**PHENIKAA UNIVERSITY**

**FACULTY OF COMPUTER SCIENCE  
   
 A logo for a university

Description automatically generated**

**Báo Cáo Đồ Án Liên Ngành**

**Nhóm 4**

**Xây dựng hệ thống quản lý không gian làm việc**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Email** |
| **Vũ Khải Hoàn** | **21012316** | **21012316@st.phenikaa-uni.edu.vn** |
| **Định Trọng Việt Phú** | **21012321** | **21012321@st.phenikaa-uni.edu.vn** |

**Giảng viên hướng dẫn: TS. Nguyễn Thị Thùy Liên**

**Hà Nội, tháng 7 năm 2025**

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc204257434)

[LỜI CẢM ƠN 4](#_Toc204257435)

[LỜI CAM ĐOAN 5](#_Toc204257436)

[Đặt vấn đề và Giải pháp 6](#_Toc204257437)

[1. Đặt vấn đề 6](#_Toc204257438)

[2. Các giải pháp đã có 6](#_Toc204257439)

[3. Giải pháp đề xuất 6](#_Toc204257440)

[Thiết kế và Triển khai 7](#_Toc204257441)

[1. Các yêu cầu chức năng 7](#_Toc204257442)

[2. Các yêu cầu phi chức năng 9](#_Toc204257443)

[3. Các ràng buộc (Constraints) 10](#_Toc204257444)

[3.1. Ràng buộc về triển khai 10](#_Toc204257445)

[3.2. Ràng buộc kinh tế 10](#_Toc204257446)

[3.3. Ràng buộc về đạo đức 10](#_Toc204257447)

[4. Mô hình hệ thống / Thiết kế giải pháp 10](#_Toc204257448)

[4.1. Các kịch bản hệ thống (Use Cases) 10](#_Toc204257449)

[4.2. Mô hình Use-case 17](#_Toc204257450)

[4.3. Mô hình lớp và đối tượng 27](#_Toc204257451)

[4.4. Các biểu đồ tuần tự 42](#_Toc204257452)

[4.5. Các màn hình giao diện người dùng 55](#_Toc204257453)

[Một số thành phần khác 69](#_Toc204257454)

[1. Kế hoạch dự án 69](#_Toc204257455)

[2. Đảm bảo thực hiện đúng làm việc nhóm 72](#_Toc204257456)

[3. Các vấn đề đạo đức và làm việc chuyên nghiệp 72](#_Toc204257457)

[4. Tác động xã hội 73](#_Toc204257458)

[5. Kế hoạch học tập và kiến thức mới 73](#_Toc204257459)

[Tài liệu tham khảo 74](#_Toc204257460)

**TABLE OF CONTENTS**

[INTRODUCTION 3](#_Toc204257434)

[ACKNOWLEDGEMENT 4](#_Toc204257435)

[COMMITMENT 5](#_Toc204257436)

[Problem statement and Solutions 6](#_Toc204257437)

[1. Problem statement 6](#_Toc204257438)

[2. Existing solutions 6](#_Toc204257439)

[3. Proposed solutions 6](#_Toc204257440)

[Design and Implementation 7](#_Toc204257441)

[1. Functional requirements 7](#_Toc204257442)

[2. Authentication 9](#_Toc204257443)

[3. Constraints 10](#_Toc204257444)

[3.1. Implementation constraints 10](#_Toc204257445)

[3.2. Ethical constraints 10](#_Toc204257446)

[3.3. Legal constraints 10](#_Toc204257447)

[4. System modeling / Solution design 10](#_Toc204257448)

[4.1. Use Cases 10](#_Toc204257449)

[4.2. Object-oriented modeling 17](#_Toc204257450)

[4.3. User interface mockups 27](#_Toc204257451)

[4.4. Use-case descriptions 42](#_Toc204257452)

[4.5. User interface components 55](#_Toc204257453)

[Additional components 69](#_Toc204257454)

[1. Project plan 69](#_Toc204257455)

[2. Ensuring proper teamwork 72](#_Toc204257456)

[3. Ethics and professionalism 72](#_Toc204257457)

[4. Social impact 73](#_Toc204257458)

[5. Learning plan and new knowledge 73](#_Toc204257459)

[References 74](#_Toc204257460)

**DANH SÁCH HÌNH ẢNH**

[Hình 1. Sơ đồ Use-Case tổng quan hệ thống 17](#_Toc204257373)

[Hình 2. Sơ đồ Use-Case của admin 18](#_Toc204257374)

[Hình 3. Sơ đồ Use-Case của user 19](#_Toc204257375)

[Hình 4. Sơ đồ Use-Case xác thực và ủy quyền 20](#_Toc204257376)

[Hình 5. Sơ đồ Use-Case quản lý không gian làm việc 20](#_Toc204257377)

[Hình 6. Sơ đồ Use-Case quản lý thành viên 21](#_Toc204257378)

[Hình 7. Sơ đồ Use-Case quản lý công việc 22](#_Toc204257379)

[Hình 8. Sơ đồ Use-Case quản lý ghi chú 23](#_Toc204257380)

[Hình 9. Sơ đồ Use-Case quản lý sự kiện 24](#_Toc204257381)

[Hình 10. Sơ đồ Use-Case quản lý mục tiêu 25](#_Toc204257382)

[Hình 11. Sơ đồ Use-Case quản lý giao dịch 26](#_Toc204257383)

[Hình 12. Sơ đồ Use-Case quản lý thông tin cá nhân 26](#_Toc204257384)

[Hình 13. Sơ đồ lớp tổng quan hệ thống 28](#_Toc204257385)

[Hình 14. Sơ đồ lớp User 29](#_Toc204257386)

[Hình 15. Sơ đồ lớp Workspace 30](#_Toc204257387)

[Hình 16. Sơ đồ lớp WorkspaceMember 31](#_Toc204257388)

[Hình 17. Sơ đồ lớp Task 32](#_Toc204257389)

[Hình 18. Sơ đồ lớp TaskAssignment 32](#_Toc204257390)

[Hình 19. Sơ đồ lớp Note 33](#_Toc204257391)

[Hình 20. Sơ đồ lớp Event 34](#_Toc204257392)

[Hình 21. Sơ đồ lớp Goal 35](#_Toc204257393)

[Hình 22. Sơ đồ lớp Transaction 36](#_Toc204257394)

[Hình 23. Sơ đồ cơ sở dữ liệu 42](#_Toc204257395)

[Hình 24. sơ đồ tuần tự chức năng đăng nhập 43](#_Toc204257396)

[Hình 25. sơ đồ tuần tự chức năng quên mật khẩu 45](#_Toc204257397)

[Hình 26. sơ đồ tuần tự chức năng đăng ký 46](#_Toc204257398)

[Hình 27. sơ đồ tuần tự chức năng quản lý không gian làm việc 48](#_Toc204257399)

[Hình 28. sơ đồ tuần tự chức năng quản lý thành viên và công việc 53](#_Toc204257400)

[Hình 29. sơ đồ tuần tự chức năng quản lý người dùng 55](#_Toc204257401)

[Hình 30. Màn hình đăng ký tài khoản 56](#_Toc204257402)

[Hình 31. Màn hình đăng nhập 56](#_Toc204257403)

[Hình 32. Màn hình gửi mail quên mật khẩu 56](#_Toc204257404)

[Hình 33. Màn hình đặt lại mật khẩu 57](#_Toc204257405)

[Hình 34. Màn hình trang chủ 57](#_Toc204257406)

[Hình 35. Màn hình sửa thông tin workspace cơ bản 58](#_Toc204257407)

[Hình 36. Màn hình chi tiết workspace 58](#_Toc204257408)

[Hình 37. Màn hình sửa thông chi tiết workspace 59](#_Toc204257409)

[Hình 38. Màn hình quản lý thành viên 59](#_Toc204257410)

[Hình 39. Màn hình quản lý công việc 60](#_Toc204257411)

[Hình 40. Màn hình chỉnh sửa công việc 60](#_Toc204257412)

[Hình 41. Màn hình gán công việc cho người dùng 60](#_Toc204257413)

[Hình 42. Màn hình quản lý ghi chú 61](#_Toc204257414)

[Hình 43. Màn hình quản lý sự kiện 61](#_Toc204257415)

[Hình 44. Màn hình quản lý mục tiêu 62](#_Toc204257416)

[Hình 45. Màn hình quản lý giao dịch 62](#_Toc204257417)

[Hình 46. Màn hình hồ sơ cá nhân 63](#_Toc204257418)

[Hình 47. Màn hình chỉnh sửa hồ sơ cá nhân 63](#_Toc204257419)

[Hình 48. Màn hình đổi mật khẩu 63](#_Toc204257420)

[Hình 49. Màn hình quản trị người dùng 64](#_Toc204257421)

[Hình 50. Màn hình đổi mật khẩu người dùng 64](#_Toc204257422)

[Hình 51. Màn hình sửa thông tin người dùng 64](#_Toc204257423)

[Hình 52. Kế hoạch dự án 71](#_Toc204257424)

[Hình 53. Biểu đồ Gantt phân công công việc theo thành viên 72](#_Toc204257425)

# LỜI MỞ ĐẦU

Hiện nay, khi công nghệ đang ngày càng gắn liền với đời sống hằng ngày, việc sử dụng phần mềm để tổ chức, quản lý và cá nhân hóa không gian sử dụng đã trở thành một nhu cầu thiết yếu. Từ không gian học tập, làm việc cho đến sinh hoạt, người dùng đều mong muốn có công cụ giúp họ tùy chỉnh linh hoạt, dễ sử dụng và tiết kiệm thời gian.

Từ thực tế đó, nhóm chúng tôi đã xây dựng một hệ thống web nhằm hỗ trợ người dùng thiết kế và quản lý không gian cá nhân một cách trực quan, tiện lợi. Bên cạnh tính năng quản lý, hệ thống còn hướng đến trải nghiệm người dùng, đảm bảo tính tương tác và dễ mở rộng. Trong suốt quá trình làm việc, nhóm không chỉ áp dụng kiến thức đã học về lập trình, cơ sở dữ liệu, mà còn học cách phối hợp nhóm và giải quyết bài toán thực tiễn.

Báo cáo này tổng hợp toàn bộ quá trình nhóm thực hiện đề tài, từ phân tích nhu cầu, xây dựng ý tưởng, đến thiết kế và triển khai sản phẩm. Nhóm xin gửi lời cảm ơn chân thành đến thầy/cô đã hướng dẫn, cùng tất cả thành viên đã đồng hành hoàn thành dự án này.

# LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến **TS. Nguyễn Thị Thùy Liên**, người đã tận tình hướng dẫn và hỗ trợ nhóm trong suốt quá trình thực hiện đồ án với đề tài **"Xây dựng hệ thống quản lý không gian làm việc"**.

Nhờ sự hướng dẫn cặn kẽ, những nhận xét sâu sắc và định hướng cụ thể của cô, nhóm chúng em đã có thể vượt qua nhiều khó khăn trong quá trình triển khai ý tưởng và hoàn thiện sản phẩm. Không chỉ giúp nhóm củng cố kiến thức chuyên môn về phát triển phần mềm, cô còn truyền đạt cho chúng em nhiều kinh nghiệm quý báu trong việc xử lý tình huống, tổ chức công việc và tiếp cận vấn đề theo tư duy thực tiễn.

Dù đã cố gắng hoàn thành đồ án với tinh thần trách nhiệm cao nhất, nhưng chắc chắn vẫn còn tồn tại những thiếu sót nhất định. Nhóm rất mong nhận được góp ý từ quý thầy cô để có thể tiếp tục cải thiện, hoàn thiện và phát triển hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn cô Nguyễn Thị Thùy Liên và tất cả những ai đã hỗ trợ, đồng hành cùng nhóm trong suốt quá trình thực hiện đề tài này.

# LỜI CAM ĐOAN

Chúng em cam đoan đây là công trình nghiên cứu của cả nhóm và được sự hướng dẫn của TS. Nguyễn Thị Thùy Liên. Các nội dung nghiên cứu trong đề tài của chúng em là trung thực và chưa được công bố dưới bất kỳ hình thức nào trước đây.

Chúng em xin chịu trách nhiệm hoàn toàn về nội dung của Đồ án môn học mà chúng em đã nộp.

# Đặt vấn đề và Giải pháp

## Đặt vấn đề

Trong thời đại số hóa, nhu cầu quản lý công việc, thời gian, tài chính, và tương tác nhóm hiệu quả ngày càng trở nên cấp thiết đối với cả cá nhân lẫn các nhóm nhỏ như gia đình, nhóm học tập, hoặc đội ngũ khởi nghiệp. Sự phát triển của công nghệ đã mang lại nhiều công cụ hỗ trợ, nhưng người dùng thường phải sử dụng nhiều ứng dụng khác nhau để đáp ứng các nhu cầu riêng lẻ, dẫn đến sự phân mảnh thông tin. Ví dụ, việc theo dõi tiến độ công việc trên Trello, ghi chú trên Notion, quản lý lịch trên Google Calendar, và theo dõi chi tiêu trên Money Lover khiến dữ liệu trở nên rời rạc, khó đồng bộ, và gây khó khăn trong việc phối hợp nhóm, đặc biệt với các nhóm nhỏ có yêu cầu linh hoạt cao.

Thực tế, các nhóm nhỏ thường gặp thách thức trong việc tìm kiếm một giải pháp tích hợp, nơi họ có thể quản lý công việc cá nhân, chia sẻ lịch trình chung, và theo dõi tài chính nhóm một cách liền mạch. Điều này đòi hỏi một hệ thống không chỉ cung cấp không gian làm việc cá nhân mà còn hỗ trợ phân quyền, giao việc, và đồng bộ dữ liệu giữa các thành viên, với giao diện thống nhất để giảm thiểu sự phức tạp.

## Các giải pháp đã có

Hiện nay, có nhiều ứng dụng phổ biến hỗ trợ từng khía cạnh riêng lẻ:

* Trello và Asana: Tập trung vào quản lý công việc và dự án, sử dụng bảng kanban hoặc danh sách nhiệm vụ.
* Google Keep và Notion: Cung cấp công cụ ghi chú và tổ chức nội dung, hỗ trợ cộng tác cơ bản.
* Money Lover và Misa: Tập trung vào quản lý tài chính cá nhân, với các tính năng theo dõi chi tiêu và lập ngân sách.
* Google Calendar: Dành cho quản lý lịch trình và sự kiện, hỗ trợ đồng bộ giữa các thiết bị.

Mặc dù các ứng dụng quản lý công việc và cuộc sống hiện nay có những điểm mạnh riêng, chúng vẫn tồn tại một số hạn chế đáng kể:

* Thiếu tích hợp và đồng bộ hóa: Thông tin về công việc, sự kiện và tài chính thường nằm rải rác trên các ứng dụng khác nhau. Điều này buộc người dùng phải nhập liệu thủ công lặp lại, dễ dẫn đến sai sót và tốn thời gian, thay vì có một cái nhìn tổng thể liền mạch về các hoạt động của mình.
* Không linh hoạt cho nhóm nhỏ: Nhiều hệ thống hiện tại được thiết kế cho các tổ chức lớn với cấu trúc phức tạp, gây khó khăn cho các nhóm nhỏ (như gia đình, nhóm bạn bè, hoặc nhóm học tập) trong việc sử dụng. Những nhóm này cần một giao diện đơn giản hơn, dễ tùy chỉnh và phù hợp với nhu cầu cộng tác không quá phức tạp.
* Trải nghiệm người dùng không nhất quán: Việc phải chuyển đổi liên tục giữa nhiều ứng dụng khác nhau với các giao diện khác biệt làm gián đoạn luồng làm việc và gây khó khăn cho người dùng trong việc làm quen. Điều này làm giảm hiệu quả và sự thuận tiện trong quá trình quản lý công việc hàng ngày.

## Giải pháp đề xuất

* Để khắc phục những hạn chế hiện có trong việc quản lý công việc và cuộc sống, chúng tôi đề xuất một hệ thống tích hợp toàn diện, giúp người dùng tổ chức công việc, ghi chú, lịch trình, mục tiêu và tài chính một cách hiệu quả. Hệ thống này được thiết kế để phục vụ cả nhu cầu cá nhân lẫn việc cộng tác trong các nhóm nhỏ.
* Giải pháp của chúng tôi bao gồm các thành phần chính sau:
* Không gian làm việc cá nhân: Mỗi người dùng sẽ có một không gian riêng tư để quản lý các công việc và thông tin cá nhân của mình. Tại đây, bạn có thể dễ dàng xem và chỉnh sửa hồ sơ, cũng như thay đổi mật khẩu một cách an toàn.
* Không gian làm việc nhóm chung: Hệ thống đặc biệt chú trọng khả năng làm việc nhóm, cho phép các thành viên cộng tác hiệu quả. Trong không gian nhóm, bạn có thể:
* Phân quyền linh hoạt: Thiết lập các vai trò khác nhau (chủ sở hữu, thành viên cấp cao, thành viên) để quản lý quyền truy cập và trách nhiệm.
* Giao việc và theo dõi: Dễ dàng giao nhiệm vụ cho từng thành viên và theo dõi tiến độ công việc chung.
* Đồng bộ lịch trình: Mọi sự kiện và lịch hẹn quan trọng của nhóm đều được đồng bộ, giúp mọi người nắm bắt thông tin chung.
* Chia sẻ và quản lý tài chính: Cho phép theo dõi các giao dịch tài chính liên quan đến dự án hoặc hoạt động chung của nhóm.
* Giao diện người dùng thống nhất và trực quan: Mặc dù dữ liệu được tách bạch rõ ràng giữa không gian cá nhân và không gian nhóm, nhưng giao diện hệ thống được thiết kế để mang lại trải nghiệm liền mạch. Người dùng có thể dễ dàng chuyển đổi và quản lý thông tin trong cả hai không gian mà không gặp bất kỳ trở ngại nào. Đối với quản trị viên, có các công cụ chuyên biệt để quản lý người dùng và hệ thống một cách hiệu quả.
* Hệ thống được xây dựng với kiến trúc rõ ràng, tách bạch giữa phần xử lý logic (backend) và phần hiển thị cho người dùng (frontend). Điều này giúp đảm bảo sự toàn diện, linh hoạt và khả năng mở rộng của giải pháp.

# Thiết kế và Triển khai

## Các yêu cầu chức năng

* 1. **Bảo mật (Đăng ký/Đăng nhập)**
* **Đăng ký và xác thực**

Tính năng cho phép người dùng mới đăng ký tài khoản.

Tính năng xác minh email sau khi đăng ký để kích hoạt tài khoản.

* **Đăng nhập và quản lý phiên**

Tính năng cho phép người dùng đăng nhập vào hệ thống với thông tin xác thực.

Tính năng quản lý phiên người dùng (đăng xuất, duy trì phiên làm việc, đăng xuất tự động).

* **Quản lý mật khẩu**

Tính năng quên mật khẩu và gửi liên kết đặt lại mật khẩu.

Tính năng đặt lại mật khẩu thông qua liên kết được gửi.

* 1. **Quản lý người dùng (Admin/Guest)**
* **Quản lý người dùng bởi Admin(phân quyền theo vai trò admin/user)**

Tính năng cho phép admin xem danh sách tất cả người dùng trong hệ thống.

Tính năng cho phép admin cập nhật thông tin(nam, avatar, …), vai trò (admin/user), và mật khẩu của người dùng.

Tính năng cho phép admin xóa tài khoản người dùng.

Tính năng cho phép admin vô hiệu hóa người dùng.

* **Quản lý thông tin cá nhân bởi Guest**

Tính năng cho phép người dùng (guest) xem và cập nhật hồ sơ cá nhân (thông tin cơ bản).

Tính năng cho phép người dùng đổi mật khẩu cá nhân.

* 1. Quản lý nội dung Workspace
* **Quản lý thông tin Workspace**

Tính năng cho phép người dùng (có quyền) tạo workspace mới với tùy chọn public/private.

Tính năng cho phép xem chi tiết thông tin workspace (tên, mô tả, trạng thái public/private, ngày tạo, người tạo).

Tính năng cho phép người dùng (owner/senior) cập nhật thông tin workspace (tên, mô tả, trạng thái public/private, giao diện).

Tính năng cho phép xóa workspace (bởi owner).

Tính năng tìm kiếm workspace dựa trên tên hoặc từ khóa.

Tính năng cho phép người dùng thay đổi màu chữ và nền.

* **Quản lý thành viên**

Tính năng hiển thị danh sách thành viên trong workspace.

Tính năng cho phép người dùng (owner/senior) thêm thành viên mới với vai trò phân quyền (owner, editor, viewer).

Tính năng cho phép cập nhật vai trò thành viên (owner, editor, viewer) dựa trên phân quyền.

Tính năng cho phép xóa thành viên khỏi workspace (bởi owner/senior).

Tính năng tìm kiếm thành viên dựa trên tên hoặc vai trò.

* **Quản lý công việc (Tasks)**

Tính năng tạo mới công việc trong workspace (chỉ owner/editor).

Tính năng xem danh sách công việc và trạng thái (chưa làm, đang làm, đã làm) dựa trên phân quyền (owner/editor/viewer).

Tính năng cập nhật thông tin và trạng thái công việc (chỉ owner/editor).

Tính năng xóa công việc (chỉ owner/editor).

Tính năng gán công việc cho thành viên (chỉ owner/editor) và quản lý phân công.

Tính năng tìm kiếm công việc dựa trên tiêu đề hoặc trạng thái.

* **Quản lý ghi chú (Notes)**

Tính năng tạo mới ghi chú trong workspace (chỉ owner/editor).

Tính năng xem danh sách ghi chú dựa trên phân quyền (owner/editor/viewer).

Tính năng cập nhật và xóa ghi chú (chỉ owner/editor).

Tính năng tìm kiếm ghi chú dựa trên nội dung hoặc ngày tạo.

* **Quản lý sự kiện (Events)**

Tính năng tạo mới sự kiện trong workspace (chỉ owner/editor).

Tính năng xem danh sách sự kiện với trạng thái (chỉ owner/editor/viewer theo phân quyền).

Tính năng cập nhật và xóa sự kiện (chỉ owner/editor).

Tính năng tìm kiếm sự kiện dựa trên tiêu đề hoặc ngày.

* **Quản lý giao dịch (Transactions)**

Tính năng tạo mới giao dịch (thu/chi) trong workspace (chỉ owner/editor).

Tính năng xem danh sách giao dịch dựa trên phân quyền (owner/editor/viewer).

Tính năng cập nhật và xóa giao dịch (chỉ owner/editor).

Tính năng tìm kiếm giao dịch dựa trên danh mục hoặc loại (thu/chi).

* **Quản lý mục tiêu (Goals)**

Tính năng tạo mới mục tiêu với tiến độ trong workspace (chỉ owner/editor).

Tính năng xem danh sách mục tiêu và tiến độ dựa trên phân quyền (owner/editor/viewer).

Tính năng cập nhật tiến độ và xóa mục tiêu (chỉ owner/editor).

Tính năng tìm kiếm mục tiêu dựa trên tiêu đề hoặc tiến độ.

* **Tổng quan và điều hướng**

Tính năng hiển thị bảng điều khiển (dashboard) với các widget tổng hợp dựa trên phân quyền (owner/editor/viewer).

Tính năng cho phép người dùng chuyển đổi giữa các section (thành viên, công việc, v.v.) theo phân quyền.

Tính năng tìm kiếm nội dung tổng quan trên dashboard (nếu áp dụng).

## Các yêu cầu phi chức năng

* 1. Hiệu suất (Performance)

Yêu cầu về thời gian phản hồi nhanh chóng cho các thao tác cơ bản (tạo workspace, xem danh sách,…).

[Bổ sung] Yêu cầu hệ thống xử lý đồng thời tối đa 50 người dùng mà không giảm hiệu suất (dựa trên khả năng của Node.js và MySQL).

* 1. Bảo mật (Security)

Yêu cầu bảo vệ dữ liệu người dùng (đăng ký, mật khẩu) và phân quyền rõ ràng.

[Bổ sung] Yêu cầu mã hóa dữ liệu nhạy cảm (như mật khẩu) bằng thuật toán như bcrypt trước khi lưu vào cơ sở dữ liệu.

* 1. Khả năng sử dụng (Usability)

Yêu cầu giao diện thân thiện, dễ sử dụng cho mọi người.

[Bổ sung] Yêu cầu hỗ trợ đa ngôn ngữ (ít nhất tiếng Việt và tiếng Anh) để mở rộng người dùng.

* 1. Khả năng mở rộng (Scalability)

Yêu cầu hệ thống có thể hỗ trợ số lượng người dùng và workspace tăng trong tương lai.

[Bổ sung] Yêu cầu hỗ trợ thêm module mới (như báo cáo thống kê) mà không ảnh hưởng đến cấu trúc hiện tại.

* 1. Độ tin cậy (Reliability)

Yêu cầu hệ thống hoạt động ổn định với ít lỗi trong quá trình sử dụng.

[Bổ sung] Yêu cầu sao lưu dữ liệu tự động hàng ngày để tránh mất mát thông tin.

* 1. Tương thích (Compatibility)

Yêu cầu hệ thống hoạt động trên các trình duyệt phổ biến (Chrome, Firefox) và thiết bị khác nhau.

[Bổ sung] Yêu cầu tương thích với độ phân giải màn hình từ 320px (di động) đến 1920px (máy tính).

## Các ràng buộc (Constraints)

### Ràng buộc về triển khai

* Sử dụng công nghệ cụ thể (Node.js, React, MySQL) do giới hạn kỹ năng nhóm.
* Hệ thống chỉ chạy trên localhost trong giai đoạn phát triển.
* Yêu cầu máy chủ có cấu hình tối thiểu (RAM 4GB, CPU 2 nhân).
* Yêu cầu hoàn thành trong 2 tháng, do hạn chế thời gian đồ án.

### Ràng buộc kinh tế

* Ngân sách giới hạn trong phạm vi đồ án học tập.
* Không sử dụng dịch vụ đám mây trả phí (AWS, Azure) mà dựa vào máy chủ cục bộ.
* Chi phí nhân lực dựa trên thời gian tự nguyện của nhóm.

### Ràng buộc về đạo đức

* Bảo vệ thông tin cá nhân (email, mật khẩu) của người dùng.
* Tránh sử dụng dữ liệu không được phép từ bên thứ ba.
* Đảm bảo tính minh bạch khi quản lý phân quyền (admin, owner).
* Yêu cầu thông báo người dùng khi dữ liệu cá nhân được thu thập hoặc thay đổi.

## Mô hình hệ thống / Thiết kế giải pháp

### Các kịch bản hệ thống (Use Cases)

#### Đăng ký tài khoản mới

**Mô tả tổng quan**: Kịch bản này cho phép người dùng mới đăng ký một tài khoản để truy cập hệ thống.

**Actors**: Người dùng (User), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng chưa có tài khoản.

**Luồng sự kiện chính**:

Người dùng truy cập trang RegisterPage.js.

Hệ thống hiển thị form nhập thông tin (tên, email, mật khẩu).

Người dùng nhập dữ liệu và gửi yêu cầu.

Hệ thống xác thực dữ liệu qua authController.js và gửi email xác nhận (VerifyEmailPage.js).

Người dùng nhận email và xác nhận tài khoản.

Hệ thống lưu thông tin vào userModel.js và thông báo đăng ký thành công.

**Luồng sự kiện thay thế**:

Nếu email đã tồn tại (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không nhận được email (bước 5), người dùng yêu cầu gửi lại qua [ResetPasswordPage.js](http://resetpasswordpage.js).

**Điều kiện kết thúc**: Tài khoản mới được tạo và người dùng có thể đăng nhập.

#### Đăng nhập vào hệ thống

**Mô tả tổng quan**: Kịch bản này cho phép người dùng đã đăng ký truy cập vào hệ thống.

**Actors**: Người dùng (User), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng có tài khoản đã xác nhận.

**Luồng sự kiện chính**:

Người dùng truy cập trang LoginPage.js.

Hệ thống hiển thị form nhập email và mật khẩu.

Người dùng nhập thông tin và gửi yêu cầu.

Hệ thống xác thực thông tin qua authController.js.

Hệ thống cấp token và chuyển hướng đến HomePage.js hoặc [WorkspaceDetailPage.js](http://workspacedetailpage.js).

**Luồng sự kiện thay thế**:

Nếu thông tin sai (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu quên mật khẩu (bước 3), người dùng chuyển đến ResetPasswordPage.js để khôi phục.

**Điều kiện kết thúc**: Người dùng được đăng nhập và truy cập hệ thống.

#### Tạo workspace mới

**Mô tả tổng quan**: Kịch bản này cho phép người dùng (vai trò Owner) tạo một không gian làm việc mới.

**Actors**: Người dùng (User, vai trò Owner), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, có quyền tạo workspace qua [authMiddleware.js](http://authmiddleware.js).

**Luồng sự kiện chính**:

Người dùng truy cập HomePage.js hoặc WorkspaceDetailPage.js.

Hệ thống hiển thị tùy chọn "Tạo workspace".

Người dùng nhập thông tin (tên, mô tả, trạng thái) qua form.

Hệ thống gửi yêu cầu đến workspaceController.js để lưu vào workspaceModel.js.

Hệ thống tạo workspace và thông báo thành công.

**Luồng sự kiện thay thế**:

Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Workspace mới được tạo và hiển thị trong danh sách.

#### Quản lý thành viên trong workspace

**Mô tả tổng quan**: Kịch bản này cho phép người dùng (Owner/Senior) thêm, chỉnh sửa hoặc xóa thành viên trong workspace.

**Actors**: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, là Owner/Senior, có danh sách từ [workspaceMemberController.js](http://workspacemembercontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageMembers.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách thành viên qua MemberList.js.

Người dùng chọn "Thêm thành viên" và nhập thông tin (AddMember.js).

Hệ thống gửi yêu cầu đến workspaceMemberController.js để cập nhật workspaceMemberModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu thành viên đã tồn tại (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Danh sách thành viên được cập nhật.

#### Tạo và quản lý công việc

**Mô tả tổng quan**: Kịch bản này cho phép người dùng tạo, chỉnh sửa hoặc hoàn thành công việc trong workspace.

**Actors**: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập [taskController.js](http://taskcontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageTasks.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách công việc hiện tại.

Người dùng chọn "Tạo công việc" và điền thông tin (ManageTasksForm.js).

Hệ thống gửi yêu cầu đến taskController.js để lưu vào taskModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu công việc đã tồn tại (bước 4), hệ thống yêu cầu chỉnh sửa.

**Điều kiện kết thúc**: Công việc mới được tạo hoặc cập nhật.

#### Gán công việc cho thành viên

**Mô tả tổng quan**: Kịch bản này cho phép người dùng (Owner/Senior) gán công việc cho các thành viên trong workspace.

**Actors**: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, có quyền truy cập [taskAssignmentController.js](http://taskassignmentcontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageTaskAssignment.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách công việc và thành viên.

Người dùng chọn công việc và thành viên để gán (qua form).

Hệ thống gửi yêu cầu đến taskAssignmentController.js để lưu vào taskAssignmentModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu thành viên không thuộc workspace (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Công việc được gán thành công.

#### Quản lý ghi chú (Notes)

**Mô tả tổng quan**: Kịch bản này cho phép người dùng tạo và quản lý ghi chú trong workspace.

**Actors**: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập [noteController.js](http://notecontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageNotes.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách ghi chú hiện tại.

Người dùng chọn "Tạo ghi chú" và nhập nội dung.

Hệ thống gửi yêu cầu đến noteController.js để lưu vào noteModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu nội dung trống (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Ghi chú được tạo hoặc cập nhật.

#### Quản lý sự kiện (Events)

**Mô tả tổng quan**: Kịch bản này cho phép người dùng tạo và quản lý sự kiện trong workspace.

**Actors**: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập [eventController.js](http://eventcontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageEvents.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách sự kiện hiện tại.

Người dùng chọn "Tạo sự kiện" và nhập thông tin (thời gian, tiêu đề).

Hệ thống gửi yêu cầu đến eventController.js để lưu vào eventModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu thời gian xung đột (bước 4), hệ thống hiển thị cảnh báo.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Sự kiện được tạo hoặc cập nhật.

#### Quản lý mục tiêu (Goals)

**Mô tả tổng quan**: Kịch bản này cho phép người dùng tạo và theo dõi mục tiêu trong workspace.

**Actors**: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập [goalController.js](http://goalcontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageGoals.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách mục tiêu hiện tại.

Người dùng chọn "Tạo mục tiêu" và nhập thông tin (tiêu đề, tiến độ).

Hệ thống gửi yêu cầu đến goalController.js để lưu vào goalModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu tiến độ không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Mục tiêu được tạo hoặc cập nhật.

#### Quản lý giao dịch (Transactions)

**Mô tả tổng quan**: Kịch bản này cho phép người dùng ghi nhận và quản lý giao dịch tài chính trong workspace.

**Actors**: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, có quyền truy cập [transactionController.js](http://transactioncontroller.js).

**Luồng sự kiện chính**:

Người dùng truy cập ManageTransactions.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách giao dịch hiện tại.

Người dùng chọn "Thêm giao dịch" và nhập thông tin (loại, số tiền).

Hệ thống gửi yêu cầu đến transactionController.js để lưu vào transactionModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu số tiền không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Nếu không có quyền (bước 1), hệ thống từ chối yêu cầu.

**Điều kiện kết thúc**: Giao dịch được ghi nhận và hiển thị.

#### Xem và chỉnh sửa hồ sơ cá nhân

**Mô tả tổng quan**: Kịch bản này cho phép người dùng xem và cập nhật thông tin cá nhân.

**Actors**: Người dùng (User), Hệ thống (System).

**Điều kiện tiên quyết**: Người dùng đã đăng nhập, có quyền truy cập [ProfilePage.js](http://profilepage.js).

**Luồng sự kiện chính**:

Người dùng truy cập ProfilePage.js.

Hệ thống hiển thị thông tin cá nhân (tên, email).

Người dùng chỉnh sửa thông tin (nếu cần) và gửi yêu cầu.

Hệ thống gửi yêu cầu đến authController.js để cập nhật userModel.js.

Hệ thống thông báo thành công và cập nhật thông tin.

**Luồng sự kiện thay thế**:

Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

**Điều kiện kết thúc**: Hồ sơ cá nhân được cập nhật.

#### Quản lý người dùng (Chức năng Admin)

**Mô tả tổng quan**: Kịch bản này cho phép admin xem, thêm, chỉnh sửa hoặc xóa người dùng trong hệ thống.

**Actors**: Admin (User, vai trò Admin), Hệ thống (System).

**Điều kiện tiên quyết**: Admin đã đăng nhập, có quyền truy cập qua adminMiddleware.js và [AdminUsersPage.js](http://adminuserspage.js).

**Luồng sự kiện chính**:

Admin truy cập [AdminUsersPage.js](http://adminuserspage.js).

Hệ thống hiển thị danh sách người dùng từ [userModel.js](http://usermodel.js).

Admin chọn "Thêm người dùng" và nhập thông tin (tên, email, vai trò).

Hệ thống gửi yêu cầu đến authController.js để lưu hoặc cập nhật.

Hệ thống thông báo thành công và cập nhật danh sách.

**Luồng sự kiện thay thế**:

Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

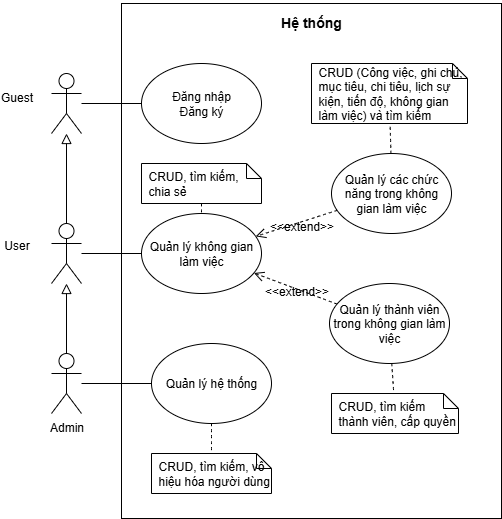
Nếu không có quyền (bước 1), hệ thống từ chối truy cập.

**Điều kiện kết thúc**: Danh sách người dùng được cập nhật.

### Mô hình Use-case

#### Sơ đồ Use-case tổng quan hệ thống

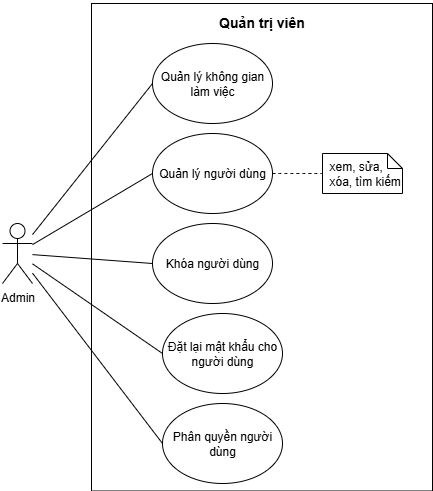
Sơ đồ tổng quan hệ thống bao gồm các chức năng chính và các tác nhân tham gia. Bao gồm người dùng với các vai trò khác nhau, hệ thống xử lý các yêu cầu như tạo workspace, quản lý công việc, ghi chú, sự kiện, mục tiêu, giao dịch, cũng như quản lý người dùng bởi admin.



Hình 1. Sơ đồ Use-Case tổng quan hệ thống

#### Sơ đồ Use-case của Admin

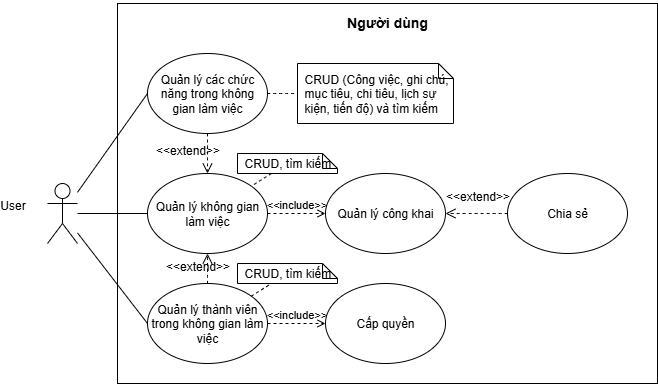
Admin có tất cả các chức năng của người dùng và các chức năng cơ bản như quyền xem danh sách, chỉnh sửa, xóa, tìm kiếm người dùng. Admin cũng có thêm các chức năng như khóa người dùng, đặt lại mật khẩu, phân quyền.



Hình 2. Sơ đồ Use-Case của admin

#### Sơ đồ Use-case của người dùng

Sơ đồ Use-case của người dùng bao gồm thêm, sửa, xóa và tìm kiếm workspace, thành viên, công việc, ghi chú, sự kiện, mục tiêu, và giao dịch. Trong đó quản người dùng có cấp quyền cho người dùng và quản lý không gian làm việc có thể chia sẻ không gian làm việc được(public/private).



Hình 3. Sơ đồ Use-Case của user

#### Sơ đồ Use-case chức năng chung của hệ thống

* + - 1. Xác thực và ủy quyền

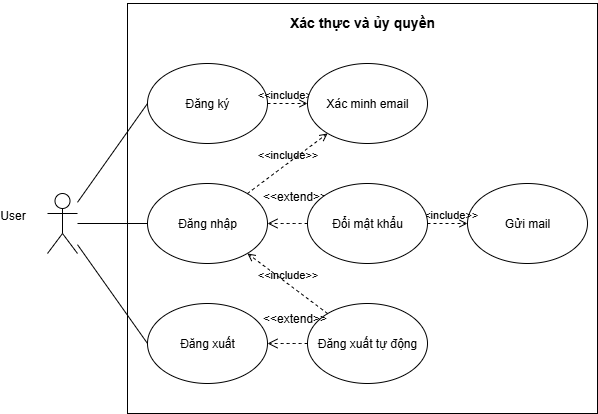
Cho phép người dùng tạo tài khoản mới để truy cập hệ thống bằng cách điền thông tin cơ bản (tên, email, mật khẩu) qua form đăng ký. Hệ thống xác thực dữ liệu và gửi email xác minh, sau đó lưu thông tin khi người dùng hoàn tất xác nhận.

Cho phép người dùng đã đăng ký truy cập hệ thống bằng cách nhập email và mật khẩu. Hệ thống xác thực thông tin, cấp token, và chuyển hướng đến trang chính nếu thành công.

Cho phép người dùng lấy lại quyền truy cập tài khoản khi quên mật khẩu. Người dùng truy cập chức năng "Quên mật khẩu" và nhập email đã đăng ký. Hệ thống gửi một liên kết đặt lại mật khẩu đến email đó. Người dùng có thể sử dụng liên kết này để đặt lại mật khẩu và truy cập lại tài khoản.

Cho phép người dùng kết thúc phiên làm việc hiện tại với hệ thống. Người dùng thực hiện thao tác đăng xuất (ví dụ: nhấp nút "Đăng xuất"). Hệ thống ngay lập tức thu hồi hoặc vô hiệu hóa token/phiên làm việc của người dùng, sau đó chuyển hướng họ trở lại trang đăng nhập hoặc trang công khai.

Ngoài ra, hệ thống sẽ tự động đăng xuất người dùng sau một khoảng thời gian không hoạt động nhất định để tăng cường bảo mật.

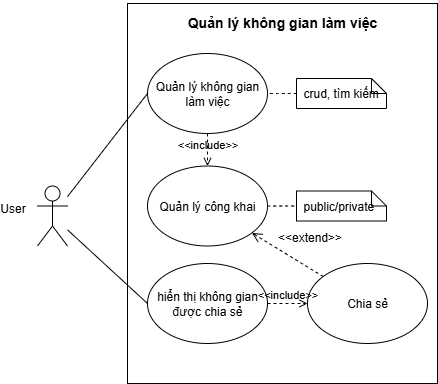


Hình 4. Sơ đồ Use-Case xác thực và ủy quyền

* + - 1. Quản lý không gian làm việc

Cho phép Người dùng (vai trò Owner) tạo mới, chỉnh sửa, xóa, tìm kiếm thông tin Workspace. Người dùng (Owner) có thể tạo mới một Workspace bằng cách điền các thông tin như tên, mô tả,... Hệ thống sẽ xử lý yêu cầu, xác thực dữ liệu, lưu các thay đổi vào cơ sở dữ liệu và cập nhật ngay lập tức danh sách Workspace hiển thị cho Owner.

Cho phép Người dùng (vai trò Owner) thiết lập chế độ truy cập (công khai hoặc riêng tư) cho Workspace của họ. Người dùng (Owner) có thể chọn chế độ hiển thị cho Workspace là "Công khai" (Public) hoặc "Riêng tư" (Private). Nếu là "Công khai", Workspace có thể được truy cập bởi bất kỳ ai hoặc qua liên kết chia sẻ. Nếu là "Riêng tư", Workspace chỉ có thể được truy cập bởi Owner và các thành viên được Owner mời cụ thể. Hệ thống sẽ cập nhật trạng thái này và áp dụng các quy tắc truy cập tương ứng.



Hình 5. Sơ đồ Use-Case quản lý không gian làm việc

* + - 1. Quản lý thành viên

Cho phép Người dùng (Owner) kiểm soát các thành viên trong Workspace và quyền hạn của họ. Người dùng (Owner) có thể thêm thành viên mới vào Workspace bằng cách nhập thông tin cần thiết. Họ cũng có thể chỉnh sửa vai trò hoặc xóa thành viên hiện có khỏi Workspace. Hệ thống sẽ xác thực quyền của Owner và xử lý các yêu cầu này, sau đó cập nhật danh sách thành viên và áp dụng quyền mới ngay lập tức trong Workspace.

**Owner:**

Có quyền quản lý tất cả các chức năng trong Workspace, bao gồm quản lý thành viên (thêm, chỉnh sửa, xóa thành viên) và quản lý công việc (ghi chú, mục tiêu, giao dịch, lịch, công việc).

**Editor:**

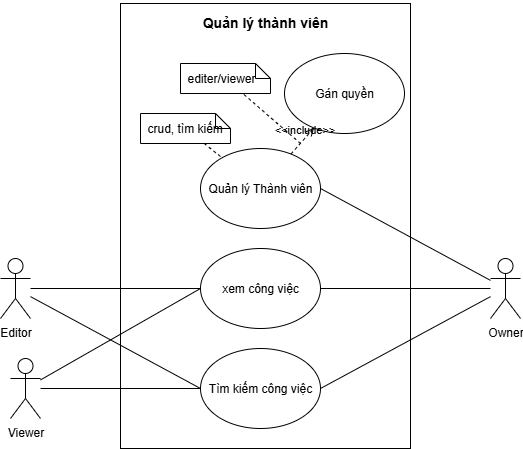
Được phép quản lý tất cả các chức năng liên quan đến quản lý công việc bên trong Workspace (ví dụ: tạo, chỉnh sửa, xóa ghi chú, mục tiêu, giao dịch, lịch, công việc).

Không được phép quản lý thành viên (tức là không thể thêm, chỉnh sửa, hoặc xóa thành viên khác khỏi Workspace).

**Viewer:**

Chỉ được phép xem tất cả các thông tin và nội dung trong Workspace.

Không có quyền chỉnh sửa hoặc quản lý bất kỳ chức năng nào, bao gồm cả thành viên và công việc.



Hình 6. Sơ đồ Use-Case quản lý thành viên

* + - 1. Quản lý công việc

Cho phép người dùng (Owner/editor) tạo, chỉnh sửa hoặc hoàn thành công việc trong Workspace. Người dùng có thể tạo công việc mới, chỉnh sửa thông tin của công việc hiện có, hoặc đánh dấu công việc là đã hoàn thành, . Hệ thống sẽ lưu trữ tất cả thông tin về công việc và cập nhật trạng thái của chúng ngay khi có bất kỳ thay đổi nào từ phía người dùng.

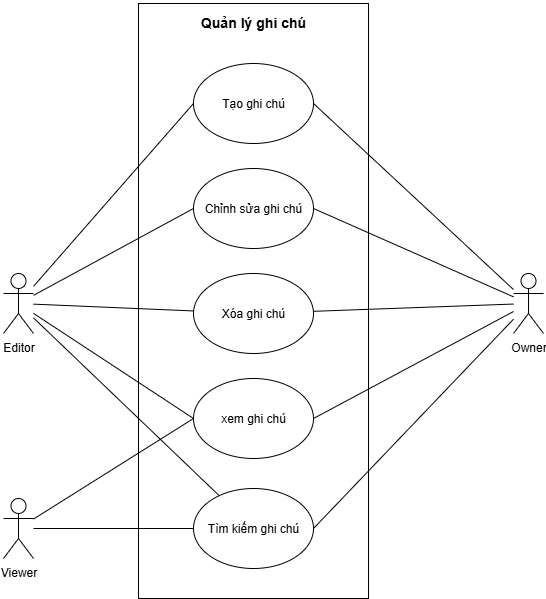
Cho phép Người dùng (Owner/editor) phân công công việc cho các thành viên trong Workspace. Người dùng có vai trò Owner hoặc editor có thể chọn một công việc cụ thể và gán cho một hoặc nhiều thành viên trong cùng Workspace. Hệ thống sẽ xác nhận quyền của người gán và lưu thông tin phân công này, đồng thời cập nhật trạng thái công việc được gán cho thành viên liên quan.



Hình 7. Sơ đồ Use-Case quản lý công việc

* + - 1. Quản lý ghi chú

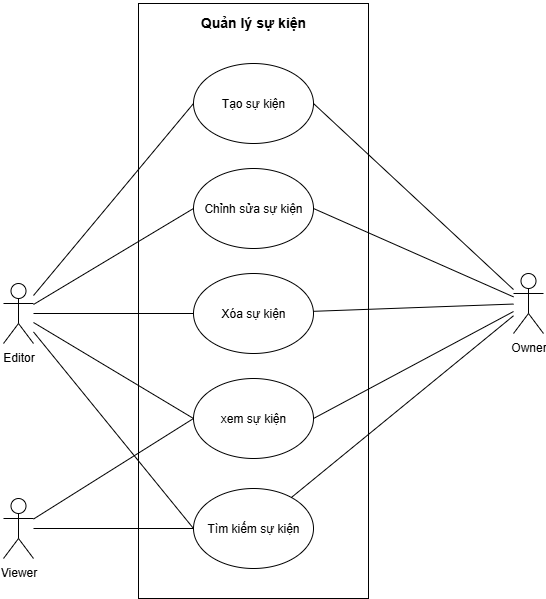
Cho phép người dùng tạo và quản lý ghi chú trong Workspace. Người dùng có thể tạo các ghi chú mới để lưu trữ thông tin nhanh hoặc quan trọng trong Workspace của mình. Hệ thống sẽ lưu lại toàn bộ nội dung ghi chú và hiển thị chúng dưới dạng danh sách hoặc các bố cục khác nhau để người dùng dễ dàng xem và quản lý.



Hình 8. Sơ đồ Use-Case quản lý ghi chú

* + - 1. Quản lý sự kiện

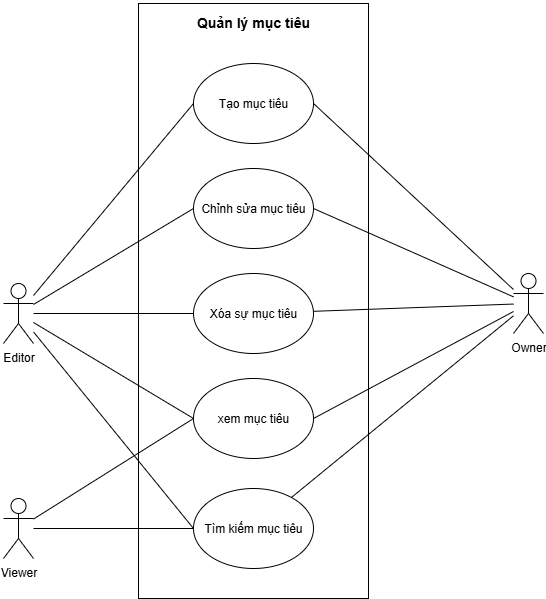
Cho phép người dùng tạo và quản lý sự kiện trong Workspace. Người dùng có thể lên lịch và tạo các sự kiện mới trong Workspace, bao gồm thông tin về thời gian, địa điểm và mô tả. Hệ thống sẽ kiểm tra các xung đột về thời gian với các sự kiện đã có để tránh trùng lặp, sau đó lưu thông tin sự kiện và hiển thị trên lịch hoặc danh sách sự kiện cho người dùng.



Hình 9. Sơ đồ Use-Case quản lý sự kiện

Quản lý mục tiêu

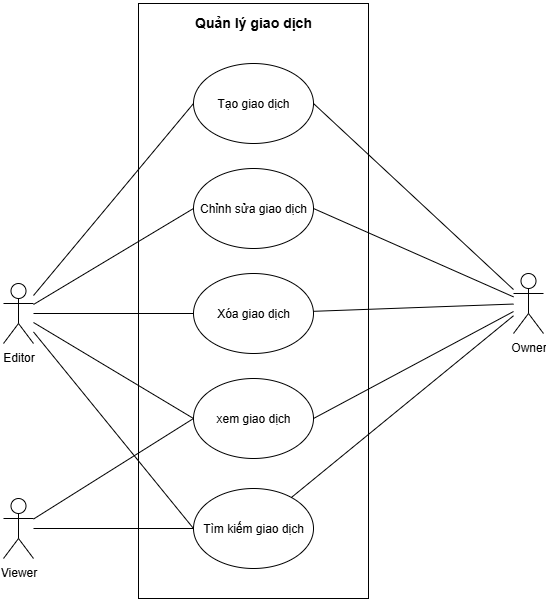
Cho phép người dùng tạo và theo dõi tiến độ mục tiêu trong Workspace. Người dùng có thể thiết lập các mục tiêu mới, đặt ra các mốc thời gian và mô tả chi tiết cho từng mục tiêu. Hệ thống sẽ lưu trữ thông tin này và cho phép người dùng cập nhật tiến độ đạt được của mỗi mục tiêu, hiển thị tổng quan để dễ dàng theo dõi.



Hình 10. Sơ đồ Use-Case quản lý mục tiêu

* + - 1. Quản lý giao dịch

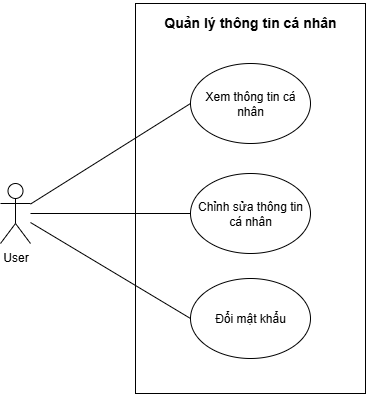
Cho phép Người dùng (Owner/Senior) ghi nhận và quản lý các giao dịch tài chính trong Workspace. Người dùng có vai trò Owner hoặc Senior có thể nhập các chi tiết về giao dịch tài chính (ví dụ: khoản thu, chi, mô tả). Hệ thống sẽ lưu trữ thông tin giao dịch này và hiển thị tổng hợp các giao dịch, giúp người dùng dễ dàng theo dõi dòng tiền trong Workspace.



Hình 11. Sơ đồ Use-Case quản lý giao dịch

* + - 1. Quản lý thông tin cá nhân

Cho phép người dùng xem và cập nhật thông tin cá nhân của họ. Người dùng có thể truy cập hồ sơ cá nhân của mình để xem các thông tin đã lưu như tên hoặc địa chỉ email. Người dùng cũng có quyền chỉnh sửa các thông tin này khi cần và có thể đổi mật khẩu. Hệ thống sẽ xác thực các thay đổi và lưu chúng vào cơ sở dữ liệu, cập nhật hồ sơ cá nhân của người dùng.



Hình 12. Sơ đồ Use-Case quản lý thông tin cá nhân

### Mô hình lớp và đối tượng

#### Các lớp chính trong hệ thống

User: Thông tin người dùng (id, name, email, password, role, created\_at, updated\_at, is\_verified, is\_active, verification\_token, verification\_expires, reset\_password\_token, reset\_password\_expires, avatar\_url).

Workspace: Không gian làm việc (id, name, description, is\_private, created\_by, created\_at, updated\_at, theme\_color, background\_image).

WorkspaceMember: Mối quan hệ giữa User và Workspace (id, user\_id, workspace\_id, role, joined\_at).

Task: Công việc (id, workspace\_id, title, description, status, deadline, created\_at, updated\_at, created\_by).

TaskAssignment: Gán công việc (id, task\_id, user\_id, assigned\_at).

Note: Ghi chú (id, user\_id, workspace\_id, content, created\_at, updated\_at).

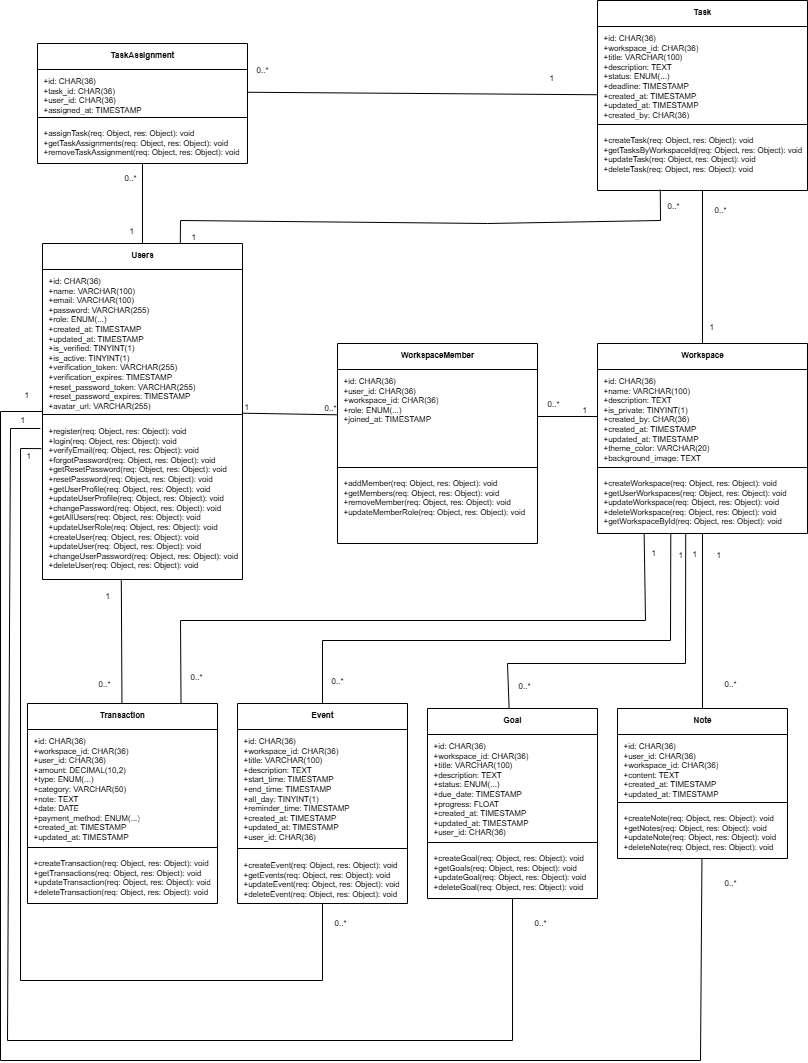
Event: Sự kiện (id, workspace\_id, title, description, start\_time, end\_time, all\_day, reminder\_time, created\_at, updated\_at, user\_id).

Goal: Mục tiêu (id, workspace\_id, title, description, status, due\_date, progress, created\_at, updated\_at, user\_id).

Transaction: Giao dịch (id, workspace\_id, user\_id, amount, type, category, note, date, payment\_method, created\_at, updated\_at).

#### Sơ đồ lớp tổng quan hệ thống

Sơ đồ lớp tổng quan thể hiện các lớp chính và mối quan hệ giữa chúng, bao gồm các liên kết (association) giữa User, Workspace, và các thực thể khác như Task, Note, Event, Goal, Transaction.



Hình 13. Sơ đồ lớp tổng quan hệ thống

#### Mô tả chi tiết các lớp và mối quan hệ

##### Lớp User

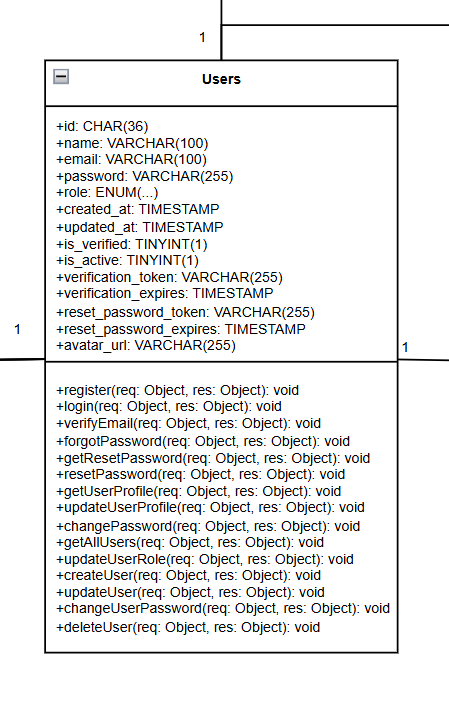
Mô tả: Lớp User đại diện cho thông tin cá nhân của người dùng trong hệ thống, bao gồm id (CHAR(36)), name (VARCHAR(100)), email (VARCHAR(100)), password (VARCHAR(255)), role (ENUM: user, admin), created\_at (TIMESTAMP), updated\_at (TIMESTAMP), is\_verified (TINYINT(1)), is\_active (TINYINT(1)), verification\_token (VARCHAR(255)), verification\_expires (TIMESTAMP), reset\_password\_token (VARCHAR(255)), reset\_password\_expires (TIMESTAMP), và avatar\_url (VARCHAR(255)). Lớp này quản lý đăng ký, đăng nhập, xác minh email, đặt lại mật khẩu, và các chức năng quản lý người dùng (bao gồm cả admin).

Mối quan hệ:

Liên kết với WorkspaceMember (một User có thể tham gia nhiều WorkspaceMember, thể hiện vai trò thành viên trong các không gian làm việc, với multiplicity 1 ở đầu User và 0..\* ở đầu WorkspaceMember).

Liên kết với TaskAssignment (một User có thể được gán nhiều TaskAssignment, thể hiện việc được phân công công việc, với multiplicity 1 ở đầu User và 0..\* ở đầu TaskAssignment).

Liên kết với Transaction (một User có thể tạo nhiều Transaction, thể hiện các giao dịch tài chính, với multiplicity 1 ở đầu User và 0..\* ở đầu Transaction).



Hình 14. Sơ đồ lớp User

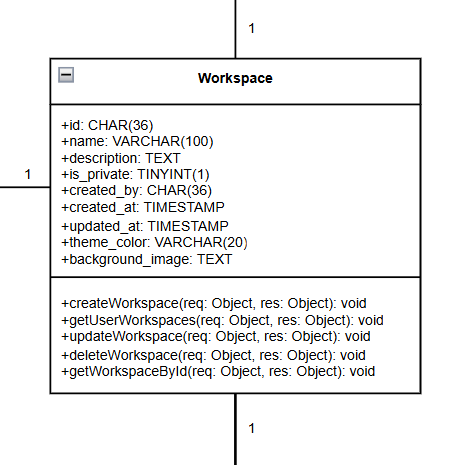
##### Lớp Workspace

Mô tả: Lớp Workspace đại diện cho không gian làm việc, với id (CHAR(36)), name (VARCHAR(100)), description (TEXT), is\_private (TINYINT(1)), created\_by (CHAR(36)), created\_at (TIMESTAMP), updated\_at (TIMESTAMP), theme\_color (VARCHAR(20)), và background\_image (TEXT). Lớp này quản lý việc tạo, cập nhật, xóa, và truy xuất thông tin không gian làm việc, bao gồm cả thiết lập quyền riêng tư và tùy chỉnh giao diện.

Mối quan hệ:

Liên kết với WorkspaceMember (một Workspace chứa nhiều WorkspaceMember, thể hiện các thành viên tham gia, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu WorkspaceMember).

Liên kết với Task, Note, Event, Goal, Transaction (một Workspace chứa nhiều thực thể tương ứng, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu các lớp khác).



Hình 15. Sơ đồ lớp Workspace

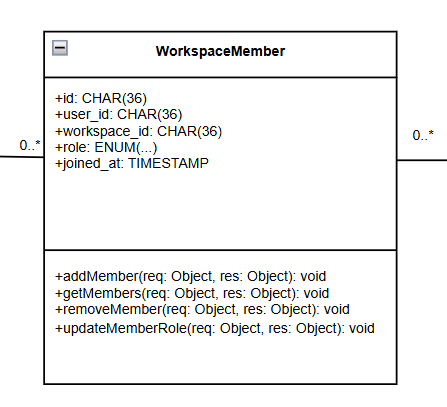
##### Lớp WorkspaceMember

Mô tả: Lớp WorkspaceMember thể hiện mối quan hệ giữa User và Workspace, với id (CHAR(36)), user\_id (CHAR(36)), workspace\_id (CHAR(36)), role (ENUM: member, senior, owner), và joined\_at (TIMESTAMP). Lớp này quản lý vai trò và quyền hạn của thành viên trong không gian làm việc, bao gồm thêm, xóa, và cập nhật vai trò.

Mối quan hệ:

Liên kết với User (một User có thể tham gia nhiều WorkspaceMember, với multiplicity 1 ở đầu User và 0..\* ở đầu WorkspaceMember).

Liên kết với Workspace (một Workspace chứa nhiều WorkspaceMember, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu WorkspaceMember).



Hình 16. Sơ đồ lớp WorkspaceMember

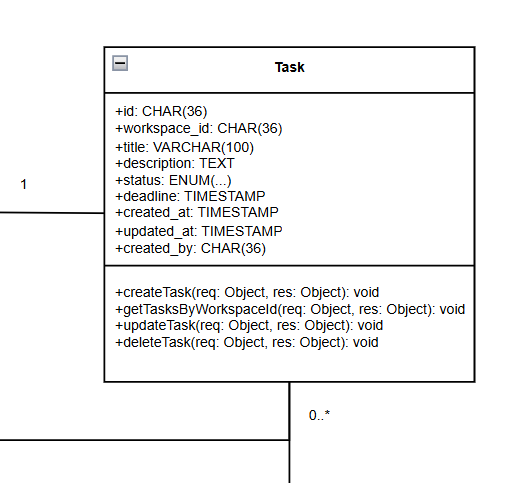
##### Lớp Task

Mô tả: Lớp Task đại diện cho công việc trong không gian làm việc, với id (CHAR(36)), workspace\_id (CHAR(36)), title (VARCHAR(100)), description (TEXT), status (ENUM: pending, in\_progress, completed), deadline (TIMESTAMP), created\_at (TIMESTAMP), updated\_at (TIMESTAMP), và created\_by (CHAR(36)). Lớp này quản lý việc tạo, cập nhật, xóa, và liệt kê công việc, với quyền hạn dựa trên vai trò trong Workspace.

Mối quan hệ:

Liên kết với TaskAssignment (một Task có thể được gán qua nhiều TaskAssignment, với multiplicity 1 ở đầu Task và 0..\* ở đầu TaskAssignment).

Liên kết với Workspace (một Workspace chứa nhiều Task, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu Task).



Hình 17. Sơ đồ lớp Task

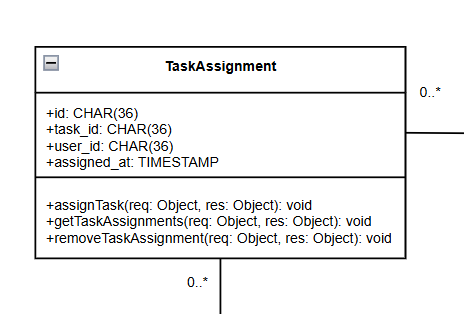
##### Lớp TaskAssignment

Mô tả: Lớp TaskAssignment quản lý việc gán công việc cho thành viên, với id (CHAR(36)), task\_id (CHAR(36)), user\_id (CHAR(36)), và assigned\_at (TIMESTAMP). Lớp này hỗ trợ gán, liệt kê, và xóa gán công việc, với quyền hạn thuộc về owner hoặc senior.

Mối quan hệ:

Liên kết với Task (một Task có nhiều TaskAssignment, với multiplicity 1 ở đầu Task và 0..\* ở đầu TaskAssignment).

Liên kết với User (một User có nhiều TaskAssignment, với multiplicity 1 ở đầu User và 0..\* ở đầu TaskAssignment).



Hình 18. Sơ đồ lớp TaskAssignment

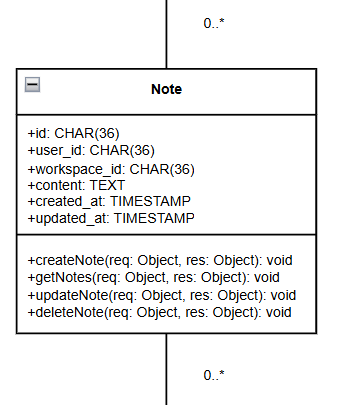
##### Lớp Note

Mô tả: Lớp Note đại diện cho ghi chú trong không gian làm việc, với id (CHAR(36)), user\_id (CHAR(36)), workspace\_id (CHAR(36)), content (TEXT), created\_at (TIMESTAMP), và updated\_at (TIMESTAMP). Lớp này quản lý việc tạo, cập nhật, xóa, và liệt kê ghi chú, với quyền hạn dựa trên vai trò hoặc người tạo.

Mối quan hệ:

Liên kết với User (một User tạo nhiều Note, với multiplicity 1 ở đầu User và 0..\* ở đầu Note).

Liên kết với Workspace (một Workspace chứa nhiều Note, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu Note).



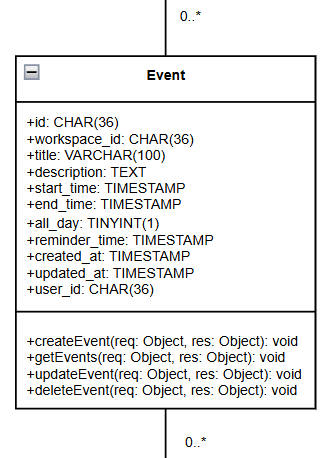
Hình 19. Sơ đồ lớp Note

##### Lớp Event

Mô tả: Lớp Event đại diện cho sự kiện trong không gian làm việc, với id (CHAR(36)), workspace\_id (CHAR(36)), title (VARCHAR(100)), description (TEXT), start\_time (TIMESTAMP), end\_time (TIMESTAMP), all\_day (TINYINT(1)), reminder\_time (TIMESTAMP), created\_at (TIMESTAMP), updated\_at (TIMESTAMP), và user\_id (CHAR(36)). Lớp này quản lý việc tạo, cập nhật, xóa, và liệt kê sự kiện, với quyền hạn thuộc về owner hoặc senior.

Mối quan hệ:

Liên kết với Workspace (một Workspace chứa nhiều Event, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu Event).



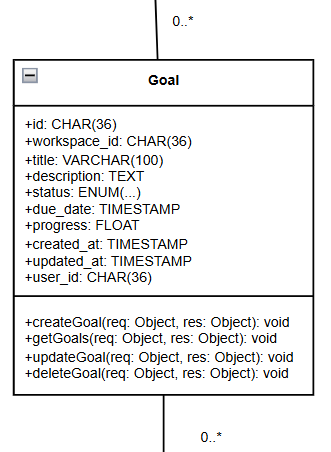
Hình 20. Sơ đồ lớp Event

##### Lớp Goal

Mô tả: Lớp Goal đại diện cho mục tiêu trong không gian làm việc, với id (CHAR(36)), workspace\_id (CHAR(36)), title (VARCHAR(100)), description (TEXT), status (ENUM), due\_date (TIMESTAMP), progress (FLOAT), created\_at (TIMESTAMP), updated\_at (TIMESTAMP), và user\_id (CHAR(36)). Lớp này quản lý việc tạo, cập nhật, xóa, và liệt kê mục tiêu, với quyền hạn thuộc về owner hoặc senior.

Mối quan hệ:

Liên kết với Workspace (một Workspace chứa nhiều Goal, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu Goal).



Hình 21. Sơ đồ lớp Goal

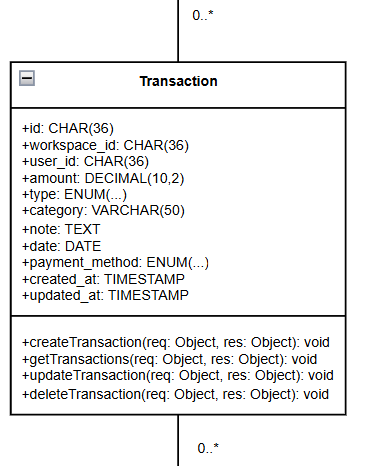
##### Lớp Transaction

Mô tả: Lớp Transaction đại diện cho giao dịch tài chính trong không gian làm việc, với id (CHAR(36)), workspace\_id (CHAR(36)), user\_id (CHAR(36)), amount (DECIMAL(10,2)), type (ENUM), category (VARCHAR(50)), note (TEXT), date (DATE), payment\_method (ENUM), created\_at (TIMESTAMP), và updated\_at (TIMESTAMP). Lớp này quản lý việc tạo, cập nhật, xóa, và liệt kê giao dịch, với quyền hạn thuộc về owner hoặc senior.

Mối quan hệ:

Liên kết với Workspace (một Workspace chứa nhiều Transaction, với multiplicity 1 ở đầu Workspace và 0..\* ở đầu Transaction).

Liên kết với User (một User tạo nhiều Transaction, với multiplicity 1 ở đầu User và 0..\* ở đầu Transaction).



Hình 22. Sơ đồ lớp Transaction

#### Sơ đồ cơ sở dữ liệu

Sơ đồ cơ sở dữ liệu bao gồm các bảng chính tương ứng với các lớp trong hệ thống, với các trường, khóa chính (Primary Key), khóa ngoại (Foreign Key), và các ràng buộc (Constraints) như sau:

* + - 1. **Bảng Users**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

name (VARCHAR(100), NOT NULL)

email (VARCHAR(100), UNIQUE, NOT NULL)

password (VARCHAR(255), NOT NULL)

role (ENUM('user', 'admin'), NOT NULL)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

is\_verified (TINYINT(1), DEFAULT 0)

is\_active (TINYINT(1), DEFAULT 1)

verification\_token (VARCHAR(255))

verification\_expires (TIMESTAMP)

reset\_password\_token (VARCHAR(255))

reset\_password\_expires (TIMESTAMP)

avatar\_url (VARCHAR(255))

Mô tả: Bảng này lưu trữ thông tin người dùng, với id là khóa chính duy nhất, email là duy nhất để tránh trùng lặp, và các trường như verification\_token hỗ trợ chức năng xác minh email.

* + - 1. **Bảng Workspaces**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

name (VARCHAR(100), NOT NULL)

description (TEXT)

is\_private (TINYINT(1), NOT NULL)

created\_by (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

theme\_color (VARCHAR(20))

background\_image (TEXT)

Mô tả: Bảng này lưu trữ thông tin không gian làm việc, với created\_by là khóa ngoại liên kết với bảng Users, thể hiện người tạo workspace.

Bảng WorkspaceMembers

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

role (ENUM('member', 'senior', 'owner'), NOT NULL)

joined\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

Mô tả: Bảng này quản lý mối quan hệ giữa User và Workspace, với user\_id và workspace\_id là khóa ngoại tạo mối quan hệ nhiều-nhiều, và role xác định vai trò thành viên.

* + - 1. **Bảng Tasks**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

title (VARCHAR(100), NOT NULL)

description (TEXT)

status (ENUM('pending', 'in\_progress', 'completed'), NOT NULL)

deadline (TIMESTAMP)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

created\_by (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

Mô tả: Bảng này lưu trữ thông tin công việc, với workspace\_id và created\_by là khóa ngoại liên kết với Workspaces và Users.

* + - 1. **Bảng TaskAssignments**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

task\_id (CHAR(36), FOREIGN KEY REFERENCES Tasks(id), NOT NULL)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

assigned\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

Mô tả: Bảng này quản lý việc gán công việc, với task\_id và user\_id là khóa ngoại tạo mối quan hệ giữa Tasks và Users.

* + - 1. **Bảng Notes**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

content (TEXT, NOT NULL)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

Mô tả: Bảng này lưu trữ ghi chú, với user\_id và workspace\_id là khóa ngoại liên kết với Users và Workspaces.

* + - 1. **Bảng Events**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

title (VARCHAR(100), NOT NULL)

description (TEXT)

start\_time (TIMESTAMP, NOT NULL)

end\_time (TIMESTAMP, NOT NULL)

all\_day (TINYINT(1), DEFAULT 0)

reminder\_time (TIMESTAMP)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

Mô tả: Bảng này lưu trữ thông tin sự kiện, với workspace\_id và user\_id là khóa ngoại liên kết với Workspaces và Users.

* + - 1. **Bảng Goals**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

title (VARCHAR(100), NOT NULL)

description (TEXT)

status (ENUM('not\_started', 'in\_progress', 'completed'), NOT NULL)

due\_date (TIMESTAMP)

progress (FLOAT, DEFAULT 0)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

Mô tả: Bảng này lưu trữ thông tin mục tiêu, với workspace\_id và user\_id là khóa ngoại liên kết với Workspaces và Users.

* + - 1. **Bảng Transactions**

Cấu trúc:

id (CHAR(36), PRIMARY KEY)

workspace\_id (CHAR(36), FOREIGN KEY REFERENCES Workspaces(id), NOT NULL)

user\_id (CHAR(36), FOREIGN KEY REFERENCES Users(id), NOT NULL)

amount (DECIMAL(10,2), NOT NULL)

type (ENUM('income', 'expense'), NOT NULL)

category (VARCHAR(50))

note (TEXT)

date (DATE, NOT NULL)

payment\_method (ENUM('cash', 'bank', 'other'), NOT NULL)

created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

updated\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

Mô tả: Bảng này lưu trữ thông tin giao dịch, với workspace\_id và user\_id là khóa ngoại liên kết với Workspaces và Users.

* + - 1. **Mối quan hệ và ràng buộc**

**Khóa ngoại (Foreign Keys):**

created\_by trong Workspaces tham chiếu đến id trong Users.

user\_id và workspace\_id trong WorkspaceMembers tham chiếu đến id trong Users và Workspaces.

workspace\_id và created\_by trong Tasks tham chiếu đến id trong Workspaces và Users.

task\_id và user\_id trong TaskAssignments tham chiếu đến id trong Tasks và Users.

user\_id và workspace\_id trong Notes tham chiếu đến id trong Users và Workspaces.

workspace\_id và user\_id trong Events tham chiếu đến id trong Workspaces và Users.

workspace\_id và user\_id trong Goals tham chiếu đến id trong Workspaces và Users.

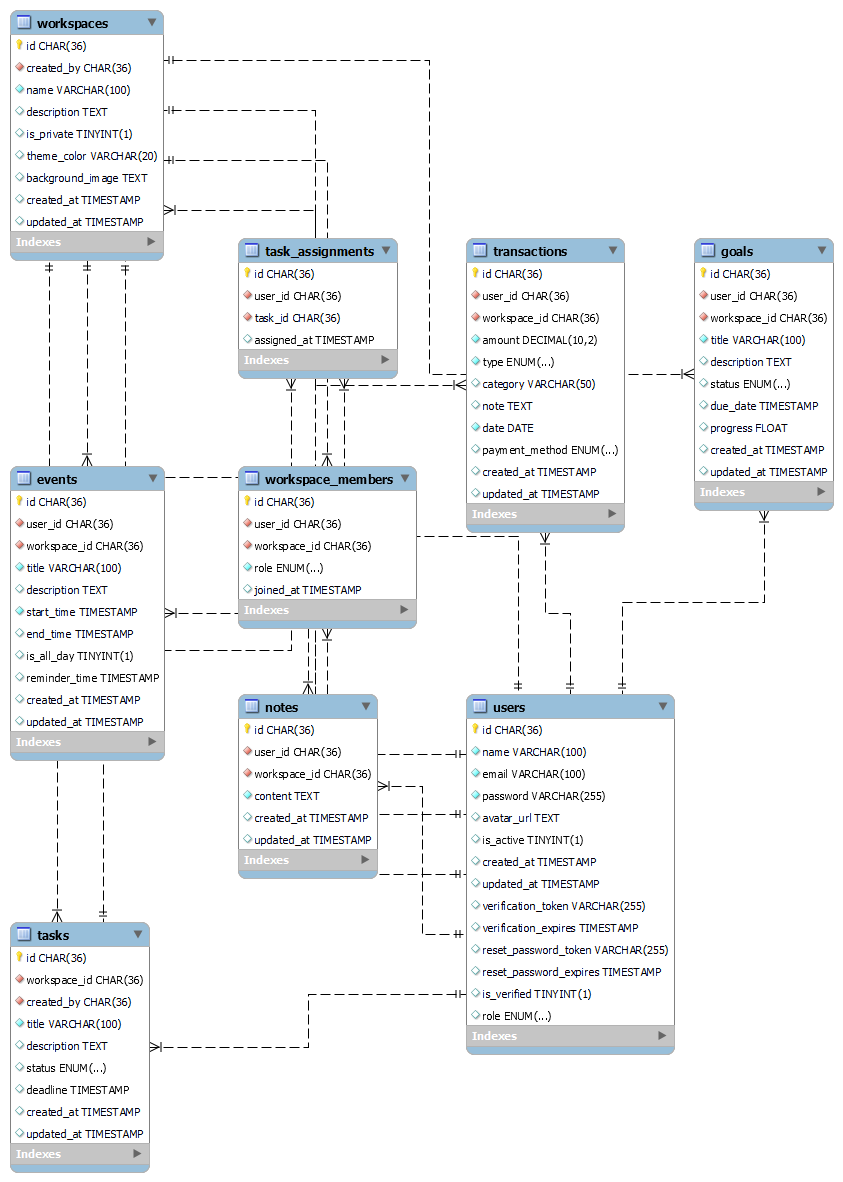
workspace\_id và user\_id trong Transactions tham chiếu đến id trong Workspaces và Users.

**Ràng buộc:**

Các trường như email trong Users có ràng buộc UNIQUE để đảm bảo không trùng lặp.

Các trường NOT NULL đảm bảo dữ liệu bắt buộc (ví dụ: title trong Tasks, amount trong Transactions).

Ràng buộc ON DELETE CASCADE hoặc ON UPDATE CASCADE có thể được áp dụng cho các khóa ngoại để xử lý xóa hoặc cập nhật liên quan (tùy cấu hình).



Hình 23. Sơ đồ cơ sở dữ liệu

### Các biểu đồ tuần tự

* + 1. Sơ đồ tuần tự chức năng Đăng nhập

Biểu đồ này mô tả luồng xử lý khi người dùng thực hiện thao tác đăng nhập vào hệ thống. Nó minh họa các bước từ khi người dùng nhập thông tin cho đến khi hệ thống xác thực và phản hồi kết quả.

Mô tả chi tiết:

Người dùng (Actor) nhập email và mật khẩu tại Trang Đăng Nhập.

Trang Đăng Nhập gửi thông tin email và mật khẩu tới Auth Controller.

Auth Controller tiếp nhận và chuyển tiếp thông tin này đến User Model.

User Model gửi yêu cầu truy vấn tất cả email và mật khẩu hiện có trong cơ sở dữ liệu MySQL.

MySQL trả về kết quả truy vấn cho User Model.

User Model thực hiện kiểm tra xem email và mật khẩu được cung cấp có hợp lệ và tồn tại trong hệ thống hay không.

Phân nhánh (Alt):

Trường hợp [Thành công]: Nếu email và mật khẩu hợp lệ:

User Model gửi trạng thái "email và pass hợp lệ" cho Auth Controller.

Auth Controller thông báo "Đăng nhập thành công" cho Trang Đăng Nhập.

Auth Controller yêu cầu Trang Chủ thực hiện "Chuyển trang".

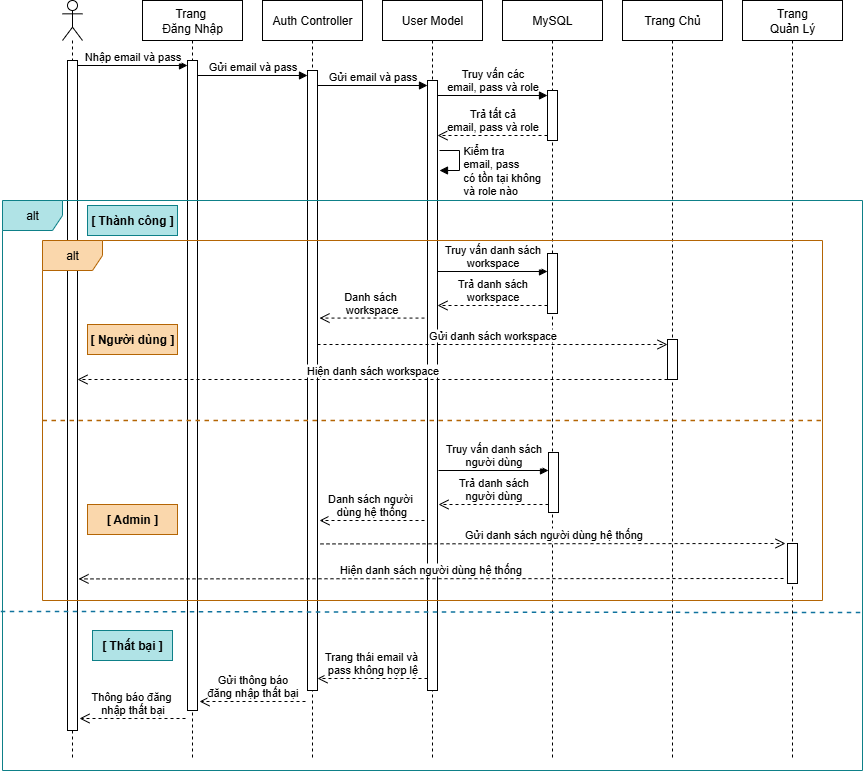
Trang Chủ hiển thị giao diện chính của ứng dụng.

Trường hợp [Thất bại]: Nếu email hoặc mật khẩu không hợp lệ/không tồn tại:

User Model gửi trạng thái "email và pass không hợp lệ" cho Auth Controller.

Auth Controller gửi thông báo "đăng nhập thất bại" cho Trang Đăng Nhập.

Trang Đăng Nhập hiển thị "Thông báo đăng nhập thất bại" cho Người dùng.



Hình 24. sơ đồ tuần tự chức năng đăng nhập

Sơ đồ tuần tự chức năng Quên mật khẩu

Biểu đồ này trình bày quá trình người dùng yêu cầu khôi phục mật khẩu khi bị quên. Nó bao gồm các bước từ khi người dùng nhập email đến khi mật khẩu mới được cập nhật và thông báo.

Mô tả chi tiết:

Người dùng (Actor) nhập email tại Trang Quên Mật Khẩu.

Trang Quên Mật Khẩu gửi email này đến Auth Controller.

Auth Controller chuyển tiếp email đến User Model.

User Model yêu cầu MySQL truy vấn kiểm tra sự tồn tại của email.

MySQL trả về kết quả truy vấn.

User Model kiểm tra sự tồn tại của email.

Phân nhánh (Alt):

Trường hợp [Email không tồn tại]:

User Model gửi trạng thái "email không tồn tại" cho Auth Controller.

Auth Controller gửi thông báo "email không tồn tại" cho Trang Quên Mật Khẩu.

Trang Quên Mật Khẩu hiển thị "Thông báo email không tồn tại" cho Người dùng.

Trường hợp [Email Tồn Tại]:

User Model gửi trạng thái "kiểm tra email để đổi mật khẩu" cho Auth Controller.

Auth Controller gửi "Thông báo kiểm tra email để đổi mật khẩu" cho Trang Quên Mật Khẩu.

Trang Quên Mật Khẩu hiển thị thông báo tương ứng cho Người dùng.

Auth Controller gửi yêu cầu "Xác minh email chính chủ" đến dịch vụ Email.

Email gửi lại thông báo xác minh (ví dụ: link reset) cho Auth Controller.

Auth Controller yêu cầu Trang Đổi Mật Khẩu thực hiện "Chuyển trang".

Trang Đổi Mật Khẩu hiển thị giao diện nhập mật khẩu mới và gửi "Thông báo đổi pass" cho Người dùng.

Người dùng nhập mật khẩu mới.

Trang Đổi Mật Khẩu gửi email và mật khẩu mới đến Auth Controller.

Auth Controller gửi email và mật khẩu mới đến User Model.

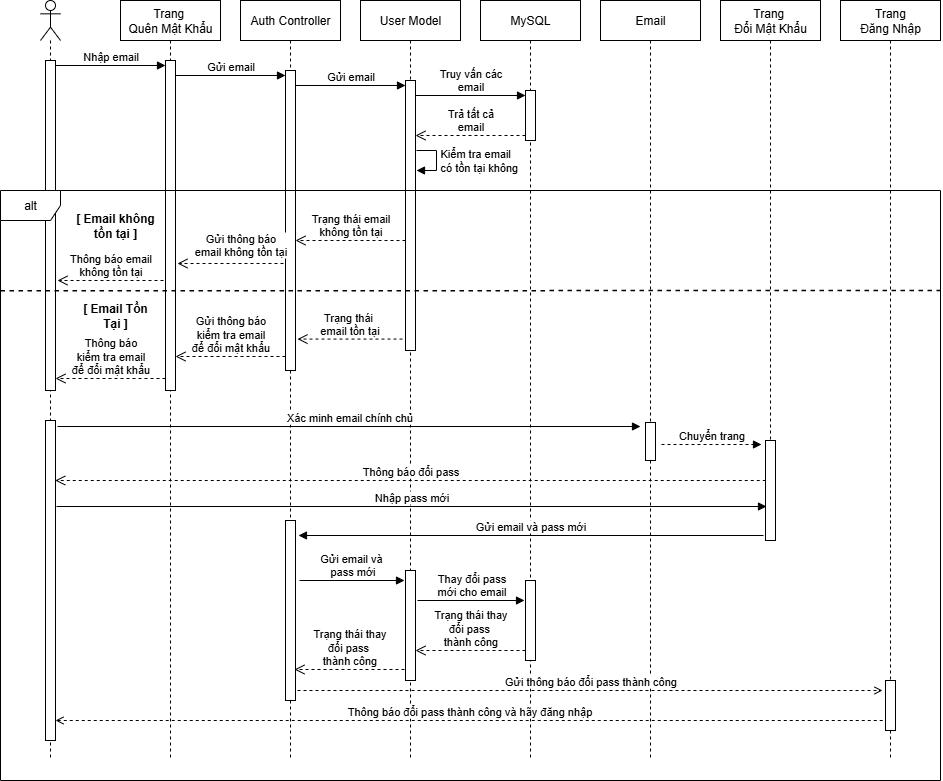
User Model yêu cầu MySQL "Thay đổi pass mới cho email" tương ứng.

MySQL xác nhận thay đổi thành công.

User Model gửi trạng thái "thay đổi đổi pass thành công" cho Auth Controller.

Auth Controller gửi "Thông báo đổi pass thành công và hãy đăng nhập" cho Trang Đăng Nhập.

Trang Đăng Nhập hiển thị thông báo và cho phép người dùng đăng nhập lại.



Hình 25. sơ đồ tuần tự chức năng quên mật khẩu

* + 1. Sơ đồ tuần tự chức năng Đăng ký

Biểu đồ này minh họa quy trình người dùng tạo tài khoản mới trong hệ thống, bao gồm các bước xác thực thông tin và gửi email xác minh.

Mô tả chi tiết:

Người dùng (Actor) nhập tên, email và mật khẩu tại Trang Đăng Ký.

Trang Đăng Ký gửi thông tin tên, email, mật khẩu tới Auth Controller.

Auth Controller chuyển tiếp thông tin này đến User Model.

User Model yêu cầu MySQL truy vấn kiểm tra tất cả tên, email, mật khẩu hiện có để đảm bảo tính duy nhất và hợp lệ.

MySQL trả về kết quả truy vấn.

User Model thực hiện kiểm tra xem tên, mật khẩu có hợp lệ và email đã tồn tại hay chưa.

Phân nhánh (Alt):

Trường hợp [Thất bại]: Nếu tên, mật khẩu không hợp lệ hoặc email đã tồn tại:

User Model gửi trạng thái "tài khoản không hợp lệ" cho Auth Controller.

Auth Controller gửi thông báo "tên, pass không hợp lệ hoặc email đã tồn tại" cho Trang Đăng Ký.

Trang Đăng Ký hiển thị "Thông báo tên, pass không hợp lệ hoặc email đã tồn tại" cho Người dùng.

Trường hợp [Thành công]: Nếu thông tin hợp lệ và email chưa tồn tại:

User Model yêu cầu MySQL "Lưu tài khoản".

MySQL xác nhận lưu tài khoản thành công.

User Model gửi trạng thái "lưu thành công" cho Auth Controller.

Auth Controller gửi yêu cầu "Gửi email xác thực tài khoản" đến dịch vụ Email.

Email gửi thông báo xác thực cho Auth Controller (ví dụ: link kích hoạt).

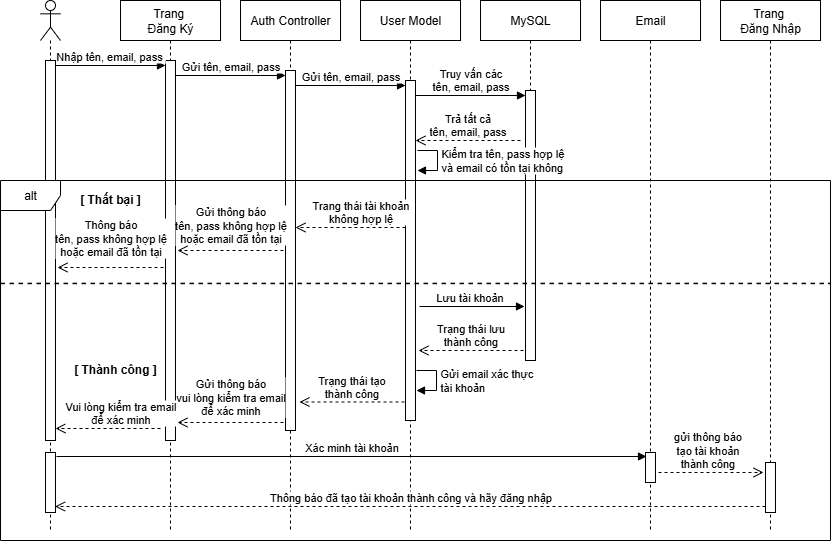
Auth Controller gửi "Vui lòng kiểm tra email để xác minh" cho Trang Đăng Ký.

Trang Đăng Ký hiển thị thông báo "Vui lòng kiểm tra email để xác minh" cho Người dùng.

Người dùng thực hiện "Xác minh tài khoản" (thông qua email).

Khi tài khoản được xác minh, Auth Controller gửi "Thông báo đã tạo tài khoản thành công và hãy đăng nhập" cho Trang Đăng Nhập.

Trang Đăng Nhập hiển thị thông báo và người dùng có thể tiến hành đăng nhập.



Hình 26. sơ đồ tuần tự chức năng đăng ký

* + 1. Quản lý workspace

**Mô tả chi tiết:**

Người dùng (Actor) truy cập ManageWorkspaces.js trong WorkspaceDetailPage.js.

Hệ thống (Frontend) hiển thị danh sách workspace hiện tại.

Người dùng chọn "Tạo workspace" và nhập thông tin (tên, mô tả, quyền riêng tư).

Hệ thống (Frontend) gửi yêu cầu đến workspaceController.js.

workspaceController.js tiếp nhận và chuyển tiếp yêu cầu đến workspaceModel.js.

workspaceModel.js xử lý dữ liệu và lưu vào cơ sở dữ liệu (qua db.js).

db.js trả về kết quả truy vấn cho workspaceModel.js.

**Phân nhánh (Alt):**

Trường hợp [Thành công]: Nếu thông tin hợp lệ:

workspaceModel.js gửi trạng thái "Lưu thành công" cho workspaceController.js.

workspaceController.js gửi thông báo "Tạo workspace thành công" cho Hệ thống (Frontend).

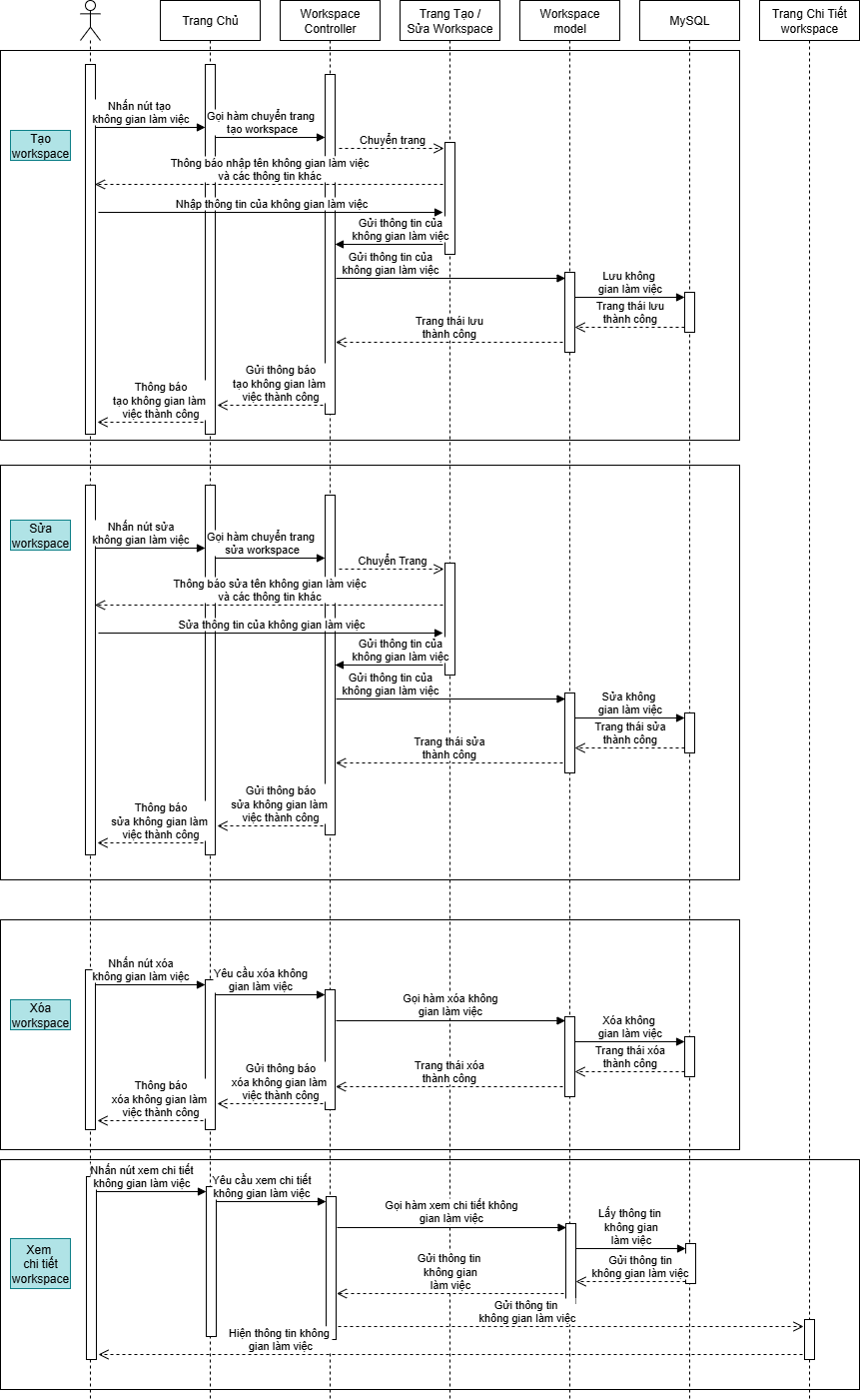
Hệ thống (Frontend) cập nhật danh sách workspace và hiển thị thông báo thành công cho Người dùng.

Trường hợp [Thất bại]: Nếu thông tin không hợp lệ (ví dụ: tên trống, quyền riêng tư không hợp lệ):

workspaceModel.js gửi trạng thái "Thông tin không hợp lệ" cho workspaceController.js.

workspaceController.js gửi thông báo "Tạo workspace thất bại" cho Hệ thống (Frontend).

Hệ thống (Frontend) hiển thị "Thông báo tạo workspace thất bại" cho Người dùng.



Hình 27. sơ đồ tuần tự chức năng quản lý không gian làm việc

* + 1. Quản lý thành viên và công việc trong workspace

**Thành viên**

Mô tả tổng quan: Cho phép người dùng (Owner/Senior) thêm, chỉnh sửa hoặc xóa thành viên trong workspace.

Actors: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, là Owner/Senior, có danh sách từ workspaceMemberController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageMembers.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách thành viên qua MemberList.js.

Người dùng chọn "Thêm thành viên" và nhập thông tin (AddMember.js).

Hệ thống gửi yêu cầu đến workspaceMemberController.js để cập nhật workspaceMemberModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu thành viên đã tồn tại (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Danh sách thành viên được cập nhật.

**Tạo và quản lý công việc**

Mô tả tổng quan: Cho phép người dùng tạo, chỉnh sửa hoặc hoàn thành công việc trong workspace.

Actors: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập taskController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageTasks.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách công việc hiện tại.

Người dùng chọn "Tạo công việc" và điền thông tin (ManageTasksForm.js).

Hệ thống gửi yêu cầu đến taskController.js để lưu vào taskModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Công việc mới được tạo hoặc cập nhật.

**Gán công việc cho thành viên**

Mô tả tổng quan: Cho phép người dùng (Owner/Senior) gán công việc cho các thành viên trong workspace.

Actors: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, có quyền truy cập taskAssignmentController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageTaskAssignment.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách công việc và thành viên.

Người dùng chọn công việc và thành viên để gán (qua form).

Hệ thống gửi yêu cầu đến taskAssignmentController.js để lưu vào taskAssignmentModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu thành viên không thuộc workspace (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Công việc được gán thành công.

**Quản lý ghi chú**

Mô tả tổng quan: Cho phép người dùng tạo và quản lý ghi chú trong workspace.

Actors: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập noteController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageNotes.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách ghi chú hiện tại.

Người dùng chọn "Tạo ghi chú" và nhập nội dung.

Hệ thống gửi yêu cầu đến noteController.js để lưu vào noteModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu nội dung trống (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Ghi chú được tạo hoặc cập nhật.

**Quản lý sự kiện**

Mô tả tổng quan: Cho phép người dùng tạo và quản lý sự kiện trong workspace.

Actors: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập eventController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageEvents.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách sự kiện hiện tại.

Người dùng chọn "Tạo sự kiện" và nhập thông tin (thời gian, tiêu đề).

Hệ thống gửi yêu cầu đến eventController.js để lưu vào eventModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu thời gian xung đột (bước 4), hệ thống hiển thị cảnh báo.

Điều kiện kết thúc: Sự kiện được tạo hoặc cập nhật.

**Quản lý mục tiêu**

Mô tả tổng quan: Cho phép người dùng tạo và theo dõi mục tiêu trong workspace.

Actors: Người dùng (User, bất kỳ vai trò), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, thuộc workspace, có quyền truy cập goalController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageGoals.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách mục tiêu hiện tại.

Người dùng chọn "Tạo mục tiêu" và nhập thông tin (tiêu đề, tiến độ).

Hệ thống gửi yêu cầu đến goalController.js để lưu vào goalModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu tiến độ không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Mục tiêu được tạo hoặc cập nhật.

**Quản lý giao dịch**

Mô tả tổng quan: Cho phép người dùng (Owner/Senior) ghi nhận và quản lý giao dịch tài chính trong workspace.

Actors: Người dùng (User, vai trò Owner/Senior), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, có quyền truy cập transactionController.js.

Luồng sự kiện chính:

Người dùng truy cập ManageTransactions.js trong WorkspaceDetailPage.js.

Hệ thống hiển thị danh sách giao dịch hiện tại.

Người dùng chọn "Thêm giao dịch" và nhập thông tin (loại, số tiền).

Hệ thống gửi yêu cầu đến transactionController.js để lưu vào transactionModel.js.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu số tiền không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Giao dịch được ghi nhận và hiển thị.



Hình 28. sơ đồ tuần tự chức năng quản lý thành viên và công việc

* + 1. Xem và chỉnh sửa hồ sơ cá nhân

Mô tả tổng quan: Cho phép người dùng xem và cập nhật thông tin cá nhân, bao gồm đổi mật khẩu.

Actors: Người dùng (User), Hệ thống (System).

Điều kiện tiên quyết: Người dùng đã đăng nhập, có quyền truy cập ProfilePage.js hoặc ResetPasswordPage.js.

Luồng sự kiện chính:

Người dùng truy cập ProfilePage.js hoặc ResetPasswordPage.js.

Hệ thống hiển thị thông tin cá nhân (tên, email) hoặc form đổi mật khẩu.

Người dùng chỉnh sửa thông tin (nếu cần) hoặc nhập mật khẩu cũ/mật khẩu mới và gửi yêu cầu.

Hệ thống gửi yêu cầu đến authController.js để cập nhật userModel.js (bao gồm xác thực mật khẩu cũ nếu đổi mật khẩu).

Hệ thống thông báo thành công và cập nhật thông tin (gửi email xác nhận nếu đổi mật khẩu).

Luồng sự kiện thay thế: Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

Điều kiện kết thúc: Hồ sơ cá nhân được cập nhật hoặc mật khẩu được đổi.

* + 1. Quản lý người dùng (Chức năng Admin)

Mô tả tổng quan: Cho phép admin xem, thêm, chỉnh sửa hoặc xóa người dùng trong hệ thống.

Actors: Admin (User, vai trò Admin), Hệ thống (System).

Điều kiện tiên quyết: Admin đã đăng nhập, có quyền truy cập qua adminMiddleware.js và AdminUsersPage.js.

Luồng sự kiện chính:

Admin truy cập AdminUsersPage.js.

Hệ thống hiển thị danh sách người dùng từ userModel.js.

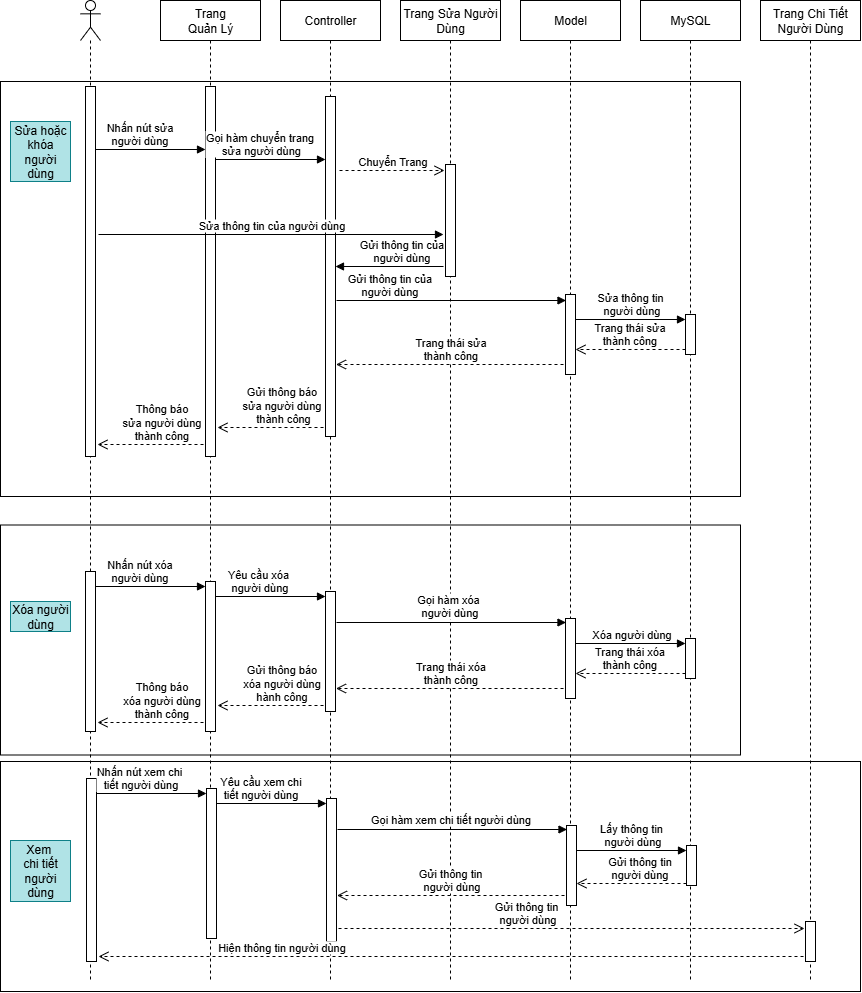
Admin chọn "Thêm người dùng" và nhập thông tin (tên, email, vai trò).

Hệ thống gửi yêu cầu đến authController.js để lưu hoặc cập nhật.

Hệ thống thông báo thành công và cập nhật danh sách.

Luồng sự kiện thay thế: Nếu thông tin không hợp lệ (bước 4), hệ thống hiển thị thông báo lỗi.

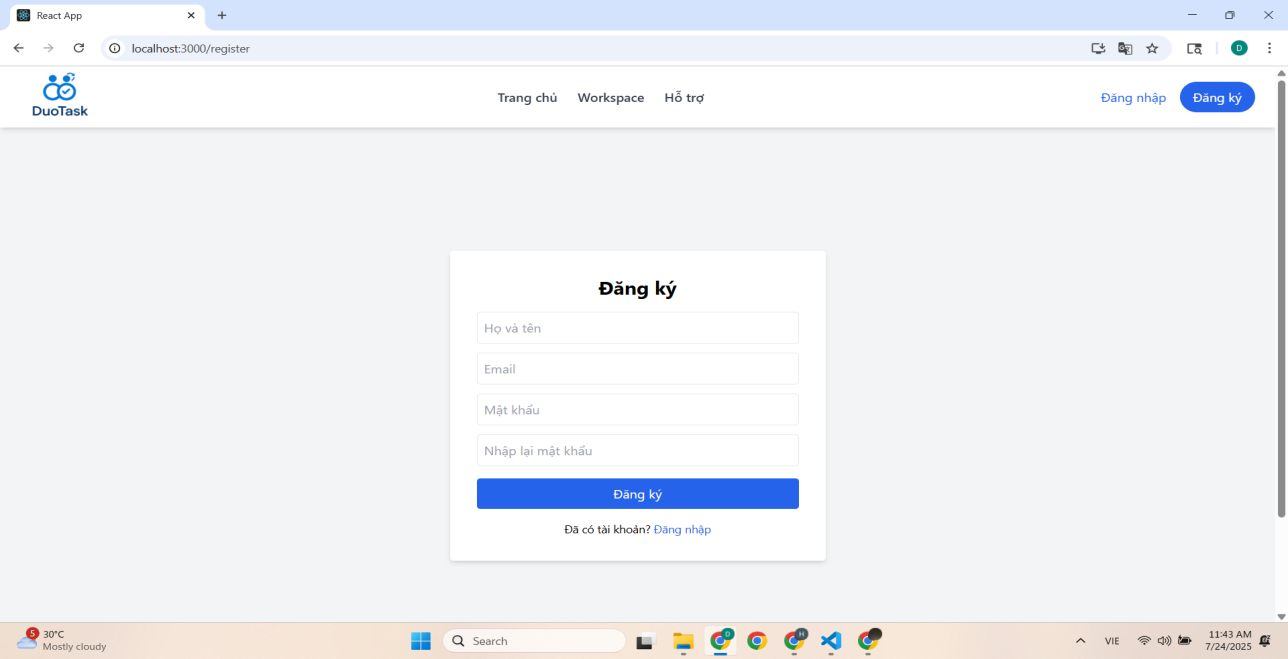
Điều kiện kết thúc: Danh sách người dùng được cập nhật.



Hình 29. sơ đồ tuần tự chức năng quản lý người dùng

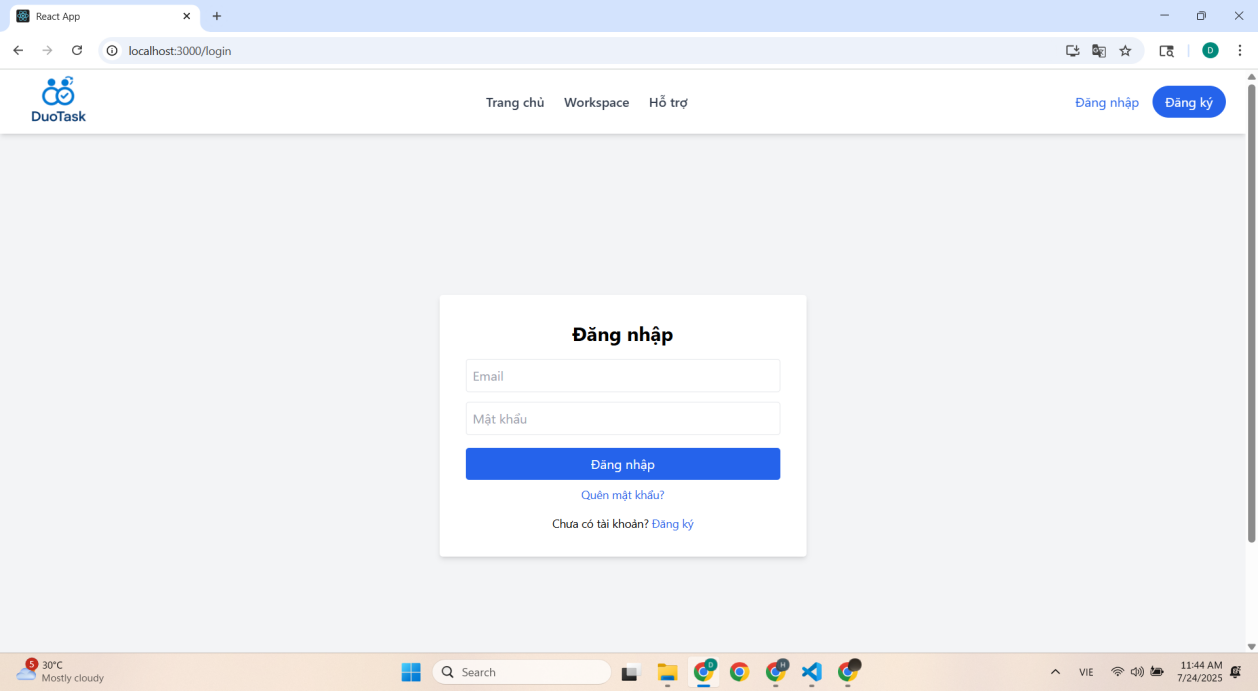
### Các màn hình giao diện người dùng

* + 1. Màn hình đăng ký tài khoản: Hiển thị form đăng ký với các trường bắt buộc (tên, email, mật khẩu) để người dùng tạo tài khoản mới, tích hợp với VerifyEmailPage.js cho xác nhận.

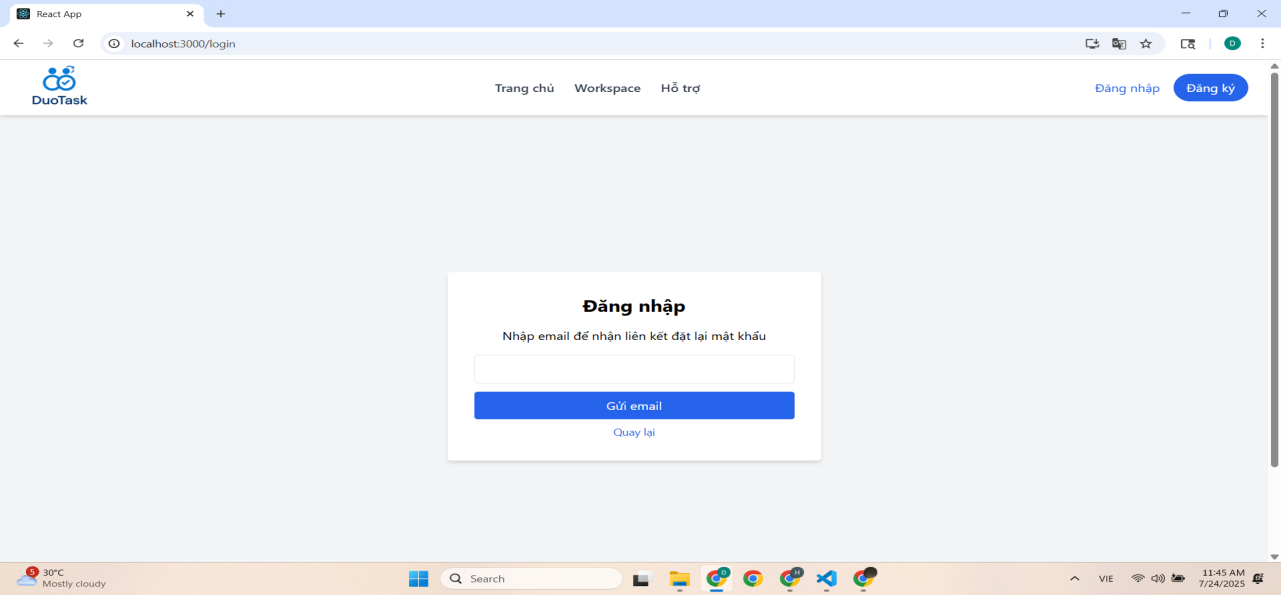


Hình 30. Màn hình đăng ký tài khoản

* + 1. Màn hình đăng nhập: Hiển thị form nhập email và mật khẩu, cho phép người dùng truy cập hệ thống sau khi xác thực thành công, liên kết với LoginPage.js.



Hình 31. Màn hình đăng nhập

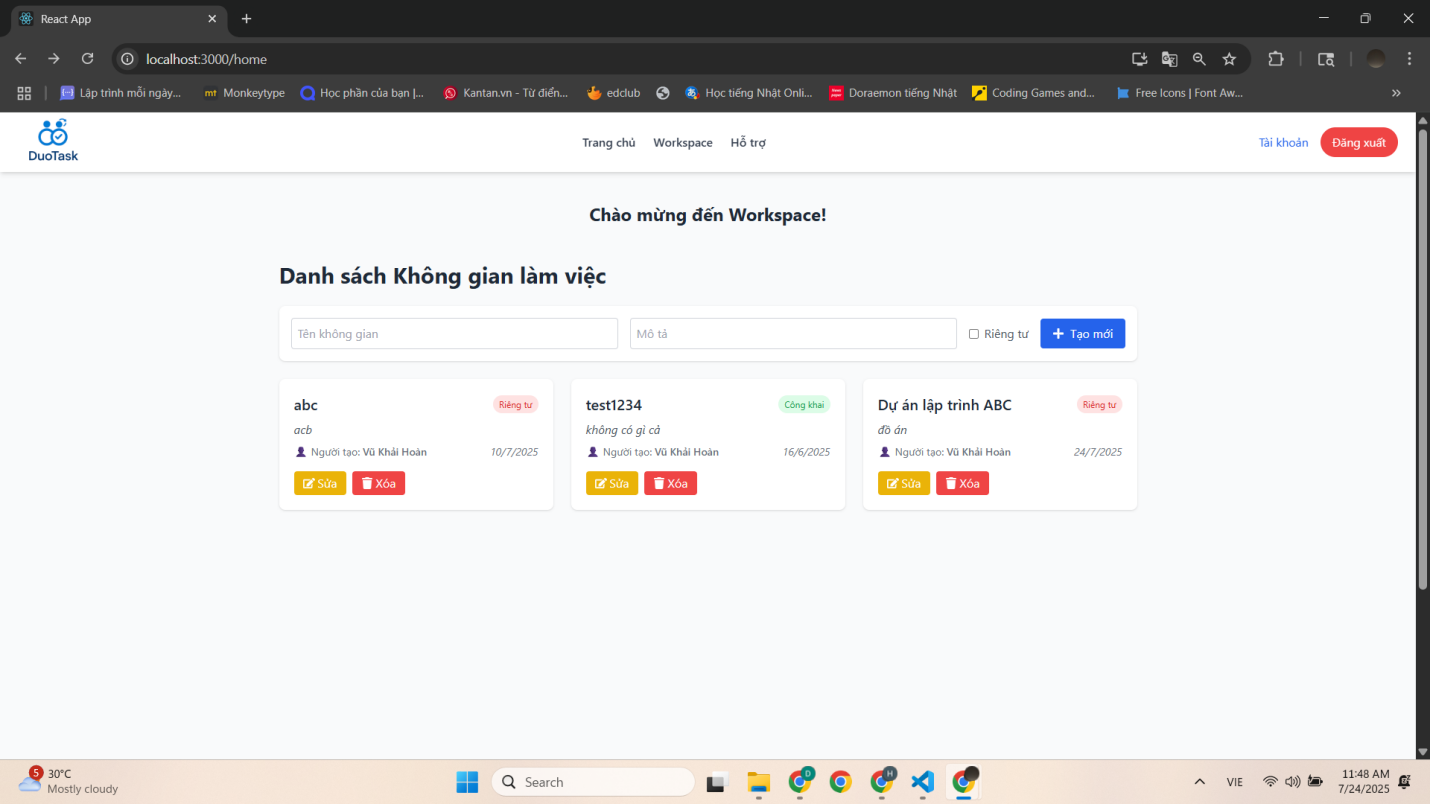


Hình 32. Màn hình gửi mail quên mật khẩu

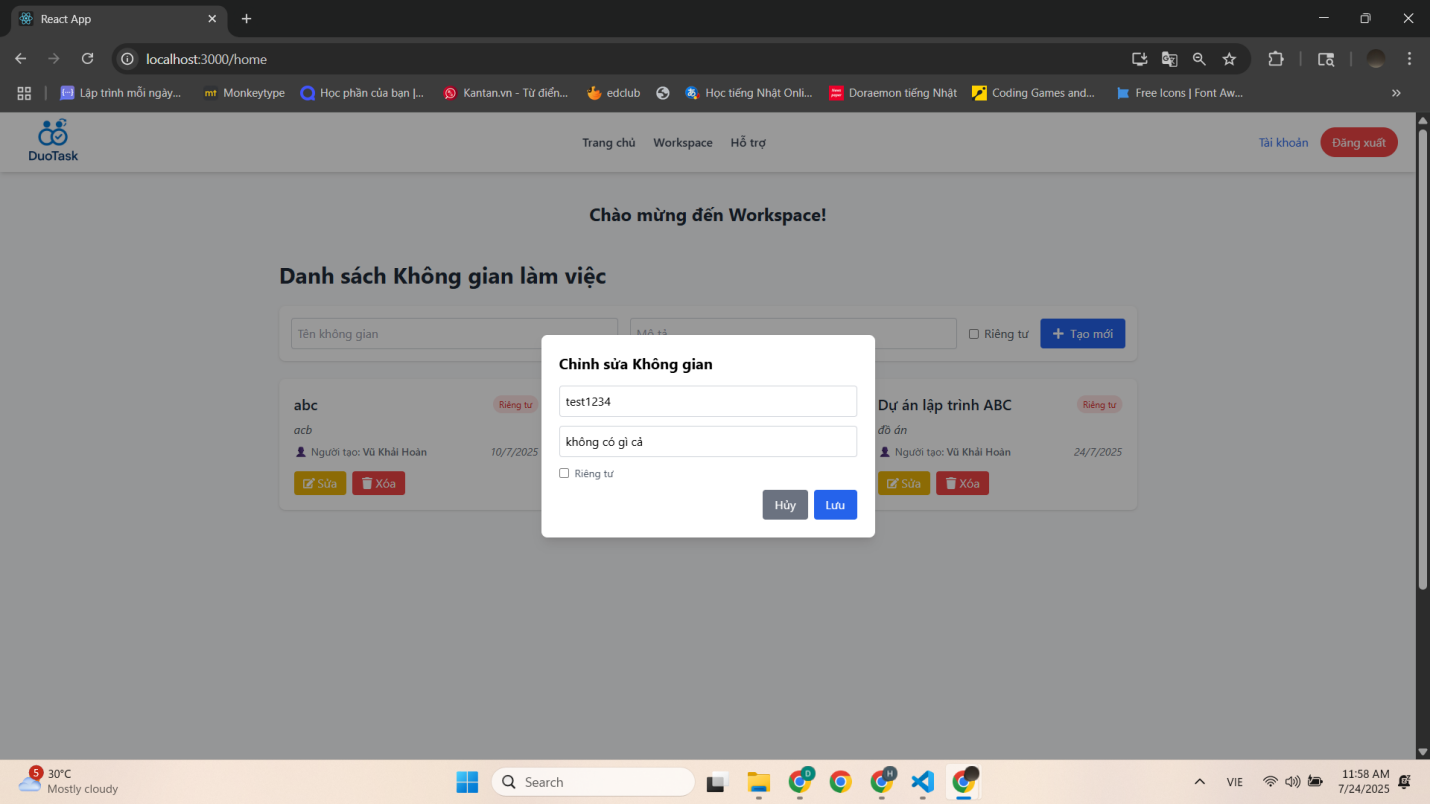


Hình 33. Màn hình đặt lại mật khẩu

* + 1. Màn hình trang chủ: Hiển thị danh sách workspace của người dùng sau khi đăng nhập, với các tùy chọn điều hướng đến các chức năng chính (WorkspaceDetailPage.js).

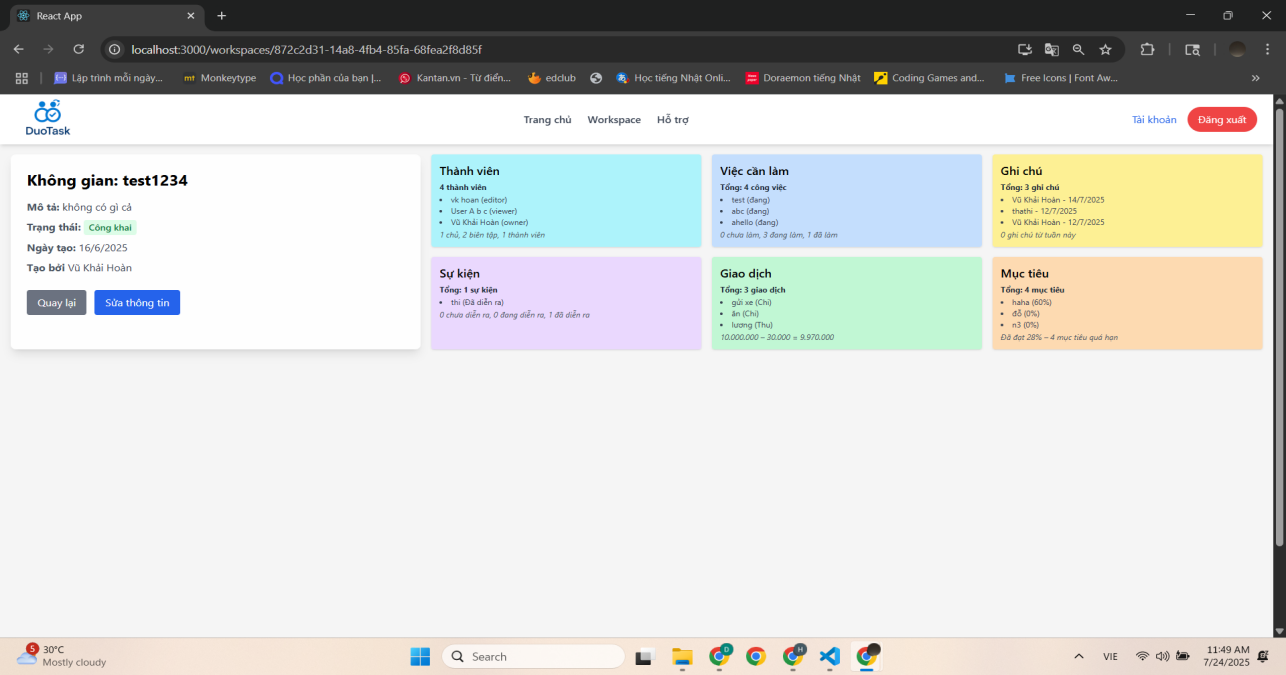


Hình 34. Màn hình trang chủ

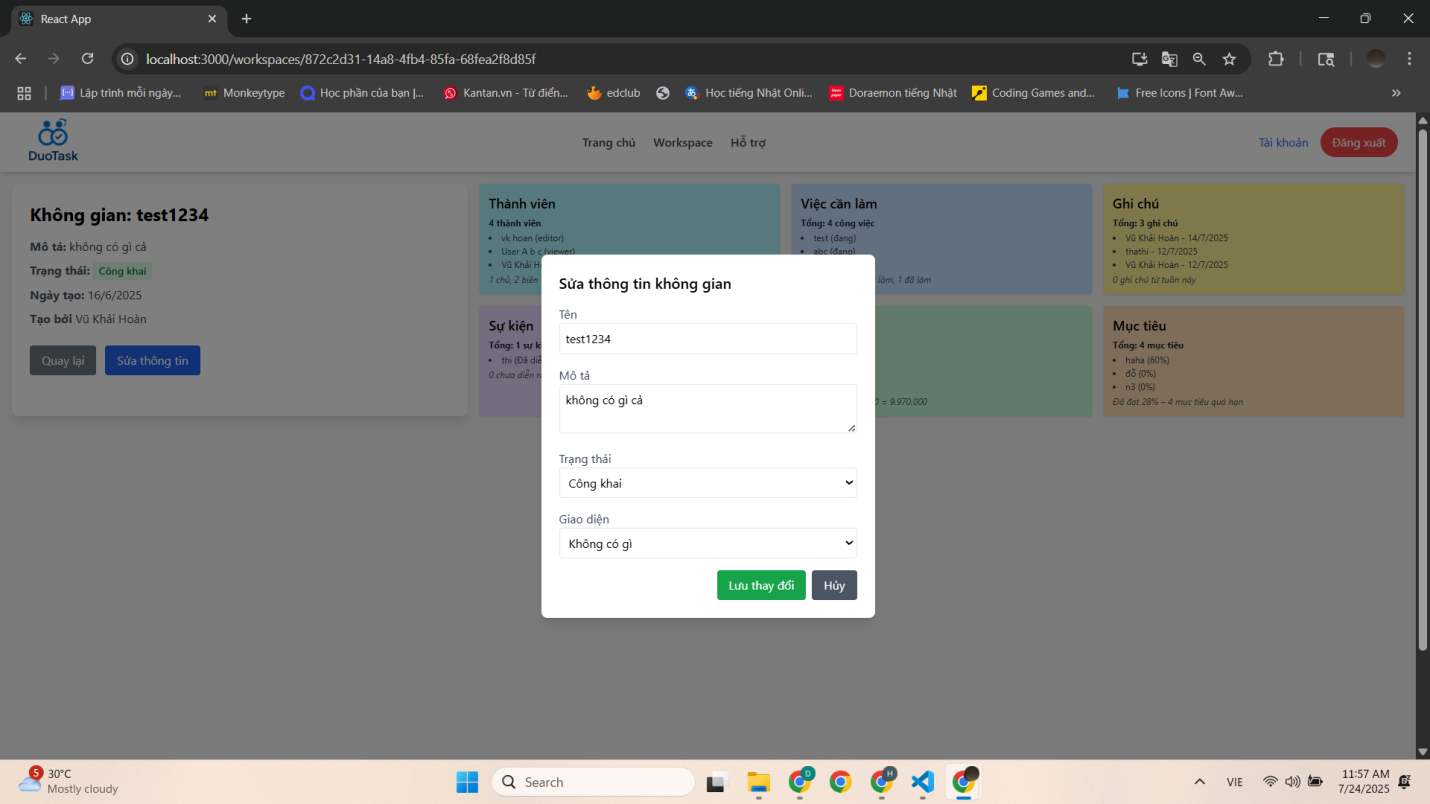


Hình 35. Màn hình sửa thông tin workspace cơ bản

* + 1. Màn hình chi tiết workspace: Hiển thị thông tin chi tiết về workspace (tên, thành viên, trạng thái public/private), với các tab quản lý công việc, ghi chú, sự kiện, v.v., liên kết với WorkspaceDetailPage.js.

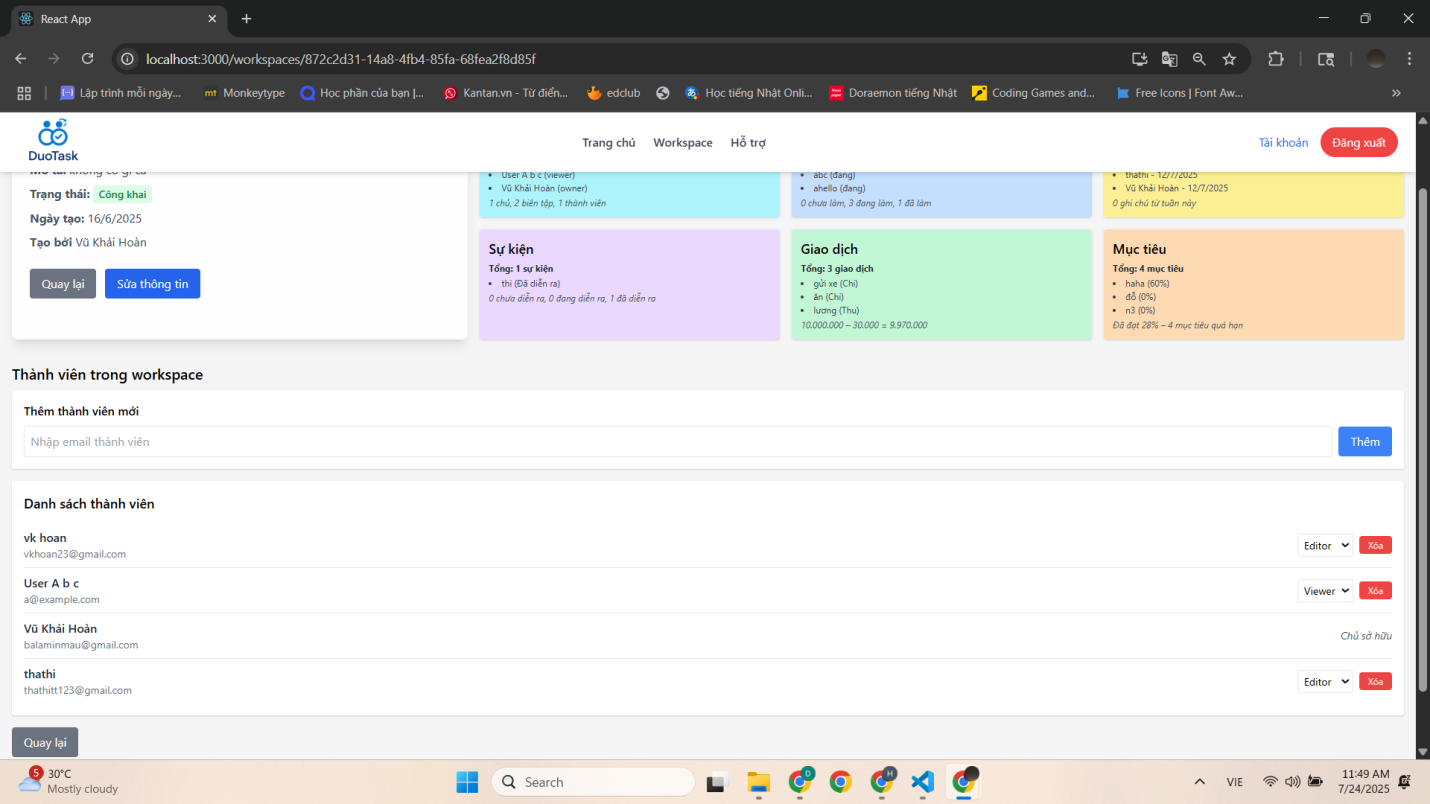


Hình 36. Màn hình chi tiết workspace



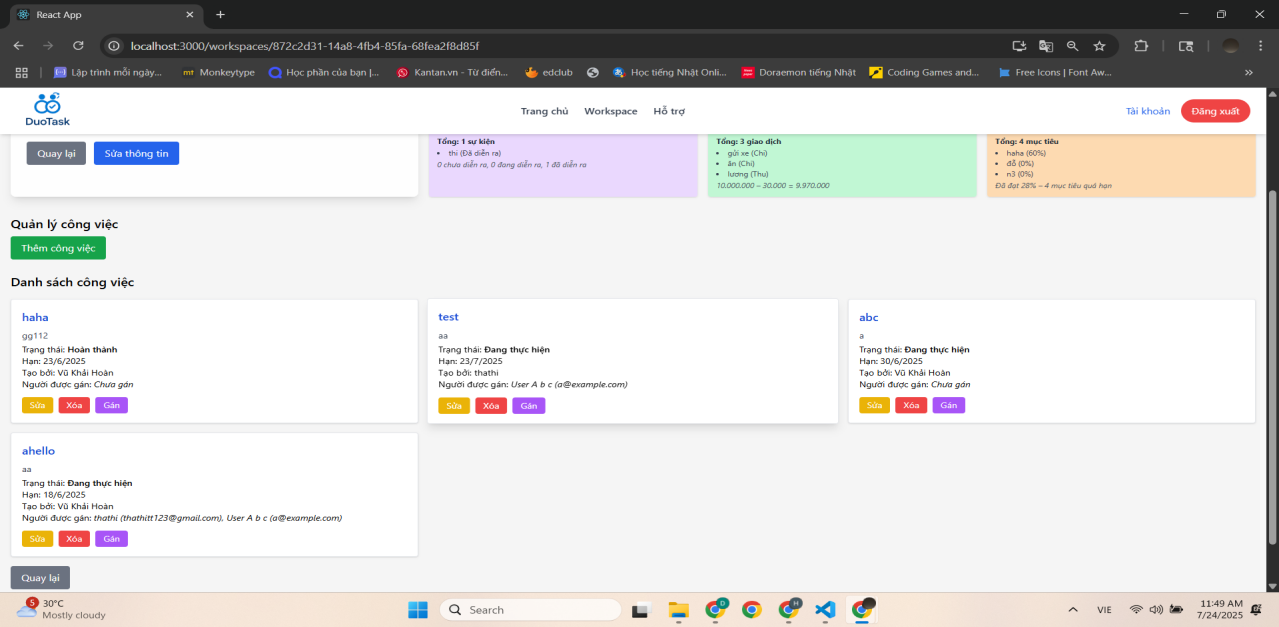
Hình 37. Màn hình sửa thông chi tiết workspace

* + 1. Màn hình quản lý thành viên: Hiển thị danh sách thành viên trong workspace, với các nút thêm, chỉnh sửa hoặc xóa, liên kết với ManageMembers.js.

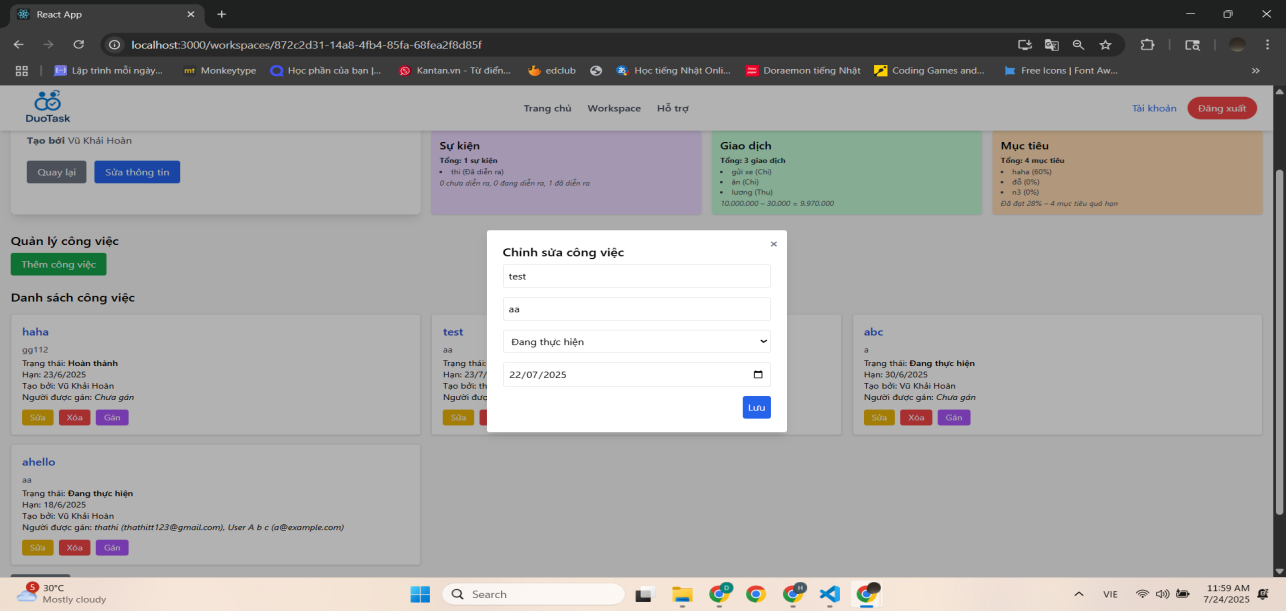


Hình 38. Màn hình quản lý thành viên

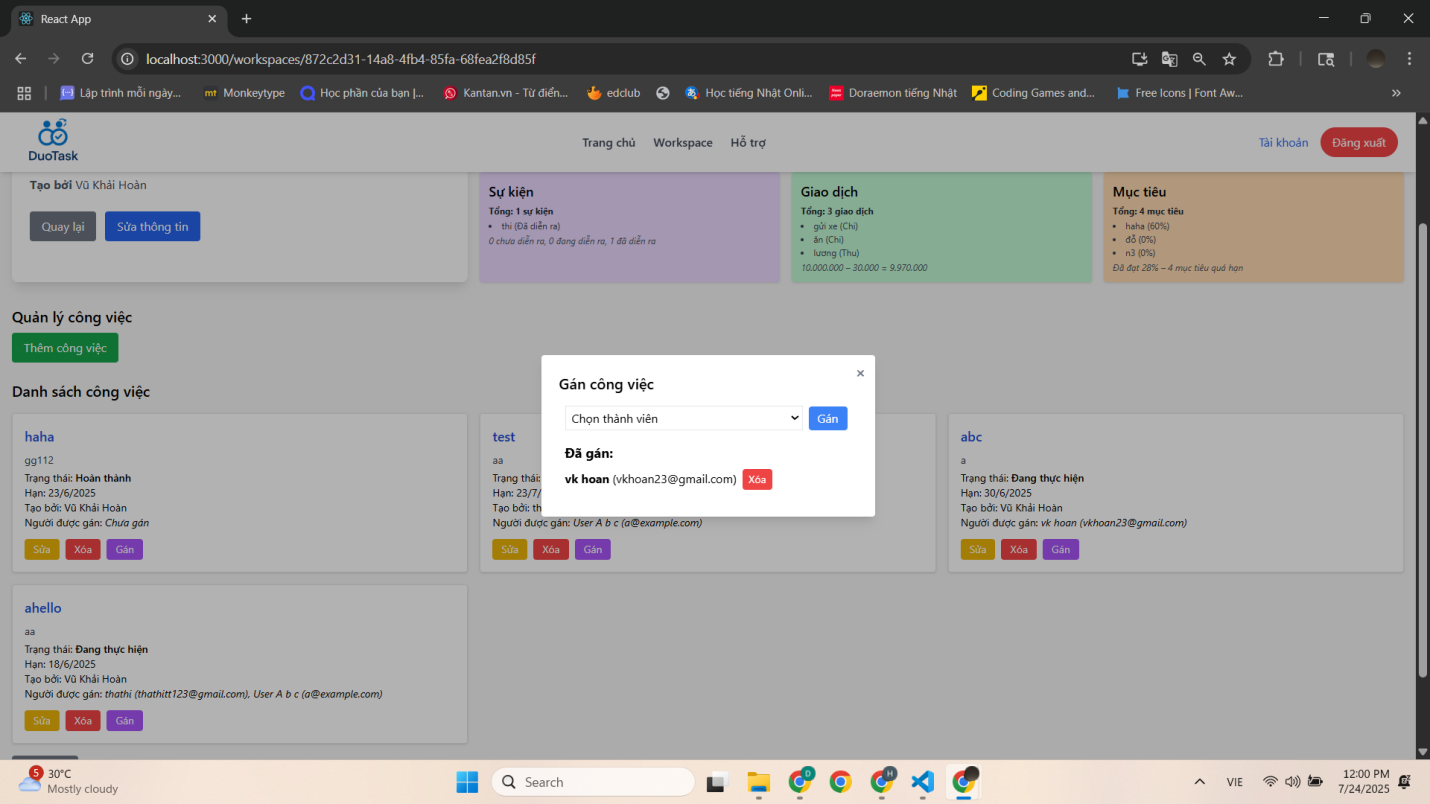
* + 1. Màn hình quản lý công việc: Hiển thị danh sách công việc, với form tạo mới và tùy chọn gán công việc, liên kết với ManageTasks.js và ManageTaskAssignment.js.



Hình 39. Màn hình quản lý công việc

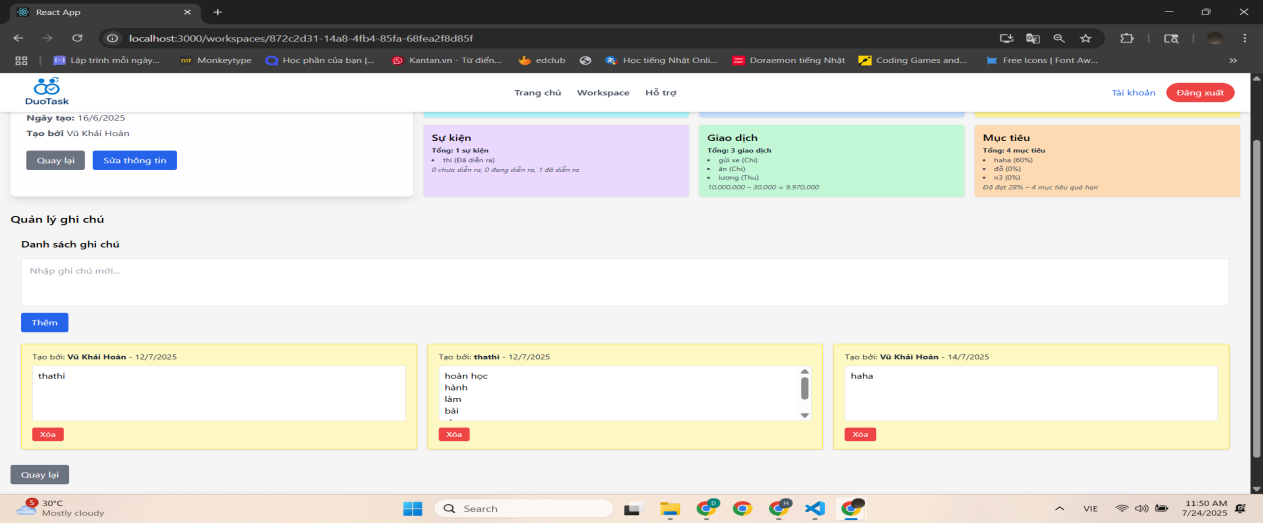


Hình 40. Màn hình chỉnh sửa công việc



Hình 41. Màn hình gán công việc cho người dùng

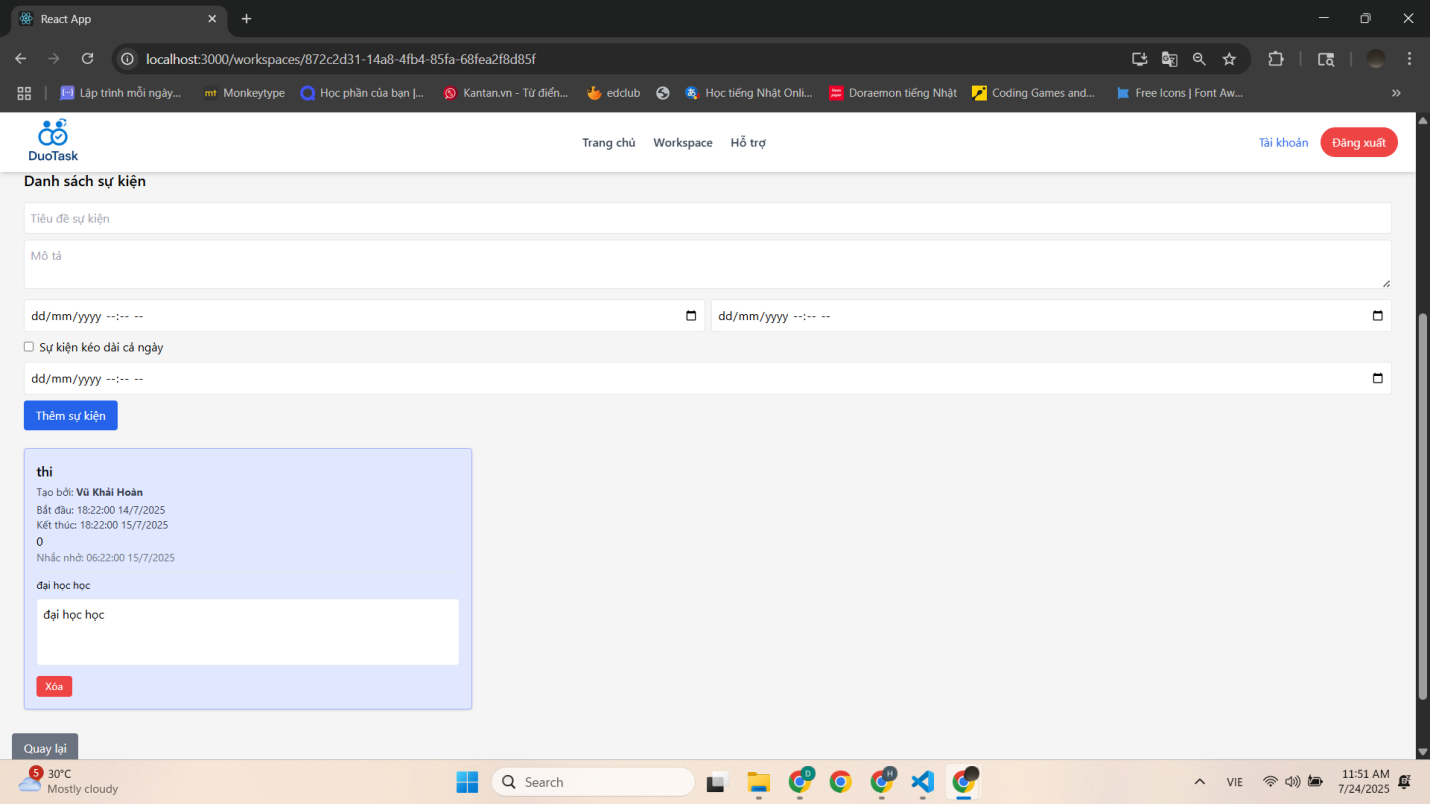
* + 1. Màn hình quản lý ghi chú: Hiển thị danh sách ghi chú và form tạo mới, liên kết với ManageNotes.js.



Hình 42. Màn hình quản lý ghi chú

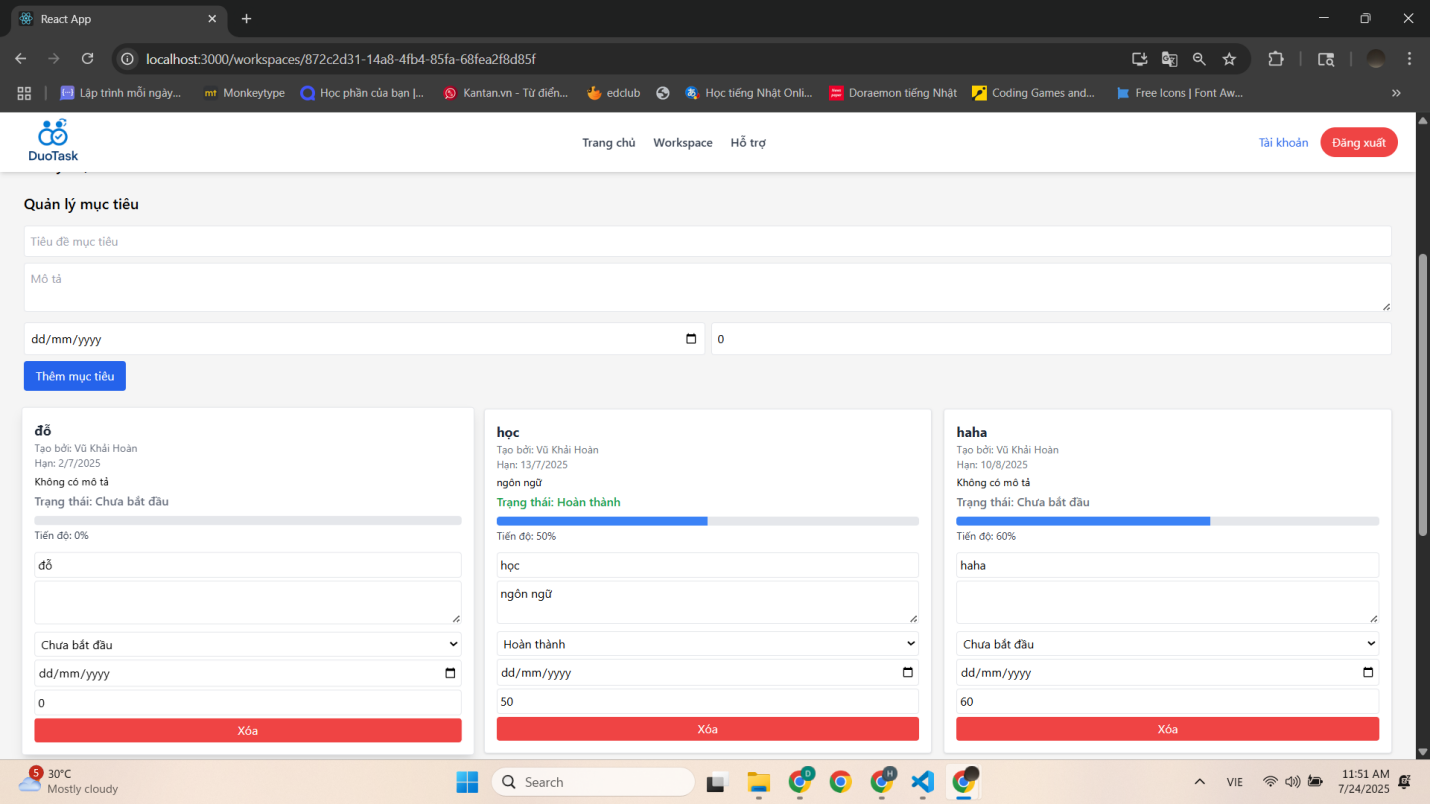
* + 1. Màn hình quản lý sự kiện: Hiển thị lịch và form tạo sự kiện, với kiểm tra xung đột thời gian, liên kết với ManageEvents.js.

(Hình ảnh): Ảnh màn hình hoặc mockup cho "Màn hình quản lý sự kiện".



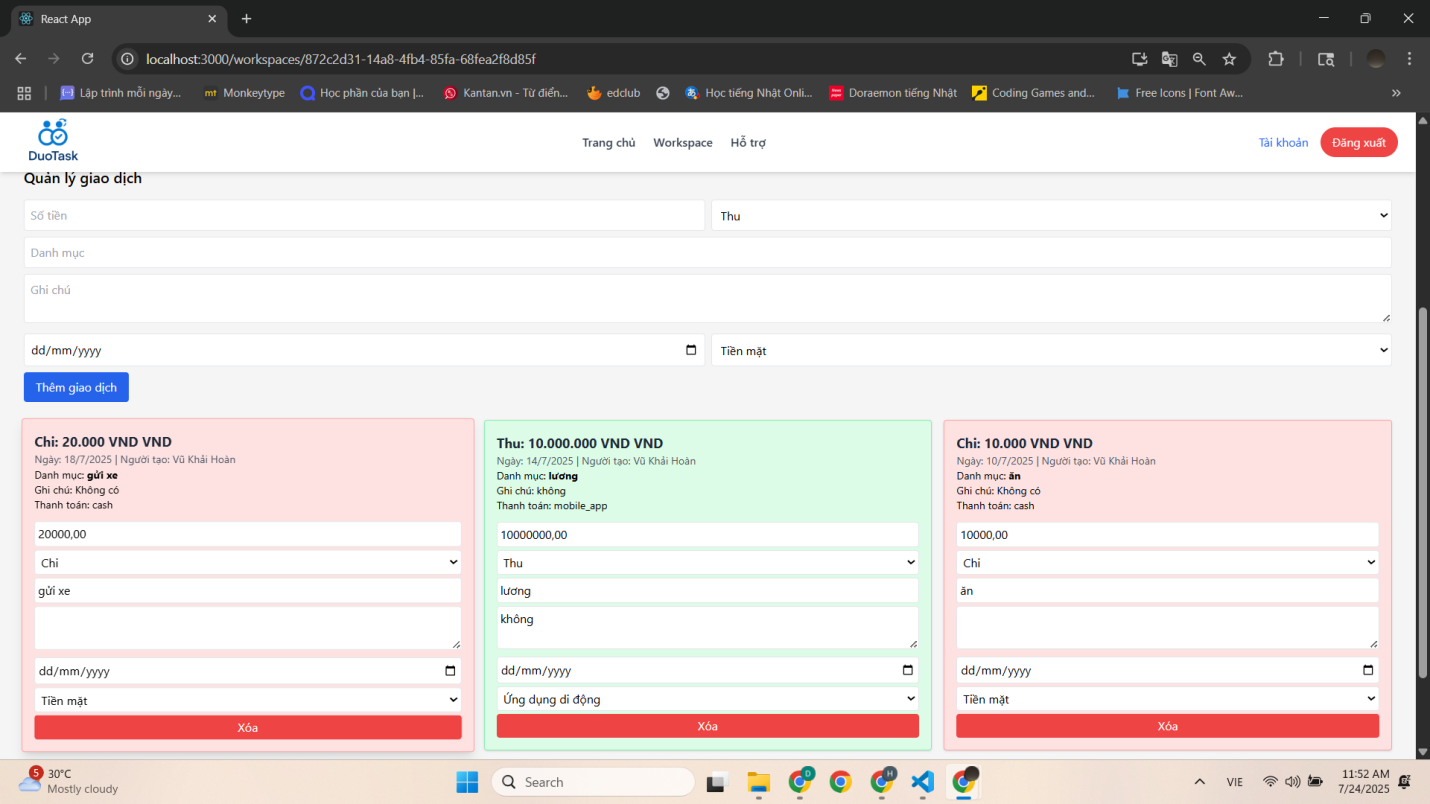
Hình 43. Màn hình quản lý sự kiện

* + 1. Màn hình quản lý mục tiêu: Hiển thị danh sách mục tiêu và form tạo mới, với theo dõi tiến độ, liên kết với ManageGoals.js.



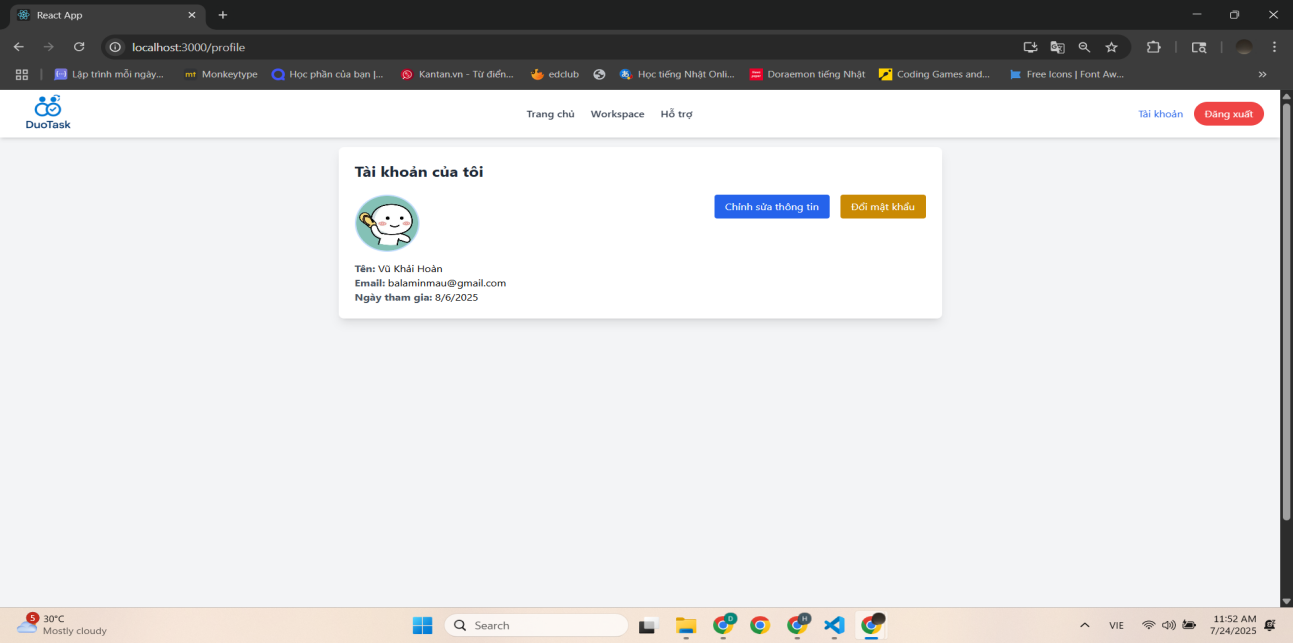
Hình 44. Màn hình quản lý mục tiêu

* + 1. Màn hình quản lý giao dịch: Hiển thị danh sách giao dịch và form ghi nhận, dành cho Owner/Senior, liên kết với ManageTransactions.js.

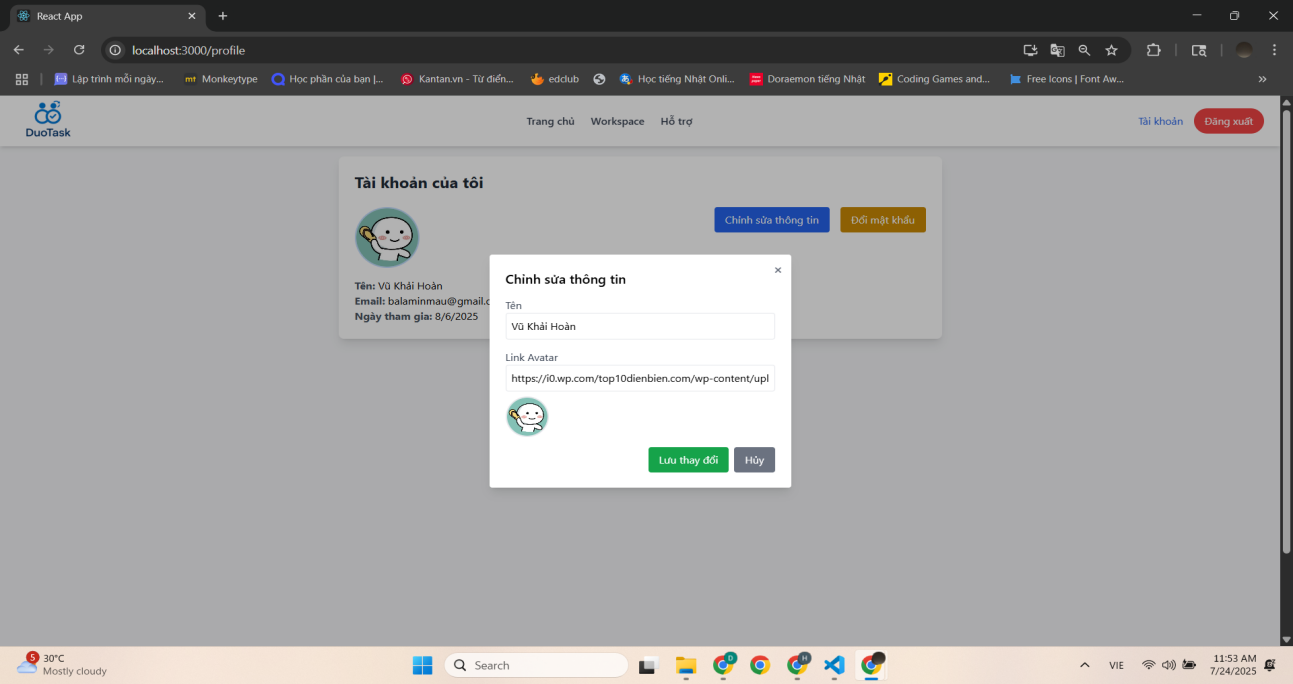


Hình 45. Màn hình quản lý giao dịch

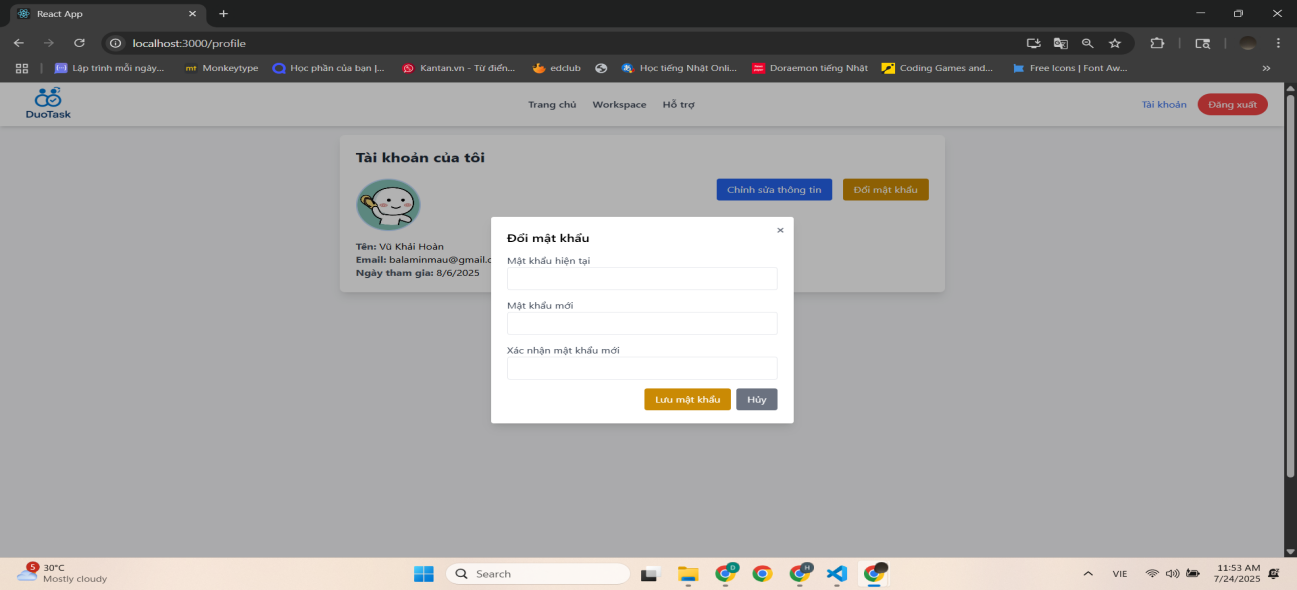
* + 1. Màn hình hồ sơ cá nhân: Hiển thị thông tin cá nhân (tên, email) và form đổi mật khẩu, liên kết với ProfilePage.js và ResetPasswordPage.js.



Hình 46. Màn hình hồ sơ cá nhân

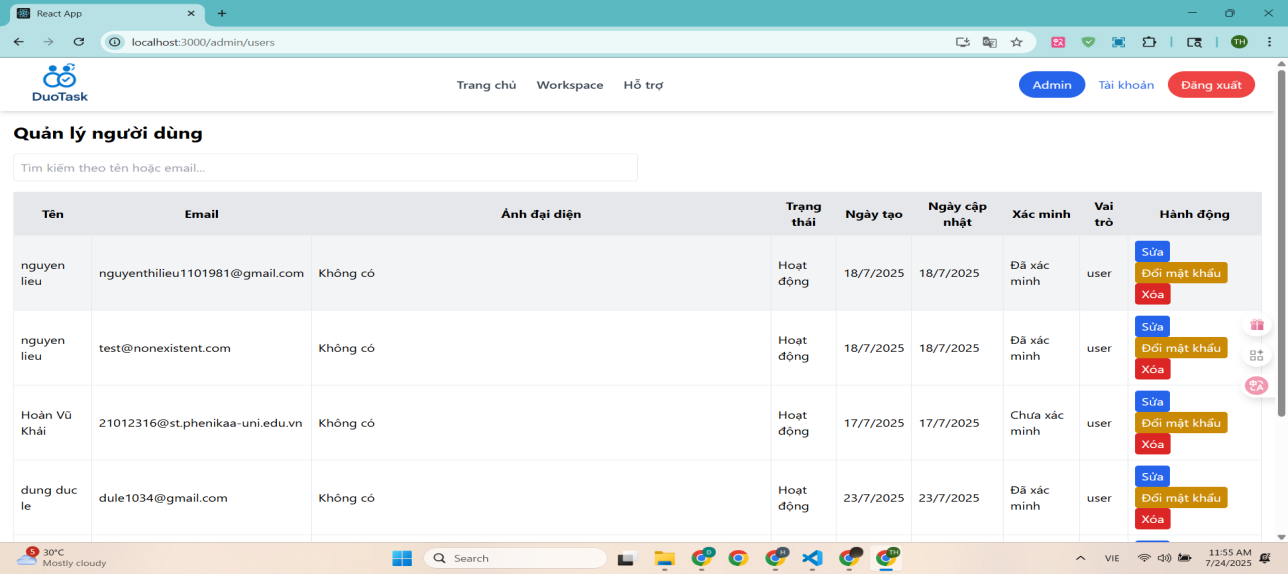


Hình 47. Màn hình chỉnh sửa hồ sơ cá nhân

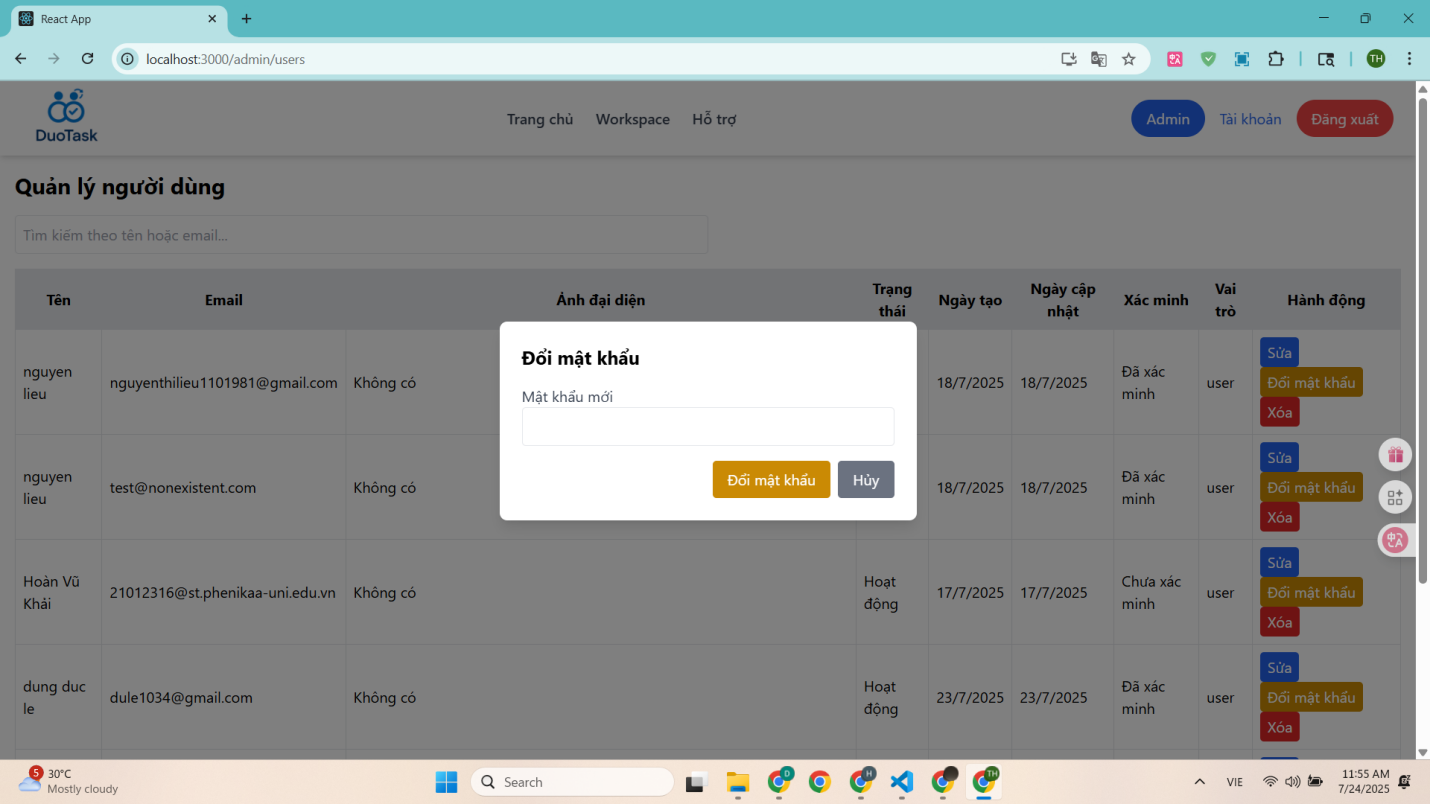


Hình 48. Màn hình đổi mật khẩu

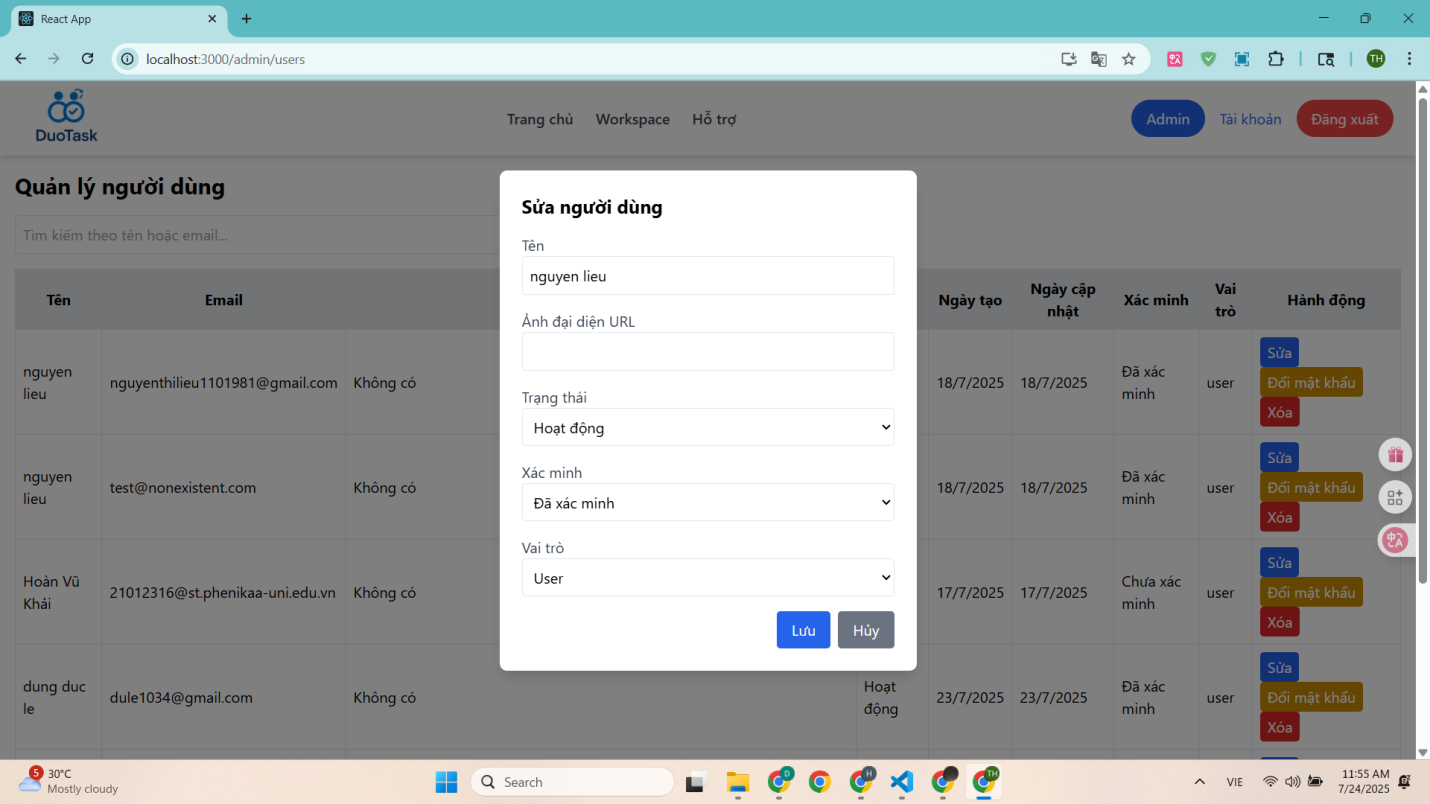
* + 1. Màn hình quản trị người dùng: Hiển thị danh sách người dùng và form quản lý (thêm, chỉnh sửa, xóa), dành cho admin, liên kết với AdminUsersPage.js.



Hình 49. Màn hình quản trị người dùng



Hình 50. Màn hình đổi mật khẩu người dùng



Hình 51. Màn hình sửa thông tin người dùng

* 1. Công nghệ sử dụng
     1. 4.6.2 Ngôn ngữ lập trình và framework

JavaScript: Là ngôn ngữ chính, được dùng để viết cả phần backend (phía máy chủ) và frontend (phía người dùng), giúp hệ thống hoạt động nhanh và đồng bộ.

Node.js: Một môi trường chạy JavaScript ở backend, cho phép xử lý các tác vụ như quản lý người dùng, workspace, và công việc một cách hiệu quả.

React.js: Một công cụ frontend giúp tạo giao diện đẹp, dễ tương tác, như các màn hình quản lý công việc hoặc hồ sơ cá nhân.

* + 1. 4.6.3 Cơ sở dữ liệu

MySQL: Một cơ sở dữ liệu quan hệ, được sử dụng để lưu trữ thông tin như danh sách người dùng, workspace, và giao dịch một cách có cấu trúc, phù hợp với việc quản lý nhóm nhỏ và đảm bảo tính nhất quán dữ liệu.

* + 1. 4.6.4 Mô hình MVC (Model-View-Controller)

Hệ thống được thiết kế theo mô hình MVC, giúp tổ chức mã nguồn rõ ràng và dễ bảo trì:

Model: Xử lý dữ liệu, như lưu thông tin người dùng, công việc, hoặc lịch trình, đảm bảo dữ liệu được cập nhật chính xác.

View: Hiển thị giao diện cho người dùng, như các trang quản lý công việc hoặc sự kiện, giúp người dùng dễ dàng thao tác.

Controller: Kết nối giữa Model và View, xử lý yêu cầu (như tạo workspace hoặc gán công việc) và gửi kết quả về giao diện.

Mô hình này giúp hệ thống phân chia công việc rõ ràng, dễ phát triển thêm tính năng trong tương lai.

* + 1. 4.6.5 Công cụ hỗ trợ phát triển

Git: Giúp nhóm làm việc cùng nhau bằng cách quản lý các phiên bản mã nguồn, tránh xung đột khi nhiều người chỉnh sửa.

Visual Studio Code: Một phần mềm lập trình thân thiện, hỗ trợ viết code cho cả React và Node.js.

Postman: Dùng để kiểm tra các chức năng của hệ thống, như xem liệu đăng nhập hoặc tạo công việc có hoạt động đúng không.

MySQL Workbench: Công cụ quản lý và thiết kế cơ sở dữ liệu MySQL, giúp tạo bảng, truy vấn, và tối ưu hóa dữ liệu.

* + 1. 4.6.6 Công nghệ bảo mật và kết nối

JSON Web Token (JWT): Được dùng để bảo vệ hệ thống, đảm bảo chỉ người dùng đã đăng nhập mới truy cập được các tính năng như quản lý thành viên hoặc tài chính.

Express.js: Một framework giúp xây dựng các giao tiếp giữa frontend và backend nhanh chóng, an toàn.

Axios: Giúp frontend gửi yêu cầu đến backend một cách mượt mà, ví dụ khi lưu công việc mới.

* + 1. 4.6.7 Công cụ thiết kế giao diện

Tailwind CSS: Một framework CSS giúp thiết kế giao diện nhanh chóng, linh hoạt, và đẹp mắt, được sử dụng để tạo các thành phần như form đăng ký, danh sách công việc, hoặc lịch sự kiện, đảm bảo giao diện thống nhất và dễ tùy chỉnh theo nhu cầu người dùng.

* + 1. 4.6.8 Kết luận về công nghệ sử dụng

Sự kết hợp của JavaScript, Node.js, React.js, MySQL, và mô hình MVC cùng các công cụ hỗ trợ như Git, Postman, MySQL Workbench, và Tailwind CSS tạo nên một hệ thống quản lý công việc toàn diện. Nó đáp ứng nhu cầu quản lý cá nhân và nhóm nhỏ, với giao diện thống nhất và khả năng mở rộng, phù hợp với các thiết kế đã mô tả ở các phần trước.

* 1. Cấu trúc dự án

Frontend: Chứa giao diện người dùng với các trang (như WorkspaceDetailPage.js) và thành phần (như TaskList.js), được xây dựng bằng React.js và Tailwind CSS.

Backend: Chứa logic xử lý với các phần như quản lý người dùng, workspace, và công việc, được phát triển bằng Node.js và Express.js.

Database: Sử dụng MySQL để lưu trữ dữ liệu, được quản lý qua MySQL Workbench.

task-management/

├── backend/

│ ├── config/

│ │ ├── db.js

│ ├── controllers/

│ │ ├── authController.js

│ │ ├── eventController.js

│ │ ├── goalController.js

│ │ ├── noteController.js

│ │ ├── taskAssignmentController.js

│ │ ├── taskController.js

│ │ ├── transactionController.js

│ │ ├── workspaceController.js

│ │ └── workspaceMemberController.js

│ ├── middleware/

│ │ ├── adminMiddleware.js

│ │ ├── authMiddleware.js

│ ├── models/

│ │ ├── eventModel.js

│ │ ├── goalModel.js

│ │ ├── noteModel.js

│ │ ├── taskAssignmentModel.js

│ │ ├── taskModel.js

│ │ ├── transactionModel.js

│ │ ├── userModel.js

│ │ ├── workspaceMemberModel.js

│ │ └── workspaceModel.js

│ ├── node\_modules/

│ ├── routes/

│ │ ├── authRoutes.js

│ ├── utils/

│ ├── .env

│ ├── package-lock.json

│ ├── package.json

│ └── server.js

└── frontend/

├── node\_modules/

├── public/

├── src/

│ ├── components/

│ │ ├── AddMember.js

│ │ ├── DashboardWidgets.js

│ │ ├── Footer.js

│ │ ├── Header.js

│ │ ├── LoadingSpinner.js

│ │ ├── ManageEvents.js

│ │ ├── ManageGoals.js

│ │ ├── ManageMembers.js

│ │ ├── ManageNotes.js

│ │ ├── ManageTaskAssignment.js

│ │ ├── ManageTasks.js

│ │ ├── ManageTasksForm.js

│ │ ├── ManageTransactions.js

│ │ ├── MemberList.js

│ │ ├── Modal.js

│ │ ├── PrivateRoute.js

│ │ └── WorkspaceList.js

│ ├── img/

│ ├── pages/

│ │ ├── AdminUsersPage.js

│ │ ├── Home.js

│ │ ├── HomePage.js

│ │ ├── LoginPage.js

│ │ ├── ProfilePage.js

│ │ ├── RegisterPage.js

│ │ ├── ResetPasswordPage.js

│ │ ├── VerifyEmailPage.js

│ │ └── WorkspaceDetailPage.js

│ ├── App.css

│ ├── App.js

│ ├── App.test.js

│ ├── index.css

│ ├── index.js

│ ├── logo.svg

│ ├── reportWebVitals.js

│ └── setupTests.js

├── .gitignore

├── package-lock.json

├── package.json

├── postcss.config.js

├── README.md

└── tailwind.config.js

# Một số thành phần khác

## Kế hoạch dự án

Tuần 1: Lập kế hoạch tổng thể + Phân tích yêu cầu + Thiết kế chức năng

Mục tiêu chính: Xác định phạm vi dự án, thu thập và phân tích yêu cầu từ người dùng, thiết kế sơ bộ các chức năng chính (đăng ký, quản lý workspace, công việc).

Kết quả đạt được sau tuần đó:

Hoàn thành sơ đồ chức năng, minh họa các luồng chính như đăng ký tài khoản, quản lý workspace.

Xây dựng phân tích Use-case, bao gồm các kịch bản như tạo workspace, gán công việc.

Thiết kế mô hình phân quyền (admin, owner, editor, viewer) để đảm bảo tính bảo mật và linh hoạt.

Lập danh sách tính năng dự kiến, bao gồm quản lý thành viên, ghi chú, và giao dịch.

Tuần 2: Thiết kế cơ sở dữ liệu + Dựng backend cơ bản (API đăng ký, đăng nhập) + Dựng giao diện cơ bản + Đăng nhập/Đăng ký + Layout tổng thể

Mục tiêu chính: Xây dựng nền tảng dữ liệu và các thành phần cơ bản của hệ thống, đảm bảo chức năng đăng nhập/đăng ký hoạt động.

Kết quả đạt được sau tuần đó:

Thiết kế và chuẩn hóa cơ sở dữ liệu MySQL với các bảng chính (user, workspace, task, transaction, v.v.), đảm bảo quan hệ giữa các thực thể.

Phát triển backend bằng Node.js và Express.js, tạo API đăng ký và đăng nhập, kiểm tra thành công qua Postman.

Xây dựng giao diện ban đầu bằng React.js và Tailwind CSS, hoàn thiện các trang auth (đăng nhập/đăng ký), menu, và header.

Đảm bảo chức năng đăng nhập/đăng ký hoạt động, tích hợp JWT để bảo mật phiên làm việc.

Tuần 3: Thiết kế & làm chức năng quản lý Workspace + chia sẻ, phân quyền

Mục tiêu chính: Hoàn thiện tính năng quản lý workspace, bao gồm tạo, chỉnh sửa, xóa, và phân quyền cho thành viên.

Kết quả đạt được sau tuần đó:

Tích hợp tính năng tạo, sửa, xóa workspace với tùy chọn public/private, quản lý qua backend.

Thêm chức năng mời người dùng vào workspace và phân quyền (owner, editor, viewer) dựa trên logic phân quyền trong backend.

Đảm bảo giao diện hiển thị thông tin workspace (tên, mô tả, trạng thái) và hỗ trợ thay đổi giao diện (màu chữ, nền).

Tuần 4: Xây dựng chức năng Ghi chú, Lịch, Mục tiêu cá nhân + Chức năng Task + quản lý thành viên nâng cao

Mục tiêu chính: Phát triển các tính năng bổ sung và nâng cao quản lý thành viên trong workspace.

Kết quả đạt được sau tuần đó:

Hoàn thành CRUD (tạo, đọc, cập nhật, xóa) cho ghi chú, lịch sự kiện, và mục tiêu cá nhân, tích hợp với giao diện.

Xây dựng chức năng task với các trạng thái (chưa làm, đang làm, đã làm) và gán task cho thành viên.

Nâng cao quản lý thành viên với tìm kiếm theo tên/vai trò và cập nhật phân quyền linh hoạt.

Tuần 5: Xác minh email + Phân quyền admin toàn hệ thống

Mục tiêu chính: Hoàn thiện bảo mật tài khoản và chức năng quản lý của admin.

Kết quả đạt được sau tuần đó:

Tích hợp tính năng xác minh email sau đăng ký, gửi thông báo qua email để kích hoạt tài khoản.

Phát triển phân quyền admin, cho phép quản lý toàn bộ người dùng (xem, cập nhật, xóa, vô hiệu hóa) qua giao diện admin.

Tuần 6: Thiết kế lại giao diện UI/UX đẹp hơn + responsive

Mục tiêu chính: Cải thiện trải nghiệm người dùng với giao diện chuyên nghiệp và hỗ trợ nhiều thiết bị.

Kết quả đạt được sau tuần đó:

Nâng cấp giao diện bằng Tailwind CSS, tạo layout chuyên nghiệp với màu sắc và bố cục tối ưu.

Đảm bảo responsive design, hỗ trợ tốt trên di động, tablet, và máy tính (từ 320px đến 1920px).

Tối ưu UX/UI với các nút điều hướng rõ ràng và hiệu ứng mượt mà.

Tuần 7: Tìm kiếm + Thống kê + Kiểm thử hệ thống toàn bộ

Mục tiêu chính: Thêm chức năng tìm kiếm, thống kê, và kiểm tra chất lượng hệ thống.

Kết quả đạt được sau tuần đó:

Tích hợp tính năng tìm kiếm và lọc (theo thời gian, trạng thái, tiêu đề) trên dashboard.

Xây dựng thống kê dưới dạng biểu đồ (số lượng task, giao dịch) bằng React.js.

Thực hiện kiểm thử toàn hệ thống, sửa lỗi và đảm bảo mọi chức năng hoạt động ổn định.

Tuần 8: Hoàn thiện tính năng + Làm báo cáo + Quay video demo + Slide thuyết trình

Mục tiêu chính: Kết thúc dự án với sản phẩm hoàn chỉnh và tài liệu trình bày.

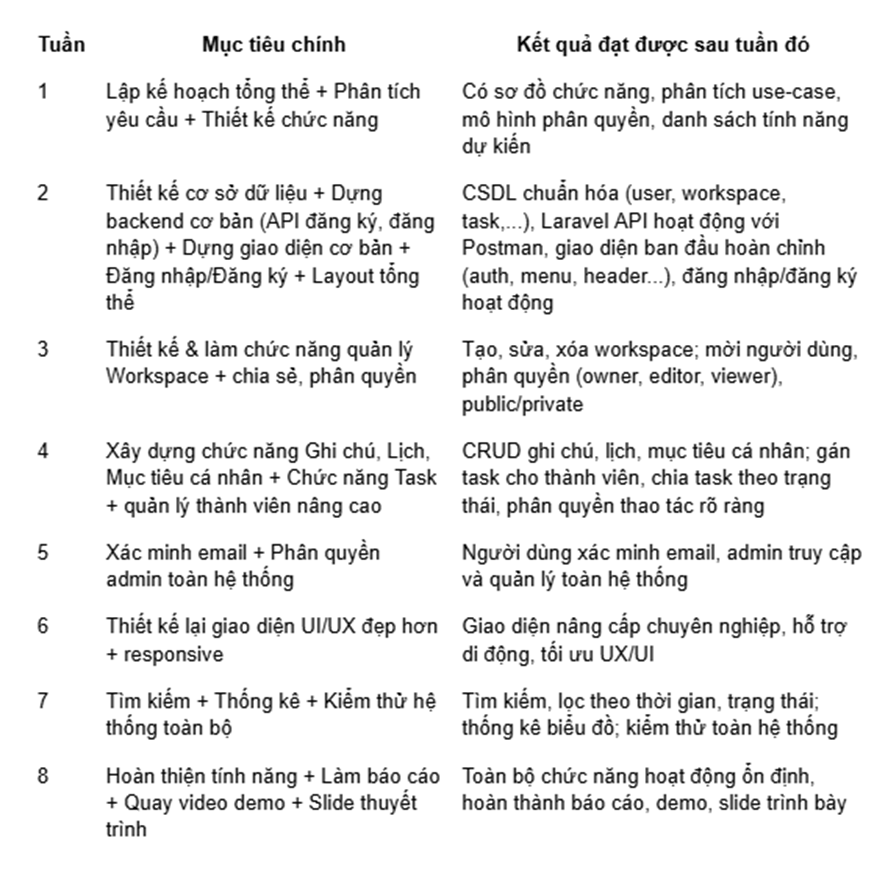
Kết quả đạt được sau tuần đó:

Hoàn thiện tất cả tính năng, đảm bảo không còn lỗi lớn (bug).

Hoàn thành báo cáo chi tiết, bao gồm tất cả các phần từ thiết kế đến triển khai.

Quay video demo minh họa cách sử dụng các tính năng chính.

Chuẩn bị slide thuyết trình với nội dung cô đọng, hình ảnh minh họa.



Hình 52. Kế hoạch dự án

## Đảm bảo thực hiện đúng làm việc nhóm

Họp định kỳ mỗi tuần 1 lần:

Tổ chức họp vào cuối mỗi tuần (thứ 6 hoặc thứ 7) qua Google Meet, kéo dài khoảng 1 giờ.

Mục tiêu: Xem lại tiến độ, phân công nhiệm vụ cho tuần tiếp theo, và giải quyết các vấn đề phát sinh.

Ghi chú cuộc họp được lưu trên Google Docs để tất cả thành viên theo dõi.

Sử dụng Google Docs/Drive để chia sẻ tài liệu:

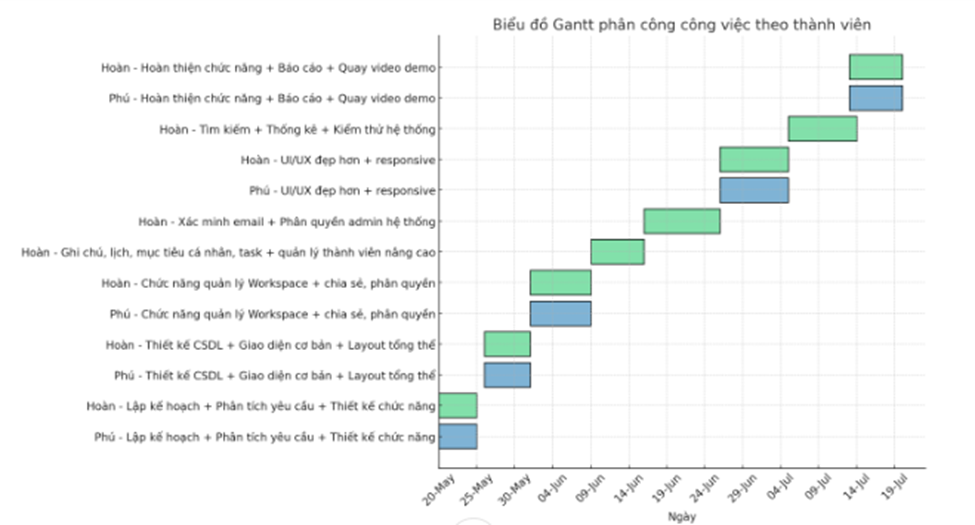
Lưu trữ tất cả tài liệu (yêu cầu, thiết kế, báo cáo) trên Google Drive, chia sẻ với quyền chỉnh sửa cho nhóm.

Sử dụng Google Docs để soạn thảo chung, theo dõi phiên bản mới nhất, và nhận phản hồi từ các thành viên.

Chia deadline nhỏ, theo dõi qua Trello hoặc GitHub:

Phân chia công việc thành các nhiệm vụ nhỏ với deadline cụ thể (ví dụ: hoàn thành API đăng ký trong 2 ngày).

Theo dõi tiến độ trên Trello với các bảng (To Do, In Progress, Done) hoặc GitHub Issues, đảm bảo mọi thành viên cập nhật trạng thái công việc.



Hình 53. Biểu đồ Gantt phân công công việc theo thành viên

## Các vấn đề đạo đức và làm việc chuyên nghiệp

Cam kết sử dụng nguồn tài nguyên hợp pháp:

Chỉ sử dụng tài liệu, tài nguyên (hình ảnh, code mẫu) có bản quyền hoặc miễn phí, được phép sử dụng cho mục đích học tập.

Trích dẫn nguồn rõ ràng nếu tham khảo từ tài liệu bên ngoài.

Không sao chép mã nguồn từ các hệ thống thương mại:

Đảm bảo tất cả mã nguồn (Node.js, React.js, MySQL) được viết bởi nhóm, tránh sao chép từ các sản phẩm thương mại như Trello hay Asana.

Kiểm tra lẫn nhau trong nhóm để đảm bảo tính độc đáo.

Giao tiếp lịch sự, minh bạch trong nhóm:

Duy trì thái độ tôn trọng, lắng nghe ý kiến mỗi thành viên trong các cuộc họp.

Báo cáo tiến độ và khó khăn một cách trung thực, tránh che giấu vấn đề.

## Tác động xã hội

Giúp cá nhân quản lý cuộc sống hiệu quả hơn:

Hỗ trợ cá nhân theo dõi công việc hàng ngày, lịch trình, và chi tiêu cá nhân, giảm áp lực từ sự phân mảnh thông tin.

Ví dụ: Sinh viên có thể dùng để quản lý bài tập và thời gian học.

Tăng hiệu suất làm việc nhóm:

Cung cấp công cụ phân quyền, gán công việc, và đồng bộ lịch, giúp nhóm nhỏ (gia đình, đội nhóm) làm việc hiệu quả hơn.

Giảm thời gian họp hành nhờ thông tin tập trung trên dashboard.

Hỗ trợ các nhóm nhỏ như gia đình, sinh viên, freelancer làm việc chuyên nghiệp:

Tích hợp quản lý tài chính và mục tiêu cá nhân, phù hợp với nhu cầu của gia đình hoặc freelancer.

Giúp sinh viên tổ chức nhóm học tập, chia sẻ tài liệu và tiến độ học tập.

## Kế hoạch học tập và kiến thức mới

Tìm hiểu thêm về công nghệ frontend/backend mới (React, Flutter,...):

Đăng ký khóa học online (như Coursera, freeCodeCamp) để học sâu về React.js, tối ưu giao diện với Tailwind CSS.

Thử nghiệm Flutter để phát triển ứng dụng di động trong tương lai, mở rộng từ dự án hiện tại.

Học về quản lý dự án phần mềm:

Nghiên cứu phương pháp Agile/Scrum qua tài liệu hoặc workshop, áp dụng vào quản lý nhóm cho dự án tiếp theo.

Thực hành sử dụng Trello hoặc Jira để lập kế hoạch chi tiết hơn.

Làm quen với mô hình Agile/Scrum:

Tham gia nhóm học tập về Agile, thực hành các sprint ngắn (2-4 tuần) để cải thiện quy trình làm việc.

Áp dụng retrospective (phản hồi sau mỗi sprint) để rút kinh nghiệm từ dự án này.

# Tài liệu tham khảo

[1] Trello – [www.trello.com](http://www.trello.com)  
[2] Notion – [www.notion.so](http://www.notion.so)  
[3] Google Calendar – calendar.google.com  
[4] Money Lover – [www.moneylover.me](http://www.moneylover.me)

<https://thinhnotes.com/chuyen-nghe-ba/use-case-diagram-va-5-sai-lam-thuong-gap/>  
<https://innovirtual.de/class-diagram-a-quick-overview/>

<https://codelearn.io/sharing/sequence-diagram-trong-uml?srsltid=AfmBOop1HjmV-6ubwnIxcjfCc0QrD-PUkFoO4uzdckNw6oZ5LnZDAwIi>

<https://thinhnotes.com/chuyen-nghe-ba/erd-la-gi/>

<https://react.dev/learn>

<https://tailwindcss.com/docs/installation/using-vite>

<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

<https://expressjs.com/>

<https://react-icons.github.io/react-icons/>

<https://dev.mysql.com/doc/workbench/en/>