu hóa truy vấn dữ liệu trong tầng Data Access**, sử dụng ORM (Object-Relational Mapping) để giảm tải cho cơ sở dữ liệu. Mô hình **MVC giúp giảm số lượng truy vấn không cần thiết**, vì Controller chỉ gọi Model khi có yêu cầu thực sự. Ngoài ra, hệ thống có thể tích hợp **caching (Redis, Memcached)** và **cân bằng tải (Load Balancer)** để đảm bảo hiệu suất không bị ảnh hưởng khi số lượng người dùng tăng lên. Nhờ đó, hệ thống có thể xử lý **20.000 giao dịch/ngày** trong điều kiện tải cao mà vẫn đảm bảo tốc độ phản hồi nhanh.

### **3. Đáp ứng Yêu Cầu Quản Lý Dữ Liệu:** Trong mô hình **MVC**, **Model** đóng vai trò quản lý dữ liệu, đảm bảo thông tin luôn chính xác và toàn vẹn. Hệ thống sử dụng **tầng Data Access** để kiểm soát quyền truy cập dữ liệu, ngăn chặn các lỗi thao tác như nhập sai số lượng tồn kho hoặc chỉnh sửa hóa đơn sau khi hoàn tất. Các quy tắc nghiệp vụ như **đơn hàng phải có ít nhất một sản phẩm, số lượng tồn kho không thể âm** được thực thi chặt chẽ trong tầng xử lý nghiệp vụ (Application Layer). Ngoài ra, dữ liệu được tổ chức theo mô hình quan hệ, đảm bảo thông tin thuốc liên kết với nhà cung cấp, đơn hàng liên kết với khách hàng và nhân viên, giúp quản lý dễ dàng và logic.

### **4. Đáp ứng Yêu Cầu Giao Diện:** Nhờ áp dụng **MVC**, hệ thống có thể dễ dàng thay đổi giao diện mà không ảnh hưởng đến logic xử lý. **Tầng Presentation (giao diện người dùng) được thiết kế linh hoạt**, giúp hiển thị thông tin thuốc, đơn hàng, khách hàng theo đúng tiêu chuẩn GPP. Giao diện hỗ trợ **đa ngôn ngữ (tiếng Việt, tiếng Anh), chế độ Dark Mode, tùy chỉnh kích thước chữ**, giúp người dùng có trải nghiệm tốt hơn. Ngoài ra, việc sử dụng kiến trúc phân lớp giúp hệ thống **dễ dàng mở rộng sang các nền tảng khác như mobile app hoặc web app** mà không cần thay đổi toàn bộ hệ thống.

### **5. Đáp ứng Yêu Cầu Bảo Mật:** Hệ thống áp dụng các **cơ chế bảo mật chặt chẽ ngay từ tầng xử lý nghiệp vụ**, đảm bảo dữ liệu và quyền truy cập được kiểm soát nghiêm ngặt. **MVC giúp phân quyền rõ ràng**, chỉ cho phép nhân viên thực hiện các thao tác phù hợp với vai trò của họ, trong khi quản trị viên có quyền kiểm soát toàn bộ hệ thống. Tầng Data Access được bảo vệ bằng **các biện pháp chống SQL Injection, CSRF, XSS**, đồng thời sử dụng **OAuth, JWT hoặc xác thực hai lớp (2FA) cho các tài khoản quan trọng**. Hệ thống cũng có **cơ chế sao lưu dữ liệu định kỳ** và **kịch bản phục hồi dữ liệu khẩn cấp**, đảm bảo tính liên tục của hoạt động ngay cả khi gặp sự cố.

### **6. Đáp ứng Yêu Cầu Tuân Thủ & Ràng Buộc:** Kiến trúc **Layered Architecture kết hợp với MVC** giúp hệ thống dễ bảo trì, mở rộng và tuân thủ các tiêu chuẩn ngành. Do các thành phần được tách biệt rõ ràng, việc nâng cấp hay thay đổi công nghệ không ảnh hưởng đến toàn bộ hệ thống. Hệ thống được thiết kế để **hoạt động tốt trên nhiều thiết bị và hệ điều hành**, dễ dàng tích hợp với **các phần mềm hóa đơn điện tử, kế toán hoặc hệ thống quản lý dược phẩm khác**. Ngoài ra, hệ thống đảm bảo **tuân thủ quy định của Bộ Y tế về quản lý nhà thuốc** và đáp ứng tiêu chuẩn GPP về quản lý chất lượng thuốc.

### Cơ sở dữ liệu mức vật lý của hệ thống

#### Chuyển đổi từ tên thực thể sang tên bảng

| **Tên thực thể** | **Tên bảng** |
| --- | --- |
| Nhân viên | tblNhanVien |
| Khách hàng | tblKhachHang |
| Loại sản phẩm | tblLoaiSanPham |
| Nhà cung cấp | tblNhaCungCap |
| Loại sản phẩm | tblLoaiSanPham |
| Sản phẩm | tblSanPham |
| Lô sản phẩm | tblLoSanPham |
| Báo cáo | tblBaoCao |
| Đơn thuốc kê đơn | tblThuocKeDon |
| Chi tiết đơn thuốc kê đơn | tblChiTietThuocKeDon |
| Hóa đơn | tblHoaDon |
| Chi tiết hóa đơn | tblChiTietHoaDon |
| Khiếu nại | tblKhieuNai |
| Đơn nhập hàng | tblDonNhap |
| Chi tiết đơn nhập hàng | tblChiTietDonNhap |

Bảng 4.1 Chuyển đổi từ tên thực thể sang tên bảng

#### Xác định bảng cơ sở dữ liệu

tblNhanVien(PK\_iMaNV, sTenDangNhap, sMatKhau, sHoTen, sChucVu, fLuongCB, fHeSoLuong, dNgayVaoLam)

tblNhaCungCap(PK\_iMaNCC, sTenNCC, sDiaChi, sSDT, sEmail, bChungChi)

tblKhachHang(PK\_iMaKH, sHoTen, sTenDangNhap, sMatKhau, sEmail, sSDT, dNgayTao)

tblLoaiSanPham(PK\_iMaLoaiSanPham, sTenLoai, sMoTa)

tblSanPham(PK\_iMaSP, sTen, sDangThuoc, sDonViTinh, iSoLuong, fGiaNhap, fGiaBan, sCachSuDung, sDieuKienBaoQuan, FK\_iMaNCC, FK\_iMaLoaiSanPham , sHanSuDung, bBietTru, bThuHoi, bCanKeDon)

tblLoSanPham(PK\_iLoId, FK\_iSanPhamId, dNgaySanXuat, dNgayHetHan, iSoLuongNhap, iSoLuongTon, dNgayNhap, FK\_iDonNhapId, fGiaNhap, sDonViTinh)

tblThuocKeDon(PK\_iMaTKD, FK\_iMaKH, FK\_iMaNV, dNgayTao, sGhiChu)

tblChiTietThuocKeDon(PK\_iMaCTTKD, FK\_iMaTKD, FK\_iMaSP, sLieuLuong, sThoiLuongSD)

tblHoaDon(PK\_iMaHD, FK\_iMaKH, dThoiGian, fTongGiaTri, sGhiChu)

tblChiTietHoaDon(PK\_iMaCTHD, FK\_iMaHD, FK\_iMaSP, iSoLuong, iLieuLuong, sGhiChu)

tblKhieuNai(PK\_iMaKN, dThoiGian, FK\_iMaKH, sNguyenNhan, bDaGiaiQuyet)

tblNhapHang(PK\_iMaNH, dNgayTao, FK\_iMaNV, FK\_iMaNCC, fTongTien)

**Ghi chú**

PK\_ trước tên cột là khóa chính (Primary Key).

FK\_ trước tên cột là khóa ngoại (Foreign Key).

#### Xác định các mối quan hệ

| Bảng | Quan hệ | Bảng |
| --- | --- | --- |
| tblNhanVien | Tạo(1-n) | tblThuocKeDon |
| tblNhanVien | Tạo(1-n) | tblDonNhap |
| tblKhachHang | Có(1-n) | tblThuocKeDon |
| tblKhachHang | Gửi(1-n) | tblKhieuNai |
| tblKhachHang | Có(1-n) | tblHoaDon |
| tblLoaiSanPham | Có(1-n) | tblSanPham |
| tblSanPham | Tồn tại(m-n) | tblChiTietHoaDon |
| tblSanPham | Tồn tại(m-n) | tblChiTietThuocKeDon |
| tblSanPham | Tồn tại(m-n) | tblLoSanPham |
| tblHoaDon | Có(1-n) | tblChiTietHoaDon |
| tblDonNhap | Có(1-n) | tblLoSanPham |
| tblThuocKeDon | Có(1-n) | tblChiTietThuocKeDon |
| tblNhaCungCap | Cung cấp(1-n) | tblDonNhap |
| tblNhaCungCap | Cung cấp(1-n) | tblSanPham |

Bảng 4.2 Xác định các mối quan hệ

#### Đặc tả dữ liệu hệ thống

| **tblNhanVien** | | |
| --- | --- | --- |
| iMaNV | INT | Mã nhân viên, khóa chính. |
| sTenDangNhap | NVARCHAR(50) | Tên đăng nhập của nhân viên. |
| sMatKhau | NVARCHAR(255) | Mật khẩu đăng nhập của nhân viên. |
| sHoTen | NVARCHAR(255) | Họ và tên đầy đủ của nhân viên. |
| sChucVu | NVARCHAR(20) | Chức vụ: 'Nhân viên' hoặc 'Quản lý'. |
| fLuongCB | DECIMAL(10,2) | Lương cơ bản của nhân viên. |
| fHeSoLuong | DECIMAL(10,2) | Hệ số lương của nhân viên. |
| dNgayVaoLam | DATE | Ngày bắt đầu làm việc của nhân viên. |

Bảng 4.3 Đặc tả dữ liệu hệ thống bảng nhân viên

| **tblNhaCungCap** | | |
| --- | --- | --- |
| iMaNCC | INT | Mã nhà cung cấp, khóa chính. |
| sTenNCC | NVARCHAR(255) | Tên nhà cung cấp. |
| sDiaChi | NVARCHAR(MAX) | Địa chỉ của nhà cung cấp. |
| sSDT | NVARCHAR(20) | Số điện thoại nhà cung cấp. |
| sEmail | NVARCHAR(255) | Email nhà cung cấp. |
| bChungChi | VARBINARY(MAX) | Chứng chỉ (file quét hoặc file đính kèm) của nhà cung cấp. |

Bảng 4.4 Đặc tả dữ liệu hệ thống bảng nhà cung cấp

| **tblKhachHang** | | |
| --- | --- | --- |
| iMaKH | INT IDENTITY(1,1) | Mã khách hàng, khóa chính. |
| sHoTen | NVARCHAR(255) | Họ và tên đầy đủ của khách hàng. |
| sTenDangNhap | NVARCHAR(50) | Tên đăng nhập (duy nhất) của khách hàng. |
| sMatKhau | NVARCHAR(255) | Mật khẩu đăng nhập của khách hàng. |
| sEmail | NVARCHAR(255) | Email của khách hàng (duy nhất, có thể cho phép NULL nếu muốn). |
| sSDT | NVARCHAR(20) | Số điện thoại khách hàng. |
| dNgayTao | DATETIME | Thời điểm tạo tài khoản, mặc định lấy thời điểm hiện tại. |

Bảng 4.5 Đặc tả dữ liệu hệ thống bảng khách hàng

| **tblLoaiSanPham** | | |
| --- | --- | --- |
| iMaLoaiSanPham | INT | Khóa chính tự tăng, định danh duy nhất cho mỗi loại sản phẩm |
| sTenLoai | NVARCHAR(max) | Tên của loại sản phẩm |
| sMoTa | NVARCHAR(max) | Mô tả của loại sản phẩm |

Bảng 4.6 Đặc tả dữ liệu hệ thống bảng loại sản phẩm

| **tblSanPham** | | |
| --- | --- | --- |
| iMaSP | INT | Mã sản phẩm, khóa chính. |
| sTen | NVARCHAR(255) | Tên sản phẩm. |
| sDangThuoc | NVARCHAR(100) | Dạng thuốc (viên, bột, siro, ...), nếu là thuốc. |
| sDonViTinh | NVARCHAR(50) | Đơn vị tính của sản phẩm. |
| iSoLuong | INT | Số lượng tồn kho (mặc định 0). |
| fGiaNhap | DECIMAL(10,2) | Giá nhập sản phẩm. |
| fGiaBan | DECIMAL(10,2) | Giá bán sản phẩm. |
| sCachSuDung | NVARCHAR(MAX) | Cách sử dụng (nếu là thuốc). |
| sDieuKienBaoQuan | NVARCHAR(MAX) | Điều kiện bảo quản (nhiệt độ, độ ẩm...). |
| iMaNCC | INT | Khóa ngoại đến bảng tblNhaCungCap(iMaNCC). |
| sHanSuDung | NVARCHAR(100) | Hạn sử dụng. |
| bBietTru | BIT | Thuốc biệt trữ (1 = Có, 0 = Không). |
| bThuHoi | BIT | Thuốc thuộc diện thu hồi (1 = Có, 0 = Không). |
| bCanKeDon | BIT | Có cần kê đơn không (1 = có, 0 = không). |

Bảng 4.7 Đặc tả dữ liệu hệ thống bảng sản phẩm

| **tblLoSanPham** | | |
| --- | --- | --- |
| iLoId | INT | Khóa chính tự tăng, định danh duy nhất cho mỗi lô sản phẩm |
| iSanPhamId | INT | Khóa ngoại tham chiếu đến bảng Sản Phẩm, xác định lô này thuộc sản phẩm nào |
| dNgaySanXuat | DATE | Ngày sản xuất của lô sản phẩm |
| dNgayHetHan | DATE | Ngày hết hạn của lô sản phẩm |
| iSoLuongNhap | INT | Số lượng sản phẩm được nhập vào trong lô |
| iSoLuongTon | INT | Số lượng sản phẩm còn lại sau khi nhập và xuất bán (nếu có) |
| dNgayNhap | DATE | Ngày nhập lô sản phẩm vào kho |
| iDonNhapId | INT | Khóa ngoại tham chiếu đến bảng Đơn Nhập, xác định lô này thuộc về đơn nhập hàng nào |
| fGiaNhap | INT | Giá nhập cho mỗi đơn vị sản phẩm trong lô |
| sDonViTinh | VARCHAR(50) | Đơn vị tính của sản phẩm (vd: chai, hộp, kg…) |

Bảng 4.8 Đặc tả dữ liệu hệ thống lô sản phẩm

| **tblThuocKeDon** | | |
| --- | --- | --- |
| iMaTKD | INT | Mã đơn thuốc kê, khóa chính. |
| iMaKH | INT | Khóa ngoại đến bảng tblKhachHang(iMaKH). |
| iMaNV | INT | Khóa ngoại đến bảng tblNhanVien(iMaNV). |
| dNgayTao | DATETIME | Thời điểm tạo đơn thuốc. |
| sGhiChu | NVARCHAR(MAX | Ghi chú đơn thuốc (nếu có). |

Bảng 4.9 Đặc tả dữ liệu hệ thống thuốc kê đơn

| **tblChiTietThuocKeDon** | | |
| --- | --- | --- |
| iMaCTTKD | INT | Mã chi tiết đơn thuốc kê, khóa chính. |
| iMaTKD | INT | Khóa ngoại đến bảng tblThuocKeDon(iMaTKD). |
| iMaSP | INT | Khóa ngoại đến bảng tblSanPham(iMaSP). |
| sLieuLuong | NVARCHAR(100) | Liều lượng chỉ định. |
| sThoiLuongSD | NVARCHAR(100) | Thời lượng sử dụng (vd: 5 ngày, 10 ngày...). |

Bảng 4.10 Đặc tả dữ liệu hệ thống chi tiết thuốc kê đơn

| **tblHoaDon** | | |
| --- | --- | --- |
| iMaHD | INT | Mã hóa đơn, khóa chính. |
| iMaKH | INT | Khóa ngoại đến bảng tblKhachHang(iMaKH). |
| dThoiGian | DATETIME | Thời điểm lập hóa đơn. |
| fTongGiaTri | DECIMAL(10,2) | Tổng giá trị thanh toán của hóa đơn. |
| sGhiChu | NVARCHAR(MAX) | Ghi chú bổ sung cho hóa đơn (nếu cần). |

Bảng 4.11 Đặc tả dữ liệu hệ thống hóa đơn

| **tblChiTietHoaDon** | | |
| --- | --- | --- |
| iMaCTHD | INT | Mã chi tiết hóa đơn, khóa chính. |
| iMaHD | INT | Khóa ngoại đến bảng tblHoaDon(iMaHD). |
| iMaSP | INT | Khóa ngoại đến bảng tblSanPham(iMaSP). |
| iSoLuong | INT | Số lượng sản phẩm trong hóa đơn. |
| iLieuLuong | INT | Liều dùng hoặc số lần dùng (trường hợp thuốc). |
| sGhiChu | NVARCHAR(MAX) | Ghi chú chi tiết (nếu có). |

Bảng 4.12 Đặc tả dữ liệu hệ thống chi tiết hóa đơn

| **tblKhieuNai** | | |
| --- | --- | --- |
| iMaKN | INT | Mã khiếu nại, khóa chính. |
| dThoiGian | DATETIME | Thời gian tạo khiếu nại. |
| iMaKH | INT | Khóa ngoại đến bảng tblKhachHang(iMaKH). |
| sNguyenNhan | NVARCHAR(MAX) | Nội dung hoặc nguyên nhân khiếu nại. |
| bDaGiaiQuyet | BIT | Đã giải quyết hay chưa (1 = đã giải quyết, 0 = chưa). |

Bảng 4.13 Đặc tả dữ liệu hệ thống khiếu nại

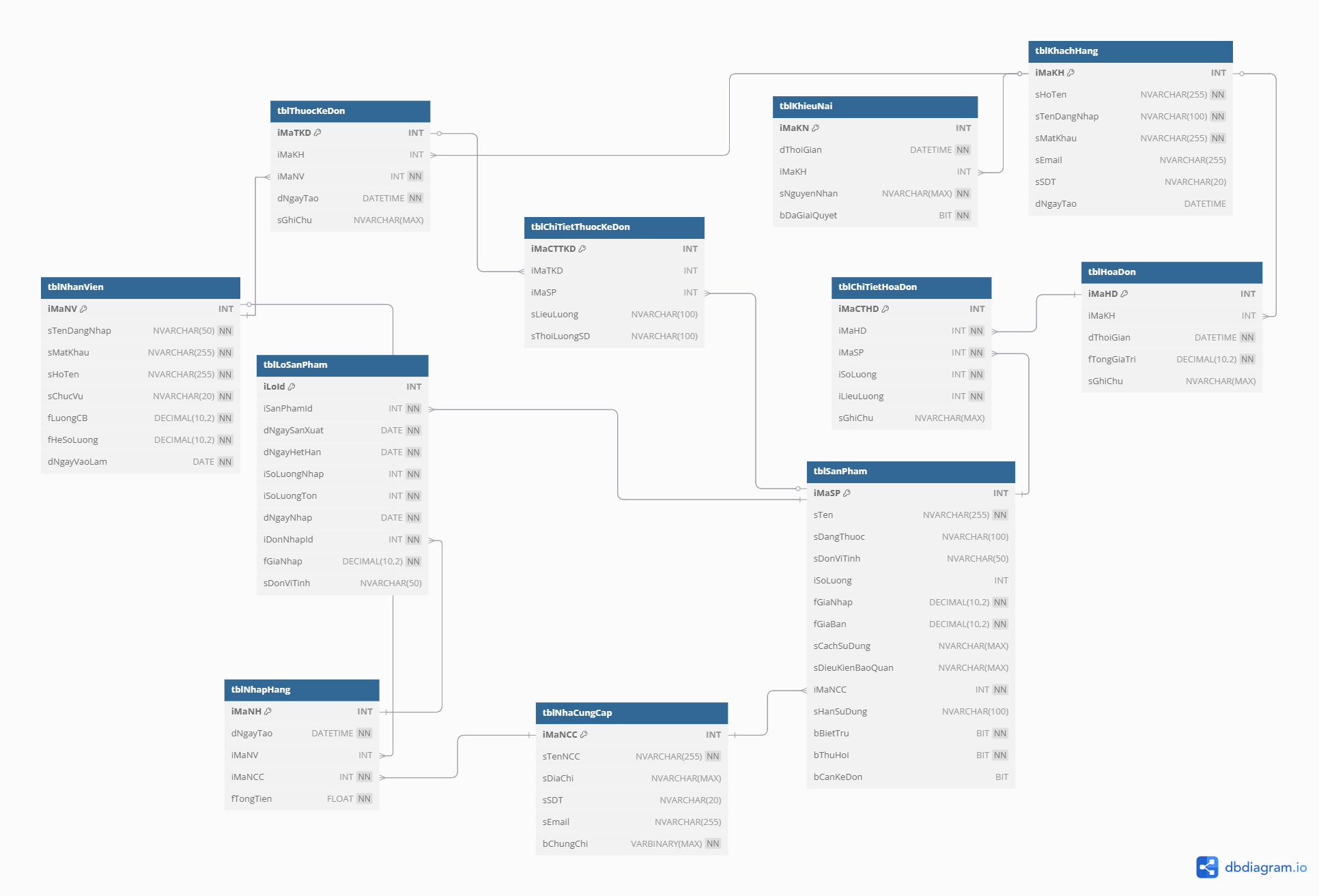
| **tblNhapHang** | | |
| --- | --- | --- |
| iMaNH | INT | Mã đơn nhập hàng, khóa chính. |
| dNgayTao | DATETIME | Thời gian lập đơn nhập. |
| iMaNV | INT | Khóa ngoại đến bảng tblNhanVien(iMaNV). |
| iMaNCC | INT | Khóa ngoại đến bảng tblNhaCungCap(iMaNCC). |
| fTongTien | FLOAT | Tổng tiền cho đơn nhập hàng này. |

Bảng 4.14 Đặc tả dữ liệu hệ thống nhập hàng

| **tblLoSanPham** | | |
| --- | --- | --- |
| lo\_id | INT | Mã lô sản phẩm, khóa chính. |
| iMaNH | INT | Khóa ngoại đến bảng tblNhapHang(iMaNH). |
| iMaSP | INT | Khóa ngoại đến bảng tblSanPham(iMaSP). |
| iSoLuong | INT | Số lượng nhập cho sản phẩm này. |
| fGiaNhap | DECIMAL(10,2) | Giá nhập (đơn giá) cho sản phẩm tại thời điểm nhập. |
| sDonViTinh | NVARCHAR(50) | Đơn vị tính ghi nhận khi nhập (có thể khác với đơn vị tính trong bảng sản phẩm). |
| dNgaySanXuat | DateTime | Ngày sản xuất |
| dHSD | NVARCHAR(50) | Hạn sử dụng (nếu là thuốc). |

Bảng 4.15 Đặc tả dữ liệu hệ thống lô sản phẩm

#### Lược đồ cơ sở dữ liệu

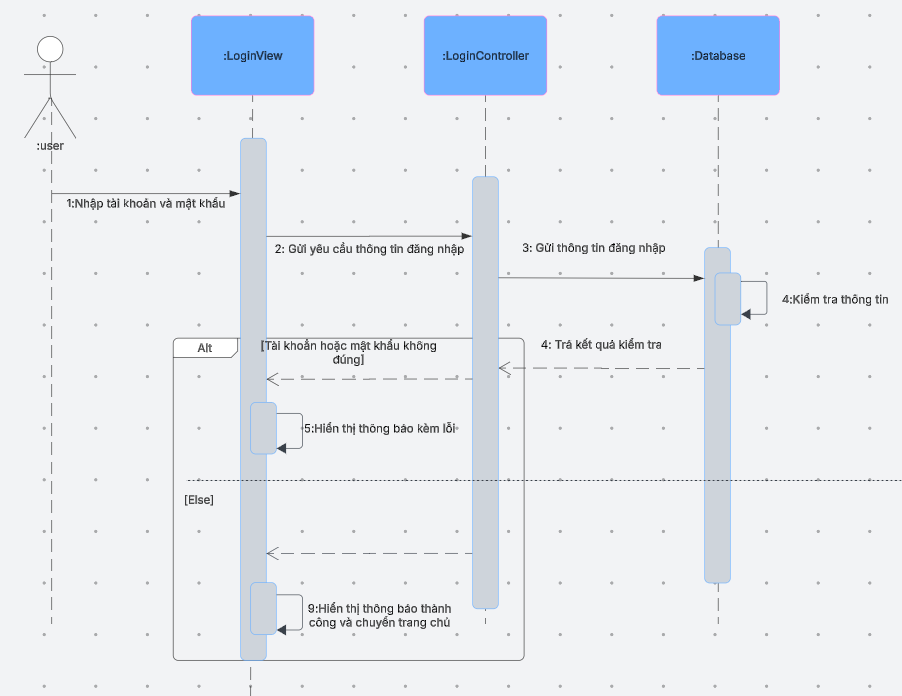


Hình 4.3: Lược đồ cơ sở dữ liệu

### Một số thiết kế chi tiết theo kiến trúc được chọn

#### Thiết kế xử lý chức năng đăng nhập

* Input: sTenDangNhap, sMatKhau
* Output: Chuyển sang trang chủ
* Process:
* Người dùng nhập thông tin tài khoản và mật khẩu và bấm nút “Đăng nhập”
* Nếu dữ liệu hợp lệ thì sẽ đưa ra thông báo “Đăng nhập thành công”, chuyển người dùng sang trang kế tiếp với quyền tương ứng.
* Nếu 1 trong 2 trường hoặc cả 2 đều trống thì đưa ra thông báo “Vui lòng nhập đầy đủ thông tin” và yêu cầu nhập lại.
* Nếu dữ liệu về tài khoản và mật khẩu không đúng, thì hiện ra thông báo “Tài khoản hoặc mật khẩu không chính xác” và yêu cầu nhập lại.
* Biểu đồ tuần tự

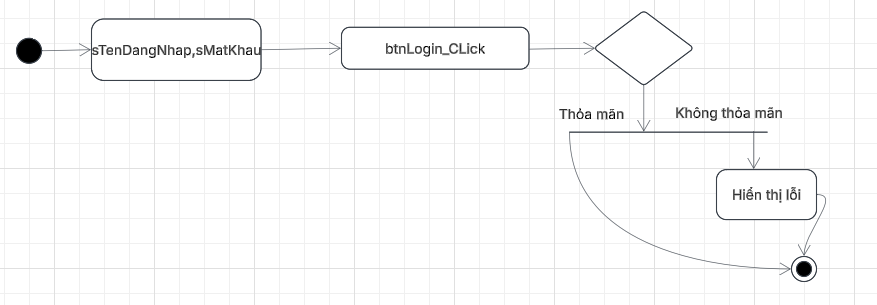


* Các unit cần có

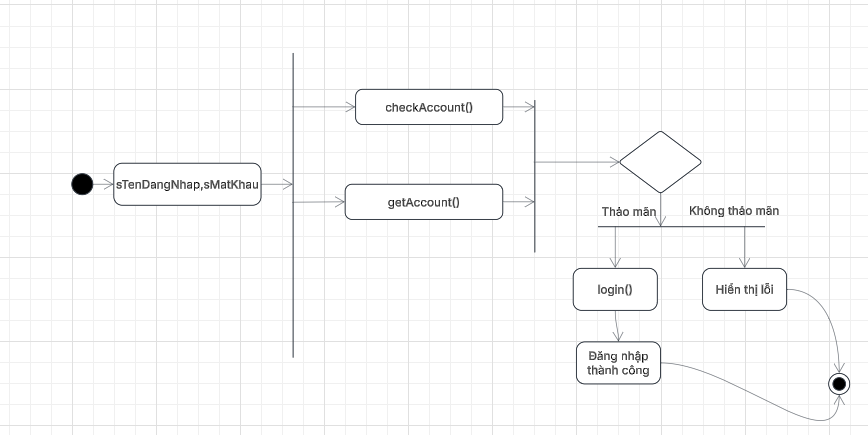
| STT | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | LoginView | btnLogin\_Click() | sTenDangNhap, sMatKhau | “Vui lòng nhập đầy đủ thông tin” hoặc không trả về |
| 2 | LoginController | login() | sTenDangNhap, sMatKhau | “Tài khoản hoặc mật khẩu không chính xác” hoặc không trả về |
| 3 | LoginModel | checkAccount() | sTenDangNhap, sMatKhau | Trả về kết quả đăng nhập thành công hay thất bại |
| 4 | Database | getAccount() | sTenDangNhap, sMatKhau | Thông tin tài khoản |

* Thiết kế từng unit

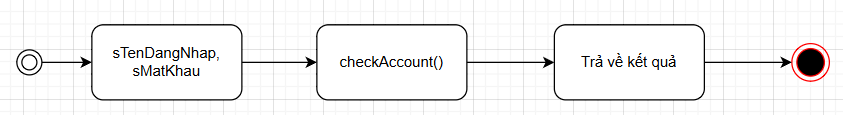
1. btnLogin\_Click()



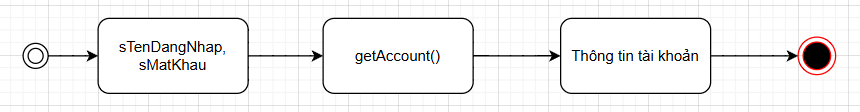
1. login()



1. checkAccount()

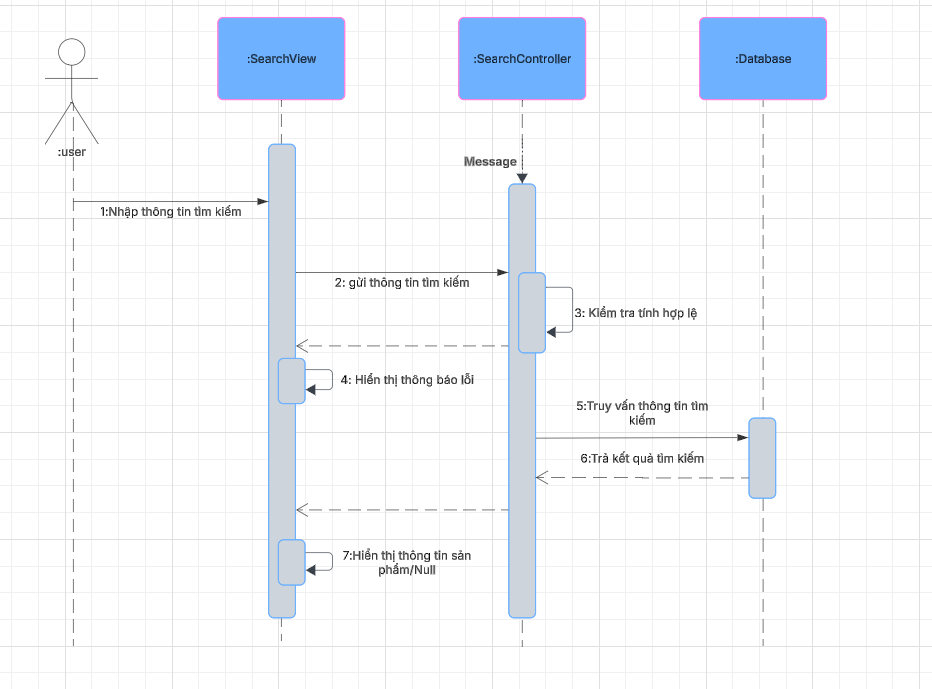


1. getAccount()



#### Thiết kế xử lý chức năng tìm kiếm sản phẩm

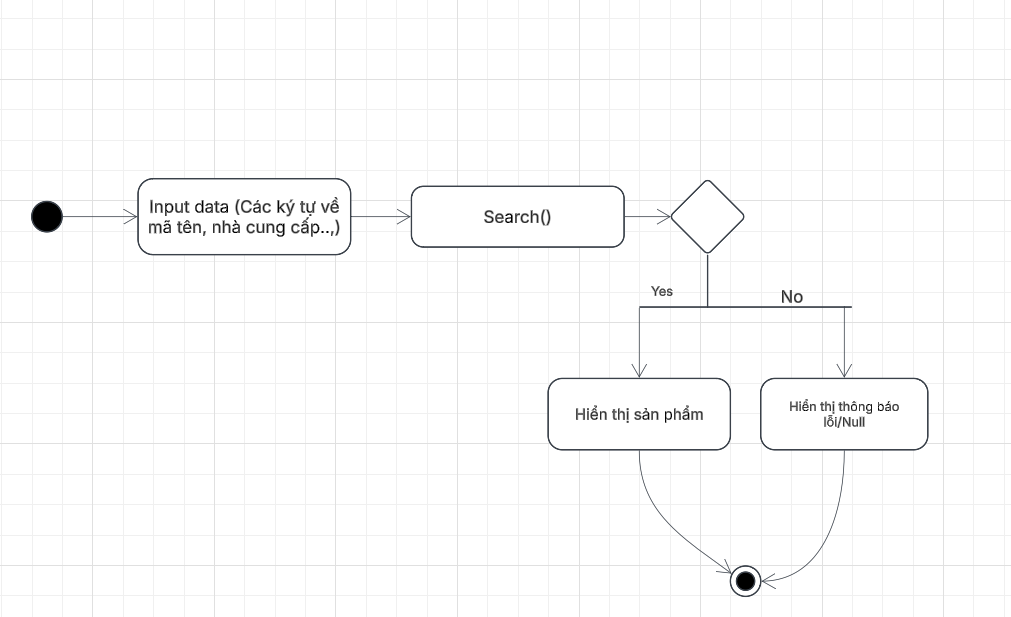
* Input: Các ký tự, lựa chọn chứa tên sản phẩm, mã sản phẩm hoặc loại thuốc, nhà sản xuất, hãng
* Output:
  + Danh sách sản phẩm phù hợp
  + Thông báo lỗi hoặc cảnh báo như không có sản phẩm phù hợp hoặc dữ liệu đầu vào không hợp lệ.
* Qui trình xử lý
  + Người dùng nhập từ khóa để tìm kiếm danh sách sản phẩm phù hợp.
  + Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào.
  + Hệ thống truy vấn từ cơ sở dữ liệu.
  + Hệ thống gửi kết quả tới giao diện người dùng
  + Đưa ra thông báo lỗi nếu có.
* Biểu đồ tuần tự:



* Các unit cần có:

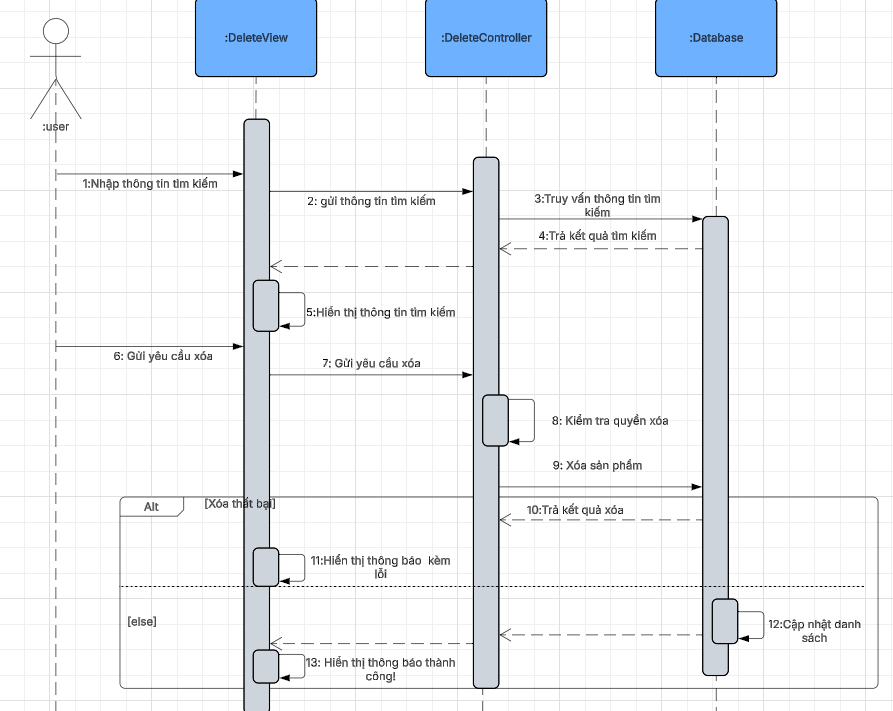
| STT | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | ProductController | search() | Các ký tự chứa tên sản phẩm, mã sản phẩm hoặc loại thuốc, nhà sản xuất, hãng. | Danh sách thông tin sản phẩm, đối tượng phù hợp hoặc thông báo không có. |

* Thiết kế cho từng unit:
  + Search():



#### Thiết kế xử lý chức năng xóa sản phẩm

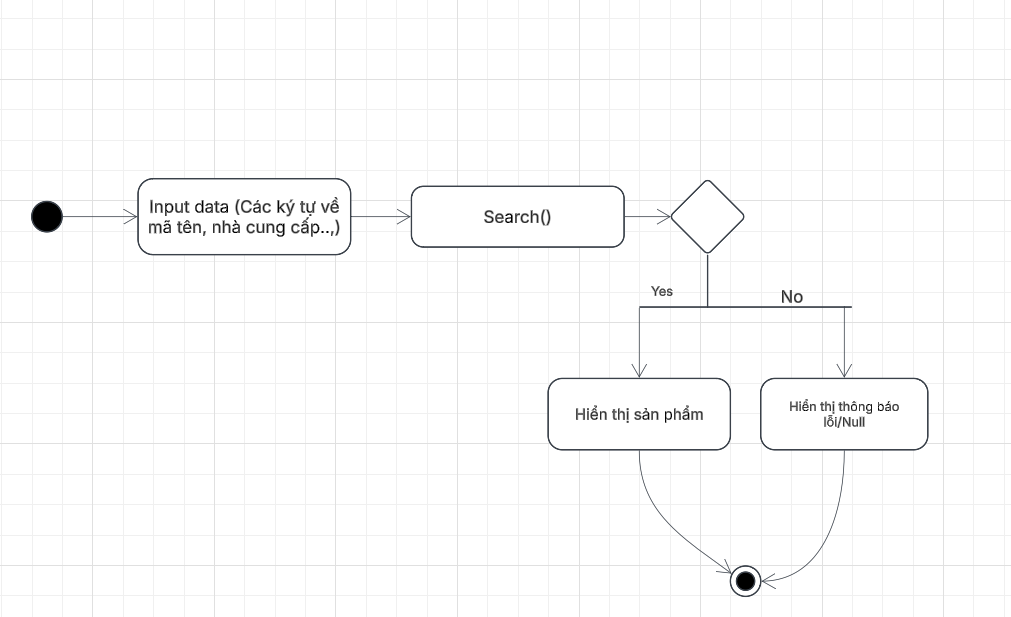
* Input: Mã sản phẩm cần xóa.
* Output: Xóa sản phẩm trong cơ sở dữ liệu.
* Quy trình xử lý:
  + Người dùng tìm kiếm sản phẩm theo mã sản phẩm cần xóa.
  + Hệ thống tìm kiếm và hiển thị thông tin sản phẩm cần xóa và hiển thị hộp thoại xác nhận. Nếu không có mã sản phẩm cần xóa hệ thống sẽ đưa thông báo.
  + Người dùng nhấn nút xóa.
  + Hệ thống sẽ kiểm tra các thông tin sau:
    - Kiểm tra quyền của người dùng.
    - Nếu có quyền, hệ thống thống thực hiện đánh dấu sản phẩm đã bị xóa trong database
    - Nếu không có quyền, hệ thống sẽ bỏ qua lệnh này.
  + Xóa thành công sẽ hiển thị thông báo.
  + Xóa thất bại sẽ hiển thị thông báo kèm theo lỗi.
* Biểu đồ tuần tự:



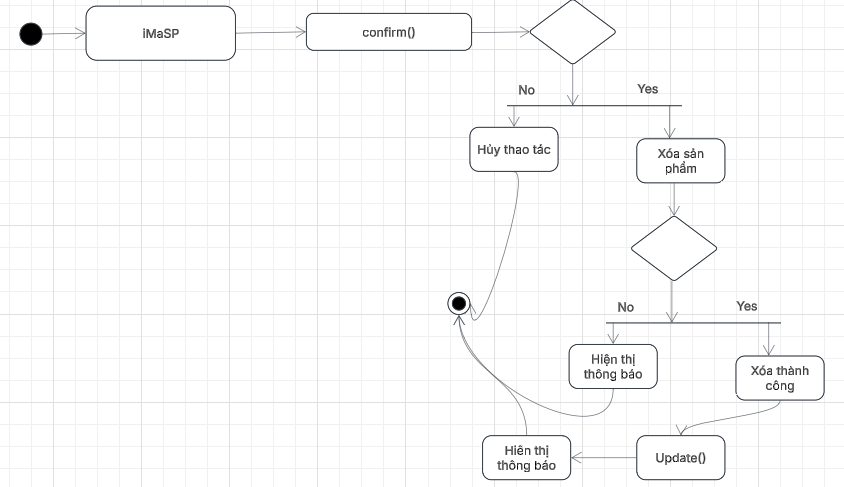
* Các unit được sử dụng trong quá trình “xóa sản phẩm”

| STT | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | ProductController | search() | Mã sản phẩm | Thông tin sản phẩm hoặc không có sản phẩm |
| 2 | ProductController | confirm() | Mã sản phẩm | Xóa thành công hay thất bại |
| 3 | ProductController | update() | Thông tin sản phẩm | Xác nhận xóa sản phẩm thành công hoặc thất bại kèm lỗi |

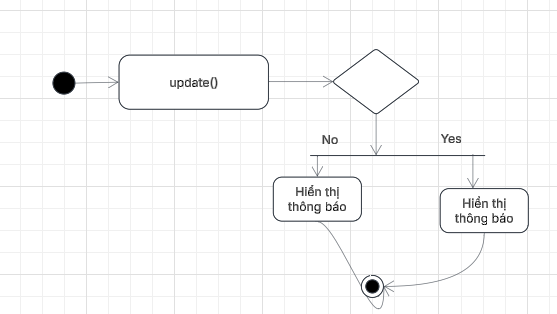
* Từng unit được thiết kế như sau:
  + search():



* + confirm():

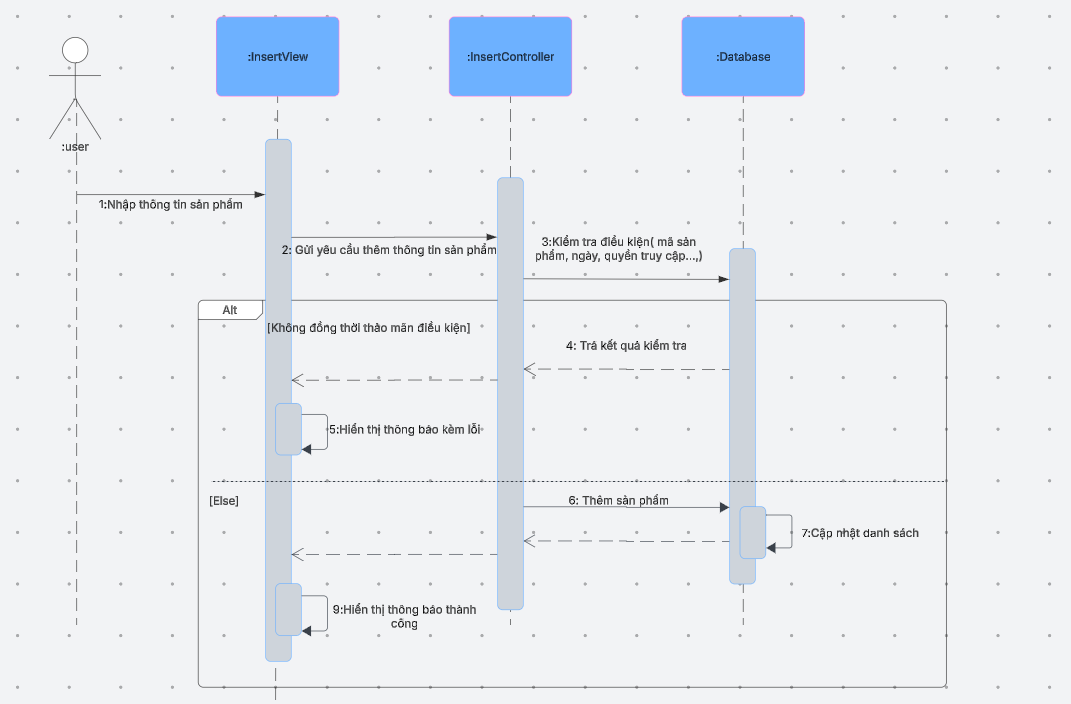


* + update():



#### Thiết kế xử lý chức năng thêm sản phẩm

* Input: Thông tin sản phẩm cần thêm.
* Output: Thêm sản phẩm thành công, sản phẩm nằm trong Database.
* Process:
  + Người dùng có thẩm quyền nhập các thông tin về sản phẩm cần được thêm gồm: Tên thuốc, mã thuốc, nhà sản xuất, giá nhập, giá bán, số lượng tồn kho, ngày sản xuất, hạn sử dụng, hình ảnh sản phẩm (tùy chọn), thông tin khác( nếu có).
  + Người dùng nhấn nút “Thêm” để xác nhận việc thêm sản phẩm mới.
  + Hệ thống kiểm tra thông tin của sản phẩm:
    - Kiểm tra trùng lặp của mã thuốc, nếu mã thuốc đã tồn tại trong Database thì đưa ra thông báo “Sản phẩm đã tồn tại”.
    - Những ràng buộc của các trường thông tin:
      * Nếu những thông tin thiết yếu, bắt buộc cần nhập bị bỏ trống thì sẽ cảnh bảo người dùng “Không thể để trống thông tin này” vào các trường bắt buộc.
      * Ngày sản xuất nhỏ hơn thời điểm hiện tại.
      * Giá cả phải phải lớn hơn 0.
  + Nếu thông tin sản phẩm hợp lệ, hệ thống thêm sản phẩm mới vào Database.
* Biểu đồ tuần tự:

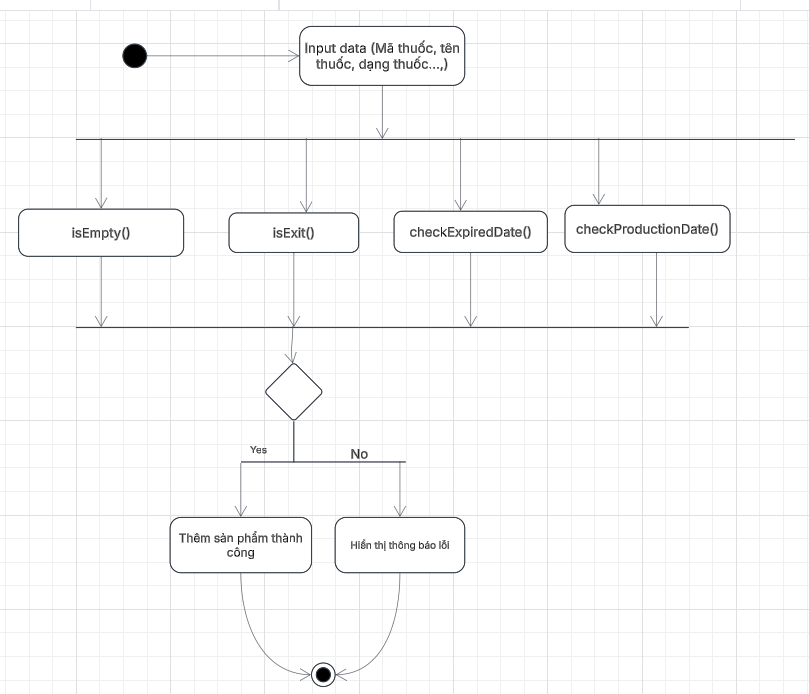


* Các unit cần có cho chức năng “ thêm sản phẩm’’

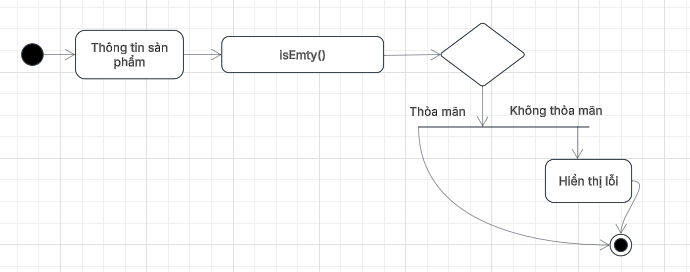
| **STT** | **Class** | **Method** | **Input** | **Output** |
| --- | --- | --- | --- | --- |
| 1 | Medicine | addMedicine() | Mã thuốc, Tên thuốc, Dạng thuốc, Đơn vị tính, Số lượng, Giá nhập, Giá bán, Cách sử dụng, Điều kiện bảo quản, Nhà cung cấp, Ngày sản xuất, Hạn sử dụng, Thuốc biệt trữ, Thuốc thuộc diện thu hồi, Kê đơn | Thêm sản phẩm thành công |
| Thêm sản phẩm thất bại |
| 2 | Medicine | isEmpty() | Mã thuốc, Tên thuốc, Dạng thuốc, Đơn vị tính, Số lượng, Giá nhập, Giá bán, Cách sử dụng, Điều kiện bảo quản, Nhà cung cấp, Ngày sản xuất, Hạn sử dụng, Thuốc biệt trữ, Thuốc thuộc diện thu hồi, Kê đơn | Bạn không được bỏ trống trường này |
| 3 | Medicine | IsExist() | Mã thuốc | Đã tồn tại thuốc có mã tương tự |
| 4 | Medicine | checkProductionDate() | Ngày sản xuất | Ngày sản xuất phải nhỏ hơn hiện tại |
| 5 | Medicine | CheckExpiredDate() | Hạn sử dụng | Ngày hết hạn phải lớn hơn hiện tại |

* Thiết kế từng unit:

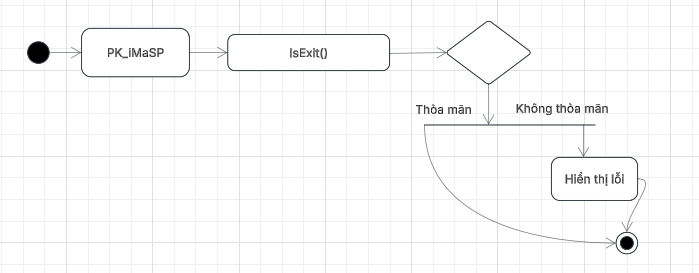
1. addMedicine()



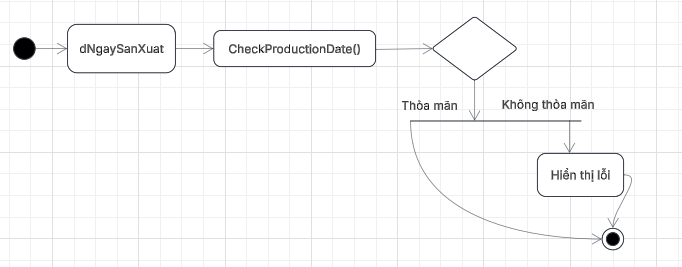
1. isEmpty()



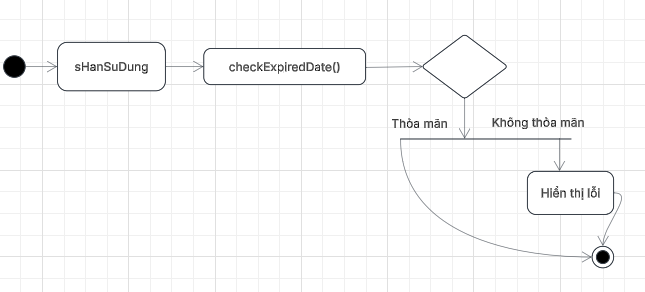
1. isExist()



1. checkProductionDate()

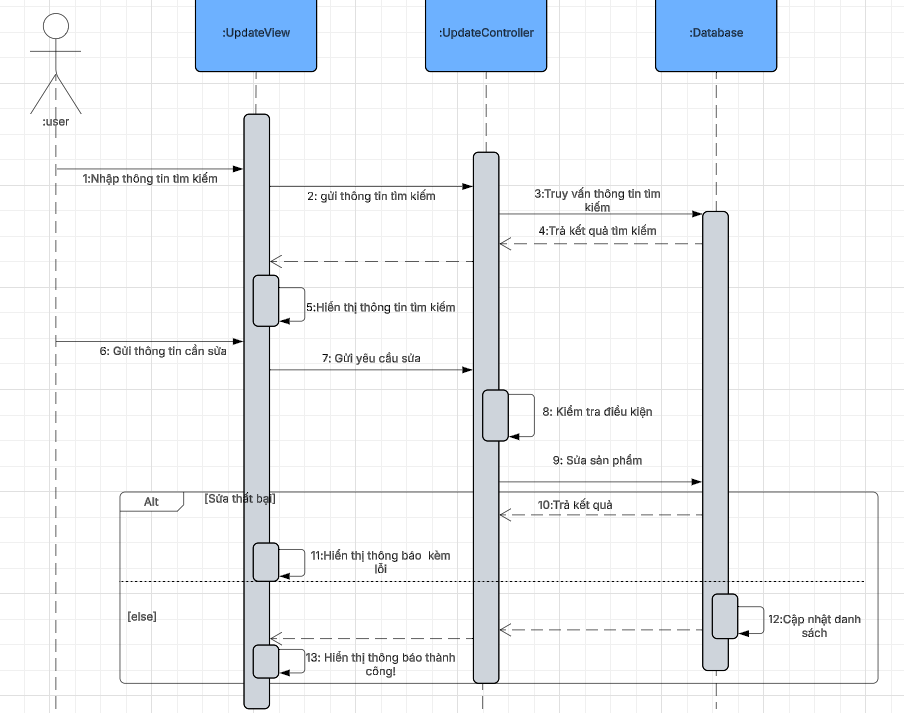


1. checkExpiredDate()



#### Thiết kế xử lý chức năng sửa sản phẩm

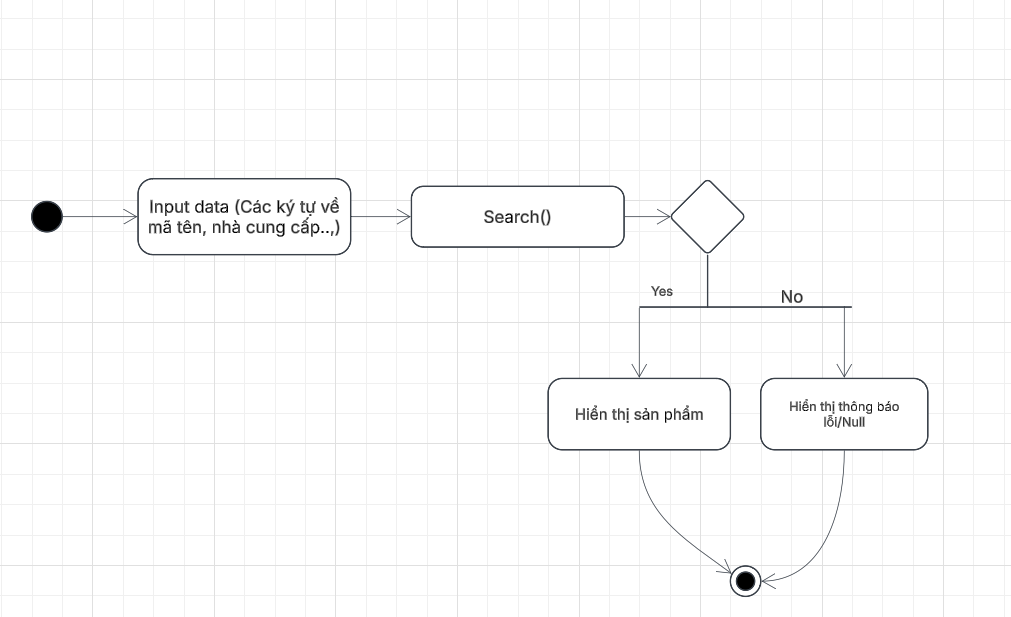
* Input: Mã sản phẩm, các thông tin sản phẩm cần sửa.
* Output: Cập nhật thông tin sản phẩm trong cơ sở dữ liệu.
* Quy trình xử lý:
  + Tìm kiếm sản phẩm theo mã hoặc tên.
  + Hiển thị thông tin sản phẩm.
  + Nhập các trường thông tin cần cập nhật.
  + Kiểm tra định dạng, độ dài, tính hợp lệ của dữ liệu nhập.
  + Gửi yêu cầu cập nhật đến lớp dữ liệu.
  + Thông báo thành công nếu cập nhật thành công.
  + Thông báo lỗi nếu cập nhật thất bại (kèm theo thông tin lỗi).
* Biểu đồ tuần tự:



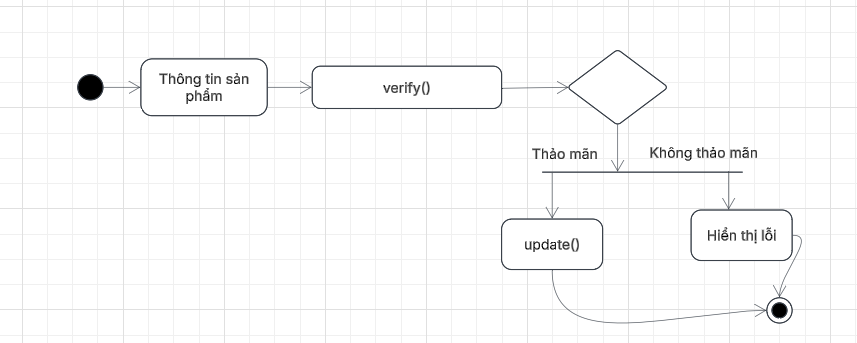
* Các unit được sử dụng trong quá trình “sửa sản phẩm”

| STT | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | ProductController | search() | Mã sản phẩm  Hoặc  Tên sản phẩm | Kết quả tìm kiếm (nếu rỗng thì thông báo) |
| 2 | ProductController | verify() | Thông tin cần cập nhật | Kết quả sau khi kiểm tra tính hợp lý của thông tin cần cập nhật |
| 3 | ProductController | update() | Thông tin cần cập nhật | Kết quả sau khi cập nhật thông tin lên cơ sở dữ liệu |

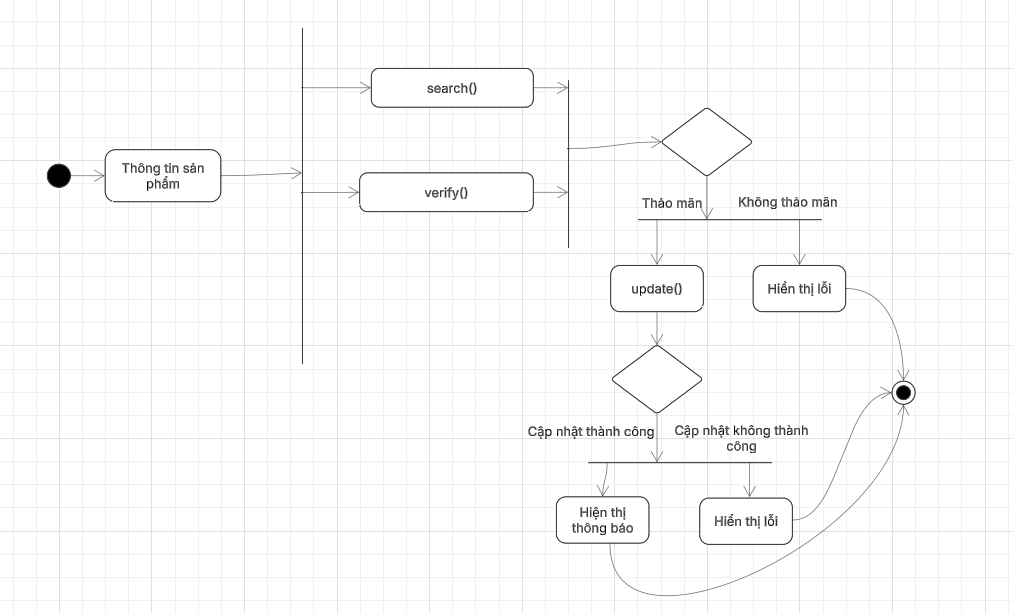
* Từng unit được thiết kế như sau :
  + search():



* + verify():



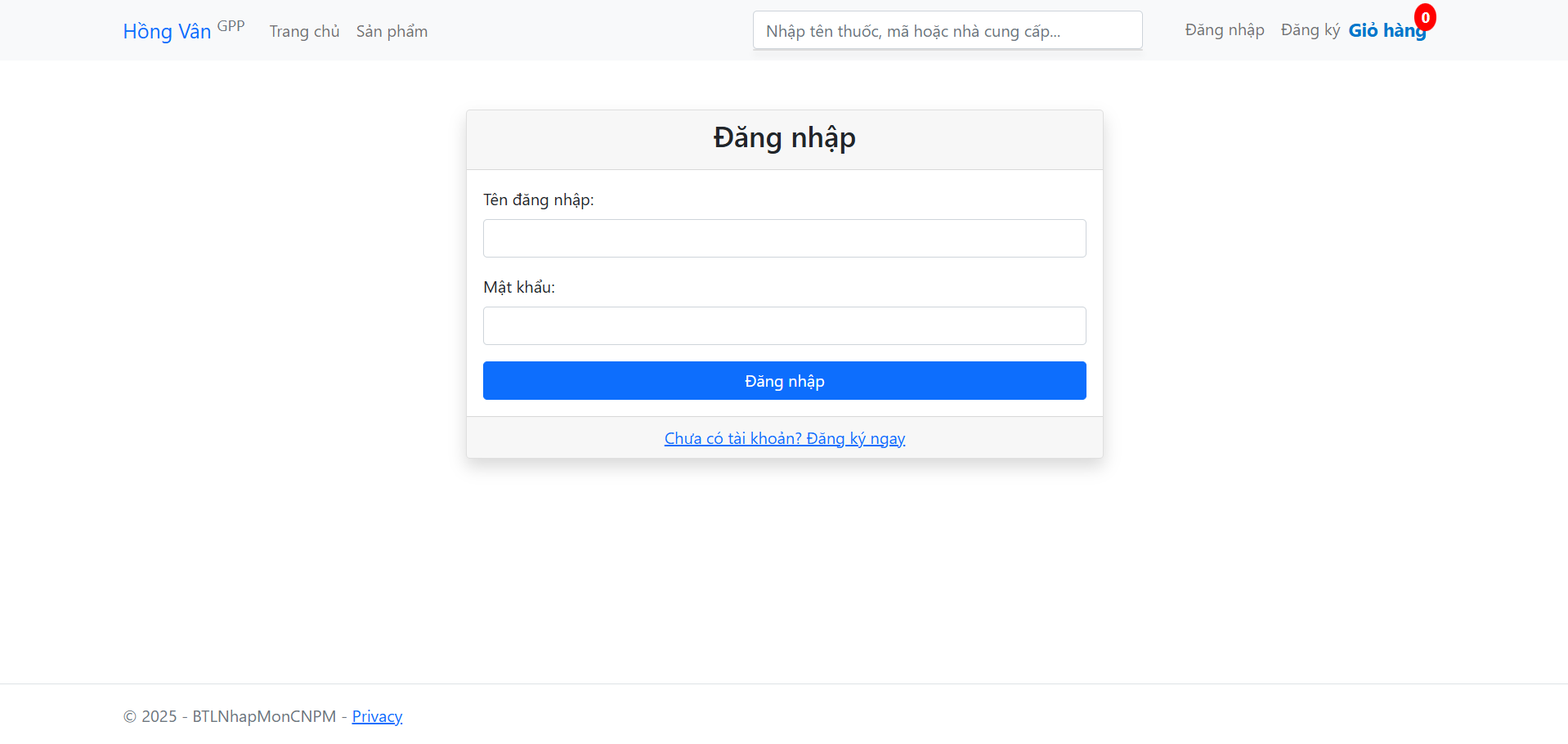
* + update():

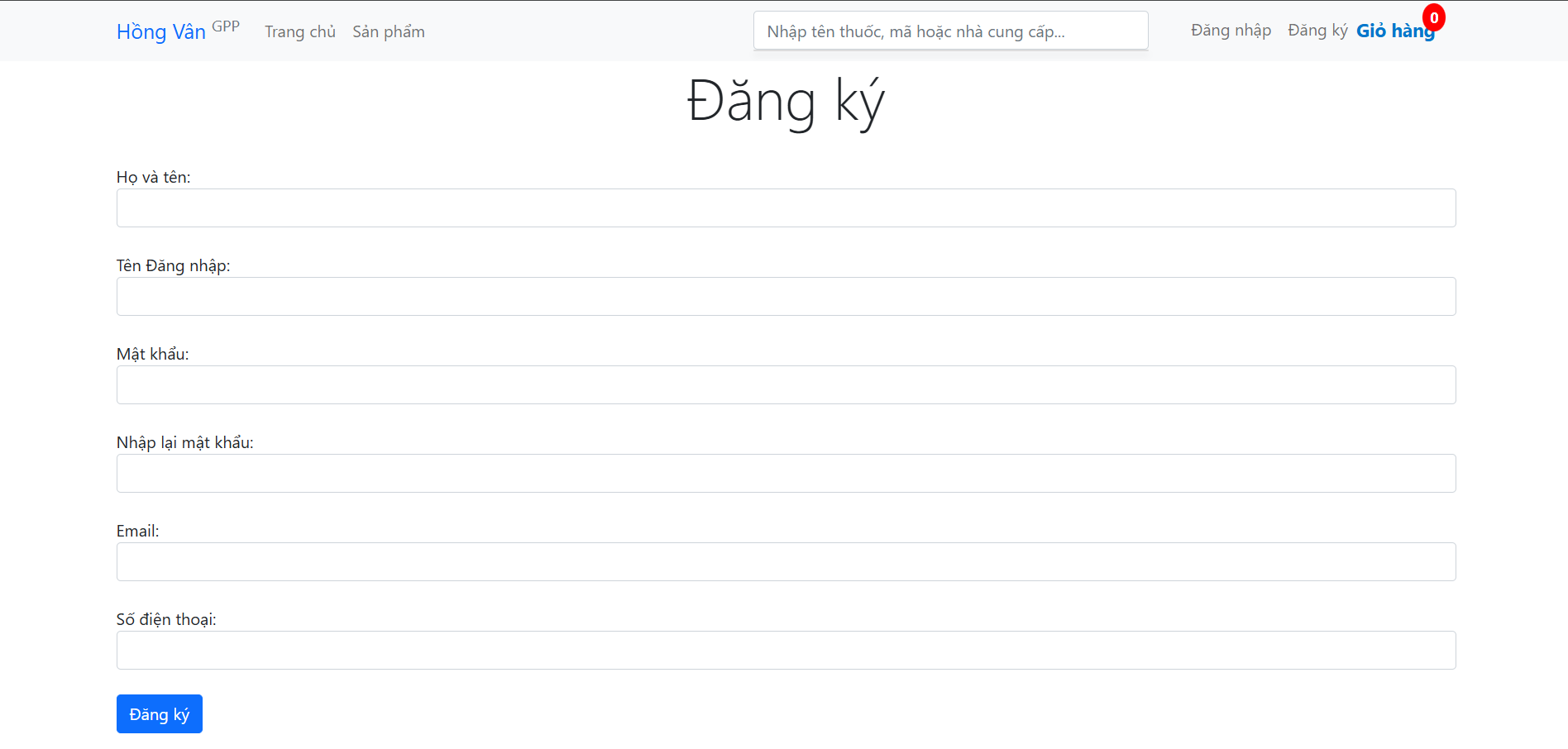


## CHƯƠNG 5. XÂY DỰNG VÀ KIỂM THỬ HỆ THỐNG

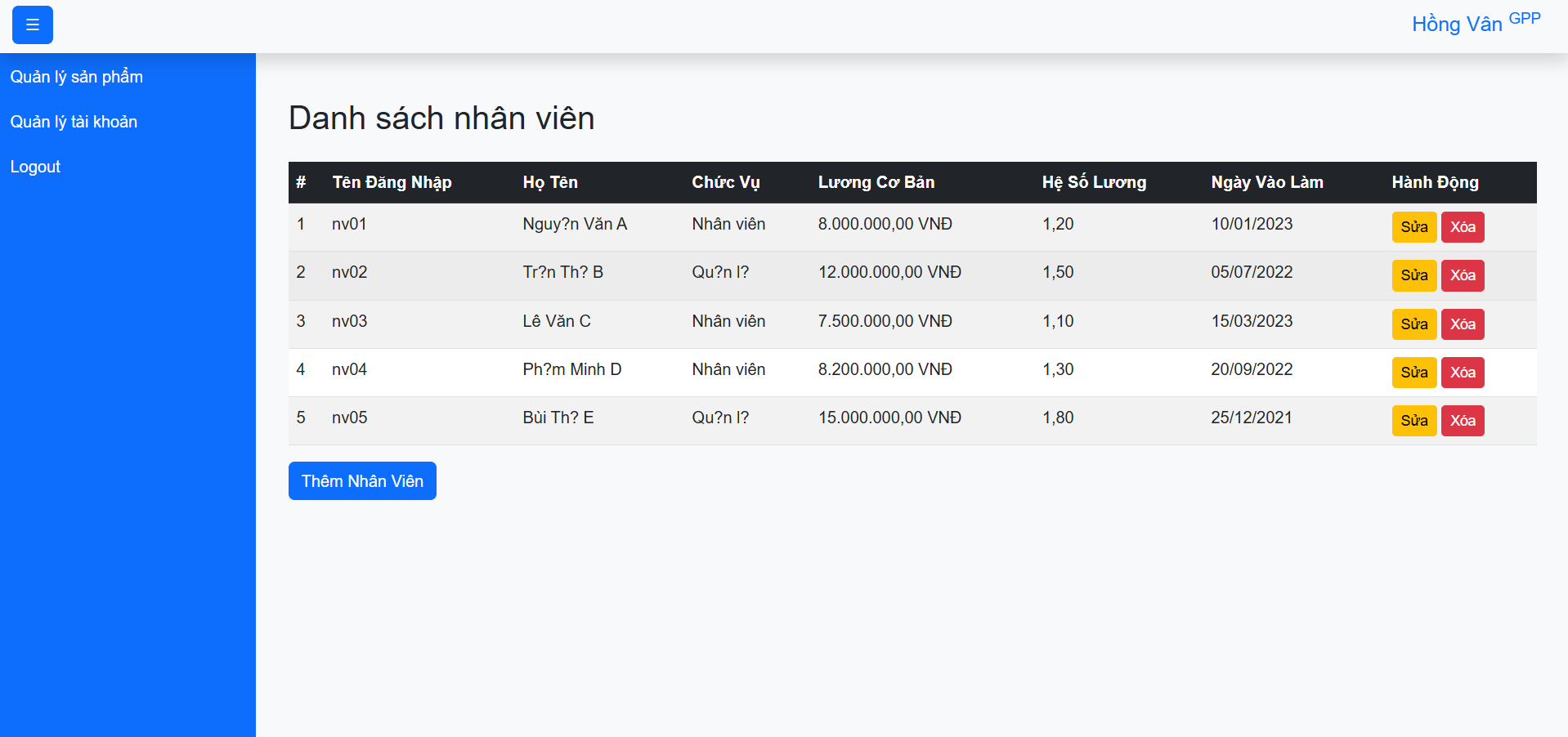
*Lựa chọn xây dựng và kiểm thử các chức năng đáp ứng cho ít nhất 20% yêu cầu quan trọng nhất của hệ thống và phải chỉ rõ Ai-làm gì tương ứng với chương này trong bảng phân công (mục 1.7)*

### Kết quả xây dựng module/nhóm chức năng

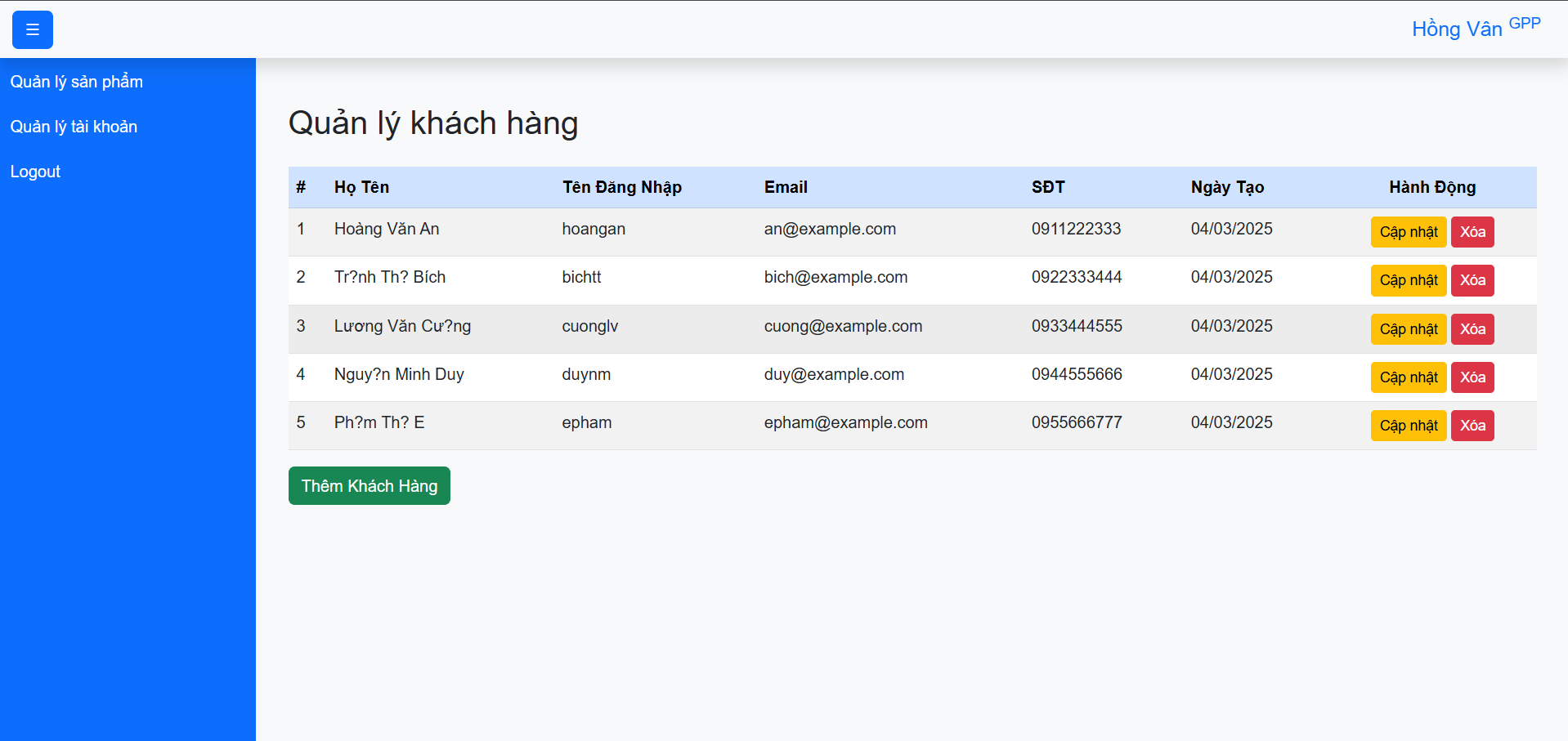
* + 1. Chức năng đăng nhập
    2. Chức năng đăng ký



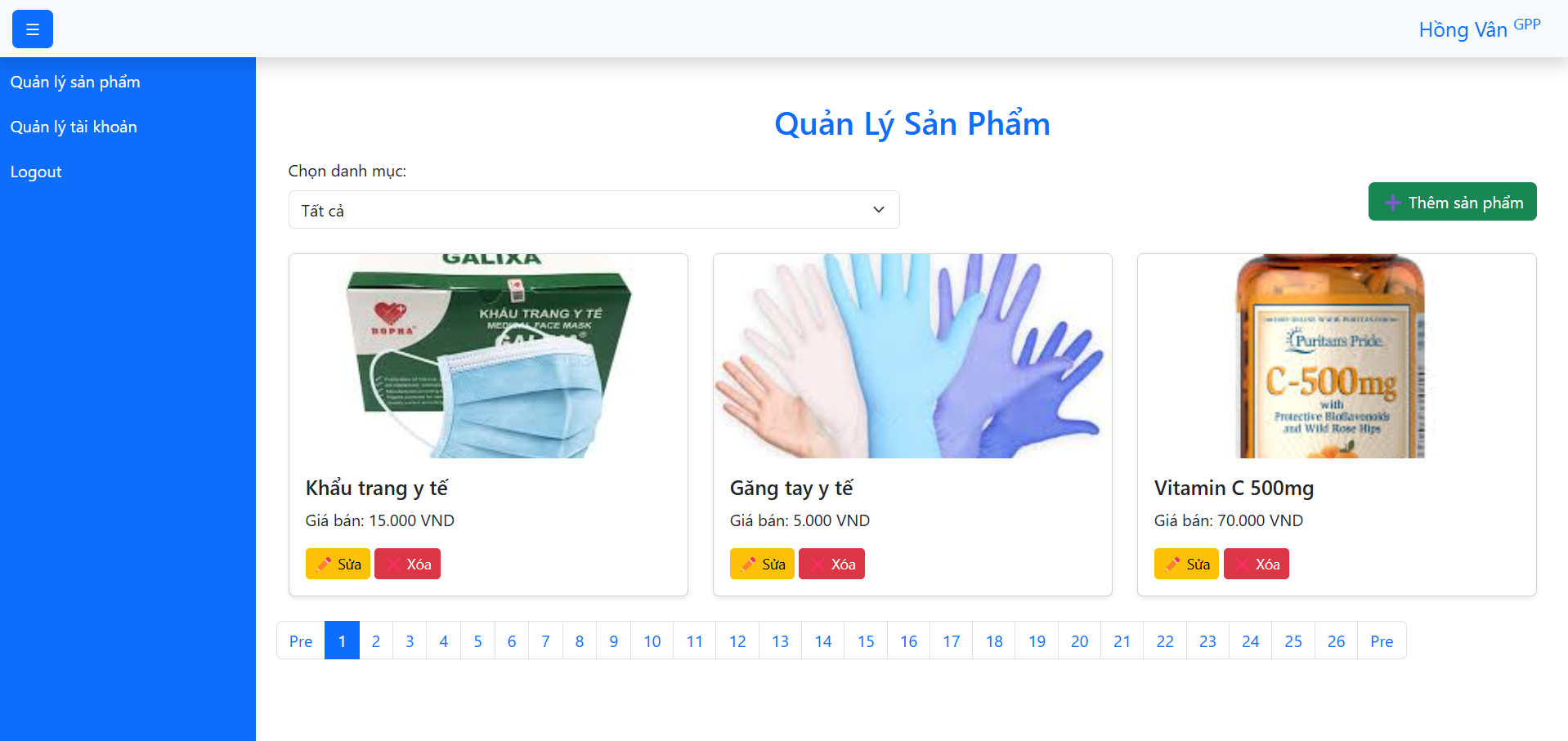
* + 1. Chức năng Danh sách tài khoản
* Quản lý tài khoản nhân viên



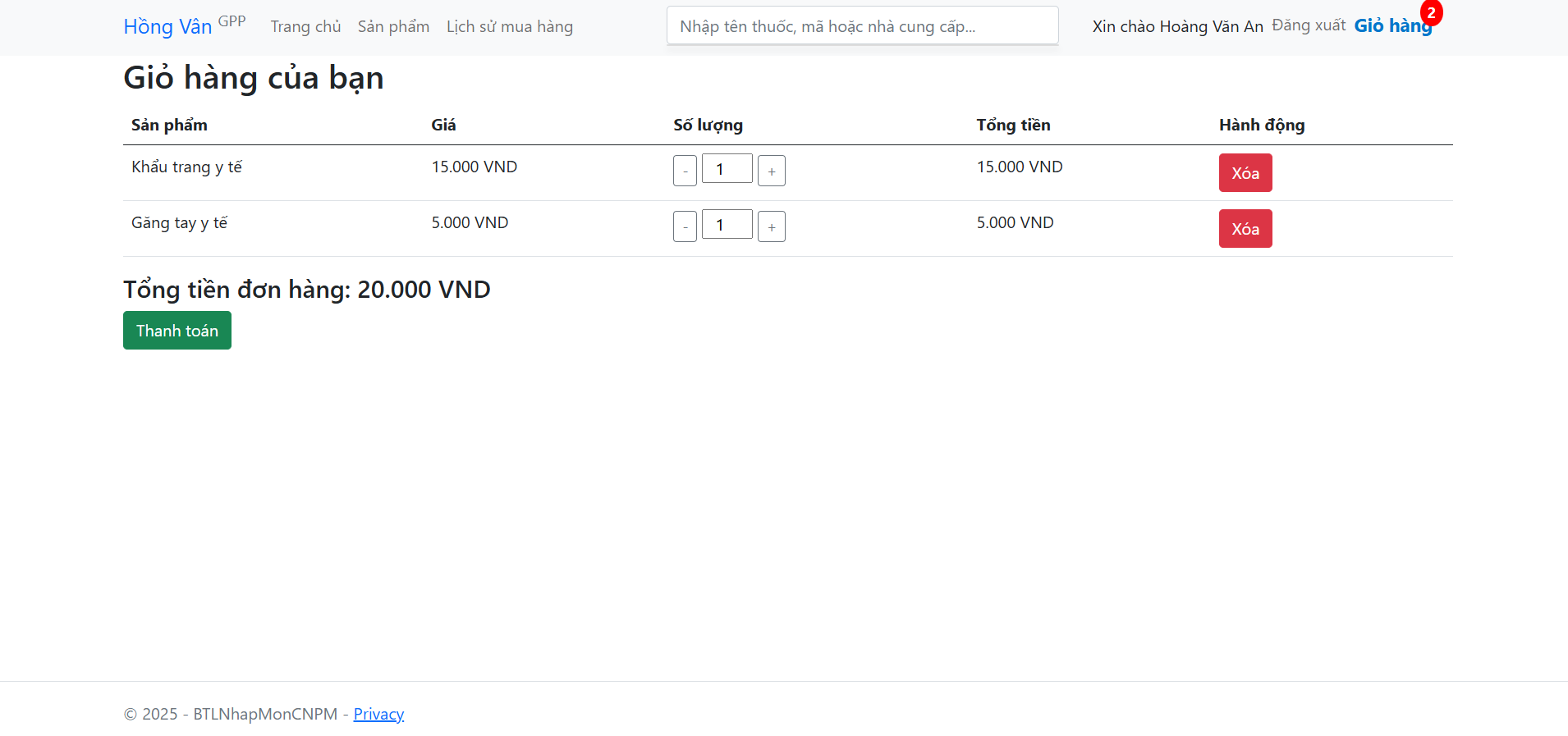
* Quản lý tài khoản khách hàng



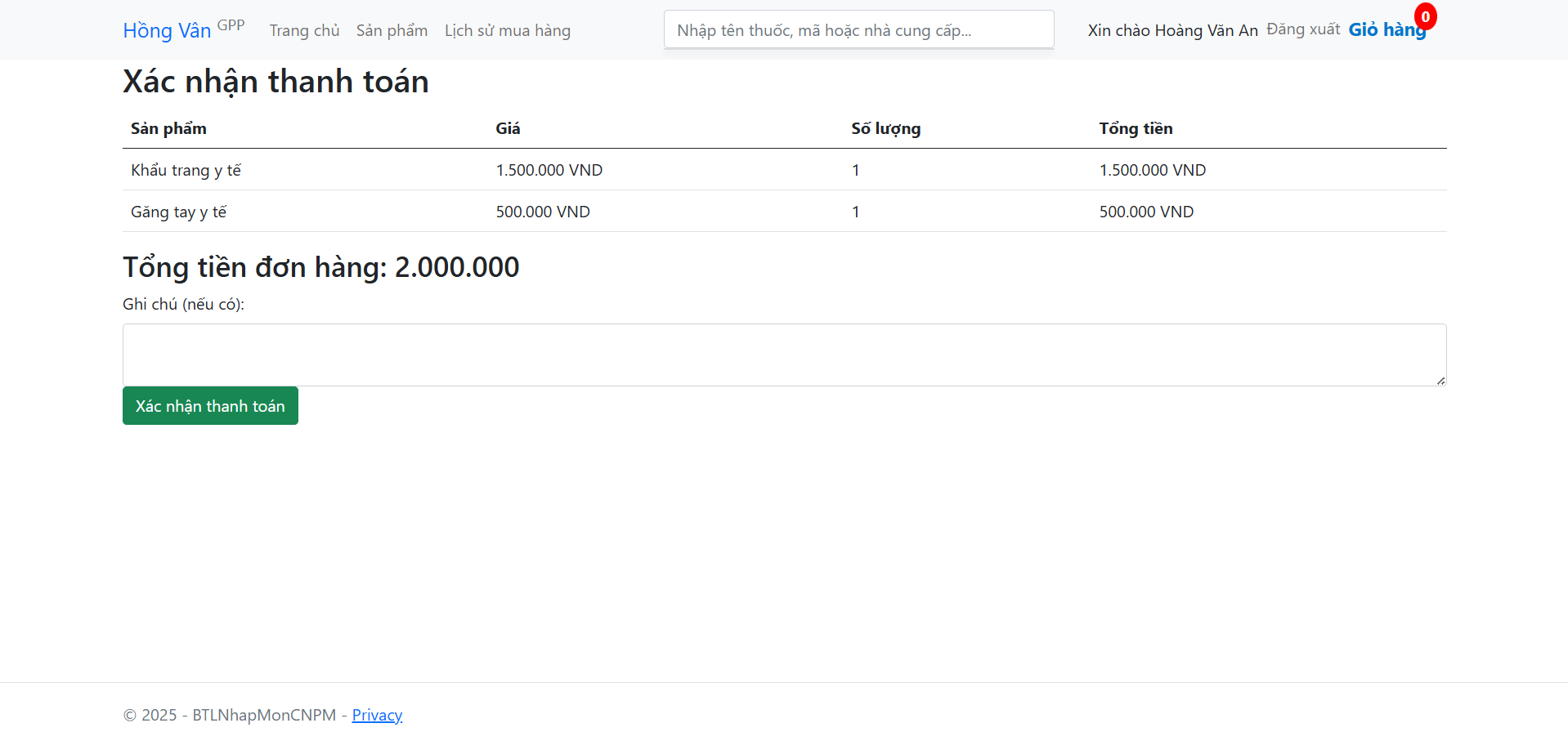
* + 1. Chức năng Quản lý sản phẩm



5.1.5 Quản lý giỏ hàng



5.1.6 Thanh toán đơn hàng



### Kiểm thử hệ thống

#### Tổng quan về kiểm thử phần mềm

* + - 1. Khái niệm kiểm thử phần mềm:

Theo IEEE “Kiểm thử là tiến trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận xét kết quả và đưa ra đánh giá về hệ thống hoặc thành phần đó”.

* + - 1. Các nguyên tắc kiểm thử

**Nguyên tắc 1: Kiểm thử luôn có lỗi -** Kiểm thử có thể cho thấy rằng phần mềm đang có lỗi, nhưng không thể chứng minh rằng một phần mềm không có lỗi. Kiểm thử làm giảm xác suất lỗi tiềm ẩn trong phần mềm, ngay cả khi đã kiểm thử nghiêm ngặt phần mềm vẫn có thể còn lỗi. Vì vậy, cần phải tìm được càng nhiều lỗi càng tốt.

**Nguyên tắc 2: Kiểm thử toàn bộ là không thể -** Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ khi kiểm thử chỉ bao gồm một số ít trường hợp thì có thể kiểm thử toàn bộ. Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên người kiểm thử có thể tập trung việc kiểm thử vào một số điểm cần thiết, có nguy cơ lỗi cao hơn. Nghĩa là phải lên kế hoạch kiểm thử, thiết kế trường hợp kiểm thử sao cho có độ bao phủ nhiều nhất và giảm thiểu rủi ro sót lỗi khi đến tay người dùng.

**Nguyên tắc 3: Kiểm thử càng sớm càng tốt -** Để tìm được lỗi sớm, các hoạt động kiểm thử nên được bắt đầu càng sớm càng tốt trong quy trình phát triển (vòng đời phát triển) phần mềm hoặc hệ thống, và nên tập trung vào các hoạt động/mục tiêu đã xác định trước.

Các hoạt động kiểm thử được bắt đầu càng sớm thì sẽ phát hiện ra lỗi sớm khi đó ít tốn công để tìm lỗi và sửa chữa.

**Nguyên tắc 4: Sự tập trung của lỗi -** Thông thường, lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Nếu xác định được điều này bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.

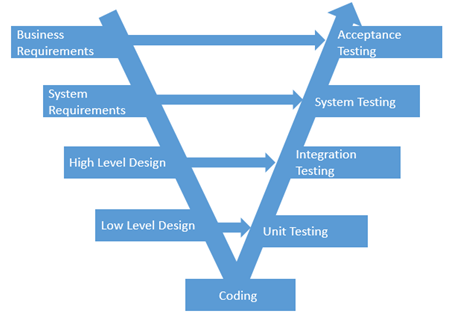
**Nguyên tắc 5: Nghịch lý thuốc trừ sâu -** Nếu bạn sử dụng cùng một tập hợp các trường hợp kiểm thử liên tục, sau đó một thời gian các trường hợp kiểm thử không tìm thấy lỗi nào mới. Hiệu quả của các trường hợp kiểm thử bắt đầu giảm xuống sau một số lần thực hiện, vì vậy luôn luôn chúng ta phải luôn xem xét và sửa đổi các trường hợp kiểm thử trên một khoảng thời gian thường xuyên.

**Nguyên tắc 6: Kiểm thử phụ thuộc vào ngữ cảnh -** Theo nguyên tắc này thì việc kiểm thử phụ thuộc vào ngữ cảnh và chúng ta phải tiếp cận kiểm thử theo nhiều ngữ cảnh khác nhau Nếu bạn đang kiểm thử ứng dụng web và ứng dụng di động bằng cách sử dụng chiến lược kiểm thử giống nhau, thì đó là sai. Chiến lược để kiểm thử ứng dụng web sẽ khác với kiểm thử ứng dụng cho thiết bị di động của Android.

**Nguyên tắc 7: Không có lỗi - sai lầm:** Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng để tung ra thị trường. Việc không tìm thấy lỗi cũng có thể là do bộ trường hợp kiểm thử được tạo ra chỉ nhằm kiểm tra những tính năng được làm đúng theo yêu cầu thay vì nhằm tìm kiếm lỗi mới

#### Mô hình kiểm thử được áp dụng trong đề tài

Hệ thống được áp dụng Test theo mô hình chữ V [4]. (I.E. Dorothy Graham, Foundations of Software Testing, 2007)



Các hoạt động phát triển phần mềm của Dev:

● Phân tích yêu cầu: Thu thập, phân tích và nghiên cứu hệ thống có những chức năng gì.

● Yêu cầu hệ thống: High level design của phần mềm được xây dựng. Hoạt động đánh giá thiết kế, xác minh tạo test plan và test case .

● Thiết kế kiến trúc: Dựa trên thiết kế mức cao, kiến trúc phần mềm được xây dựng. Các module , sơ đồ kiến trúc, bảng cơ sở dữ liệu, chi tiết về công nghệ được hoàn thành.

● Thiết kế module: Các module hoặc thành phần được thiết kế riêng, Các method, class, giao diện, kiểu dữ liệu … được hoàn tất.

● Code: Thực hiện viết code từ các hoạt động bên Dev, bên Test áp dụng kiểm thử theo các giai đoạn trong mô hình chữ V

○ Unit test: Kiểm tra các đoạn code trong dự án để đảm bảo các module hoạt động chính xác.

○ Integration test: Kiểm thử các “module” hay các “component” khi chúng tích hợp với nhau.

○ System test: Kiểm thử cả một hệ thống hoàn chỉnh khi toàn bộ các “module” hay “component” được tích hợp.

○ Acceptance test: Kiểm thử hệ thống hoàn chỉnh như được cài đặt cho người dùng cuối.

#### Kế hoạch kiểm thử

* + - 1. Giới Thiệu

Theo IEEE “Kiểm thử là tiến trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận xét kết quả và đưa ra đánh giá về hệ thống hoặc thành phần đó”.

* + - * 1. Mục Đích

Tài liệu kế hoạch kiểm thử này được lập ra nhằm xác định thông tin cơ bản về dự án cũng như các thành phần chức năng cần kiểm thử. Đồng thời, tài liệu này giúp xác định những yêu cầu về tài nguyên cần thiết cho quá trình kiểm thử, đảm bảo việc chuẩn bị đầy đủ các điều kiện để thực hiện kiểm thử hiệu quả. Ngoài ra, tài liệu cũng liệt kê các yêu cầu kiểm thử cụ thể (Test Requirements), giúp định hướng rõ ràng phạm vi và mục tiêu kiểm thử. Bên cạnh đó, các chiến lược kiểm thử phù hợp sẽ được đề xuất nhằm đảm bảo chất lượng và hiệu suất của hệ thống. Cuối cùng, tài liệu này cũng sẽ tổng hợp danh sách các tài liệu cần được lập sau khi hoàn thành kiểm thử, phục vụ việc đánh giá và lưu trữ kết quả kiểm thử một cách có hệ thống.

* + - * 1. Chú giải viết tắt

| Từ viết tắt | Mô tả | Ghi chú |
| --- | --- | --- |
| UI | User Interface |  |
| SRS | Software Requirement Specification |  |
| UAT | User Acceptance test |  |

* + - * 1. Tài liệu đầu vào

| **Tên tài liệu** | **Ngày thực hiện** |
| --- | --- |
| Tài liệu SRS | 23/02/2025 |
| Hồ sơ thiết kế | 10/03/2025 |

* + - * 1. Phạm vi

Tài liệu kiểm thử này được áp dụng cho việc kiểm thử những chức năng của phần mềm Quản lý hoạt động cho nhà thuốc Hồng Vân được đặc tả trong tài liệu SRS

Các chức năng thực hiện test:

* **Quản lý sản phẩm:** 
  + Chức năng tìm kiếm sản phẩm
  + Chức năng thêm sản phẩm
  + Chức năng xóa sản phẩm
  + Chức năng sửa sản phẩm
* **Quản lý đặt hàng:** 
  + Chức năng tìm kiếm sản phẩm.
  + Chức năng thêm sản phẩm, điều chỉnh giỏ hàng.
  + Lịch sử mua hàng.
    - * 1. Danh sách rủi ro

| **Risk** | **Giải pháp giảm tránh Risk** |
| --- | --- |
| Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm thử app | Lập kế hoạch khóa training để nâng cao kỹ năng của các thành viên |
| Thiếu hợp tác, ảnh hưởng tiêu cực đến năng suất của thành viên | Khuyến khích mỗi thành viên trong nhóm thực hiện nhiệm vụ của mình và truyền cảm hứng cho họ để họ nỗ lực nhiều hơn. |
| Các thành viên có quỹ thời gian hạn hẹp để tham gia dự án | Nắm bắt được thông tin về thời gian của các thành viên sắp xếp công việc phù hợp đối với quỹ thời gian của từng người |

* + - 1. Requirements for Test

1. Xác định yêu cầu
   1. Yêu cầu đầu vào

| **STT** | **Mức độ kiểm thử** | **Yêu cầu đầu vào** |
| --- | --- | --- |
| 1 | Kiểm thử đơn vị | - Tài liệu SRS  - Hồ sơ thiết kế hệ thống  - Mã nguồn: gồm chức năng đăng nhập, đổi mật khẩu, quản lý sản phẩm  Testcase  - Môi trường kiểm thử |
| 2 | Kiểm thử tích hợp | - Module đã được kiểm thử đơn lẻ  - Kế hoạch kiểm thử  - Tài liệu thiết kế  - Đặc tả chức năng  - Các test case, test data  - Môi trường kiểm thử |
| 3 | Kiểm thử hệ thống | - Hệ thống phần mềm đã hoàn thành  - Đặc tả yêu cầu  - Bản thiết kế hệ thống  - Testcase và test data  - Các tiêu chuẩn kiểm thử  - Môi trường kiểm thử |
| 4 | Kiểm thử chấp nhận | - Hệ thống phần mềm đã hoàn thành  - Đặc tả yêu cầu chức năng và phi chức năng  - Kế hoạch kiểm thử chấp nhận  - Testcase và test data  - Các tiêu chuẩn kiểm thử  - Môi trường kiểm thử |

* 1. Yêu cầu đầu ra

| STT | Mức độ kiểm thử | Yêu cầu đầu ra |
| --- | --- | --- |
| 1 | Kiểm thử đơn vị | Kết quả kiểm thử  Lỗi nghiêm trọng được sửa và đóng.  Tính tương thích  Báo cáo kiểm thử |
| 2 | Kiểm thử tích hợp | Kết quả kiểm thử  Tất cả các bug nghiêm trọng hoặc ưu tiên đã được sửa và đóng.  Integration Testing đã hoàn thành tất cả các yêu cầu chức năng và hiệu suất.  Tính tương thích  Khả năng cài đặt đã được xác minh.  Các bản ghi và báo cáo về kết quả kiểm thử |
| 3 | Kiểm thử hệ thống | Tất cả test case System Testing đã được thực hiện.  Lỗi ưu tiên đã được sửa và đóng.  Hệ thống đang đáp ứng tất cả các yêu cầu nghiệp vụ và chức năng.  Hệ thống đang hỗ trợ tất cả các phần cứng và phần mềm dự định của nó.  Báo cáo kết quả kiểm thử |
| 4 | Kiểm thử chấp nhận | Hoàn thành và phê duyệt bởi Test Management team.  Quy trình nghiệp vụ thiết yếu đã được bao phủ.  Không có critical defect.  Báo cáo kết quả kiểm thử |

* Tiêu chí kiểm tra chấp nhận:
  + Tiêu chí đình chỉ kiểm thử (Suspension Criteria): Nếu có 40% trường hợp kiểm thử thất bại thì tạm dừng kiểm thử cho đến khi nhóm phát triển sửa chữa tất cả các trường hợp thất bại.
  + Tiêu chí đầu vào (Entry Criteria)
    - (Các) môi trường thử nghiệm có sẵn
    - Dữ liệu thử nghiệm có sẵn
    - Code đã được hợp nhất thành công
    - Đã hoàn thành Unit Test
  + Tiêu chí đầu ra
    - > 98% Test case được thực thi (Run rate)
    - > 98% pass rate of Test Case
    - Không có lỗi nghiêm trọng và mức độ nghiêm trọng cao
      1. Lịch trình công việc

| **STT** | **Nội dung** | **Sản phẩm** |
| --- | --- | --- |
| 1 | Lập kế hoạch kiểm thử | Tài liệu “Kế hoạch kiểm thử” |
| 2 | Thiết kế Test Case | Tài liệu Test Case |
| 3 | Đánh giá các Test Case | Tài liệu Test Case |
| 4 | Thực thi các Test Case | Tài liệu Test Case |
| 5 | Ghi nhận và đánh giá kết quả kiểm thử | Tài liệu Test Case |

* + - 1. Chiến lược kiểm thử (Test Strategy)

| Type of Tests | Stage of Test | | | |
| --- | --- | --- | --- | --- |
| Unit | Integration | System | Acceptance |
| <Function Test > |  | X | X | X |
| <User Interface test> |  | X | X | X |
| <Regression Test> |  | X |  |  |
| <Usability Test> |  |  | X | X |
| <Statement Test> | X |  |  |  |

● **Test types**

**Function Testing:** *Kiểm tra chức năng là để xác minh việc chấp nhận, xử lý và truy xuất dữ liệu phù hợp cũng như việc triển khai phù hợp các quy tắc kinh doanh. Loại thử nghiệm này dựa trên kỹ thuật hộp đen; đó là xác minh ứng dụng và các quy trình nội bộ của nó bằng cách tương tác với ứng dụng thông qua Giao diện người dùng đồ họa (GUI) và phân tích đầu ra hoặc kết quả. Được xác định bên dưới là phác thảo về thử nghiệm được đề xuất cho từng ứng dụng:*

| Test Objective: | Đảm bảo chức năng mục tiêu thử nghiệm phù hợp, bao gồm điều hướng, nhập dữ liệu, xử lý và truy xuất. |
| --- | --- |
| Technique: | Thực thi từng trường hợp sử dụng, luồng trường hợp sử dụng hoặc chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác minh những điều sau:   * Kết quả mong đợi xảy ra khi dữ liệu hợp lệ được sử dụng. * Các thông báo lỗi hoặc cảnh báo thích hợp được hiển thị khi dữ liệu không hợp lệ được sử dụng. * Sử dụng công cụ test |
| Completion Criteria: | * Tất cả các thử nghiệm theo kế hoạch đã được thực hiện. * Tất cả các lỗi được xác định đã được giải quyết và đóng lại |
| Special Considerations: | Xác định hoặc mô tả các hạng mục hoặc vấn đề ảnh hưởng đến việc triển khai và thực hiện kiểm thử chức năng |

User Interface Testing:*Kiểm thử giao diện người dùng là đảm bảo rằng Giao diện người dùng cung cấp cho người dùng quyền truy cập và điều hướng phù hợp thông qua các chức năng của mục tiêu thử nghiệm. Ngoài ra, kiểm thử giao diện người dùng đảm bảo rằng các đối tượng trong giao diện người dùng hoạt động như mong đợi và tuân thủ các tiêu chuẩn của công ty hoặc ngành.*

| Test Objective: | Xác minh những điều sau đây:   * Điều hướng qua mục tiêu thử nghiệm phản ánh đúng chức năng và yêu cầu kinh doanh, bao gồm từ cửa sổ này sang cửa sổ khác, trường này sang trường khác và sử dụng các phương thức truy cập (phím tab, di chuyển chuột, phím tăng tốc) * Các đối tượng và đặc điểm của cửa sổ, chẳng hạn như menu, kích thước, vị trí, trạng thái và tiêu điểm tuân theo các tiêu chuẩn. |
| --- | --- |
| Technique: | Tạo hoặc sửa đổi các kiểm tra cho từng cửa sổ để xác minh trạng thái điều hướng và đối tượng thích hợp cho từng cửa sổ ứng dụng và đối tượng. |
| Completion Criteria: | Mỗi cửa sổ được xác minh thành công để phù hợp với phiên bản điểm chuẩn hoặc trong tiêu chuẩn chấp nhận được |
| Special Considerations: | Không phải tất cả các thuộc tính cho các đối tượng tùy chỉnh và bên thứ ba đều có thể được truy cập. |

Regression Testing: *Kiểm tra hồi quy là một hoạt động bảo trì cần thiết nhằm chứng tỏ rằng mã không bị ảnh hưởng bất lợi bởi các thay đổi. Xác định kích hoạt cho kiểm tra hồi quy (áp dụng cho các dự án bảo trì), thời gian và phạm vi kiểm tra hồi quy.*

| Test Objective: | Kiểm thử hồi quy là để xác thực các phần đã sửa đổi của phần mềm, để đảm bảo rằng việc sửa đổi đó không gây ra lỗi ở các phần khác. |
| --- | --- |
| Technique: | * Sử dụng lại tập hợp các ca kiểm thử từ một bộ kiểm thử hiện có để kiểm thử một mô-đun đã sửa đổi. * Sử dụng công cụ Rational Robot: Tạo một số kịch bản kiểm tra chức năng. Xác định lịch thực hiện kiểm thử tự động tại đây * 80% Test case được chọn ngẫu nhiên từ Test case hiện có * Xây dựng cơ sở hạ tầng phân tích chương trình. Chúng tôi đang xây dựng một cơ sở hạ tầng có thể mở rộng để thực hiện và đánh giá phân tích chương trình. Dựa trên kết quả phân tích, chúng tôi xác định phạm vi kiểm tra hồi quy |
| Completion Criteria: | * Tất cả các trường hợp kiểm thử được thực hiện và thông qua * Tất cả các trường hợp kiểm thử được chọn đều được thực hiện và thông qua |
| Special Considerations: |  |

Usability test: *Kiểm thử khả năng sử dụng là một kỹ thuật được triển khai trong thiết kế tương tác lấy người dùng làm trung tâm để đánh giá sản phẩm hoặc dịch vụ bằng cách thử nghiệm nó với một số người dùng.*

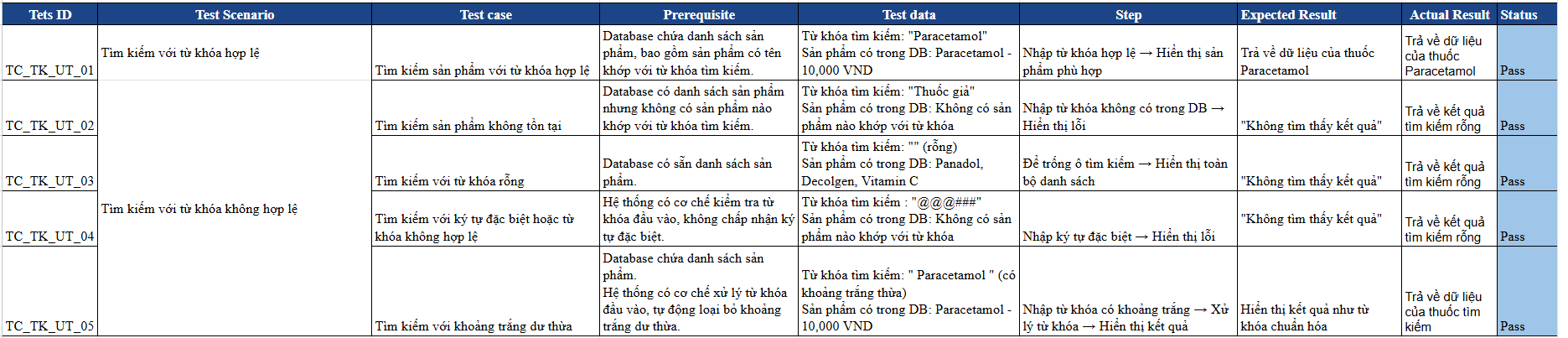
| Test Objective: | Nó chủ yếu nhằm mục đích kiểm tra tính dễ sử dụng của sản phẩm và chức năng. |
| --- | --- |
| Technique: | * Kiểm tra về thời gian tải thực tế . * Kiểm tra xem màu sắc của văn bản tương phản trên nền có phù hợp không. * Kiểm tra xem kích thước phông chữ và khoảng cách giữa các văn bản có thể đọc được chính xác không, có gây rối mắt hay không. * Kiểm tra xem trang không tìm thấy được thiết kế tùy chỉnh nào không. * Kiểm tra xem người dùng có dễ dàng nhận ra điều hướng trang. Bố cục cho các khu vực điều hướng có dễ nhận biết hay không. * Kiểm tra nếu các tùy chọn điều hướng có dễ hiểu và ngắn gọn hay không. * Kiểm tra xem số lượng nút / liên kết có hợp lý không. * Kiểm tra xem phong cách của các liên kết có nhất quán trên tất cả các trang và dễ hiểu không. * Kiểm tra xem khi tìm kiếm có dễ dàng truy cập hay không. |
| Completion Criteria: | * Tất cả các trường hợp kiểm thử được thực hiện và thông qua * Tất cả các trường hợp kiểm thử được chọn đều được thực hiện và thông qua |
| Special Considerations: |  |

Statement testing: *Statement testing là một kỹ thuật kiểm thử hộp trắng (white-box testing) giúp xác định các test case để thực thi một số câu lệnh của một chức năng nào đó.*

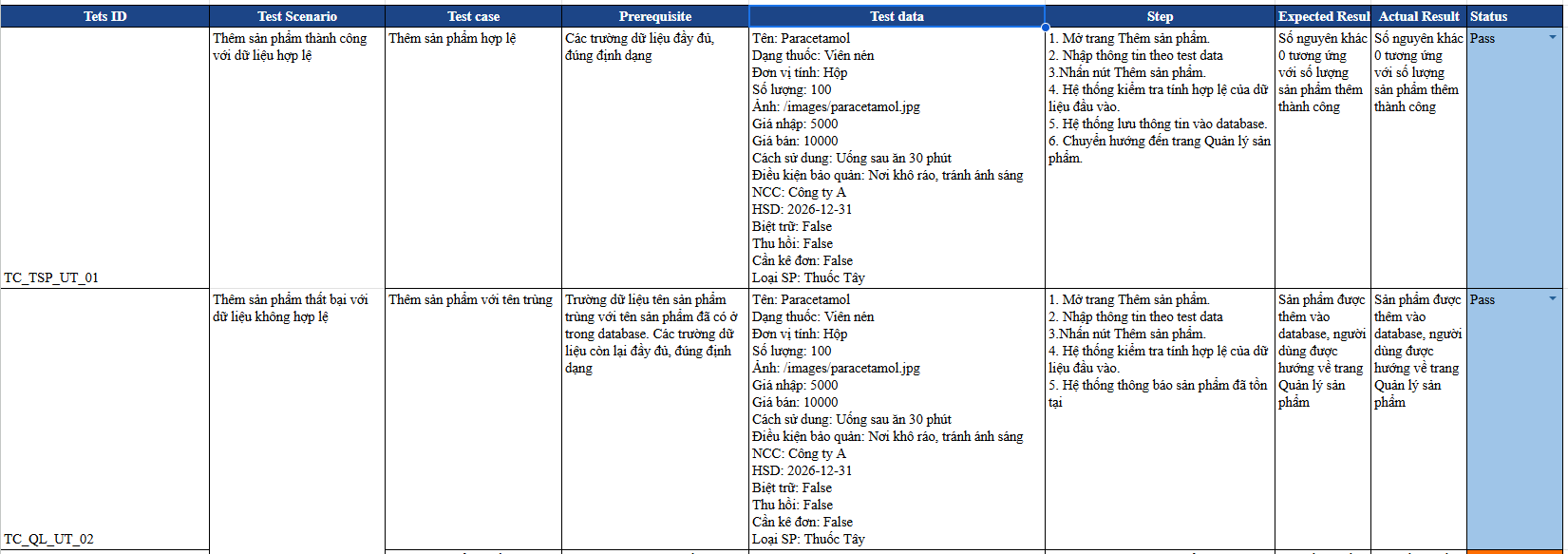
| Test Objective: | Statement testing (kiểm thử câu lệnh) là kỹ thuật dựa vào source code (mã nguồn) để xác định các test case (trường hợp kiểm thử) cần thiết để thực thi một số câu lệnh (statement) của một chức năng hay class nào đó |
| --- | --- |
| Technique: | * Dựa vào mã nguồn (source code) của chương trình, function hay class đang được kiểm để xác định test case. * Mỗi test case được viết ra là để thực thi (execute) một số câu lệnh cụ thể nào đó trong đoạn code. |
| Completion Criteria: | Statement coverage của 1 chức năng/hệ thống được xác định theo công thức: số câu lệnh được thực thi bởi bộ test case chia cho tổng số câu lệnh của chức năng/hệ thống đó. Thường thì độ bao phủ sẽ được biểu diễn theo dạng % (phần trăm). Số lượng test case tối thiểu để bao phủ đầy đủ (hết 100%) cho một đoạn mã nguồn là số lượng test case thực thi (đi qua) mọi câu lệnh ít nhất một lần. |
| Special Considerations: |  |

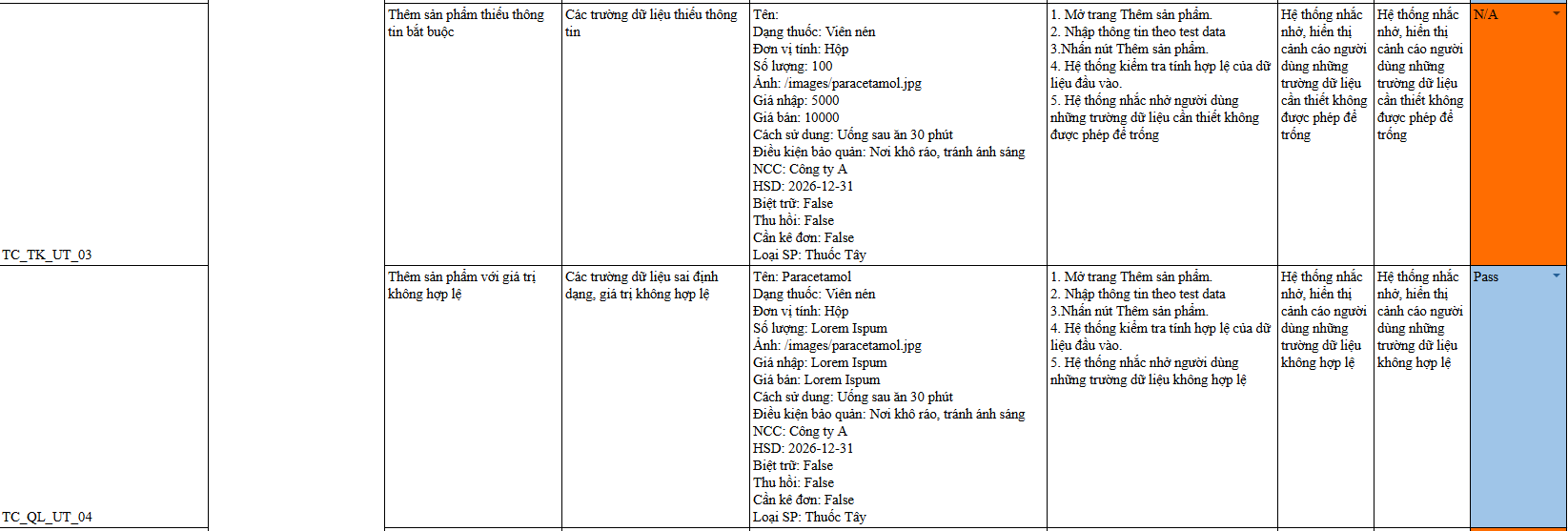
#### 

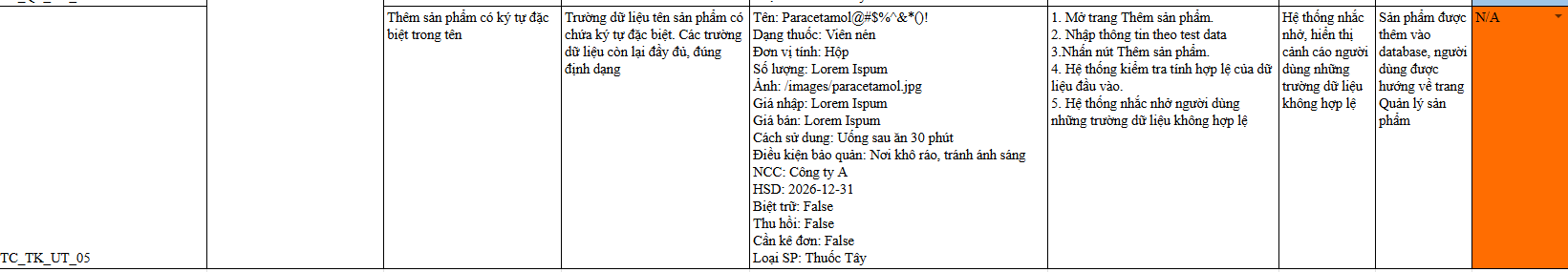
* + 1. Kết quả kiểm thử.
       1. Unit Testing
* Tìm kiếm sản phẩm



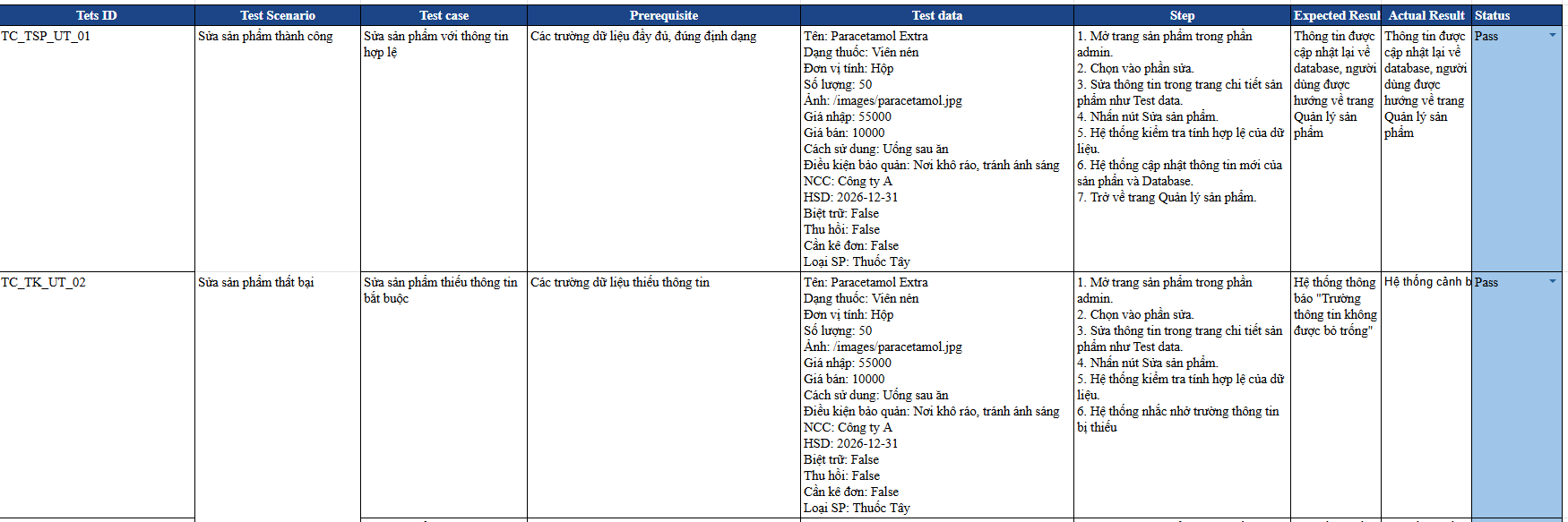
* Thêm sản phẩm

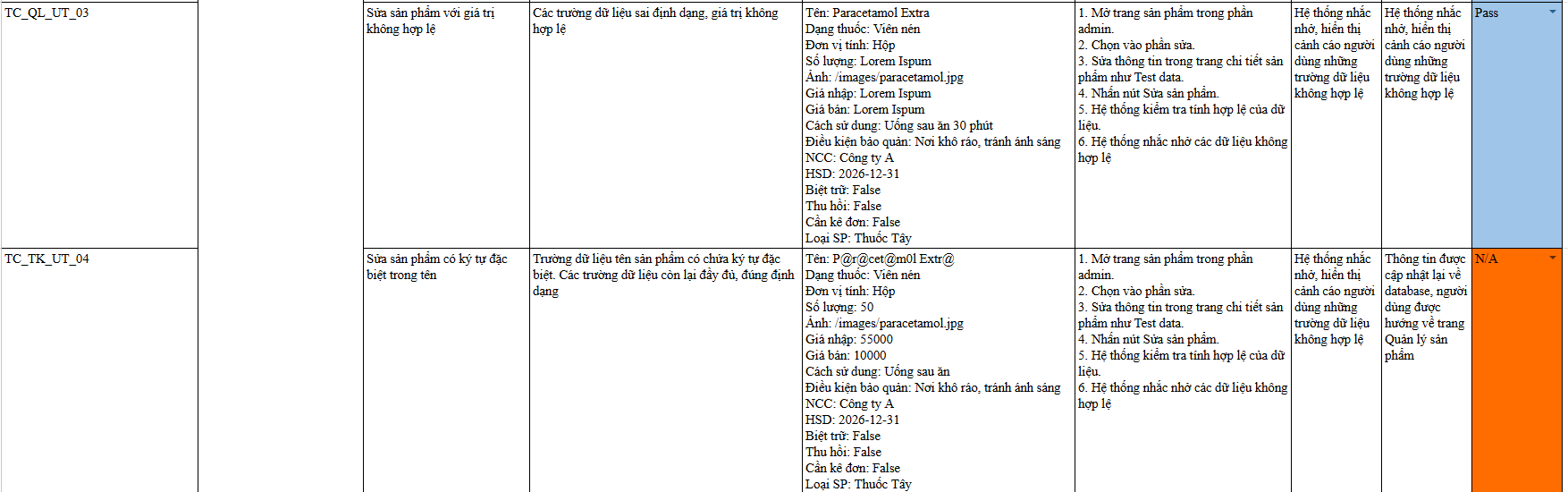




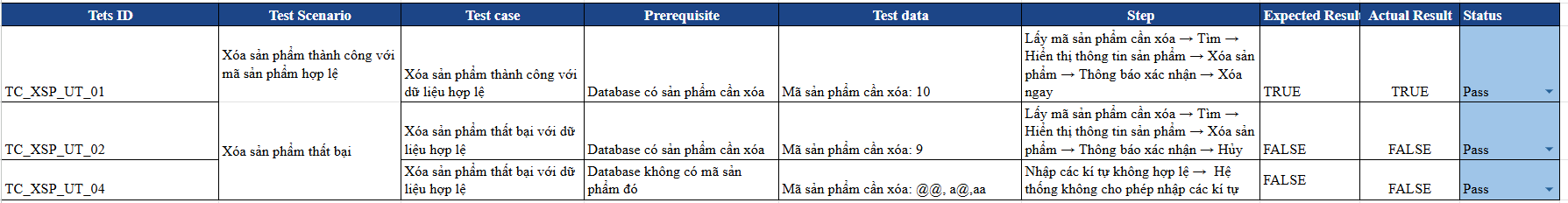


* Sửa sản phẩm

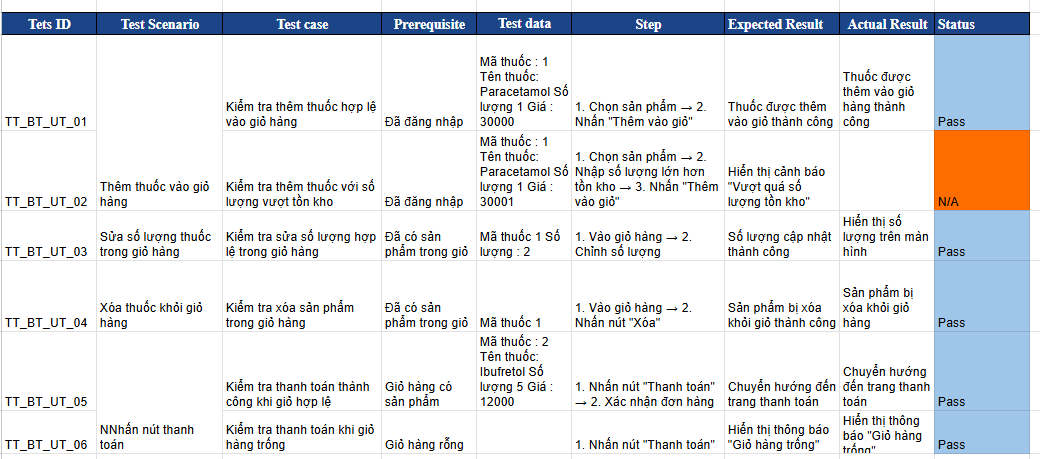




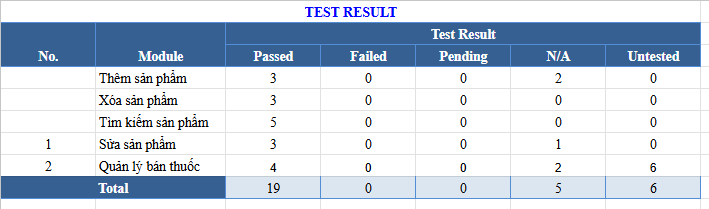
* Xóa sản phẩm



* Quản lý bán thuốc



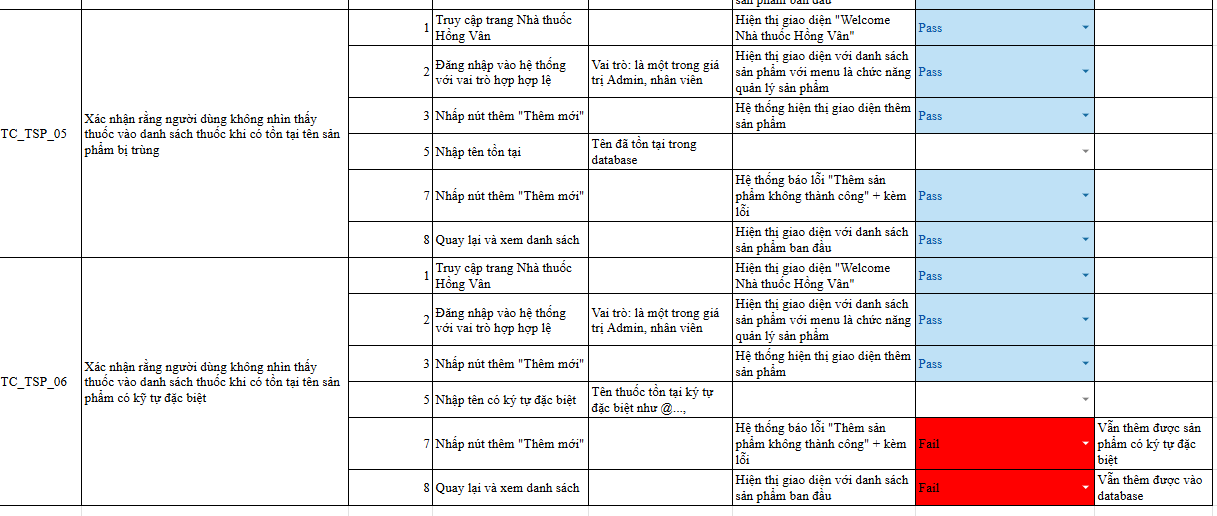
* Test result:



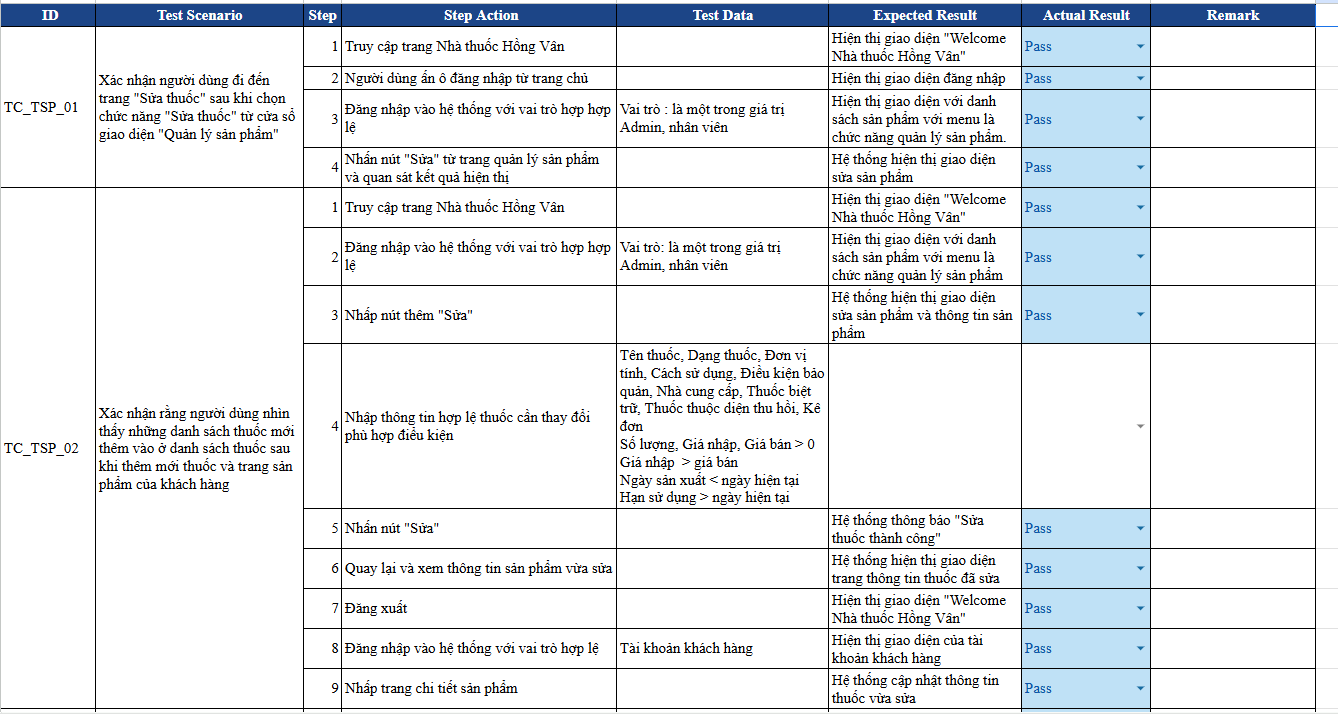
* + - 1. Integration Testing
* Thêm

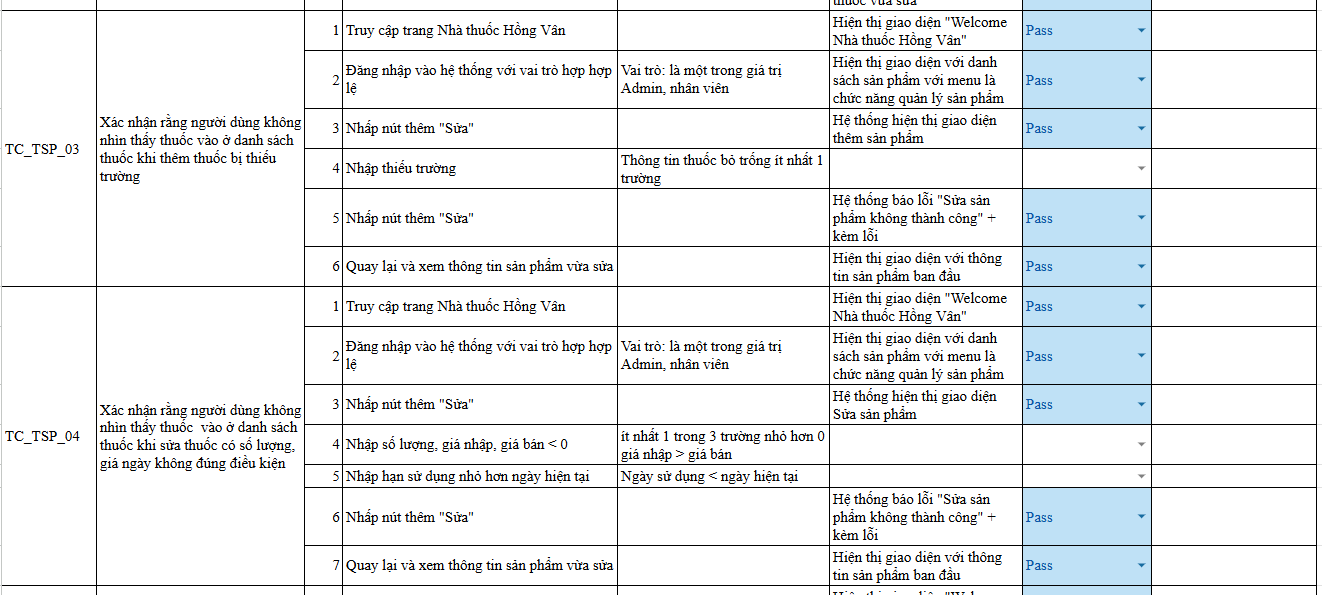


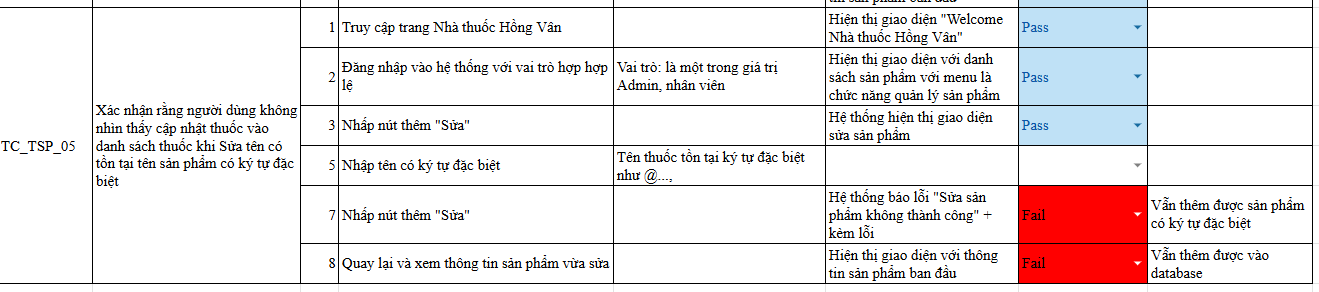




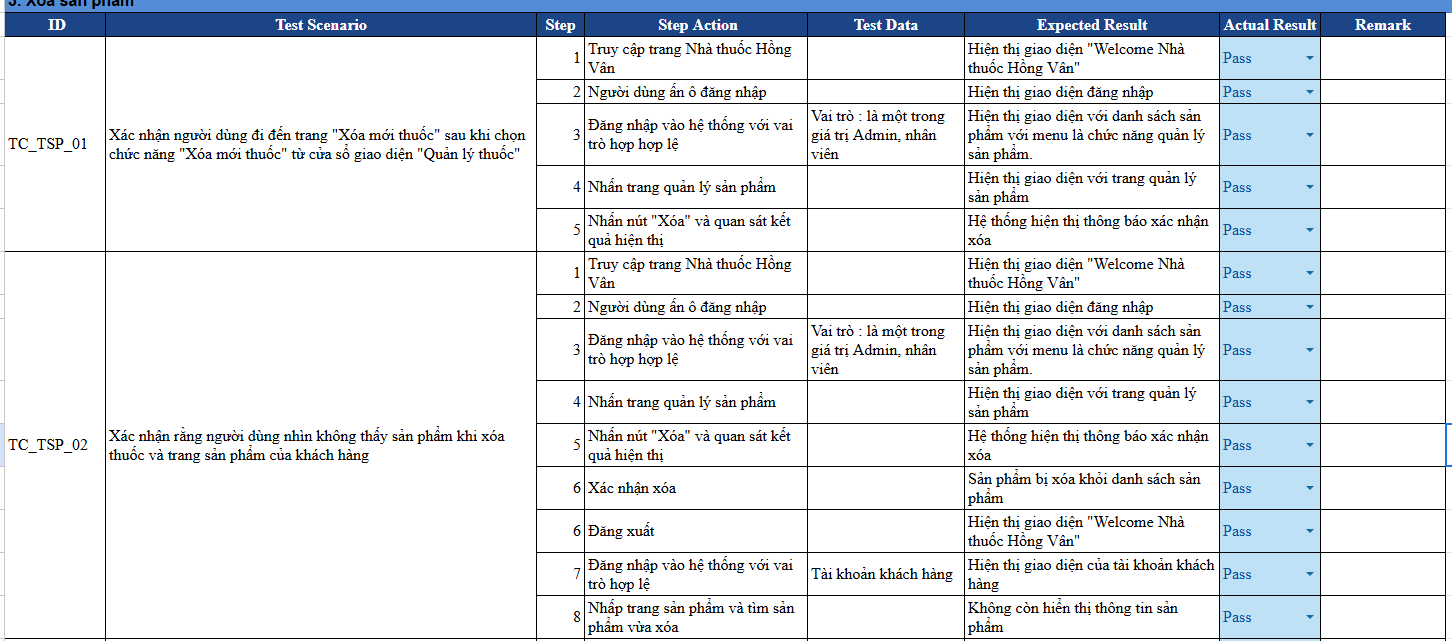
* Sửa

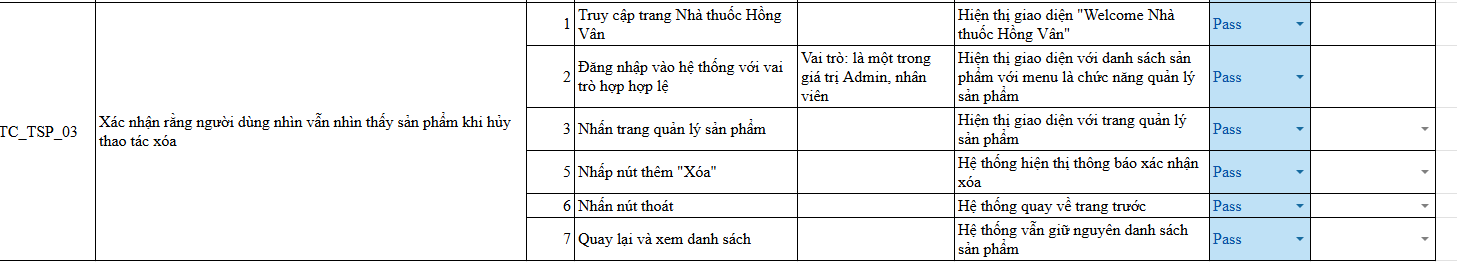




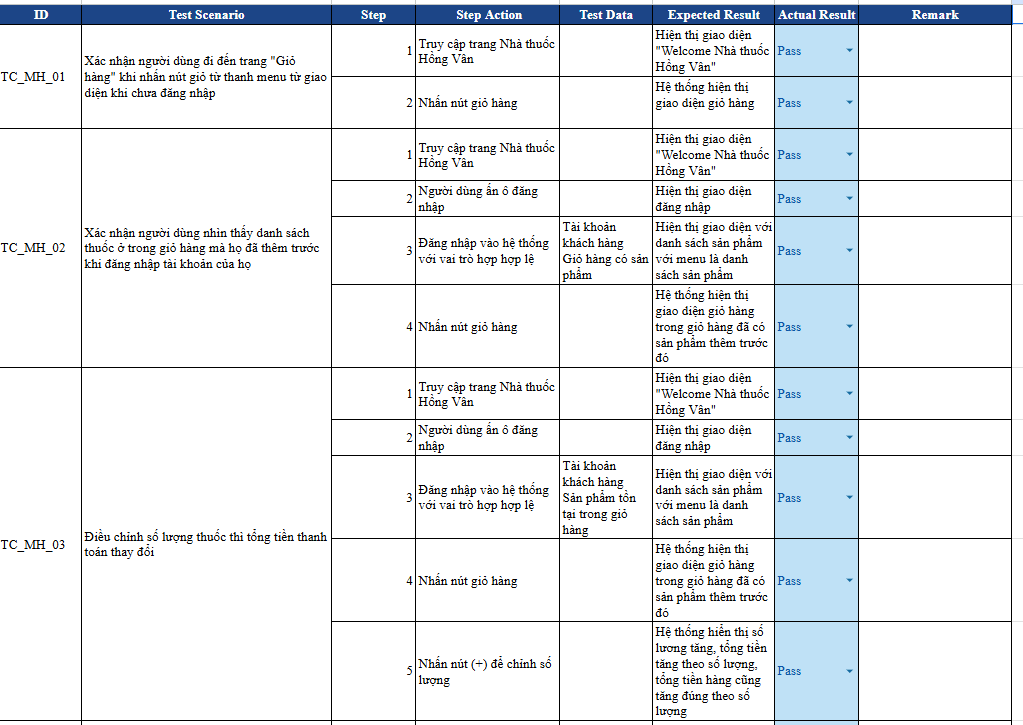


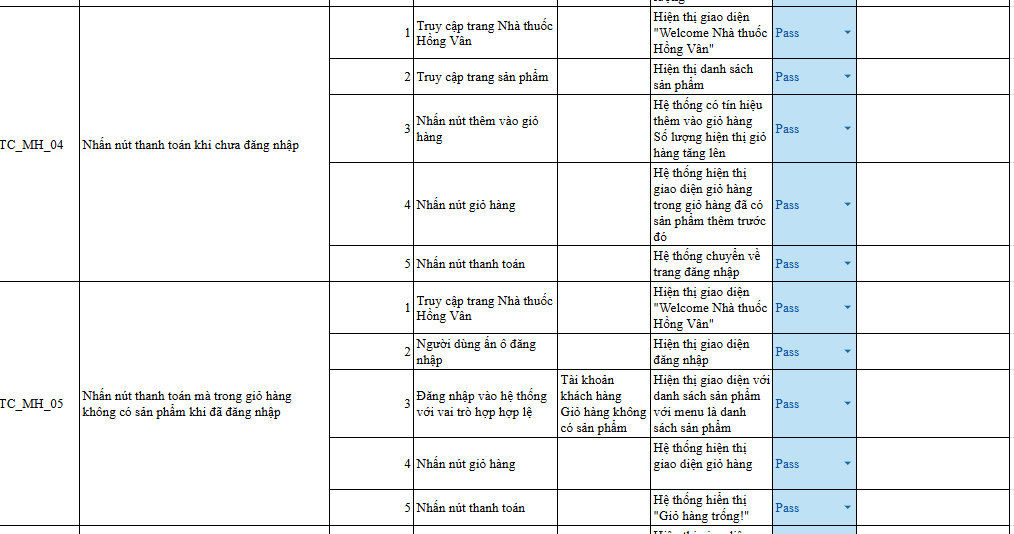
* Xóa

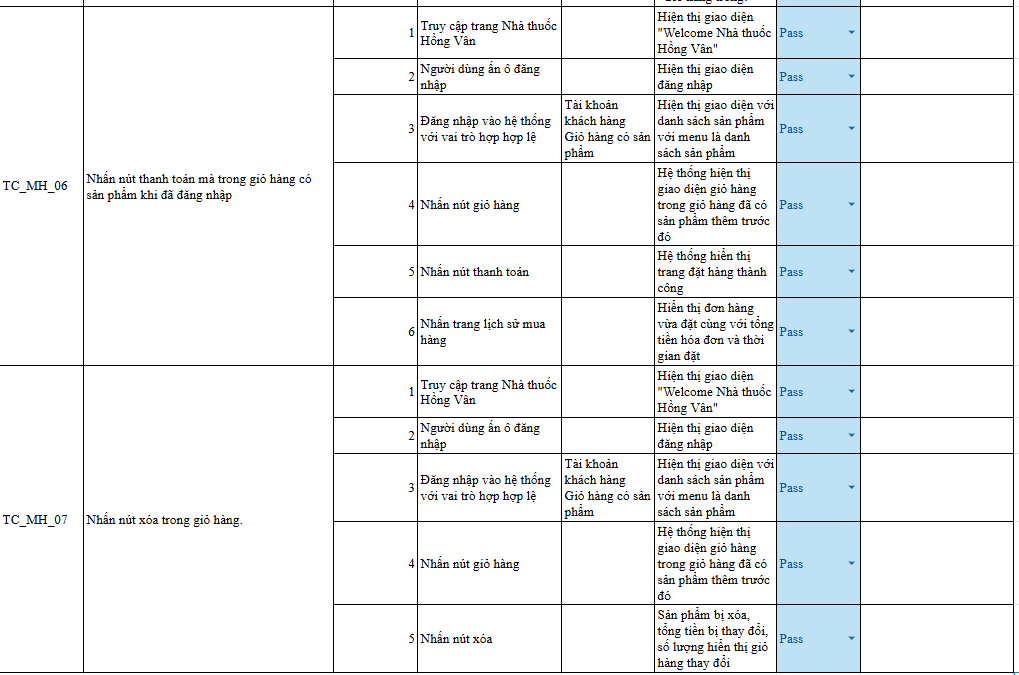




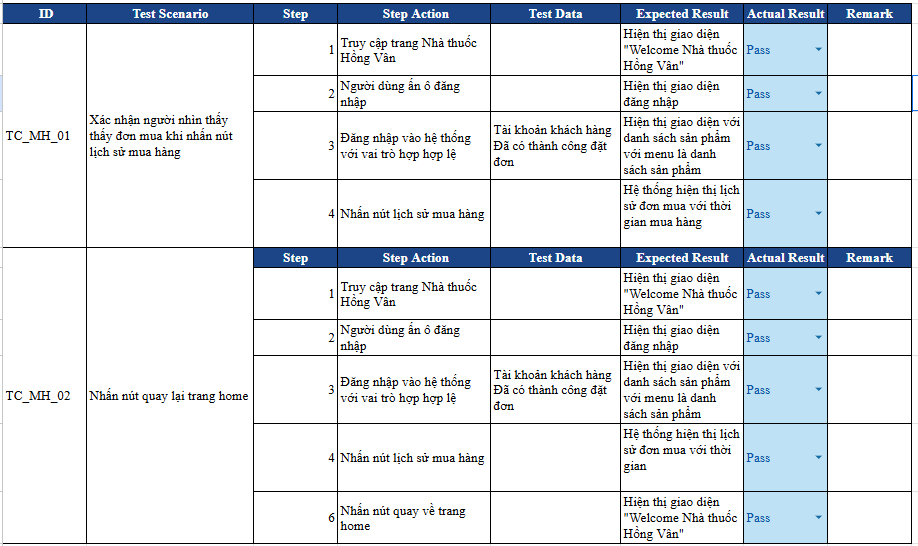
* Giỏ hàng



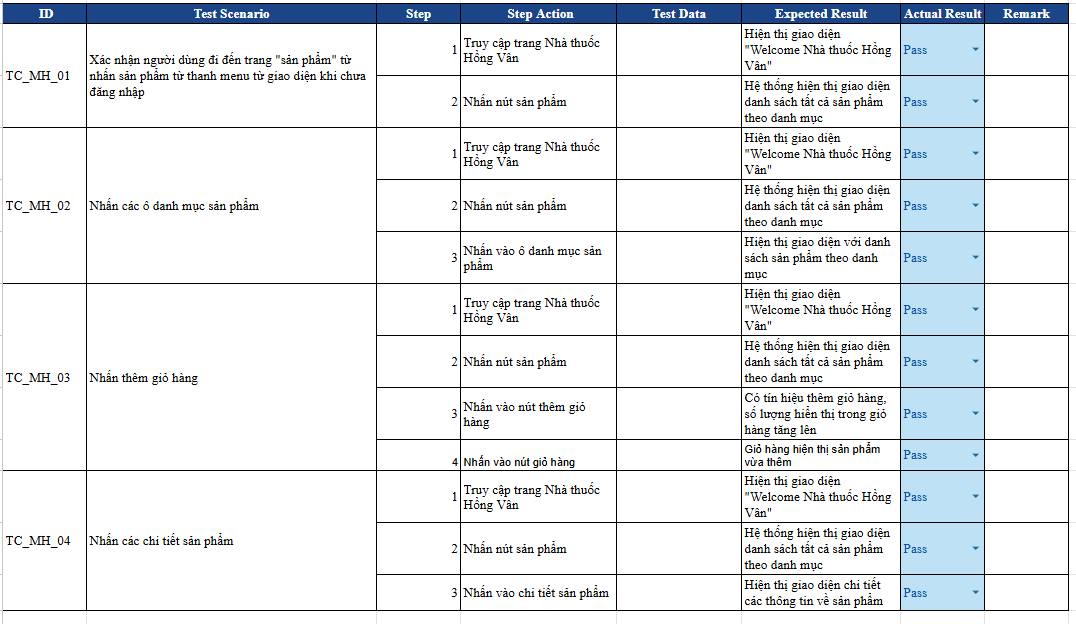




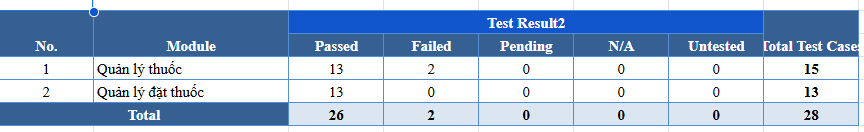
* Lịch sử mua hàng



* Danh sách sản phẩm



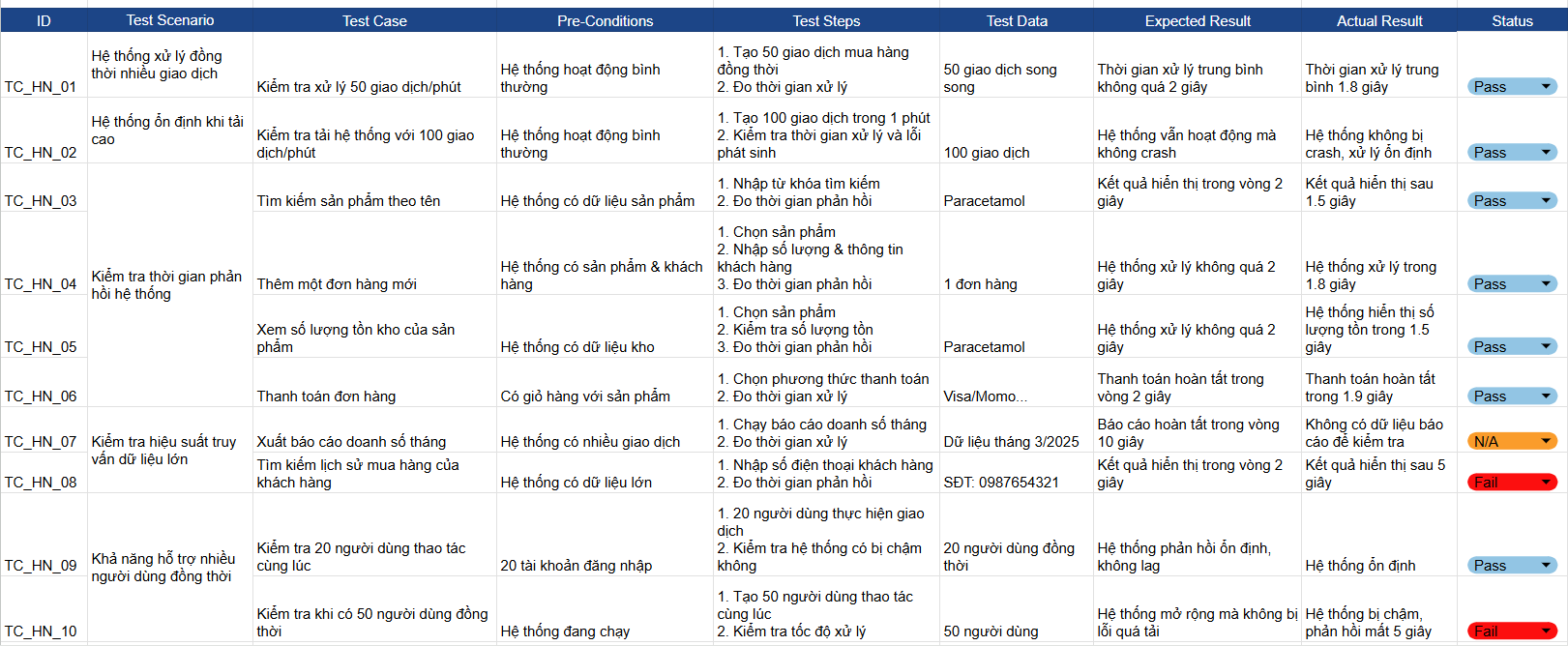
* Test result

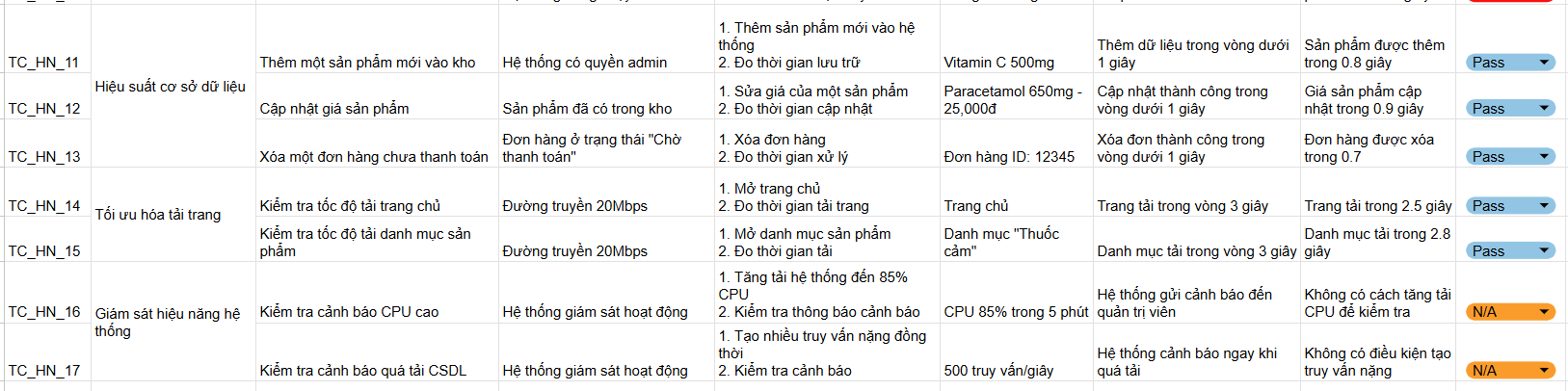


* + - 1. System Testing
* Kiểm tra độ tin cậy



* Kiểm tra hiệu năng

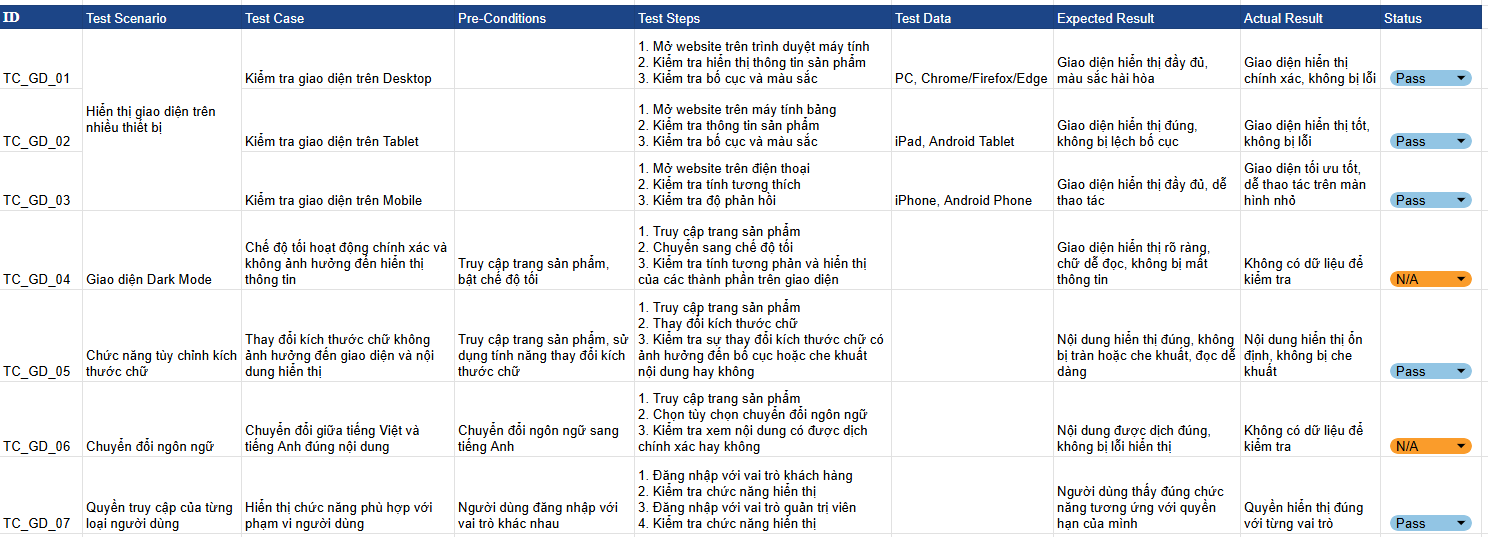




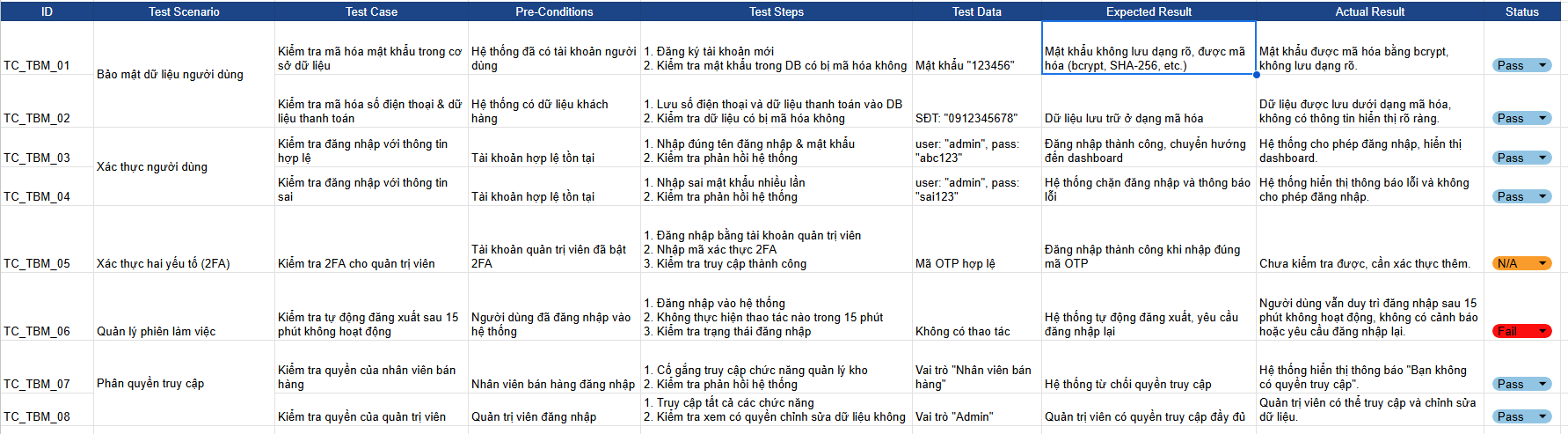
* Kiểm tra dữ liệu

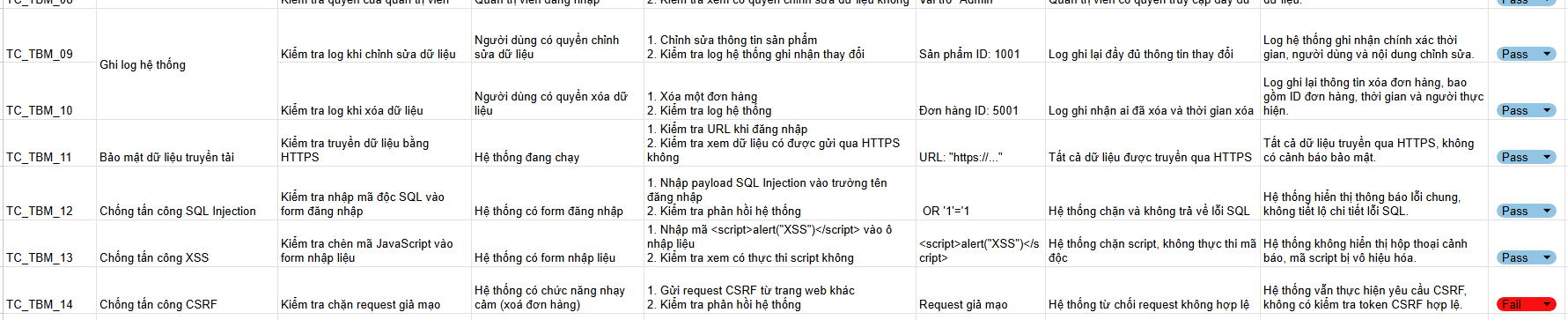


* Kiểm tra giao diện

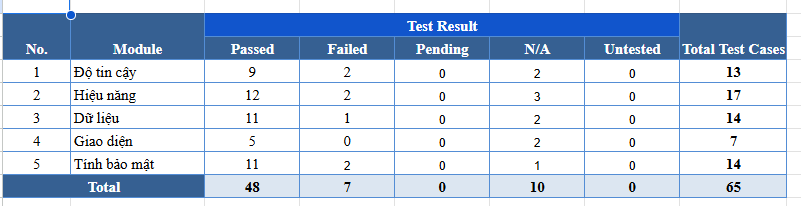


* Kiểm tra tính bảo mật

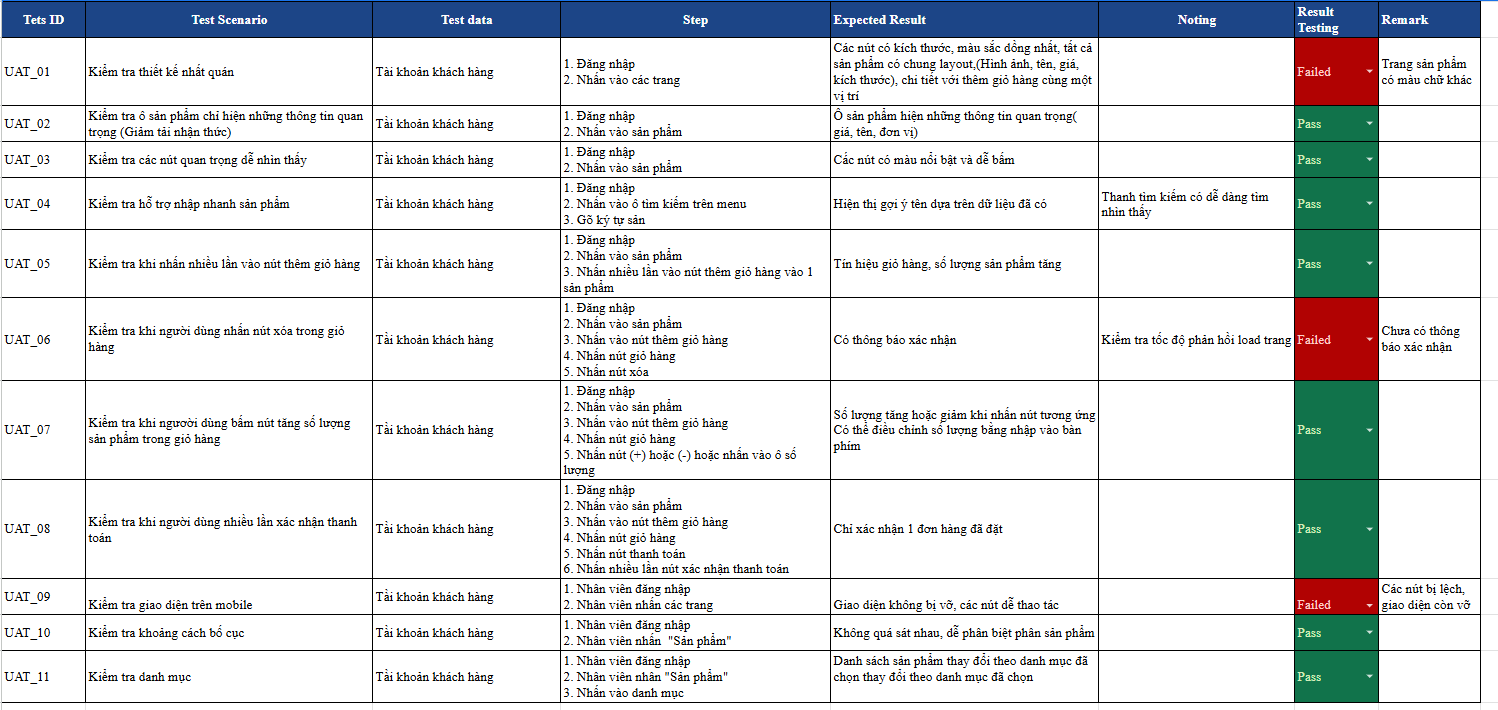




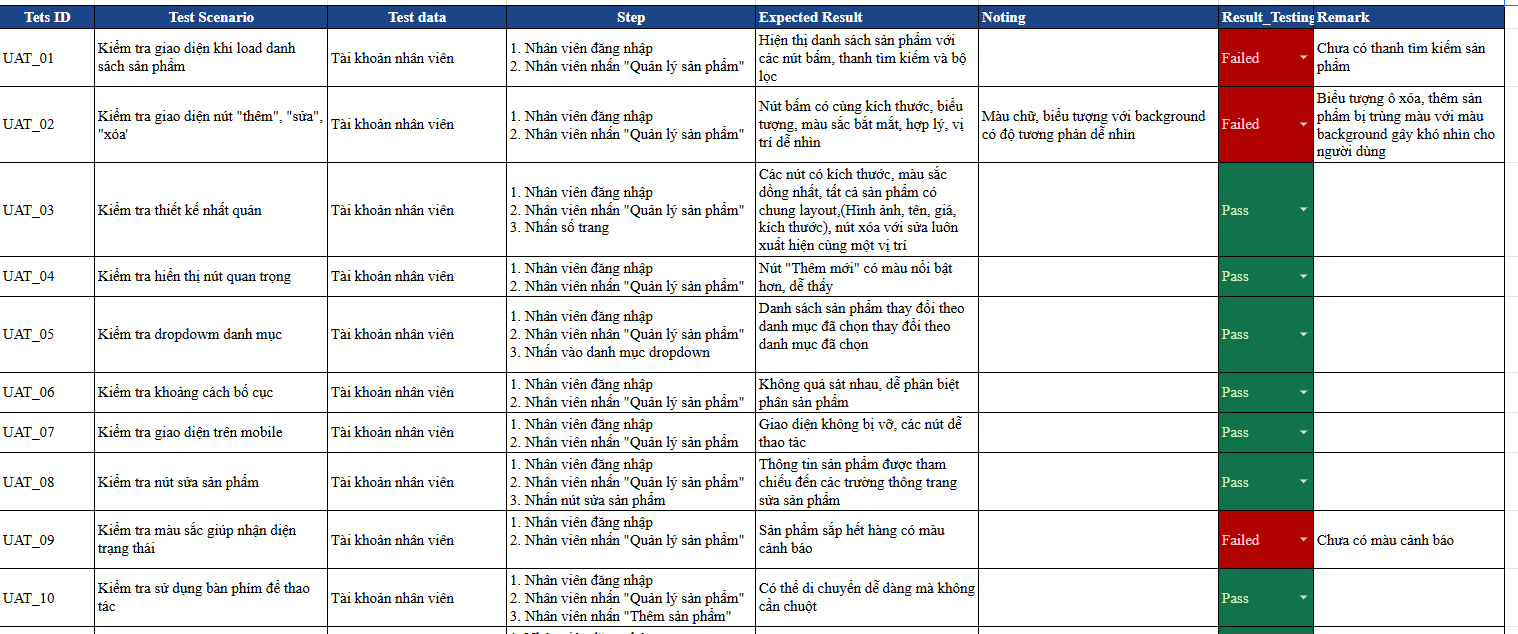
* Test result:

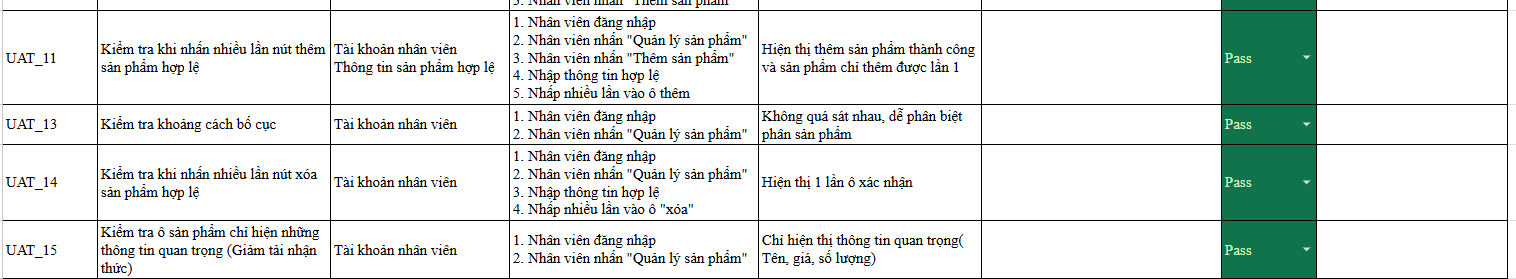


* + - 1. Acceptance Testing
* Quản lý đặt thuốc

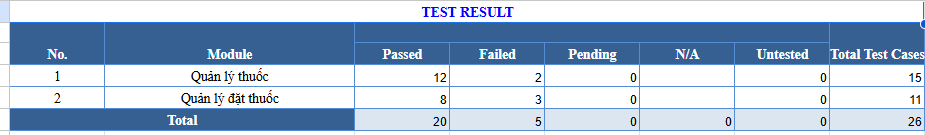


* Quản lý thuốc





* Test result:



* + 1. Resource.
* Source code: <https://github.com/honganhss/BTLNhapMonCNPM>
* Unit test: [G12\_Unit Test](https://docs.google.com/spreadsheets/d/1kjqbnnEcZu3s-XyjM9F2OhkJCSTByZ4nTAv0K9K-Ulw/edit?gid=1433020941#gid=1433020941)
* System test: [G12\_SystemTest](https://docs.google.com/spreadsheets/d/1DnhtkwN04PVIuWXmQXs7SOEvwNNeUe7OFfOHJlFHKe4/edit?gid=474662979#gid=474662979)
* Integration test: [G12\_Integration\_Test](https://docs.google.com/spreadsheets/d/1qwVNp0m_N1a18rKtSTLuC8PNgYoD2PwnGvFEU_cD5PU/edit?gid=1865560927#gid=1865560927)
* Acceptance test: [G12 Acceptance Testing](https://docs.google.com/spreadsheets/d/1JRwnhbxAA_FlbW-ZF2usiUWrtUvIaQkaOkRig5KOEXw/edit?gid=0#gid=0)
  + 1. DELIVERABLES.

| **STT** | **Sản phẩm** | **Ngày giao** | **Cung cấp bởi** | **Giao cho** |
| --- | --- | --- | --- | --- |
| 1 | Test Plan | 10/3/2025 | Nguyễn Mai Hương  Trần Thị Mai Thi | DT - G12 |
| 2 | Unit Test cases | 13/3/2025 | Nguyễn Mai Hương  Nguyễn Văn Mạnh | DT - G12  rAI-G12 |
| 3 | Integration Test Cases | 14/3/2025 | Trần Thị Mai Thi | DT - G12 |
| 4 | System Test cases | 16/3/2025 | Nguyễn Mai Hương | DT - G12 |
| 5 | Acceptance Test cases | 16/3/2025 | Trần Thị Mai Thi | DT - G12 |

## CHƯƠNG 6. KẾT LUẬN

### Đánh giá chất lượng phần mềm

Đánh giá chất lượng phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GPP:

* **Theo các yếu tố chất lượng:**
* Tính đúng đắn: Phần mềm không gây ra các lỗi lớn và sai sót trong quá trình hoạt động, các lỗi đã được phát hiện trong quá trình kiểm thử đã được khắc phục trước khi đến tay khách hàng.
* Tính tin cậy: Phần mềm không phát sinh các lỗi và hoạt động ổn định trong 1 thời gian dài tiến hành dùng thử.
* Tính khả dụng: phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GPP được tạo ra có giao diện, hình ảnh, màu sắc hài hòa bắt mắt, thân thiện với người dùng, có khả năng thích ứng với các thiết bị và hệ điều hành khác nhau.
* Tính toàn vẹn: Phần mềm có các tính năng bảo mật ở mức cơ bản cho thông tin và dữ liệu của người dùng
* Tính hiệu quả: Phần mềm chưa được chạy thử và hoạt động trong 1 môi trường thực tế, chưa đảm bảo được khả năng có thể xảy ra lỗi hoặc sự cố hay không khi hoạt động trên môi trường thực tế trong 1 khoảng thời gian dài.
* Tính khả chuyển: Phần mềm chưa hoạt động được đa nền tảng, đa thiết bị.
* Tính tái sử dụng: Có thể sử dụng mã nguồn của phần mềm phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GPP để phát triển các dự án quản lý nhà thuốc sau này.
* Tính bảo trì:
* Tính linh hoạt: Phần mềm có khả được bảo trì và nâng cấp 1 cách dễ dàng mà không ảnh hường đến các thành phần khác trong hệ thống nhờ lựa chọn mô hình
* Tính kiểm thử được: Phần mềm có thể kiểm thử 1 cách dễ dàng nhờ các thành phần phân tách thành các module độc lập và các yêu cầu rõ ràng từ khách hàng về phần mềm nhờ kiến trúc Layered Architecture kết hợp với kiến trúc MVC giúp hệ thống vận hành ổn định nhờ việc tách biệt các thành phần thành nhiều lớp độc lập và hệ thống có thể dễ dàng thay đổi giao diện mà không ảnh hưởng đến logic xử lý.
* **Theo các pha SDLC:**
* Phân tích yêu cầu: Đã thu thập được các thông tin sơ khai về phần mềm phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GP, phân tích đúng yêu cầu của 1 quản lý nhà thuốc và đúng theo yêu cầu của người dùng.
* Thiết kế: Hệ thống có giao diện đơn giản, dễ nhìn, quen thuộc thuận tiện cho người dùng dễ sử dụng
* Lập trình: Các sinh viên thực hiện hoạt động phát triển phần mềm đã tiến hành lập trình đầy đủ các chức năng cơ bản cần có cho 1 phần mềm quản lý nhà thuốc: đăng ký, đăng nhập, quản lý thuốc,...
* Kiểm thử: Đã phát hiện và loại bỏ được hết các lỗi nghiêm trọng, độ bao phủ của các test case cao, chưa tiến hành kiểm thử hết được tất cả các chức năng của phần mềm
* Quá trình phát triển tiếp tục và cải tiến sản phẩm: Các tester liên tục thực hiện lại các kiểm thử đã được thực hiện trước đó để khắc phục các lỗi phát sinh cũng như phát triển thêm các tính năng mới cho phần mềm hệ thống quản lý nhà thuốc.

### Ưu điểm & Nhược điểm của đề tài

* Ưu điểm
* Tính chuyên nghiệp: Phần mềm quản lý nhà thuốc Hồng vân chứa nhiều dữ liệu và hoạt động trong môi trường thực tế trong một khoảng thời gian dài, do đó chưa đủ dữ liệu để đánh giá độ hiệu quả và đáp ứng của phần mềm trong thực tế.
* Tính độc đáo: Phần mềm quản lý nhà thuốc Hồng Vân chứa dữ liệu người dùng, bao gồm các thông tin cá nhân và quá trình mua hàng, do đó nó đòi hỏi tính bảo mật cao để đảm bảo an toàn thông tin người dùng.
* Tính tiện ích: Phần mềm hỗ trợ danh mục quản lý thuốc, kiểm soát hạn sử dụng một cách chính xác.
* Tính khả chuyển: Phần mềm có khả năng mở rộng và nâng cấp để đáp ứng nhu cầu phát triển của nhà thuốc, có thể tích hợp với các hệ thống thanh toán điện tử,...
* Nhược điểm
* Chưa được kiểm thử thực tế một cách toàn diện: Dù đã triển khai trong môi trường thực tế, nhưng phần mềm vẫn cần được đánh giá qua quá trình kiểm thử chặt chẽ để đảm bảo tính ổn định, hiệu suất, và khả năng xử lý dữ liệu lớn.
* Đòi hỏi tính bảo mật cao: Dữ liệu cá nhân và thông tin liên quan đến thuốc yêu cầu chính sách bảo mật nghiêm ngặt để tránh rò rỉ dữ liệu hoặc bị tấn công từ bên ngoài. Cần áp dụng các biện pháp bảo vệ như mã hóa dữ liệu, kiểm soát quyền truy cập và cập nhật bảo mật thường xuyên.
* Quản lý nội dung: Việc cập nhật danh mục thuốc, thông tin nhà cung cấp, giá cả và quy định pháp lý cần được thực hiện thường xuyên để đảm bảo tính chính xác và phù hợp với quy định của Bộ Y tế.
* Chi phí và nguồn lực: Phát triển và duy trì phần mềm quản lý nhà thuốc đạt chuẩn GPP đòi hỏi đầu tư về công nghệ, nhân lực, hệ thống máy chủ, bảo mật, và quản lý dữ liệu. Do đó, phần mềm quản lý nhà thuốc Hồng Vân đạt chuẩn GPP cần có nguồn lực đủ lớn để đáp ứng các yêu cầu này, đồng thời đối mặt với rủi ro tài chính và quản lý nguồn lực trong quá trình phát triển và hoạt động.

### Hướng phát triển

* *Xét theo các yếu tố chất lượng.*
* Hoàn thiện và mở rộng các chức năng: tiếp tục phát triển các tính năng nhưu tích hợp AI để gợi ý thuốc hoặc hỗ trợ tư vấn trực tuyến. Nâng cấp báo cáo để giúp nhà thuốc có cái nhìn tổng quan hơn về hoạt động kinh doanh. Cần xây dựng các kịch bản sao lưu,phục hồi dữ liệu khi gặp rủi ro tránh mất mát dữ liệu.
* Cải thiện bảo mật : Ứng dụng các công nghệ bảo mật tiên tiến, mã hóa dữ liệu đầu cuối và giám sát an ninh mạng
* Tối ưu hóa tốc độ xử lý để nâng cao hiệu suất và trải nghiệm ứng dụng
* Cần hoàn thành bản thiết kế tiến trình hệ thống và xác định được các tình huống kiểm soát người dùng.
* Cần tiếp tục viết chương trình để có thể hoàn thành các chức năng đã được phân tích.
* *Xét theo các pha SDLC*
* Nên lập kế hoạch và triển khai: Sau khi có thiết kế phần mềm, kế hoạch triển khai phần mềm cũng cần được đưa ra để đảm bảo rằng dự án được triển khai một cách hiệu quả.

….

## TÀI LIỆU THAM KHẢO

[1] IEEE. (2011). *IEEE Standard for Software and System Requirements Engineering (IEEE Std 29148-2011)*. Institute of Electrical and Electronics Engineers.

[2] Bộ Y Tế. (2018, 01 22). *Thông Tư Quy định về thực hành tốt cơ sở bán lẻ thuốc* [Số: 02/2018/TT-BYT]. Văn phòng Chính phủ.

## PHỤ LỤC A: QUY TẮC ĐẶT TÊN

- Tên đặt theo tiếng anh hoặc tiếng Việt không dấu, có nghĩa và viết hoa chữ cái đầu mỗi từ, viết liền

- Tên bao gồm: tên class, tên biến, tên hàm, tên file,...

- Tên gắn liền với đặc điểm, chức năng tương ứng từng đối tượng

- Đặt tên trong cơ sở dữ liệu:

+ Tên thuộc tính: <chữ đầu tiên của kiểu dữ liệu><tên thuộc tính>

VD: iMaNV,...

+ Tên bảng: tbl<tenbang>

VD: tblNguyenLieu

- Project name convention

| **STT** | **Identifier type** | **Convention** |
| --- | --- | --- |
| 1 | Tên folder, namespace, variable | PascalCase |
| 2 | Tên file, class, method | camelCase |
| 3 | Constant | Uppercase |

- Design Convention

| **STT** | **Type** | **Convention** |
| --- | --- | --- |
| 1 | Label | lb |
| 2 | Textbox | tb |
| 3 | Button | btn |
| 4 | Checkbox | ck |
| 5 | Radio button | rb |
| 6 | ComboBox | cb |
| 7 | ListBox | lb |
| 8 | DataGridView | dvg |
| 9 | Panel | panel |

## PHỤ LỤC B: CÁC BIỂU MẪU, TÀI LIỆU THU THẬP ĐƯỢC

1. [SRS Document](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document)
2. [SRS Template](https://googlegroups.com/group/do_an_H3T/attach/2cf8749f7fbb2d1b/Template%20-%20SRS.docx?part=0.1) 1
3. [SRS Template 2](https://qndev.github.io/resources/SRS.pdf)
4. [G12.Hồ sơ khảo sát](https://docs.google.com/document/d/1ALCzONCEs7G-Ox7ErvC1V8i3IytmMqq2UcjOaFArXH0/edit?usp=drive_link)
5. [IEEE 29148:2011](https://drive.google.com/file/d/1mcaxdfvAZMI9cxPVM_EbeCk94PpuF73F/view?usp=drive_link)
6. [TT\_BYT](https://drive.google.com/file/d/11VwZrraFDrP74-NtwTWBm8uJ93V97m7m/view?usp=drive_link)

## PHỤ LỤC C: CÁC BIÊN BẢN TIẾN ĐỘ

[G12.Tiến độ công việc](https://docs.google.com/spreadsheets/d/12vFWyBBGif4XYIxHvN3KaTogdLE4B4Y4MXWvRrsr7pI/edit?usp=sharing)

## PHỤ LỤC D: CÁC BIÊN BẢN RÀ SOÁT KỸ THUẬT

[G12.Biên Bản rà soát kỹ thuật 15/2/25](https://docs.google.com/document/d/1VZK35lPnr4tC-XmE6NhVQgvdDI9UTBlvRJHMZg7nd8E/edit?usp=sharing)

## PHỤ LỤC E: HỒ SƠ TOÀN BỘ DỰ ÁN

1. [Hồ sơ khảo sát](https://docs.google.com/document/d/1ALCzONCEs7G-Ox7ErvC1V8i3IytmMqq2UcjOaFArXH0/edit?usp=sharing)
2. [Hồ sơ phân tích](https://docs.google.com/document/d/19pF6bx8D_kEcg23OYYQcQWlnGhJ8FuQYJWPvIFq3ko0/edit?usp=sharing)
3. [Hồ sơ thiết kế](https://docs.google.com/document/u/3/d/1GMlLzu13DR5tgYWxQ1Rr8bIvO6wkwQ7qs2NY9JgcMZw/edit)
4. [Mã nguồn](https://drive.google.com/drive/folders/1BZs3QMKoNiT7kln68RS5ooWfTrsu15YY?usp=drive_link)
5. [Hồ sơ kiểm thử](https://drive.google.com/drive/folders/1GLvwWPvlit1XPkom3URcqgDGPy1XD5Uf)