Đặc Tả Yêu Cầu Phần Mềm

cho

<Dự Án>

**Phiên bản X.X được phê chuẩn**

**Được chuẩn bị bởi <Tác giả>**

**<Tổ chức>**

**<Ngày tạo ra bản đặc tả>**

**Mục Lục**

[1. Giới thiệu 1](#_Toc221150633)

[1.1 Mục tiêu 1](#_Toc221150634)

[1.2 Phạm vi sản phẩm 1](#_Toc221150635)

[1.3 Bảng chú giải thuật ngữ 2](#_Toc221150636)

[1.4 Tài liệu tham khảo 2](#_Toc221150637)

[1.5 Bố cục tài liệu 3](#_Toc221150638)

[2. Mô tả tổng quan 4](#_Toc221150639)

[2.1 Bối cảnh của sản phẩm 4](#_Toc221150640)

[2.2 Các chức năng của sản phẩm 5](#_Toc221150641)

[2.3 Đặc điểm người sử dụng 5](#_Toc221150642)

[2.4 Môi trường vận hành 6](#_Toc221150643)

[2.5 Các ràng buộc về thực thi và thiết kế 6](#_Toc221150644)

[2.6 Các giả định và phụ thuộc 7](#_Toc221150645)

[3. Các yêu cầu giao tiếp bên ngoài 7](#_Toc221150646)

[3.1 Giao diện người sử dụng 7](#_Toc221150647)

[3.2 Giao tiếp phần cứng 11](#_Toc221150648)

[3.3 Giao tiếp phần mềm 12](#_Toc221150649)

[3.4 Giao tiếp truyền thông tin 13](#_Toc221150650)

[4. Các tính năng của hệ thống 15](#_Toc221150651)

[4.1 Thêm dự án 15](#_Toc221150652)

[4.2 Xóa dự án 16](#_Toc221150653)

[4.3 Chỉnh sửa thông tin dự án 17](#_Toc221150654)

[4.4 Chức năng đăng ký tài khoản 18](#_Toc221150655)

[4.5 Chức năng đăng nhập 19](#_Toc221150656)

[4.6 Chức năng Quản lý và Phân quyền Người dùng: 20](#_Toc221150657)

[4.7 Tính năng Nhận thông báo 21](#_Toc221150658)

[4.8 Chức năng tạo công việc 22](#_Toc221150659)

[4.9 Chức năng xem công việc 23](#_Toc221150660)

[4.10 Chức năng bình luận công việc 24](#_Toc221150661)

[4.11 Chức năng cập nhật trạng thái công việc 25](#_Toc221150662)

[4.12 Chức năng phân quyền dự án 26](#_Toc221150663)

[4.13 Chức năng xem thống kê 27](#_Toc221150664)

[5 Các yêu cầu phi chức năng 28](#_Toc221150665)

[5.1 Yêu cầu thực thi 28](#_Toc221150666)

[5.2 Yêu cầu an toàn 29](#_Toc221150667)

[5.3 Yêu cầu bảo mật 30](#_Toc221150668)

[5.4 Các đặc điểm chất lượng phần mềm 31](#_Toc221150669)

[5.5 Các quy tắc nghiệp vụ 33](#_Toc221150670)

[6 Các yêu cầu khác 34](#_Toc221150671)

**Theo dõi phiên bản tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Ngày** | **Lý do thay đổi** | **Phiên bản** |
|  |  |  |  |
|  |  |  |  |

# Giới thiệu

## Mục tiêu

Mục tiêu:

* Làm cơ sở để đánh giá khả năng tái sử dụng, bảo trì và mở rộng của sản phẩm cuối cùng.
* Xác định và Thống nhất: Cung cấp bản mô tả chi tiết, chính xác về các yêu cầu chức năng (quản lý dự án, công việc, người dùng) và yêu cầu kỹ thuật (Java, kiến trúc đa tầng, Design Patterns) để làm cơ sở thống nhất giữa nhóm phát triển
* Làm nền tảng thực thi: Cung cấp thông tin đầu vào để nhóm thiết kế xây dựng các biểu đồ UML, nhóm lập trình xây dựng Core Framework/Plugin và nhóm kiểm thử thực hiện JUnit Test
* Đảm bảo chất lượng: Thiết lập các tiêu chí để đạt được khả năng tái sử dụng cao, dễ bảo trì, dễ mở rộng và thích ứng tốt với các thay đổi thông qua kiến trúc tách biệt

Nhóm người sử dụng:

* Nhóm Thiết kế (Designers): Sử dụng tài liệu để xây dựng các biểu đồ UML (Use Case, Class, Sequence diagram) và thiết kế kiến trúc hệ thống Plugin/HostContext.
* Nhóm Phát triển (Developers): Căn cứ vào các yêu cầu chức năng và kỹ thuật để thực hiện lập trình Java, xây dựng PluginLoader và giao diện Dashboard.
* Nhóm Kiểm thử (Testers): Dựa trên các mô tả chức năng để viết Unit Test bằng JUnit và thực hiện Integration Test nhằm đảm bảo chất lượng phần mềm.
* Quản lý dự án: Sử dụng tài liệu để lập kế hoạch, phân chia các giai đoạn (Sprints) và theo dõi tiến độ.
* Người quản lý: Sử dụng tài liệu để theo dõi, hướng dẫn và đánh giá kết quả thực hiện đồ án của nhóm
* Khách hàng hoặc chủ đầu tư: Đọc tài liệu đề xác nhận rằng phần mềm được thiết kế đúng với nhu cầu và mục tiêu kinh doanh của họ trước khi đội ngũ kỹ thuật bắt tay vào lập trình.

## Phạm vi sản phẩm

* Tên sản phẩm: Hệ thống quản lý công việc nhóm - TeamWork Master.
* Mô tả và Mục đích: TeamWork Master là ứng dụng Java quy mô vừa, đóng vai trò là nền tảng cộng tác tập trung cho các nhóm dự án. Mục đích của phần mềm là số hóa quy trình quản lý, giúp các thành viên phối hợp nhịp nhàng trong việc phân công, theo dõi tiến độ và hoàn thành mục tiêu chung một cách minh bạch, khoa học.
* Chức năng chính:
* Quản trị tập trung: Quản lý dự án, phân quyền người dùng (Admin/Manager/Member).
* Điều phối nhiệm vụ: Khởi tạo công việc, gán người thực hiện, đặt deadline và cập nhật trạng thái (To Do, In Progress, Done).
* Tương tác & Thông báo: Hỗ trợ bình luận trực tiếp trong công việc và tự động gửi thông báo nhắc nhở tiến độ.
* Kiến trúc mở: Sử dụng hệ thống Plugin cho phép mở rộng thêm tính năng mới mà không cần can thiệp vào mã nguồn lõi.
* Lợi ích và Liên hệ chiến lược:
* Về vận hành: Tăng hiệu suất làm việc nhóm, giảm thiểu sai sót do quên việc và đảm bảo tính kỷ luật thông qua hệ thống báo cáo, thông báo thời gian thực.
* Về kỹ thuật: Đảm bảo tính linh hoạt và dễ bảo trì nhờ áp dụng các mẫu thiết kế và kiến trúc đa tầng.
* Về chiến lược: Hỗ trợ tốt cho mô hình làm việc Agile và quản trị dựa trên dữ liệu. Thiết kế ưu tiên khả năng tái sử dụng giúp tổ chức tối ưu hóa chi phí phát triển và dễ dàng mở rộng quy mô trong tương lai.

## Bảng chú giải thuật ngữ

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ / Từ viết tắt | Định nghĩa / Giải thích |
| 1 | OOP | Lập trình hướng đối tượng |
| 2 | UML | Ngôn ngữ mô hình hóa thống nhất |
| 3 | Plugin | Phần mềm bổ sung tích hợp vào ứng dụng |
| 4 | DBMS | Hệ quản trị cơ sở dữ liệu |
| 5 | JVM | Máy ảo java |
| 6 | REQ | Yêu cầu |
| 7 | PER | Hiệu năng hoặc Yêu cầu thực thi |
| 8 | SQF | An toàn |
| 9 | SEC | Bảo mật |

## Tài liệu tham khảo

### 1.4.1. What is Use Case Diagram?

* Tiêu đề: What is Use Case Diagram?
* Tác giả: Visual Paradigm (không ghi rõ cá nhân)
* Trang web / Nguồn: Visual-Paradigm.com – Guide to UML & Unified Modeling Language
* URL: [https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/?utm_source=chatgpt.com)

Mô tả: Tài liệu này giới thiệu định nghĩa Use Case Diagram theo UML, mục đích sử dụng của sơ đồ use case trong phân tích yêu cầu hệ thống, cách hiển thị các thành phần như Actor, Use Case, các kiểu quan hệ (extends, include, generalization) và cách sơ đồ này hỗ trợ phát triển phần mềm. Nó định vị Use Case Diagram như một loại sơ đồ hành vi trong bộ UML.

### 1.4.2. Software Requirements Specification (SRS).

* Tiêu đề: How to Write a Software Requirements Specification (SRS) Document
* Tác giả: Gerhard Krüger — Perforce Software (Application Lifecycle Management, Software Quality)
* Nơi đăng: Blog chính thức của Perforce (Perforce.com)
* Ngày xuất bản: 11 July 2025
* URL: [https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document?utm_source=chatgpt.com)
* Loại tài liệu: Bài hướng dẫn trực tuyến (blog/guide) về cách viết SRS
* Mô tả nội dung: Bài viết này giải thích mục đích và vai trò của Software Requirements Specification (SRS), các bước thực hiện để viết một SRS rõ ràng và hữu dụng trong dự án phần mềm, các best practices, so sánh giữa việc viết bằng Word so với dùng công cụ quản lý yêu cầu, và lợi ích khi dùng công cụ Perforce ALM trong việc quản lý SRS.

### 1.4.3. Các mẫu thiết kế giao diện người dùng

* Tiêu đề: Các mẫu thiết kế giao diện người dùng
* Nguồn / Nơi cung cấp: Website BAC – Chuyên trang chia sẻ kiến thức cho nhà thiết kế UI/UX và Business Analyst
* Định dạng tài liệu: Bài viết trực tuyến (HTML – blog kiến thức)
* Ngày đăng: 07 Tháng 10, 2021
* Tác giả / Nhà sản xuất nội dung: Ban biên tập BAC (không nêu tên cá nhân cụ thể)
* Mục tiêu nội dung: Giới thiệu và tổng hợp một số mẫu thiết kế giao diện người dùng (UI patterns) – những cách thiết kế giao diện đã được chứng minh là hữu ích trong thực tế, giúp hỗ trợ thiết kế UI đúng với nhu cầu và trực quan cho người dùng.
* Đối tượng đọc: Designers, UI/UX practitioners, developers, Business Analysts, người mới học thiết kế giao diện.
* URL: [https://www.bacs.vn/vi/blog/kien-thuc/cac-mau-thiet-ke-giao-dien-nguoi-dung-phan-1-20894.html](https://www.bacs.vn/vi/blog/kien-thuc/cac-mau-thiet-ke-giao-dien-nguoi-dung-phan-1-20894.html?utm_source=chatgpt.com)

### 1.4.4 Quy trình phát triển phần mềm và các mô hình phát triển phần mềm

* Tiêu đề: QUY TRÌNH PHÁT TRIỂN PHẦN MỀM VÀ CÁC MÔ HÌNH PHÁT TRIỂN PHẦN MỀM
* Tác giả / Người đăng: Thao Gold (tài khoản tác giả trên Viblo)
* Nơi đăng: Viblo – nền tảng chia sẻ kiến thức kỹ thuật cho lập trình viên và chuyên gia CNTT
* Ngày xuất bản: 2 tháng 6, 2020
* Định dạng: Bài viết trực tuyến trên web (HTML)
* URL: [https://viblo.asia/p/quy-trinh-phat-trien-phan-mem-va-cac-mo-hinh-phat-trien-phan-mem-gAm5ybP8Kdb](https://viblo.asia/p/quy-trinh-phat-trien-phan-mem-va-cac-mo-hinh-phat-trien-phan-mem-gAm5ybP8Kdb?utm_source=chatgpt.com)
* Mô tả nội dung: Bài viết trình bày các khái niệm cơ bản về quy trình phát triển phần mềm (software development process) và mô tả chi tiết các mô hình phát triển phần mềm phổ biến như Waterfall (thác nước), V-Model (mô hình chữ V), Spiral (xoắn ốc) và Agile.

## Bố cục tài liệu

-Tài liệu Đặc tả Yêu cầu Phần mềm (SRS) này được tổ chức thành các phần chính như sau, nhằm mô tả toàn diện hệ thống TeamWork Master từ tổng quan đến chi tiết kỹ thuật:

* Phần 2 - Mô tả tổng quan: Cung cấp cái nhìn bao quát về bối cảnh sản phẩm, mô tả các nhóm người dùng (Admin, Manager, Member), môi trường vận hành (Java/Vue.js), các giả định và những ràng buộc thiết kế quan trọng (đặc biệt là kiến trúc Microkernel/Plugin).
* Phần 3 - Các yêu cầu giao tiếp bên ngoài: Mô tả chi tiết về các giao diện tương tác của hệ thống, bao gồm:
  + Giao diện người dùng (UI) trên nền tảng Web (Vue.js).
  + Các giao thức giao tiếp phần mềm (RESTful API, cơ chế nạp Plugin).
  + Giao tiếp phần cứng và các yêu cầu về truyền thông tin.
* Phần 4 - Các tính năng của hệ thống: Đây là phần trọng tâm, liệt kê chi tiết các yêu cầu chức năng (Functional Requirements) cụ thể như: Quản lý dự án, Công việc, Người dùng, Thông báo, Thống kê và Phân quyền. Mỗi tính năng được mô tả kèm theo mức độ ưu tiên, tác nhân và chuỗi đáp ứng.
* Phần 5 - Các yêu cầu phi chức năng: Xác định các tiêu chuẩn kỹ thuật khắt khe về Hiệu năng (PER), An toàn (SAF), Bảo mật (SEC), Chất lượng phần mềm (QA) và các Quy tắc nghiệp vụ (BR) mà hệ thống bắt buộc phải tuân thủ.
* Phần 6 - Các yêu cầu khác: Bao gồm các yêu cầu về Cơ sở dữ liệu, vấn đề Pháp lý/Bản quyền, mục tiêu Tái sử dụng (Design for Reuse) và quy cách Đóng gói/Phân phối sản phẩm.
* Phụ lục: Chứa các mô hình phân tích bổ sung và danh sách các vấn đề chưa được xác định (TBD).

**Đề xuất các mục nên đọc với từng nhóm người đọc:**

|  |  |  |
| --- | --- | --- |
| Nhóm đối tượng | Các mục trọng tâm cần đọc | Mục đích |
| Khách hàng / Chủ đầu tư | Phần 1, 2, và danh sách tính năng ở Phần 4 (4.1 - 4.13). | Hiểu phạm vi dự án, xác nhận các chức năng nghiệp vụ đáp ứng đúng nhu cầu kinh doanh. |
| Quản lý dự án (PM) | Toàn bộ tài liệu, đặc biệt chú trọng Phần 2 (Ràng buộc) và Phần 4. | Lên kế hoạch phát triển, theo dõi tiến độ dựa trên mức độ ưu tiên của từng tính năng. |
| Nhóm Thiết kế (Architects) | Phần 2 (Ràng buộc kiến trúc), Phần 3 (Giao tiếp), Phần 5 (Phi chức năng) và Phần 6 (Tái sử dụng). | Thiết kế kiến trúc hệ thống Core/Plugin, cơ sở dữ liệu và các giao diện API. |
| Nhóm Lập trình viên (Developers) | Frontend (Vue.js): Đọc kỹ Phần 3.1, 3.3, 3.4 và chi tiết Phần 4.  Backend (Java): Đọc kỹ Phần 2.5, 3.2, 3.3, Phần 5 (SAF, SEC, PER) và Phần 6. | Hiện thực hóa các yêu cầu chức năng và đảm bảo tuân thủ các quy tắc kỹ thuật/an toàn. |
| Nhóm Kiểm thử (Testers) | Toàn bộ tài liệu, đặc biệt là Phần 4 (để viết Test Case chức năng) và Phần 5 (để kiểm thử hiệu năng/bảo mật). | Xây dựng kịch bản kiểm thử (Test Plan) đảm bảo sản phẩm đạt chất lượng yêu cầu. |

# Mô tả tổng quan

## Bối cảnh của sản phẩm

* Nguồn gốc và Ngữ cảnh: Dự án ra đời nhằm đáp ứng nhu cầu quản lý công việc theo mô hình cộng tác hiện đại dành cho các nhóm quy mô vừa. Thay vì sử dụng các công cụ rời rạc như email hay bảng tính, TeamWork Master cung cấp một môi trường tích hợp. Sản phẩm này là kết quả của việc áp dụng các nguyên lý thiết kế phần mềm quan trọng như:
* Design for Change: Khả năng thích ứng với thay đổi thông qua kiến trúc đa tầng.
* Extensibility: Khả năng mở rộng thông qua cơ chế nạp lớp động
* Mối quan hệ với các hệ thống khác: Mặc dù là sản phẩm độc lập, TeamWork Master được thiết kế để có thể dễ dàng tích hợp vào một hệ sinh thái phần mềm doanh nghiệp lớn hơn trong tương lai. Hệ thống bao gồm hai thành phần chính tương tác chặt chẽ:
* Hệ thống Host (Core): Đóng vai trò là "Container" quản lý vòng đời ứng dụng, cung cấp ngữ cảnh (HostContext) và điều phối các dịch vụ dùng chung.
* Hệ thống Plugins: Các thành phần chức năng (như Widget quản lý công việc, Widget thống kê) được phát triển tách biệt và được nạp vào Core khi ứng dụng khởi chạy.
* Sơ đồ khối và các thành phần chính: Cấu trúc của hệ thống và sự tương tác giữa các thành phần được mô tả qua các lớp sau:
* Lớp Giao diện (User Interface): Dashboard Frame đóng vai trò giao diện chính, cung cấp hệ thống Menu, Layout và là nơi hiển thị các Widgets từ Plugin. Thành phần này tương tác trực tiếp với người dùng.
* Lớp Điều khiển & Logic (Business Logic - Plugin Loader): Sử dụng Java Reflection để quét và nạp các Plugin. Đây là trung tâm điều phối, giải quyết các tương tác giữa các Plugin và Core thông qua các Interfaces đã được định nghĩa.
* Lớp Dữ liệu (Data Access Layer): Quản lý việc lưu trữ và truy xuất thông tin về dự án, công việc và người dùng. Hệ thống áp dụng mẫu thiết kế để đảm bảo dữ liệu luôn nhất quán giữa các luồng xử lý (Concurrency).
* Tương tác ngoài (External Interactions): Người dùng (Actors): Admin (quản trị hệ thống), Manager (quản lý dự án), và Member (thực hiện công việc). Mỗi đối tượng tương tác với hệ thống qua các quyền hạn khác nhau.

## Các chức năng của sản phẩm

* Chức năng sản phẩm:
* Quản lý dự án: Tạo, chỉnh sửa, xóa dự án; phân quyền quản lý dự án.
* Quản lý công việc: Tạo công việc, gán người thực hiện, đặt deadline, trạng thái(To Do, In Progress, Done, Cancel…).
* Quản lý người dùng: Đăng ký, đăng nhập, phân vai trò (Admin, Manager, Member).
* Thông báo: Gửi thông báo khi có thay đổi trạng thái công việc hoặc deadline sắp đến.
* Bình luận & trao đổi: Cho phép thảo luận trong từng công việc.
* Thống kê & báo cáo: Số lượng công việc theo trạng thái, tiến độ dự án.

## Đặc điểm người sử dụng

* Quản trị viên hệ thống (Administrator) - Nhóm hỗ trợ
* Tần suất sử dụng: Thỉnh thoảng (khi cần thiết lập hệ thống).
* Kinh nghiệm: Có kiến thức về quản trị phần mềm và hệ thống.
* Chức năng sử dụng: Quản lý tài khoản người dùng (cấp quyền, khóa tài khoản), quản lý danh mục dự án tổng thể và cấu hình các Plugins cho toàn bộ ứng dụng.
* Mức bảo mật: Cao nhất, có quyền truy cập vào toàn bộ dữ liệu cấu hình hệ thống.
* Quản lý dự án (Manager) - Nhóm quan trọng
* Tần suất sử dụng: Thường xuyên hàng ngày.
* Kinh nghiệm: Am hiểu quy trình quản lý công việc và điều phối nhân sự.
* Chức năng sử dụng: Tạo và thiết lập dự án; phân công nhiệm vụ cho các thành viên; thiết lập deadline; theo dõi biểu đồ tiến độ và phê duyệt trạng thái hoàn thành công việc. Đây là nhóm người dùng quyết định luồng vận hành của dữ liệu trên hệ thống.
* Mức bảo mật: Cao (trong phạm vi các dự án được giao quản lý).
* Thành viên nhóm (Member) - Nhóm đông đảo nhất
* Tần suất sử dụng: Rất thường xuyên (liên tục trong ngày làm việc).
* Kinh nghiệm: Người sử dụng phổ thông, cần giao diện trực quan, dễ dùng.
* Chức năng sử dụng: Tiếp nhận công việc được giao; cập nhật trạng thái thực hiện (To Do, In Progress, Done); thực hiện bình luận, trao đổi và đính kèm tài liệu trong từng nhiệm vụ.
* Mức bảo mật: Trung bình (chỉ xem và sửa đổi các công việc liên quan đến bản thân hoặc dự án đang tham gia).

**🡪** Nhóm quan trọng nhất: Manager (Quản lý dự án) và Member (Thành viên). Vì mục tiêu cốt lõi của phần mềm là "hợp tác hiệu quả", các tính năng cập nhật trạng thái công việc phải được ưu tiên hàng đầu để đáp ứng nhu cầu tương tác liên tục của hai nhóm này. Nhóm ít quan trọng hơn: Administrator. Tuy đóng vai trò duy trì hệ thống nhưng các chức năng của Admin không trực tiếp tạo ra giá trị cộng tác hàng ngày cho dự án.

## Môi trường vận hành

* Phần mềm TeamWork Master được thiết kế để hoạt động đa nền tảng trên môi trường Java. Yêu cầu vận hành cụ thể:
  + Phần cứng: Máy tính cá nhân có bộ vi xử lý đa nhân (Multi-core CPU) để hỗ trợ xử lý đa luồng và RAM tối thiểu 4GB để đảm bảo hiệu suất cho giao diện đồ họa.
  + Hệ điều hành: Tương thích với Windows (10/11), macOS hoặc Linux.
  + Môi trường phần mềm: Yêu cầu cài đặt Java Runtime Environment (JRE) 11 trở lên. Hệ thống cần quyền truy cập thư mục cục bộ để nạp các tệp Plugin (.jar/.class).
  + Thành phần khác: Có thể cùng tồn tại với các phần mềm văn phòng khác mà không gây xung đột tài nguyên.

## Các ràng buộc về thực thi và thiết kế

Quá trình phát triển hệ thống Quản lý công việc nhóm (TeamTask) phải tuân thủ nghiêm ngặt các ràng buộc sau đây để đảm bảo tính ổn định và khả năng mở rộng của kiến trúc Microkernel:

* Ràng buộc về Kiến trúc và Thiết kế:
  + Nguyên tắc "Hộp đen" (Black-box): Các Plugin KHÔNG được phép truy cập trực tiếp vào Cơ sở dữ liệu. Mọi thao tác lấy dữ liệu, ghi log, hoặc hiển thị thông báo đều phải thông qua "Hợp đồng" là interface IHostContext do hệ thống Core cung cấp.
  + Tính Cô lập (Isolation): Lỗi phát sinh từ một Plugin (ví dụ: lỗi ngoại lệ do lập trình viên bên thứ ba viết sai) không được phép làm sập (crash) hệ thống Core. Core phải có cơ chế bắt lỗi (Exception Handling) khi thực thi phương thức execute() của Plugin.
* Ràng buộc về Công nghệ và Ngôn ngữ:
* Ngôn ngữ lập trình: Hệ thống phải được phát triển bằng Java (JDK 17 trở lên). Đây là yêu cầu bắt buộc để tận dụng cơ chế ClassLoader mạnh mẽ của Java cho việc nạp động các file .jar (Plugin) vào thời gian chạy (runtime).
* Giao diện: Sử dụng Java Swing hoặc JavaFX hoặc vuejs để đảm bảo phần mềm có thể chạy trên nhiều hệ điều hành (Windows, macOS, Linux) mà không cần viết lại mã giao diện.
* Ràng buộc về Phần cứng và Hiệu năng:
* Do hướng tới các nhóm làm việc quy mô vừa sử dụng trên máy tính văn phòng, hệ thống lõi (Core) khi khởi động không được chiếm quá 250MB RAM.
* Quá trình quét và nạp các Plugin mới vào bộ nhớ không được làm "treo" giao diện người dùng quá 3 giây.
* Ràng buộc về Cấu trúc Thư mục:
* Hệ thống yêu cầu phải có một thư mục cụ thể (ví dụ: /plugins/) nằm cùng cấp với file chạy chính của Core. Chỉ các file có định dạng .jar nằm trong thư mục này mới được hệ thống nhận diện là Plugin hợp lệ.

## Các giả định và phụ thuộc

* Các giả định (Assumptions):
* Giả định về Môi trường người dùng: Máy tính của người dùng cuối (cả Quản lý và Thành viên) đều đã được cài đặt sẵn môi trường thực thi Java (JRE - Java Runtime Environment) phiên bản 17 trở lên. Nếu giả định này sai, hệ thống Core sẽ không thể khởi chạy.
* Giả định về Kết nối mạng: Do đây là ứng dụng cộng tác nhóm, giả định rằng hệ thống mạng nội bộ (LAN) hoặc kết nối Internet luôn ở trạng thái ổn định để các máy trạm có thể giao tiếp với Cơ sở dữ liệu tập trung theo thời gian thực.
* Giả định về Tính tuân thủ của Plugin: Giả định rằng các lập trình viên bên thứ ba (hoặc đội ngũ phát triển sau này) khi viết Plugin mở rộng sẽ tuân thủ nghiêm ngặt "Hợp đồng" (Interface IPlugin và IHostContext). Nếu Plugin cố tình can thiệp vào các luồng không được phép, nó có thể vi phạm các ràng buộc an toàn của hệ thống.
* Các phụ thuộc (Dependencies):
* Phụ thuộc vào Thư viện mã nguồn mở (Third-party Libraries): Dự án này muốn tái sử dụng các thành phần phần mềm có sẵn, cụ thể:
* Phụ thuộc vào thư viện Jackson (để phân tích cú pháp dữ liệu JSON khi Plugin và Core trao đổi dữ liệu).
* Phụ thuộc vào thư viện Log4j (để ghi nhận nhật ký hệ thống - log). Bất kỳ lỗ hổng bảo mật nào từ các thư viện này cũng sẽ ảnh hưởng trực tiếp đến ứng dụng.
* Phụ thuộc vào Hệ quản trị cơ sở dữ liệu (DBMS): Hệ thống Core phụ thuộc vào MySQL để lưu trữ dữ liệu người dùng và công việc. Việc thay đổi cấu trúc dữ liệu hoặc gián đoạn dịch vụ từ phía máy chủ MySQL sẽ khiến toàn bộ ứng dụng bị tê liệt.
* Phụ thuộc vào Hệ điều hành: Mặc dù viết bằng Java có tính đa nền tảng, việc truy xuất hệ thống file (để quét thư mục /plugins/ và nạp file .jar) phụ thuộc vào quyền (Permission) đọc/ghi (Read/Write) của hệ điều hành Windows/macOS đang cấp cho ứng dụng.

# Các yêu cầu giao tiếp bên ngoài

## Giao diện người sử dụng

### Nguyên tắc thiết kế chung

* Công nghệ Frontend: Sử dụng Vue.js kết hợp với các công nghệ web tiêu chuẩn (HTML5, CSS3, JavaScript/TypeScript). Giao diện giao tiếp với hệ thống Backend (Java Core) thông qua chuẩn RESTful API.
* Tính thích ứng (Responsiveness): Giao diện có khả năng tự động điều chỉnh kích thước để hiển thị tốt trên các độ phân giải màn hình khác nhau (Laptop, Desktop, Tablet).
* Trải nghiệm người dùng (UX): Tuân thủ quy tắc "3 Click" - người dùng có thể truy cập bất kỳ chức năng chính nào trong tối đa 3 lần nhấp chuột.
* Ngôn ngữ: Hỗ trợ hiển thị đầy đủ bảng mã Unicode (UTF-8) để hiển thị Tiếng Việt chính xác.

### Bố cục màn hình (Layout)

Màn hình làm việc chính (Dashboard) được tổ chức theo cấu trúc 3 vùng độc lập để tối ưu hóa không gian làm việc:

* Thanh điều hướng bên trái (Sidebar Navigation):
* Chứa menu chính để truy cập các module: Tổng quan, Dự án, Thành viên, và Cài đặt.
* Hiển thị danh sách các dự án đang tham gia để chuyển đổi nhanh.
* Trạng thái: Có thể thu gọn (collapse) để mở rộng không gian làm việc.
* Vùng làm việc trung tâm (Main View - Router View):
  + Đây là nơi hiển thị nội dung chính dựa trên điều hướng của người dùng (ví dụ: Bảng Kanban, Danh sách công việc, Biểu đồ thống kê).
  + Sử dụng cơ chế vue-router để chuyển đổi giữa các màn hình mà không cần tải lại toàn bộ trang.
* Thanh công cụ phía trên (Top Bar / Action Bar):
  + Góc trái: Hiển thị đường dẫn (Breadcrumb) và tên màn hình hiện tại.
  + Góc phải: Chứa thông tin tài khoản người dùng, nút Thông báo (Notification Bell) , và các công cụ mở rộng từ Plugin.

### Các thành phần giao diện chuẩn (UI Components)

Hệ thống sử dụng bộ thư viện thành phần (Component Library) thống nhất để đảm bảo tính nhất quán:

* Bảng Kanban (Kanban Board):
* Hỗ trợ thao tác Kéo & Thả (Drag & Drop) sử dụng HTML5 Drag API hoặc thư viện Vue Draggable để chuyển trạng thái công việc giữa các cột (To Do, In Progress, Done).
* Hộp thoại (Modals/Dialogs):
  + Sử dụng các cửa sổ nổi (Modal) đè lên giao diện chính cho các tác vụ cần sự tập trung như: Tạo mới dự án, Chỉnh sửa công việc, hoặc Xác nhận xóa.
  + Hộp thoại phải có nút "Đóng" (X) và nút hành động rõ ràng (Lưu/Hủy).
* Thông báo hệ thống (Notifications):
* Thông báo thành công/thông tin: Hiển thị dạng "Toast Message" (thông báo nổi tạm thời) ở góc trên bên phải màn hình và tự động biến mất sau 3 giây.
* Thông báo lỗi: Các lỗi nghiêm trọng (mất kết nối, lỗi server) hiển thị dạng Alert hoặc Modal yêu cầu người dùng xác nhận

### Phím tắt và Tiện ích (Shortcuts)

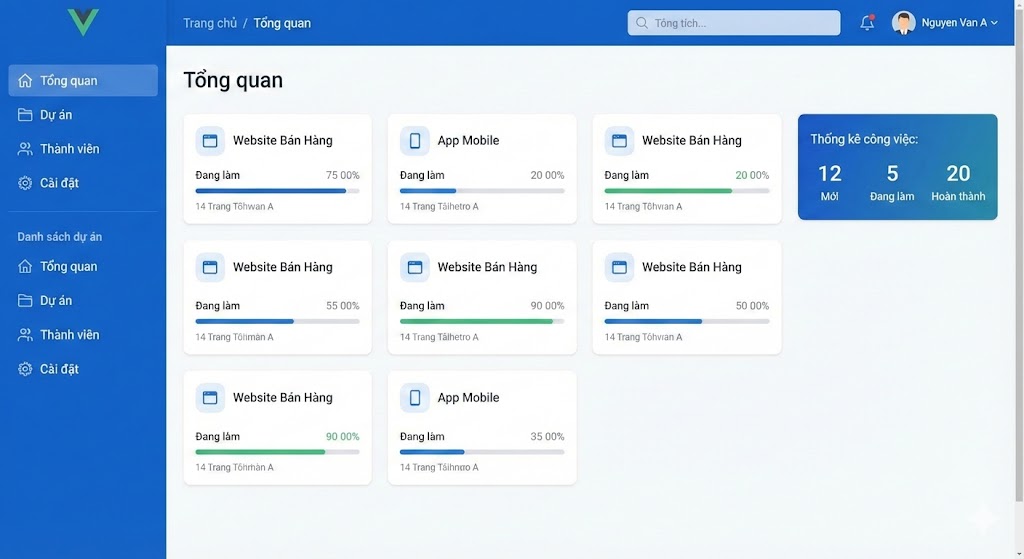
* Ctrl + N: Mở Modal tạo công việc mới nhanh.
* Ctrl + F: Focus vào ô tìm kiếm công việc.
* Esc: Đóng các hộp thoại (Modal) đang mở.
* Ctrl + N: Tạo công việc mới nhanh (Quick Add Task).
* Ctrl + F: Mở thanh tìm kiếm công việc.
* Ctrl + P: Mở giao diện Trình quản lý Plugin (Plugin Manager).

### Các màn hình chính cần thiết kế

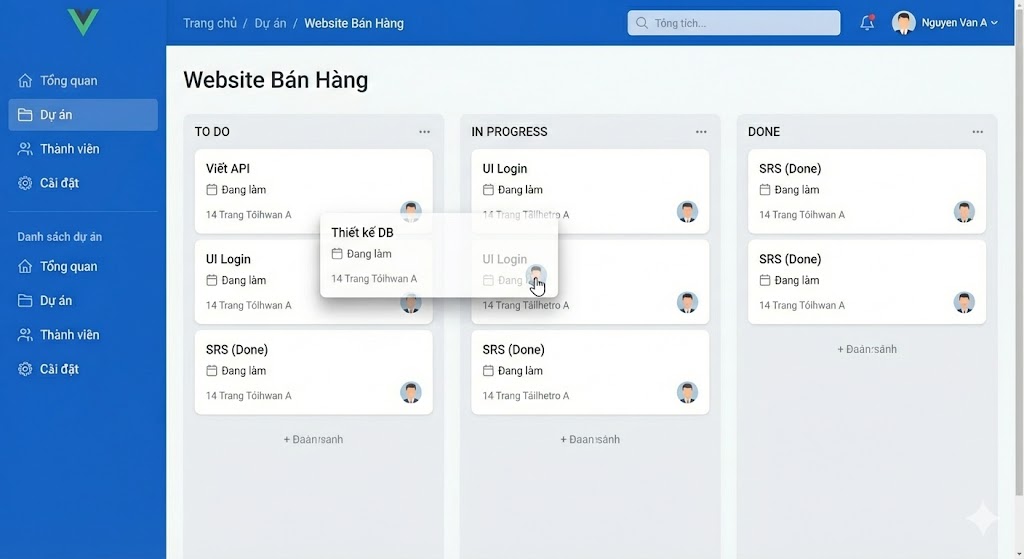
* Màn hình Đăng nhập/Đăng ký: Form nhập liệu bảo mật, kiểm tra định dạng email/mật khẩu ngay tại phía client (Client-side validation).
* Dashboard tổng quan: Hiển thị danh sách dự án và các widget thống kê.
* Chi tiết dự án (Kanban/List View): Giao diện quản lý công việc chính.
* Trang Quản lý Plugin: Giao diện cho phép Admin tải lên file .jar (gửi về backend) và bật/tắt các tính năng mở rộng.

### Thiết kế giao diện

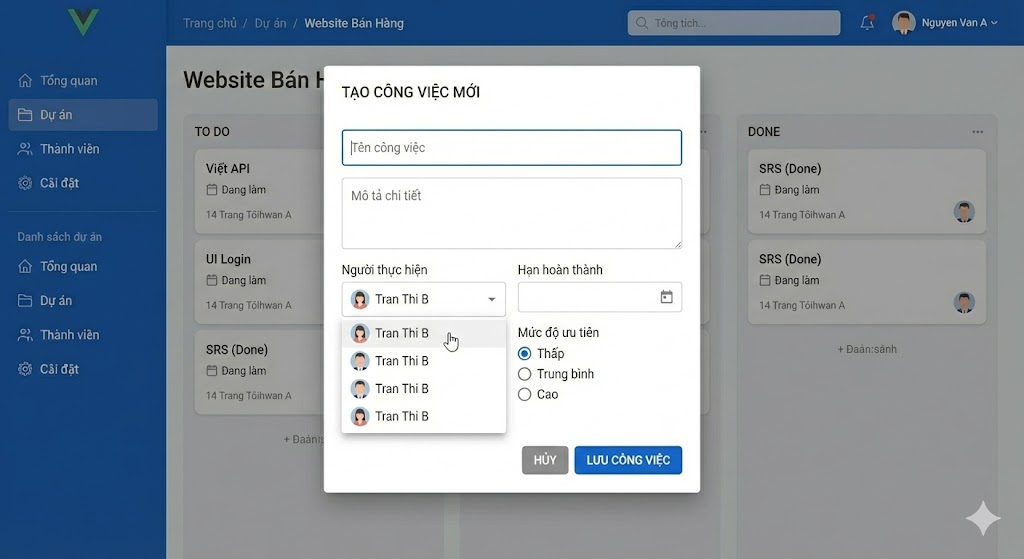
* Giao diện Dashboard Tổng quan (Màn hình chính): Đây là màn hình đầu tiên người dùng nhìn thấy sau khi đăng nhập. Nó cung cấp một cái nhìn tổng quan về các dự án đang tham gia và các thống kê quan trọng. Giao diện tuân thủ bố cục 3 phần với thanh điều hướng bên trái, thanh công cụ phía trên và khu vực nội dung chính ở giữa.



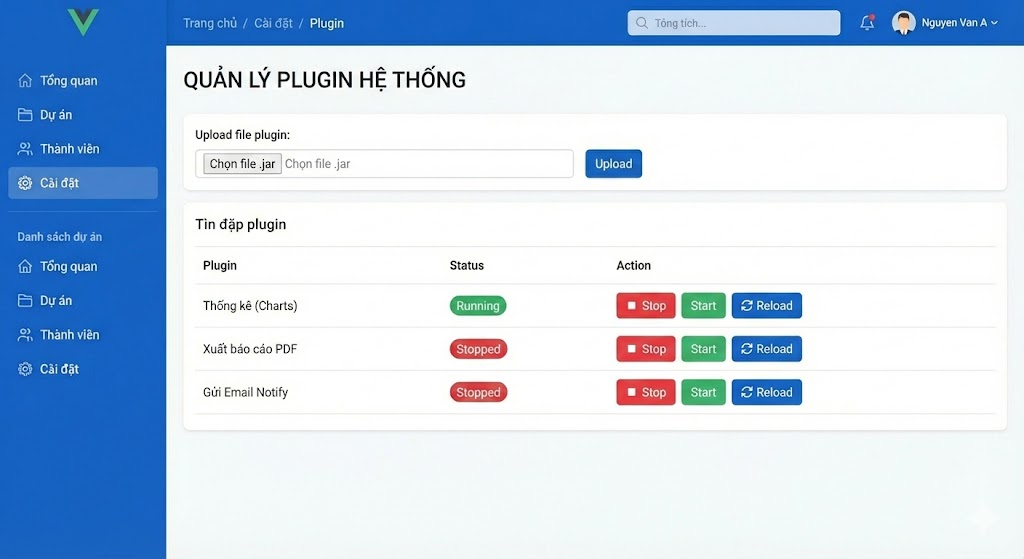
* Giao diện Chi tiết Dự án (Kanban Board): Đây là giao diện làm việc chính của người dùng. Các công việc được tổ chức trên một bảng Kanban với các cột trạng thái "TO DO", "IN PROGRESS", và "DONE". Người dùng có thể dễ dàng thay đổi trạng thái công việc bằng thao tác Kéo & Thả (Drag & Drop), một tính năng được Vue.js hỗ trợ rất tốt.



* Giao diện Tạo/Chỉnh sửa Công việc (Hộp thoại Modal)Khi tạo mới hoặc chỉnh sửa một công việc, một hộp thoại (modal) sẽ xuất hiện đè lên giao diện chính. Điều này giúp người dùng tập trung vào việc nhập liệu mà không bị phân tâm. Hộp thoại chứa đầy đủ các trường thông tin cần thiết như Tên, Mô tả, Người thực hiện, và Hạn hoàn thành.



* Giao diện Quản lý Plugin (Trang Quản trị): Đây là trang dành cho người quản trị (Admin) để quản lý các tính năng mở rộng của hệ thống. Người dùng có thể tải lên các file plugin (.jar) và quản lý trạng thái hoạt động của chúng. Giao diện này giữ nguyên bố cục chung để đảm bảo tính nhất quán của toàn bộ ứng dụng.



## Giao tiếp phần cứng

### Thiết bị hiển thị (Display Devices - Client Side)

* Đặc điểm vật lý: Hỗ trợ đa dạng các thiết bị hiển thị bao gồm màn hình máy tính (Desktop/Laptop), máy tính bảng (Tablet) và điện thoại thông minh (Smartphone).
* Đặc điểm logic (Rendering):

+ Độ phân giải: Giao diện được thiết kế theo nguyên lý Responsive Web Design.

* + - Tối ưu nhất trên độ phân giải chiều rộng ≥ 1280px (Laptop/Desktop) để hiển thị đầy đủ Bảng Kanban 3 cột.
    - Hỗ trợ hiển thị trên độ phân giải di động (Mobile viewport) nhưng sẽ ẩn bớt các cột hoặc chuyển sang dạng danh sách dọc (List View).

+ Màu sắc: Yêu cầu màn hình màu (sRGB) để phân biệt rõ các trạng thái công việc + (Xanh - To Do, Cam - In Progress, Xám - Done) và các thông báo lỗi (Đỏ).

### Thiết bị nhập liệu (Input Devices - Client Side)

* Bàn phím (Keyboard): Hỗ trợ nhập liệu văn bản và các phím tắt điều hướng (Ctrl+N, Esc, Enter).
* Thiết bị trỏ (Mouse/Trackpad):
* Sử dụng để thực hiện các thao tác Click, Hover (rê chuột).
* Quan trọng nhất là thao tác Drag & Drop (Kéo & Thả) để di chuyển thẻ công việc giữa các cột trạng thái.
* Màn hình cảm ứng (Touch Screen): Hỗ trợ các thao tác chạm (Tap) và vuốt (Swipe) trên các thiết bị máy tính bảng hoặc màn hình cảm ứng, thay thế cho chuột truyền thống.

### Thiết bị lưu trữ (Storage Devices - Server Side)

* Đọc (Read Operation):
* Hệ thống Backend yêu cầu quyền truy cập trực tiếp vào Ổ cứng (HDD/SSD) của máy chủ để quét và nạp các tệp tin .jar từ thư mục /plugins.
* Giao thức: Sử dụng java.io.File hoặc java.nio để tương tác với File System của hệ điều hành.
* Ghi (Write Operation):
* Ghi dữ liệu vào Cơ sở dữ liệu (Database files) nếu sử dụng database nhúng (H2/SQLite).
* Ghi nhật ký hệ thống (System Logs) để phục vụ việc bảo trì và gỡ lỗi.
* Lưu trữ phía Client (Browser Storage):
* Sử dụng bộ nhớ cục bộ của thiết bị người dùng (LocalStorage hoặc SessionStorage) để lưu trữ Token xác thực (JWT) và các cấu hình giao diện cá nhân (Theme, Language) nhằm giảm tải cho Server.

### Giao tiếp mạng (Network Interface - NIC)

 Máy chủ (Server): Yêu cầu cổng mạng hoạt động ổn định để lắng nghe các yêu cầu HTTP/HTTPS từ phía Client tại cổng cấu hình (ví dụ: Port 8080).

 Máy khách (Client): Yêu cầu kết nối mạng (Wifi/LAN/4G) để tải giao diện Vue.js và đồng bộ dữ liệu JSON thời gian thực.

 Giao thức: Phần mềm sử dụng ngăn xếp giao thức TCP/IP của hệ điều hành để truyền tải dữ liệu.

## Giao tiếp phần mềm

### Giao tiếp giữa Frontend (Vue.js) và Backend (Java)

* Giao thức (Protocol): HTTP/1.1 hoặc HTTP/2 qua kết nối bảo mật (HTTPS).
* Cơ chế trao đổi: Sử dụng chuẩn RESTful API.
* Frontend (Vue.js): Sử dụng thư viện HTTP Client (như Axios) để gửi các yêu cầu (GET, POST, PUT, DELETE) tới các Endpoints của Backend.
* Backend (Java): Các Controller (ví dụ: Servlet hoặc Spring Controller) tiếp nhận yêu cầu, xử lý nghiệp vụ và trả về kết quả.
* Định dạng dữ liệu:
* Request (Yêu cầu): Dữ liệu gửi đi đóng gói dạng JSON (Javascript Object Notation).
* Response (Phản hồi): Trả về JSON chứa dữ liệu nghiệp vụ hoặc mã lỗi HTTP (200 OK, 400 Bad Request, 500 Internal Server Error).

### Giao tiếp Nội bộ: Core Framework và Plugins (Backend)

* Cơ chế nạp (Loading Mechanism):
* Hệ thống Core sử dụng Java Reflection API và ClassLoader để quét thư mục /plugins/ và nạp động các file .jar vào thời gian chạy (Runtime).
* Hợp đồng giao tiếp (Interfaces):
* IPlugin: Mọi Plugin (bên thứ ba) bắt buộc phải hiện thực Interface này để Core có thể điều khiển vòng đời (Initialize, Start, Stop).
* IHostContext: Core cung cấp Interface này cho Plugin, cho phép Plugin gọi ngược (Callback) để truy xuất dữ liệu chung hoặc ghi log hệ thống.
* Cách thức tương tác:
* Khi Admin tải lên một file .jar từ giao diện Vue.js, Backend sẽ lưu file xuống đĩa cứng, kích hoạt PluginLoader để nạp class, và đăng ký các chức năng mới vào hệ thống mà không cần khởi động lại Server.

### Giao tiếp với Cơ sở dữ liệu (Database)

* Thành phần kết nối: Backend Java sử dụng JDBC Driver (hoặc thông qua các ORM Framework như Hibernate/JPA).
* Hệ quản trị CSDL: MySQL (hoặc H2/SQLite cho phiên bản nhúng).
* Dịch vụ:
* Thực hiện các truy vấn SQL (SELECT, INSERT, UPDATE, DELETE) để quản lý dữ liệu Dự án, Công việc và Người dùng.
* Quản lý Transaction (Giao dịch) để đảm bảo tính toàn vẹn dữ liệu (ví dụ: khi xóa Dự án thì phải xóa hết Công việc con).

### Giao tiếp với các Thư viện và Công cụ (Libraries & Tools)

* Phía Frontend (Vue.js Ecosystem):
* Vue Router: Quản lý điều hướng trang và lịch sử duyệt web (SPA Navigation).
* Pinia / Vuex: Quản lý trạng thái ứng dụng (State Management) để đồng bộ dữ liệu giữa các Component.
* Axios: Thư viện giao tiếp HTTP bất đồng bộ (Asynchronous) với Backend.
* Phía Backend (Java Ecosystem):
* Jackson / Gson: Thư viện chuyển đổi (Parsing) qua lại giữa Java Objects và JSON chuỗi để gửi cho Frontend.
* Log4j / SLF4J: Ghi nhận nhật ký hoạt động hệ thống và lỗi từ cả Core lẫn Plugin.

## Giao tiếp truyền thông tin

### Tổng quan

* Hệ thống quản lý công việc nhóm yêu cầu các chức năng giao tiếp và truyền thông tin để hỗ trợ việc trao đổi dữ liệu giữa người dùng với hệ thống, giữa các thành viên trong nhóm, và giữa hệ thống với các dịch vụ bên ngoài. Các chức năng truyền thông này đảm bảo việc cập nhật, đồng bộ và thông báo thông tin kịp thời, chính xác và an toàn.

### 3.4.2 Các hình thức giao tiếp truyền thông

**a. Giao tiếp giữa người dùng và hệ thống**

* Người dùng truy cập hệ thống thông qua trình duyệt web hoặc ứng dụng client.
* Giao tiếp được thực hiện thông qua giao thức HTTP/HTTPS.
* Các chức năng chính bao gồm:
* Đăng nhập/đăng xuất
* Tạo, chỉnh sửa, phân công công việc
* Trao đổi thông tin trong nhóm
* Theo dõi tiến độ và báo cáo

**b. Giao tiếp giữa các thành viên trong nhóm**

Hệ thống hỗ trợ:

* Gửi thông báo trong hệ thống (notification)
* Bình luận (comment) trên từng công việc
* Trao đổi tin nhắn nội bộ (nếu có)
* Dữ liệu được truyền thông qua máy chủ trung tâm theo mô hình Client–Server.

**c. Giao tiếp với các dịch vụ bên ngoài**

Hệ thống có thể tích hợp với:

* Thư điện tử (Email) để gửi thông báo, nhắc hạn công việc
* Các dịch vụ lịch (Calendar) hoặc lưu trữ đám mây
* Giao tiếp thông qua các API chuẩn RESTful.

### 3.4.3 Định dạng thông điệp

Dữ liệu truyền giữa client và server sử dụng định dạng:

* JSON (chủ yếu)
* XML (tùy chọn, nếu tích hợp hệ thống khác)

### 3.4.4 Chuẩn và giao thức truyền thông

Các chuẩn và giao thức được sử dụng bao gồm:

* HTTP/HTTPS: truyền dữ liệu giữa client và server
* SMTP: gửi thư điện tử thông báo
* RESTful API: giao tiếp với các hệ thống và dịch vụ bên ngoài
* WebSocket (nếu có): cập nhật trạng thái công việc và thông báo theo thời gian thực

### 3.4.5 Mã hóa và bảo mật truyền thông

* Toàn bộ dữ liệu truyền qua mạng sử dụng HTTPS (SSL/TLS) để mã hóa.
* Xác thực người dùng thông qua:
* Tên đăng nhập và mật khẩu được mã hóa
* Token xác thực (JWT hoặc tương đương)
* Phân quyền truy cập dữ liệu theo vai trò người dùng (quản lý, thành viên).
* Ngăn chặn truy cập trái phép và tấn công trung gian (Man-in-the-Middle).

### 3.4.6 Tỷ lệ truyền dữ liệu và hiệu năng

Hệ thống được thiết kế để:

* Xử lý đồng thời nhiều yêu cầu từ nhiều người dùng
* Tối ưu dung lượng dữ liệu truyền bằng cách nén (gzip)
* Dữ liệu truyền chủ yếu là văn bản và metadata, không yêu cầu băng thông lớn.

### 3.4.7 Đồng bộ dữ liệu

Hệ thống quản lý công việc nhóm hỗ trợ cơ chế đồng bộ dữ liệu nhằm đảm bảo thông tin luôn nhất quán, cập nhật kịp thời và chính xác giữa các người dùng, thiết bị và thành phần hệ thống.

**a. Mục tiêu đồng bộ dữ liệu**

* Đảm bảo mọi thay đổi (tạo, cập nhật, xóa công việc, bình luận, trạng thái) được phản ánh kịp thời tới các thành viên liên quan.
* Tránh xung đột dữ liệu khi nhiều người dùng thao tác đồng thời.
* Hỗ trợ người dùng làm việc trên nhiều thiết bị.
* Tăng trải nghiệm người dùng thông qua cập nhật gần thời gian thực.

**b. Phạm vi dữ liệu được đồng bộ**

* Thông tin công việc (task): tiêu đề, mô tả, trạng thái, người phụ trách, thời hạn.
* Bình luận và trao đổi trong công việc.
* Thông báo (notification).
* Tiến độ thực hiện và báo cáo.
* Thông tin phiên đăng nhập và trạng thái người dùng (online/offline – nếu có).

**c. Hình thức đồng bộ**

Hệ thống hỗ trợ các hình thức đồng bộ sau:

* Đồng bộ thời gian thực (Real-time synchronization).
* Đồng bộ gần thời gian thực (Near real-time).
* Đồng bộ khi đăng nhập lại.

**d. Kỹ thuật đồng bộ dữ liệu**

Hệ thống áp dụng một hoặc kết hợp các kỹ thuật sau:

* WebSocket.
* Polling định kỳ.
* Event-based synchronization.

**e. Xử lý xung đột dữ liệu**

* Hệ thống ưu tiên dữ liệu cập nhật mới nhất (last-write-wins) hoặc theo quyền người dùng.
* Cảnh báo người dùng khi xảy ra xung đột chỉnh sửa.
* Ghi nhận lịch sử thay đổi (log) để phục hồi khi cần thiết.

# Các tính năng của hệ thống

## Thêm dự án

### 4.1.1 Mô tả và mức ưu tiên

-Tính năng này cho phép người dùng có quyền quản lý (Manager) tạo lập một không gian làm việc mới (Project). Mỗi dự án sẽ đóng vai trò là một container chứa các công việc (tasks), tài liệu và danh sách thành viên tham gia.

- Mức độ ưu tiên: Cao

* Đây là tính năng nền tảng; nếu không có dự án, người dùng không thể tạo tác vụ hay gán quyền
* Đánh giá: Lợi ích (9/9) - Chi phí phát triển (3/9) - Rủi ro (2/9)

### 4.1.2 Tác nhân / Chuỗi đáp ứng

- Tác nhân (Manager): Tại màn hình Dashboard chính, người dùng nhấn vào nút chức năng “Tạo dự án mới” (+).

-Hệ thống: Hiển thị hộp thoại (Dialog) nhập thông tin dự án gồm: Tên dự án, Mô tả ngắn, Ngày bắt đầu và Ngày dự kiến kết thúc.

-Tác nhân: Nhập các thông tin cần thiết và nhấn nút “Lưu”.

-Hệ thống: Thực hiện kiểm tra tính hợp lệ của dữ liệu ( Validate):

* Nếu dữ liệu hợp lệ: Hệ thống lưu thông tin vào cơ sở dữ liệu, tạo ID dự án mới, đóng hộp thoại và hiển thị thông báo “Tạo dự án thành công”. Giao diện tự động chuyển hướng vào trang chi tiết của dự án vừa tạo.
* Nếu dữ liệu không hợp lệ ( ví dụ: Tên dự án bị trống): Hệ thống giữ nguyên hộp thoại và hiển thị dòng thông báo lỗi màu đỏ ngay dưới trường dữ liệu bị sai.

### 4.1.3 Các yêu cầu chức năng

- Các yêu cầu dưới đây quy định chi tiết về hành vi của hệ thống khi xử lý tính năng tạo dự án:

* REQ-1: Hệ thống phải cung cấp giao diện để nhập liệu thông tin dự án bao gồm các trường:
* Tên dự án (Bắt buộc, tối đa 100 ký tự).
* Mô tả dự án (Tùy chọn, văn bản nhiều dòng).
* Ngày bắt đầu (Bắt buộc, mặc định là ngày hiện tại),
* Ngày kết thúc (Tùy chọn)
* REQ-2: Hệ thống phải kiểm tra logic nghiệp vụ cho trường ngày tháng: "Ngày kết thúc" (nếu có) phải lớn hơn hoặc bằng "Ngày bắt đầu". Nếu vi phạm, hệ thống phải hiển thị thông báo lỗi: *"Ngày kết thúc không được nhỏ hơn ngày bắt đầu"*.
* REQ-3: Khi một dự án được tạo thành công, hệ thống phải tự động gán người dùng tạo dự án đó (Current User) vai trò là "Chủ dự án" (Project Owner) với toàn quyền quản trị.
* REQ-4: Hệ thống phải tự động khởi tạo 03 cột trạng thái công việc mặc định cho dự án mới, bao gồm: *"Cần làm" (To Do)*, *"Đang làm" (In Progress)*, và *"Đã xong" (Done)*.
* REQ-5: Hệ thống phải đảm bảo tên dự án là duy nhất trong phạm vi tài khoản của người quản lý đó. Nếu trùng tên, hệ thống yêu cầu nhập tên khác để tránh nhầm lẫn.
* REQ-6: (Liên quan đến Plugin) Hệ thống phải kích hoạt các Plugin mặc định (Core Plugins) cho dự án mới ngay khi khởi tạo, trừ khi người dùng cấu hình khác trong phần Cài đặt chung.

## Xóa dự án

### 4.2.1 Mô tả và mức ưu tiên

- Tính năng này cho phép người dùng có quyền Quản lý (Manager) loại bỏ hoàn toàn một dự án khỏi hệ thống. Hành động này sẽ xóa vĩnh viễn dự án cùng toàn bộ công việc (Tasks), tài liệu và dữ liệu liên quan được lưu trữ bởi Core và các Plugin.

- Mức độ ưu tiên: Trung bình (Medium).

* Tính năng này quan trọng để quản lý dung lượng và dọn dẹp hệ thống, nhưng ít được sử dụng thường xuyên hơn tính năng "Tạo" hoặc "Sửa".
* Đánh giá: Lợi ích (6/9) - Chi phí (4/9) - Rủi ro (9/9 - do liên quan đến mất mát dữ liệu vĩnh viễn).

### 4.2.2 Tác nhân / Chuỗi đáp ứng

Quá trình tương tác để xóa một dự án diễn ra như sau:

* Tác nhân (Manager): Tại màn hình "Cài đặt dự án" hoặc từ danh sách dự án trên Dashboard, người dùng chọn chức năng "Xóa dự án".
* Hệ thống: Hiển thị một hộp thoại cảnh báo (Confirmation Dialog) với nội dung nhấn mạnh về việc dữ liệu sẽ bị mất vĩnh viễn. Hệ thống yêu cầu người dùng xác nhận lại (ví dụ: nhấn nút "Đồng ý" hoặc nhập lại tên dự án để xác nhận).
* Tác nhân: Thực hiện hành động xác nhận.
* Hệ thống: Kiểm tra quyền hạn của người dùng.
* Nếu hợp lệ: Hệ thống tiến hành xóa dữ liệu logic trong Core, gửi tín hiệu xóa đến các Plugin liên quan, sau đó thông báo "Đã xóa thành công" và điều hướng người dùng về trang chủ.
* Nếu không hợp lệ (ví dụ: mất kết nối, không đủ quyền): Hiển thị thông báo lỗi.

### 4.2.3 Các yêu cầu chức năng

* REQ-1: Hệ thống chỉ cho phép người dùng có vai trò là "Chủ dự án" (Project Owner) hoặc "Quản trị viên hệ thống" (Admin) thực hiện quyền xóa dự án. Các thành viên (Member) không nhìn thấy nút này.
* REQ-2: Hệ thống bắt buộc phải hiển thị hộp thoại xác nhận (Confirmation Popup) trước khi thực hiện lệnh xóa. Thông báo phải ghi rõ: "Hành động này không thể hoàn tác. Toàn bộ công việc trong dự án sẽ bị xóa."
* REQ-3: (Yêu cầu về tính toàn vẹn dữ liệu - Cascade Delete) Khi xóa một dự án, hệ thống Core phải tự động xóa tất cả các thực thể con phụ thuộc vào dự án đó, bao gồm:
* Danh sách công việc (Task list).
* Các cột trạng thái (Status columns).
* Lịch sử hoạt động (Activity Logs) của dự án đó.
* REQ-4: (Yêu cầu tích hợp Plugin) Trước khi hoàn tất việc xóa trong Core, hệ thống phải phát đi một sự kiện (Event) hoặc gọi phương thức onProjectDelete(String projectId) trong Interface IPlugin. Điều này cho phép các Plugin đã cài đặt (ví dụ: Plugin lưu trữ file đính kèm) tự động dọn dẹp dữ liệu rác liên quan đến dự án bị xóa.
* REQ-5: Sau khi xóa thành công, hệ thống phải tự động làm mới (refresh) danh sách dự án ở thanh Sidebar để dự án vừa xóa biến mất ngay lập tức mà không cần người dùng tải lại trang.

## Chỉnh sửa thông tin dự án

### 4.3.1 Mô tả và mức ưu tiên

Tính năng này cho phép người dùng có quyền Quản lý (Manager/Owner) cập nhật lại các thông tin meta-data của một dự án đã tồn tại, như tên gọi, mô tả phạm vi, hoặc điều chỉnh thời gian thực hiện (deadline) khi kế hoạch thay đổi.

* Mức độ ưu tiên: Cao (High).
* Trong thực tế quản lý dự án, việc thay đổi phạm vi hoặc gia hạn thời gian xảy ra rất thường xuyên.
* Đánh giá: Lợi ích (8/9) - Chi phí (3/9) - Rủi ro (3/9).

### 4.3.2 Tác nhân / Chuỗi đáp ứng

Quá trình tương tác để cập nhật thông tin dự án diễn ra như sau:

* Tác nhân (Manager): Truy cập vào trang chi tiết của dự án hoặc danh sách dự án, chọn nút "Cài đặt" (Settings) hoặc biểu tượng "Chỉnh sửa" (Edit) tương ứng với dự án cần sửa.
* Hệ thống: Hiển thị form chứa thông tin hiện tại của dự án. Các trường dữ liệu cũ được điền sẵn (pre-filled) để người dùng tiện theo dõi.
* Tác nhân: Thực hiện thay đổi nội dung mong muốn (ví dụ: dời ngày kết thúc) và nhấn nút "Lưu thay đổi".
* Hệ thống: Kiểm tra tính hợp lệ của dữ liệu mới (Validation).
* Nếu hợp lệ: Hệ thống cập nhật bản ghi trong cơ sở dữ liệu, ghi lại nhật ký thay đổi, và hiển thị thông báo "Cập nhật thành công". Giao diện người dùng lập tức hiển thị thông tin mới.
* Nếu không hợp lệ (ví dụ: Ngày kết thúc mới lại nhỏ hơn ngày bắt đầu cũ): Hệ thống từ chối lưu và hiển thị thông báo lỗi cụ thể.

### 4.3.3 Các yêu cầu chức năng

Các yêu cầu dưới đây quy định cụ thể về phạm vi dữ liệu được phép sửa và các ràng buộc logic:

* REQ-1: Hệ thống cho phép người dùng chỉnh sửa các trường thông tin sau:
* Tên dự án.
* Mô tả dự án.
* Trạng thái dự án (ví dụ: Chuyển từ "Đang chạy" sang "Tạm dừng" hoặc "Đóng").
* Ngày kết thúc dự kiến.
* REQ-2: Hệ thống KHÔNG cho phép chỉnh sửa "Ngày bắt đầu" nếu dự án đã có các công việc (Tasks) phát sinh hoặc đã hoạt động quá 24 giờ, nhằm đảm bảo tính toàn vẹn của lịch sử tiến độ. (Hoặc có thể cho phép nhưng phải hiện cảnh báo).
* REQ-3: (Ràng buộc nghiệp vụ) Nếu người dùng thay đổi "Ngày kết thúc" của dự án sớm hơn trước đó, hệ thống phải kiểm tra xem có công việc con (Task) nào có hạn chót (Deadline) nằm ngoài ngày kết thúc mới hay không.
* Nếu có: Hệ thống phải hiển thị cảnh báo danh sách các công việc bị ảnh hưởng và yêu cầu người dùng xác nhận hoặc điều chỉnh lại các công việc đó.
* REQ-4: Hệ thống chỉ cho phép người dùng có vai trò "Chủ dự án" (Owner) thực hiện việc chỉnh sửa. Các thành viên khác chỉ có quyền xem (Read-only).
* REQ-5: (Audit Log) Mọi thao tác chỉnh sửa thông tin dự án phải được hệ thống tự động ghi lại vào "Nhật ký hoạt động" (Activity Log) của dự án đó, bao gồm: Người sửa, Thời gian sửa, Dữ liệu cũ -> Dữ liệu mới.
* REQ-6: (Real-time Update) Khi thông tin dự án được cập nhật thành công, hệ thống phải gửi tín hiệu thời gian thực (qua WebSocket) để cập nhật lại tiêu đề/thông tin trên màn hình của tất cả các thành viên đang online trong dự án đó mà không cần họ phải tải lại trang (F5).

## Chức năng đăng ký tài khoản

### 4.4.1 Mô tả và mức ưu tiên

* Mô tả: Tính năng này cho phép người dùng mới (chưa có tài khoản) cung cấp các thông tin cá nhân cần thiết để tạo lập một danh tính trong hệ thống. Việc đăng ký thành công sẽ cấp cho người dùng quyền truy cập vào ứng dụng với vai trò mặc định (thường là Member), từ đó có thể tham gia vào các dự án và nhận công việc.
* Mức độ ưu tiên: Cao (High). Đây là chức năng thiết yếu đầu vào; nếu không có tính năng này, người dùng không thể truy cập và sử dụng các chức năng cộng tác của phần mềm.

### 4.4.2 Tác nhân / Chuỗi đáp ứng

* Tác nhân (Actor): Người dùng khách (Guest) - người chưa có tài khoản hoặc chưa đăng nhập vào hệ thống.
* Chuỗi đáp ứng (Stimulus/Response Sequences):
* Hành động: Người dùng chọn liên kết hoặc nút "Đăng ký" từ màn hình Đăng nhập.
* Đáp ứng: Hệ thống hiển thị Biểu mẫu Đăng ký bao gồm các trường: Tên đăng nhập, Mật khẩu, Xác nhận mật khẩu, Họ và tên, Email.
* Hành động: Người dùng nhập thông tin vào các trường và nhấn nút "Hoàn tất đăng ký".
* Đáp ứng: Hệ thống thực hiện kiểm tra (validate) dữ liệu đầu vào.
* Trường hợp thành công: Hệ thống tạo mới bản ghi người dùng trong cơ sở dữ liệu, hiển thị thông báo "Đăng ký thành công" và chuyển hướng người dùng về màn hình Đăng nhập.
* Trường hợp thất bại: Hệ thống giữ nguyên màn hình và hiển thị thông báo lỗi cụ thể (ví dụ: "Tên đăng nhập đã tồn tại", "Mật khẩu không khớp") ngay tại trường dữ liệu bị lỗi.

### 4.4.3 Các yêu cầu chức năng

* Dưới đây là các yêu cầu chi tiết để thực hiện tính năng đăng ký:
* REQ-1: Hệ thống phải cung cấp giao diện biểu mẫu đăng ký gồm các trường thông tin bắt buộc: Tên đăng nhập (Username), Mật khẩu (Password), Nhập lại mật khẩu (Confirm Password), và Email.
* REQ-2: Hệ thống phải kiểm tra tính duy nhất của Tên đăng nhập và Email. Nếu người dùng nhập giá trị đã tồn tại trong hệ thống, hệ thống phải báo lỗi và yêu cầu nhập lại.
* REQ-3: Hệ thống phải yêu cầu độ dài mật khẩu tối thiểu (ví dụ: 6 ký tự) để đảm bảo tính bảo mật cơ bản.
* REQ-4: Hệ thống phải so sánh chuỗi ký tự ở trường "Mật khẩu" và "Nhập lại mật khẩu". Nếu không trùng khớp, hệ thống phải ngăn chặn việc gửi dữ liệu và thông báo cho người dùng.
* REQ-5: Hệ thống phải gán vai trò mặc định là Member (Thành viên) cho tài khoản mới đăng ký. (Vai trò Admin hoặc Manager sẽ được cấp hoặc nâng quyền bởi Quản trị viên sau đó để đảm bảo an toàn) .
* REQ-6: Hệ thống phải mã hóa mật khẩu (hashing) trước khi lưu trữ vào cơ sở dữ liệu hoặc file hệ thống, không được lưu dưới dạng văn bản thuần (plain text) để đảm bảo yêu cầu an toàn.
* REQ-7: Hệ thống phải cung cấp nút "Hủy bỏ" hoặc "Quay lại" để người dùng có thể thoát khỏi màn hình đăng ký và trở về màn hình đăng nhập mà không tạo tài khoản.

## Chức năng đăng nhập

### 4.5.1 Mô tả và mức ưu tiên

* Mô tả: Tính năng đăng nhập cho phép người dùng đã có tài khoản xác thực danh tính của mình với hệ thống. Sau khi xác thực thành công, hệ thống sẽ cấp quyền truy cập vào các tài nguyên và chức năng tương ứng với vai trò của người dùng (Admin, Manager, hoặc Member). Đây là cổng kiểm soát an ninh chính của ứng dụng.
* Mức độ ưu tiên: Cao (High). Đây là tính năng bắt buộc phải hoạt động chính xác để đảm bảo tính bảo mật và phân quyền của hệ thống.

### 4.5.2 Tác nhân / Chuỗi đáp ứng

* Tác nhân (Actor): Người dùng (User) - bao gồm cả Admin, Manager và Member khi chưa đăng nhập.
* Chuỗi đáp ứng (Stimulus/Response Sequences):
* Hành động: Người dùng mở ứng dụng hoặc chọn "Đăng nhập".
* Đáp ứng: Hệ thống hiển thị Màn hình Đăng nhập với các trường nhập liệu cho Tên đăng nhập và Mật khẩu.
* Hành động: Người dùng nhập thông tin xác thực và nhấn nút "Đăng nhập".
* Đáp ứng: Hệ thống kiểm tra thông tin trong cơ sở dữ liệu.
* *Trường hợp thành công:* Hệ thống xác định vai trò người dùng, tải ngữ cảnh (Context) phù hợp và chuyển hướng vào màn hình chính (Dashboard).
* *Trường hợp thất bại:* Hệ thống hiển thị thông báo lỗi "Tên đăng nhập hoặc mật khẩu không đúng" và yêu cầu nhập lại.

### 4.5.3 Các yêu cầu chức năng

Các yêu cầu chi tiết để thực hiện tính năng đăng nhập bao gồm:

* REQ-1: Hệ thống phải cung cấp giao diện đăng nhập gồm hai trường bắt buộc: Tên đăng nhập (Username) và Mật khẩu (Password).
* REQ-2: Hệ thống phải che giấu các ký tự trong trường Mật khẩu (hiển thị dạng dấu chấm hoặc sao) để đảm bảo riêng tư khi nhập liệu.
* REQ-3: Hệ thống phải thực hiện xác thực (Authentication) bằng cách so sánh tên đăng nhập và mã hóa (hash) mật khẩu nhập vào với dữ liệu đã lưu trữ.
* REQ-4: Nếu thông tin đăng nhập đúng, hệ thống phải truy xuất Vai trò (Role) của người dùng (Admin, Manager, Member) để thiết lập quyền hạn cho phiên làm việc hiện tại (ví dụ: Admin thấy nút quản lý người dùng, Member chỉ thấy dự án của mình).
* REQ-5: Hệ thống phải hiển thị thông báo lỗi chung chung (ví dụ: "Thông tin đăng nhập không chính xác") nếu tên đăng nhập không tồn tại hoặc mật khẩu sai, nhằm tránh việc kẻ xấu dò tìm tên tài khoản.
* REQ-6: Hệ thống phải cung cấp liên kết hoặc nút chuyển hướng sang màn hình Đăng ký cho người dùng chưa có tài khoản.
* REQ-7: (Tùy chọn nâng cao) Hệ thống nên hỗ trợ chức năng "Ghi nhớ đăng nhập" (Remember Me) để lưu thông tin phiên làm việc cho lần truy cập sau (nếu chính sách bảo mật cho phép).

## 4.6 Chức năng Quản lý và Phân quyền Người dùng:

### 4.6.1 Mô tả và mức ưu tiên

* Mô tả: Tính năng này cung cấp cho Quản trị viên (Administrator) quyền kiểm soát toàn bộ tài khoản người dùng trong hệ thống. Admin có thể xem danh sách thành viên, tìm kiếm và quan trọng nhất là thực hiện phân quyền (nâng cấp hoặc hạ cấp vai trò giữa Member và Manager) cũng như khóa các tài khoản vi phạm.
* Mức độ ưu tiên: Trung bình (Medium). Mặc dù không ảnh hưởng trực tiếp đến việc tạo công việc hàng ngày, nhưng đây là tính năng thiết yếu để duy trì trật tự và cấu trúc quản lý của tổ chức.

### 4.6.2 Tác nhân / Chuỗi đáp ứng

* Tác nhân (Actor): Quản trị viên (Administrator).
* Chuỗi đáp ứng (Stimulus/Response Sequences):
* Hành động: Admin chọn menu "Quản lý người dùng" từ Dashboard.
* Đáp ứng: Hệ thống hiển thị danh sách tất cả người dùng với các thông tin tóm tắt (Tên, Email, Vai trò hiện tại, Trạng thái).
* Hành động: Admin tìm kiếm một người dùng cụ thể và chọn "Chỉnh sửa" hoặc thay đổi vai trò trên giao diện.
* Đáp ứng: Hệ thống hiển thị hộp thoại xác nhận hoặc cập nhật trực tiếp.
* Hành động: Admin xác nhận thay đổi (ví dụ: chuyển từ Member sang Manager).
* Đáp ứng: Hệ thống cập nhật quyền hạn mới vào cơ sở dữ liệu và thông báo "Cập nhật thành công". Người dùng được cấp quyền sẽ thấy thay đổi này trong lần đăng nhập tiếp theo.

### Các yêu cầu chức năng

Các yêu cầu chi tiết cho việc quản lý và phân quyền:

* REQ-1: Hệ thống phải cho phép Admin xem danh sách toàn bộ người dùng đã đăng ký, bao gồm các cột thông tin: ID, Tên đăng nhập, Họ tên, Email, Vai trò (Role) và Trạng thái (Active/Locked).
* REQ-2: Hệ thống phải cung cấp công cụ tìm kiếm hoặc lọc người dùng theo Tên hoặc Email để dễ dàng quản lý khi số lượng tài khoản lớn.
* REQ-3: Hệ thống phải cho phép Admin thay đổi vai trò của một tài khoản. Cụ thể:
* Nâng quyền Member lên Manager (để họ có thể tạo và quản lý dự án).
* Hạ quyền Manager xuống Member.
* Bổ nhiệm Admin khác (tùy chọn theo chính sách).
* REQ-4: Hệ thống phải cho phép Admin thay đổi trạng thái hoạt động của tài khoản:
* Khóa (Lock/Ban): Ngăn người dùng đăng nhập.
* Kích hoạt (Unlock/Active): Mở lại quyền truy cập cho tài khoản bị khóa.
* REQ-5: Hệ thống phải ngăn chặn việc Admin tự tước quyền quản trị của chính mình hoặc tự khóa tài khoản của chính mình để tránh lỗi không còn ai quản lý hệ thống.
* REQ-6: Hệ thống phải ghi lại nhật ký (Log) khi có sự thay đổi về vai trò người dùng (ví dụ: "Admin A đã cấp quyền Manager cho User B") để phục vụ việc tra cứu lịch sử.

## Tính năng Nhận thông báo

### 4.7.1 Mô tả và mức ưu tiên

* Mô tả: Tính năng này giúp người dùng nắm bắt kịp thời các sự kiện quan trọng liên quan đến công việc và dự án của mình. Hệ thống sẽ tự động sinh ra thông báo trong các trường hợp: được phân công công việc mới, trạng thái công việc thay đổi, có bình luận mới, hoặc khi công việc sắp đến hạn chót (deadline).
* Mức độ ưu tiên: Cao (High). Đây là tính năng cốt lõi giúp duy trì sự cộng tác liên tục. Nếu thiếu tính năng này, người dùng buộc phải kiểm tra thủ công từng dự án để biết thay đổi, làm giảm đáng kể hiệu suất làm việc nhóm.

### Tác nhân / Chuỗi đáp ứng

* Tác nhân (Actor): Người dùng đã đăng nhập (Member, Manager), Hệ thống (Bộ đếm thời gian - System Timer).
* Chuỗi đáp ứng (Stimulus/Response Sequences):
* Trường hợp 1: Sự kiện từ người dùng khác
* Hành động: Quản lý gán một công việc cho Thành viên A (hoặc thay đổi trạng thái công việc của A).
* Đáp ứng: Hệ thống xác định Thành viên A là đối tượng nhận tin. Hệ thống tạo một bản ghi thông báo và hiển thị tín hiệu (ví dụ: chấm đỏ hoặc số lượng tin chưa đọc) trên biểu tượng chuông tại Dashboard của Thành viên A.
* Trường hợp 2: Sự kiện từ thời gian (Deadline)
* Hành động: Luồng nền (Background Thread/Timer) của hệ thống quét các công việc có deadline sắp đến (ví dụ: còn dưới 24 giờ).
* Đáp ứng: Hệ thống tự động gửi thông báo cảnh báo đến người phụ trách công việc đó mà không cần tác động của con người.
* Trường hợp 3: Xem thông báo
* Hành động: Người dùng nhấp vào biểu tượng thông báo.
* Đáp ứng: Hệ thống hiển thị danh sách các thông báo mới nhất.
* Hành động: Người dùng chọn một thông báo cụ thể.
* Đáp ứng: Hệ thống đánh dấu thông báo là "Đã đọc" và điều hướng người dùng đến màn hình chi tiết của công việc liên quan.

### Các yêu cầu chức năng

Các yêu cầu chi tiết để thực hiện tính năng thông báo:

* REQ-1: Hệ thống phải tự động tạo thông báo gửi đến người dùng liên quan trong các tình huống sau:
* Khi người dùng được gán vào một công việc mới.
* Khi trạng thái công việc họ đang làm bị thay đổi (ví dụ: từ "In Progress" sang "Done" hoặc bị "Reject").
* Khi có bình luận mới trong công việc mà họ tham gia.
* REQ-2: Hệ thống phải có cơ chế quét thời gian thực (hoặc định kỳ) để kiểm tra hạn chót. Hệ thống phải gửi thông báo cảnh báo trước thời điểm deadline một khoảng thời gian quy định (ví dụ: trước 24 giờ và trước 1 giờ).
* REQ-3: Giao diện Dashboard phải có khu vực hiển thị thông báo (ví dụ: Icon cái chuông trên thanh Menu) và hiển thị trực quan số lượng thông báo chưa đọc (Unread Badge).
* REQ-4: Danh sách thông báo phải hiển thị tóm tắt nội dung: Tiêu đề công việc, Loại sự kiện (Deadline/Assign/Comment), Thời gian nhận thông báo.
* REQ-5: Hệ thống phải cho phép người dùng nhấp vào thông báo để đi thẳng tới màn hình chi tiết của công việc đó (Deep linking trong ứng dụng).
* REQ-6: Hệ thống phải hỗ trợ phân biệt trạng thái "Đã đọc" và "Chưa đọc". Khi người dùng nhấp vào xem, thông báo phải chuyển sang trạng thái "Đã đọc".
* REQ-7: (Yêu cầu mở rộng cho Plugin) Hệ thống cần thiết kế Interface để hỗ trợ việc gửi thông báo qua Email trong tương lai (thông qua Plugin mở rộng) mà không cần sửa code cốt lõi.

## Chức năng tạo công việc

### 4.8.1 Mô tả và mức ưu tiên

- Mô tả: Chức năng tạo công việc cho phép người dùng tạo mới một nhiệm vụ trong hệ thống để phân công cho các thành viên trong nhóm. Khi tạo công việc, người dùng có thể nhập các thông tin như: tên công việc, mô tả chi tiết, người thực hiện, thời hạn hoàn thành, mức độ ưu tiên, trạng thái ban đầu và các tài liệu liên quan (nếu có). Chức năng này giúp nhóm tổ chức công việc một cách khoa học, tránh bỏ sót nhiệm vụ và đảm bảo các thành viên nắm rõ trách nhiệm của mình.

- Mức độ ưu tiên: Cao. Là chức năng cốt lõi của hệ thống. Nếu không có chức năng tạo công việc thì hệ thống sẽ không thể vận hành đúng mục đích quản lý công việc nhóm.

### 4.8.2 Tác nhân / Chuỗi đáp ứng

-Tác nhân (Actor): Trưởng nhóm, thành viên được cấp quyền tạo công việc.

Chuỗi đáp ứng (Stimulus/Response Sequences):

* Hành động: Người dùng chọn menu “Tạo công việc” từ trang quản lý dự án.  
  Đáp ứng: Hệ thống hiển thị biểu mẫu tạo công việc với các trường thông tin cần thiết (Tên công việc, Mô tả, Người phụ trách, Thời hạn, Mức độ ưu tiên,…).
* Hành động: Người dùng nhập thông tin công việc và nhấn nút “Lưu”.  
  Đáp ứng: Hệ thống kiểm tra tính hợp lệ của dữ liệu (trường bắt buộc, định dạng ngày tháng,…).
* Hành động: Người dùng xác nhận tạo công việc.  
  Đáp ứng: Hệ thống lưu công việc vào cơ sở dữ liệu và hiển thị thông báo “Tạo công việc thành công”.
* Hành động: Công việc được tạo thành công.  
  Đáp ứng: Hệ thống gửi thông báo đến người được phân công và cập nhật danh sách công việc.

### Các yêu cầu chức năng

* REQ-1: Hệ thống phải cung cấp giao diện để người dùng tạo mới công việc.
* REQ-2: Hệ thống phải yêu cầu nhập các trường bắt buộc: tên công việc, người thực hiện, hạn hoàn thành.
* REQ-3: Hệ thống phải cho phép nhập mô tả chi tiết cho công việc.
* REQ-4: Hệ thống phải kiểm tra định dạng ngày tháng hợp lệ.
* REQ-5: Hệ thống phải thông báo lỗi khi thiếu hoặc sai thông tin.
* REQ-6: Hệ thống phải lưu công việc vào cơ sở dữ liệu sau khi tạo thành công.
* REQ-7: Hệ thống phải gửi thông báo cho người được giao việc.
* REQ-8: Hệ thống phải không cho phép tạo công việc trùng tên trong cùng một dự án (TBD).

## Chức năng xem công việc

### 4.9.1 Mô tả và mức ưu tiên

- Mô tả: Chức năng xem công việc cho phép người dùng theo dõi danh sách các công việc trong nhóm theo từng dự án, từng thành viên hoặc từng trạng thái. Người dùng có thể xem thông tin tổng quát và chi tiết của từng công việc. Thông qua chức năng này, người dùng nắm được tiến độ chung của nhóm, từ đó điều chỉnh kế hoạch làm việc phù hợp.

* Mức độ ưu tiên: Cao. Việc theo dõi công việc là nhu cầu thường xuyên của người dùng. Nếu không xem được công việc, người dùng sẽ không biết mình cần làm gì và tiến độ hiện tại ra sao.

### 4.9.2 Tác nhân / Chuỗi đáp ứng

Tác nhân (Actor): Trưởng nhóm, thành viên nhóm.Chuỗi đáp ứng (Stimulus/Response Sequences):

* Hành động: Người dùng chọn menu “Danh sách công việc” từ trang chính.  
  Đáp ứng: Hệ thống hiển thị danh sách các công việc kèm thông tin tóm tắt (Tên công việc, Người phụ trách, Trạng thái, Hạn hoàn thành).
* Hành động: Người dùng sử dụng chức năng tìm kiếm hoặc bộ lọc.  
  Đáp ứng: Hệ thống hiển thị danh sách công việc phù hợp với điều kiện tìm kiếm.
* Hành động: Người dùng chọn một công việc cụ thể.  
  Đáp ứng: Hệ thống hiển thị trang chi tiết công việc với đầy đủ thông tin liên quan.
* Hành động: Người dùng quay lại danh sách hoặc chuyển sang công việc khác.  
  Đáp ứng: Hệ thống cập nhật giao diện theo thao tác của người dùng.

### 4.9.3 Các yêu cầu chức năng

* REQ-1: Hệ thống phải hiển thị danh sách công việc theo dự án hoặc nhóm.
* REQ-2: Hệ thống phải cho phép lọc công việc theo trạng thái, người thực hiện, thời hạn.
* REQ-3: Hệ thống phải hỗ trợ tìm kiếm công việc theo tên.
* REQ-4: Hệ thống phải hiển thị thông tin chi tiết: mô tả, trạng thái, hạn, người phụ trách.
* REQ-5: Hệ thống phải cập nhật trạng thái công việc khi có thay đổi.
* REQ-6: Hệ thống phải thông báo khi không có dữ liệu phù hợp.
* REQ-7: Hệ thống phải phân quyền xem công việc theo vai trò người dùng.

## Chức năng bình luận công việc

### 4.10.1 Mô tả và mức ưu tiên

- Mô tả: Chức năng bình luận công việc cho phép các thành viên trao đổi ý kiến, cập nhật tiến độ, hỏi đáp và phản hồi trực tiếp trên từng công việc. Tất cả nội dung trao đổi được lưu trữ tập trung giúp nhóm dễ dàng theo dõi lịch sử làm việc.

- Mức độ ưu tiên: Trung bình. Chức năng này hỗ trợ giao tiếp trong nhóm, giúp nâng cao hiệu quả phối hợp. Tuy nhiên, hệ thống vẫn có thể hoạt động cơ bản mà chưa cần chức năng này.

### 4.10.2 Tác nhân / Chuỗi đáp ứng

Tác nhân (Actor): Trưởng nhóm, thành viên nhóm. Chuỗi đáp ứng (Stimulus/Response Sequences):

* Hành động: Người dùng mở trang chi tiết một công việc.  
  Đáp ứng: Hệ thống hiển thị nội dung công việc và khu vực bình luận.
* Hành động: Người dùng nhập nội dung bình luận vào ô nhập liệu.  
  Đáp ứng: Hệ thống hiển thị nội dung đang được soạn thảo.
* Hành động: Người dùng nhấn nút “Gửi”.  
  Đáp ứng: Hệ thống kiểm tra nội dung bình luận (không rỗng, không vi phạm quy định).
* Hành động: Người dùng xác nhận gửi bình luận.  
  Đáp ứng: Hệ thống lưu bình luận vào cơ sở dữ liệu và hiển thị ngay trên giao diện.
* Hành động: Có bình luận mới được thêm vào.  
  Đáp ứng: Hệ thống gửi thông báo đến các thành viên liên quan và cập nhật lịch sử trao đổi.

### Các yêu cầu chức năng

* REQ-16: Hệ thống phải cho phép người dùng thêm bình luận.
* REQ-17: Hệ thống phải hiển thị tên người gửi và thời gian gửi.
* REQ-18: Hệ thống phải không cho phép gửi bình luận rỗng.
* REQ-19: Hệ thống phải cho phép người dùng chỉnh sửa bình luận của mình (TBD).
* REQ-20: Hệ thống phải cho phép xóa bình luận của mình.
* REQ-21: Hệ thống phải lưu trữ toàn bộ lịch sử trao đổi.
* REQ-22: Hệ thống phải gửi thông báo khi có bình luận mới.

## Chức năng cập nhật trạng thái công việc

### 4.11.1 Mô tả và mức ưu tiên

- Mô tả:Cho phép người dùng thay đổi trạng thái tiến độ của một nhiệm vụ cụ thể (To Do, In Progress, Done, Cancel) để phản ánh quá trình thực hiện thực tế. Trạng thái này là cơ sở để hệ thống tính toán tiến độ dự án và gửi thông báo nhắc nhở.

- Mức độ ưu tiên: Cao. Đây là tính năng cốt lõi để theo dõi tiến độ và kích hoạt các luồng thông báo tự động

### 4.11.2 Tác nhân / Chuỗi đáp ứng

* Tác nhân (Actor): Member (người được gán việc), Manager (quản lý dự án).
* Chuỗi đáp ứng (Stimulus/Response Sequences):
* Hành động: Người dùng chọn một công việc cụ thể trên bảng Kanban hoặc danh sách.
* Hành động: Người dùng chọn trạng thái mới từ danh sách thả xuống hoặc kéo thả thẻ công việc sang cột trạng thái khác.
* Đáp ứng: Hệ thống thực hiện kiểm tra (validate) quyền hạn và logic nghiệp vụ.
  + - Trường hợp thành công: Hệ thống cập nhật trạng thái mới vào cơ sở dữ liệu , ghi lại nhật ký thay đổi (Audit Log) , hiển thị thông báo "Cập nhật thành công" và gửi thông báo cho các thành viên liên quan (quan hệ <<extend>> Nhận thông báo).
    - Trường hợp thất bại: Hệ thống từ chối thay đổi, giữ nguyên trạng thái cũ và hiển thị lỗi cụ thể (ví dụ: "Bạn không có quyền cập nhật công việc này" hoặc "Công việc đã hoàn thành không thể hủy").

### 4.11.3 Các yêu cầu chức năng

* REQ-UP-1: Hệ thống phải cung cấp giao diện tương tác kéo-thả (Drag & Drop) để người dùng thay đổi trạng thái nhanh chóng trên bảng Kanban.
* REQ-UP-2: Hệ thống chỉ cho phép người được gán việc (Assignee) hoặc Quản lý (Manager/Owner) thực hiện việc thay đổi trạng thái (tuân thủ quy tắc BR-3).
* REQ-UP-3: Khi trạng thái chuyển sang "Done", hệ thống phải tự động kiểm tra và khóa quyền chỉnh sửa các trường thông tin khác của công việc đó (tuân thủ quy tắc BR-3).
* REQ-UP-4: Hệ thống phải đảm bảo việc cập nhật trạng thái diễn ra trên luồng riêng biệt (Background Thread) để không gây treo giao diện người dùng (tuân thủ yêu cầu PER-2).
* REQ-UP-5: Hệ thống phải kích hoạt sự kiện thông báo (Event-based) ngay lập tức tới module "Nhận thông báo" khi trạng thái được lưu thành công.
* REQ-UP-6: Mọi thao tác thay đổi trạng thái phải được hệ thống tự động ghi lại vào "Nhật ký hoạt động" (Activity Log) gồm: Người sửa, Trạng thái cũ -> Trạng thái mới, và Thời gian sửa.

## Chức năng phân quyền dự án

### 4.12.1 Mô tả và mức ưu tiên

- Mô tả: Cho phép Quản lý dự án (Manager) hoặc Chủ dự án (Owner) thiết lập và điều chỉnh vai trò của các thành viên trong phạm vi một dự án cụ thể. Việc này nhằm kiểm soát quyền truy cập, thao tác dữ liệu và đảm bảo tính bảo mật dựa trên mô hình phân quyền theo vai trò (RBAC)

- Mức độ ưu tiên: Cao. Đảm bảo tính bảo mật, đúng trách nhiệm và ngăn chặn các truy cập trái phép vào tài nguyên dự án.

### 4.12.2 Tác nhân / Chuỗi đáp ứng

- Tác nhân (Actor): Manager (Quản lý dự án/Chủ dự án).

- Chuỗi đáp ứng (Stimulus/Response Sequences):

* Hành động: Manager truy cập vào trang chi tiết dự án, chọn mục "Cài đặt" hoặc biểu tượng "Thành viên".
* Hành động: Manager chọn một thành viên từ danh sách và chỉ định vai trò mới (ví dụ: nâng quyền từ Member lên Manager dự án).
* Hành động: Manager nhấn "Xác nhận" để lưu thay đổi.
* Đáp ứng: Hệ thống thực hiện kiểm tra quyền hạn và logic nghiệp vụ.
* Trường hợp thành công: Hệ thống cập nhật quyền hạn mới vào cơ sở dữ liệu , ghi lại lịch sử vào nhật ký hoạt động (Audit Log) , và hiển thị thông báo "Cập nhật quyền thành công".
* Trường hợp thất bại: Nếu người thực hiện không phải Owner hoặc Admin, hệ thống từ chối yêu cầu và hiển thị thông báo lỗi: "Bạn không có quyền phân quyền cho dự án này".

### 4.12.3 Các yêu cầu chức năng

* REQ-AUTH-1: Hệ thống phải thực thi cơ chế phân quyền dựa trên vai trò (RBAC) nghiêm ngặt tại cả lớp Giao diện (UI) và lớp Nghiệp vụ (Business Logic).
* REQ-AUTH-2: Manager chỉ được phép phân quyền trong phạm vi các dự án mà họ sở hữu hoặc được chỉ định quản lý (tuân thủ quy tắc BR-1).
* REQ-AUTH-3: Hệ thống phải tự động ẩn hoặc vô hiệu hóa các tính năng nhạy cảm (như "Xóa dự án", "Cài đặt hệ thống") đối với người dùng không có vai trò phù hợp.
* REQ-AUTH-4: Mọi thao tác thay đổi vai trò người dùng phải được ghi lại tự động vào "Nhật ký hoạt động" (Activity Log), bao gồm: Người thực hiện, Người được sửa, Vai trò cũ -> Vai trò mới, và Thời gian.
* REQ-AUTH-5: (Tính an toàn) Hệ thống phải ngăn chặn việc người quản lý tự tước quyền quản trị của chính mình trong dự án nếu dự án đó chỉ còn một Manager.

## 4.13 Chức năng xem thống kê

### 4.13.1 Mô tả và mức ưu tiên

- Mô tả: Chức năng này cho phép người dùng tổng hợp dữ liệu từ các dự án và công việc để cung cấp cái nhìn trực quan về hiệu suất và tiến độ thông qua các biểu đồ (hình tròn, hình cột). Chức năng được thiết kế dưới dạng Plugin để có thể nạp/rút linh hoạt mà không ảnh hưởng đến mã nguồn lõi.

- Mức độ ưu tiên: Trung bình . Cung cấp dữ liệu quan trọng để Quản lý điều chỉnh kế hoạch và nguồn lực kịp thời.

### 4.13.2 Tác nhân / Chuỗi đáp ứng

**-** Tác nhân (Actor): Manager (Quản lý), Admin (Quản trị viên), Member (Thành viên).

- Chuỗi đáp ứng (Stimulus/Response Sequences):

* Hành động: Người dùng chọn tính năng "Thống kê" từ Dashboard chính hoặc từ menu của một dự án cụ thể.
* Đáp ứng: Hệ thống nhận diện yêu cầu và kích hoạt Plugin Thống kê thông qua bộ nạp PluginLoader.
* Hành động: Hệ thống khởi tạo một tiến trình chạy ngầm (SwingWorker) để truy vấn và tính toán số liệu từ dự án mà không làm khóa giao diện chính (non-blocking UI).
* Đáp ứng:
  + Trường hợp có dữ liệu: Hệ thống hiển thị các biểu đồ tỷ lệ trạng thái công việc (To Do, In Progress, Done, Cancel) và thanh tiến độ hoàn thành của dự án.
  + Trường hợp không có dữ liệu: Hệ thống hiển thị màn hình trống kèm thông báo "Hiện tại chưa có dữ liệu công việc để thực hiện thống kê".

### 4.13.1 Các yêu cầu chức năng

* REQ-STAT-1: Chức năng thống kê phải được thiết kế hoàn toàn dưới dạng Plugin, tuân thủ nghiêm ngặt Interface của hệ thống lõi để đảm bảo tính mở rộng và tái sử dụng (Design for Reuse).
* REQ-STAT-2: Hệ thống phải xử lý việc truy vấn và tính toán dữ liệu thống kê trên một luồng (thread) riêng biệt bằng SwingWorker, đảm bảo giao diện người dùng (UI) vẫn phản hồi mượt mà dưới 0.5 giây ngay cả khi dữ liệu lớn.
* REQ-STAT-3: Dữ liệu thống kê phải được cập nhật tự động (gần thời gian thực) thông qua cơ chế Timer hoặc khi nhận được tín hiệu thay đổi trạng thái công việc từ hệ thống lõi.
* REQ-STAT-4: Plugin thống kê chỉ được phép truy xuất dữ liệu thông qua đối tượng HostContext được cung cấp, không được phép can thiệp trực tiếp vào cấu trúc cơ sở dữ liệu để đảm bảo tính đóng gói (Information Hiding).
* REQ-STAT-5: Hệ thống phải hỗ trợ hiển thị ít nhất hai loại biểu đồ phổ biến: Biểu đồ tròn (tỷ lệ trạng thái) và Biểu đồ cột (tiến độ theo thời gian).

# Các yêu cầu phi chức năng

## Yêu cầu thực thi

Do TeamWork Master chuyển sang mô hình ứng dụng web đơn trang (SPA) với kiến trúc Client-Server, các yêu cầu về hiệu năng sẽ tập trung vào tốc độ phản hồi của API, thời gian tải trang trên trình duyệt và trải nghiệm mượt mà khi tương tác với giao diện Vue.js.

* PER-1: Thời gian phản hồi API (API Latency)
* Yêu cầu: Các API RESTful phục vụ các thao tác cơ bản (CRUD) phải có thời gian phản hồi trung bình dưới 500ms trong điều kiện mạng ổn định. Đối với các tác vụ phức tạp (như Thống kê, Xuất báo cáo), thời gian phản hồi không được vượt quá 3 giây.
* Lý do (Rationale): Đảm bảo người dùng không phải chờ đợi lâu khi lưu công việc hoặc chuyển trạng thái, giữ cho dòng chảy công việc (workflow) không bị gián đoạn.
* PER-2: Tốc độ tải trang ban đầu (Initial Page Load)
* Yêu cầu: Thời gian để người dùng nhìn thấy nội dung chính (First Contentful Paint) khi truy cập vào Dashboard phải dưới 2 giây.
* Giải pháp kỹ thuật: Áp dụng kỹ thuật Lazy Loading (tải chậm) cho các Module và Component của Vue.js. Chỉ tải mã nguồn của các màn hình khi người dùng thực sự truy cập vào đó, thay vì tải toàn bộ ứng dụng ngay từ đầu.
* PER-3: Hiệu năng hiển thị danh sách lớn (Rendering Performance)
* Yêu cầu: Bảng Kanban hoặc Danh sách công việc phải duy trì tốc độ khung hình 60fps (cuộn mượt mà) ngay cả khi hiển thị lên tới 500 công việc trên một màn hình.
* Lý do (Rationale): Trình duyệt sẽ tốn nhiều tài nguyên để vẽ (render) DOM.
* Giải pháp: Cần sử dụng kỹ thuật Virtual Scrolling (chỉ render các phần tử đang hiển thị trong vùng nhìn thấy) để giảm tải cho trình duyệt.
* PER-4: Trải nghiệm bất đồng bộ (Non-blocking UI)
* Yêu cầu: Giao diện tuyệt đối không được bị "treo" (freeze) khi đang chờ dữ liệu từ Server.
* Chi tiết:
  + Khi người dùng thực hiện hành động (ví dụ: nhấn nút "Lưu"), hệ thống phải hiển thị ngay lập tức trạng thái phản hồi (như Spinner quay hoặc Skeleton Loading) để người dùng biết hệ thống đang xử lý.
  + Các tác vụ nặng phải được xử lý ngầm (Background) và thông báo kết quả sau thông qua Toast Notification.
* PER-5: Thời gian khởi động Server và Nạp Plugin (Backend Startup)
* Yêu cầu: Thời gian khởi động của Backend Java, bao gồm việc quét và nạp động (Dynamic Loading) 5 Plugin cơ bản thông qua Reflection, không được vượt quá 15 giây.
* Lý do (Rationale): Mặc dù Server chạy liên tục, nhưng thời gian khởi động nhanh giúp quá trình phát triển, bảo trì và triển khai lại (Redeploy) của nhóm lập trình viên diễn ra thuận lợi hơn.
* PER-6: Tối ưu băng thông mạng (Network Optimization)
* Yêu cầu: Dung lượng tải xuống của gói ứng dụng Frontend (JS/CSS bundle size) trong lần truy cập đầu tiên phải được nén (Gzip/Brotli) xuống dưới 2MB.
* Lý do (Rationale): Đảm bảo ứng dụng có thể truy cập tốt ngay cả khi người dùng sử dụng mạng 4G hoặc Wifi công cộng tốc độ thấp.

## Yêu cầu an toàn

Mục này xác định các cơ chế bảo vệ nhằm ngăn chặn khả năng mất mát dữ liệu, lỗi hệ thống do xung đột tài nguyên hoặc các thao tác sai lầm của người dùng trong môi trường ứng dụng Web.

* SAF-1: Cô lập lỗi Plugin (Backend Plugin Isolation)
* Mô tả: Hệ thống Backend (Java Core) phải đảm bảo rằng lỗi phát sinh từ một Plugin (ví dụ: lỗi logic tính toán, lỗi ném ngoại lệ Runtime) không được làm sập (crash) toàn bộ Web Server.
* Hoạt động an toàn:
  + Mọi lời gọi hàm tới Plugin thông qua IHostContext hoặc IPlugin phải được bọc trong khối try-catch tại lớp Service.
  + Nếu Plugin gặp lỗi trong quá trình xử lý một Request HTTP, hệ thống phải ghi lại lỗi vào Log, ngắt quy trình của riêng Request đó và trả về mã lỗi HTTP 500 (Internal Server Error) cho Client, nhưng vẫn phải đảm bảo Server tiếp tục phục vụ các Request khác bình thường.
* SAF-2: Tính toàn vẹn dữ liệu giao dịch (Transactional Data Integrity)
* Mô tả: Đảm bảo dữ liệu không bị sai lệch hoặc lưu trữ không hoàn chỉnh khi xảy ra lỗi hệ thống (như mất điện, mất kết nối DB) giữa chừng.
* Hoạt động an toàn:
  + Hệ thống Backend phải áp dụng cơ chế Giao dịch cơ sở dữ liệu (Database Transactions - ACID) cho các tác vụ phức tạp liên quan đến nhiều bảng (ví dụ: Khi xóa Dự án, phải xóa cả Công việc và File đính kèm).
  + Nguyên tắc "All or Nothing": Nếu bất kỳ bước nào trong quy trình thất bại, hệ thống phải tự động Rollback (khôi phục) toàn bộ dữ liệu về trạng thái ban đầu, ngăn chặn tình trạng dữ liệu "mồ côi" (Orphan data).
* SAF-3: An toàn đa luồng trong môi trường Web (Concurrency Safety)
* Mô tả: Do Backend phải phục vụ hàng trăm Request đồng thời từ nhiều người dùng, hệ thống phải đảm bảo không xảy ra xung đột dữ liệu (Race Condition).
* Hoạt động an toàn:
  + Các biến trạng thái chia sẻ (Shared State) trong các lớp Singleton Service của Java phải được quản lý chặt chẽ (hoặc tốt nhất là thiết kế Stateless).
  + Sử dụng các cấu trúc dữ liệu an toàn luồng (ConcurrentHashMap) hoặc từ khóa synchronized khi truy cập vào các tài nguyên chung như bộ nhớ đệm (Cache) hoặc danh sách Plugin đang hoạt động.
* SAF-4: Xác nhận thao tác phá hủy phía Client (Client-side Confirmation)
* Mô tả: Ngăn chặn người dùng vô tình xóa mất dữ liệu quan trọng do thao tác nhầm trên giao diện Web.
* Hoạt động an toàn:
  + Trên giao diện Vue.js, đối với mọi hành động mang tính phá hủy vĩnh viễn (như "Xóa Dự Án", "Xóa Thành Viên", "Hủy Công Việc"), hệ thống bắt buộc phải hiển thị một Hộp thoại cảnh báo (Modal Dialog) chặn màn hình.
  + Nút xác nhận (ví dụ: "Xóa ngay") nên được thiết kế màu đỏ và yêu cầu người dùng thao tác thêm một bước (như nhập lại tên dự án hoặc đợi 3 giây) mới có thể nhấn được.
* SAF-5: An toàn dữ liệu đầu vào (Input Validation Safety)
* Mô tả: Ngăn chặn việc hệ thống xử lý các dữ liệu rác hoặc không hợp lệ do lỗi nhập liệu của người dùng.
* Hoạt động an toàn: Áp dụng cơ chế kiểm tra 2 lớp (Double Validation):
  + Lớp 1 (Vue.js): Kiểm tra ngay trên trình duyệt (Form Validation) để báo lỗi tức thì cho người dùng (ví dụ: Ngày kết thúc không được nhỏ hơn ngày bắt đầu).
  + Lớp 2 (Backend Java): Kiểm tra lại toàn bộ dữ liệu khi nhận được API Request để đảm bảo an toàn ngay cả khi người dùng tắt JavaScript hoặc tấn công qua API.
* SAF-6: Xử lý sự cố mất kết nối mạng (Network Resilience)
* Mô tả: Đảm bảo trải nghiệm người dùng không bị gián đoạn đột ngột khi kết nối mạng của họ không ổn định.
* Hoạt động an toàn:
  + Khi Vue.js gửi Request và gặp lỗi mất mạng (Network Error), giao diện không được hiển thị màn hình trắng hoặc lỗi kỹ thuật khó hiểu.
  + Thay vào đó, phải hiển thị thông báo "Mất kết nối Internet. Vui lòng kiểm tra lại" và giữ nguyên trạng thái dữ liệu người dùng đang nhập liệu để họ không phải nhập lại từ đầu khi có mạng trở lại.

## Yêu cầu bảo mật

Mục này quy định các biện pháp kỹ thuật nhằm bảo vệ dữ liệu người dùng, đảm bảo tính riêng tư và ngăn chặn các truy cập trái phép vào hệ thống thông qua Internet hoặc API.

* SEC-1: Xác thực và Quản lý Phiên (Authentication & Session Management)
* Yêu cầu: Hệ thống phải sử dụng cơ chế xác thực không trạng thái (Stateless) thông qua JSON Web Token (JWT) thay vì session truyền thống để phù hợp với kiến trúc RESTful API.
* Chi tiết:
  + Mật khẩu người dùng lưu trong Database phải được mã hóa (Hashing) bằng thuật toán mạnh (như BCrypt hoặc Argon2). Tuyệt đối không lưu văn bản thuần.
  + Khi đăng nhập thành công, Backend trả về một chuỗi AccessToken (có thời hạn ngắn, ví dụ: 30 phút) và RefreshToken. Frontend (Vue.js) lưu trữ token này (khuyến nghị lưu trong HttpOnly Cookie hoặc LocalStorage) để đính kèm vào Header của các Request tiếp theo (Authorization: Bearer <token>).
* SEC-2: Kiểm soát truy cập API (API Authorization / RBAC)
* Yêu cầu: Việc phân quyền phải được thực hiện triệt để tại Backend Java, không được chỉ dựa vào việc ẩn nút trên giao diện Vue.js.
* Chi tiết:
  + Mọi API Endpoint nhạy cảm (như DELETE /api/projects/{id}, POST /api/users/ban) phải có Middleware hoặc Interceptor chặn lại để kiểm tra vai trò (Role) của người gọi.
  + Ví dụ: Nếu một Member cố tình dùng công cụ (như Postman) để gửi lệnh xóa dự án, Backend phải trả về lỗi 403 Forbidden.
* SEC-3: Bảo mật kênh truyền thông (Secure Communication - HTTPS)
* Yêu cầu: Toàn bộ dữ liệu trao đổi giữa Client (Trình duyệt) và Server bắt buộc phải được mã hóa thông qua giao thức HTTPS (TLS/SSL).
* Lý do: Ngăn chặn các cuộc tấn công "nghe lén" (Sniffing) hoặc "trung gian" (Man-in-the-Middle) để đánh cắp Token đăng nhập hoặc dữ liệu công việc nhạy cảm khi người dùng truy cập từ mạng Wifi công cộng.
* SEC-4: Cơ chế "Hộp cát" cho Plugin (Plugin Sandboxing)
* Yêu cầu: Vì Plugin (file .jar) được tải lên và chạy trực tiếp trên Server, hệ thống Core phải cô lập quyền hạn của chúng.
* Chi tiết:
  + Security Manager: Cấu hình Java Security Manager để ngăn chặn Plugin thực hiện các hành vi nguy hiểm như: Đọc/Ghi các file hệ thống (ngoài thư mục cho phép), mở kết nối mạng lạ (Socket), hoặc gọi lệnh Shell của hệ điều hành (Runtime.exec).
  + Plugin chỉ được phép tương tác với dữ liệu thông qua các hàm API công khai mà IHostContext cung cấp.
* SEC-5: Phòng chống tấn công Web (Web Vulnerability Prevention)
* Yêu cầu: Hệ thống phải được cấu hình để chống lại các lỗ hổng web phổ biến (OWASP Top 10).
* Chi tiết:
  + CORS (Cross-Origin Resource Sharing): Backend phải cấu hình chặt chẽ chính sách CORS, chỉ cho phép các Domain/Port cụ thể của Frontend (ví dụ: localhost:3000 hoặc domain thật) được phép gọi API.
  + Chống XSS (Cross-Site Scripting): Frontend (Vue.js) phải tự động Escape các ký tự đặc biệt khi hiển thị nội dung do người dùng nhập (như mô tả công việc, bình luận) để ngăn chặn mã độc JavaScript thực thi trên trình duyệt người khác.
  + SQL Injection: Backend bắt buộc sử dụng PreparedStatement hoặc các thư viện ORM (như Hibernate) khi truy vấn Database, tuyệt đối không nối chuỗi SQL thủ công.

## Các đặc điểm chất lượng phần mềm

QA-1: Tính mở rộng và Linh hoạt (Extensibility & Flexibility)

* Mô tả: Hệ thống phải giữ vững kiến trúc Microkernel (đặc trưng của đồ án), cho phép thêm tính năng mới mà không cần biên dịch lại mã nguồn lõi (Core) hay Frontend.
* Tiêu chí kiểm tra:
  + Backend: Hệ thống phải hỗ trợ nạp nóng (Hot-loading) hoặc nạp động các file .jar vào thư mục /plugins trên Server.
  + Frontend: Giao diện Vue.js phải được thiết kế dạng "Dynamic Rendering". Khi Backend báo về có Plugin mới (ví dụ: Plugin Thống kê), giao diện phải tự động hiển thị thêm mục Menu hoặc Widget tương ứng dựa trên cấu hình JSON trả về, không cần sửa code Vue.js cứng (Hard-code).
  + Tuân thủ nguyên lý Open/Closed Principle (Mở cho mở rộng, đóng cho sửa đổi).

QA-2: Tính khả chuyển và Tương thích (Portability & Compatibility)

* Mô tả: Ứng dụng phải hoạt động thống nhất trên nhiều môi trường khác nhau.
* Tiêu chí kiểm tra:
  + Cross-Browser: Giao diện Vue.js phải hiển thị và hoạt động đúng (không vỡ layout, không lỗi JS) trên các trình duyệt phổ biến phiên bản mới nhất: Google Chrome, Mozilla Firefox, Microsoft Edge, và Safari.
  + Cross-Platform (Backend): Server Java (đóng gói dạng .jar) phải chạy được trên Windows, macOS, và Linux (Ubuntu/CentOS) mà không cần sửa đổi đường dẫn file (sử dụng File.separator hoặc đường dẫn tương đối).
  + Responsive: Giao diện phải tự động thích ứng với các kích thước màn hình: Desktop (>1200px), Laptop, và Tablet.

QA-3: Tính dễ sử dụng và Trải nghiệm người dùng (Usability & UX)

* Mô tả: Giao diện Web phải trực quan, giảm thiểu thao tác thừa và cung cấp phản hồi tức thì.
* Tiêu chí kiểm tra:
  + Quy tắc "3 Click": Người dùng phải truy cập được bất kỳ tính năng chính nào (Tạo task, Xem chi tiết, Báo cáo) trong tối đa 3 lần nhấp chuột.
  + Direct Manipulation: Hỗ trợ thao tác Kéo & Thả (Drag & Drop) trực quan cho việc chuyển trạng thái công việc trên bảng Kanban.
  + Feedback: Mọi hành động (Lưu, Xóa, Tải) đều phải có thông báo phản hồi (Toast Message) hoặc hiệu ứng Loading để người dùng biết hệ thống đang xử lý.
  + Error Messages: Thông báo lỗi phải dùng ngôn ngữ tự nhiên (ví dụ: "Email không đúng định dạng") thay vì mã lỗi kỹ thuật.

QA-4: Tính có thể bảo trì (Maintainability)

* Mô tả: Mã nguồn phải được tổ chức khoa học để dễ dàng sửa lỗi và nâng cấp bởi các lập trình viên khác nhau (Frontend và Backend).
* Tiêu chí kiểm tra:
  + Frontend (Vue.js): Phải chia nhỏ thành các Components tái sử dụng (Button, Modal, TaskCard). Mã nguồn tuân thủ chuẩn ESLint và Prettier.
  + Backend (Java): Tuân thủ mô hình phân lớp (Controller - Service - Repository). Các class và method phải có JavaDoc.
  + API: Tài liệu API (Swagger/OpenAPI) phải được cập nhật đồng bộ với code để team Frontend có thể tích hợp chính xác.

QA-5: Tính tin cậy (Reliability)

* Mô tả: Hệ thống phải hoạt động ổn định và có khả năng tự phục hồi khi gặp lỗi cục bộ (Resilience).
* Tiêu chí kiểm tra:
  + Frontend Resilience: Nếu một API bị lỗi (500 Error), giao diện không được "trắng trang" (White Screen of Death). Phải hiển thị thông báo lỗi nhẹ nhàng và cho phép người dùng thử lại (Retry).
  + Backend Stability: Nếu một Plugin bị lỗi (Crash/Exception), Core Framework phải bắt được ngoại lệ (Catch Exception), cô lập lỗi đó và vẫn duy trì hoạt động của các chức năng cốt lõi (như Đăng nhập, Xem dự án).

QA-6: Tính có thể tái sử dụng (Reusability)

* Mô tả: Các thành phần nền tảng phải được thiết kế để có thể dùng cho dự án khác.
* Tiêu chí kiểm tra:
  + Core Framework: Bộ khung Backend (PluginLoader, Authentication, BaseController) phải tách biệt hoàn toàn với nghiệp vụ "Quản lý công việc". Có thể mang bộ khung này đi làm phần mềm "Quản lý Sinh viên" mà không cần sửa code Core.
  + UI Library: Các Component Vue.js cơ bản (Input, Dropdown, Table) nên được tách thành thư mục riêng (ví dụ: /components/base) để dễ dàng tái sử dụng cho các màn hình khác nhau.

## Các quy tắc nghiệp vụ

BR-1: Quy tắc phân cấp quyền hạn (Role Hierarchy Policy)

* Mô tả: Hệ thống vận hành dựa trên 3 cấp độ quyền hạn chồng nhau: Admin > Manager > Member.
* Chi tiết:
  + Admin (Quản trị viên): Có toàn quyền truy cập vào tất cả dữ liệu, cấu hình hệ thống và quản lý tất cả người dùng. Admin có thể xem và can thiệp vào bất kỳ dự án nào.
  + Manager (Quản lý dự án): Chỉ có quyền quản lý (Sửa/Xóa Dự án, Tạo/Gán công việc) đối với các dự án mà họ là người tạo ra (Owner) hoặc được chỉ định làm đồng quản lý. Đối với các dự án khác mà họ tham gia với tư cách thành viên, họ chỉ có quyền hạn như một Member.
  + Member (Thành viên): Chỉ có quyền xem thông tin dự án mình tham gia và chỉnh sửa trạng thái (kéo thả) các công việc được gán cho chính mình. Member tuyệt đối không được phép xóa Dự án, xóa Công việc hoặc mời thành viên mới.

BR-2: Quy tắc về Thời gian và Hạn chót (Time & Deadline Validity)

* Mô tả: Đảm bảo tính logic của dữ liệu thời gian để tránh các sai sót trong lập kế hoạch.
* Chi tiết:
  + Logic ngày tháng: "Ngày bắt đầu" (Start Date) của một công việc phải luôn nhỏ hơn hoặc bằng "Ngày kết thúc/Hạn chót" (Due Date). Giao diện Vue.js phải chặn không cho người dùng chọn ngày sai.
  + Ràng buộc cha-con: Hạn chót của một công việc con (Task) không được phép vượt quá Hạn chót của Dự án cha (Project). Nếu người dùng cố tình đặt deadline của công việc trễ hơn deadline dự án, Backend phải từ chối và trả về lỗi nghiệp vụ.

BR-3: Quy tắc Luồng trạng thái công việc (Workflow Logic)

* Mô tả: Kiểm soát sự chuyển dịch trạng thái của công việc trên bảng Kanban để đảm bảo quy trình làm việc chuẩn.
* Chi tiết:
  + Trạng thái khởi tạo: Một công việc khi mới được tạo phải luôn ở trạng thái mặc định là "To Do" (Cần làm).
  + Quyền chuyển trạng thái: Chỉ có Người được giao việc (Assignee) hoặc Quản lý dự án (Manager) mới được phép kéo thả công việc sang trạng thái "In Progress" hoặc "Done".
  + Khóa trạng thái "Done": Công việc đã ở trạng thái "Done" (Hoàn thành) sẽ bị khóa, không được phép chỉnh sửa nội dung (mô tả, đính kèm). Muốn chỉnh sửa, người có thẩm quyền (Manager) phải chuyển trạng thái ngược lại về "In Progress" (Re-open).

BR-4: Quy tắc Định danh duy nhất (Unique Identity)

* Mô tả: Đảm bảo không có sự trùng lặp dữ liệu định danh trong hệ thống cơ sở dữ liệu.
* Chi tiết:
  + Tài khoản: Tên đăng nhập (Username) và Email phải là duy nhất trên toàn hệ thống.
  + Mã dự án: ID của dự án phải là duy nhất.
  + Xử lý xung đột: Nếu một tài khoản đã bị xóa (soft delete), các thông tin này vẫn không được phép tái sử dụng ngay lập tức để đảm bảo tính toàn vẹn của lịch sử Log.

BR-5: Quy tắc Bảo toàn dữ liệu dự án (Data Preservation)

* Mô tả: Ngăn chặn việc mất mát dữ liệu quan trọng do thao tác xóa nhầm hoặc xóa không đúng quy trình.
* Chi tiết:
  + Điều kiện xóa: Không được phép xóa một Dự án đang ở trạng thái "Active" (Đang chạy). Người dùng phải chuyển trạng thái dự án sang "Closed" hoặc "Canceled" trước khi thực hiện lệnh xóa.
  + Xóa dây chuyền (Cascading Delete): Khi một Dự án bị xóa hợp lệ, toàn bộ các thực thể con phụ thuộc (Công việc, Bình luận, File đính kèm) phải được xóa theo tự động để tránh rác dữ liệu (Orphan data).

BR-6: Quy tắc về Plugin (Plugin Integrity Policy)

* Mô tả: Quy định chặt chẽ về việc chấp nhận các module mở rộng (.jar) vào hệ thống Backend.
* Chi tiết:
  + Hợp đồng Interface: Hệ thống Core chỉ chấp nhận nạp các file .jar có chứa lớp hiện thực đúng giao diện IPlugin mà Core quy định. Các file không đạt chuẩn sẽ bị từ chối nạp và báo lỗi.
  + Xung đột Plugin: Nếu nạp hai Plugin có cùng ID định danh, hệ thống sẽ ưu tiên Plugin có số phiên bản (Version) cao hơn.

# Các yêu cầu khác

### 6.1 Yêu cầu về Cơ sở dữ liệu và Lưu trữ (Database Requirements)

Do đặc thù là ứng dụng Desktop hoạt động độc lập (Standalone) và cần đảm bảo tính "Khả chuyển" (Portability), hệ thống không sử dụng các hệ quản trị cơ sở dữ liệu cài đặt phức tạp (như SQL Server hay Oracle).

* OTH-1: Hệ thống phải sử dụng Cơ sở dữ liệu nhúng (Embedded Database) như H2, SQLite hoặc cơ chế lưu trữ dựa trên tệp tin cấu trúc (Structured Files: XML/JSON) để lưu trữ dữ liệu dự án.
* OTH-2: Các tệp dữ liệu phải được lưu trữ trong thư mục người dùng (User Home Directory) để tránh các vấn đề về quyền truy cập (Permission Denied) trên hệ điều hành Windows/Linux.
* OTH-3: Dữ liệu cấu hình (như theme, ngôn ngữ, vị trí cửa sổ) phải được lưu trong tệp config.properties chuẩn của Java để dễ dàng chỉnh sửa thủ công khi cần.

### 6.2 Yêu cầu về Quốc tế hóa và Văn hóa (Internationalization & Cultural Requirements)

* OTH-4: Ứng dụng phải hỗ trợ bảng mã Unicode (UTF-8) một cách toàn diện để cho phép người dùng nhập tên công việc, mô tả và bình luận bằng Tiếng Việt có dấu mà không bị lỗi font.
* OTH-5: Giao diện mặc định của phần mềm sử dụng Tiếng Anh (hoặc Tiếng Việt - tùy nhóm chọn), nhưng định dạng ngày tháng (Date/Time Format) phải tuân theo văn hóa người dùng máy tính tại Việt Nam: dd/MM/yyyy HH:mm (ví dụ: 31/01/2026 14:30) thay vì định dạng Mỹ (MM/dd/yyyy).

### 6.3 Yêu cầu về Pháp lý và Bản quyền (Legal & Copyright Requirements)

* OTH-6 (Tuân thủ giấy phép mã nguồn mở): Vì dự án sử dụng các thư viện bên thứ ba (Third-party libraries) như *JUnit* (Eclipse Public License) hay *FlatLaf* (Apache License 2.0), phần mềm phải bao gồm một tệp văn bản LICENSE hoặc NOTICE liệt kê đầy đủ các giấy phép này để tuân thủ quy định về sở hữu trí tuệ.
* OTH-7 (Liêm chính học thuật): Mã nguồn của phần Core Framework và các thuật toán xử lý chính phải do nhóm tự phát triển. Các đoạn mã tham khảo từ Internet (StackOverflow, GitHub) phải được chú thích rõ nguồn gốc trong comment của code.

### 6.4 Mục tiêu Tái sử dụng (Reuse Objectives)

Đây là yêu cầu đặc biệt quan trọng của đồ án môn học này ("Design for Reuse").

* OTH-8: Thành phần Core Framework (bao gồm bộ nạp Plugin PluginLoader, giao diện IGUI, và cơ chế EventBus) phải được thiết kế tách biệt hoàn toàn với nghiệp vụ "Quản lý công việc". Mục tiêu là bộ Core này có thể được mang sang sử dụng để xây dựng một phần mềm khác (ví dụ: Phần mềm Quản lý Sinh viên) mà không cần sửa đổi mã nguồn (Reused without modification).

### 6.5 Yêu cầu về Đóng gói và Phân phối (Packaging & Deployment)

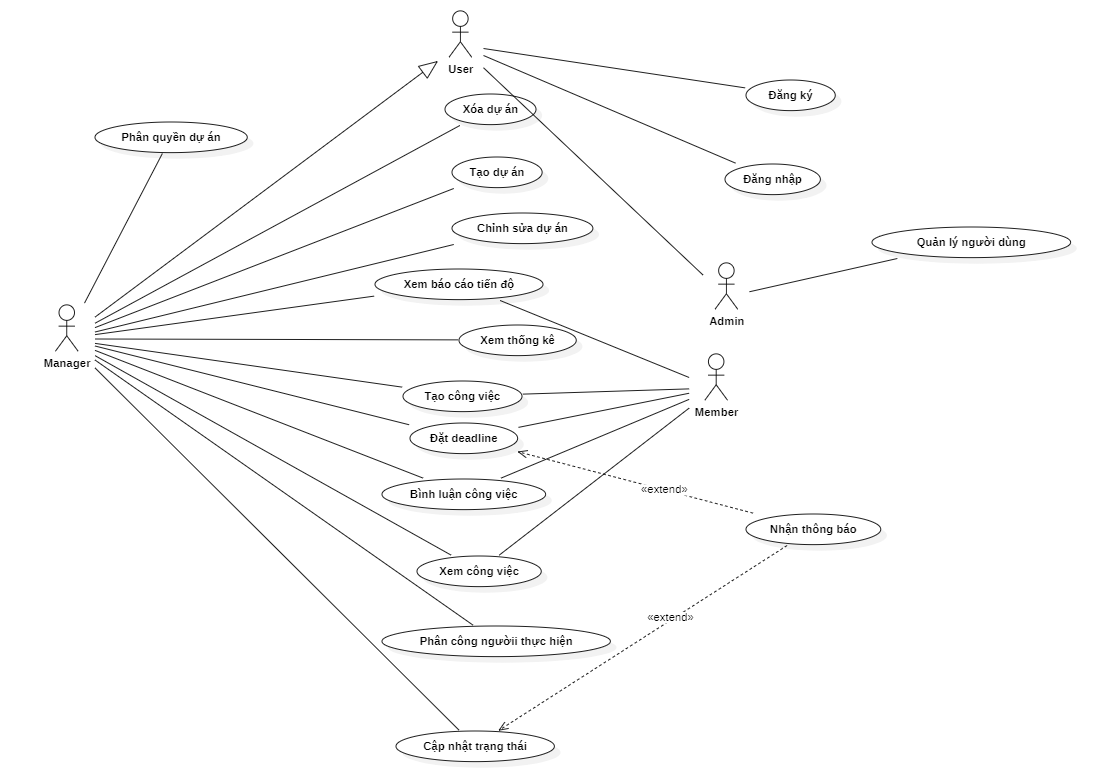
* OTH-9: Sản phẩm cuối cùng phải được đóng gói thành tệp Executable JAR (.jar) duy nhất. Người dùng chỉ cần nhấp đúp chuột (double-click) vào file này để chạy chương trình (với điều kiện máy đã cài JRE).
* OTH-10: Cấu trúc thư mục khi bàn giao sản phẩm phải rõ ràng:
  + TeamWorkMaster.jar (File chạy chính)
  + /plugins (Thư mục chứa các file .jar mở rộng)
  + /libs (Thư mục chứa thư viện phụ thuộc nếu không đóng gói chung)
  + README.txt (Hướng dẫn cài đặt nhanh)

### 6.6 Yêu cầu về Tài liệu hướng dẫn (Documentation)

* OTH-11: Ngoài tài liệu đặc tả (SRS) và thiết kế (SDD), nhóm phát triển phải cung cấp tài liệu Hướng dẫn phát triển Plugin (Plugin Developer Guide). Tài liệu này hướng dẫn lập trình viên khác cách tạo ra một file .jar tuân thủ Interface của hệ thống để mở rộng tính năng cho phần mềm.

**Phụ lục A: Các mô hình phân tích**

**Sơ đồ usecase**

****

**Sơ đồ class**

****

**Phụ lục B: TBD – Danh sách sẽ được xác định**

<Thu thập một danh sách được đánh số của các tham khảo TBD (To Be Determine) mà chúng vẫn còn trong tài liệu đặc tả.>