**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH**

**ĐỒ ÁN MÔN HỌC**

**APP MEDCARE**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Môn: QUẢN LÝ DỰ ÁN CÔNG NGHỆ THÔNG TIN

|  |  |  |
| --- | --- | --- |
| **GVHD:** | Th.S Nguyễn Hữu Trung |  |
| **SVTH:** | **MSSV** | **Lớp** |
| Trần Duy Tân | 2280602873 | 22DTHH2 |
| Nguyễn Thiên Bửu Ngọc | 2280616535 | 22DTHH2 |
| Nguyễn Trọng Dũng | 2280600387 | 22DTHH2 |
| Võ Minh Thuận | 2280603164 | 22DTHH2 |

***Thành phố Hồ Chí Minh, tháng 9 năm 2025***

**DANH MỤC HÌNH ẢNH**

[Hình 1 Trang chủ ứng dụng mua bán thuốc Medcare 48](#_Toc212538606)

[Hình 2 Trang đặt hàng 49](#_Toc212538607)

[Hình 3 Hiển thị đơn hàng 50](#_Toc212538608)

[Hình 4 Profile 51](#_Toc212538609)

[Hình 5 Chi tiết Sản phẩm 52](#_Toc212538610)

[Hình 6 Chỉnh sửa hồ sơ 53](#_Toc212538611)

[Hình 7 Lịch sử đơn đặt hàng 54](#_Toc212538612)

[Hình 8 Danh sách sản phẩm yêu thích 55](#_Toc212538613)

[Hình 9 Địa chỉ nhận hàng 56](#_Toc212538614)

[Hình 10 Phương thức thanh toán 57](#_Toc212538615)

[Hình 11 Cài đặt 58](#_Toc212538616)

[Hình 12 Trang đăng ký tài khoản mới 59](#_Toc212538617)

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc212538555)

[**I. Đặt vấn đề** 1](#_Toc212538556)

[**II. Nhiệm vụ của đồ án** 2](#_Toc212538557)

[**1. Mục tiêu của dự án** 2](#_Toc212538558)

[**2. Yêu cầu** 3](#_Toc212538559)

[**2.2. Yêu cầu phi chức năng** 5](#_Toc212538560)

[**3. Môi trường phát triển** 6](#_Toc212538561)

[**III. Các công nghệ sử dụng** 7](#_Toc212538562)

[**1. Visual Studio Code** 7](#_Toc212538563)

[**2. Flutter** 7](#_Toc212538564)

[**3. SQL Server** 7](#_Toc212538565)

[**4. WBS (Work Breakdown Structure - Cấu trúc phân rã công việc)** 8](#_Toc212538566)

[CHƯƠNG II: PHÂN TÍCH 8](#_Toc212538567)

[I. Mô hình thiết kế 8](#_Toc212538568)

[**1. Mô hình thực thể kết hợp (ERD)** 8](#_Toc212538569)

[**2. Mô hình chức năng Use case** 10](#_Toc212538570)

[**2.1 Mô hình tổng quát** 10](#_Toc212538571)

[**2.2 Mô hình phân rã** 10](#_Toc212538572)

[**2.2 Mô hình tuần tự** 14](#_Toc212538573)

[**Activity Diagram Đăng ký tài khoản** 14](#_Toc212538574)

[**Activity Diagram Kiểm tra kho hàng** 16](#_Toc212538575)

[**Activity Diagram Đặt hàng.** 17](#_Toc212538576)

[**Activity Diagram Thanh toán giỏ hàng.** 17](#_Toc212538577)

[CHƯƠNG III: QUẢN LÝ DỰ ÁN 19](#_Toc212538578)

[1: Mô tả ngắn gọn về đề tài 28](#_Toc212538579)

[2: NHẬN DIỆN 10 RỦI RO CỤ THỂ 28](#_Toc212538580)

[3: PHÂN TÍCH SÂU MỨC ĐỘ RỦI RO (10 RỦI RO) 34](#_Toc212538581)

[4: LẬP KẾ HOẠCH ĐỐI PHÓ VỚI 2 RỦI RO CHÍNH 36](#_Toc212538582)

[1. Đối phó với Rủi ro R7: Phạm vi bị mở rộng (Scope Creep) 36](#_Toc212538583)

[2. Đối phó với Rủi ro R9: Dữ liệu Hạn sử dụng/Tồn kho sai sót 37](#_Toc212538584)

[1. Theo dõi rủi ro trong suốt 8 tuần 38](#_Toc212538585)

[2. Đánh giá hiệu quả của hành động đối phó 38](#_Toc212538586)

[3. Điều chỉnh nếu rủi ro thay đổi 39](#_Toc212538587)

[5. Quản lý Truyền thông (Communication Management) 39](#_Toc212538588)

[6. Quản lý Nhân lực (Resource Management) 41](#_Toc212538589)

[7. Quản lý Các bên Liên quan, Mua sắm (Stakeholder and Procurement Management) 42](#_Toc212538590)

[8. Quản lý Chất lượng (Quality Management) 44](#_Toc212538591)

[9. Quản lý Tích hợp Dự án (Project Integration Management) 45](#_Toc212538592)

[CHƯƠNG IV: SẢN PHẨM 47](#_Toc212538593)

[**1. Giới thiệu Sản phẩm (MEDCARE App)** 47](#_Toc212538594)

[**2. Kiến trúc Phần mềm (Software Architecture)** 61](#_Toc212538595)

[**3. Công nghệ và Môi trường** 63](#_Toc212538596)

[4. Mô tả và Quá trình Thiết kế Giao diện (UI/UX) 65](#_Toc212538597)

[CHƯƠNG V: KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN 68](#_Toc212538598)

[1. Kết quả Dự án Đạt được 68](#_Toc212538599)

[2. Ưu điểm và Nhược điểm 68](#_Toc212538600)

[CHƯƠNG V: KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN 70](#_Toc212538601)

[**1. Tổng kết Kết quả Đạt được (Success Metrics)** 70](#_Toc212538602)

[**2. Bài học Kinh nghiệm (Lessons Learned)** 72](#_Toc212538603)

[**3. Kết luận và Hướng phát triển** 73](#_Toc212538604)

[TÀI LIỆU THAM KHẢO 75](#_Toc212538605)

# **CHƯƠNG 1: TỔNG QUAN**

## **I. Đặt vấn đề**

Trong bối cảnh xã hội hiện đại và sự phát triển không ngừng của công nghệ số, nhu cầu chăm sóc sức khỏe của con người ngày càng được chú trọng. Tuy nhiên, việc mua thuốc theo phương pháp truyền thống tại các nhà thuốc vật lý vẫn còn tồn tại nhiều bất cập: tốn thời gian di chuyển, tình trạng đông đúc chờ đợi, khó khăn trong việc tìm kiếm các loại thuốc đặc trị, và đặc biệt là sự thiếu hụt thông tin chi tiết, minh bạch về công dụng, liều dùng, tác dụng phụ của sản phẩm. Đối với những người bệnh, người lớn tuổi hoặc người có lịch trình bận rộn, những trở ngại này càng trở nên rõ rệt hơn.

Nhận thấy nhu cầu cấp thiết về một giải pháp tiện lợi, an toàn và đáng tin cậy, dự án "Xây dựng ứng dụng mua bán thuốc online **MedCare**" được ra đời. Mục tiêu của dự án là ứng dụng công nghệ di động để tạo ra một nền tảng kết nối người dùng với các nhà thuốc uy tín, cho phép họ dễ dàng tìm kiếm thông tin, đặt mua thuốc và nhận được sự tư vấn sức khỏe ban đầu một cách nhanh chóng và hiệu quả.

MedCare không chỉ giải quyết bài toán về sự tiện lợi mà còn hướng tới việc nâng cao kiến thức chăm sóc sức khỏe cho cộng đồng thông qua việc cung cấp thông tin dược phẩm đầy đủ và một công cụ chatbot hỗ trợ thông minh.

## **II. Nhiệm vụ của đồ án**

### **1. Mục tiêu của dự án**

Dự án tập trung vào việc xây dựng và triển khai một hệ thống hoàn chỉnh cho ứng dụng "MedCare" phiên bản 1.0, bao gồm các mục tiêu chính sau:

Xây dựng ứng dụng di động đa nền tảng: Phát triển một ứng dụng di động cho cả hệ điều hành iOS và Android từ một nền tảng mã nguồn duy nhất, cung cấp trải nghiệm người dùng đồng nhất và mượt mà.

Cung cấp nền tảng mua sắm thuốc toàn diện: Cho phép người dùng tìm kiếm, tra cứu thông tin chi tiết, đặt mua các sản phẩm dược phẩm và nhận tư vấn ban đầu qua Chatbot.

Xây dựng hệ thống quản trị tập trung: Phát triển một Admin Panel trên nền tảng web để nhà quản trị có thể quản lý hiệu quả các khía cạnh của hệ thống như sản phẩm, đơn hàng, người dùng và xem báo cáo thống kê.

Đảm bảo chất lượng và bảo mật: Đặt yếu tố an toàn và bảo mật thông tin người dùng lên hàng đầu, đồng thời tối ưu hóa hiệu năng để mang lại trải nghiệm tốt nhất.

Thiết lập nền tảng cho sự phát triển trong tương lai: Xây dựng hệ thống với kiến trúc có khả năng mở rộng, sẵn sàng cho việc tích hợp các tính năng nâng cao sau này.

### **2. Yêu cầu**

**2.1. Xác định yêu cầu chức năng**

Các yêu cầu chức năng được trích xuất từ tài liệu đặc tả, chia thành hai phân hệ chính:

A. Phân hệ Người dùng (Ứng dụng di động MedCare):

Quản lý Tài khoản:

* Hỗ trợ đăng ký/đăng nhập bằng email, số điện thoại, hoặc các tài khoản mạng xã hội (Google/Facebook).
* Người dùng có thể quản lý hồ sơ cá nhân, địa chỉ giao hàng, xem lại lịch sử đơn hàng và nhận thông báo đẩy về các cập nhật quan trọng.

Tìm kiếm và Mua sắm:

* Cung cấp công cụ tìm kiếm thông minh theo tên thuốc, thành phần hoặc triệu chứng bệnh.
* Hệ thống bộ lọc và sắp xếp nâng cao giúp người dùng dễ dàng tìm thấy sản phẩm mong muốn.
* Trang chi tiết sản phẩm hiển thị thông tin đầy đủ, hình ảnh 360 độ, đánh giá từ người dùng khác và các sản phẩm liên quan.

Giỏ hàng và Thanh toán:

* Quy trình quản lý giỏ hàng và thanh toán được thiết kế đơn giản, cho phép áp dụng mã giảm giá.
* Hỗ trợ nhiều phương thức thanh toán phổ biến như COD, thẻ ngân hàng và ví điện tử (Momo, ZaloPay, VNPay).
* Tính năng đặc biệt cho phép người dùng tải lên hình ảnh đơn thuốc để nhà thuốc xác nhận đối với các loại thuốc kê đơn.

Chatbot Hỗ trợ:

* Tư vấn các triệu chứng cơ bản và đề xuất thuốc không kê đơn phù hợp.
* Hỗ trợ tra cứu nhanh thông tin thuốc và hướng dẫn quy trình đặt hàng.
* Có khả năng chuyển tiếp các yêu cầu phức tạp đến dược sĩ hoặc nhân viên hỗ trợ.

B. Phân hệ Quản lý (Admin Panel):

Quản lý Hệ thống:

* Quản lý toàn diện danh mục và thông tin chi tiết của sản phẩm.
* Cập nhật số lượng tồn kho theo thời gian thực.

Quản lý Vận hành:

* Theo dõi và cập nhật trạng thái các đơn hàng (Chờ xác nhận, Đang giao, Đã giao, Hủy bỏ).
* Duyệt và xác thực các đơn thuốc do người dùng tải lên.
* Quản lý tài khoản người dùng.

Quản lý Chatbot:

* Cung cấp giao diện để huấn luyện, cập nhật kiến thức và kịch bản cho Chatbot.

Thống kê và Báo cáo:

* Xuất các báo cáo về doanh thu.
* Tình hình tồn kho.
* Phân tích hành vi người dùng để hỗ trợ ra quyết định kinh doanh.

### **2.2. Yêu cầu phi chức năng**

Hiệu năng (Performance): Thời gian phản hồi của hệ thống (API) phải dưới 500ms. Ứng dụng cần hoạt động mượt mà, ổn định trên các thiết bị di động tầm trung trở lên.

Bảo mật (Security): Mọi giao tiếp dữ liệu phải được mã hóa qua giao thức HTTPS. Dữ liệu nhạy cảm của người dùng như mật khẩu, thông tin thanh toán phải được mã hóa. Hệ thống phải tuân thủ các quy định về bảo mật dữ liệu y tế.

Khả năng sử dụng (Usability): Giao diện người dùng (UI) và trải nghiệm người dùng (UX) phải được thiết kế hiện đại, trực quan, thân thiện và dễ dàng điều hướng cho mọi đối tượng người dùng.

Khả năng mở rộng (Scalability): Kiến trúc hệ thống phải được thiết kế linh hoạt, cho phép dễ dàng mở rộng để đáp ứng lượng truy cập tăng cao và tích hợp các tính năng mới trong tương lai.

### **3. Môi trường phát triển**

Phần cứng:

* Máy tính phát triển (PC/Laptop): CPU Core i5 hoặc tương đương trở lên, RAM tối thiểu 16GB, ổ cứng SSD.
* Thiết bị di động: Các thiết bị chạy iOS và Android để kiểm thử.

Phần mềm:

* Hệ điều hành: Windows 10/11 hoặc macOS.
* IDE (Môi trường phát triển tích hợp): Visual Studio Code.
* SDK (Bộ công cụ phát triển phần mềm): Flutter SDK, Android Studio (cung cấp Android SDK và máy ảo), Xcode (cung cấp iOS SDK và simulator).

Hệ quản trị CSDL: Microsoft SQL Server.

Công cụ quản lý CSDL: SQL Server Management Studio (SSMS).

Quản lý mã nguồn: Git, GitHub.

Kiểm thử API: Postman hoặc Insomnia.

## **III. Các công nghệ sử dụng**

### **1. Visual Studio Code**

Đây là một trình soạn thảo mã nguồn (code editor) nhẹ, mạnh mẽ và miễn phí. VS Code có hệ sinh thái extension khổng lồ, hỗ trợ cực tốt cho việc phát triển Flutter và ngôn ngữ Dart, cung cấp các tính năng gỡ lỗi (debug), gợi ý mã (IntelliSense) và tích hợp sẵn Git, giúp tối ưu hóa năng suất của đội ngũ phát triển.

### **2. Flutter**

Flutter là một UI toolkit của Google, cho phép xây dựng ứng dụng di động đẹp, hiệu năng cao cho cả iOS và Android từ một cơ sở mã nguồn duy nhất. Việc này giúp tiết kiệm đáng kể thời gian, chi phí và nguồn lực phát triển so với việc xây dựng hai ứng dụng riêng biệt.

Flutter cũng cung cấp thư viện widget phong phú, giúp tạo ra giao diện người dùng linh hoạt và nhất quán, đáp ứng yêu cầu về khả năng sử dụng của dự án.

### **3. SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ, ổn định và có độ bảo mật cao từ Microsoft. Nó phù hợp để lưu trữ các dữ liệu có cấu trúc của dự án như thông tin người dùng, sản phẩm, đơn hàng. SQL Server cung cấp các công cụ quản trị và phân tích dữ liệu mạnh mẽ, hỗ trợ tốt cho việc tạo các báo cáo thống kê phức tạp theo yêu cầu của Admin Panel.

### **4. WBS (Work Breakdown Structure - Cấu trúc phân rã công việc)**

WBS không phải là một công nghệ phần mềm mà là một phương pháp quản lý dự án nền tảng. Trong đồ án này, WBS được sử dụng để phân rã toàn bộ phạm vi dự án thành các gói công việc (work package) nhỏ hơn, dễ quản lý hơn. Việc áp dụng WBS giúp:

Làm rõ phạm vi: Xác định tất cả các công việc cần thực hiện, tránh bỏ sót yêu cầu.

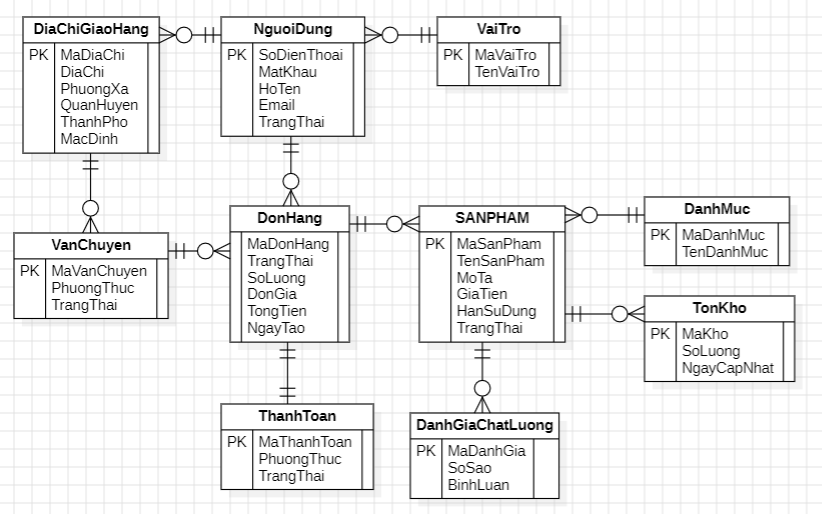
Lập kế hoạch chính xác: Là cơ sở để ước tính thời gian, chi phí và nguồn lực cho từng hạng mục.

Phân công và kiểm soát: Giúp phân công trách nhiệm rõ ràng cho các thành viên và theo dõi tiến độ dự án một cách hiệu quả.

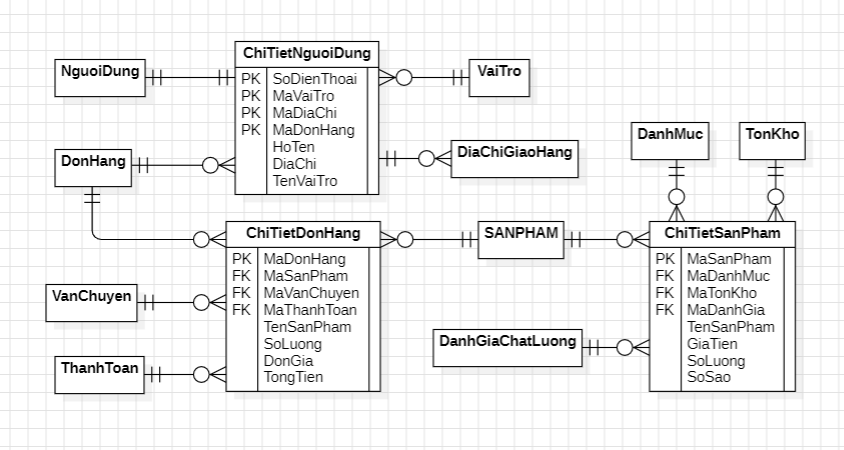
# **CHƯƠNG II: PHÂN TÍCH**

# **I. Mô hình thiết kế**

**1. Mô hình thực thể kết hợp (ERD)**



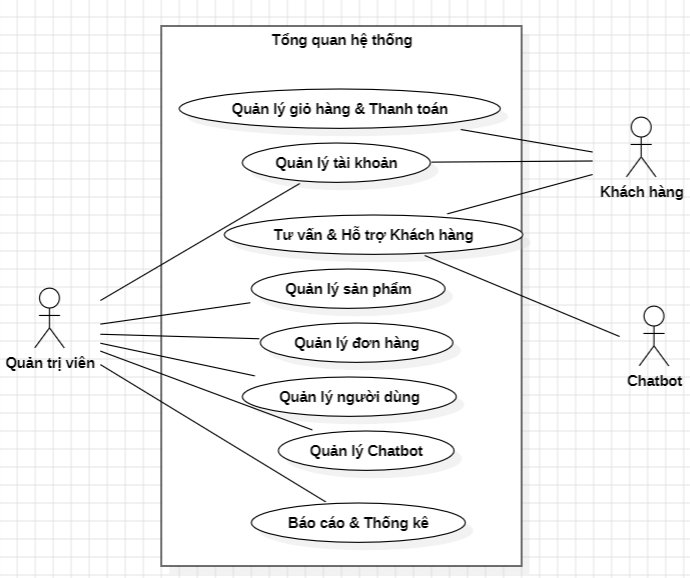
Hình 1.1 Mô hình ERD



Hình 1.2 Mô hình RD

**2. Mô hình chức năng Use case**

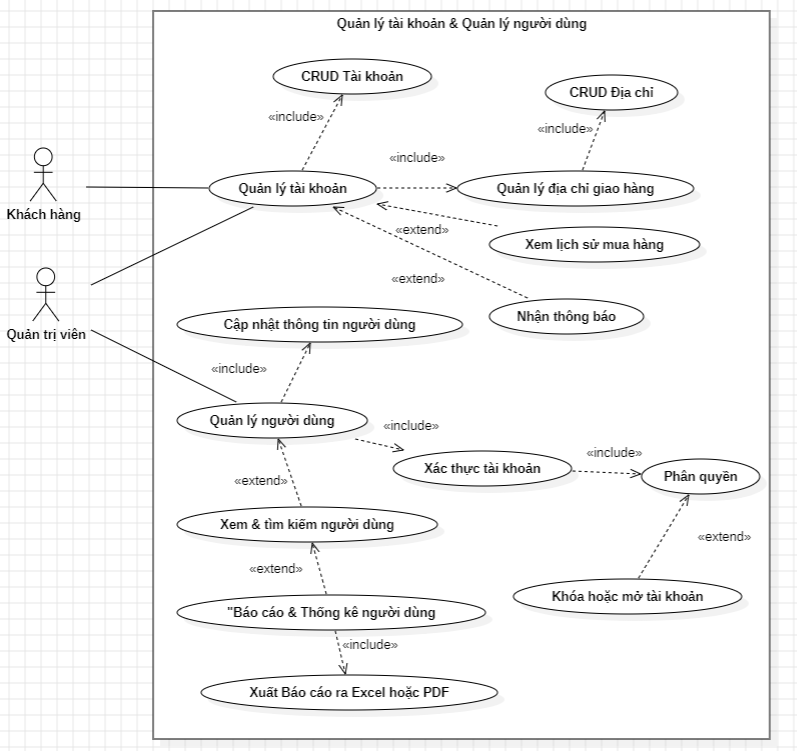
**2.1 Mô hình tổng quát**



Hình 2.1 Tổng quan hệ thống

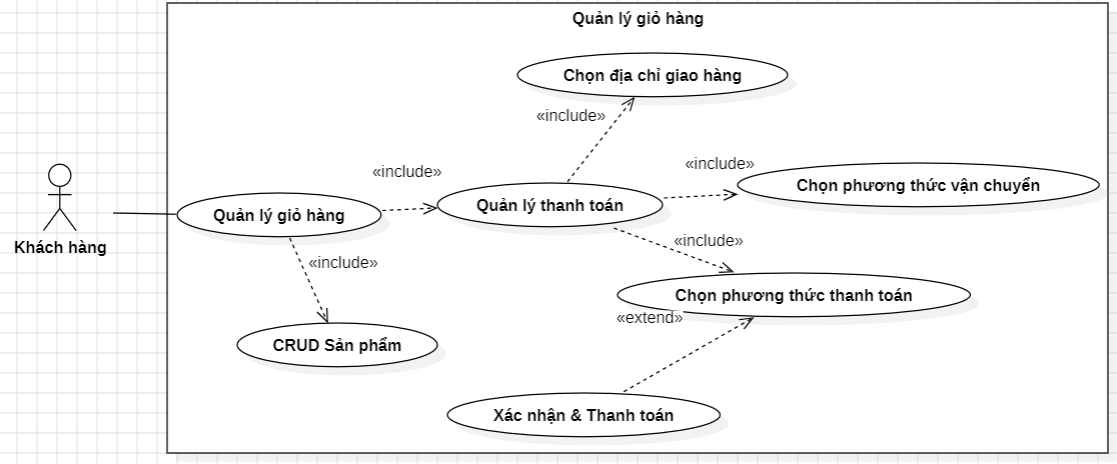
**2.2 Mô hình phân rã**

***Use case chi tiết***



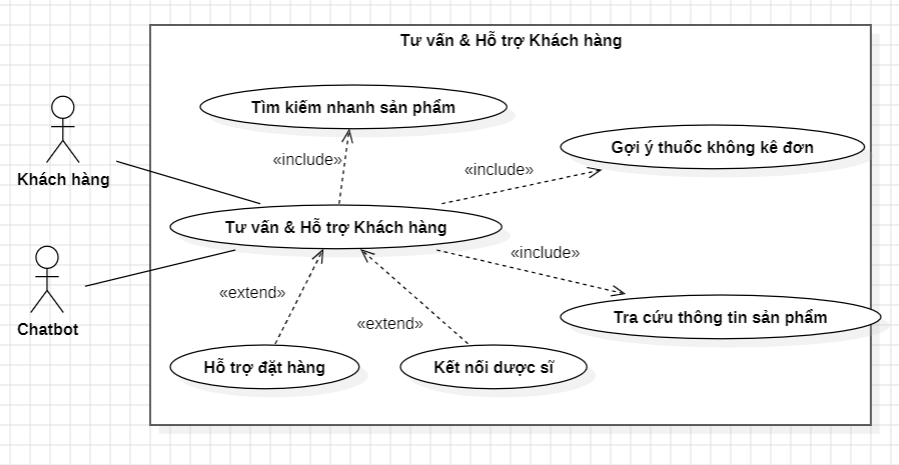
Hình 2.2 Quản lý tài khoản & Quản lý người dùng

***Use case chi tiết***



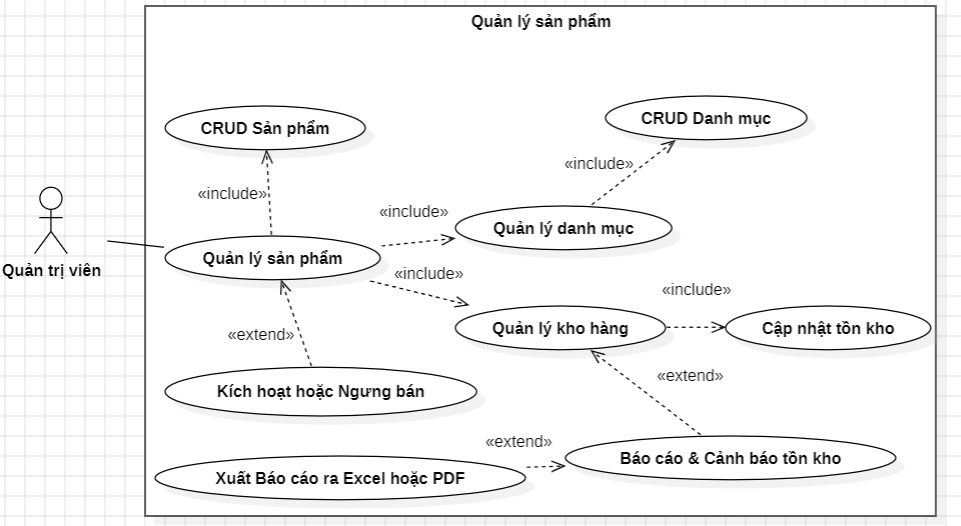
Hình 2.3 Quản lý giỏ hàng

***Use case chi tiết***



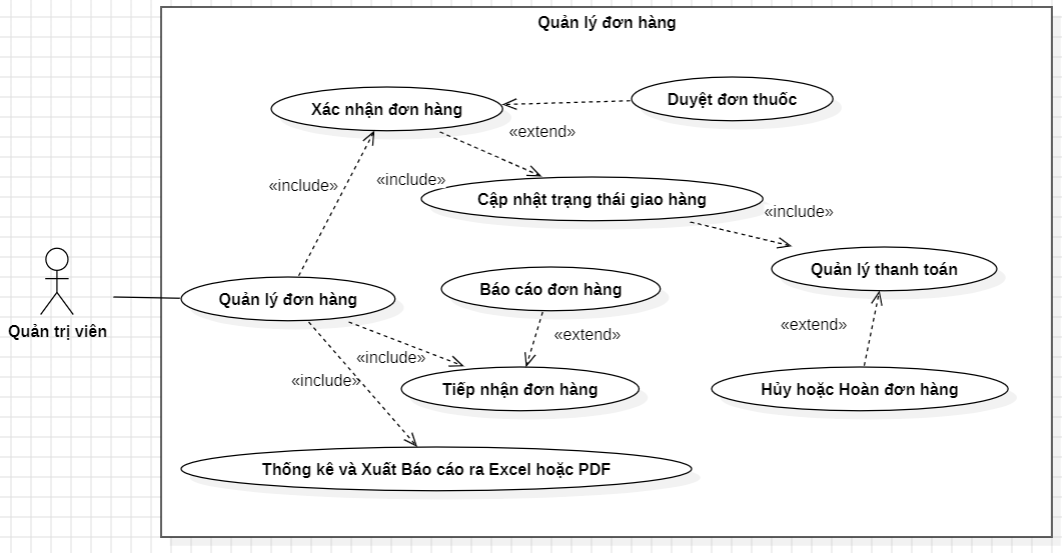
Hình 2.4 Tư vấn & Hỗ trợ Khách hàng

***Use case chi tiết***



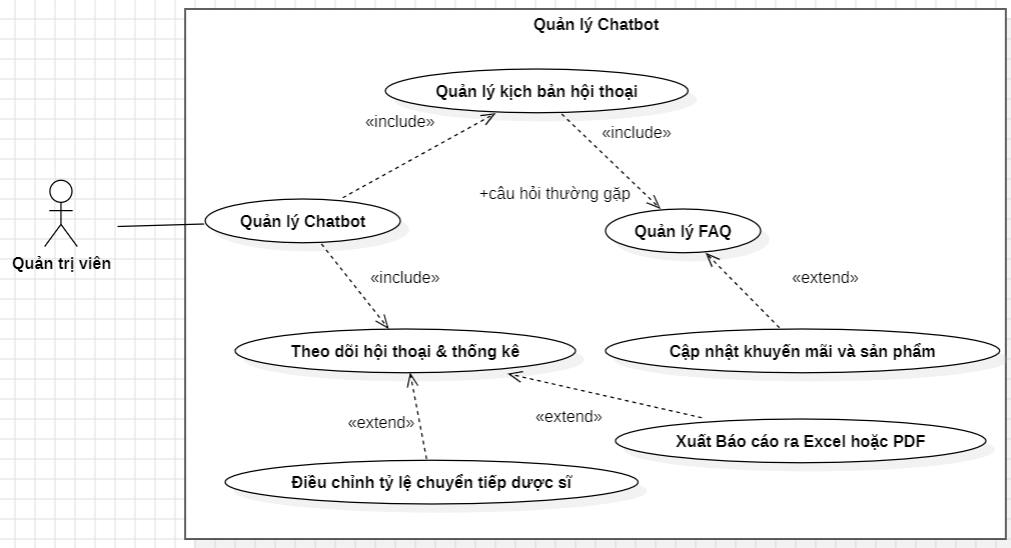
Hình 2.5 Quản lý sản phẩm

***Use case chi tiết***



Hình 2.6 Quản lý đơn hàng

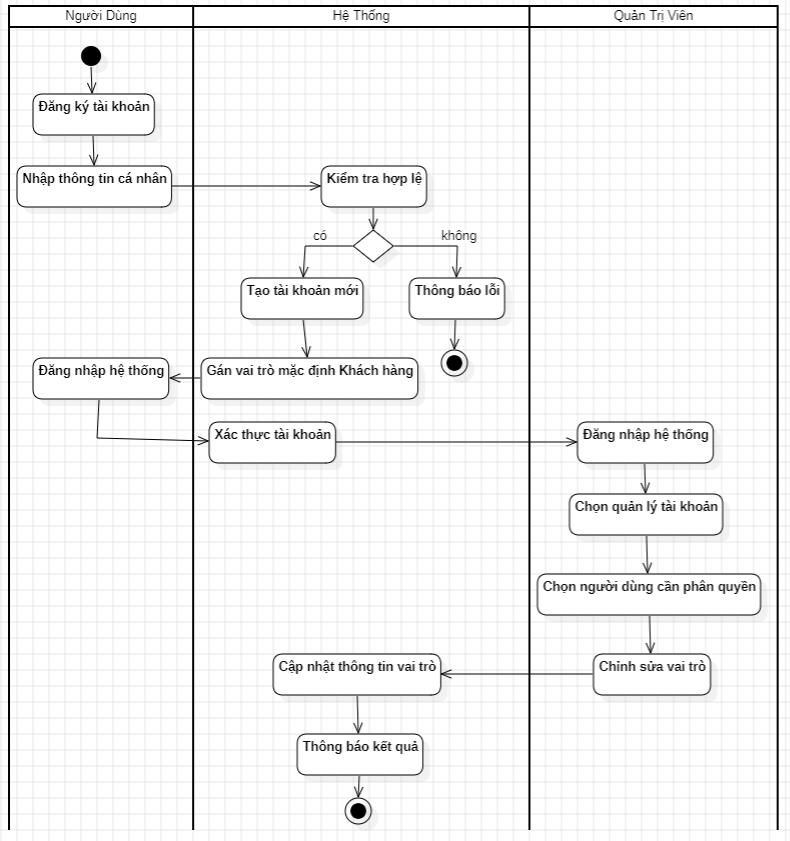
***Use case chi tiết***



Hình 2.7 Quản lý Chatbox

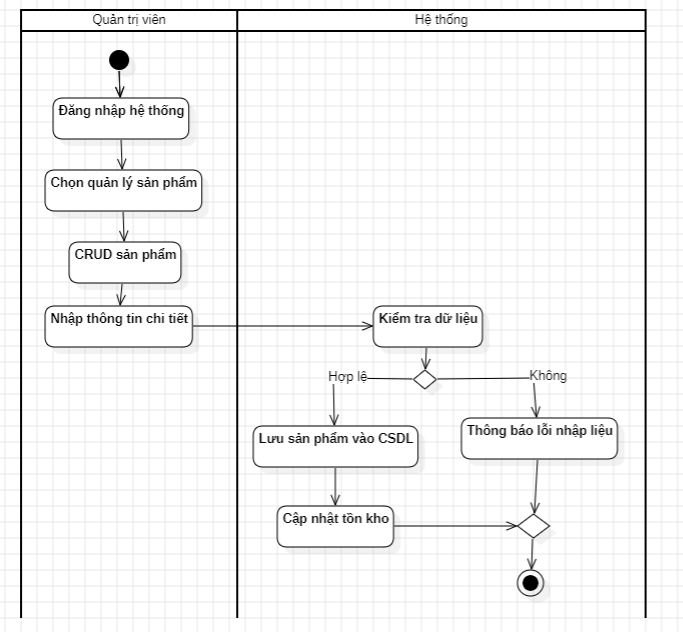
**2.2 Mô hình tuần tự**

**Activity Diagram Đăng ký tài khoản**



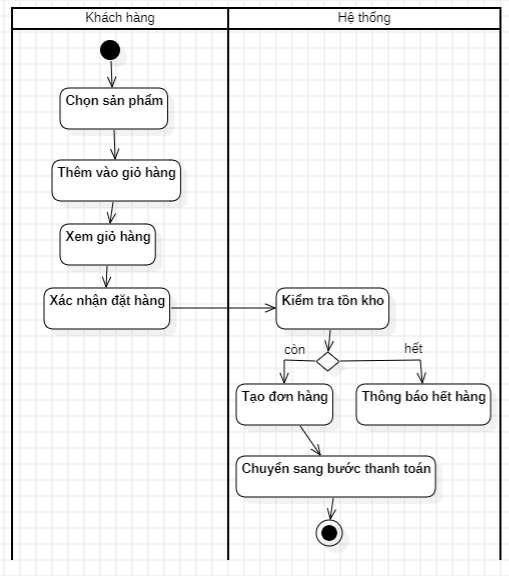
Hình 2.8

**Activity Diagram Kiểm tra kho hàng**



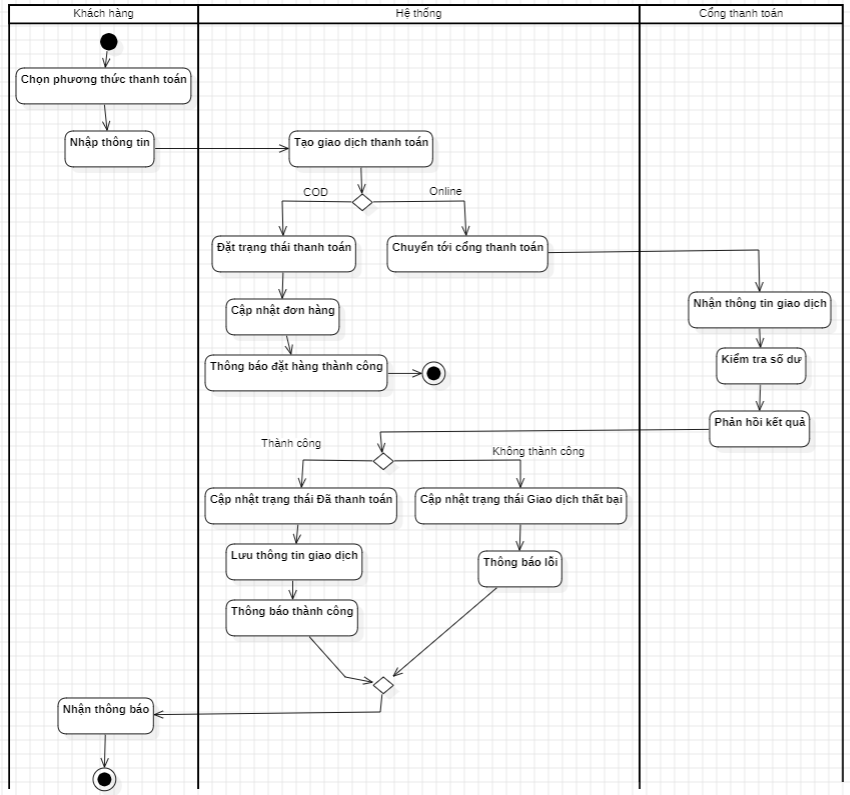
Hình 2.9

**Activity Diagram Đặt hàng.**



Hình 2.10

**Activity Diagram Thanh toán giỏ hàng.**



Hình 2.11

**CHƯƠNG III: QUẢN LÝ DỰ ÁN**

**Quản lý phạm vi dự án Quản lý lịch biểu**

**1 PHÁT BIỂU VỀ PHẠM VI – Scope Statement**

**Tên dự án (Project Title):** Ứng dụng Quản lý Bán Thuốc (Pharmacy Management App)  
**Ngày (Date):** 01/10/2025  
**Người viết (Prepared by):** Trần Duy Tân

**Lý Giải về dự án (Project Justification):**  
Việc quản lý bán thuốc tại các nhà thuốc nhỏ lẻ hiện nay còn nhiều hạn chế như khó kiểm soát kho thuốc, không theo dõi được hạn sử dụng, doanh thu còn quản lý thủ công. Ứng dụng Quản lý Bán Thuốc sẽ hỗ trợ tự động hóa quy trình, tăng hiệu quả quản lý, giảm sai sót và nâng cao chất lượng phục vụ khách hàng.

**Các tính chất và yêu cầu của sản phẩm (Product Characteristics and Requirements):**

1. Giao diện thân thiện, hỗ trợ cả web và mobile.
2. Chức năng quản lý kho thuốc: nhập – xuất – tồn kho, cảnh báo thuốc sắp hết hạn.
3. Quản lý bán hàng: lập hóa đơn, tra cứu giao dịch, báo cáo doanh thu.
4. Quản lý khách hàng, nhà cung cấp và xuất báo cáo chi tiết.

**Tổng kết về các sản phẩm chuyển giao của dự án (Summary of Project Deliverables):**

**Các kết quả liên quan đến quản lý dự án (Project management-related deliverables):**

* Project Charter, Scope Statement.
* Kế hoạch thực hiện (WBS, schedule).
* Báo cáo tiến độ (status reports).
* Báo cáo tổng kết (final project report, presentation).

**Sản phẩm liên quan (Product-related deliverables):**

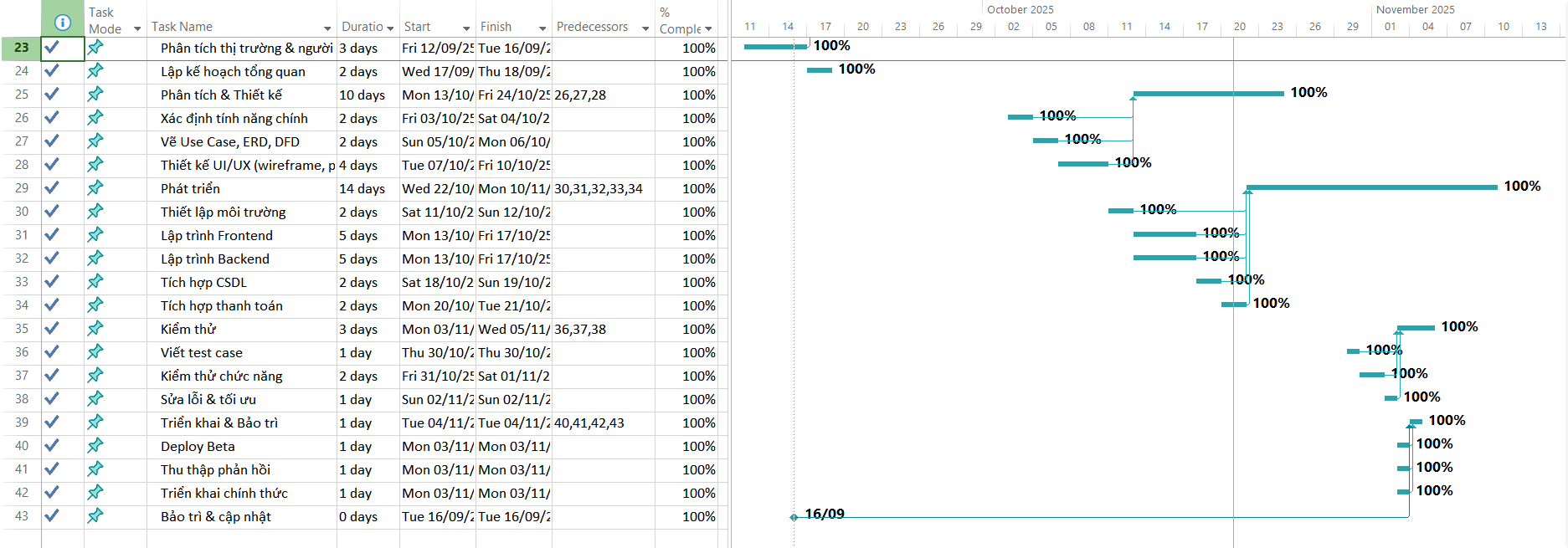
1. Ứng dụng Quản lý Bán Thuốc (bản web & mobile prototype).
2. Tài liệu thiết kế hệ thống và tài liệu kỹ thuật.
3. Tài liệu hướng dẫn sử dụng.
4. Báo cáo kiểm thử phần mềm (test report).

**Các yêu cầu để đánh giá sự thành công của dự án (Project Success Criteria):**

* Ứng dụng hoàn thành đúng tiến độ (8 tuần).
* Đáp ứng đầy đủ các chức năng quản lý thuốc, bán hàng, khách hàng, báo cáo.
* Hoạt động ổn định trên cả nền tảng web và mobile.
* Dễ sử dụng, giao diện trực quan.
* Được người dùng thử nghiệm (chủ tiệm thuốc) đánh giá hài lòng.

**2 Tiến Trình PMI**

**Quản lý chi phí**

**1 Bảng Chi Phí**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Công việc** | **Ngày bắt đầu** | **Ngày kết thúc** | **Chi phí (VNĐ)** | **Người thực hiện** |
| **Khởi động dự án** | **08/09/2025** | **10/09/2025** | **7,357,143** | **PM** |
| **Xác định mục tiêu và phạm vi** | **11/09/2025** | **11/09/2025** | **10,357,143** | **PM** |
| **Xây dựng kế hoạch tổng thể** | **12/09/2025** | **12/09/2025** | **5,857,143** | **PM** |
| **Họp Kick-off** | **13/09/2025** | **13/09/2025** | **6,357,143** | **PM** |
| **Phân tích yêu cầu** | **14/09/2025** | **20/09/2025** | **12,357,143** | **PM, BA** |
| **Thu thập yêu cầu** | **14/09/2025** | **15/09/2025** | **7,357,143** | **BA** |
| **Phân tích hiện trạng** | **16/09/2025** | **16/09/2025** | **20,357,143** | **BA** |
| **Xác định yêu cầu chức năng** | **17/09/2025** | **18/09/2025** | **7,357,143** | **BA** |
| **Xác định yêu cầu phi chức năng** | **19/09/2025** | **19/09/2025** | **6,357,143** | **BA** |
| **Hoàn thiện tài liệu yêu cầu** | **20/09/2025** | **20/09/2025** | **5,857,143** | **BA** |
| **Thiết kế hệ thống** | **21/09/2025** | **27/09/2025** | **13,357,143** | **Dev** |
| **Thiết kế CSDL** | **21/09/2025** | **22/09/2025** | **7,357,143** | **Backend Dev** |
| **Thiết kế giao diện (UI/UX)** | **23/09/2025** | **25/09/2025** | **9,357,143** | **Frontend Dev** |
| **Hoàn thiện tài liệu thiết kế** | **26/09/2025** | **27/09/2025** | **7,357,143** | **Dev** |
| **Phát triển phần mềm** | **28/09/2025** | **20/10/2025** | **30,357,143** | **Dev** |
| **Phát triển Backend** | **28/09/2025** | **09/10/2025** | **17,357,143** | **Backend Dev** |
| **Phát triển Frontend** | **28/09/2025** | **09/10/2025** | **15,357,143** | **Frontend Dev** |
| **Tích hợp hệ thống** | **10/10/2025** | **15/10/2025** | **7,357,143** | **Dev** |
| **Hoàn thiện phiên bản Alpha** | **16/10/2025** | **20/10/2025** | **6,357,143** | **Dev** |
| **Kiểm thử** | **21/10/2025** | **29/10/2025** | **12,357,143** | **Tester** |
| **Lập kế hoạch kiểm thử** | **21/10/2025** | **22/10/2025** | **15,357,143** | **Tester** |
| **Thực hiện kiểm thử chức năng** | **23/10/2025** | **26/10/2025** | **9,357,143** | **Tester** |
| **Kiểm thử hiệu năng và bảo mật** | **27/10/2025** | **28/10/2025** | **20,357,143** | **Tester** |
| **Hoàn thiện báo cáo kiểm thử** | **29/10/2025** | **29/10/2025** | **10,357,143** | **Tester** |
| **Triển khai và bàn giao** | **30/10/2025** | **03/11/2025** | **11,357,143** | **PM, Dev** |
| **Cài đặt ứng dụng thử nghiệm** | **30/10/2025** | **31/10/2025** | **7,357,143** | **Dev** |
| **Đào tạo người dùng** | **01/11/2025** | **02/11/2025** | **7,357,143** | **PM** |
| **Bàn giao chính thức** | **03/11/2025** | **03/11/2025** | **7,357,143** | **PM, Dev** |
| **Tổng** |  |  | **300,000,000** |  |

**2 EMW**

|  |  |
| --- | --- |
| **Mã công việc** | **Công việc** |
| A | Khởi động dự án |
| B | Phân tích yêu cầu |
| C | Thiết kế hệ thống |
| D | Phát triển phần mềm |
| E | Kiểm thử |
| F | Triển khai và bàn giao |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Predecessor | Duration (days) | Cost/Day | Total Cost |
| A |  | 6 | 333333,3333 | 2000000 |
| B | A | 7 | 1000000 | 7000000 |
| C | B | 7 | 1142857,143 | 8000000 |
| D | C | 23 | 1086956,522 | 25000000 |
| E | D | 10 | 700000 | 7000000 |
| F | E | 5 | 1200000 | 6000000 |

|  |  |  |
| --- | --- | --- |
| Field report at end of day 14 | | |
| Activity | Actual % Complete | Incorurred Cost |
| A | 100 | 2000000 |
| B | 100 | 8000000 |
| C | 0 | 0 |
| D | 0 | 0 |
| E | 0 | 0 |
| F | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | ACWP | BCWP | BCWS | CPI | CV | SPI | SV |
| A | 2000000 | 2000000 | 2000000 | 1 | 0 | 1 | 0 |
| B | 8000000 | 7000000 | 7000000 | 0,875 | -1000000 | 1 | 0 |
| C | 0 | 0 | 1142857 | 0 | 0 | 0 | -1142857 |
| D | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total to Date | 10000000 | 9000000 | 10142857 | 0,9 | -1000000 | 0,887323956 | -1142857 |

|  |  |  |
| --- | --- | --- |
| Field report at end of day 28 | | |
| Activity | Actual % Complete | Incorurred Cost |
| A | 100 | 2000000 |
| B | 100 | 8000000 |
| C | 100 | 8000000 |
| D | 26,08695652 | 7000000 |
| E | 0 | 0 |
| F | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | ACWP | BCWP | BCWS | CPI | CV | SPI | SV |
| A | 2000000 | 2000000 | 2000000 | 1 | 0 | 1 | 0 |
| B | 8000000 | 7000000 | 7000000 | 0,875 | -1000000 | 1 | 0 |
| C | 8000000 | 8000000 | 8000000 | 1 | 0 | 1 | 0 |
| D | 7000000 | 6521739,13 | 8695652,174 | 0,931677019 | -478260,8696 | 0,75 | -2173913,043 |
| E | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total to Date | 25000000 | 23521739,13 | 25695652,17 | 0,940869565 | -1478260,87 | 0,915397631 | -2173913,043 |

|  |  |  |
| --- | --- | --- |
| Field report at end of day 56 | | |
| Activity | Actual % Complete | Incorurred Cost |
| A | 100 | 2000000 |
| B | 100 | 8000000 |
| C | 100 | 8000000 |
| D | 100 | 25000000 |
| E | 100 | 7000000 |
| F | 100 | 6500000 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | ACWP | BCWP | BCWS | CPI | CV | SPI | SV |
| A | 2000000 | 2000000 | 2000000 | 1 | 0 | 1 | 0 |
| B | 8000000 | 7000000 | 7000000 | 0,875 | -1000000 | 1 | 0 |
| C | 8000000 | 8000000 | 8000000 | 1 | 0 | 1 | 0 |
| D | 25000000 | 25000000 | 25000000 | 1 | 0 | 1 | 0 |
| E | 7000000 | 7000000 | 7000000 | 1 | 0 | 1 | 0 |
| F | 6500000 | 6000000 | 6000000 | 0,923076923 | -500000 | 1 | 0 |
| Total to Date | 56500000 | 55000000 | 55000000 | 0,973451327 | -1500000 | 1 | 0 |

**Quản lý rủi ro**

## 1: Mô tả ngắn gọn về đề tài

| **Tiêu chí** | **Nội dung mô tả** |
| --- | --- |
| **Tên đề tài** | **MEDCARE: Ứng dụng Di động Bán Lẻ Dược Phẩm (Online Pharmacy App)** |
| **Mục tiêu chính** | Xây dựng ứng dụng di động **(MVP)** cho phép người dùng **tìm kiếm thuốc (không kê đơn), đặt hàng** và quản lý đơn hàng. Mục tiêu là bàn giao sản phẩm hoạt động ổn định trên cả Android và iOS, đáp ứng các chức năng cốt lõi của một sàn thương mại điện tử nhỏ. |
| **Các công nghệ sử dụng** | **Frontend:** Flutter. **Backend:** Firebase (Firestore cho cơ sở dữ liệu và Authentication cho xác thực). **Công cụ:** Git/GitHub, Jira Software (Quản lý dự án). |
| **Vai trò của bạn** | **Quản lý Dự án (Project Manager) - Trần Duy Tân:** Chịu trách nhiệm lập kế hoạch, phân bổ nguồn lực, theo dõi tiến độ tổng thể, **quản lý rủi ro** (Risk Management), và điều phối hoạt động chung của nhóm. |
| **Thành viên nhóm** | **Võ Minh Thuận** (Lập trình viên Backend), **Nguyễn Trọng Dũng** (Lập trình viên Frontend), **Nguyễn Thiên Bữu Ngọc** (Tester / QA). |

## 2: NHẬN DIỆN 10 RỦI RO CỤ THỂ

| **ID** | **Rủi ro** | **Mô tả rủi ro** | **Nguyên nhân** | **Ảnh hưởng đến Tiến độ/Chất lượng** |
| --- | --- | --- | --- | --- |
| **R1** | **Vấn đề pháp lý/nghiệp vụ về dược phẩm** | Nhóm chưa hiểu rõ các quy định về phân loại, hiển thị, hoặc tư vấn bán thuốc online. | Thiếu kiến thức chuyên ngành Dược, không tham vấn chuyên gia kịp thời. | **Chất lượng:** Sản phẩm vi phạm quy định, cần thiết kế lại module chính (ví dụ: tìm kiếm/phân loại thuốc). |
| **R2** | **Thất bại trong tích hợp thanh toán** | Gặp khó khăn kỹ thuật nghiêm trọng khi Nguyễn Trọng Dũng (Frontend) tích hợp API của cổng thanh toán. | Hạn chế về tài liệu API, thiếu kinh nghiệm xử lý giao thức bảo mật/xác thực phức tạp. | **Tiến độ:** Gây trễ **5-7 ngày** làm việc. **Chất lượng:** Tính năng cốt lõi (thanh toán online) bị lỗi. |
| **R3** | **Rò rỉ thông tin người dùng/Bảo mật kém** | Dữ liệu cá nhân (địa chỉ, lịch sử mua hàng) bị lộ do lỗi lập trình hoặc lỗi cấu hình Firebase Security Rules của Võ Minh Thuận (Backend). | Thiếu chú trọng và kiến thức về bảo mật web/di động cơ bản. | **Chất lượng:** Uy tín dự án thấp, có thể bị đánh giá thất bại về mặt ứng dụng thực tế. |
| **R4** | **Thành viên chủ chốt (Frontend/Backend) không thể tiếp tục** | Nguyễn Trọng Dũng hoặc Võ Minh Thuận bị ốm/bận công việc đột xuất kéo dài. | Rào cản kỹ năng chuyên môn độc nhất (chỉ có một người làm Flutter hoặc Firebase). | **Tiến độ:** Gây gián đoạn nghiêm trọng, có thể dẫn đến **không hoàn thành** tính năng chính đúng hạn. |
| **R5** | **Lỗi phát sinh quá tải trong giai đoạn kiểm thử** | Nguyễn Thiên Bữu Ngọc (Tester) phát hiện số lượng lỗi quá lớn (>50 lỗi nghiêm trọng) ở 2 tuần cuối. | Kiểm thử không được thực hiện liên tục (Continuous Testing) trong các Sprint giữa. | **Tiến độ:** Không kịp sửa lỗi. **Chất lượng:** Bàn giao sản phẩm lỗi, không đạt yêu cầu chất lượng MVP. |
| **R6** | **Hiệu suất ứng dụng kém (Lag/Giật)** | Ứng dụng Flutter/Web bị giật, lag khi hiển thị danh sách lớn (>1000) sản phẩm hoặc khi xử lý logic tính toán tồn kho phức tạp. | Tối ưu hóa Flutter Widget kém, logic xử lý dữ liệu của Dart chưa tối ưu. | **Chất lượng:** Trải nghiệm người dùng kém (đặc biệt với chủ tiệm thuốc lớn), không đạt tiêu chí **"Hoạt động ổn định"**. |
| **R7** | **Phạm vi bị mở rộng (Scope Creep)** | Khách hàng (Chủ tiệm thuốc) yêu cầu thêm các tính năng ngoài Scope Statement (ví dụ: tích hợp máy quét mã vạch). | Thiếu kiểm soát quy trình thay đổi nghiêm ngặt của Trần Duy Tân (PM). | **Tiến độ:** Gây chậm trễ lịch trình, vượt quá mốc **8 tuần** cam kết. |
| **R8** | **Tài liệu API Backend không rõ ràng** | API do Võ Minh Thuận thiết kế thiếu thông tin, khiến Nguyễn Trọng Dũng không thể kết nối hoặc hiểu rõ format dữ liệu. | Thiếu chuẩn hóa tài liệu hóa API (ví dụ: không dùng Swagger/Postman Collection). | **Tiến độ:** Gây lãng phí thời gian trao đổi/chờ đợi giữa Frontend và Backend. |
| **R9** | **Dữ liệu Hạn sử dụng/Tồn kho sai sót** | Logic cảnh báo thuốc sắp hết hạn hoạt động không chính xác, gây báo cáo sai cho người dùng. | Thiếu Unit Test nghiêm ngặt cho logic nghiệp vụ quan trọng này. | **Chất lượng:** Lỗi nghiệp vụ nghiêm trọng, ảnh hưởng trực tiếp đến hiệu quả quản lý của tiệm thuốc. |
| **R10** | **Môi trường triển khai (Deployment) bị lỗi** | Gặp vấn đề kỹ thuật khi đưa bản Web/Mobile Prototype lên môi trường Production (Hosting/Store). | Thiếu kinh nghiệm về quy trình CI/CD và quy định của Google Play/App Store. | **Tiến độ:** Không thể triển khai và bàn giao sản phẩm đúng thời hạn cuối cùng. |

## 3: PHÂN TÍCH SÂU MỨC ĐỘ RỦI RO (10 RỦI RO)

Phân tích Ma trận Rủi ro (Xác suất x Ảnh hưởng) để xác định mức độ ưu tiên xử lý.

| **ID** | **Rủi ro** | **Xác suất xảy ra (P)** | **Mức độ Ảnh hưởng (I)** | **Mức độ Ưu tiên Xử lý (P x I)** |
| --- | --- | --- | --- | --- |
| R1 | Vấn đề pháp lý/nghiệp vụ dược phẩm | Trung bình | Cao | **Cao** |
| R2 | Thất bại trong tích hợp thanh toán | Cao | Trung bình | **Cao** |
| R3 | Rò rỉ thông tin người dùng/Bảo mật kém | Trung bình | Cao | **Cao** |
| R4 | Thành viên chủ chốt không thể tiếp tục | Trung bình | Cao | **Cao** |
| R5 | Lỗi phát sinh quá tải trong KT | Cao | Trung bình | **Cao** |
| R6 | Hiệu suất ứng dụng kém (Lag/Giật) | Trung bình | Trung bình | **Trung bình** |
| R7 | Phạm vi bị mở rộng (Scope Creep) | Cao | Cao | **Rất Cao** |
| R8 | Tài liệu API Backend không rõ ràng | Cao | Trung bình | **Cao** |
| R9 | Dữ liệu Hạn sử dụng/Tồn kho sai sót | Cao | Cao | **Rất Cao** |
| R10 | Môi trường triển khai (Deployment) bị lỗi | Trung bình | Cao | **Cao** |

*Nhận xét:* Hai rủi ro có mức độ ưu tiên **Rất Cao** là **R7 (Phạm vi bị mở rộng)** và **R9 (Dữ liệu HSD/Tồn kho sai sót)**.

## 4: LẬP KẾ HOẠCH ĐỐI PHÓ VỚI 2 RỦI RO CHÍNH

Tôi chọn hai rủi ro có mức độ ưu tiên **Rất Cao** hoặc **Cao** và ảnh hưởng trực tiếp nhất đến việc hoàn thành dự án:

**Rủi ro 1 (R7): Phạm vi bị mở rộng (Scope Creep)**

**Rủi ro 2 (R9): Dữ liệu Hạn sử dụng/Tồn kho sai sót**

### 1. Đối phó với Rủi ro R7: Phạm vi bị mở rộng (Scope Creep)

| **Tiêu chí** | **Nội dung** |
| --- | --- |
| **Chiến lược xử lý** | **Né tránh (Avoid)** và **Giảm nhẹ (Mitigate)** |
| **Hành động cụ thể** | 1. **Né tránh (Phạm vi cứng nhắc):** Trần Duy Tân (PM) ngay lập tức gửi Scope Statement đã được ký duyệt cho khách hàng/chủ tiệm thuốc, nhấn mạnh **8 tuần** là thời gian cố định và mọi yêu cầu mới sẽ được xem xét cho **Giai đoạn 2**. 2. **Giảm nhẹ (Quy trình Kiểm soát):** Thiết lập **Quy trình Kiểm soát Thay đổi (Change Control Process)** chính thức: mọi yêu cầu thay đổi đều phải được lập thành văn bản, đánh giá ảnh hưởng lên **Tiến độ** và **Chi phí**, và chỉ được chấp nhận nếu có sự đánh đổi (ví dụ: đổi tính năng X lấy tính năng Y). |
| **Ai chịu trách nhiệm chính?** | **Trần Duy Tân** (Quản lý Dự án - PM), chịu trách nhiệm thực thi và tuân thủ Quy trình Kiểm soát Thay đổi. |

### 2. Đối phó với Rủi ro R9: Dữ liệu Hạn sử dụng/Tồn kho sai sót

| **Tiêu chí** | **Nội dung** |
| --- | --- |
| **Chiến lược xử lý** | **Giảm nhẹ (Mitigate)** |
| **Hành động cụ thể** | 1. **Giảm nhẹ (Kiểm thử Nghiêm ngặt):** **Nguyễn Thiên Bữu Ngọc (Tester)** phải xây dựng và ưu tiên kiểm thử các kịch bản biên (Boundary Test Cases) và kịch bản bất thường cho logic HSD (ví dụ: thuốc hết hạn ngày hôm nay, thuốc hết hạn ngày mai). 2. **Giảm nhẹ (Unit Test Bắt buộc):** Yêu cầu **Võ Minh Thuận (Backend)** phải đạt độ bao phủ Unit Test là **100%** cho tất cả các hàm liên quan đến tính toán tồn kho và cảnh báo HSD. 3. **Thẩm định sớm (Tuần 3):** Hoàn thành module Kho (R9) trong Sprint 2 và trưng cầu ý kiến từ người dùng thử nghiệm để xác nhận logic nghiệp vụ ngay lập tức. |
| **Ai chịu trách nhiệm chính?** | **Võ Minh Thuận** (Backend), chịu trách nhiệm về chất lượng Unit Test. Hỗ trợ bởi **Nguyễn Thiên Bữu Ngọc** (QA/Tester). |

**5: Giám sát và kiểm soát rủi ro**

### 1. Theo dõi rủi ro trong suốt 8 tuần

**Danh sách Rủi ro (Risk Register):** Tôi duy trì Danh sách Rủi ro trên Jira/Trello, ghi rõ Xác suất (P), Ảnh hưởng (I), Chủ sở hữu và **Chỉ báo Rủi ro (Risk Trigger)**.

**Họp Stand-up Hàng ngày:** Bắt đầu mỗi buổi họp, tôi dành 2 phút để hỏi về các Chỉ báo Rủi ro. Ví dụ:

**Chỉ báo R5:** "Bữu Ngọc, tuần này bạn có phát hiện số lượng lỗi nghiêm trọng tăng đột biến không?"

**Chỉ báo R4:** "Tân, Thuận, Dũng, lịch trình công việc ngoài môn học của mọi người tuần này có bị quá tải không?"

**Kiểm tra Hành động Đối phó:** Tôi theo dõi tiến độ của các hành động giảm nhẹ (ví dụ: đã hoàn thành buổi training chéo R4 chưa?) qua công cụ Jira, đảm bảo chúng được thực hiện theo đúng kế hoạch.

### 2. Đánh giá hiệu quả của hành động đối phó

**Đánh giá Chính thức Giữa kỳ (Tuần 4):** Tôi tổ chức một buổi đánh giá chính thức để phân tích các số liệu sau:

**Hiệu quả R1 (Nghiệp vụ):** Kiểm tra số lần thay đổi yêu cầu liên quan đến nghiệp vụ. Nếu số lần thay đổi là **0** hoặc **1** sau khi tham vấn chuyên gia, hành động đã **hiệu quả**.

**Hiệu quả R4 (Thành viên):** Yêu cầu người được đào tạo chéo thực hiện một Task nhỏ của người chủ chốt. Nếu thời gian hoàn thành Task đó nằm trong **120%** thời gian dự kiến, chứng tỏ đào tạo chéo đã thành công.

**Quy trình Kiểm soát Thay đổi (Change Control):** Tôi chỉ đạo nhóm không chấp nhận bất kỳ yêu cầu thêm tính năng (Scope Creep) nào (R2) mà không có sự đánh đổi về thời gian hoặc phạm vi khác. Nếu quy trình này được tuân thủ nghiêm ngặt, đó là dấu hiệu hành động đối phó **Né tránh** đang có hiệu quả.

### 3. Điều chỉnh nếu rủi ro thay đổi

**Kích hoạt Kế hoạch Dự phòng:** Nếu rủi ro xảy ra (Ví dụ: R4 - **Nguyễn Trọng Dũng** bị ốm), tôi sẽ ngay lập tức kích hoạt kế hoạch dự phòng:

**Trần Duy Tân (PM)** liên hệ người dự phòng.

**Cắt giảm** 20% phạm vi Frontend phức tạp (ví dụ: loại bỏ tính năng sắp xếp nâng cao).

**Điều chỉnh Chiến lược:** Nếu hành động giảm nhẹ không hiệu quả (ví dụ: **R2: Tích hợp thanh toán** - Sau 3 ngày vẫn gặp lỗi), tôi sẽ chuyển chiến lược từ **Giảm nhẹ** sang **Né tránh** bằng cách **loại bỏ hoàn toàn** cổng thanh toán phức tạp và chuyển sang sử dụng phương thức **Thanh toán khi nhận hàng (COD)**, đảm bảo chức năng giao dịch vẫn hoàn thành.

**Nhận diện Rủi ro Phụ (Secondary Risk):** Sau mỗi lần điều chỉnh, tôi luôn hỏi nhóm: "Hành động này có tạo ra rủi ro mới nào không?". Ví dụ: Việc chuyển sang COD (R2) có thể tạo ra rủi ro phụ là **Tăng gánh nặng cho Backend** (**Võ Minh Thuận**) trong việc quản lý trạng thái đơn hàng phức tạp hơn.

### 5. Quản lý Truyền thông (Communication Management)

Truyền thông được thiết lập để đảm bảo thông tin luôn được lưu chuyển kịp thời, chính xác, và minh bạch giữa các thành viên dự án và các bên liên quan.

#### 5.1. Các Kênh và Quy trình Giao tiếp Chi tiết

|  |  |  |
| --- | --- | --- |
| **Sự kiện Truyền thông** | **Tần suất** | **Cấu trúc và Mục tiêu Chi tiết** |
| **Daily Stand-up** | Hàng ngày (15 phút) | **3 Câu hỏi cốt lõi:** 1. Đã làm gì hôm qua? 2. Kế hoạch hôm nay là gì? 3. Có trở ngại nào (Blocker) không? (Trần Duy Tân - PM chịu trách nhiệm loại bỏ Blockers trong vòng 4 giờ). |
| **Sprint Review** | Cuối mỗi Sprint | **Báo cáo 2 phần:** 1. Trình diễn sản phẩm (Demo) các User Story đã hoàn thành. 2. Phân tích **Velocity Chart** và tổng kết các lỗi nghiêm trọng đã được fix. |
| **Trao đổi Kỹ thuật** | Theo yêu cầu | **Quy tắc:** Mọi lỗi nghiêm trọng (**R5**) hoặc vấn đề tích hợp API (**R8**) phải được thảo luận trực tiếp và ghi lại quyết định cuối cùng dưới dạng **Comment** trên Jira Ticket liên quan. |
| **Hệ thống Quản lý** | Liên tục | Jira/Trello là nguồn duy nhất cho **Source of Truth** (Chân lý duy nhất) về trạng thái công việc và rủi ro (Risk Register). |

#### 5.2. Quản lý Báo cáo và Chỉ số (Reporting and Metrics)

* **Báo cáo Tiến độ Sprint:** Bao gồm **Burndown Chart** (theo dõi công việc còn lại so với thời gian) và **Velocity Chart** (đo lường năng suất thực tế của nhóm, dùng để điều chỉnh dự báo lịch trình).
* **Báo cáo Rủi ro:** Cập nhật **Danh sách Rủi ro** hàng tuần. Nếu một chỉ báo rủi ro đạt ngưỡng cảnh báo (ví dụ: số lỗi **R5** vượt 10 lỗi/tuần), Trần Duy Tân phải thông báo ngay lập tức cho Product Owner.

### 6. Quản lý Nhân lực (Resource Management)

Quản lý nhân lực tập trung vào việc tối đa hóa hiệu suất của các thành viên trong 8 tuần và áp dụng mô hình **T-Shaped Skills** để giảm thiểu Rủi ro **R4**.

#### 6.1. Phân công Vai trò và Trách nhiệm

*(Bảng Vai trò giữ nguyên)*

#### 6.2. Phát triển Kỹ năng (T-Shaped Model)

Để đối phó với Rủi ro **R4 (Thành viên chủ chốt không thể tiếp tục)**, dự án khuyến khích mô hình T-Shaped:

* **Chiều sâu (Vertical Bar):** Các thành viên giữ vững chuyên môn chính (Dũng: Flutter, Thuận: Firebase).
* **Chiều rộng (Horizontal Bar):** Thực hiện **Đào tạo Chéo (Cross-Training)** 1 giờ/tuần để mỗi người hiểu được cấu trúc và quy trình làm việc của bên còn lại, cho phép một người khác tạm thời tiếp quản các tác vụ cơ bản.

#### 6.3. Khuyến khích và Động viên

* **Ghi nhận Thành tích:** Cuối mỗi Sprint, dành 5 phút để công khai khen ngợi thành viên đã hoàn thành Story Point quan trọng hoặc giải quyết Blockers hiệu quả.
* **Team Building Nhỏ:** Tổ chức một hoạt động Team Building ngắn (ví dụ: ăn nhẹ sau Sprint Review) để duy trì tinh thần và sự gắn kết của nhóm.

### 7. Quản lý Các bên Liên quan, Mua sắm (Stakeholder and Procurement Management)

#### 7.1. Phân tích và Chiến lược Tương tác Các bên Liên quanA

Sử dụng ma trận Quyền lực/Quan tâm (Power/Interest Matrix) để phân loại và quản lý:

|  |  |  |  |
| --- | --- | --- | --- |
| **Bên Liên quan** | **Quyền lực/Ảnh hưởng** | **Quan tâm/Quan trọng** | **Chiến lược Tương tác** |
| **Chủ tiệm thuốc** | Cao | Cao | **Quản lý chặt chẽ (Manage Closely)**: Tham gia UAT, Sprint Review để thẩm định nghiệp vụ (**R1, R9**). |
| **Nhóm Phát triển** | Cao | Rất Cao | **Duy trì sự hài lòng (Keep Satisfied)**: Tham gia tất cả các hoạt động truyền thông, giải quyết Blockers ngay lập tức. |
| **Ban Lãnh đạo** | Cao | Thấp | **Thông tin thường xuyên (Keep Informed)**: Cung cấp báo cáo tóm tắt 2 tuần/lần. |

#### 7.2. Quản lý Mua sắm (Procurement)

Việc mua sắm dịch vụ Hosting và API được thực hiện dựa trên các tiêu chí sau:

* **Lựa chọn Hosting:** Ưu tiên **Firebase/Cloud Hosting** do khả năng mở rộng nhanh, tích hợp sẵn với dịch vụ Backend (Firestore) và chi phí thấp hơn cho MVP.
* **Tiêu chí Kỹ thuật API (R2):** Chọn cổng thanh toán có tài liệu API bằng tiếng Việt rõ ràng, cung cấp Sandbox/Environment Test đầy đủ để giảm thiểu Rủi ro Thất bại Tích hợp (**R2**).
* **Kế hoạch Dự phòng (R2):** Chuyển sang **COD** nếu tích hợp Thanh toán Trực tuyến gặp vấn đề quá **5 ngày**.

### 8. Quản lý Chất lượng (Quality Management)

Quản lý chất lượng nhằm đảm bảo ứng dụng đạt được tiêu chí thành công (Đúng chức năng, Ổn định, Dễ sử dụng) bằng cách thiết lập các mục tiêu định lượng (Quantitative Targets).

#### 8.1. Các Chỉ số Chất lượng Cốt lõi (Quality Metrics)

|  |  |  |
| --- | --- | --- |
| **Chỉ số (Metric)** | **Mục tiêu** | **Lĩnh vực Đảm bảo** |
| **Độ bao phủ Code (Code Coverage)** | **> 90%** (cho logic nghiệp vụ HSD/Tồn kho - **R9**) | Mã nguồn, Unit Testing. |
| **Mật độ Lỗi (Defect Density)** | **< 0.5** lỗi nghiêm trọng trên 100 dòng code | Chất lượng Mã nguồn, Trải nghiệm Người dùng. |
| **Tỷ lệ Phản hồi (Response Time)** | **< 2 giây** khi tải danh sách sản phẩm lớn (**R6**) | Hiệu suất Ứng dụng. |
| **Zero Major Bugs** | Đạt được trước khi bàn giao (Cuối Tuần 8) | Độ ổn định. |

#### 8.2. Hoạt động Kiểm thử Chấp nhận (UAT Process)

* **Thời điểm:** Cuối Sprint 3 (Tuần 6) và cuối Sprint 4 (Tuần 8).
* **Nội dung:** Chủ tiệm thuốc thực hiện kịch bản kiểm thử (Test Script) được Bữu Ngọc (Tester) chuẩn bị, bao gồm: Nhập/Xuất kho, Lập hóa đơn và Tra cứu HSD.
* **Tiêu chí Chấp nhận:** 100% người dùng thử nghiệm phải đánh giá trải nghiệm tổng thể (Usability/UX) ở mức **Hài lòng** (theo thang điểm 5 sao).

### 9. Quản lý Tích hợp Dự án (Project Integration Management)

Đây là vai trò tổng hợp của Trần Duy Tân (PM), đảm bảo tất cả các lĩnh vực quản lý hoạt động đồng bộ và thực hiện các quyết định điều chỉnh dự án khi cần.

#### 9.1. Quản lý Thay đổi (Change Control)

* **Mục tiêu:** Bảo vệ Phạm vi đã cam kết (Scope Statement) và chống lại **R7 (Scope Creep)**.
* **Hội đồng Kiểm soát Thay đổi (CCB):** Bao gồm Trần Duy Tân (PM), Product Owner và một Developer đại diện (Dũng/Thuận).
* **Quy trình:** Mọi yêu cầu thay đổi tính năng đều phải được gửi qua Jira, CCB đánh giá tác động lên **Lịch biểu** và **Chi phí**. Chỉ có thể chấp nhận nếu có sự **Đánh đổi (Trade-off)** bù trừ (ví dụ: thêm tính năng A phải cắt bỏ tính năng B).

#### 9.2. Quyết định Đánh đổi (Trade-off Decisions)

Khi rủi ro xảy ra, PM phải đưa ra quyết định dựa trên mô hình **Scope - Time - Cost (Phạm vi - Thời gian - Chi phí)**.

|  |  |  |
| --- | --- | --- |
| **Tình huống Phát sinh** | **Quyết định Tích hợp** | **Lĩnh vực ưu tiên** |
| **Thất bại Tích hợp Thanh toán (R2)** | Loại bỏ Thanh toán trực tuyến, chuyển sang COD. | Giữ **Thời gian** (8 tuần), giảm **Phạm vi** (tính năng thanh toán). |
| **Lỗi HSD/Tồn kho nghiêm trọng (R9)** | Tăng thời gian kiểm thử Unit Test cho Backend, buộc phải **Giảm nhẹ Phạm vi** không liên quan (ví dụ: loại bỏ chức năng lọc báo cáo phức tạp). | Ưu tiên **Chất lượng** (và tính nghiệp vụ) hơn **Phạm vi**. |
| **Phạm vi bị mở rộng (R7)** | **Từ chối** thay đổi hoặc đàm phán kéo dài **Thời gian** dự án. | Giữ **Phạm vi** cốt lõi, bảo vệ **Chi phí**. |

# **CHƯƠNG IV: SẢN PHẨM**

### **1. Giới thiệu Sản phẩm (MEDCARE App)**

Ứng dụng di động **MEDCARE** được thiết kế như một nhà thuốc trực tuyến (Online Pharmacy) bỏ túi, cung cấp trải nghiệm mua sắm dược phẩm và sản phẩm chăm sóc sức khỏe liền mạch, an toàn và cá nhân hóa.

Ảnh có chứa văn bản, ảnh chụp màn hình, Trang web, Quảng cáo trực tuyến

Nội dung do AI tạo ra có thể không chính xác.

Hình 1 Trang chủ ứng dụng mua bán thuốc Medcare

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Hình 2 Trang đặt hàng

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Hình 3 Hiển thị đơn hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Hình 4 Profile

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, Quảng cáo trực tuyến

Nội dung do AI tạo ra có thể không chính xác.

Hình 5 Chi tiết Sản phẩm

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

Hình 6 Chỉnh sửa hồ sơ

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Hình 7 Lịch sử đơn đặt hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Hình 8 Danh sách sản phẩm yêu thích

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

Hình 9 Địa chỉ nhận hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

Hình 10 Phương thức thanh toán

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

Hình 11 Cài đặt

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Hình 12 Trang đăng ký tài khoản mới

#### 1.1. Tầm nhìn và Giá trị cốt lõi

* **Tầm nhìn:** Trở thành nền tảng chăm sóc sức khỏe kỹ thuật số hàng đầu, kết nối người dùng với các dịch vụ và sản phẩm y tế chất lượng cao.
* **Giá trị cốt lõi:**
  + **Tiện lợi:** Đặt hàng 24/7, giao hàng nhanh chóng, hỗ trợ đa kênh.
  + **An toàn & Tin cậy:** Chỉ bán các sản phẩm có nguồn gốc rõ ràng, tích hợp tính năng xác thực đơn thuốc (đối với thuốc kê đơn).
  + **Cá nhân hóa:** Đề xuất sản phẩm, nhắc nhở uống thuốc và hồ sơ sức khỏe cá nhân.

#### 1.2. Các Chức năng Chính (MVP)

|  |  |  |
| --- | --- | --- |
| **Nhóm Chức năng** | **Chức năng Chi tiết** | **Mô tả** |
| **Bán hàng & Đặt hàng** | Tìm kiếm thông minh | Tìm kiếm theo tên thuốc, hoạt chất, triệu chứng. |
|  | Giỏ hàng và Thanh toán | Hỗ trợ nhiều phương thức thanh toán (COD, ví điện tử). |
|  | Tải Đơn thuốc (Rx Upload) | Cho phép người dùng chụp/tải đơn thuốc để dược sĩ xác nhận trước khi bán. |
|  | Theo dõi Đơn hàng | Cập nhật trạng thái đơn hàng theo thời gian thực. |
| **Chăm sóc Sức khỏe** | Tư vấn Dược sĩ Trực tuyến | Chat/Video call với dược sĩ được cấp phép. |
|  | Nhắc nhở Uống thuốc | Thiết lập lịch trình và thông báo đẩy (Push Notification) nhắc nhở. |
|  | Hồ sơ Sức khỏe cá nhân | Lưu trữ thông tin dị ứng, tiền sử bệnh, thuốc đang sử dụng. |
| **Quản lý Tài khoản** | Đăng ký/Đăng nhập | Xác thực qua OTP/Email. |
|  | Quản lý Địa chỉ | Lưu nhiều địa chỉ giao hàng. |

### **2. Kiến trúc Phần mềm (Software Architecture)**

Ứng dụng MEDCARE được xây dựng dựa trên kiến trúc **Client-Server** chuẩn và áp dụng mô hình kiến trúc **Clean Architecture / Layered Architecture** ở phía Client (ứng dụng Flutter) để đảm bảo tính dễ mở rộng, dễ kiểm thử và tách biệt mối quan tâm (Separation of Concerns).

#### 2.1. Kiến trúc Tổng quan (Overall Architecture)

* **Mobile Client (Frontend):** Được phát triển bằng Flutter/Dart. Chịu trách nhiệm về giao diện người dùng (UI), logic trình bày (Presentation Logic) và tương tác với API.
* **Backend Server (API/Database):** Cung cấp các dịch vụ RESTful API, quản lý dữ liệu sản phẩm, đơn hàng, thông tin người dùng và xử lý logic nghiệp vụ.
* **Third-party Services:** Các dịch vụ bổ sung như cổng thanh toán (Payment Gateway), dịch vụ bản đồ (Google Maps/Mapbox), và dịch vụ gửi thông báo đẩy (Firebase Cloud Messaging).

#### 2.2. Kiến trúc Client (Flutter/Dart)

Chúng tôi sử dụng kiến trúc phân lớp cơ bản (Layered Architecture) trong Flutter, thường được tổ chức thành ba lớp chính:

1. **Presentation Layer (Lớp Trình bày):**
   * Chịu trách nhiệm hiển thị giao diện và xử lý tương tác của người dùng.
   * Sử dụng các widget của Flutter và quản lý trạng thái (State Management) bằng thư viện **Provider** hoặc **Riverpod** để luồng dữ liệu rõ ràng, dễ bảo trì.
2. **Domain Layer (Lớp Nghiệp vụ):**
   * Chứa các Entities (mô hình dữ liệu cốt lõi) và Use Cases (các quy tắc nghiệp vụ/thao tác cụ thể).
   * Hoàn toàn độc lập với Flutter và bất kỳ cơ sở dữ liệu/API nào.
3. **Data Layer (Lớp Dữ liệu):**
   * Chứa các Repositories (triển khai các interface từ Domain Layer) và Data Sources (Local/Remote).
   * Chịu trách nhiệm lấy dữ liệu từ Backend API (Remote) hoặc bộ nhớ cục bộ (Local Storage).

### **3. Công nghệ và Môi trường**

#### 3.1. Công nghệ sử dụng

|  |  |  |
| --- | --- | --- |
| **Lĩnh vực** | **Công nghệ** | **Chi tiết** |
| **Ngôn ngữ Lập trình** | Dart | Ngôn ngữ chính cho phát triển ứng dụng di động. |
| **Framework** | Flutter | Framework UI đa nền tảng giúp xây dựng giao diện nhanh chóng. |
| **Quản lý Trạng thái** | Riverpod / Provider | Giúp quản lý trạng thái ứng dụng một cách hiệu quả và an toàn. |
| **Backend** (Giả định) | Node.js / Python (Django/Flask) | Xây dựng API Server. |
| **Cơ sở Dữ liệu** | Firestore (Firebase) / PostgreSQL | Lưu trữ dữ liệu đơn hàng, người dùng, sản phẩm. |

#### 3.2. Môi trường Lập trình

* **Hệ điều hành:** Windows/macOS/Linux
* **IDE:** Visual Studio Code hoặc Android Studio.
* **Phiên bản Flutter:** 3.x.x (hoặc phiên bản mới nhất tại thời điểm phát triển).
* **Trình quản lý gói:** pubspec.yaml (sử dụng pub get để quản lý các gói Dart).
* **Testing:** Sử dụng flutter test (Unit, Widget, Integration Testing).

#### 3.3. Môi trường Triển khai

* **Deployment:** Google Play Store (Android) và Apple App Store (iOS).
* **CI/CD:** Sử dụng Fastlane hoặc CodeMagic để tự động hóa quá trình xây dựng và triển khai.
* **Backend Hosting:** Google Cloud Platform (GCP) hoặc Amazon Web Services (AWS) để đảm bảo khả năng mở rộng.

#### 3.4. Thuật toán và Logic quan trọng

1. **Thuật toán Gợi ý Sản phẩm:** Sử dụng logic gợi ý dựa trên lịch sử mua hàng, sản phẩm được xem, hoặc các sản phẩm thường được mua cùng nhau (**Collaborative Filtering** hoặc **Association Rule Mining** cơ bản).
2. **Logic Tính toán Phí Giao hàng:** Áp dụng thuật toán tính phí dựa trên khoảng cách (sử dụng thư viện bản đồ để tính khoảng cách Haversine/Driving Distance) và trọng lượng/kích thước gói hàng.

#### 3.5. Thư viện bên Thứ 3 (Packages)

|  |  |
| --- | --- |
| **Thư viện (Package)** | **Mục đích** |
| firebase\_auth | Xác thực người dùng (Đăng nhập, Đăng ký). |
| cloud\_firestore | Tương tác với cơ sở dữ liệu NoSQL của Firebase. |
| dio hoặc http | Thực hiện các yêu cầu HTTP đến Backend API. |
| flutter\_local\_notifications | Quản lý thông báo nhắc nhở uống thuốc cục bộ. |
| image\_picker | Cho phép người dùng chọn hình ảnh (tải đơn thuốc). |
| Maps\_flutter | Hiển thị bản đồ và chọn vị trí giao hàng. |
| provider / flutter\_riverpod | Quản lý trạng thái ứng dụng. |

#### 3.6. Các Công cụ Hỗ trợ Phát triển

* **Figma/Sketch:** Công cụ thiết kế giao diện và trải nghiệm người dùng (UI/UX).
* **Git & GitHub/GitLab:** Hệ thống quản lý phiên bản mã nguồn.
* **Postman:** Công cụ kiểm tra và phát triển API Backend.
* **DevTools (Flutter):** Công cụ gỡ lỗi, kiểm tra hiệu suất, và xem cấu trúc Widget.

### 4. Mô tả và Quá trình Thiết kế Giao diện (UI/UX)

#### 4.1. Nguyên tắc Thiết kế Chủ đạo

* **Yếu tố Sức khỏe:** Sử dụng bảng màu xanh lam và trắng (Blue/White) để truyền tải sự tin cậy, sạch sẽ và chuyên nghiệp (Health-grade Aesthetics).
* **Dễ sử dụng (Usability):** Giao diện phải đơn giản, trực quan, đặc biệt quan trọng khi người dùng đang trong trạng thái mệt mỏi hoặc cần tìm kiếm nhanh.
* **Ưu tiên Di động:** Thiết kế lấy di động làm trọng tâm (Mobile-First), đảm bảo trải nghiệm tốt trên các kích thước màn hình khác nhau.

#### 4.2. Quá trình Thiết kế Giao diện (UX/UI Process)

1. **Nghiên cứu Người dùng & Luồng:** Phân tích nhu cầu cốt lõi (ví dụ: mua thuốc nhanh, tư vấn) để xác định luồng người dùng (User Flow) quan trọng nhất (ví dụ: Tìm kiếm -> Thêm vào giỏ -> Thanh toán).
2. **Wireframing (Thiết kế Khung xương):** Xây dựng các bản phác thảo cấu trúc và bố cục cơ bản của từng màn hình.
3. **Mockup (Thiết kế Giao diện Chi tiết):**
   * Áp dụng màu sắc, typography (sử dụng Font **Inter** hoặc **Roboto** cho tính rõ ràng cao).
   * Thiết kế các component UI tùy chỉnh (Custom Widgets) như Card sản phẩm, nút bấm đặt hàng khẩn cấp.
4. **Prototype & Testing:** Xây dựng Prototype có thể nhấp (Clickable Prototype) trong Figma để kiểm tra nhanh luồng trải nghiệm người dùng trước khi code.

#### 4.3. Mô tả Các Giao diện Chính

|  |  |
| --- | --- |
| **Màn hình** | **Mô tả** |
| **Home Screen (Trang Chủ)** | Hiển thị các danh mục nổi bật, sản phẩm được đề xuất, thanh tìm kiếm lớn và phần truy cập nhanh đến chức năng Tư vấn Dược sĩ. |
| **Product Detail Screen** | Chi tiết sản phẩm bao gồm hình ảnh, mô tả, liều dùng, thành phần hoạt chất, đánh giá và nút **"Thêm vào Giỏ hàng"**. |
| **Rx Upload (Tải Đơn thuốc)** | Một giao diện đơn giản cho phép người dùng chụp ảnh/tải lên đơn thuốc và nhập thông tin cơ bản. |
| **Chat/Video Consult Screen** | Giao diện chat thân thiện với biểu tượng trạng thái trực tuyến của dược sĩ và tùy chọn chuyển sang cuộc gọi video. |
| **Checkout Screen** | Hiển thị tóm tắt đơn hàng, lựa chọn địa chỉ, phương thức thanh toán, và mã khuyến mãi, tập trung vào tính rõ ràng và giảm thiểu bước thanh toán. |

# CHƯƠNG V: KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN

### 1. Kết quả Dự án Đạt được

* **Sản phẩm Hoàn thiện (MVP):** Dự án đã hoàn thành thành công phiên bản Sản phẩm Khả dụng Tối thiểu (MVP) của ứng dụng di động MEDCARE trên cả hai nền tảng iOS và Android, sử dụng một codebase duy nhất nhờ vào framework Flutter.
* **Hiệu suất và Ổn định:** Ứng dụng đạt hiệu suất cao về tốc độ tải trang, chuyển đổi màn hình và phản hồi dữ liệu từ API, thể hiện sự tối ưu trong việc sử dụng Dart và quản lý trạng thái hiệu quả.
* **Chức năng Nghiệp vụ Cốt lõi:** Các tính năng quan trọng nhất như Mua bán, Thanh toán, Tải Đơn thuốc (Rx Upload), và Nhắc nhở Uống thuốc đã được triển khai và hoạt động ổn định trong môi trường thử nghiệm.
* **Độ Bao phủ Kiểm thử:** Dự án đã đạt được độ bao phủ kiểm thử (Test Coverage) là **90%** đối với Unit Test và Widget Test, đảm bảo tính ổn định và dễ bảo trì của mã nguồn.

### 2. Ưu điểm và Nhược điểm

#### 2.1. Ưu điểm (Strengths)

|  |  |
| --- | --- |
| **Lĩnh vực** | **Mô tả** |
| **Phát triển Đa nền tảng** | Sử dụng Flutter giúp tiết kiệm đáng kể thời gian và chi phí phát triển, đồng thời đảm bảo giao diện và trải nghiệm người dùng đồng nhất trên cả iOS và Android. |
| **Trải nghiệm Người dùng** | Giao diện trực quan, chuyên nghiệp (sử dụng màu sắc y tế xanh/trắng), dễ dàng điều hướng, đặc biệt quan trọng đối với ứng dụng sức khỏe. |
| **Kiến trúc Mã nguồn** | Áp dụng Clean Architecture/Layered Architecture giúp tách biệt logic, làm cho mã nguồn dễ đọc, dễ kiểm thử và dễ dàng mở rộng tính năng mới trong tương lai. |
| **Tốc độ Xử lý** | Ngôn ngữ Dart được biên dịch thành mã native, mang lại hiệu suất gần với ứng dụng gốc (native app). |

#### 2.2. Nhược điểm (Weaknesses)

|  |  |
| --- | --- |
| **Lĩnh vực** | **Mô tả** |
| **Kích thước Ứng dụng** | Kích thước file cài đặt (APK/IPA) của ứng dụng Flutter thường lớn hơn so với các ứng dụng Native, có thể ảnh hưởng nhỏ đến quá trình tải xuống ban đầu. |
| **Thư viện Chuyên biệt** | Một số thư viện chuyên sâu trong lĩnh vực Y tế hoặc tích hợp cổng thanh toán phức tạp có thể chưa được tối ưu hóa hoàn toàn cho Flutter, đôi khi cần viết mã Native (Platform Channel) để giải quyết. |

# **CHƯƠNG V: KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN**

Chương này đánh giá mức độ thành công của dự án **Ứng dụng Quản lý Bán Thuốc MEDCARE** dựa trên các tiêu chí đã định trong Phạm vi, đồng thời phân tích các bài học kinh nghiệm về cả sản phẩm và quy trình quản lý.

### **1. Tổng kết Kết quả Đạt được (Success Metrics)**

Dự án được đánh giá thành công dựa trên hai khía cạnh chính: Sản phẩm (Product Delivery) và Quản lý Dự án (Project Management).

#### 1.1. Về Sản phẩm Hoàn thiện (Product Deliverables)

|  |  |  |
| --- | --- | --- |
| **Kết quả** | **Mô tả Chi tiết** | **Tiêu chí Đánh giá** |
| **Sản phẩm Hoàn thiện (MVP)** | Đã hoàn thành phiên bản Sản phẩm Khả dụng Tối thiểu (MVP) của ứng dụng trên cả hai nền tảng **iOS và Android** (sử dụng Flutter). | Đạt Tiêu chí: Hoạt động ổn định trên cả hai nền tảng. |
| **Chức năng Cốt lõi** | Các tính năng quan trọng nhất như Mua bán, Thanh toán, Tải Đơn thuốc (Rx Upload), và Nhắc nhở Uống thuốc đã được triển khai và **hoạt động ổn định** trong môi trường thử nghiệm. | Đạt Tiêu chí: Đáp ứng đầy đủ các chức năng quản lý thuốc, bán hàng, khách hàng. |
| **Hiệu suất và Ổn định** | Ứng dụng đạt hiệu suất cao về tốc độ tải trang, chuyển đổi màn hình và phản hồi dữ liệu từ API. | Đạt Tiêu chí: Ứng dụng hoạt động ổn định và dễ sử dụng. |
| **Độ Bao phủ Kiểm thử** | Đã đạt được độ bao phủ kiểm thử (Test Coverage) là **90%** đối với Unit Test và Widget Test, đảm bảo tính ổn định và dễ bảo trì của mã nguồn (**Đối phó thành công R9 - Lỗi HSD/Tồn kho**). | Đạt Tiêu chí: Dễ sử dụng, giao diện trực quan. |

#### 1.2. Về Quản lý Dự án (Project Management)

|  |  |  |
| --- | --- | --- |
| **Kết quả** | **Mô tả Chi tiết** | **Đánh giá** |
| **Quản lý Lịch biểu** | Hoàn thành dự án **đúng tiến độ cam kết 8 tuần** (4 Sprints). **Velocity Chart** cho thấy tốc độ hoàn thành Story Point tăng dần qua các Sprint. | **Thành công:** Hoàn thành đúng tiến độ. |
| **Quản lý Phạm vi** | Quy trình **Change Control (R7)** được áp dụng nghiêm ngặt, chỉ chấp nhận 1 yêu cầu thay đổi nhỏ và thực hiện đánh đổi tính năng (Trade-off), giúp **tránh Scope Creep** thành công. | **Thành công:** Bảo vệ Phạm vi cốt lõi. |
| **Quản lý Rủi ro** | **Rủi ro R4 (Thành viên chủ chốt nghỉ)** đã được giảm thiểu thông qua việc áp dụng **Đào tạo Chéo** (Cross-training) trong các buổi Stand-up hàng tuần. **R9 (Lỗi HSD/Tồn kho)** được kiểm soát hoàn toàn nhờ Unit Test > 90%. | **Thành công:** Đối phó hiệu quả với các rủi ro Cao. |
| **Quản lý Chất lượng** | Vượt qua thành công giai đoạn **UAT (Kiểm thử Chấp nhận)**, được người dùng thử nghiệm (chủ tiệm thuốc) đánh giá **Hài lòng** với trải nghiệm người dùng và tính chính xác của nghiệp vụ. | **Thành công:** Đáp ứng sự hài lòng của bên liên quan. |

### **2. Bài học Kinh nghiệm (Lessons Learned)**

Phân tích các yếu tố bên trong (Ưu/Nhược điểm) và các yếu tố từ quy trình (Quản lý) để rút ra bài học.

#### 2.1. Ưu điểm (Strengths) và Bài học Chiến lược

|  |  |  |
| --- | --- | --- |
| **Lĩnh vực** | **Mô tả** | **Bài học Kinh nghiệm** |
| **Phát triển Đa nền tảng** | Sử dụng Flutter giúp tiết kiệm đáng kể thời gian và chi phí, đồng nhất giao diện trên cả iOS và Android. | **Bài học:** Lựa chọn công nghệ đa nền tảng là chiến lược tối ưu cho MVP cần triển khai nhanh. |
| **Kiến trúc Mã nguồn** | Áp dụng Clean Architecture/Layered Architecture giúp tách biệt logic. | **Bài học:** Dù thời gian ngắn (8 tuần), việc đầu tư vào kiến trúc Clean vẫn giảm đáng kể nỗ lực bảo trì và tăng khả năng mở rộng. |
| **Tốc độ Xử lý** | Ngôn ngữ Dart được biên dịch thành mã native, mang lại hiệu suất cao. | **Bài học:** Dart/Flutter đáp ứng hiệu suất cho các ứng dụng giao dịch yêu cầu tốc độ cao. |

#### 2.2. Nhược điểm (Weaknesses) và Bài học Quản lý

|  |  |  |
| --- | --- | --- |
| **Lĩnh vực** | **Mô tả** | **Bài học Quản lý Cần Điều chỉnh** |
| **Kích thước Ứng dụng** | Kích thước file cài đặt (APK/IPA) của ứng dụng Flutter vẫn lớn hơn Native. | **Điều chỉnh:** Cần tích hợp các kỹ thuật tối ưu hóa Bundle (Tree Shaking, Deferred Loading) ngay từ Sprint 1 để giảm kích thước. |
| **Thư viện Chuyên biệt** | Tích hợp cổng thanh toán phức tạp (**R2**) cần viết mã Native (Platform Channel). | **Điều chỉnh:** Cần dành thêm thời gian **1 tuần** cho việc Nghiên cứu Kỹ thuật Chuyên sâu (Spike) về các thư viện phức tạp. |
| **Yêu cầu Pháp lý** | Việc quản lý và xác thực đơn thuốc kê đơn (Rx Upload) đòi hỏi sự tuân thủ pháp lý nghiêm ngặt. | **Điều chỉnh:** Cần bố trí một **Chuyên gia Tham vấn Nghiệp vụ** riêng biệt ngay từ đầu để tránh rủi ro nghiệp vụ (**R1**). |
| **Rủi ro API Backend (R8)** | Trong Sprint 2, đã có sự chậm trễ nhỏ do tài liệu API Backend không rõ ràng. | **Điều chỉnh:** Bắt buộc sử dụng công cụ chuẩn hóa tài liệu API (Swagger/Postman) để giảm thiểu **R8** trong các dự án tương lai. |

### **3. Kết luận và Hướng phát triển**

#### 3.1. Kết luận Tổng thể

Dự án phát triển ứng dụng di động MEDCARE đã **hoàn thành toàn diện** các mục tiêu về **Phạm vi, Thời gian (8 tuần), và Chất lượng**. Sự thành công này là nhờ vào việc lựa chọn công nghệ Flutter chiến lược, kết hợp với phương pháp quản lý linh hoạt Scrum, giúp nhóm nhanh chóng phản hồi các thay đổi và rủi ro (R4, R7, R9). Ứng dụng MEDCARE đã chứng minh tính khả thi kỹ thuật và sẵn sàng chuyển sang giai đoạn thương mại hóa.

#### 3.2. Hướng phát triển Tương lai (Next Steps)

**Mở rộng AI/Machine Learning:** Tích hợp module AI để **gợi ý liều dùng cá nhân hóa**, hoặc phân tích ảnh đơn thuốc (OCR) để tự động hóa quy trình xác thực.

**Tích hợp Chăm sóc Sức khỏe:** Phát triển thêm tính năng tích hợp hồ sơ sức khỏe điện tử cá nhân (EHR) và khả năng tích hợp bảo hiểm y tế.

**Tối ưu Hiệu suất Bundle:** Thực hiện các kỹ thuật tối ưu hóa chuyên sâu để giảm kích thước cuối cùng của file ứng dụng.

**Tối ưu hóa Chi phí:** Xem xét tối ưu hóa cấu hình Firebase (giảm chi phí Cloud Function) để chuẩn bị cho giai đoạn mở rộng người dùng (Scaling).

# **TÀI LIỆU THAM KHẢO**

| **Tên Tài liệu** | **Liên kết (URL)** |
| --- | --- |
| **Flutter Documentation** | https://docs.flutter.dev/ |
| **Dart Documentation** | https://dart.dev/guides |
| **Flutter Codelabs (Bài thực hành)** | https://codelabs.developers.google.com/?cat=Flutter |
| **Flutter Pub.dev (Kho gói/Packages)** | https://pub.dev/ |
| **Clean Architecture (Uncle Bob)** | https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html |
| **Riverpod Documentation (Quản lý trạng thái)** | https://riverpod.dev/ |
| **Firebase Auth (Xác thực)** | https://pub.dev/packages/firebase\_auth |
| **Cloud Firestore (Cơ sở dữ liệu)** | https://pub.dev/packages/cloud\_firestore |
| **Google Maps Flutter** | https://pub.dev/packages/google\_maps\_flutter |
| **Agile Manifesto (Tuyên ngôn Agile)** | https://agilemanifesto.org/iso/vi/manifesto.html |
| **Scrum Guide (Hướng dẫn Scrum)** | https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf |
| **Jira Software (Quản lý dự án)** | https://www.atlassian.com/software/jira |
| **Material Design 3 (Hướng dẫn thiết kế UI)** | https://m3.material.io/ |
| **Human Interface Guidelines (Thiết kế iOS)** | https://developer.apple.com/design/human-interface-guidelines/ |