TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM

**VIỆN CÔNG NGHỆ VIỆT - NHẬT**

ĐỒ ÁN MÔN HỌC

ỨNG DỤNG QUẢN LÝ VÀ LƯU TRỮ FILE TRỰC TUYẾN

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 Nhật Trường** MSSV: 2280603493 Lớp: 22DTHJA2 2/ **Đặng Hoàng Việt** MSSV: 2280603659 Lớp: 22DTHJA2

#### TP. Hồ Chí Minh, 2025

# LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ thông tin vào quản lý và lưu trữ dữ liệu ngày càng trở nên cần thiết. Nhu cầu lưu trữ, chia sẻ và truy cập tệp tin trực tuyến một cách nhanh chóng, an toàn và tiện lợi đang trở thành xu hướng phổ biến trong các tổ chức, doanh nghiệp cũng như đối với cá nhân. Các hệ thống quản lý và lưu trữ file trực tuyến ra đời đã giúp người dùng dễ dàng quản lý dữ liệu, cộng tác và chia sẻ thông tin ở bất kỳ đâu, vào bất kỳ thời điểm nào.

Nhận thấy tầm quan trọng của vấn đề này, nhóm chúng em đã lựa chọn đề tài “Ứng dụng quản lý và lưu trữ file trực tuyến” nhằm xây dựng một hệ thống hỗ trợ người dùng trong việc tải lên, quản lý, chia sẻ và tìm kiếm tệp tin một cách hiệu quả. Hệ thống hướng đến việc cung cấp giải pháp tối ưu cho quá trình lưu trữ và đồng bộ dữ liệu, giúp nâng cao hiệu suất làm việc và khả năng cộng tác trong môi trường số.

Trong quá trình thực hiện, nhóm đã áp dụng các kiến thức về Java Spring Boot, Spring Security, RESTful API, HTML, CSS, JavaScript, và các công nghệ cơ sở dữ liệu hiện đại để phát triển hệ thống. Mặc dù đã nỗ lực hoàn thiện trong phạm vi thời gian cho phép, nhóm nhận thấy vẫn còn những thiếu sót nhất định. 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 Nhật Trường Đặng Hoàng Việt**

# 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 "Ứng dụng quản lý và lưu trữ file trực tuyến" 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 Nhật Trường Đặng Hoàng Việt

# BẢNG PHÂN CÔNG

|  |  |  |
| --- | --- | --- |
| **Họ & Tên** | **Công việc** | **Ghi chú** |
| Nguyễn Nhật Trường | * Thiết kế CSDL. * Làm chức năng quản lý dự án. * Làm chức năng quản lý công việc. * Làm chức năng quản lý thành viên trong dự án. | 27.5% |
| Nguyễn Lê Tiến Dũng | * Làm chức năng quản lý nhóm làm việc. * Làm bảng kaban kéo thả trạng thái công việc. * Làm chức năng quản lý tài khoản và dự án cho người dùng admin. | 27.5% |
| Đặng Hoàng Việt | * Làm chức năng báo cáo và ghi chú. * Thiết kế giao diện. * Làm slide. | 25% |
| Nguyễn Hữu Nhân | * Làm chức năng trò chuyện nhóm. * Làm báo cáo. | 20% |

# 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.* 14](#_bookmark14)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# DANH MỤC HÌNH ẢNH

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# CHƯƠNG 1: TỔNG QUAN

## Lý do chọn đề tài

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, dữ liệu đã trở thành một trong những tài sản quan trọng nhất của cá nhân và doanh nghiệp. Việc lưu trữ, quản lý và chia sẻ tệp tin một cách an toàn, tiện lợi và hiệu quả là nhu cầu cấp thiết trong hầu hết các lĩnh vực, đặc biệt là trong môi trường làm việc số và học tập trực tuyến hiện nay.

Tuy nhiên, trên thực tế, nhiều tổ chức và cá nhân vẫn đang sử dụng các phương pháp lưu trữ truyền thống như lưu file trên thiết bị cá nhân, ổ cứng hoặc chia sẻ qua email. Những phương pháp này tiềm ẩn nhiều rủi ro như mất dữ liệu khi thiết bị hư hỏng, khó đồng bộ giữa các thiết bị, hạn chế trong việc chia sẻ và quản lý quyền truy cập. Ngoài ra, khi khối lượng dữ liệu ngày càng lớn, việc sắp xếp và tìm kiếm file trở nên mất nhiều thời gian và thiếu hiệu quả.

Chính vì vậy, nhóm chúng em quyết định lựa chọn đề tài “Ứng dụng quản lý và lưu trữ file trực tuyến” với mong muốn xây dựng một hệ thống giúp người dùng có thể tải lên, lưu trữ, quản lý và chia sẻ file trực tuyến một cách dễ dàng, an toàn và nhanh chóng. Hệ thống sẽ giúp giảm thiểu các rủi ro về mất mát dữ liệu, đồng thời tăng tính linh hoạt trong làm việc và học tập, đặc biệt là trong môi trường cộng tác từ xa.

## Mục tiêu đề tài

Đề tài “Ứng dụng quản lý và lưu trữ file trực tuyến” được thực hiện với mục tiêu xây dựng một hệ thống hỗ trợ người dùng trong việc lưu trữ, quản lý và chia sẻ tệp tin một cách an toàn, tiện lợi và hiệu quả. Ứng dụng giúp người dùng có thể truy cập dữ liệu mọi lúc, mọi nơi, đồng thời đảm bảo tính bảo mật và đồng bộ trong quá trình sử dụng. Cụ thể, đề tài hướng đến các mục tiêu sau:

* Xây dựng một website quản lý và lưu trữ file trực tuyến cho phép người dùng tải lên, tải xuống, chỉnh sửa và xóa tệp tin dễ dàng.
* Hỗ trợ quản lý dữ liệu khoa học, giúp người dùng phân loại, tìm kiếm và truy cập file nhanh chóng, thuận tiện.
* Đảm bảo an toàn và bảo mật thông tin, thông qua cơ chế phân quyền truy cập, xác thực người dùng và bảo vệ dữ liệu trên hệ thống.
* Tiết kiệm thời gian và nâng cao hiệu suất làm việc, giúp người dùng giảm thiểu rủi ro mất dữ liệu, đồng thời tăng tính linh hoạt khi chia sẻ và cộng tác trực tuyến.

## Phương pháp nghiên cứu đề tài

Nghiên cứu lý thuyết và thu thập dữ liệu: Nhóm tiến hành tìm hiểu và tổng hợp kiến thức liên quan đến mạng máy tính, giao thức truyền thông (HTTP, TCP/IP), mô hình client–server, cũng như các công nghệ hỗ trợ truyền tải và trao đổi dữ liệu qua mạng. Bên cạnh đó, nhóm tham khảo tài liệu về lập trình socket, giao tiếp giữa client và server, cùng các nguyên lý trong phát triển ứng dụng web mạng. Ngoài ra, nhóm khảo sát các nền tảng lưu trữ trực tuyến phổ biến như Google Drive, Dropbox, OneDrive để hiểu rõ cách thức hoạt động và yêu cầu về bảo mật, truyền tải dữ liệu.

Phân tích và thiết kế hệ thống: Dựa trên kết quả nghiên cứu, nhóm tiến hành phân tích yêu cầu chức năng và phi chức năng của hệ thống, tập trung vào khả năng giao tiếp mạng giữa client và server, truyền nhận file qua giao thức HTTP, và xử lý yêu cầu đồng thời từ nhiều người dùng. Hệ thống được thiết kế theo mô hình client–server, trong đó Spring Boot đóng vai trò server xử lý các yêu cầu RESTful API, còn giao diện web đóng vai trò client gửi và nhận dữ liệu thông qua mạng. Ngoài ra, nhóm xây dựng mô hình cơ sở dữ liệu phục vụ cho việc lưu trữ thông tin người dùng, siêu dữ liệu của file và log truy cập. Giao diện người dùng (UI) được thiết kế thân thiện, hỗ trợ thao tác tải lên, tải xuống và chia sẻ file dễ dàng.

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. Quá trình phát triển tập trung vào việc xây dựng API truyền tải file, xử lý yêu cầu đồng thời, bảo mật khi truyền dữ liệu, và quản lý phiên làm việc (session) của người dùng. Hệ thống được kiểm thử trong môi trường mạng cục bộ (localhost) để đảm bảo khả năng kết nối, truyền nhận dữ liệu ổn định, và duy trì tốc độ xử lý khi có nhiều người dùng truy cập đồng thời.

## Ý nghĩa của việc nghiên cứu đề tài

Việc xây dựng ứng dụng quản lý và lưu trữ file trực tuyến có ý nghĩa quan trọng trong bối cảnh công nghệ thông tin ngày càng phát triển và nhu cầu làm việc, học tập từ xa trở nên phổ biến. Hệ thống không chỉ giúp người dùng lưu trữ, chia sẻ và quản lý tệp tin một cách an toàn, tiện lợi và nhanh chóng, mà còn góp phần tăng cường khả năng kết nối và trao đổi dữ liệu qua mạng giữa nhiều người dùng ở các vị trí khác nhau. Ứng dụng này mang lại giải pháp thiết thực trong việc tối ưu hóa quá trình làm việc, giảm thiểu rủi ro mất dữ liệu và đảm bảo tính bảo mật, đồng bộ khi truyền tải thông tin qua mạng Internet.

Bên cạnh ý nghĩa thực tiễn, đề tài còn mang giá trị học thuật cao khi giúp chúng em vận dụng kiến thức đã học trong học phần Lập trình mạng máy tính vào việc phát triển một hệ thống thực tế. Thông qua quá trình nghiên cứu và triển khai, chúng em được củng cố hiểu biết về mô hình client–server, giao thức HTTP, lập trình socket, và truyền nhận dữ liệu qua mạng, đồng thời rèn luyện kỹ năng phân tích, thiết kế, lập trình và làm việc nhóm.

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 giải quyết vấn đề trong môi trường mạng, và sẵn sàng áp dụng vào các dự án công nghệ 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.

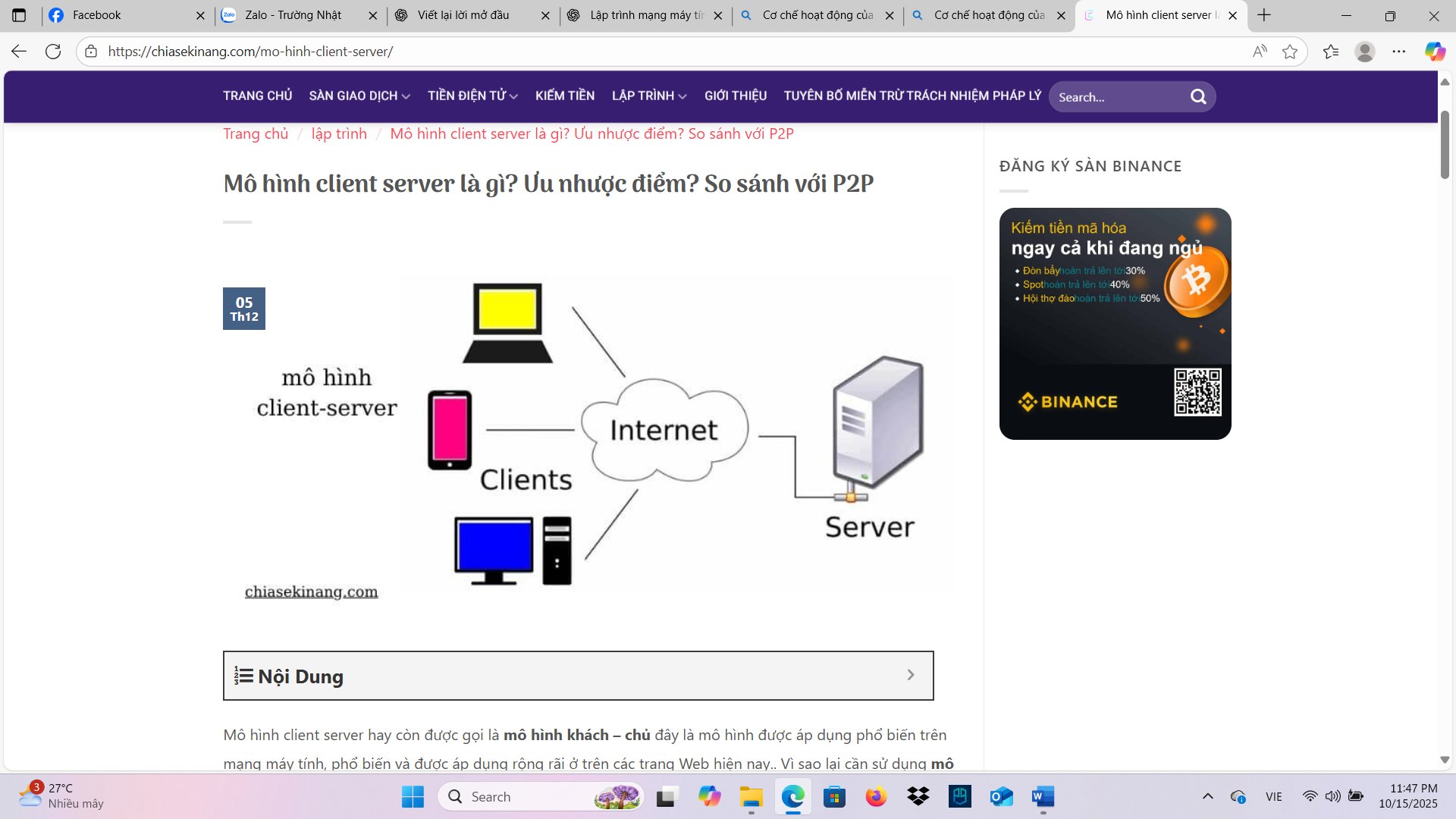
Ngày nay, lập trình mạng không chỉ giới hạn trong việc xây dựng các ứng dụng web hay dịch vụ Internet mà còn mở rộng sang nhiều lĩnh vực khác như IoT (Internet of Things), hệ thống phân tán, điện toán đám mây, trò chơi trực tuyến (online games) hay ứng dụng di động kết nối thời gian thực. Với sự phát 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, file tải lên, yêu cầu tìm kiếm, v.v.) và gửi yêu cầu tương ứng đến Server thông qua các giao thức mạng (phổ biến nhất là HTTP/HTTPS). Trong hệ thống đề tài “Ứng dụng quản lý và lưu trữ file trực tuyến”, Client chính là giao diện web (Frontend), nơi người dùng có thể thao tác trực tiếp như tải tệp lên, xem danh sách tệp, hoặc xóa file.

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, tương tác với cơ sở dữ liệu, xác thực người dùng và quản lý toàn bộ quy trình lưu trữ file. Khi Client gửi yêu cầu đến, Server sẽ phân tích dữ liệu, thực hiện thao tác cần thiết (ví dụ: lưu file, truy vấn danh sách, xóa dữ liệu, v.v.), sau đó gửi phản hồi (response) về cho Client dưới dạng dữ liệu JSON hoặc thông báo trạng thái.



*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 hoặc thiết bị có thể hiểu và giao tiếp được với nhau, việc sử dụng các giao thức truyền thông (communication protocols) là vô cùng cần thiết. Giao thức truyền thông có thể hiểu là tập hợp các quy tắc, tiêu chuẩn và quy định về cách thức thiết lập kết nối, trao đổi, xử lý và kết thúc quá trình truyền dữ liệu giữa các bên tham gia trong mạng máy tính. Mỗi giao thức đảm nhiệm một vai trò riêng trong các tầng của mô hình mạng (chẳng hạn như mô hình TCP/IP hoặc OSI), giúp đảm bảo dữ liệu được truyền đi đúng định dạng, đúng thứ tự, đầy đủ và an toàn.

Trong hệ thống “Ứng dụng quản lý và lưu trữ file trực tuyến”, việc truyền tải và trao đổi dữ liệu giữa Client (giao diện người dùng) và Server (ứng dụng Spring Boot) được thực hiện thông qua nhiều giao thức khác nhau, tùy theo chức năng và yêu cầu cụ thể của hệ thống. Dưới đây là các giao thức tiêu biểu được sử dụng:

* TCP (Transmission Control Protocol): TCP là một trong những giao thức quan trọng nhất trong bộ giao thức Internet, nằm ở tầng Transport (vận chuyển) của mô hình TCP/IP. TCP đảm bảo truyền dữ liệu đáng tin cậy giữa hai điểm đầu cuối bằng cách thiết lập một kết nối hai chiều (connection-oriented) trước khi trao đổi dữ liệu. Nó có khả năng kiểm tra lỗi, sắp xếp lại các gói tin theo đúng thứ tự, và tự động gửi lại gói tin nếu phát hiện mất mát trong quá trình truyền. Nhờ đó, TCP trở thành nền tảng vững chắc cho các giao thức tầng cao hơn như HTTP, FTP hay SMTP. Trong hệ thống này, TCP là cơ chế cơ bản giúp duy trì kết nối ổn định giữa client và server, đảm bảo file và dữ liệu người dùng được gửi đi chính xác, không bị lỗi hoặc gián đoạn.
* HTTP/HTTPS (HyperText Transfer Protocol / Secure): HTTP là giao thức được sử dụng phổ biến nhất trong ứng dụng web, cho phép truyền tải dữ liệu dưới dạng các yêu cầu (request) và phản hồi (response) giữa trình duyệt (client) và máy chủ web (server). HTTP hoạt động dựa trên cơ chế stateless – nghĩa là mỗi yêu cầu được xử lý độc lập, không lưu lại trạng thái giữa các lần truy cập. Trong các hệ thống hiện đại, HTTPS là phiên bản nâng cao của HTTP, bổ sung lớp mã hóa SSL/TLS, giúp đảm bảo tính bảo mật, xác thực và toàn vẹn dữ liệu khi truyền tải trên Internet. Trong đề tài này, khi người dùng tải file, đăng nhập, hoặc xem danh sách file, tất cả yêu cầu đều được gửi thông qua giao thức HTTP hoặc HTTPS, đảm bảo dữ liệu được truyền đi một cách an toàn và tin cậy.
* RESTful API (Representational State Transfer): RESTful API không phải là một giao thức, mà là một kiến trúc thiết kế dựa trên giao thức HTTP, được sử dụng rộng rãi trong việc xây dựng các hệ thống web hiện đại. REST quy định cách thức client và server trao đổi dữ liệu thông qua các phương thức HTTP tiêu chuẩn như: GET – lấy dữ liệu từ server, POST – gửi dữ liệu mới lên server, PUT hoặc PATCH – cập nhật dữ liệu, DELETE – xóa dữ liệu. Dữ liệu trao đổi giữa hai bên thường sử dụng định dạng JSON (JavaScript Object Notation) vì tính gọn nhẹ, dễ đọc và dễ xử lý. Trong ứng dụng quản lý và lưu trữ file trực tuyến, RESTful API là cầu nối quan trọng giữa frontend và backend, giúp các thao tác như tải file, xem danh sách file, chỉnh sửa thông tin người dùng hay xóa file được thực hiện nhanh chóng và chuẩn hóa.
* WebSocket (mở rộng trong tương lai): Bên cạnh các giao thức truyền thống như HTTP, hệ thống có thể được mở rộng để hỗ trợ WebSocket – một giao thức truyền thông hai chiều (full-duplex) cho phép client và server duy trì một kênh kết nối liên tục. Không giống HTTP, nơi client phải gửi yêu cầu mới nhận được phản hồi, WebSocket cho phép server chủ động gửi dữ liệu ngược lại cho client bất cứ khi nào có sự kiện xảy ra. Điều này đặc biệt hữu ích cho các tính năng truyền dữ liệu thời gian thực (real-time), chẳng hạn như: thông báo tải file hoàn tất, cập nhật tiến trình upload, cảnh báo hệ thống, hay thông tin chia sẻ file giữa nhiều người dùng cùng lúc.

Tổng kết lại, việc lựa chọn và sử dụng hợp lý các giao thức truyền thông không chỉ đảm bảo hệ thống hoạt động ổn định, hiệu quả, mà còn giúp nâng cao trải nghiệm người dùng và tăng cường bảo mật dữ liệu. Nhờ sự kết hợp giữa các giao thức như TCP, HTTP/HTTPS và RESTful API, ứng dụng quản lý và lưu trữ file trực tuyến có thể vận hành một cách mượt mà, đáng tin cậy và dễ mở rộng trong tương lai.

### 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). Nhờ socket, các chương trình có thể trao đổi dữ liệu theo thời gian thực mà không cần phụ thuộc hoàn toàn vào các giao thức tầng cao như HTTP.

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.

Mặc dù trong đề tài “Ứng dụng quản lý và lưu trữ file trực tuyến”, hệ thống chủ yếu sử dụng HTTP/RESTful API (vốn 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. Bên dưới mọi giao thức web, 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 quản lý và lưu trữ file trực tuyến”, lập trình mạng đóng vai trò then chốt trong việc kết nối, trao đổi và xử lý dữ liệu giữa các thành phần của hệ thống. Toàn bộ quá trình giao tiếp giữa người dùng (client) và máy chủ (server) đều được thực hiện thông qua các cơ chế và giao thức lập trình mạng. Việc ứng dụng các kỹ thuật lập trình mạng không chỉ giúp hệ thống hoạt động trơn tru, ổn định, mà còn đảm bảo hiệu năng, bảo mật và khả năng mở rộng trong tương lai. Cụ thể, lập trình mạng được triển khai xuyên suốt qua nhiều tầng (layer) trong hệ thống như sau:

**Tầng giao tiếp (Communication Layer)**: Đây là tầng chịu trách nhiệm thiết lập và duy trì việc trao đổi dữ liệu giữa client và server. Trong đề tài, các yêu cầu của người dùng như tải lên (upload), tải xuống (download), xem danh sách file, hay xóa file đều được gửi thông qua HTTP request. Mỗi request chứa thông tin về đường dẫn API, dữ liệu cần gửi (file, metadata, token, v.v.) và được server phản hồi lại bằng HTTP response chứa mã trạng thái (status code) cùng dữ liệu JSON.

**Tầng xử lý (Processing Layer)**: Sau khi nhận được dữ liệu từ client, server (xây dựng bằng Spring Boot) sẽ đảm nhận vai trò xử lý chính. Tầng này thực hiện các nghiệp vụ cốt lõi như xác thực người dùng, kiểm tra dữ liệu đầu vào, xử lý logic lưu trữ file, và tương tác với cơ sở dữ liệu. Mỗi yêu cầu đến từ client đều được định tuyến đến một controller tương ứng, nơi áp dụng lập trình mạng thông qua các RESTful API để thực thi hành động cụ thể.

**Tầng bảo mật (Security Layer)**: Trong môi trường mạng, bảo mật là yếu tố bắt buộc để đảm bảo an toàn dữ liệu và quyền riêng tư của người dùng. Hệ thống trong đề tài áp dụng giao thức HTTPS – phiên bản bảo mật của HTTP, giúp mã hóa dữ liệu truyền đi giữa client và server bằng chứng chỉ SSL/TLS, ngăn chặn việc nghe lén hoặc giả mạo dữ liệu. Ngoài ra, hệ thống còn sử dụng JSON Web Token (JWT) để xác thực và phân quyền người dùng. Khi người dùng đăng nhập thành công, server sẽ gửi lại một mã token. Mỗi lần client gửi request, token này sẽ được đính kèm vào header của yêu cầu. Server kiểm tra token để xác định người dùng hợp lệ, từ đó cho phép hoặc từ chối truy cập tài nguyên. Cách tiếp cận này giúp hệ thống đảm bảo tính bảo mật, toàn vẹn dữ liệu và ngăn ngừa truy cập trái phép.

**Tầng lưu trữ (Storage Layer)**: Tầng này đảm nhiệm việc lưu trữ và truy xuất dữ liệu – bao gồm cả thông tin người dùng và các file được tải lên. Server sẽ giao tiếp với cơ sở dữ liệu (SQL Server) thông qua các giao thức mạng nội bộ (TCP) để lưu trữ thông tin metadata của file (tên, dung lượng, ngày tải lên, người sở hữu, v.v.). Bên cạnh đó, hệ thống có thể mở rộng để lưu trữ file trên dịch vụ đám mây (Cloud Storage) như Google Cloud Storage hoặc Amazon S3. Khi đó, server sẽ sử dụng API của dịch vụ cloud để truyền tải file thông qua Internet. Tất cả các hoạt động này – từ kết nối, gửi dữ liệu đến xác nhận lưu trữ thành công – đều được điều khiển bằng các kỹ thuật lập trình mạng.

* 1. Công nghệ sử dụng

**Ngôn ngữ và nền tảng lập trình**:

* Java: Ngôn ngữ chính cho backend, có tính ổn định, bảo mật và khả năng mở rộng cao.
* Spring Boot: Framework hỗ trợ phát triển ứng dụng web nhanh chóng, tích hợp RESTful API, bảo mật và quản lý dependency thuận tiện.

**Cơ sở dữ liệu**:

* SQL Server: Hệ quản trị cơ sở dữ liệu quan hệ được dùng để lưu trữ thông tin người dùng, thư mục, tệp tin và lịch sử thao tác.
* Spring Data JPA: Cung cấp cơ chế ORM (Object-Relational Mapping), giúp thao tác dữ liệu dễ dàng mà không cần viết nhiều câu lệnh SQL.

**Giao diện người dùng (Frontend)**:

* HTML5: Xây dựng cấu trúc và nội dung trang web.
* CSS3: Tạo kiểu dáng, bố cục và định dạng hiển thị.
* JavaScript: Tăng tính tương tác, xử lý các thao tác động như tải file, hiển thị thông báo, báo cáo trực tuyến.

**Công nghệ lưu trữ file**: Lưu trữ cục bộ (Local Storage): Lưu file trực tiếp trên máy chủ.

**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 hệ thống là xây dựng một nền tảng web giúp người dùng:

* Lưu trữ và quản lý tệp tin trực tuyến một cách an toàn và tiện lợi.
* Tải lên (upload) và tải xuống (download) file từ bất kỳ thiết bị nào có kết nối mạng.
* Chia sẻ file giữa các người dùng hoặc nhóm làm việc.
* Quản lý dung lượng và thư mục cá nhân, hỗ trợ sắp xếp và tìm kiếm dễ dàng.
* Theo dõi và ghi lại lịch sử thao tác (log) để đảm bảo tính minh bạch và truy vết khi cần.
* Đồng bộ dữ liệu qua mạng giúp truy cập linh hoạt mà không phụ thuộc vào máy tính cục bộ.

Hệ thống được thiết kế hướng đến:

* Tính bảo mật cao, đảm bảo dữ liệu người dùng không bị rò rỉ.
* Tính mở rộng, có thể tích hợp thêm các dịch vụ như lưu trữ đám mây hoặc chia sẻ qua API.
* Tính thực tiễn, có thể áp dụng trong môi trường doanh nghiệp, trường học hoặc cá nhân.

1. * 1. Đối tượng sử dụng hệ thống

Người dùng cá nhân:

* Thực hiện các thao tác cơ bản như đăng nhập, tải lên, tải xuống, xóa hoặc đổi tên file.
* Quản lý thư mục cá nhân và chia sẻ file cho người khác (nếu được cấp quyền).
* Xem lịch sử thao tác để theo dõi các hành động đã thực hiện.

Quản trị viên (Admin):

* Quản lý toàn bộ hệ thống: người dùng, dung lượng, và quyền truy cập.
* Theo dõi log hoạt động của các tài khoản.
* Quản lý tài nguyên máy chủ, tối ưu dung lượng lưu trữ.
* Thiết lập cấu hình bảo mật, chính sách phân quyền và cấp quyền chia sẻ.
  + 1. Yêu cầu chức năng

Chức năng quản lý tài khoản:

* Đăng ký tài khoản mới.
* Đăng nhập và đăng xuất hệ thống.
* Thay đổi mật khẩu, cập nhật thông tin cá nhân.
* Xác thực người dùng bằng token (JWT).

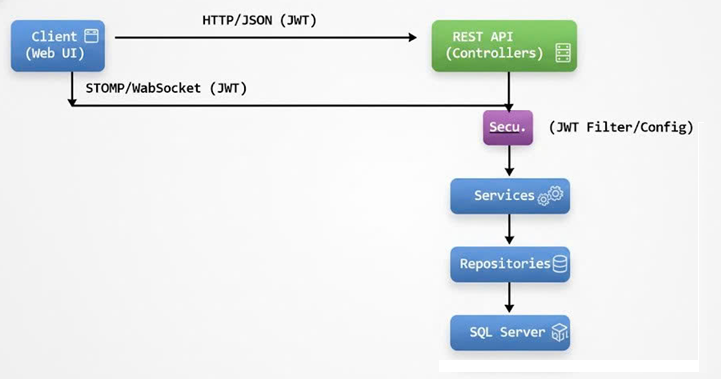
Chức năng quản lý tệp tin:

* Tải lên (upload) file từ máy tính lên hệ thống.
* Tải xuống (download) file từ hệ thống về máy người dùng.
* Xóa, đổi tên, hoặc di chuyển file giữa các thư mục.
* Quản lý thư mục cá nhân (tạo mới, đổi tên, xóa thư mục).
* Xem danh sách tệp theo thư mục, thời gian hoặc loại tệp.

Chức năng chia sẻ và thông báo:

* Gửi và nhận thông báo khi có thao tác quan trọng (upload, chia sẻ file).
* Hỗ trợ thông báo thời gian thực (realtime) thông qua WebSocket.
* Cho phép chia sẻ file qua đường dẫn hoặc quyền truy cập cụ thể.
  1. Kiến trúc hệ thống
     1. Mô hình tổng thể

Hệ thống được thiết kế theo mô hình client–server, trong đó các thành phần giao tiếp với nhau thông qua RESTful API và WebSocket. Mục tiêu của kiến trúc là phân tách rõ ràng giữa giao diện người dùng (Frontend), xử lý nghiệp vụ (Backend) và lưu trữ dữ liệu (Database), giúp hệ thống dễ bảo trì, mở rộng và đảm bảo tính bảo mật.



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

Client ↔ REST API ↔ Database:

* Client (HTML/CSS/JavaScript): Là giao diện người dùng chạy trực tiếp trên trình duyệt web, có nhiệm vụ hiển thị danh sách file, thư mục, dung lượng lưu trữ và các thông báo trạng thái upload. Khi người dùng thực hiện thao tác như tải file, tạo thư mục, đăng nhập hoặc chia sẻ file, hệ thống frontend sẽ gửi yêu cầu HTTP đến server thông qua REST API dưới dạng JSON để được xử lý.
* REST API (Spring Boot): Là lớp trung gian chịu trách nhiệm xử lý toàn bộ logic nghiệp vụ của hệ thống. REST API tiếp nhận request từ client, kiểm tra hợp lệ, xác thực và phân quyền người dùng theo vai trò. Các thao tác CRUD (Create, Read, Update, Delete) được thực hiện qua các endpoint như /api/auth (đăng nhập, đăng ký), /api/files (upload, download, chia sẻ, xóa), /api/folders (quản lý thư mục) và /api/admin/logs (quản lý lịch sử thao tác). Hệ thống sử dụng JPA/Hibernate để kết nối SQL Server, ánh xạ các đối tượng Java với bảng dữ liệu (ORM). Sau khi xử lý, API trả về phản hồi dạng JSON cho client.
* Database (SQL Server): Lưu trữ toàn bộ thông tin cần thiết của hệ thống bao gồm người dùng (User), tệp tin (File), thư mục (Folder) và nhật ký hoạt động (Logs). Các bảng được thiết kế theo chuẩn hóa 3NF và có chỉ mục (index) để tối ưu hiệu suất truy vấn. Dữ liệu được truy xuất hoặc ghi thông qua kết nối TCP (port 1433) bằng JDBC Driver của Microsoft, đảm bảo tính ổn định và bảo mật trong giao tiếp với REST API.

Client ↔ WebSocket Broker:

* Kết nối thời gian thực (Real-time Communication): Client sử dụng WebSocket (hoặc SockJS nếu không hỗ trợ WebSocket gốc) để kết nối với server thông qua endpoint /ws. Sau khi thiết lập, kết nối hai chiều giữa client và server được duy trì liên tục, cho phép trao đổi dữ liệu thời gian thực mà không cần gửi lại nhiều request HTTP.
* WebSocket Broker (Spring Internal Message Broker): Được tích hợp sẵn trong Spring Boot, hoạt động như hệ thống phân phối thông điệp nội bộ. Các module backend có thể phát (publish) thông báo đến các topic như /topic/upload-status (tiến trình tải file), /topic/share-notification (thông báo chia sẻ file) hoặc /topic/system-alert (cảnh báo hệ thống). Client có thể đăng ký (subscribe) các topic này để nhận thông tin ngay lập tức mà không cần tải lại trang.
* Ứng dụng thực tế: Khi người dùng tải file lên, backend gửi thông báo thời gian thực về tiến độ. Khi file được chia sẻ, người nhận sẽ nhận được thông báo tức thì trên giao diện web. Ngoài ra, khi có thay đổi trong thư mục chung như thêm, sửa hoặc xóa file, tất cả client liên quan đều được cập nhật ngay lập tức.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

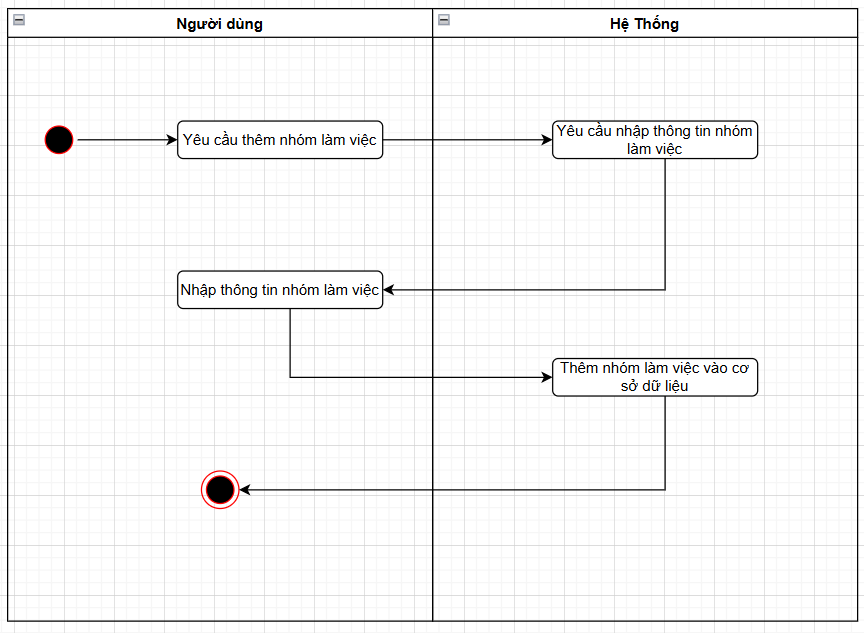
*Bảng 3.1: Các thành phần chính của hệ thống.*

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

## Sơ đồ đăng nhập

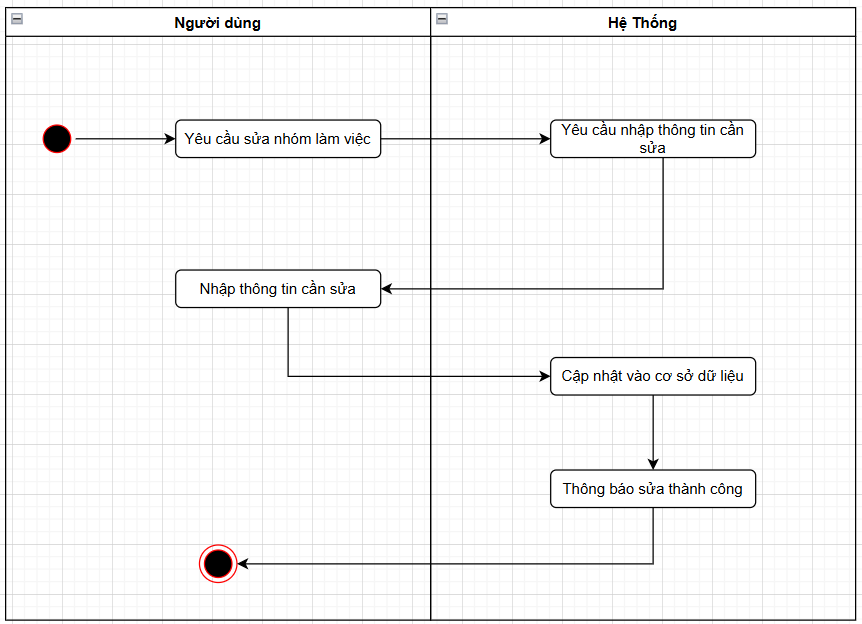
*Hình 4.1: Mô hình hoạt động đăng nhập.*

## Sơ đồ hoạt động tạo nhóm làm việc



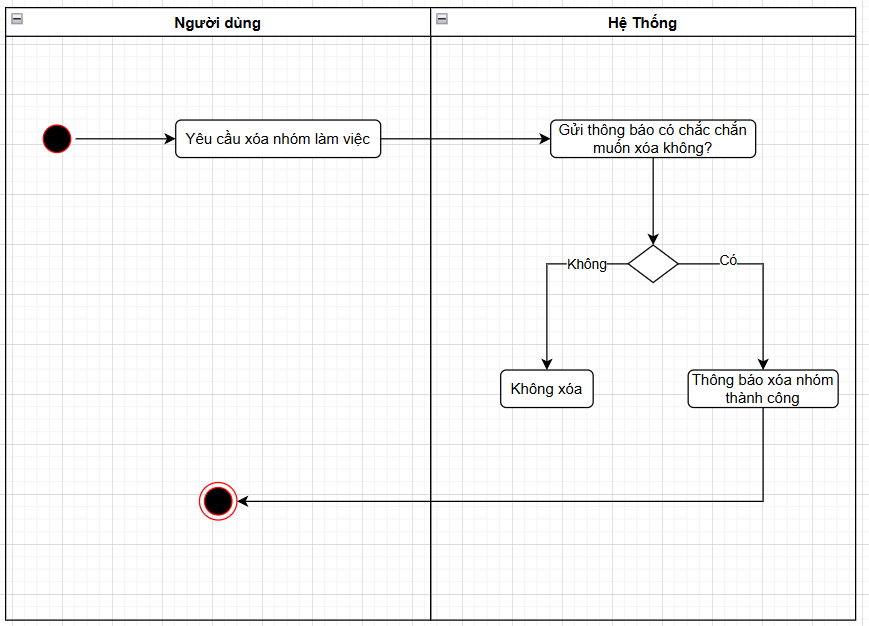
*Hình 4.2: Mô hình hoạt động tạo nhóm làm việc.*

## Sơ đồ hoạt động sửa nhóm làm việc



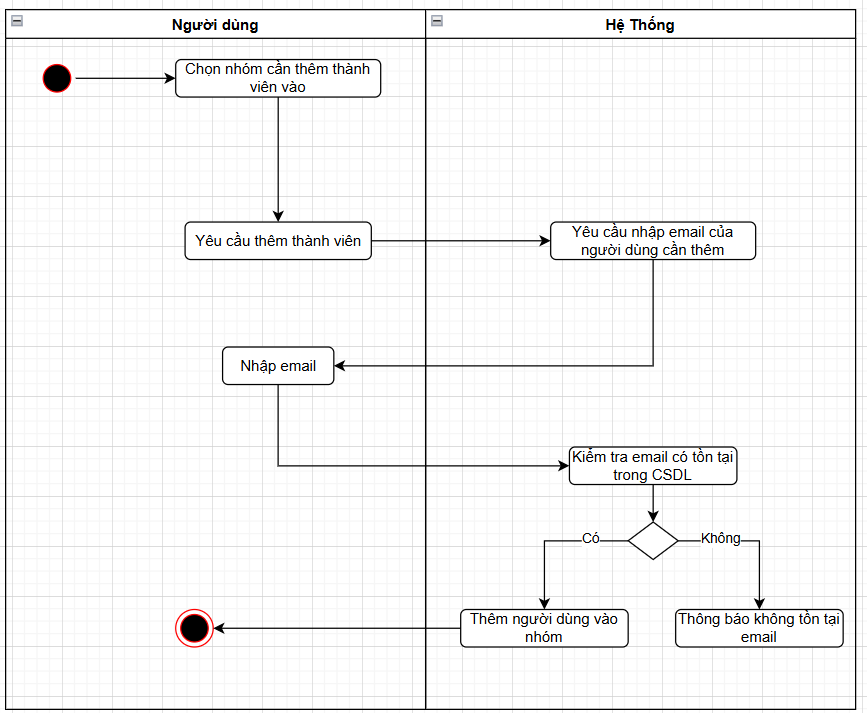
*Hình 4.3: Mô hình hoạt động sửa nhóm làm việc.*

## Sơ đồ hoạt động xóa nhóm làm việc



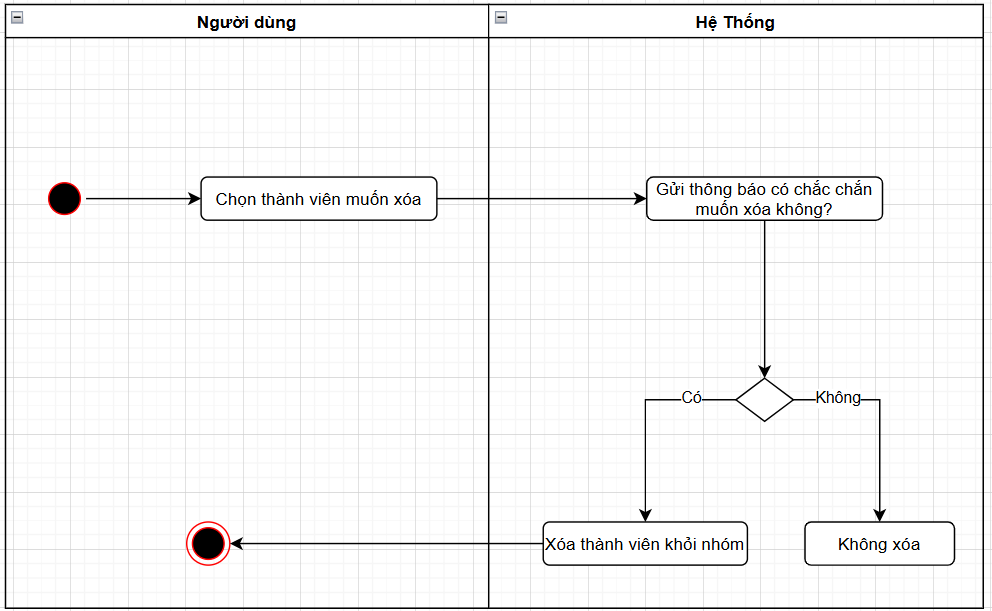
*Hình 4.4: Mô hình hoạt động xóa nhóm làm việc.*

## Sơ đồ hoạt động thêm thành viên vào nhóm



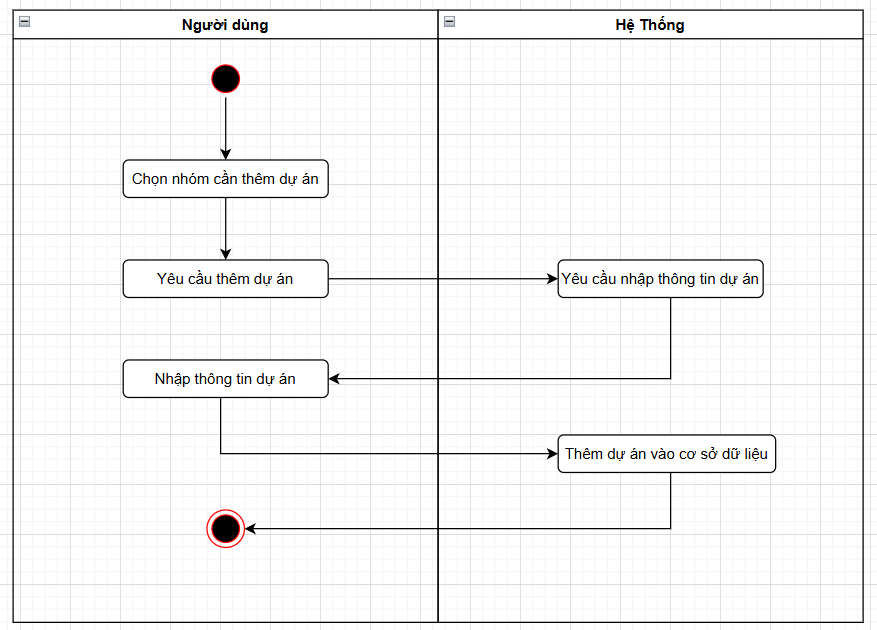
*Hình 4.5: Mô hình hoạt động thêm thành viên vào nhóm.*

## Sơ đồ hoạt động xóa thành viên khỏi nhóm



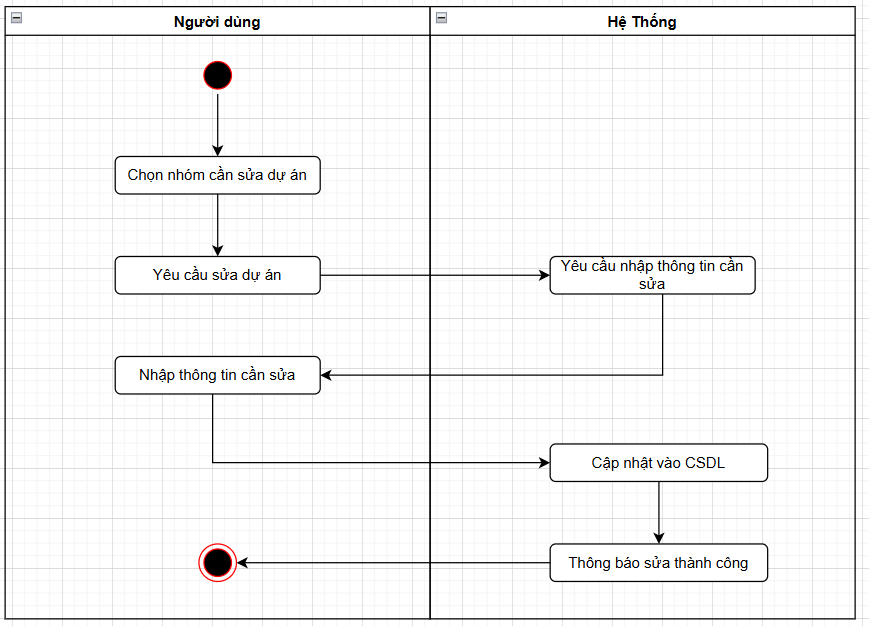
*Hình 4.6: Mô hình hoạt động xóa thành viên khỏi nhóm.*

## Sơ đồ hoạt động tạo dự án



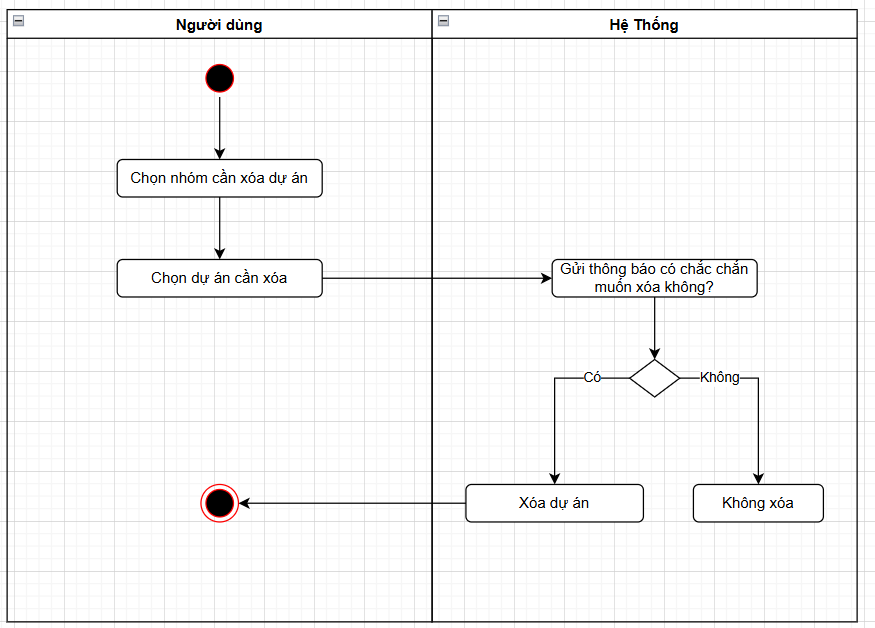
*Hình 4.7: Mô hình hoạt động tạo dự án.*

## Sơ đồ hoạt động sửa dự án



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

## Sơ đồ hoạt động xóa dự án

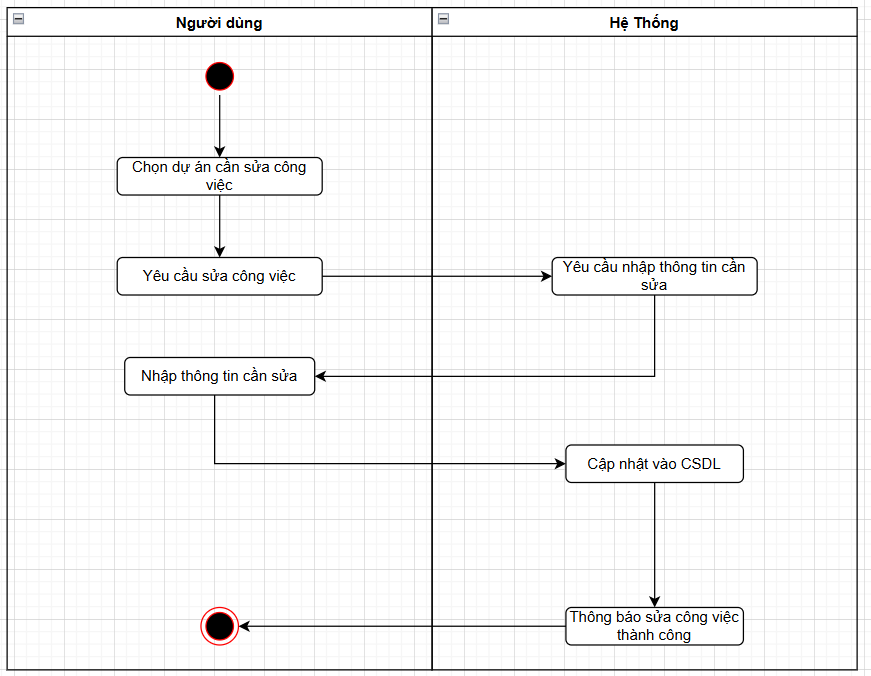


*Hình 4.9: Mô hình hoạt động xóa dự án.*

## Sơ đồ hoạt động thêm công việc

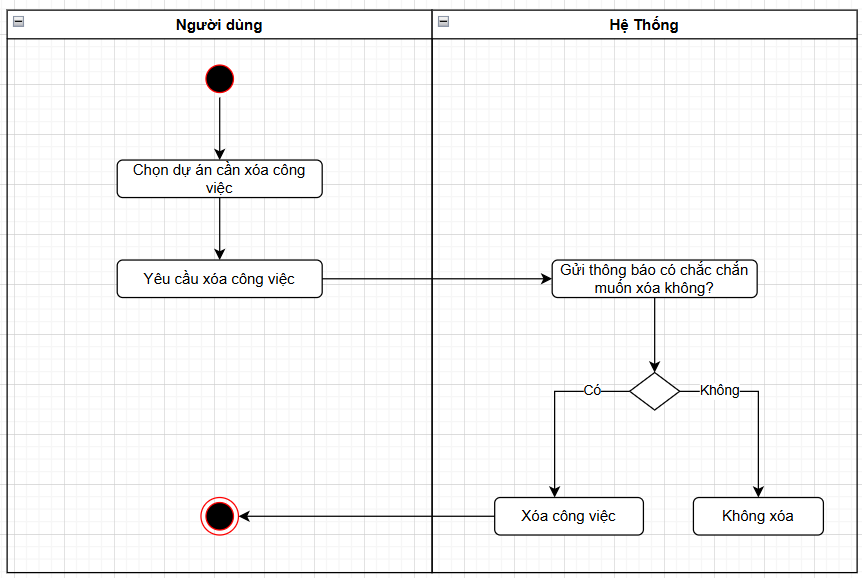
*Hình 4.10: Mô hình hoạt động thêm công việc.*

## Sơ đồ hoạt động sửa công việc



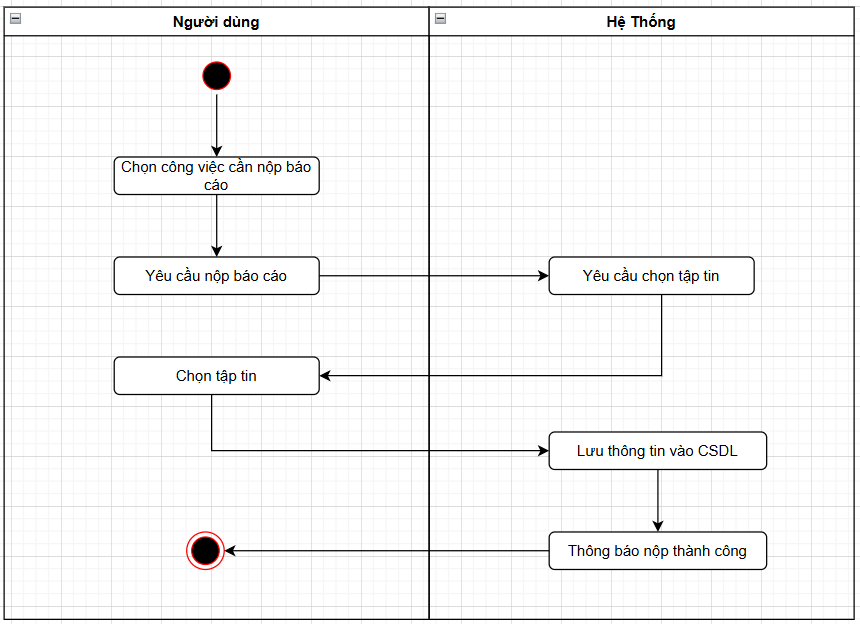
*Hình 4.11: Mô hình hoạt động sửa công việc.*

## Sơ đồ hoạt động xóa công việc



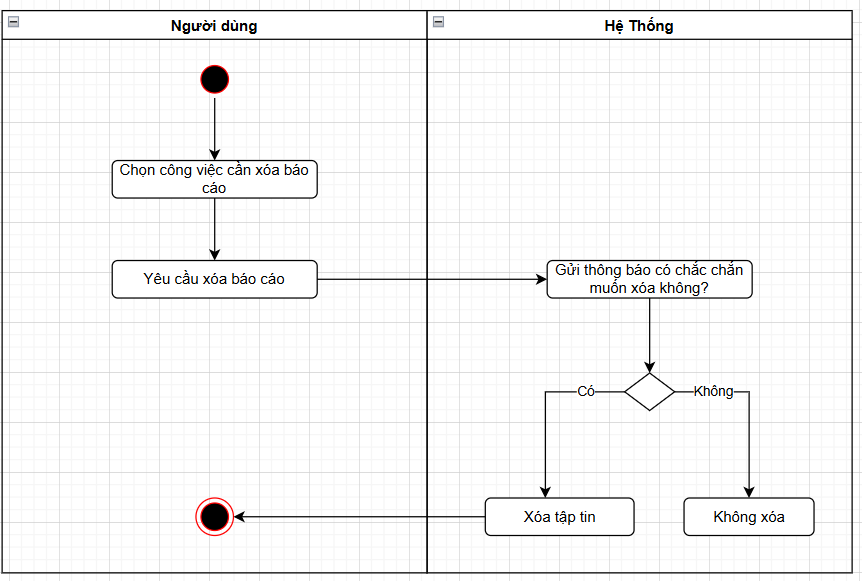
*Hình 4.12: Mô hình hoạt động xóa công việc.*

## Sơ đồ hoạt động nộp báo cáo



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

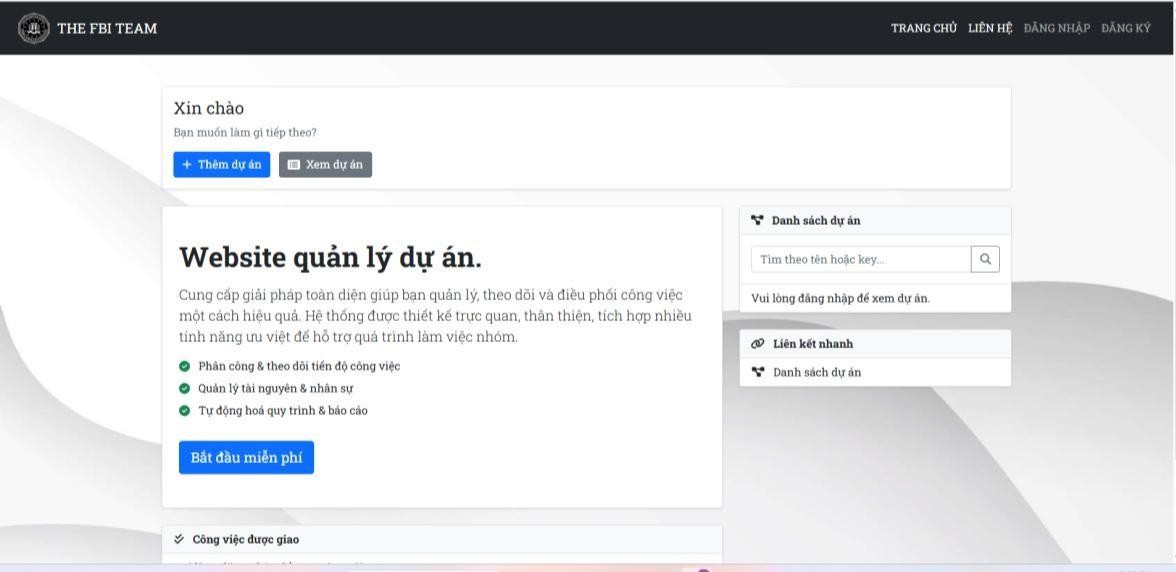
## Sơ đồ hoạt động xóa báo cáo



*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

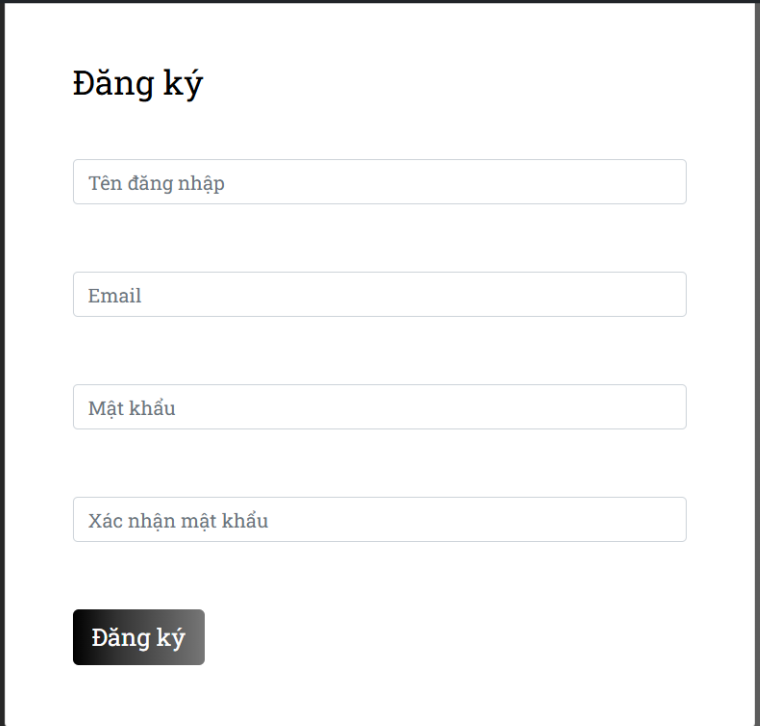
****

*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

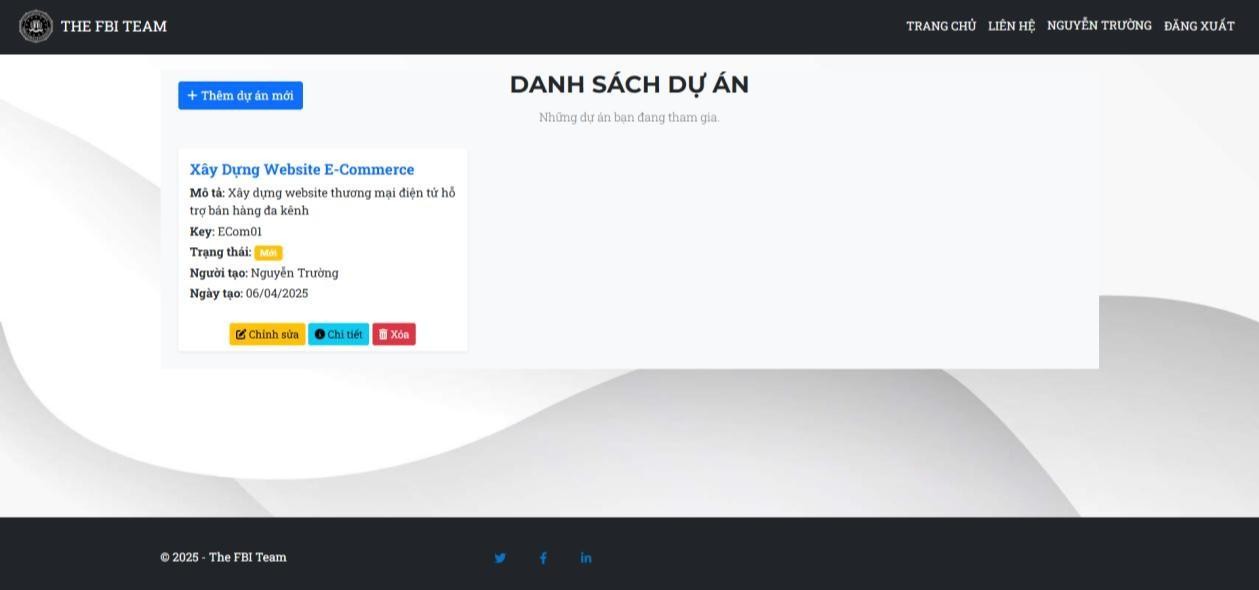
****

*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

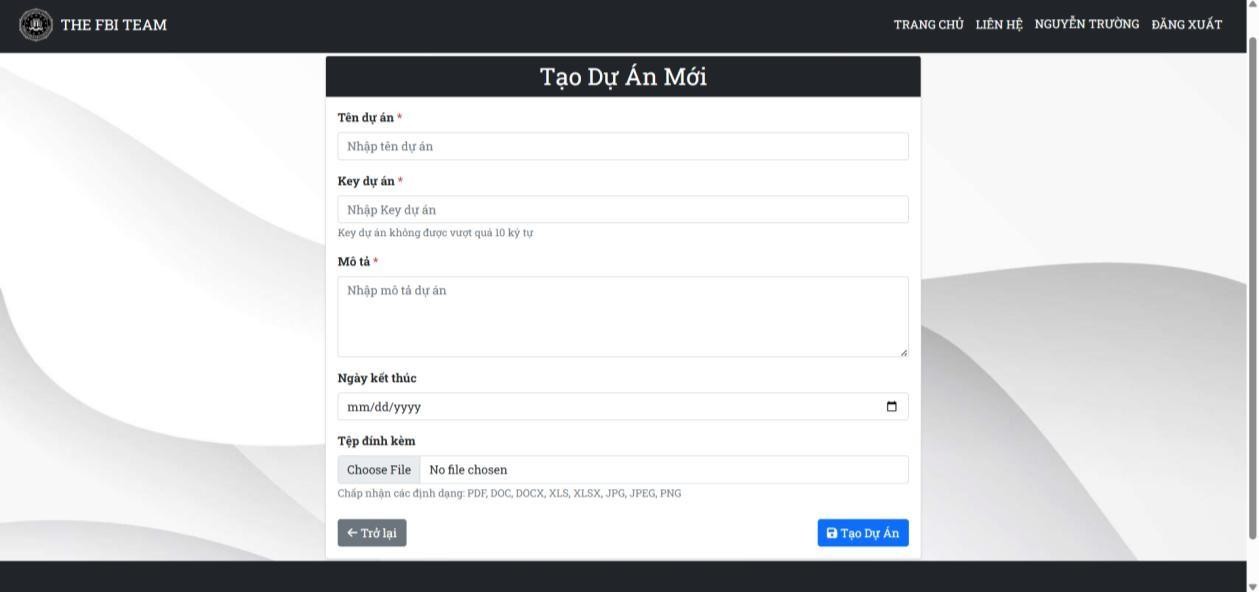
****

*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

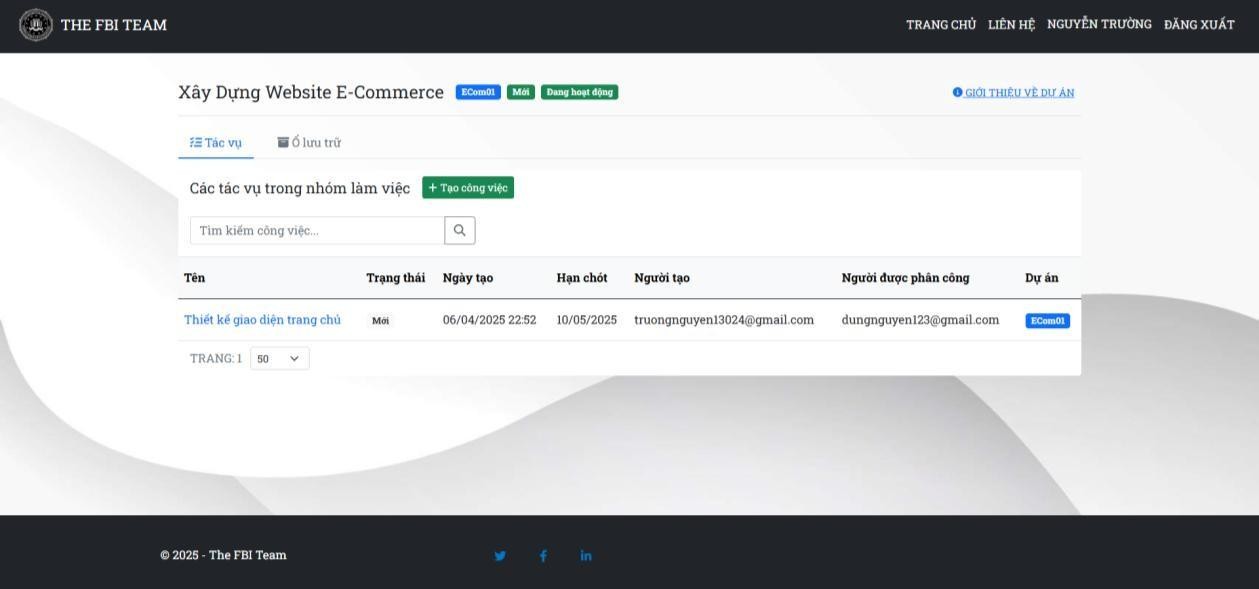
****

*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

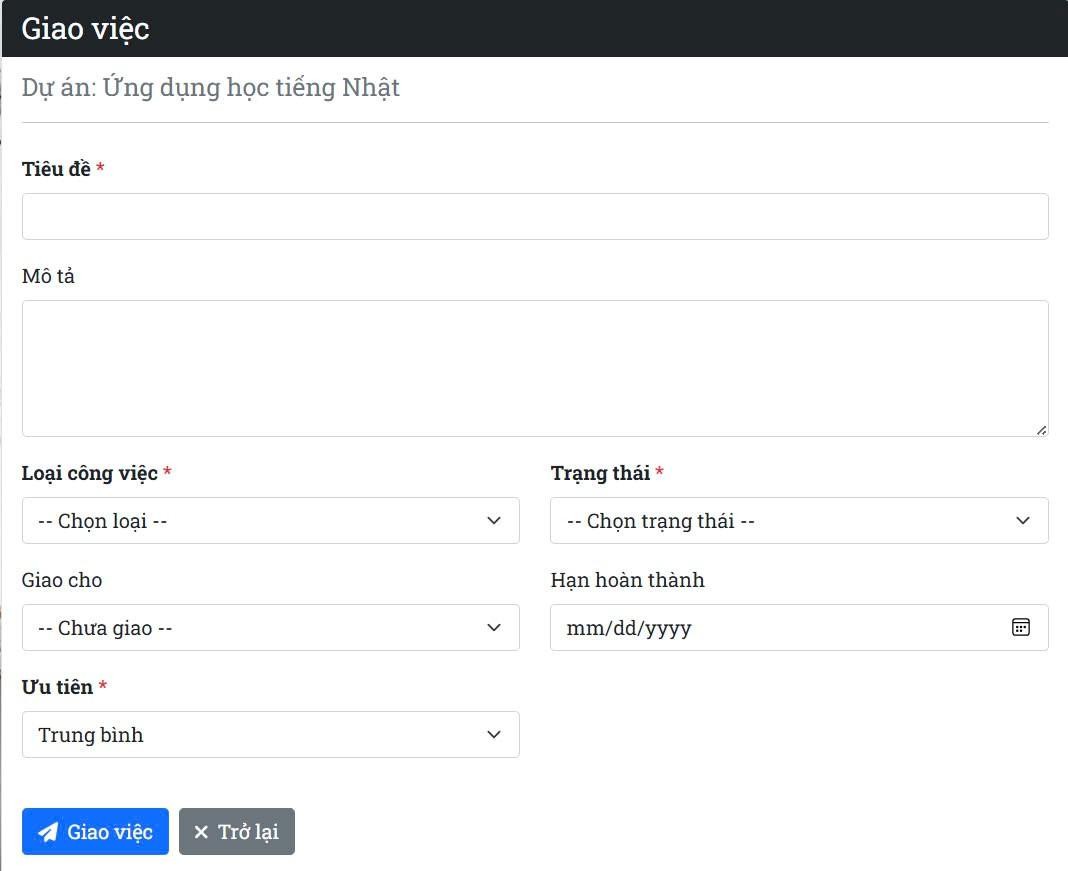


*Hình 5.8: Giao diện thêm thành viên vào dự án.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

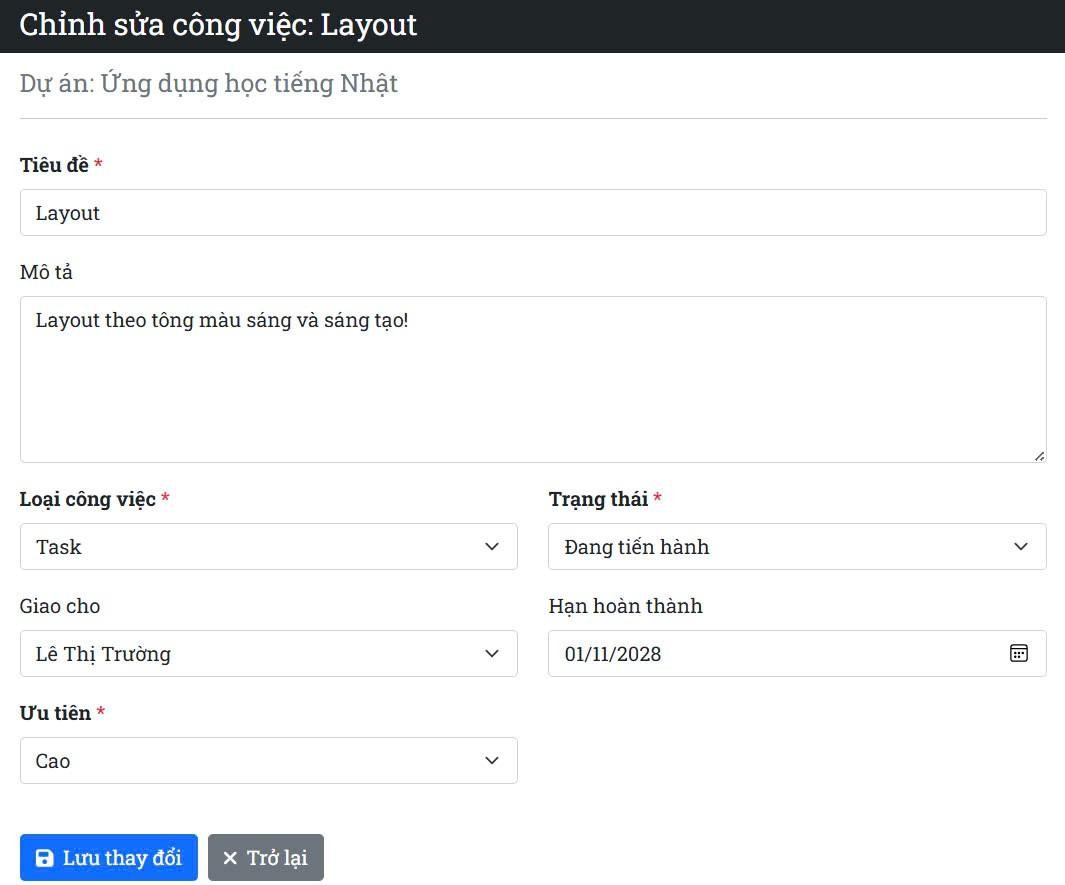
****

*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

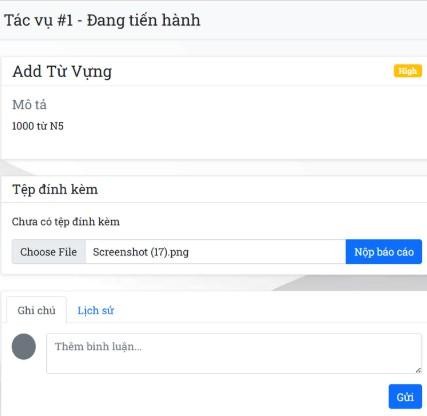
****

*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

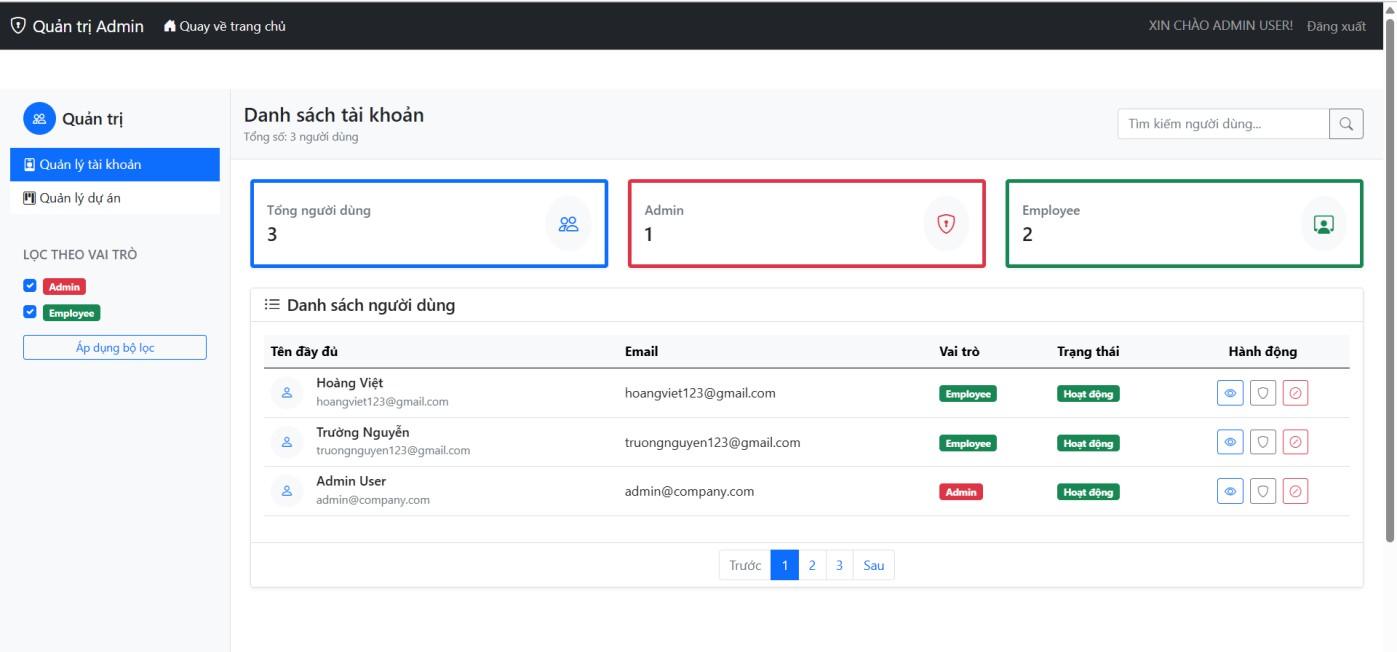
****

*Hình 5.11: Giao diện nộp báo cáo.*

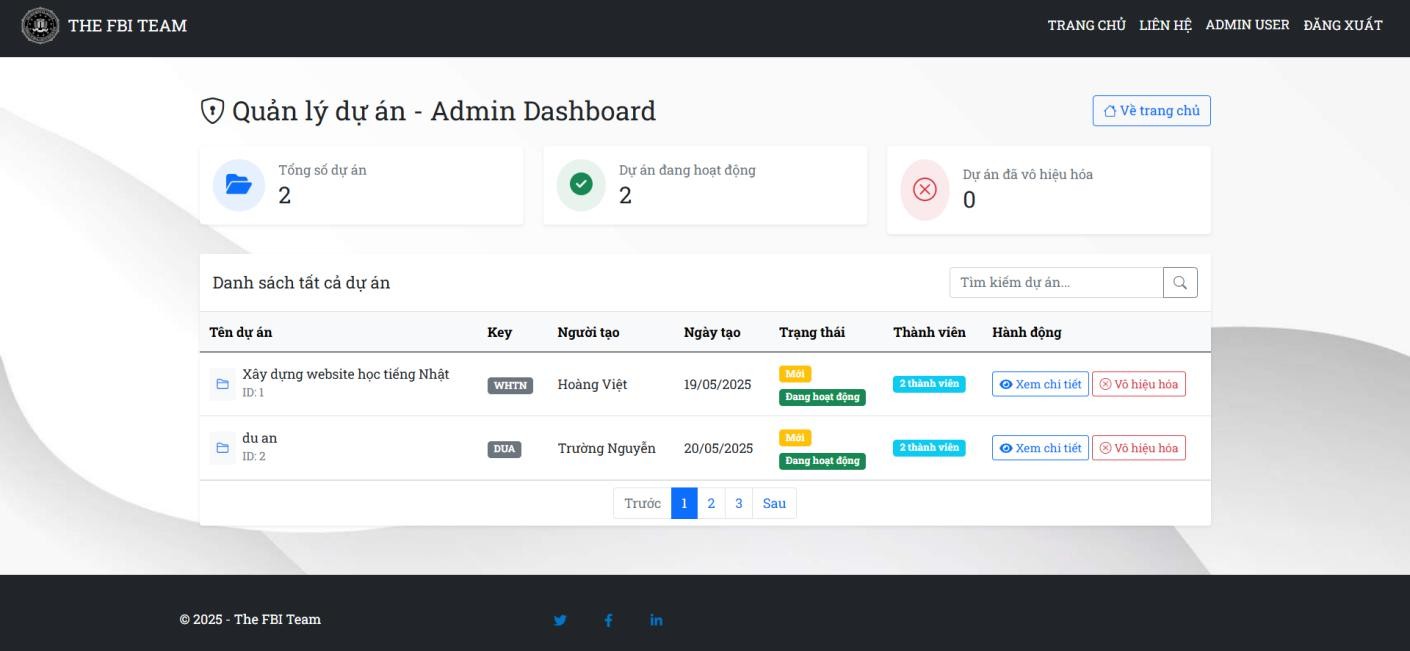
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thao tác** | **Ý nghĩa** | **Màu sắc** | **Ý nghĩa màu sắc** |
| 1 | Công việc | Tiêu đề của công việc cần sửa | - | - |
| 2 | Chọn file | Thêm file cần báo cáo | - | - |
| 3 | Thời gian nộp/cập nhật lúc | Thời gian nộp hay chỉnh sửa báo 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.*

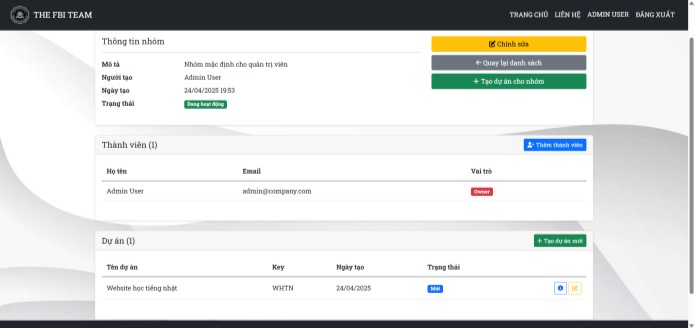
**

*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

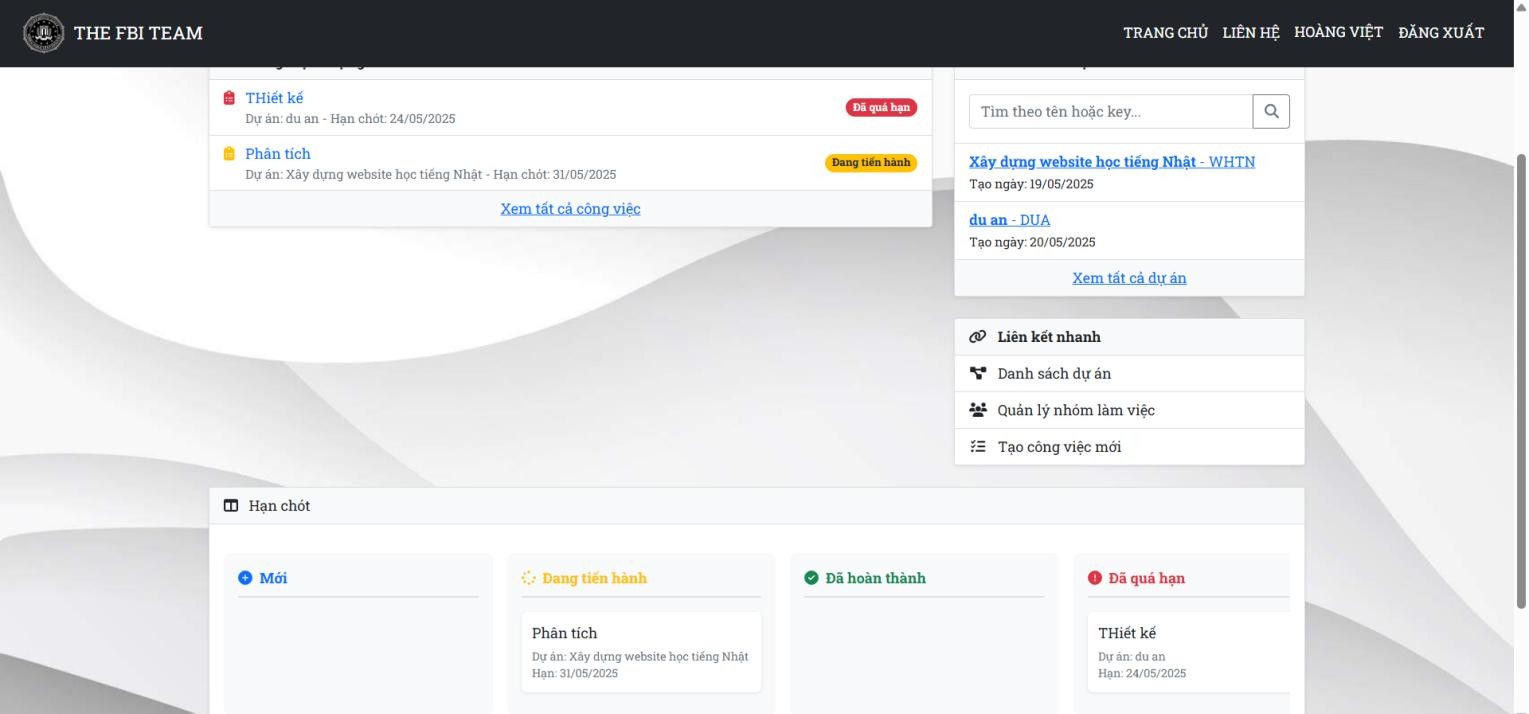


*Hình 5.14: Giao diện chức năng quản lý nhóm làm việc.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

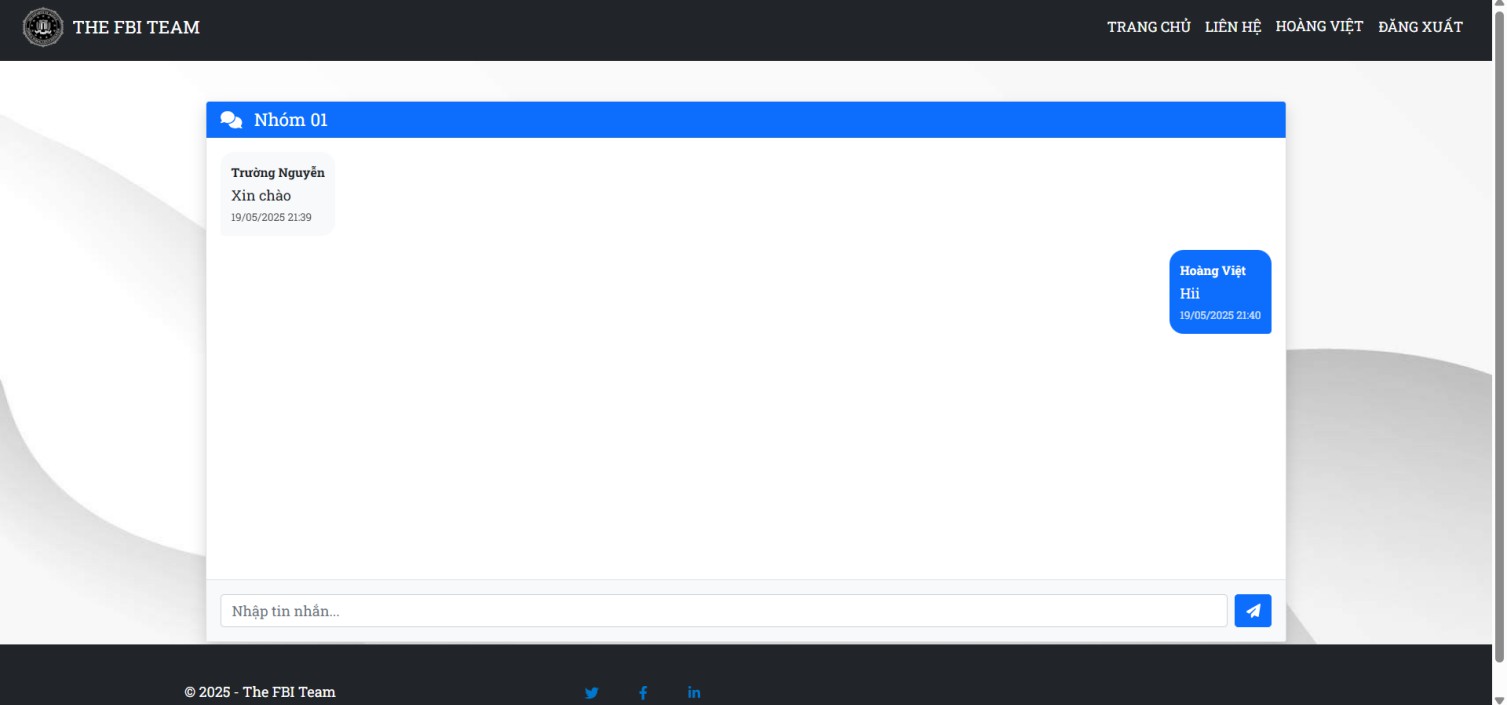


*Hình 5.15: Giao diện chức năng kéo thả thay đổi trạng thái công việc.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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)