TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM

**VIỆN CÔNG NGHỆ VIỆT - NHẬT**

ĐỒ ÁN MÔN HỌC

CHATTING APP

Ngành: **Công Nghệ Thông Tin**

Học phần: **Lập trình mạng máy tính**

Giảng viên hướng dẫn: **ThS. Trần Ngọc Kiến Phúc**

#### Sinh viên thực hiện:

1/ **Nguyễn Lê Tiến Dũng** MSSV: 2280600401 Lớp: 22DTHJA2 2/ **Nguyễn Hữu Nhân** MSSV: 2280602174 Lớp: 22DTHJA2

#### TP. Hồ Chí Minh, 2025

# LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu giao tiếp, kết nối và chia sẻ thông tin tức thời đã trở thành một phần không thể thiếu trong công việc và cuộc sống. Các ứng dụng trò chuyện trực tuyến (Chatting app) ra đời đã phá vỡ mọi rào cản về không gian và thời gian, cho phép người dùng dễ dàng kết nối, cộng tác và duy trì liên lạc với nhau một cách nhanh chóng, an toàn và tiện lợi.

Nhận thấy tầm quan trọng của vấn đề này, nhóm chúng em đã lựa chọn đề tài “Xây dựng Ứng dụng Chatting thời gian thực” nhằm xây dựng một hệ thống cho phép người dùng giao tiếp hiệu quả. Hệ thống hướng đến việc cung cấp một nền tảng trò chuyện ổn định, bảo mật và thân thiện, giúp nâng cao khả năng tương tác và hiệu suất làm việc trong môi trường số.

Trong quá trình thực hiện, nhóm đã áp dụng các kiến thức về lập trình mạng, cụ thể là các công nghệ như Java Spring Boot, Spring Security, WebSocket, RESTful API, cùng với HTML, CSS, JavaScript để phát triển hệ thống. Mặc dù đã nỗ lực hết mình, nhóm nhận thấy sản phẩm vẫn còn những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy cô để có thể hoàn thiện đề tài và phát triển hệ thống ngày càng tốt hơn.

Sinh viên thực hiện **Nguyễn Lê Tiến Dũng Nguyễn Hữu Nhân**

# LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến quý Thầy Cô Khoa Công Nghệ Thông Tin, Trường Đại học Công nghệ TP.HCM (HUTECH), đã tận tình giảng dạy và trang bị cho chúng em những kiến thức quý báu về Công nghệ Thông tin, giúp chúng em có nền tảng vững chắc để thực hiện đồ án này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến ThS. Trần Ngọc Kiến Phúc – người đã trực tiếp hướng dẫn, hỗ trợ và tạo điều kiện thuận lợi để chúng em có thể hoàn thành đề tài. Những đóng góp, định hướng và những kiến thức mà thầy truyền đạt đã giúp chúng em rất nhiều trong quá trình nghiên cứu và phát triển hệ thống.

Chúng em cũng xin gửi lời cảm ơn đến bạn bè và các anh chị đã luôn động viên, hỗ trợ chúng em trong suốt quá trình học tập và thực hiện đồ án. Sự giúp đỡ quý báu đó là nguồn động lực để chúng em hoàn thành tốt nhiệm vụ của mình.

*TP. Hồ Chí Minh, ngày 15 tháng 10 năm 2025*

# LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan rằng nội dung của Đồ Án Học Phần "Chatting App" là sản phẩm do chính nhóm thực hiện. Những nội dung trình bày trong báo cáo là kết quả của quá trình học tập, nghiên cứu và làm việc nghiêm túc của nhóm.

Toàn bộ tài liệu tham khảo được sử dụng trong quá trình thực hiện đồ án đều có nguồn gốc rõ ràng và được trích dẫn hợp pháp. Nhóm chúng em xin chịu hoàn toàn trách nhiệm về tính trung thực và chính xác của nội dung trong báo cáo này.

Nhóm chúng em xin cam đoan và chịu trách nhiệm trước nhà trường về lời cam đoan này.

*TP. Hồ Chí Minh, ngày 15 tháng 10 năm 2025*

##### Người cam đoan Nguyễn Lê Tiến Dũng Nguyễn Hữu Nhân

# BẢNG PHÂN CÔNG

|  |  |  |
| --- | --- | --- |
| **Họ & Tên** | **Công việc** | **Ghi chú** |
| Nguyễn Lê Tiến Dũng | * Thiết kế CSDL. * Làm backend cho sản phẩm. * Làm báo cáo. * Viết các kịch bản kiểm thử sản phẩm. | 50% |
| Nguyễn Hữu Nhân | * Làm frontend cho sản phẩm. * Làm slide thuyết trình. * Kiểm thử sản phẩm. * Deloy sản phẩm | 50% |

# MỤC LỤC

[LỜI MỞ ĐẦU 1](#_bookmark0)

[LỜI CẢM ƠN 2](#_bookmark1)

[LỜI CAM ĐOAN 3](#_bookmark2)

[BẢNG PHÂN CÔNG 4](#_bookmark3)

[MỤC LỤC 5](#_bookmark4)

[DANH MỤC BẢNG 8](#_bookmark5)

[DANH MỤC HÌNH ẢNH 9](#_bookmark6)

[CHƯƠNG 1: TỔNG QUAN 11](#_bookmark7)

* 1. [Lý do chọn đề tài 11](#_bookmark8)
  2. [Mục tiêu đề tài 11](#_bookmark9)
  3. [Phương pháp nghiên cứu đề tài 12](#_bookmark10)
  4. [Ý nghĩa của việc nghiên cứu đề tài 12](#_bookmark11)

[CHƯƠNG 2: KHẢO SÁT 13](#_bookmark12)

* 1. [Quy trình quản lý 13](#_bookmark13)
  2. [Nghiệp vụ 14](#_bookmark15)
  3. [Xác định yêu cầu chức năng 14](#_bookmark16)

[CHƯƠNG 3: MÔ HÌNH HÓA NGHIỆP VỤ 17](#_bookmark18)

* 1. [SƠ ĐỒ CHỨC NĂNG (USE CASE) 17](#_bookmark19)
     1. [Sơ đồ tổng quát 17](#_bookmark20)
     2. [Sơ đồ đăng nhập 17](#_bookmark22)
     3. [Sơ đồ tài khoản 18](#_bookmark25)
     4. [Sơ đồ quản lý nhóm làm việc. 19](#_bookmark28)
     5. [Sơ đồ quản lý dự án. 20](#_bookmark31)
     6. [Sơ đồ quản lý công việc 21](#_bookmark34)
     7. [Sơ đồ báo cáo công việc. 22](#_bookmark37)
  2. [SƠ ĐỒ QUAN HỆ CƠ SỞ DỮ LIỆU 23](#_bookmark40)

[CHƯƠNG 4: SƠ ĐỒ HOẠT ĐỘNG (ACTIVITY DIAGRAM) 24](#_bookmark42)

* 1. [Sơ đồ đăng nhập 24](#_bookmark43)
  2. [Sơ đồ hoạt động tạo nhóm làm việc 24](#_bookmark45)
  3. [Sơ đồ hoạt động sửa nhóm làm việc 25](#_bookmark47)
  4. [Sơ đồ hoạt động xóa nhóm làm việc 25](#_bookmark49)
  5. [Sơ đồ hoạt động thêm thành viên vào nhóm 26](#_bookmark51)
  6. [Sơ đồ hoạt động xóa thành viên khỏi nhóm 26](#_bookmark53)
  7. [Sơ đồ hoạt động tạo dự án 27](#_bookmark55)
  8. [Sơ đồ hoạt động sửa dự án 27](#_bookmark57)
  9. [Sơ đồ hoạt động xóa dự án 28](#_bookmark59)
  10. [Sơ đồ hoạt động thêm công việc 28](#_bookmark61)
  11. [Sơ đồ hoạt động sửa công việc 29](#_bookmark63)
  12. [Sơ đồ hoạt động xóa công việc 29](#_bookmark65)
  13. [Sơ đồ hoạt động nộp báo cáo 30](#_bookmark67)
  14. [Sơ đồ hoạt động xóa báo cáo 30](#_bookmark69)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN 31](#_bookmark71)

* 1. [Giao diện trang chủ 31](#_bookmark72)
  2. [Giao diện đăng nhập 31](#_bookmark75)
  3. [Giao diện đăng ký 32](#_bookmark78)
  4. [Giao diện danh sách dự án 33](#_bookmark81)
  5. [Giao diện tạo dự án 33](#_bookmark84)
  6. [Giao diện chi tiết dự án 34](#_bookmark87)
  7. [Giao diện chỉnh sửa dự án 35](#_bookmark90)
  8. [Giao diện thêm thành viên vào dự án 35](#_bookmark93)
  9. [Giao diện giao việc cho thành viên trong dự án 36](#_bookmark96)
  10. [Giao diện chỉnh sửa công việc 37](#_bookmark99)
  11. [Giao diện nộp báo cáo 38](#_bookmark102)
  12. [Giao diện chức năng vô hiệu hóa tài khoản và dự án cho người dùng admin 39](#_bookmark105)
  13. [Giao diện chức năng quản lý nhóm làm việc 40](#_bookmark109)
  14. [Giao diện chức năng kéo thả thay đổi trạng thái công việc 41](#_bookmark112)
  15. [Giao diện chức năng trò chuyện giữa các thành viên trong nhóm 42](#_bookmark115)

[CHƯƠNG 6: HƯỚNG DẪN SỬ DỤNG 44](#_bookmark118)

* 1. [Truy cập Hệ thống 44](#_bookmark119)
  2. [Quản lý Tài khoản 44](#_bookmark120)
  3. [Quản lý Dự án 44](#_bookmark121)
  4. [Quản lý Công việc 45](#_bookmark122)
  5. [Ghi chú, Báo cáo & Tìm kiếm 45](#_bookmark123)

[CHƯƠNG 7: KẾT LUẬN 47](#_bookmark124)

* 1. [Kết quả đạt được 47](#_bookmark125)
  2. [Nhược điểm của hệ thống 48](#_bookmark126)
  3. [Các tính năng đã phát triển được 48](#_bookmark127)
  4. [Các tính năng chưa thực hiện được 49](#_bookmark128)
  5. [Hướng phát triển 49](#_bookmark129)

[TÀI LIỆU THAM KHẢO 50](#_bookmark130)

# DANH MỤC BẢNG

*[Bảng 2.1: Khảo sát người dùng.](#_bookmark14)* [14](#_bookmark14)

*[Bảng 2.2: Mô tả chức năng](#_bookmark17)* [16](#_bookmark17)

*[Bảng 3.1: Mô tả sơ đồ đăng nhập.](#_bookmark24)* [18](#_bookmark24)

*[Hình 3.3: Sơ đồ tài khoản](#_bookmark26)* [18](#_bookmark26)

*[Bảng 3.2: Mô tả sơ đồ tài khoản.](#_bookmark27)* [19](#_bookmark27)

*[Bảng 3.3: Mô tả sơ đồ quản lý nhóm làm việc](#_bookmark30)* [20](#_bookmark30)

*[Bảng 3.4: Mô tả sơ đồ quản lý dự án.](#_bookmark33)* [21](#_bookmark33)

*[Bảng 3.5: Mô tả sơ đồ quản lý công việc](#_bookmark36)* [22](#_bookmark36)

*[Bảng 3.6: Mô tả sơ đồ báo cáo công việc](#_bookmark39)* [23](#_bookmark39)

*[Bảng 5.1: Ý nghĩa giao diện trang chủ](#_bookmark74)* [31](#_bookmark74)

*[Bảng 5.2: Mô tả giao diện đăng nhập](#_bookmark77)* [32](#_bookmark77)

*[Bảng 5.3: Mô tả giao diện đăng ký.](#_bookmark80)* [32](#_bookmark80)

*[Bảng 5.4: Ý nghĩa giao diện danh sách dự án.](#_bookmark83)* [33](#_bookmark83)

*[Bảng 5.5: Mô tả giao diện tạo dự án.](#_bookmark86)* [34](#_bookmark86)

*[Bảng 5.6: Mô tả giao diện chi tiết dự án.](#_bookmark89)* [35](#_bookmark89)

*[Bảng 5.7: Ý nghĩa giao diện chỉnh sửa dự án.](#_bookmark92)* [35](#_bookmark92)

*[Bảng 5.8: Ý nghĩa giao diện thêm thành dự án.](#_bookmark95)* [36](#_bookmark95)

*[Bảng 5.9: Ý nghĩa giao diện giao việc cho thành viên trong dự án.](#_bookmark98)* [37](#_bookmark98)

*[Bảng 5.10: Ý nghĩa giao diện chỉnh sửa công việc](#_bookmark101)* [38](#_bookmark101)

*[Bảng 5.11: Ý nghĩa giao diện nộp báo cáo](#_bookmark104)* [39](#_bookmark104)

*[Bảng 5.12: Ý nghĩa giao diện chức năng vô hiệu hóa tài khoản và dự án cho người](#_bookmark108) [dùng admin](#_bookmark108)* [40](#_bookmark108)

*[Bảng 5.13: Ý nghĩa giao diện chức năng quản lý nhóm làm việc](#_bookmark111)* [41](#_bookmark111)

*[Hình 5.15: Giao diện chức năng kéo thả thay đổi trạng thái công việc](#_bookmark113)* [42](#_bookmark113)

*[Bảng 5.14: Ý nghĩa giao diện chức năng kéo thả thay đổi trạng thái](#_bookmark114)* [42](#_bookmark114)

*[Bảng 5.15: Ý nghĩa giao diện chức năng trò chuyện](#_bookmark117)* [43](#_bookmark117)

# DANH MỤC HÌNH ẢNH

*[Hình 3.1: Sơ đồ tổng quát](#_bookmark21)* [17](#_bookmark21)

*[Hình 3.2: Sơ đồ đăng nhập](#_bookmark23)* [17](#_bookmark23)

*[Hình 3.3: Sơ đồ tài khoản](#_bookmark26)* [18](#_bookmark26)

*[Hình 3.4: Sơ đồ quản lý nhóm làm việc](#_bookmark29)* [19](#_bookmark29)

*[Hình 3.5: Sơ đồ quản lý dự án.](#_bookmark32)* [20](#_bookmark32)

*[Hình 3.6: Sơ đồ quản lý công việc](#_bookmark35)* [21](#_bookmark35)

*[Hình 3.7: Sơ đồ báo cáo công việc](#_bookmark38)* [22](#_bookmark38)

*[Hình 3.8: Mô hình Diagrams](#_bookmark41)* [23](#_bookmark41)

*[Hình 4.1: Mô hình hoạt động đăng nhập](#_bookmark44)* [24](#_bookmark44)

*[Hình 4.2: Mô hình hoạt động tạo nhóm làm việc](#_bookmark46)* [24](#_bookmark46)

*[Hình 4.3: Mô hình hoạt động sửa nhóm làm việc](#_bookmark48)* [25](#_bookmark48)

*[Hình 4.4: Mô hình hoạt động xóa nhóm làm việc](#_bookmark50)* [25](#_bookmark50)

*[Hình 4.5: Mô hình hoạt động thêm thành viên vào nhóm](#_bookmark52)* [26](#_bookmark52)

*[Hình 4.6: Mô hình hoạt động xóa thành viên khỏi nhóm](#_bookmark54)* [26](#_bookmark54)

*[Hình 4.7: Mô hình hoạt động tạo dự án](#_bookmark56)* [27](#_bookmark56)

*[Hình 4.8: Mô hình hoạt động sửa dự án.](#_bookmark58)* [27](#_bookmark58)

*[Hình 4.9: Mô hình hoạt động xóa dự án.](#_bookmark60)* [28](#_bookmark60)

*[Hình 4.10: Mô hình hoạt động thêm công việc](#_bookmark62)* [28](#_bookmark62)

*[Hình 4.11: Mô hình hoạt động sửa công việc](#_bookmark64)* [29](#_bookmark64)

*[Hình 4.12: Mô hình hoạt động xóa công việc](#_bookmark66)* [29](#_bookmark66)

*[Hình 4.13: Mô hình hoạt động xóa công việc](#_bookmark68)* [30](#_bookmark68)

*[Hình 4.14: Mô hình hoạt động xóa báo cáo](#_bookmark70)* [30](#_bookmark70)

*[Hình 5.1: Giao diện trang chủ](#_bookmark73)* [31](#_bookmark73)

*[Hình 5.2: Giao diện đăng nhập](#_bookmark76)* [31](#_bookmark76)

*[Hình 5.3: Giao diện đăng ký.](#_bookmark79)* [32](#_bookmark79)

*[Hình 5.4: Giao diện danh sách dự án.](#_bookmark82)* [33](#_bookmark82)

*[Hình 5.5: Giao diện tạo dự án.](#_bookmark85)* [33](#_bookmark85)

*[Hình 5.6: Giao diện chi tiết dự án.](#_bookmark88)* [34](#_bookmark88)

*[Hình 5.7: Giao diện chỉnh sửa dự án.](#_bookmark91)* [35](#_bookmark91)

*[Hình 5.8: Giao diện thêm thành viên vào dự án](#_bookmark94)* [36](#_bookmark94)

*[Hình 5.9: Giao diện giao việc cho thành viên trong dự án.](#_bookmark97)* [36](#_bookmark97)

*[Hình 5.10: Giao diện chỉnh sửa công việc.](#_bookmark100)* [37](#_bookmark100)

*[Hình 5.11: Giao diện nộp báo cáo](#_bookmark103)* [38](#_bookmark103)

*[Hình 5.12: Giao diện chức năng vô hiệu hóa tài khoản cho người dùng admin](#_bookmark106)* [39](#_bookmark106)

*[Hình 5.13: Giao diện chức năng vô hiệu hóa dự án cho người dùng admin](#_bookmark107)* [39](#_bookmark107)

*[Hình 5.14: Giao diện chức năng quản lý nhóm làm việc.](#_bookmark110)* [41](#_bookmark110)

*[Hình 5.15: Giao diện chức năng kéo thả thay đổi trạng thái công việc](#_bookmark113)* [42](#_bookmark113)

*[Hình 5.16: Giao diện chức năng trò chuyện giữa các thành viên trong nhóm](#_bookmark116)* [43](#_bookmark116)

# CHƯƠNG 1: TỔNG QUAN

## Lý do chọn đề tài

Trong thời đại kết nối số, giao tiếp tức thời đã trở thành nền tảng cho mọi hoạt động xã hội, từ công việc cộng tác, học tập trực tuyến cho đến trao đổi thông tin cá nhân. Khả năng kết nối và chia sẻ thông điệp một cách nhanh chóng, ổn định và bảo mật không còn là một lựa chọn mà đã trở thành một yêu cầu thiết yếu. Các ứng dụng trò chuyện (chatting app) chính là công cụ đáp ứng nhu cầu đó, đóng vai trò trung tâm trong việc duy trì dòng chảy thông tin liên tục trong thế giới hiện đại.

Tuy nhiên, dưới góc độ của người học chuyên ngành mạng máy tính, việc xây dựng một ứng dụng chat không chỉ đơn thuần là tạo ra một sản phẩm. Đây là một cơ hội quý báu để đối mặt và giải quyết những thách thức cốt lõi của ngành: làm thế nào để thiết lập và duy trì hàng ngàn kết nối đồng thời? Làm thế nào để đảm bảo tin nhắn được gửi và nhận gần như ngay lập tức (real-time) mà vẫn toàn vẹn? Và làm thế nào để bảo mật dữ liệu trên đường truyền? Các giao thức truyền thống như HTTP có những hạn chế nhất định trong việc xử lý giao tiếp hai chiều liên tục, đòi hỏi phải có những giải pháp chuyên biệt hơn.

Chính vì vậy, nhóm chúng em đã quyết định lựa chọn đề tài “Xây dựng Ứng dụng Chatting thời gian thực”. Đề tài này là một bài toán thực tế lý tưởng để chúng em có thể áp dụng sâu sắc các kiến thức đã học về lập trình mạng, đặc biệt là về mô hình client-server, giao thức WebSocket, quản lý phiên và xử lý đa luồng. Thông qua việc phát triển ứng dụng, chúng em mong muốn không chỉ củng cố nền tảng lý thuyết mà còn tích lũy kinh nghiệm thực tiễn trong việc xây dựng một hệ thống mạng hoàn chỉnh, hiệu quả và có khả năng mở rộng.

## Mục tiêu đề tài

Đề tài “Xây dựng Ứng dụng Chatting thời gian thực” được thực hiện với mục tiêu chính là xây dựng một hệ thống giao tiếp linh hoạt, cho phép người dùng kết nối, gửi và nhận tin nhắn một cách tức thời, an toàn và hiệu quả. Ứng dụng hướng đến việc cung cấp một nền tảng giao tiếp ổn định, giúp người dùng có thể duy trì liên lạc mọi lúc, mọi nơi. Cụ thể, đề tài tập trung vào các mục tiêu sau:

* Xây dựng một ứng dụng chat hoàn chỉnh: Phát triển một hệ thống client-server cho phép người dùng đăng ký, đăng nhập, quản lý danh sách bạn bè và trò chuyện thời gian thực trong các phòng chat riêng tư hoặc phòng chat nhóm.
* Triển khai giao tiếp thời gian thực: Áp dụng giao thức WebSocket để đảm bảo tin nhắn được gửi và nhận gần như ngay lập tức, mang lại trải nghiệm tương tác mượt mà và liền mạch cho người dùng.
* Đảm bảo tính an toàn và bảo mật: Xây dựng cơ chế xác thực người dùng an toàn, quản lý phiên làm việc hiệu quả và mã hóa dữ liệu trên đường truyền để bảo vệ thông tin và quyền riêng tư của người dùng.
* Củng cố kiến thức và kỹ năng thực tiễn: Áp dụng các kiến thức đã học về lập trình mạng, cơ sở dữ liệu và các công nghệ liên quan (Spring Boot, Spring Security, v.v.) vào một dự án thực tế, từ đó nâng cao kỹ năng phân tích, thiết kế và phát triển hệ thống phần mềm.

## Phương pháp nghiên cứu đề tài

Để đạt được các mục tiêu đã đề ra, nhóm đã tiến hành thực hiện đề tài theo các bước sau:

Nghiên cứu lý thuyết và thu thập dữ liệu: Nhóm tiến hành tìm hiểu sâu về kiến thức liên quan đến mạng máy tính, đặc biệt là giao thức WebSocket và sự khác biệt so với giao thức HTTP truyền thống trong giao tiếp real-time. Bên cạnh đó, nhóm nghiên cứu về mô hình publish-subscribe, giao thức STOMP, kiến trúc client-server cho ứng dụng chat, và các nguyên lý trong việc xây dựng ứng dụng web hiện đại. Ngoài ra, nhóm đã khảo sát các ứng dụng trò chuyện phổ biến như Messenger, Zalo, Telegram để hiểu rõ kiến trúc, luồng hoạt động, các tính năng cần thiết và các thách thức về hiệu năng, bảo mật trong giao tiếp thời gian thực.

Phân tích và thiết kế hệ thống: Dựa trên kết quả nghiên cứu, nhóm tiến hành phân tích và xác định các yêu cầu chức năng (đăng nhập, gửi/nhận tin nhắn, quản lý bạn bè, tạo phòng chat) và phi chức năng (hiệu năng, bảo mật, độ trễ thấp). Hệ thống được thiết kế theo kiến trúc client-server, kết hợp giữa RESTful API cho các tác vụ quản lý (người dùng, phòng chat) và WebSocket cho việc truyền tin nhắn. Nhóm đã xây dựng mô hình cơ sở dữ liệu để lưu trữ thông tin người dùng, tin nhắn, danh sách bạn bè và các phòng chat. Giao diện người dùng (UI) được thiết kế để trực quan, dễ sử dụng và có khả năng cập nhật động khi có tin nhắn mới hoặc thông báo.

Xây dựng và phát triển hệ thống: Nhóm triển khai hệ thống dựa trên Java Spring Boot cho phía server và HTML, CSS, JavaScript cho phía client. Phía server, nhóm tập trung vào việc cấu hình và xây dựng WebSocket endpoint, sử dụng STOMP làm giao thức con để đơn giản hóa việc gửi nhận tin nhắn đến các "topic" hoặc "queue". Phía client, nhóm sử dụng thư viện SockJS và Stomp.js để thiết lập kết nối WebSocket ổn định và tương tác với server. Quá trình phát triển tập trung vào việc xử lý các sự kiện chat (như gửi/nhận tin nhắn, thông báo "đang gõ"), quản lý trạng thái online/offline của người dùng và đảm bảo tính toàn vẹn của dữ liệu.

## Ý nghĩa của việc nghiên cứu đề tài

Việc xây dựng ứng dụng chat thời gian thực có ý nghĩa quan trọng trong bối cảnh thế giới số ngày càng phát triển, nơi nhu cầu kết nối và giao tiếp tức thời trở nên thiết yếu. Về mặt thực tiễn, hệ thống không chỉ là một công cụ liên lạc mà còn là nền tảng cho sự hợp tác trong công việc, học tập và duy trì các mối quan hệ xã hội. Ứng dụng này mang lại giải pháp cho việc trao đổi thông tin nhanh chóng, phá vỡ rào cản về không gian và địa lý, từ đó nâng cao hiệu suất làm việc nhóm và củng cố kết nối giữa các cá nhân.

Bên cạnh ý nghĩa thực tiễn, đề tài còn mang giá trị học thuật cao, giúp chúng em vận dụng kiến thức chuyên sâu từ học phần Lập trình mạng máy tính vào việc giải quyết một bài toán thực tế. Thông qua quá trình nghiên cứu và triển khai, chúng em được củng cố hiểu biết về mô hình client-server, sự khác biệt và ứng dụng của các giao thức mạng như HTTP và WebSocket, đồng thời rèn luyện kỹ năng xử lý kết nối đồng thời, truyền dữ liệu hai chiều và đảm bảo an toàn thông tin trên môi trường mạng.

Những kinh nghiệm tích lũy được từ quá trình thực hiện đề tài sẽ là nền tảng quan trọng, giúp chúng em nâng cao tư duy lập trình, khả năng phân tích và giải quyết vấn đề. Đây là những kỹ năng cốt lõi, giúp chúng em tự tin hơn để sẵn sàng áp dụng vào các dự án công nghệ phức tạp và quy mô lớn hơn trong tương lai.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

## Lập trình mạng

### Khái niệm chung

Lập trình mạng (Network Programming) là một lĩnh vực quan trọng trong khoa học máy tính, tập trung vào việc phát triển các ứng dụng có khả năng giao tiếp, truyền tải và trao đổi dữ liệu thông qua mạng máy tính. Hiểu một cách đơn giản, lập trình mạng cho phép các phần mềm hoặc thiết bị khác nhau có thể “nói chuyện” với nhau thông qua một hệ thống mạng — từ mạng cục bộ (LAN) cho đến mạng diện rộng (WAN) hoặc Internet toàn cầu. Mục tiêu chính của lập trình mạng là tạo ra sự kết nối giữa nhiều máy tính, thiết bị hoặc dịch vụ nhằm chia sẻ thông tin, dữ liệu, tài nguyên và thực hiện các tác vụ phân tán một cách hiệu quả.

Trong lập trình mạng, một hệ thống thường bao gồm hai thành phần cơ bản là client (máy khách) và server (máy chủ). Client là bên gửi yêu cầu, trong khi server là bên tiếp nhận, xử lý yêu cầu và gửi phản hồi lại cho client. Giao tiếp giữa hai bên được thực hiện thông qua các giao thức truyền thông (protocols) như TCP/IP, HTTP, FTP, SMTP, v.v. Các giao thức này quy định cách thức thiết lập kết nối, truyền dữ liệu, kiểm soát lỗi và bảo mật thông tin khi truyền tải qua mạng. Nhờ đó, các ứng dụng có thể hoạt động trơn tru dù chạy trên các thiết bị, hệ điều hành hay vị trí địa lý khác nhau.

Ví dụ, trong các hệ thống web hiện nay, khi người dùng thao tác trên trình duyệt để tải lên một tệp tin, trình duyệt (client) sẽ gửi một yêu cầu HTTP kèm dữ liệu file đến máy chủ (server) – thường được xây dựng bằng các nền tảng như Spring Boot, Node.js hoặc ASP.NET. Máy chủ sẽ tiếp nhận yêu cầu, xử lý thông tin, lưu trữ file vào hệ thống hoặc dịch vụ lưu trữ đám mây, cập nhật dữ liệu vào cơ sở dữ liệu, sau đó gửi phản hồi (response) về cho trình duyệt. Quá trình tưởng chừng đơn giản này thực chất được điều khiển và kiểm soát bằng hàng loạt kỹ thuật lập trình mạng phức tạp, đảm bảo dữ liệu được truyền đi chính xác, an toàn và nhanh chóng.

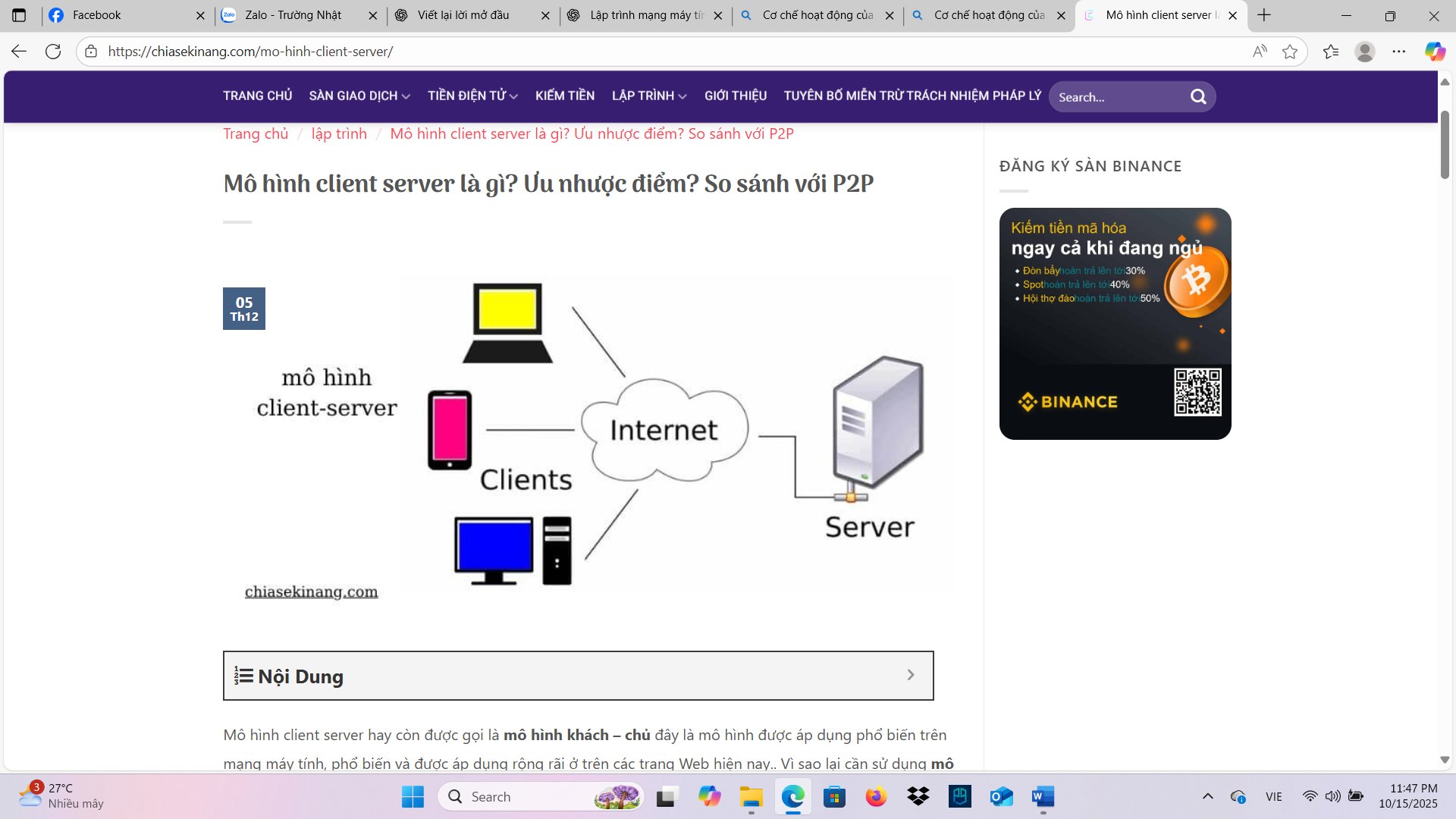
Ngày nay, lập trình mạng không chỉ giới hạn trong việc xây dựng các ứng dụng web hay dịch vụ Internet mà còn mở rộng sang nhiều lĩnh vực khác như IoT (Internet of Things), hệ thống phân tán, điện toán đám mây, trò chơi trực tuyến (online games) hay ứng dụng di động kết nối thời gian thực. Với sự phát triển mạnh mẽ của công nghệ mạng và Internet, lập trình mạng đã trở thành một kỹ năng thiết yếu đối với các lập trình viên, giúp họ xây dựng được những ứng dụng hiện đại, có khả năng kết nối linh hoạt và hoạt động hiệu quả trong môi trường toàn cầu hóa..

### Mô hình Client – Server

Mô hình Client–Server là kiến trúc nền tảng và phổ biến nhất trong lập trình mạng, đóng vai trò quan trọng trong việc tổ chức và vận hành các ứng dụng mạng hiện nay. Đây là mô hình hoạt động dựa trên nguyên tắc phân chia vai trò rõ ràng giữa bên cung cấp dịch vụ (Server) và bên sử dụng dịch vụ (Client). Cách tiếp cận này giúp hệ thống trở nên linh hoạt, dễ mở rộng và dễ bảo trì, đồng thời đảm bảo việc xử lý dữ liệu, truyền thông tin và phản hồi được diễn ra một cách có tổ chức.

Trong mô hình này, Client là các ứng dụng hoặc thiết bị đóng vai trò gửi yêu cầu đến Server. Client có thể là trình duyệt web, ứng dụng di động, phần mềm desktop hoặc bất kỳ thiết bị nào có khả năng kết nối mạng. Client chịu trách nhiệm giao tiếp với người dùng, thu thập dữ liệu (như thông tin đăng nhập, nội dung tin nhắn, yêu cầu kết bạn, v.v.) và gửi yêu cầu tương ứng đến Server. Trong hệ thống đề tài “Ứng dụng Chatting”, Client chính là giao diện web (Frontend), nơi người dùng có thể thao tác trực tiếp như gửi tin nhắn, xem danh sách bạn bè, hoặc tạo một phòng chat mới.

Ngược lại, Server là thành phần đảm nhận vai trò tiếp nhận, xử lý và phản hồi các yêu cầu từ Client. Server có thể hiểu là “bộ não” của hệ thống, nơi chứa toàn bộ logic nghiệp vụ, cơ sở dữ liệu, và các dịch vụ phục vụ cho ứng dụng. Trong đề tài này, Server được xây dựng bằng Spring Boot, một framework mạnh mẽ của Java, cho phép xử lý các yêu cầu HTTP, quản lý kết nối WebSocket, tương tác với cơ sở dữ liệu, và xác thực người dùng. Khi Client gửi yêu cầu, Server sẽ phân tích dữ liệu, thực hiện thao tác cần thiết (ví dụ: lưu tin nhắn, chuyển tiếp tin nhắn đến người nhận, truy vấn lịch sử trò chuyện), sau đó gửi phản hồi về cho Client dưới dạng dữ liệu JSON hoặc thông qua kết nối WebSocket.



*Hình 2.1: Mô hình Client - Server.*

Cơ chế hoạt động của mô hình Client–Server diễn ra theo ba bước cơ bản:

* Client khởi tạo kết nối: Client gửi yêu cầu đến Server thông qua địa chỉ IP của máy chủ và cổng (port) dịch vụ được định sẵn.
* Server lắng nghe và xử lý yêu cầu: Server luôn trong trạng thái “listen” trên cổng đó, chờ đợi yêu cầu đến. Khi nhận được yêu cầu, Server sẽ xử lý theo đúng logic nghiệp vụ.
* Server gửi phản hồi: Sau khi hoàn thành xử lý, Server trả kết quả lại cho Client. Phản hồi này có thể là dữ liệu, thông báo thành công, lỗi hoặc nội dung trang web cần hiển thị.

Nhờ mô hình Client–Server, việc chia tách giữa giao diện người dùng và xử lý nghiệp vụ trở nên rõ ràng, giúp lập trình viên dễ dàng phát triển, bảo trì và mở rộng hệ thống. Ngoài ra, mô hình này còn hỗ trợ khả năng mở rộng quy mô (scalability) khi nhu cầu truy cập tăng cao – ví dụ, có thể triển khai nhiều server xử lý song song hoặc thêm các client truy cập mà không ảnh hưởng đến cấu trúc tổng thể. Chính vì vậy, Client–Server được xem là nền tảng cơ bản và cốt lõi cho hầu hết các ứng dụng mạng hiện đại, từ website, ứng dụng di động cho đến các hệ thống phân tán quy mô lớn.

### Giao thức truyền thông

Trong lập trình mạng, để các ứng dụng có thể giao tiếp được với nhau, việc sử dụng các giao thức truyền thông là vô cùng cần thiết. Giao thức là một tập hợp các quy tắc và tiêu chuẩn về cách thức thiết lập kết nối, trao đổi và xử lý dữ liệu. Trong hệ thống “Ứng dụng Chatting”, việc truyền tải dữ liệu giữa Client và Server được thực hiện thông qua sự kết hợp của nhiều giao thức cốt lõi:

* TCP (Transmission Control Protocol): Là giao thức nền tảng nằm ở tầng vận chuyển, TCP đảm bảo việc truyền dữ liệu một cách đáng tin cậy. Nó thiết lập một kết nối ổn định, kiểm soát lỗi và đảm bảo các gói tin được gửi đi và nhận lại một cách toàn vẹn theo đúng thứ tự. Trong ứng dụng chat, TCP là nền móng cho cả HTTP và WebSocket, giúp mọi tin nhắn và dữ liệu quản lý người dùng không bị thất lạc hay hỏng hóc trên đường truyền.
* HTTP/HTTPS (HyperText Transfer Protocol / Secure) và RESTful API: HTTP là giao thức chính cho các tác vụ không yêu cầu tương tác thời gian thực. Kết hợp với kiến trúc RESTful API, HTTP xử lý các chức năng quản lý của ứng dụng như: đăng ký/đăng nhập tài khoản (POST), lấy danh sách bạn bè (GET), tạo phòng chat (POST), hay tải lịch sử tin nhắn. Dữ liệu được trao đổi qua lại giữa client và server chủ yếu dưới dạng JSON, giúp việc giao tiếp trở nên tường minh và chuẩn hóa.
* WebSocket và STOMP (Simple Text Oriented Messaging Protocol): Đối với các tính năng đòi hỏi sự tương tác tức thời, WebSocket là giao thức không thể thiếu. Khác với HTTP, WebSocket thiết lập một kênh kết nối hai chiều (full-duplex) và liên tục, cho phép server chủ động đẩy dữ liệu (như tin nhắn mới, thông báo "đang gõ phím") đến client ngay lập tức mà không cần client phải gửi yêu cầu. Để việc quản lý và định tuyến tin nhắn qua WebSocket trở nên đơn giản hơn, nhóm đã sử dụng STOMP như một giao thức con, giúp gửi tin nhắn đến đúng người nhận hoặc các kênh (topics) cụ thể một cách hiệu quả..

Tóm lại, sự kết hợp thông minh giữa HTTP (cho các tác vụ quản lý) và WebSocket (cho giao tiếp thời gian thực) là chìa khóa tạo nên một ứng dụng chat hiện đại. Trong khi HTTP đảm bảo tính ổn định cho các hoạt động nền, WebSocket mang lại sự sống động và tức thời cho trải nghiệm trò chuyện của người dùng.

### Socket trong lập trình mạng

Trong lập trình mạng, Socket đóng vai trò như “cánh cửa giao tiếp” giữa hai chương trình đang chạy trên cùng một máy tính hoặc trên các thiết bị khác nhau trong mạng. Nói một cách đơn giản, socket là điểm cuối (endpoint) của một kết nối mạng, cho phép hai ứng dụng – thường là client và server – có thể gửi và nhận dữ liệu qua lại một cách trực tiếp. Mỗi socket được định danh duy nhất bởi địa chỉ IP (đại diện cho thiết bị) và số cổng (port) (đại diện cho ứng dụng cụ thể đang hoạt động trên thiết bị đó).

Khi một chương trình muốn giao tiếp qua mạng, nó phải tạo ra một socket. Sau đó, socket này sẽ được gắn (bind) với một địa chỉ IP cụ thể và một cổng (port) xác định. Một socket có thể hoạt động ở phía server (chờ kết nối) hoặc client (gửi yêu cầu kết nối). Khi client khởi tạo kết nối đến server, hệ thống sẽ tạo ra một kênh truyền thông hai chiều giữa hai socket tương ứng. Từ đó, cả hai bên có thể gửi và nhận dữ liệu thông qua luồng (stream) kết nối đó..

Trong ngôn ngữ lập trình Java, socket được hỗ trợ trực tiếp thông qua các lớp có sẵn trong thư viện chuẩn java.net, bao gồm hai lớp quan trọng:

* ServerSocket: đại diện cho phía máy chủ (server). ServerSocket có nhiệm vụ “lắng nghe” (listen) trên một cổng cố định để chờ các yêu cầu kết nối từ client. Khi có một client kết nối đến, ServerSocket sẽ tạo ra một Socket riêng biệt cho kết nối đó, giúp server có thể phục vụ nhiều client cùng lúc.
* Socket: đại diện cho phía máy khách (client). Khi client muốn kết nối đến server, nó tạo ra một Socket chỉ định địa chỉ IP và cổng của server, sau đó gửi yêu cầu kết nối. Nếu server chấp nhận, kết nối được thiết lập và hai bên có thể bắt đầu trao đổi dữ liệu.

Trong đề tài “Ứng dụng Chatting”, cả hai giao thức chính là HTTP và WebSocket đều được xây dựng trên nền tảng socket TCP. Việc hiểu rõ cơ chế hoạt động của socket giúp ta nắm vững bản chất của quá trình truyền thông mạng. Đặc biệt, kiến thức về socket là nền tảng cốt lõi để hiểu cách WebSocket có thể tạo ra một kênh giao tiếp hai chiều, liên tục, giúp tin nhắn được gửi và nhận tức thời. Bên dưới mọi giao thức web hiện đại, chính socket là thành phần chịu trách nhiệm truyền tải gói dữ liệu từ client đến server và ngược lại.

* + 1. Ứng dụng lập trình mạng trong đề tài

Trong hệ thống “Ứng dụng Chatting”, lập trình mạng đóng vai trò xương sống, là nền tảng cho mọi hoạt động kết nối, trao đổi và xử lý dữ liệu giữa người dùng (client) và máy chủ (server). Việc ứng dụng các kỹ thuật lập trình mạng không chỉ giúp hệ thống hoạt động ổn định, mà còn đảm bảo hiệu năng, bảo mật và khả năng tương tác thời gian thực. Cụ thể, lập trình mạng được triển khai xuyên suốt qua các tầng chức năng trong hệ thống như sau:

**Tầng giao tiếp (Communication Layer)**: Đây là tầng chịu trách nhiệm thiết lập và duy trì luồng trao đổi dữ liệu giữa client và server. Trong ứng dụng chat, tầng này có sự kết hợp của hai loại giao thức chính:

* Giao tiếp bất đồng bộ (Request-Response) qua HTTP: Các chức năng mang tính quản lý và không yêu cầu tương tác tức thời được xử lý qua RESTful API trên nền giao thức HTTP. Ví dụ: đăng ký, đăng nhập, lấy danh sách bạn bè, tạo phòng chat, hay tải lịch sử tin nhắn cũ. Mỗi yêu cầu từ client sẽ được server xử lý và phản hồi lại bằng dữ liệu JSON.
* Giao tiếp thời gian thực (Real-time) qua WebSocket: Các chức năng đòi hỏi sự tương tác tức thời như gửi/nhận tin nhắn, thông báo "đang gõ phím", hay cập nhật trạng thái online được thực hiện qua giao thức WebSocket. WebSocket tạo ra một kênh kết nối hai chiều, liên tục, cho phép server chủ động đẩy dữ liệu đến client ngay khi có sự kiện xảy ra, mang lại trải nghiệm trò chuyện liền mạch.

**Tầng xử lý (Processing Layer)**: Trong môi trường mạng, bảo mật là yếu tố sống còn. Hệ thống áp dụng các biện pháp bảo mật sau:

* Xác thực người dùng và quản lý phiên làm việc.
* Định tuyến (routing) tin nhắn WebSocket đến đúng người nhận hoặc phòng chat.
* Xử lý các yêu cầu RESTful API (thêm bạn, tạo phòng, v.v.).
* Tương tác với cơ sở dữ liệu để lưu trữ và truy xuất thông tin.

**Tầng bảo mật (Security Layer)**: Trong môi trường mạng, bảo mật là yếu tố sống còn. Hệ thống áp dụng các biện pháp bảo mật sau:

* Giao thức HTTPS: Mọi giao tiếp qua HTTP đều có thể được nâng cấp lên HTTPS, sử dụng mã hóa SSL/TLS để bảo vệ dữ liệu trên đường truyền, chống lại nguy cơ nghe lén hoặc tấn công "man-in-the-middle".
* Xác thực dựa trên phiên (Session-based Authentication): Thay vì JWT, hệ thống sử dụng cơ chế session. Khi người dùng đăng nhập thành công, server sẽ tạo một phiên (session) và gửi một định danh phiên (session ID) về cho client (thường được lưu trong cookie). Trong các yêu cầu HTTP tiếp theo, client gửi kèm session ID này để server xác thực. Phiên đăng nhập này cũng được dùng để xác thực và bảo mật cho kết nối WebSocket được thiết lập sau đó.

**Tầng lưu trữ (Storage Layer)**: Tầng này đảm nhiệm việc lưu trữ và truy xuất toàn bộ dữ liệu của ứng dụng. Server sẽ giao tiếp với cơ sở dữ liệu (SQL Server) thông qua các giao thức mạng nội bộ (TCP) để quản lý các thông tin quan trọng như:

* Tài khoản và thông tin cá nhân của người dùng.
* Danh sách bạn bè và các mối quan hệ.
* Thông tin về các phòng chat (thành viên, tên phòng, v.v.).
* Lịch sử tin nhắn của các cuộc trò chuyện.
  1. Công nghệ sử dụng

**Ngôn ngữ và Nền tảng Backend:**

* Java 21: Ngôn ngữ lập trình chính, nổi bật với tính ổn định, bảo mật và hệ sinh thái mạnh mẽ.
* Spring Boot 3.x: Framework cốt lõi giúp phát triển ứng dụng phía server nhanh chóng, tích hợp sẵn các module quan trọng như Spring Web (RESTful API), Spring Data JPA, Spring Security và Spring WebSocket.

**Giao tiếp thời gian thực:**

* WebSocket: Giao thức nền tảng cho việc giao tiếp hai chiều, cho phép server chủ động gửi dữ liệu đến client.
* STOMP (Simple Text Oriented Messaging Protocol): Giao thức con hoạt động trên WebSocket, giúp đơn giản hóa việc gửi và định tuyến tin nhắn đến các kênh (topics) hoặc người dùng cụ thể.

**Cơ sở dữ liệu:**

* Microsoft SQL Server: Hệ quản trị cơ sở dữ liệu quan hệ được sử dụng để lưu trữ toàn bộ dữ liệu của ứng dụng như thông tin người dùng, tin nhắn, phòng chat, danh sách bạn bè.
* Spring Data JPA / Hibernate: Cung cấp cơ chế ORM (Object-Relational Mapping), giúp ánh xạ các đối tượng Java với bảng trong cơ sở dữ liệu, đơn giản hóa việc thao tác dữ liệu.

**Giao diện người dùng (Frontend)**:

* ReactJS: Một thư viện JavaScript mạnh mẽ để xây dựng giao diện người dùng theo kiến trúc component, giúp tạo ra các ứng dụng đơn trang (Single Page Application - SPA) có tính tương tác cao.
* HTML5, CSS3: Nền tảng để xây dựng cấu trúc và định hình giao diện cho các component React.
* SockJS và STOMP.js: Các thư viện JavaScript được tích hợp vào ứng dụng React để xử lý việc kết nối và giao tiếp thời gian thực với server qua WebSocket.

**Bảo mật**:

* Spring Security: Framework cung cấp các giải pháp bảo mật toàn diện.
* Session-based Authentication: Cơ chế xác thực người dùng dựa trên phiên làm việc.
* BCrypt: Thuật toán mã hóa mật khẩu mạnh để lưu trữ mật khẩu người dùng an toàn.

**Công cụ và môi trường phát triển**:

* IDE: Visual Studio Code.
* Hệ điều hành: Windows 11.
* Quản lý mã nguồn: Git & GitHub.
* Kiểm thử API: Postman.
* Quản lý thư viện & build project: Apache Maven.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

* 1. Khảo sát và phân tích yêu cầu hệ thống
     1. Mục tiêu của hệ thống

Mục tiêu chính của đề tài là xây dựng một nền tảng chat trực tuyến (Chatting App) hoạt động trên nền tảng web, cho phép người dùng kết nối và giao tiếp với nhau một cách nhanh chóng, an toàn và tiện lợi:

* Xây dựng chức năng giao tiếp cốt lõi: cho phép người dùng gửi và nhận tin nhắn văn bản, hình ảnh, và tệp tin trong thời gian thực, hỗ trợ tạo các phòng chat cá nhân (1-1) và phòng chat nhóm (nhiều người dùng), cung cấp các tính năng tương tác như chỉnh sửa, xóa tin nhắn và xem lịch sử cuộc trò chuyện.
* Quản lý người dùng và các mối quan hệ xã hội: xây dựng hệ thống đăng ký, đăng nhập và quản lý thông tin cá nhân an toàn, cho phép người dùng gửi, chấp nhận/từ chối lời mời kết bạn và quản lý danh sách bạn bè.
* Nâng cao trải nghiệm người dùng: cung cấp các phản hồi tức thời như trạng thái "đang gõ phím" (typing indicator) và trạng thái online/offline của người dùng, xây dựng hệ thống thông báo (notifications) để người dùng không bỏ lỡ các tin nhắn hay lời mời kết bạn mới, thiết kế giao diện người dùng (UI) hiện đại, thân thiện và dễ sử dụng trên nền tảng ReactJS.
* Định hướng thiết kế hệ thống: đảm bảo an toàn cho dữ liệu và thông tin cá nhân của người dùng thông qua cơ chế xác thực session-based và mã hóa mật khẩu, tối ưu hóa tốc độ gửi và nhận tin nhắn, đảm bảo độ trễ thấp ngay cả khi có nhiều người dùng truy cập đồng thời, hệ thống có khả năng ứng dụng trong thực tế, làm nền tảng cho các giải pháp giao tiếp trong cộng đồng, đội nhóm hoặc doanh nghiệp.

1. * 1. Đối tượng sử dụng hệ thống

Người dùng cá nhân:

* Là những người dùng cá nhân muốn kết nối và trò chuyện với bạn bè, đồng nghiệp hoặc gia đình.
* Thực hiện các thao tác cốt lõi như: đăng ký tài khoản, đăng nhập, tìm kiếm người dùng khác, gửi/chấp nhận lời mời kết bạn.
* Giao tiếp thông qua việc gửi tin nhắn, tệp tin trong các cuộc trò chuyện cá nhân (1-1) hoặc trong các phòng chat nhóm.
* Quản lý thông tin cá nhân, danh sách bạn bè và các phòng chat mà mình tham gia.
  + 1. Yêu cầu chức năng

Dựa trên mục tiêu và đối tượng sử dụng, hệ thống được xây dựng với các nhóm chức năng chính sau:

**Chức năng quản lý người dùng và xác thực:**

* Đăng ký tài khoản mới với các thông tin cơ bản.
* Đăng nhập/đăng xuất an toàn thông qua cơ chế xác thực dựa trên phiên (Session-based).
* Cho phép người dùng quản lý và cập nhật thông tin cá nhân (CRUD).

**Chức năng quản lý bạn bè:**

* Gửi lời mời kết bạn đến người dùng khác.
* Chấp nhận hoặc từ chối lời mời kết bạn.
* Hủy lời mời đã gửi hoặc xóa bạn bè khỏi danh sách.
* Hiển thị danh sách bạn bè của người dùng.

**Chức năng quản lý phòng chat (Chat Rooms):**

* Tự động tạo phòng chat riêng tư (1-1) khi hai người dùng kết bạn.
* Cho phép người dùng tạo các phòng chat nhóm với nhiều thành viên.
* Thêm hoặc xóa thành viên khỏi phòng chat nhóm.
* Cho phép cập nhật thông tin phòng chat (ví dụ: tên phòng).
* Xóa phòng chat.

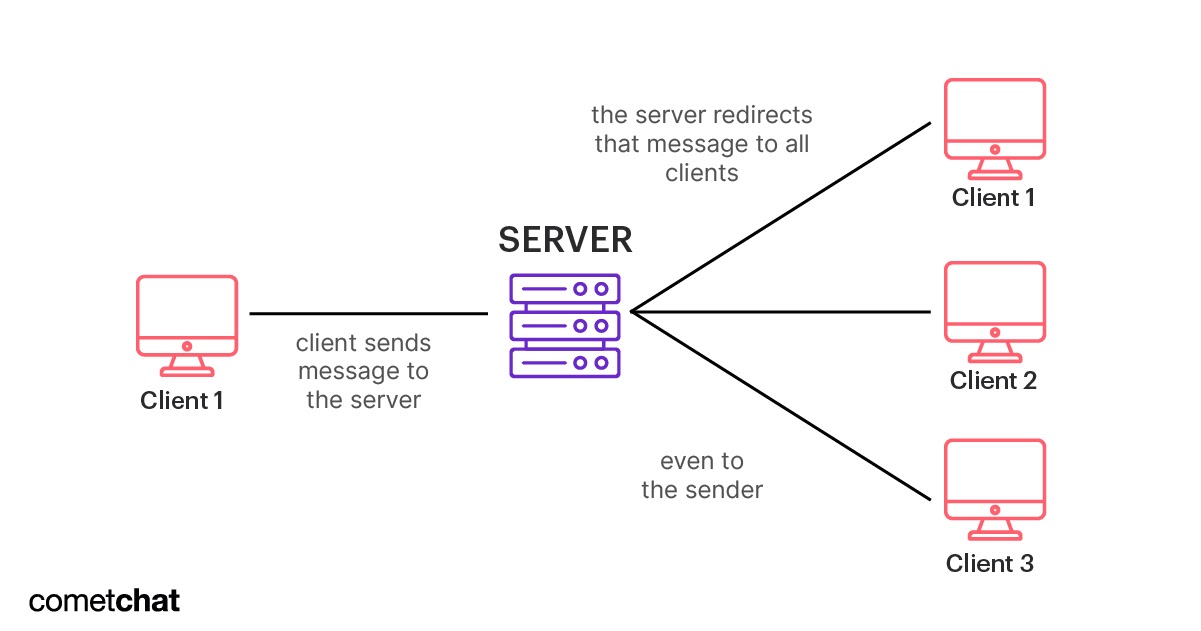
**Chức năng nhắn tin (Messaging):**

* Gửi và nhận tin nhắn văn bản trong thời gian thực.
* Hỗ trợ gửi tệp tin và media (hình ảnh, video) thông qua cơ chế upload multipart.
* Cho phép chỉnh sửa hoặc xóa tin nhắn đã gửi.
* Hiển thị trạng thái tin nhắn (đã gửi, đã xem).

**Chức năng thông báo và trạng thái:**

* Hiển thị thông báo (notifications) về các sự kiện mới (tin nhắn mới, lời mời kết bạn).
* Hiển thị trạng thái hoạt động của người dùng (online/offline).
* Hiển thị chỉ báo "đang gõ phím" (typing indicator) để tăng tính tương tác.
  1. Kiến trúc hệ thống
     1. Mô hình tổng thể

Hệ thống được xây dựng dựa trên mô hình kiến trúc Client-Server hiện đại, phân tách rõ ràng vai trò giữa giao diện người dùng (Frontend) và logic xử lý nghiệp vụ (Backend). Phía Client là một ứng dụng đơn trang (SPA) được phát triển bằng ReactJS, chịu trách nhiệm hiển thị giao diện và xử lý tương tác người dùng. Trong khi đó, phía Server là một ứng dụng web mạnh mẽ được xây dựng bằng Java Spring Boot, đảm nhiệm toàn bộ logic nghiệp vụ, xác thực, và quản lý kết nối. Để lưu trữ dữ liệu, hệ thống sử dụng cơ sở dữ liệu quan hệ Microsoft SQL Server. Sự giao tiếp giữa Client và Server được thực hiện qua hai kênh chính: HTTP/RESTful API cho các tác vụ quản lý không yêu cầu tính tức thời (như đăng nhập, quản lý bạn bè), và WebSocket (với STOMP) cho các chức năng yêu cầu tương tác thời gian thực như gửi/nhận tin nhắn và cập nhật trạng thái..



*Hình 3.1: Mô hình tổng thể hệ thống.*

Để hiện thực hóa kiến trúc phân lớp, mã nguồn phía server được tổ chức thành các package rõ ràng, mỗi package đảm nhiệm một vai trò cụ thể:

* Config: Chứa các lớp cấu hình cho ứng dụng, bao gồm WebSocketConfig để thiết lập giao tiếp thời gian thực và SecurityConfig để định nghĩa các quy tắc bảo mật.
* Controller: Là điểm cuối tiếp nhận các yêu cầu HTTP và tin nhắn WebSocket từ client, sau đó điều hướng đến tầng Service.
* DTO (Data Transfer Object): Chứa các đối tượng dùng để truyền dữ liệu giữa các tầng, giúp che giấu cấu trúc của Entity và tăng tính linh hoạt.
* Entity: Bao gồm các lớp thực thể được ánh xạ trực tiếp với các bảng trong cơ sở dữ liệu thông qua JPA.
* Exception: Chứa các lớp xử lý ngoại lệ tùy chỉnh, giúp quản lý lỗi một cách nhất quán.
* Repository: Chứa các interface kế thừa từ JpaRepository, chịu trách nhiệm truy cập và thao tác với cơ sở dữ liệu.
* Service: Chứa toàn bộ logic nghiệp vụ của ứng dụng, điều phối hoạt động giữa Controller và Repository.

Kiến trúc phía Frontend (ReactJS):

* Components: Giao diện người dùng được chia thành các component nhỏ, có chức năng riêng biệt như ChatWindow, MessageList, UserList, MessageInput, v.v.
* API/Services Layer: Một lớp riêng biệt được tạo ra để quản lý việc giao tiếp với backend. Lớp này chứa các hàm gọi RESTful API (sử dụng thư viện như Axios) và quản lý kết nối WebSocket (sử dụng SockJS và Stomp.js).
* State Management: Trạng thái của ứng dụng (thông tin người dùng đăng nhập, danh sách bạn bè, tin nhắn hiện tại) được quản lý bằng các React Hooks (useState, useEffect) và Context API để chia sẻ trạng thái giữa các component.

Thiết kế CSDL: Cơ sở dữ liệu được thiết kế theo mô hình quan hệ với các bảng chính bao gồm Users, Friendships, Chat\_Rooms, Room\_Members, Messages, và Notifications. Cấu trúc này đảm bảo dữ liệu được lưu trữ một cách có tổ chức, nhất quán và dễ dàng truy vấn.

|  |  |  |  |
| --- | --- | --- | --- |
| Thành phần | Vai trò | Giao tiếp | Công nghệ chính |
| Client (Frontend) | Giao diện người dùng, gửi request và nhận phản hồi | HTTP/HTTPS, WebSocket | HTML, CSS, JavaScript |
| REST API (Backend) | Xử lý logic nghiệp vụ, xác thực, phân quyền, giao tiếp DB và storage | HTTP/HTTPS, TCP | Java Spring Boot, JPA/Hibernate |
| Database (SQL Server) | Lưu trữ metadata (Users, Files, Folders, Logs) | TCP | Microsoft SQL Server |
| WebSocket Broker | Gửi và nhận thông báo thời gian thực giữa server và client | WebSocket/STOMP | Spring WebSocket, SockJS |

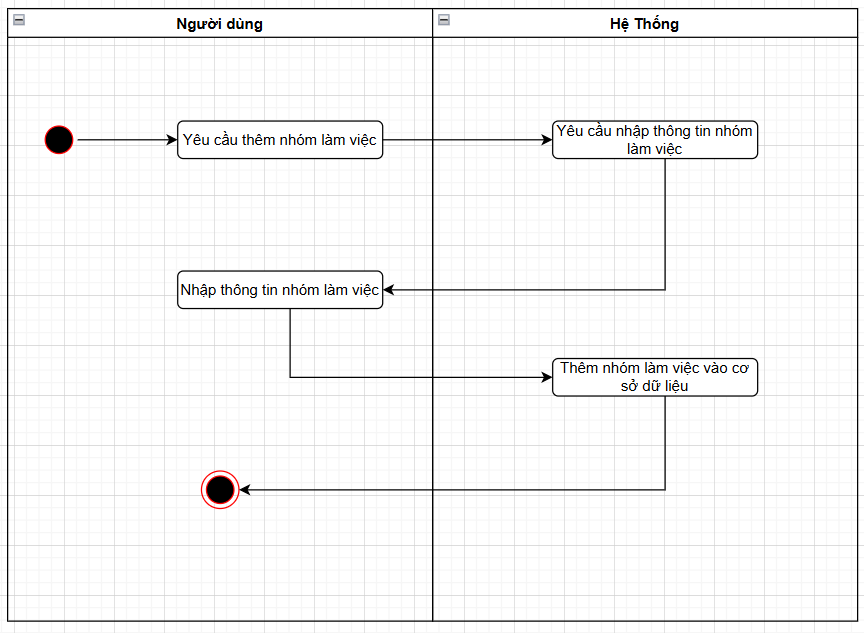
*Bảng 3.1: Các thành phần chính của hệ thống.*

# CHƯƠNG 4: SƠ ĐỒ HOẠT ĐỘNG (ACTIVITY DIAGRAM)

## Sơ đồ đăng nhập

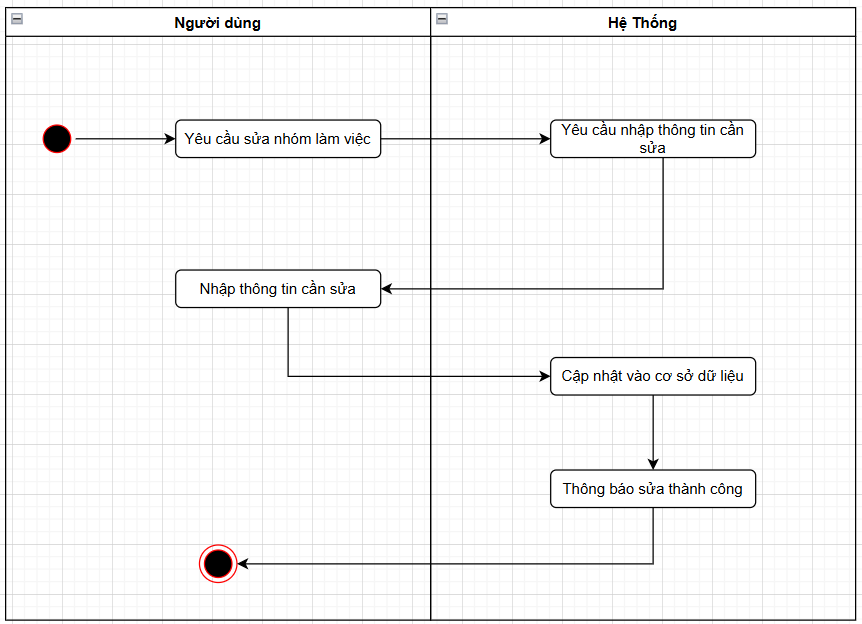
*Hình 4.1: Mô hình hoạt động đăng nhập.*

## Sơ đồ hoạt động tạo nhóm làm việc



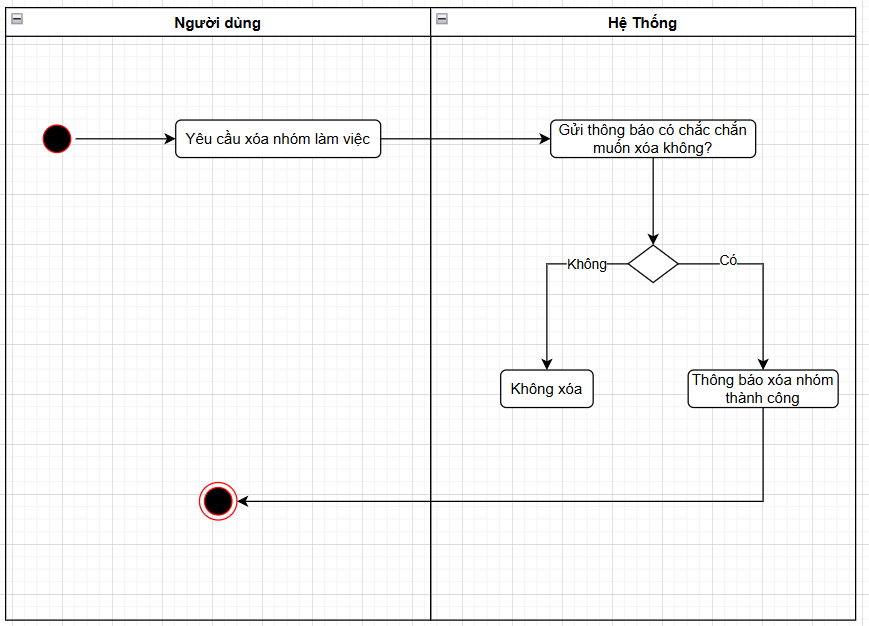
*Hình 4.2: Mô hình hoạt động tạo nhóm làm việc.*

## Sơ đồ hoạt động sửa nhóm làm việc



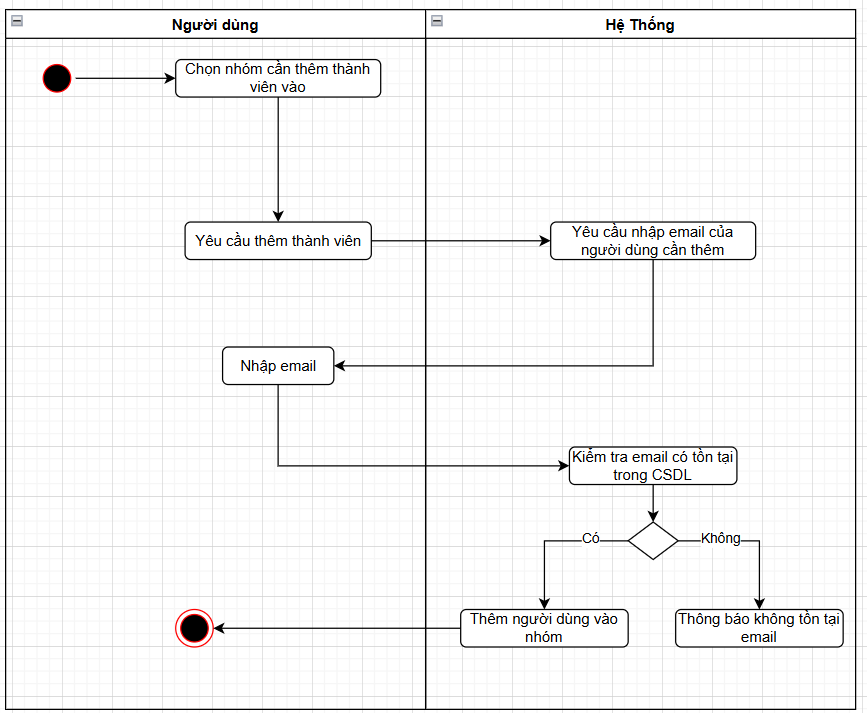
*Hình 4.3: Mô hình hoạt động sửa nhóm làm việc.*

## Sơ đồ hoạt động xóa nhóm làm việc



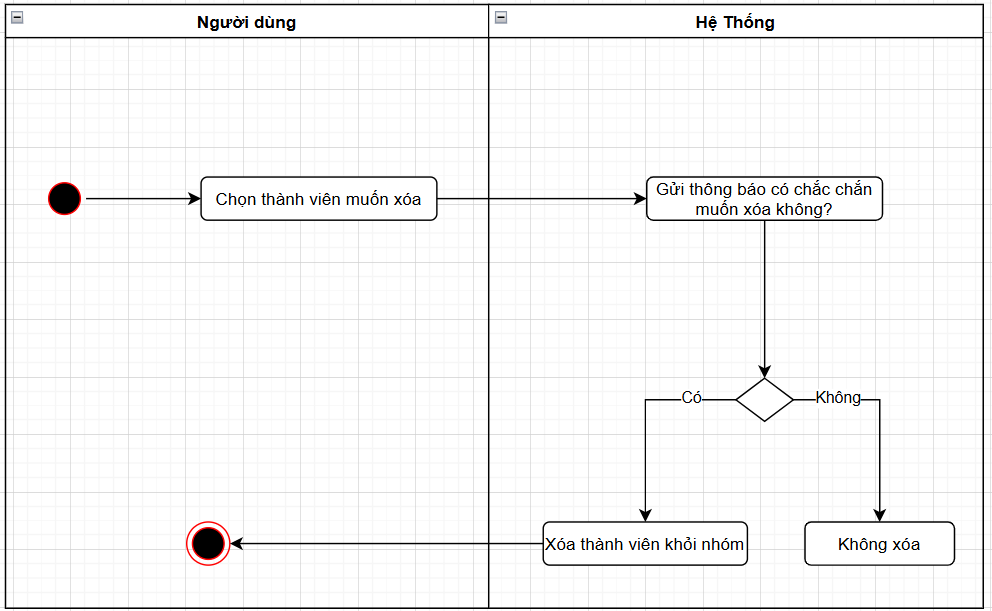
*Hình 4.4: Mô hình hoạt động xóa nhóm làm việc.*

## Sơ đồ hoạt động thêm thành viên vào nhóm



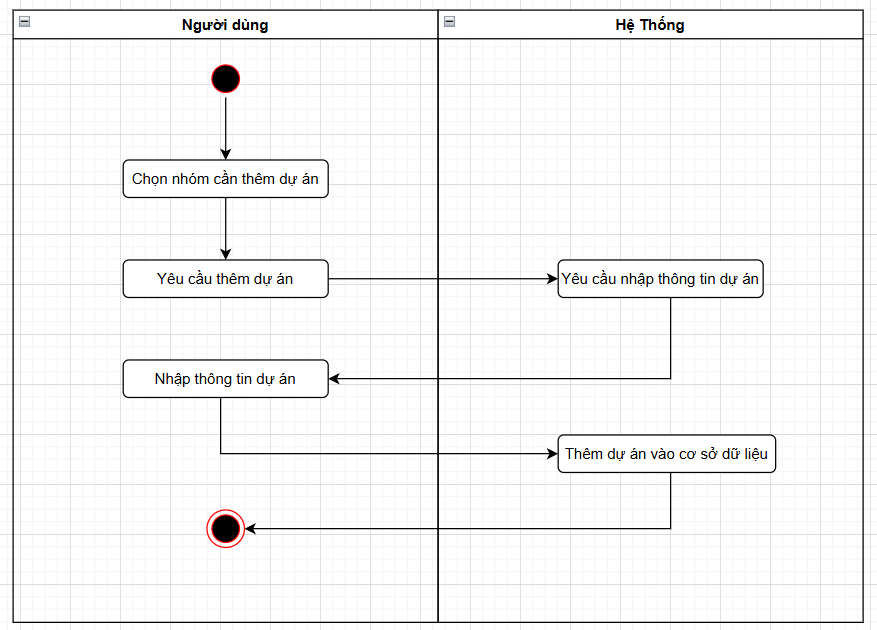
*Hình 4.5: Mô hình hoạt động thêm thành viên vào nhóm.*

## Sơ đồ hoạt động xóa thành viên khỏi nhóm



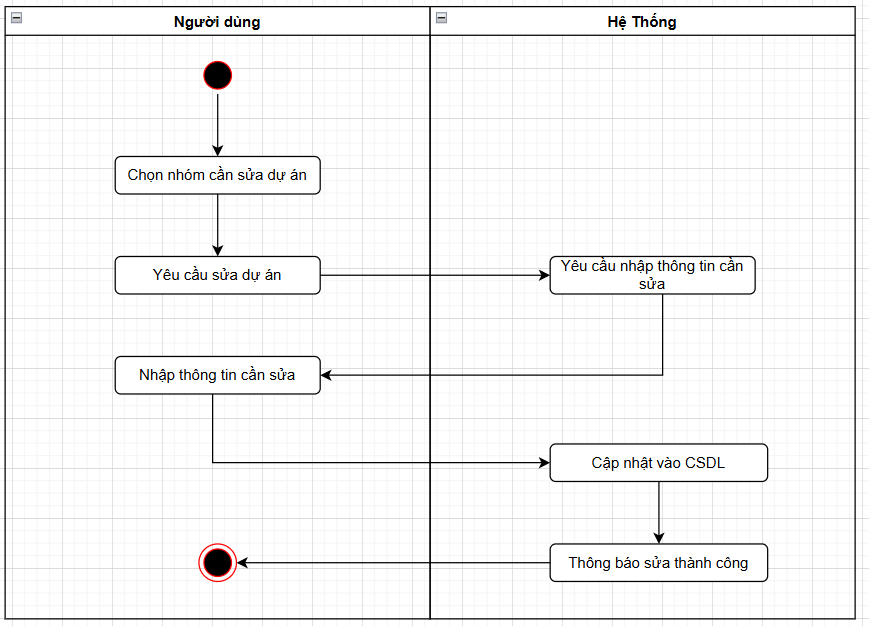
*Hình 4.6: Mô hình hoạt động xóa thành viên khỏi nhóm.*

## Sơ đồ hoạt động tạo dự án



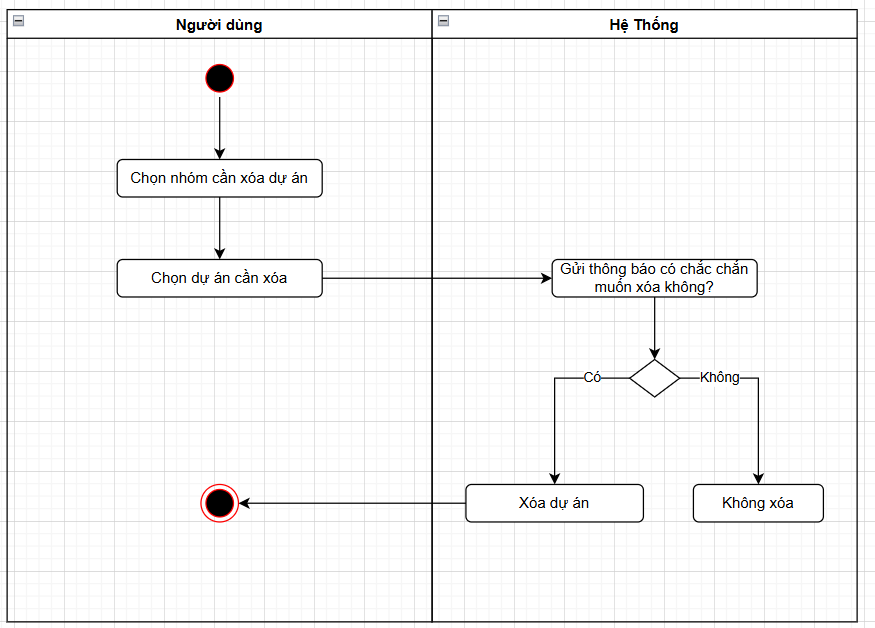
*Hình 4.7: Mô hình hoạt động tạo dự án.*

## Sơ đồ hoạt động sửa dự án



*Hình 4.8: Mô hình hoạt động sửa dự án.*

## Sơ đồ hoạt động xóa dự án

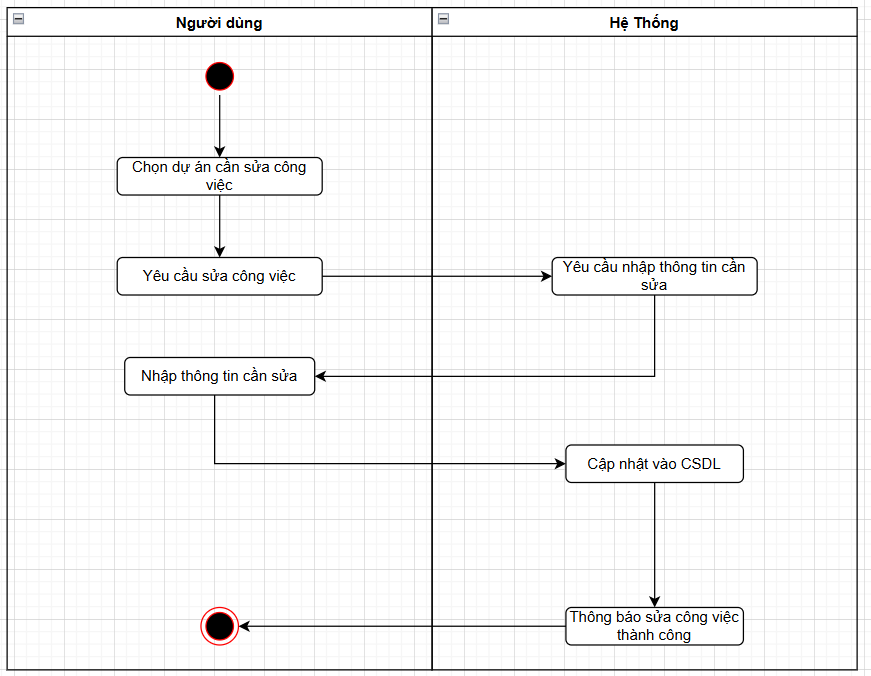


*Hình 4.9: Mô hình hoạt động xóa dự án.*

## Sơ đồ hoạt động thêm công việc

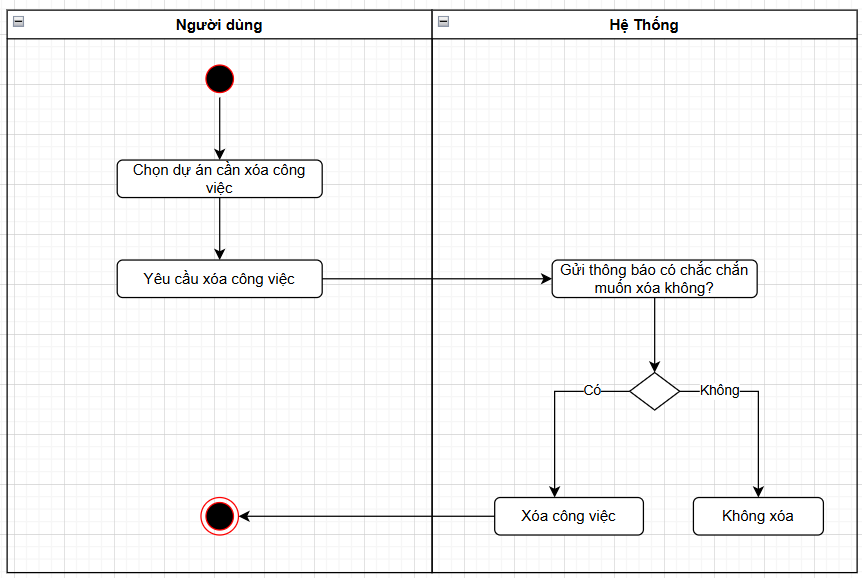
*Hình 4.10: Mô hình hoạt động thêm công việc.*

## Sơ đồ hoạt động sửa công việc



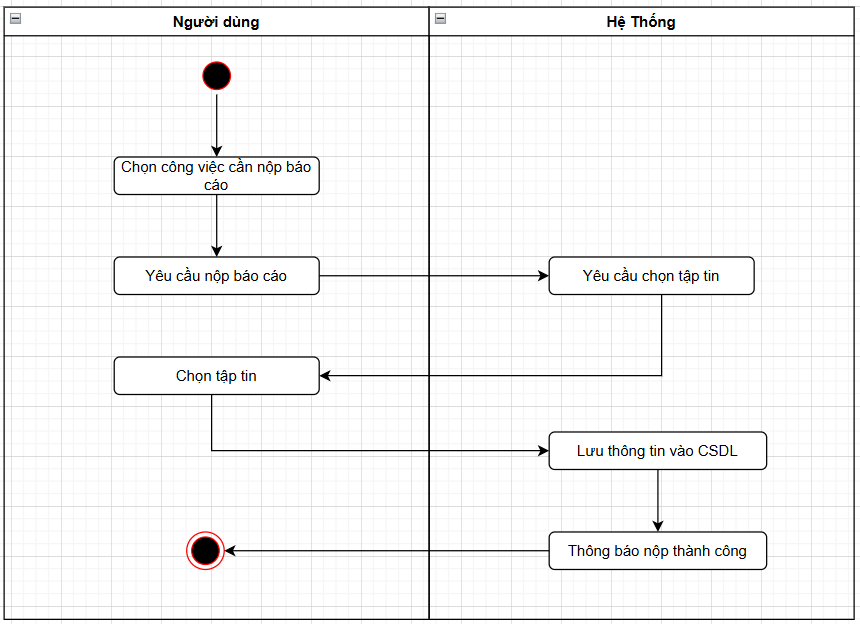
*Hình 4.11: Mô hình hoạt động sửa công việc.*

## Sơ đồ hoạt động xóa công việc



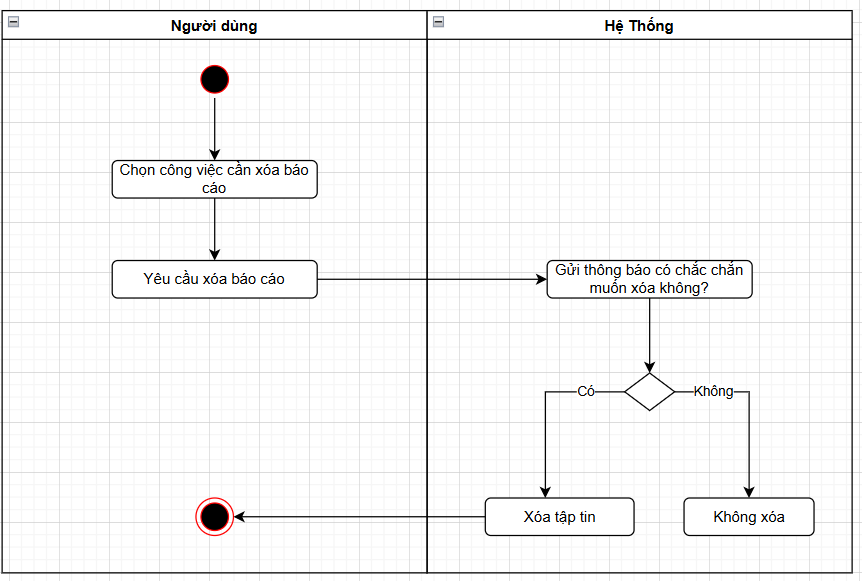
*Hình 4.12: Mô hình hoạt động xóa công việc.*

## Sơ đồ hoạt động nộp báo cáo



*Hình 4.13: Mô hình hoạt động xóa công việc.*

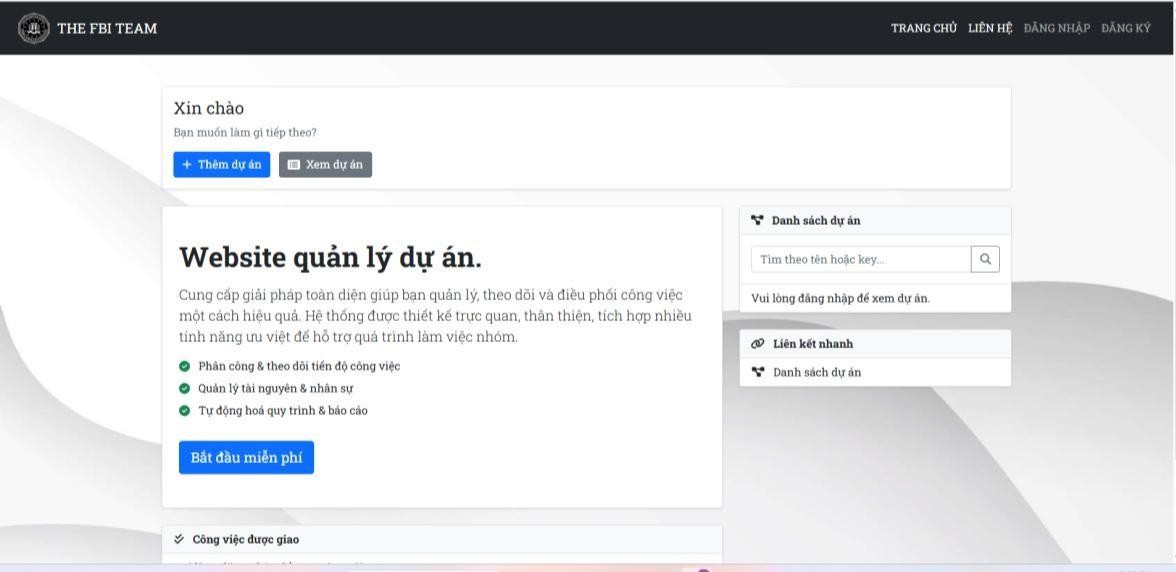
## Sơ đồ hoạt động xóa báo cáo



*Hình 4.14: Mô hình hoạt động xóa báo cáo.*

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN

## Giao diện trang chủ

****

*Hình 5.1: Giao diện trang chủ.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Nút thêm dự  án | Tạo ra một công việc mới | Màu  xanh | Thể hiện đây là hành động  chính cần thực hiện. |
| 2 | Nút xem dự án | Xem danh sách dự án của người dùng | Màu xám | Thể hiện một hành động phụ hoặc ít quan trọng hơn so  với nút chính. |
| 3 | Nút bắt đầu  miễn phí | Chuyển đến trang đăng  nhập để bắt đầu sử dụng | Màu  xanh | Thể hiện đây là hành động  chính cần thực hiện. |

*Bảng 5.1: Ý nghĩa giao diện trang chủ.*

## Giao diện đăng nhập

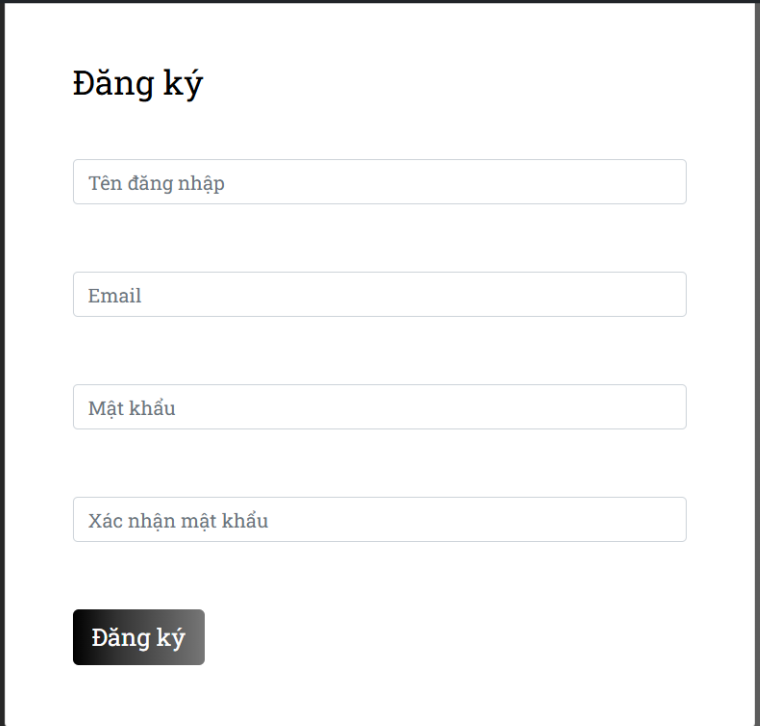
****

*Hình 5.2: Giao diện đăng nhập.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Nhập tài khoản | Nhập email đã đăng kí | - | - |
| 2 | Mật khẩu | Nhập mật khẩu đã đăng kí | - | - |
| 3 | Tùy chọn lưu tài khoản  cho lần đăng kí sau | Nhớ tài khoản cho lần  đăng nhập sau | - | - |
| 4 | Nút đăng nhập | Đăng nhập vào tài khoản | Màu đen | Tạo sự mạnh mẽ, chuyên  nghiệp, đáng tin cậy, nổi bật. |
| 5 | Nút quên mật khẩu | Lấy lại mật khẩu |  |  |
| 6 | Nút Đăng ký | Chuyển qua form đăng kí cho ai chưa có tài  khoản | Màu đỏ | Biểu thị sự khẩn cấp, cảnh báo hoặc thúc đẩy  hành động. |

*Bảng 5.2: Mô tả giao diện đăng nhập.*

## Giao diện đăng ký

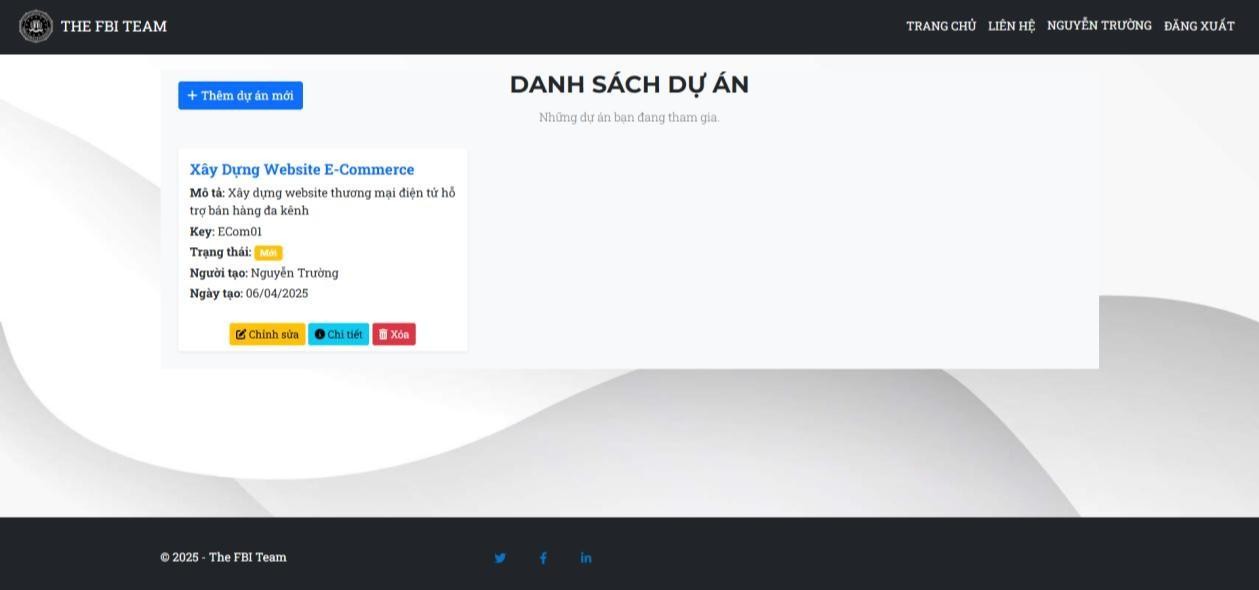
****

*Hình 5.3: Giao diện đăng ký.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tên đăng nhập | Nhập tên đăng nhập | - | - |
| 2 | Email | Nhập email | - | - |
| 3 | Mật khẩu | Nhập mật khẩu | - | - |
| 4 | Xác nhận mật  khẩu | Nhập lại mật khẩu  đăng nhập | - | - |
| 5 | Nút đăng kí | Đăng kí tài khoản  mới | Màu đen | Tạo độ tương phản cao so với nền  trắng của biểu mẫu đăng ký. |

*Bảng 5.3: Mô tả giao diện đăng ký.*

## Giao diện danh sách dự án

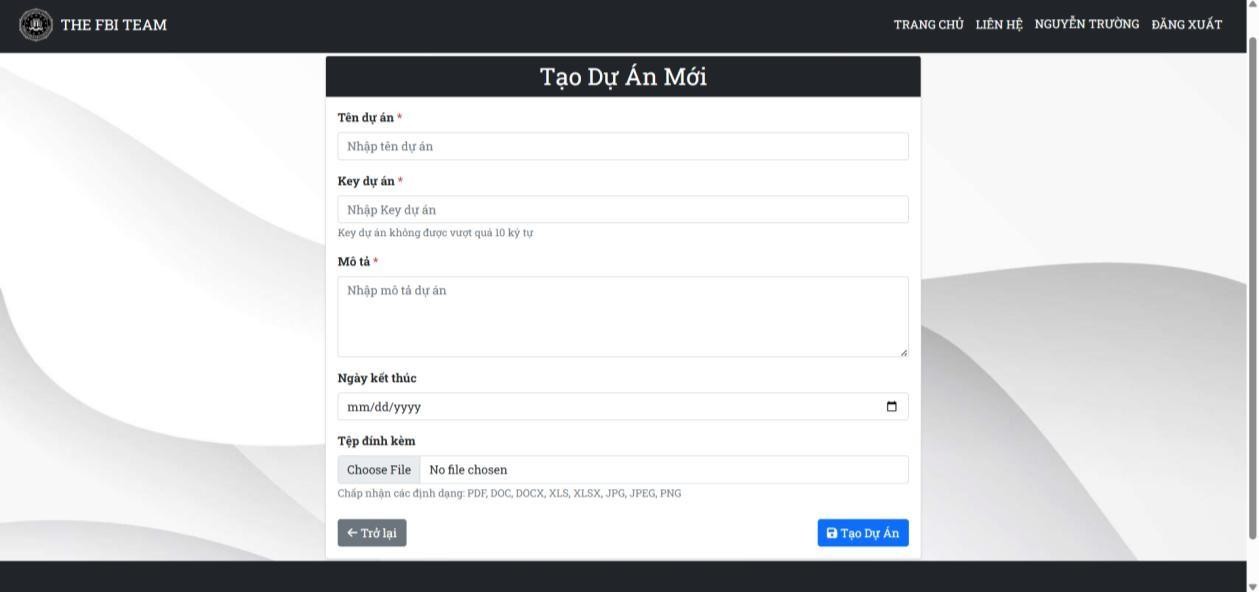
****

*Hình 5.4: Giao diện danh sách dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Thêm dự án | Tạo ra một dự án mới | - | - |
| 2 | Nút Chỉnh sửa | Sửa lại dự án đã tạo | Màu vàng | Liên quan đến sự cảnh báo nhẹ  hoặc thông báo cần chú ý. |
| 3 | Nút Chi tiết | Hiện thêm thông tin về dự án đã chọn | Màu xanh | Thể hiện sự tin cậy và an toàn. |
| 4 | Nút xóa | Xóa dự án đã chọn | Màu đỏ | Mang ý nghĩa cảnh báo, nguy hiểm, không thể hoàn tác. |

*Bảng 5.4: Ý nghĩa giao diện danh sách dự án.*

## Giao diện tạo dự án

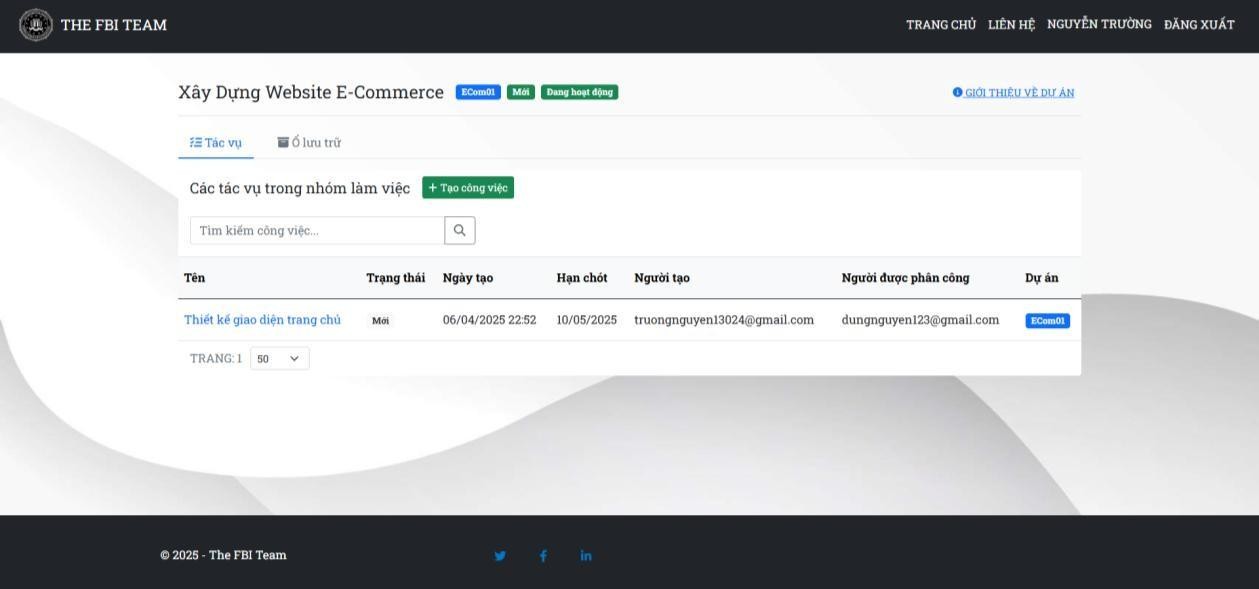
****

*Hình 5.5: Giao diện tạo dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tên dự án | Nhập tên cho dự án | - | - |
| 2 | Mô tả | Nhập mô tả cho dự án | - |  |
| 3 | Key | Từ khóa tìm kiếm cho dự án | - | - |
| 4 | Nút Tạo dự án | Tạo ra dự án và lưu vào CSDL | Màu xanh | Thể hiện đây là hành động chính cần thực  hiện. |
| 5 | Nút trở lại | Quay về trang chủ | Màu xám | Thể hiện một hành động  phụ hoặc ít quan trọng hơn so với nút chính. |

*Bảng 5.5: Mô tả giao diện tạo dự án.*

## Giao diện chi tiết dự án

****

*Hình 5.6: Giao diện chi tiết dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác / Thành phần** | **Ý nghĩa / Chức năng** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Xây Dựng  Website E- Commerce | Hiển thị tên của dự án hiện tại. | Chữ đen, nền trắng | Giao diện chính thống, dễ đọc, dễ nhận diện |
| 2 | Mới, Đang  hoạt động | Trạng thái hiện tại của  dự án. | Xanh lá | dự án đang triển khai thực  tế |
| 3 | Ô tìm kiếm công việc | Tìm kiếm công việc trong dự án theo từ  khóa. | - | - |
| 4 | Nút Tạo công  việc | Giao việc cho thành  viên trong dự án. | Xanh lá | biểu thị hành động tạo  mới, tích cực |
| 5 | Bảng danh  sách công việc | Hiển thị toàn bộ công việc thuộc dự án | Bảng trắng đen với viền | Phân biệt rõ ràng các dòng công việc, dễ theo dõi |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Giới thiệu dự  án | Chuyển hướng đến  trang mô tả dự án | Xanh dương | đây là liên kết, có thể nhấn  vào để xem thêm |

*Bảng 5.6: Mô tả giao diện chi tiết dự án.*

## Giao diện chỉnh sửa dự án

****

*Hình 5.7: Giao diện chỉnh sửa dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tên dự án | Tên của dự án cần sửa | - | - |
| 2 | Mô tả | Mô tả của dự án | - | - |
| 3 | Key | Từ khóa tìm kiếm của dự án | - | - |
| 4 | Trạng thái | Dự án còn hoạt động hay không | - | - |
| 5 | Người tạo | Người đã tạo ra dự án | - | - |
| 6 | Ngày tạo | Thời điểm dự án được tạo ra | - | - |
| 7 | Nút Lưu | Lưu thông tin mới vào dự án và  CSDL | Màu xanh | Thể hiện đây là hành  động chính cần thực hiện. |
| 8 | Nút quay lại | Trở về chi tiết dự án | Màu xám | Thể hiện một hành động  phụ. |

*Bảng 5.7: Ý nghĩa giao diện chỉnh sửa dự án.*

## Giao diện thêm thành viên vào dự án

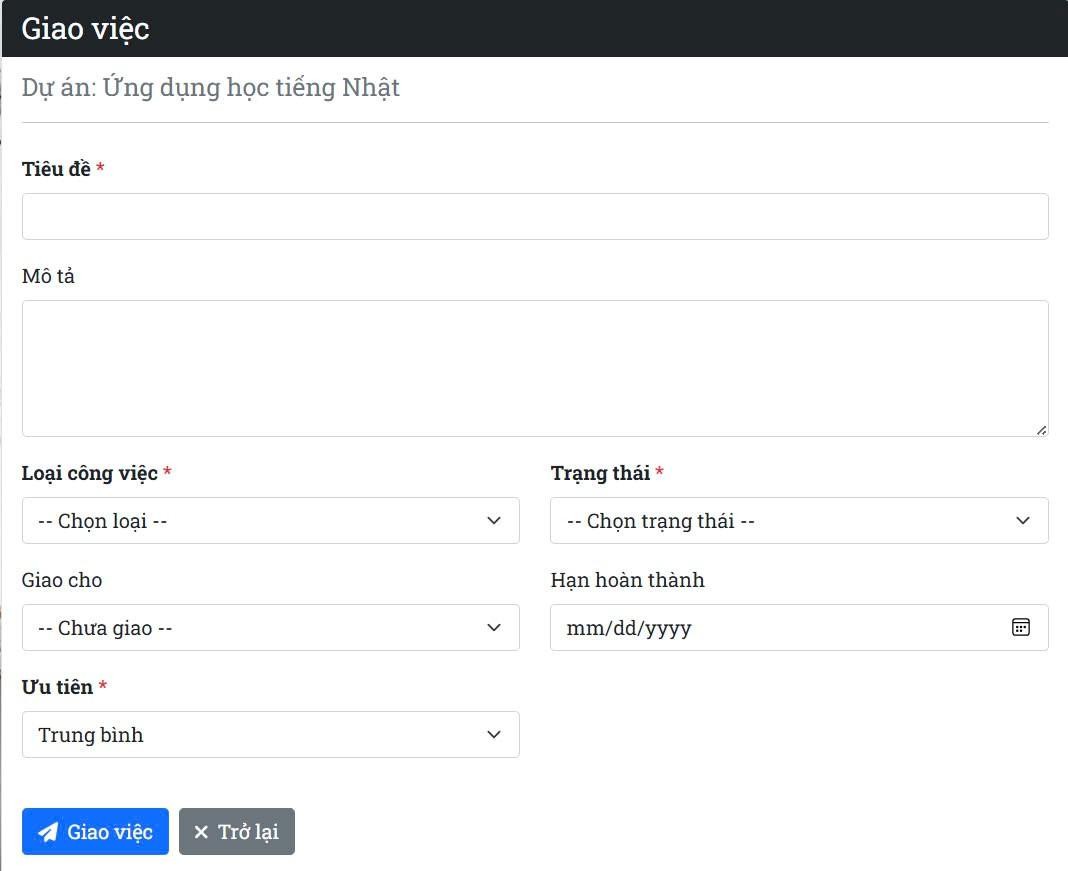


*Hình 5.8: Giao diện thêm thành viên vào dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Thêm người dùng | Thêm người dùng khác vào dự án của bạn | - | - |
| 2 | Nút thêm thành  viên | Lưu người dùng vào dự  án và CSDL | Màu xanh | Thể hiện đây là hành động  chính cần thực hiện. |
| 3 | Nút trở lại | Trở về chi tiết dự án | Màu xám | Thể hiện một hành động phụ. |

*Bảng 5.8: Ý nghĩa giao diện thêm thành dự án.*

## Giao diện giao việc cho thành viên trong dự án

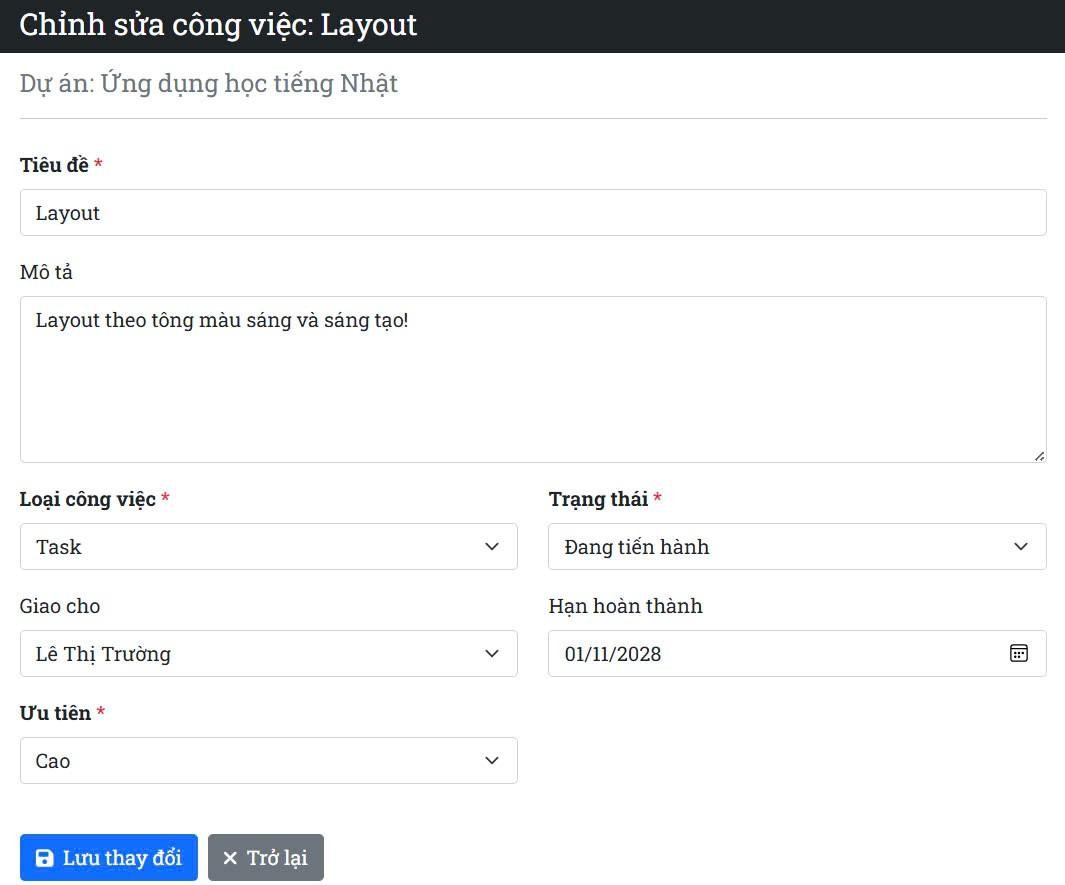
****

*Hình 5.9: Giao diện giao việc cho thành viên trong dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tiêu đề | Nhập tên cho công việc được tạo | - | - |
| 2 | Mô tả | Nhập mô tả cho công việc | - | - |
| 3 | Chọn trang  thái | Dự án có còn hoạt động  hay không | - | - |
| 4 | Giao cho | Chọn thành viên trong dự  án để giao việc | - | - |
| 5 | Hạn hoàn  thành | Chọn ngày kết thúc công  việc | - | - |
| 6 | Ưu tiên | Mức độ ưu tiên của công việc | - | - |
| 7 | Nút giao  việc | Lưu và quay trở về chi  tiết dự án | Màu xanh | Thể hiện đây là hành động  chính cần thực hiện. |
| 8 | Nút trở lại | Trở lại chi tiết dự án | Màu xám | Thể hiện một hành động phụ. |

*Bảng 5.9: Ý nghĩa giao diện giao việc cho thành viên trong dự án.*

## Giao diện chỉnh sửa công việc

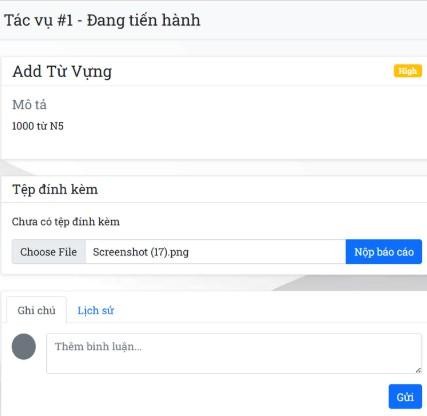
****

*Hình 5.10: Giao diện chỉnh sửa công việc.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tiêu đề | Tiêu đề của công việc cần sửa | - | - |
| 2 | Mô tả | Mô tả của công việc | - | - |
| 3 | Trạng thái | Dự án còn hoạt động hay không | - | - |
| 4 | Mức độ  ưu tiên | Mức độ ưu tiên của công việc | - | - |
| 6 | Giao cho | Chọn thành viên để thêm giao việc | - | - |
| 7 | Hạn hoàn  thành | Chọn ngày kết thúc công việc | - | - |
| 8 | Nút Lưu | Lưu vào danh sách và cập nhật  CSDL | Màu xanh | Thể hiện đây là hành động  chính cần thực hiện. |
| 9 | Nút hủy | Hủy hành động chỉnh sửa công việc | Màu xám | Thể hiện một hành động phụ. |

*Bảng 5.10: Ý nghĩa giao diện chỉnh sửa công việc.*

## Giao diện nộp báo cáo

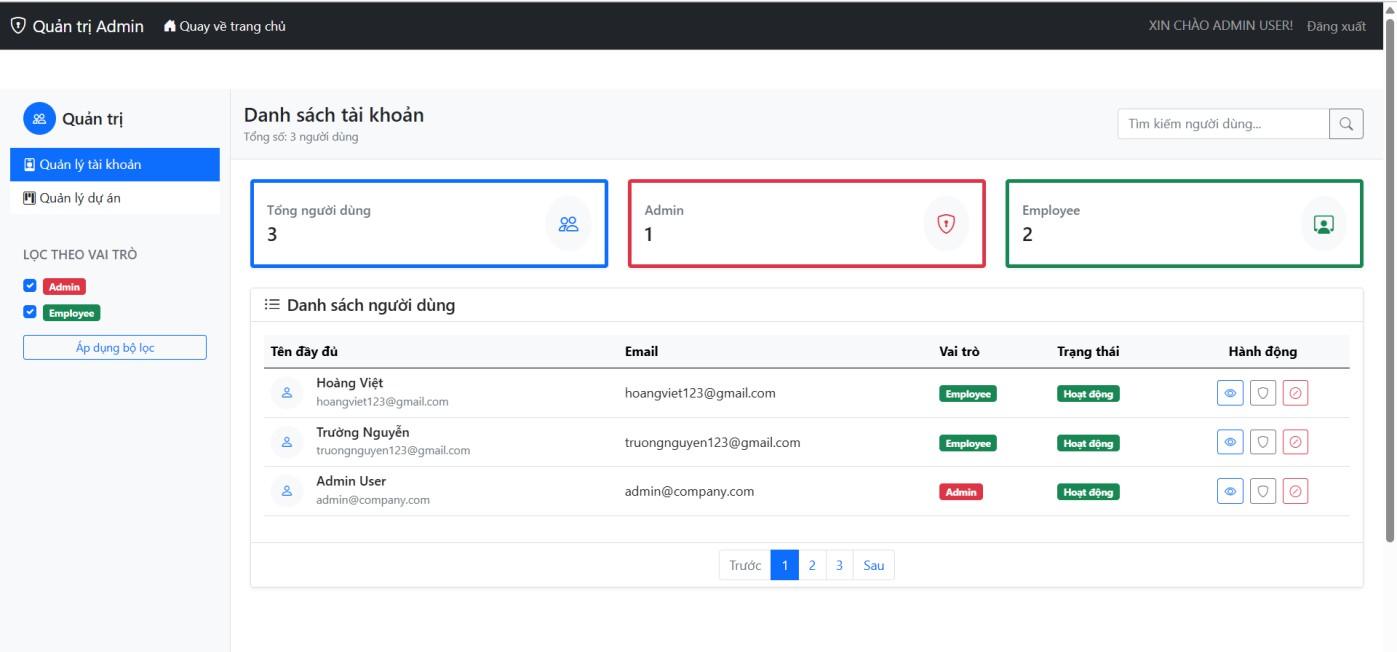
****

*Hình 5.11: Giao diện nộp báo cáo.*

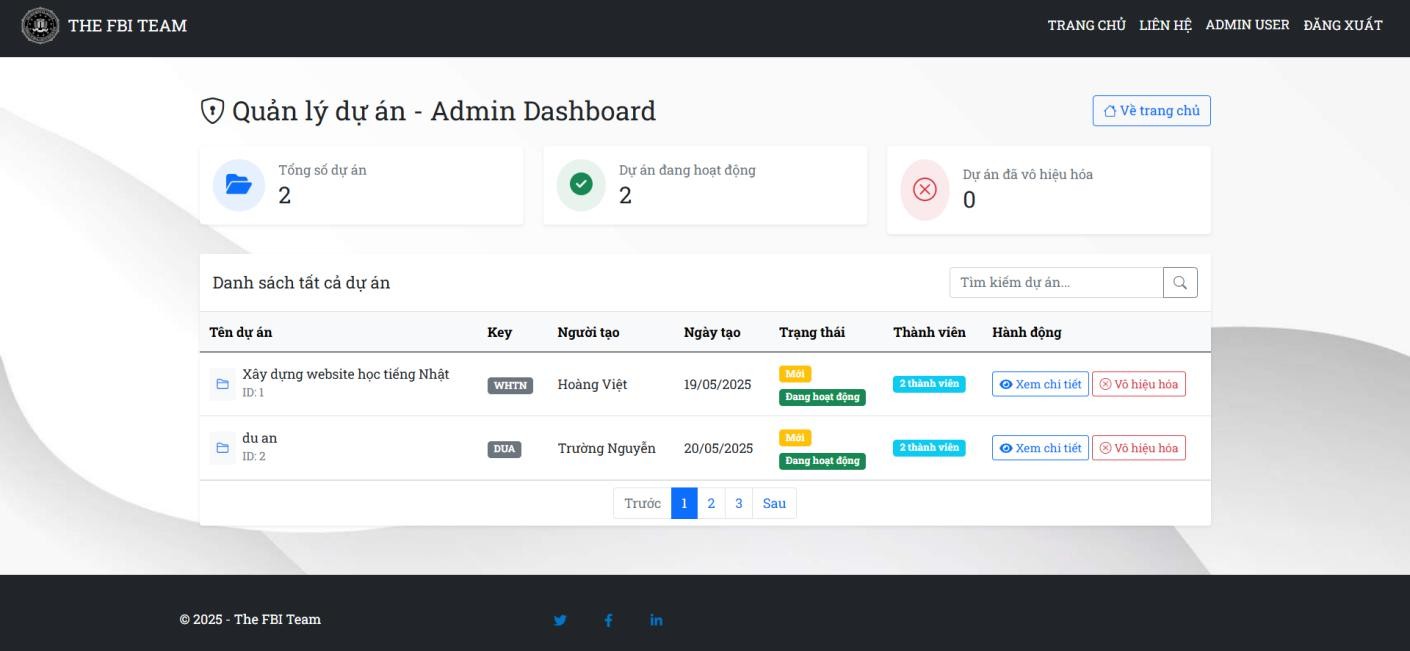
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Công việc | Tiêu đề của công việc cần sửa | - | - |
| 2 | Chọn file | Thêm file cần báo cáo | - | - |
| 3 | Thời gian nộp/cập nhật lúc | Thời gian nộp hay chỉnh sửa báo cáo | - | - |
| 4 | Nộp báo cáo/ Lưu chỉnh sửa | Gửi báo cáo cho trưởng dự án/ công  việc | Màu xanh | Thể hiện đây là hành động chính cần thực hiện. |
| 6 | Nút trở lại / hủy | Hủy bỏ thao tác | Màu xám | Thể hiện một hành động phụ. |

*Bảng 5.11: Ý nghĩa giao diện nộp báo cáo.*

## Giao diện chức năng vô hiệu hóa tài khoản và dự án cho người dùng admin

****

*Hình 5.12: Giao diện chức năng vô hiệu hóa tài khoản cho người dùng admin.*

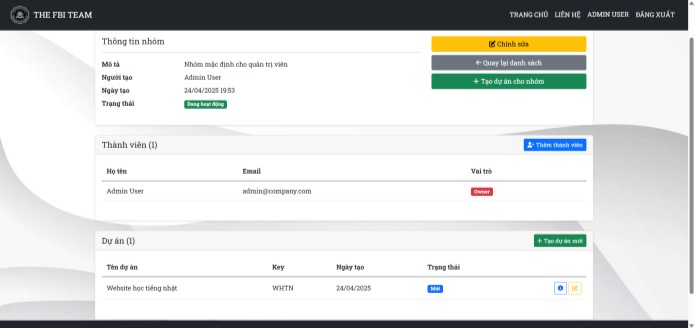
**

*Hình 5.13: Giao diện chức năng vô hiệu hóa dự án cho người dùng admin.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tổng số người dùng | Hiển thị số lượng tổng tài khoản | Xanh dương | Thể hiện thông tin chung và tổng quan |
| 2 | Admin | Hiển thị số lượng tài khoản quản trị  viên | Đỏ | Vai trò quan trọng |
| 3 | Employee | Hiển thị số  lượng tài khoản nhân viên | Xanh lá | Vai trò nhân viên |
| 4 | Danh sách người dùng | Quản lý thông tin  chi tiết của từng tài khoản | - | - |
| 6 | Bộ lọc vai trò | Lọc theo vai trò | - | - |
| 7 | Tổng số dự án | Hiển thị tổng số dự án | - | - |
| 8 | Dự án đang hoạt động | Hiển thị số lượng dự án đang hoạt  động | - | - |
| 9 | Dự án đã vô hiệu hóa | Hiển thị số lượng dự án đã vô hiệu hóa | - | - |
| 10 | Danh sách tất cả dự án | Tìm kiếm, xem chi tiết, vô hiệu  hóa dự án | - | - |
| 11 | Nút phân trang | Chuyển đổi giữa các danh sách dự án | - | - |
| 12 | Nút xem chi tiết | Xem chi tiết dự án | Xanh dương | Thể hiện chi tiết |
| 13 | Nút vô hiệu hóa | Vô hiệu hóa dự án | Đỏ | Thể hiện hành động quan  trọng |

*Bảng 5.12: Ý nghĩa giao diện chức năng vô hiệu hóa tài khoản và dự án cho người dùng admin.*

## Giao diện chức năng quản lý nhóm làm việc

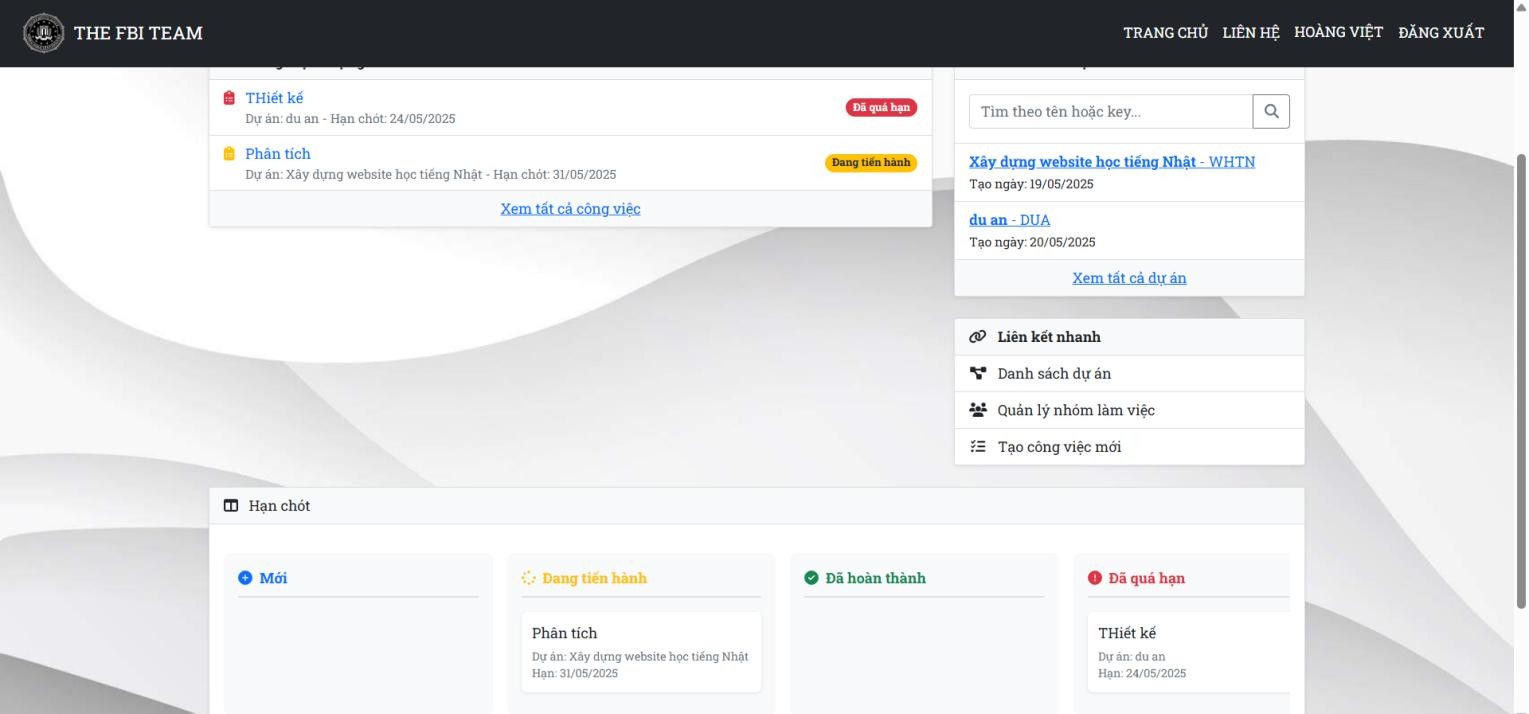


*Hình 5.14: Giao diện chức năng quản lý nhóm làm việc.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Tạo nhóm mới | Cho phép thêm  mới một nhóm làm việc | Xanh dương | Tạo mới, thể hiện hành động tích cực |
| 2 | Nhóm quản trị | Thể hiện thông tin nhóm quản trị | - | - |
| 3 | Nút chi tiết | Nhấn để xem chi tiết nhóm | Xanh dương | Thể hiện thao tác điều hướng |
| 4 | Nút xóa | Nhấn để xóa nhóm | Đỏ | Thể hiện hành động không thể hoàn tác. |
| 5 | Nút chỉnh sửa | Nhấn để chỉnh sửa thông tin nhóm | Vàng | Thể hiện thao tác điều  hướng |
| 6 | Nút quay lại danh sách | Nhấn để quay lại danh sách | Xám | Thể hiện thao tác điều  hướng |
| 7 | Tạo dự án cho nhóm/Tạo dự án  mới | Nhấn để tạo dự án mới cho nhóm | Xanh lá | Thể hiện hành động tích cực |
| 8 | Nút thêm thành viên | Nhấn để thêm  thành viên mới vào nhóm | Xanh biển | Thể hiện thao tác điều  hướng |

*Bảng 5.13: Ý nghĩa giao diện chức năng quản lý nhóm làm việc*

## Giao diện chức năng kéo thả thay đổi trạng thái công việc

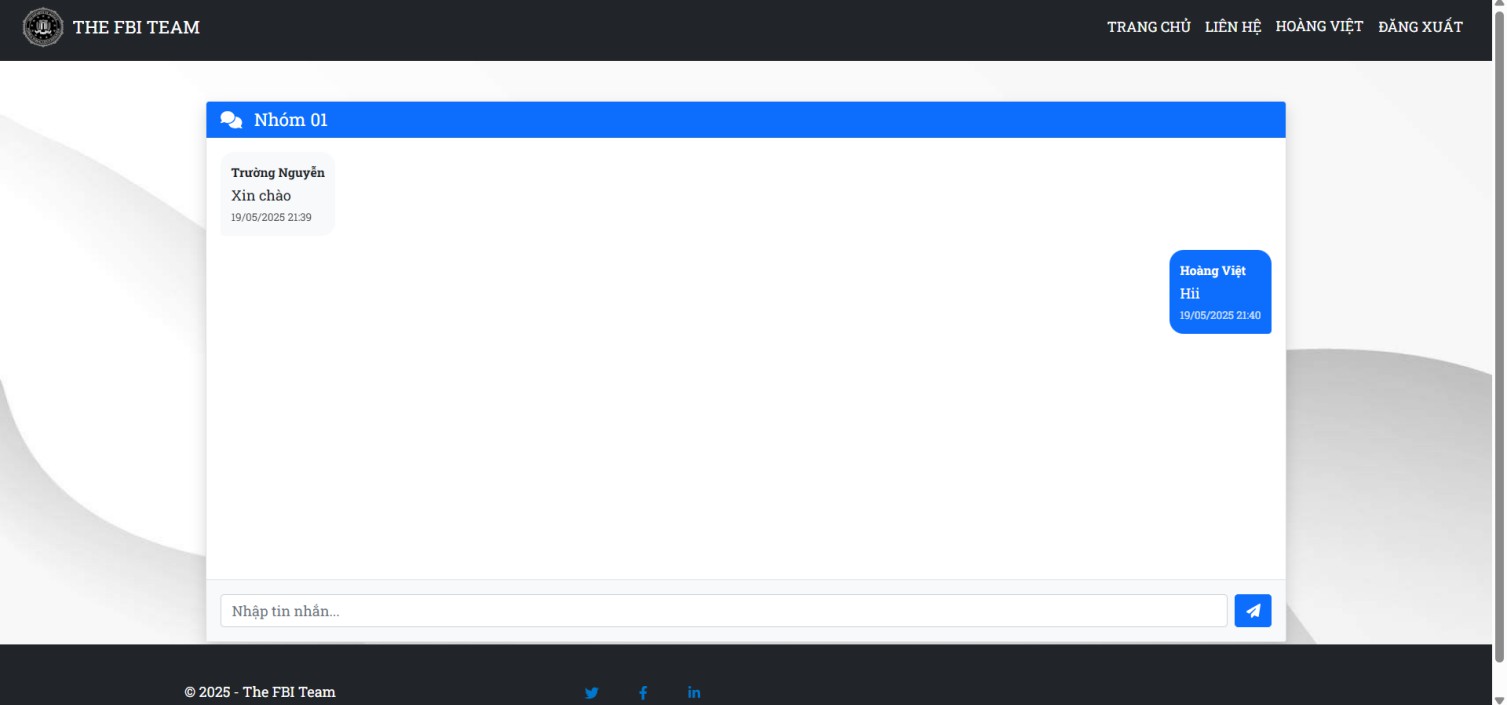


*Hình 5.15: Giao diện chức năng kéo thả thay đổi trạng thái công việc.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | "Mới" (trong khối "Hạn chót") | Nhãn trạng thái công việc | Xanh dương | Đang chờ xử lý, hoặc có thể là đang thực hiện |
| 2 | "Đang tiến hành" (trong khối "Hạn  chót") | Nhãn trạng thái công việc | Xanh dương | Đang thực hiện, cần chú ý |
| 3 | "Đã hoàn thành" (trong khối "Hạn chót") | Nhãn trạng thái công việc | Xanh dương | Hoàn thành, thành công |
| 4 | "Đã quá hạn" (trong khối "Hạn chót") | Nhãn trạng thái công việc | Xanh dương | Quá hạn, cần hành động khẩn cấp, cảnh báo |
| 6 | Thanh cuộn ngang | Cho phép người dùng cuộn để xem thêm các mục nếu danh sách  dài. | - | - |

*Bảng 5.14: Ý nghĩa giao diện chức năng kéo thả thay đổi trạng thái*

## Giao diện chức năng trò chuyện giữa các thành viên trong nhóm



*Hình 5.16: Giao diện chức năng trò chuyện giữa các thành viên trong nhóm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Ô nhập liệu tin nhắn "Nhập tin nhắn..." | Vùng để người dùng nhập nội dung tin nhắn  muốn gửi | - | - |
| 2 | Nút gửi tin nhắn | Gửi tin nhắn đã nhập | Xanh dương | Màu xanh dương và icon trực quan cho hành động  gửi, tạo cảm giác thân thiện, dễ sử dụng |
| 3 | Nội dung tin nhắn bên trái | Hiển thị tin nhắn của  người nhận | Xám | Thể hiện phân biệt tin nhắn của mình |
| 4 | Nội dung tin nhắn bên phải | Hiển thị tin nhắn của người  gửi | Xanh dương | Thể hiện người gửi |

*Bảng 5.15: Ý nghĩa giao diện chức năng trò chuyện*

# CHƯƠNG 6: HƯỚNG DẪN SỬ DỤNG

## Truy cập Hệ thống

**Đăng ký tài khoản:** Truy cập trang đăng ký. Nhập đầy đủ thông tin: Họ tên, Email, Mật khẩu, Xác nhận mật khẩu. Nhấn nút "Đăng kí". Hệ thống sẽ lưu thông tin tài khoản của bạn.

**Đăng nhập:** Truy cập trang đăng nhập. Nhập Email và Mật khẩu đã đăng ký. (Tùy chọn) Chọn ô "Lưu tài khoản" để ghi nhớ thông tin cho lần đăng nhập sau. Nhấn nút "Đăng nhập". Nếu thông tin chính xác, bạn sẽ được chuyển vào trang chính của hệ thống. Nếu nhập sai, hệ thống sẽ báo lỗi và yêu cầu nhập lại. Nếu quên mật khẩu, nhấn vào liên kết "Quên mật khẩu" (chức năng này có thể chưa hoàn thiện).

## Quản lý Tài khoản

Thay đổi thông tin cá nhân: Sau khi đăng nhập, tìm đến mục quản lý tài khoản (thường ở góc giao diện). Hệ thống hiển thị thông tin tài khoản hiện tại. Thay đổi các thông tin cá nhân cần thiết. Lưu lại thay đổi. Hệ thống sẽ cập nhật thông tin vào cơ sở dữ liệu.

## Quản lý Dự án

**Xem danh sách dự án (Trang chủ):** Sau khi đăng nhập, bạn sẽ thấy trang chủ hiển thị danh sách các dự án bạn tham gia hoặc quản lý. Mỗi dự án sẽ có các nút chức năng như: Chi tiết, Chỉnh sửa, Xóa. Có nút "Thêm dự án" để tạo dự án mới.

**Tạo dự án mới:** Trên trang chủ, nhấn nút "Thêm dự án". Điền các thông tin: Tên dự án, Mô tả, Key (Từ khóa tìm kiếm dự án). Nhấn nút "Tạo dự án". Dự án mới sẽ được lưu và hiển thị trên trang chủ.

**Xem chi tiết dự án:** Trên trang chủ, nhấn nút "Chi tiết" của dự án muốn xem. Trang chi tiết sẽ hiển thị thông tin đầy đủ của dự án và danh sách các công việc thuộc dự án đó. Tại đây có các chức năng liên quan đến công việc và dự án như: Thêm công việc, Lọc/Tìm kiếm công việc, Chỉnh sửa dự án,...

**Chỉnh sửa thông tin dự án:** Từ trang chủ, nhấn nút "Chỉnh sửa" của dự án. HOẶC từ trang chi tiết dự án, nhấn nút "Chỉnh sửa". Thay đổi các thông tin cần thiết:

Tên dự án, mô tả, Key, Trạng thái (Hoạt động/Không hoạt động). Nhấn nút "Lưu" để cập nhật thay đổi.

**Xóa dự án:** Trên trang chủ, nhấn nút "Xóa" của dự án muốn xóa.

**Lưu ý:** Theo mô tả, chỉ người tạo dự án mới có quyền xóa dự án đó.

**Thêm thành viên vào dự án:** Truy cập giao diện thêm thành viên (có thể nằm trong trang chi tiết dự án hoặc có nút riêng). Nhập thông tin người dùng cần thêm (ví dụ: ID hoặc Tên đăng nhập). Nhấn nút "Thêm". Người dùng sẽ được thêm vào dự án.

**Lưu ý:** Chỉ người tạo dự án có quyền thêm/xóa thành viên.

## Quản lý Công việc

**Thêm công việc mới vào dự án:** Vào trang "Chi tiết dự án". Nhấn nút "Thêm công việc". Điền thông tin công việc: Tiêu đề, Mô tả. Chọn Trạng thái công việc. Chọn thành viên để "Giao cho". Chọn "Hạn hoàn thành". Chọn mức độ "Ưu tiên" (Cao, Trung bình, Thấp). Nhấn nút "Giao" (hoặc "Tạo công việc"). Công việc sẽ được tạo và gán cho thành viên được chọn.

**Xem chi tiết công việc:** Trong trang "Chi tiết dự án", danh sách công việc được hiển thị.

**Chỉnh sửa công việc:** Trong trang "Chi tiết dự án", tìm đến công việc cần sửa và nhấn nút "Chỉnh sửa". Thay đổi các thông tin cần thiết: Tiêu đề, Mô tả, Trạng thái, Mức độ ưu tiên, Người được giao, Hạn hoàn thành. Nhấn nút "Lưu" để cập nhật.

**Xóa công việc:** Trong trang "Chi tiết dự án", tìm đến công việc cần xóa và nhấn nút "Xóa". Lưu ý chỉ người tạo mới được xóa dự án

**Giao việc cho thành viên:** Chức năng này được thực hiện khi bạn "Thêm công việc" hoặc "Chỉnh sửa công việc" bằng cách chọn thành viên trong mục "Giao cho".

## Ghi chú, Báo cáo & Tìm kiếm

**Nộp báo cáo công việc:** Trong trang "Chi tiết dự án", tìm đến công việc đã hoàn thành và nhấn nút "Nộp báo cáo". Chọn tệp báo cáo từ máy tính của bạn. Nhấn nút "Nộp báo cáo". Tệp sẽ được tải lên và lưu lại.

**Tìm kiếm dự án:** Sử dụng chức năng tìm kiếm trên Trang chủ, nhập tên hoặc "Key" của dự án để tìm.

**Tìm kiếm công việc:** Trong trang "Chi tiết dự án", sử dụng chức năng tìm kiếm để lọc công việc theo tên.

**Ghi chú cho công việc:** Trong trang "Chi tiết công việc", nhập ghi chú vào ô “Thêm ghi chú” sau đó nhấn chuột vào nút “Gửi Ghi Chú”.

# CHƯƠNG 7: KẾT LUẬN

## Kết quả đạt được

Việc phát triển một hệ thống quản lý dự án trực tuyến mang lại nhiều lợi ích đáng kể, giúp nâng cao hiệu suất làm việc nhóm và tối ưu hóa quy trình quản lý dự án.

Hệ thống giúp tập trung hóa toàn bộ thông tin liên quan đến dự án, bao gồm danh sách các dự án đang thực hiện, thông tin chi tiết về từng công việc, thành viên tham gia, tiến độ công việc và tài liệu liên quan. Thay vì phải lưu trữ dữ liệu rời rạc trên nhiều nền tảng khác nhau, người dùng có thể dễ dàng truy cập và quản lý thông tin ngay trên một nền tảng duy nhất. Điều này không chỉ giúp giảm thiểu sự thất lạc dữ liệu mà còn tăng cường khả năng kiểm soát và giám sát tiến độ dự án.

Hệ thống giúp số hóa các quy trình quản lý dự án, bao gồm tạo dự án, lập kế hoạch công việc, phân công nhiệm vụ, theo dõi tiến độ. Nhờ vậy, các công việc thủ công được giảm bớt, tiết kiệm thời gian và nâng cao hiệu quả trong việc quản lý nhóm. Người quản lý có thể dễ dàng theo dõi toàn bộ tiến độ công việc, nhận diện các vấn đề tồn đọng và đưa ra quyết định kịp thời để đảm bảo tiến độ chung của dự án.

Việc sử dụng hệ thống quản lý dự án giúp các thành viên trong nhóm phối hợp làm việc hiệu quả hơn. Các nhiệm vụ được phân công rõ ràng, trạng thái công việc được cập nhật thường xuyên và thông tin luôn minh bạch. Điều này giúp tránh tình trạng chồng chéo công việc, giảm sai sót do nhầm lẫn và tăng cường tính trách nhiệm của từng thành viên trong nhóm.

Hệ thống được xây dựng dựa trên nền tảng ASP.NET Core theo mô hình MVC và sử dụng Entity Framework Core để quản lý cơ sở dữ liệu. Điều này giúp hệ thống có kiến trúc rõ ràng, dễ mở rộng và bảo trì. Nhờ vào công nghệ hiện đại, hệ thống có thể hỗ trợ tích hợp với nhiều nền tảng khác nhau trong tương lai, từ các công cụ quản lý khác đến các nền tảng lưu trữ đám mây, giúp nâng cao tính linh hoạt và hiệu quả sử dụng.

Dù chưa hoàn thiện đầy đủ, hệ thống đã bước đầu hỗ trợ theo dõi lịch sử tạo và giao việc. Điều này đặt nền móng cho việc đánh giá hiệu suất làm việc của từng thành viên cũng như giúp người quản lý dễ dàng theo dõi tiến trình thực hiện các công việc

quan trọng.

## Nhược điểm của hệ thống

Bên cạnh những ưu điểm nêu trên, hệ thống vẫn còn một số hạn chế nhất định, cần được khắc phục để đáp ứng tốt hơn nhu cầu thực tế.

Do giới hạn về thời gian và nguồn lực, hệ thống chưa thể triển khai đầy đủ các tính năng nâng cao thường có trong các phần mềm quản lý dự án chuyên nghiệp. Một số tính năng quan trọng như biểu đồ Gantt, bảng Kanban, hệ thống báo cáo chi tiết và quản lý rủi ro vẫn chưa được phát triển. Việc thiếu những tính năng này có thể khiến việc theo dõi tiến độ dự án chưa thực sự tối ưu.

Giao diện hiện tại chủ yếu tập trung vào chức năng cốt lõi và chưa được tối ưu hóa hoàn toàn về trải nghiệm người dùng (UI/UX). Thiết kế giao diện còn đơn giản, chưa có nhiều yếu tố hỗ trợ trực quan cho người dùng, đặc biệt là khi quản lý các dự án lớn với nhiều công việc.

Hệ thống mới chỉ hỗ trợ phân quyền cơ bản, cho phép người tạo dự án có quyền quản lý thành viên và giao việc. Tuy nhiên, chưa có hệ thống phân quyền chi tiết như Admin, Quản lý dự án và Thành viên với các quyền hạn cụ thể. Điều này có thể gây khó khăn trong việc quản lý nhóm lớn, khi cần phân chia rõ ràng trách nhiệm giữa các vai trò.

Hiện tại, hệ thống chưa hỗ trợ các tính năng như bình luận trực tiếp trên công việc, thông báo tức thời (real-time notifications) hay chat nhóm. Điều này khiến việc trao đổi thông tin trong nhóm còn bị hạn chế, đặc biệt là khi làm việc từ xa.

Các tính năng bảo mật cơ bản đã được áp dụng, nhưng nếu triển khai hệ thống trong môi trường thực tế với quy mô lớn hơn, cần bổ sung các biện pháp bảo mật nâng cao để đảm bảo an toàn dữ liệu.

## Các tính năng đã phát triển được

Hệ thống đã hoàn thành các chức năng chính như:

* + - Quản lý người dùng: đăng ký, đăng nhập, chỉnh sửa thông tin cá nhân.
    - Quản lý dự án: tạo, chỉnh sửa, xóa dự án, thêm thành viên.
    - Quản lý công việc: tạo, chỉnh sửa, xóa công việc, phân công nhiệm vụ.
    - Báo cáo và tìm kiếm: tìm kiếm dự án, công việc, nộp tài liệu.

## Các tính năng chưa thực hiện được

Một số tính năng quan trọng chưa được triển khai bao gồm:

* + - Hệ thống phân quyền chi tiết.
    - Dashboard tổng quan dự án. Biểu đồ Gantt, Kanban board. Hệ thống thông báo real-time.
    - Chức năng bình luận và thảo luận nhóm. Hệ thống báo cáo và thống kê nâng cao.

## Hướng phát triển

Trong tương lai, hệ thống có thể phát triển theo nhiều hướng nhằm nâng cao hiệu quả quản lý và đáp ứng tốt hơn nhu cầu của người dùng. Trước hết, cần hoàn thiện các tính năng còn thiếu như hệ thống phân quyền chi tiết, bảng điều khiển tổng quan (dashboard) và hệ thống thông báo. Bên cạnh đó, việc cải thiện giao diện người dùng cũng rất quan trọng để nâng cao trải nghiệm, giúp hệ thống dễ sử dụng hơn trên cả nền tảng desktop và thiết bị di động.

Ngoài ra, khả năng cộng tác cũng cần được tăng cường thông qua việc tích hợp tính năng chat nhóm và thông báo real-time, giúp các thành viên có thể trao đổi thông tin một cách nhanh chóng và hiệu quả hơn. Hệ thống cũng nên phát triển thêm các module báo cáo mạnh mẽ, hỗ trợ xuất báo cáo chi tiết để giúp nhà quản lý có cái nhìn tổng quan về dự án.

Bảo mật cũng là một yếu tố quan trọng cần được chú trọng. Việc áp dụng các biện pháp bảo vệ dữ liệu tối ưu hơn sẽ giúp hệ thống an toàn và đáng tin cậy hơn. Bên cạnh đó, hệ thống có thể được tích hợp với các nền tảng khác như Google Drive và Lịch Google để mở rộng khả năng quản lý công việc. Cuối cùng, một hướng phát triển quan trọng là xây dựng ứng dụng di động, giúp người dùng có thể quản lý công việc mọi lúc, mọi nơi, nâng cao tính linh hoạt và hiệu suất làm việc.

# TÀI LIỆU THAM KHẢO

1. Freeman, Adam. Pro ASP.NET Core MVC. Apress.
2. https://learn.microsoft.com/vi-vn/aspnet/core/
3. https://learn.microsoft.com/vi-vn/ef/core/
4. [https://www.expro.vn/amp/thiet-ke-website-quan-ly-du-an-xay-dung-chuyen-](https://www.expro.vn/amp/thiet-ke-website-quan-ly-du-an-xay-dung-chuyen-nghiep/) [nghiep/](https://www.expro.vn/amp/thiet-ke-website-quan-ly-du-an-xay-dung-chuyen-nghiep/)
5. <https://www.bitrix24.vn/?redirect_from=www.bitrix24.com%2F>
6. [https://www.atlassian.com/software/jira](http://www.atlassian.com/software/jira)