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

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**-----**--------------------------------

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ NHÂN VIÊN**

**GVHD :** TS. Lương Thị Hồng Lan

**Sinh viên :** Dương Bá Trung Kiên

**Mã sinh viên :** 2021600212

**Hà Nội – 2025**

# LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới tập thể thầy cô trong Trường Công Nghệ Thông Tin và Truyền Thông, những người đã luôn đồng hành, giảng dạy và truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt quá trình học tập và nghiên cứu tại trường.

Đặc biệt, em xin gửi lời tri ân chân thành tới cô TS. Lương Thị Hồng Lan, người đã tận tình hướng dẫn, dành nhiều thời gian và tâm huyết để định hướng, góp ý và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn đến toàn thể thầy cô Trường Đại học Công nghiệp Hà Nội đã tạo điều kiện, chia sẻ kiến thức và kinh nghiệm, giúp em có nền tảng vững chắc để triển khai và hoàn thiện đề tài: *“****Xây dựng Hệ thống Quản Lý Nhân Viên****”*.

Mặc dù đã nỗ lực hết mình, song do hạn chế về thời gian và kinh nghiệm, đồ án khó tránh khỏi những thiếu sót. Em kính mong nhận được những ý kiến đóng góp quý báu từ thầy cô và các bạn để đề tài được hoàn thiện hơn, đồng thời giúp em có thêm động lực và kinh nghiệm trên con đường học tập, nghiên cứu và làm việc sau này.

Em xin trân trọng cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc14424)

[MỤC LỤC 2](#_Toc32063)

[DANH MỤC HÌNH ẢNH 4](#_Toc14135)

[DANH MỤC THUẬT NGỮ VIẾT TẮT 5](#_Toc20085)

[LỜI MỞ ĐẦU 6](#_Toc23482)

[CHƯƠNG 1: KIẾN THỨC CƠ SỞ 13](#_Toc5025)

[1.1 Tổng quan về hệ thống quản lý nhân sự 13](#_Toc701)

[*1.1.1 Khái niệm hệ thống quản lý nhân sự là gì 13*](#_Toc3874)

[*1.1.2 Khảo sát hệ thống quản lý nhân sự 14*](#_Toc16948)

[*1.1.3 Phân tích yêu cầu chức năng 16*](#_Toc12270)

[*1.1.4 Phân tích yêu cầu phi chức năng 19*](#_Toc25312)

[1.2 Công nghệ sử dụng 21](#_Toc25302)

[*1.2.1 React 21*](#_Toc21255)

[*1.2.2 Node.js 23*](#_Toc9041)

[*1.2.3 Express.js 25*](#_Toc15244)

[*1.2.4 MongoDB 26*](#_Toc4868)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 29](#_Toc14806)

[2.1 Phân tích use case 29](#_Toc21668)

[2.2 Đặc tả usecase 30](#_Toc20889)

[*2.2.1 Use case đăng nhập 30*](#_Toc16649)

[*2.2.2 Use case quản lý chức vụ 32*](#_Toc5249)

[*2.2.3 Use case quản lý phòng ban 34*](#_Toc30386)

[*2.2.4 Use case quản lý khen thưởng, kỷ luật 36*](#_Toc13444)

[*2.2.5 Use case quản lý người dùng 38*](#_Toc8014)

[*2.2.7 Use case quản lý đơn xin nghỉ phép 42*](#_Toc15843)

[*2.2.8 Use case quản lý dự án 44*](#_Toc8764)

[*2.2.9 Use case quản lý thông tin cá nhân 46*](#_Toc24087)

[*2.2.10 Use case quản lý học vấn 48*](#_Toc13378)

[*2.2.11 Use case quản lý thông tin gia đình 50*](#_Toc1852)

[*2.2.12 Use case quản lý kinh nghiệm làm việc 52*](#_Toc30161)

[*2.2.13 Use case quản lý nghỉ phép 54*](#_Toc31670)

[*2.2.14 Use case lịch sử nhận lương 56*](#_Toc7308)

[*2.2.15 Use case quản lý khiếu nại 58*](#_Toc27687)

[2.3 Thiết kế cơ sở dữ liệu 60](#_Toc18929)

[CHƯƠNG 3: THỰC NGHIỆM 62](#_Toc25072)

[3.1 Quy trình kiểm thử 62](#_Toc79)

[3.2 Kết quả kiểm thử 63](#_Toc17101)

[3.3 Kết quả 64](#_Toc28786)

[KẾT LUẬN 74](#_Toc24148)

[TÀI LIỆU THAM KHẢO 78](#_Toc28076)

# DANH MỤC HÌNH ẢNH

[Hình 2.2 Biểu đồ trình tự Use case đăng nhập 31](#_Toc15449)

[Hình 2.3 Biểu đồ Use case quản lý chức vụ 33](#_Toc803)

[Hình 2.4 Biểu đồ Use case quản lý phòng ban 35](#_Toc23430)

[Hình 2.5 Biểu đồ Use case quản lý khen thưởng, kỷ luật 37](#_Toc15497)

[Hình 2.6 Biểu đồ Use case quản lý người dùng 39](#_Toc18860)

[Hình 2.7 Biểu đồ Use case quản lý lương của quản lý 41](#_Toc1456)

[Hình 2.8 Biểu đồ Use case quản lý đơn xin nghỉ phép 43](#_Toc2962)

[Hình 2.9 Biểu đồ Use case quản lý dự án 45](#_Toc24465)

[Hình 2.10 Biểu đồ Use case quản lý thông tin cá nhân 47](#_Toc14417)

[Hình 2.11 Biểu đồ Use case quản lý học vấn 49](#_Toc3199)

[Hình 2.12 Biểu đồ Use case quản lý thông tin gia đình 51](#_Toc18044)

[Hình 2.13 Biểu đồ Use case quản lý kinh nghiệm làm việc 53](#_Toc7091)

[Hình 2.14 Biểu đồ Use case quản lý nghỉ phép 55](#_Toc27766)

[Hình 2.15 Biểu đồ Use case lịch sử nhận lương 57](#_Toc15144)

[Hình 2.16 Use case quản lý khiếu nại 59](#_Toc13713)

[Hình 2.17 Thiết kế CSDL trong mongoDB 61](#_Toc22157)

[Hình 3.1: Giao diện đăng nhập 66](#_Toc14303)

[Hình 3.2: Giao diện quản lý chức vụ 67](#_Toc10311)

[Hình 3.3: Giao diện quản lý lương 67](#_Toc13929)

[Hình 3.4: Giao diện quản lý phòng ban 68](#_Toc22691)

[Hình 3.5: Giao diện quản lý dự án 68](#_Toc3079)

[Hình 3.6: Giao diện quản lý khiếu nại 69](#_Toc25555)

[Hình 3.7: Giao diện quản lý người dùng 69](#_Toc1525)

[Hình 3.8: Giao diện quản lý đơn xin nghỉ phép 70](#_Toc25776)

[Hình 3.9: Giao diện quản lý khen thưởng, kỷ luật 70](#_Toc29707)

[Hình 3.12 Giao diện quản lý thông tin gia đình 72](#_Toc4748)

[Hình 3.13 Giao diện quản lý thông tin kinh nghiệm làm việc 72](#_Toc9155)

[Hình 3.14 Giao diện tạo đơn xin nghỉ phép 73](#_Toc29343)

[Hình 3.15 Giao diện tạo đơn nghỉ phép 73](#_Toc25720)

[Hình 3.17 Giao diện thông tin khen thưởng/kỷ luật 74](#_Toc468)

[Hình 3.18 Giao diện lịch sử nhận lương 75](#_Toc26236)

# DANH MỤC THUẬT NGỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Tên** | **Mô tả** |
| XSS | Cross-Site Scripting |
| CSRF | Cross-Site Request Forgery |
| UI | Giao diện người dùng |
| non-blocking I/O | Xử lý bất đồng bộ |
| JWT | Xử lý xác thực bằng token jsonwebtoken |
| Server | Máy chủ của phần mềm |
| API | Phương thức giao tiếp với ứng dụng |

# LỜI MỞ ĐẦU

**Tổng quan đề tài**

Trong bối cảnh các doanh nghiệp hiện đại ngày càng mở rộng quy mô và phát triển, việc quản lý nhân sự hiệu quả là yếu tố quan trọng giúp tối ưu hóa quy trình làm việc và nâng cao năng suất. Hệ thống quản lý nhân viên không chỉ hỗ trợ việc lưu trữ, xử lý thông tin nhân viên mà còn giúp quản lý các quy trình liên quan như quản lý lương, nghỉ phép, kinh nghiệm làm việc, và khen thưởng, kỷ luật của từng nhân viên.

Sự phát triển mạnh mẽ của công nghệ web cùng với các hệ quản trị cơ sở dữ liệu hiện đại đã tạo điều kiện thuận lợi cho việc xây dựng các hệ thống quản lý nhân sự trực tuyến. Những hệ thống này giúp doanh nghiệp dễ dàng quản lý, theo dõi và cập nhật thông tin từ xa, đồng thời hạn chế tối đa các sai sót thường gặp trong quá trình xử lý thủ công.

Trong bối cảnh đó, dự án được triển khai với mục tiêu phát triển một Hệ thống Quản lý Nhân sự ứng dụng các công nghệ tiên tiến như ReactJS cho giao diện người dùng, Node.js và Express cho tầng xử lý nghiệp vụ phía máy chủ, kết hợp với MongoDB để lưu trữ dữ liệu. Hệ thống được thiết kế nhằm cung cấp một nền tảng quản lý toàn diện, hỗ trợ ba nhóm đối tượng chính trong doanh nghiệp: Quản trị viên, Quản lý nhân sự và Nhân viên.

Thông qua hệ thống, Quản trị viên có thể quản lý các thông tin liên quan đến các phòng ban, chức vụ, nhân viên và dự án của công ty. Quản lý nhân sự có thể thực hiện các chức năng như quản lý thông tin nhân viên, lương thưởng, và duyệt đơn xin nghỉ phép. Nhân viên cũng có thể tự quản lý và cập nhật thông tin cá nhân, học vấn, kinh nghiệm làm việc, theo dõi lương hàng tháng và đơn xin nghỉ phép của mình một cách dễ dàng.

Việc áp dụng hệ thống này không chỉ giúp tự động hóa các quy trình quản lý nhân sự, mà còn góp phần nâng cao tính minh bạch và hiệu quả trong hoạt động doanh nghiệp. Hệ thống cung cấp giao diện người dùng thân thiện, dễ sử dụng và có khả năng mở rộng trong tương lai để phù hợp với nhu cầu phát triển của doanh nghiệp.

**Lý do chọn đề tài**

Trong quá trình học tập và tìm hiểu về các hệ thống thông tin quản lý, đặc biệt là quản lý nhân sự, em nhận thấy rằng nhu cầu quản lý nhân viên một cách hiệu quả luôn là bài toán quan trọng đối với các doanh nghiệp. Một hệ thống quản lý nhân sự không chỉ đơn thuần là lưu trữ thông tin mà còn hỗ trợ các quy trình quản lý phức tạp như theo dõi lương thưởng, quản lý nghỉ phép, đánh giá hiệu quả làm việc, và quản lý hồ sơ nhân viên một cách chi tiết và chính xác.

Thực tế cho thấy, nhiều doanh nghiệp hiện nay vẫn áp dụng phương pháp quản lý nhân sự thủ công hoặc sử dụng các hệ thống đã lỗi thời, thiếu tính linh hoạt. Điều này không chỉ tiêu tốn nhiều thời gian và công sức mà còn tiềm ẩn nguy cơ sai sót trong quá trình nhập liệu và xử lý thông tin. Do đó, việc phát triển một hệ thống quản lý nhân sự trực tuyến với giao diện thân thiện, thao tác nhanh chóng và tích hợp các công nghệ hiện đại như React, Node.js và MongoDB là yêu cầu cấp thiết, mang tính thực tiễn cao và phù hợp với xu thế chuyển đổi số trong doanh nghiệp.

Bên cạnh ý nghĩa thực tiễn, đề tài còn tạo cơ hội để em vận dụng những kiến thức đã được học vào quá trình xây dựng sản phẩm thực tế, từ khâu thiết kế cơ sở dữ liệu, phát triển backend cho đến triển khai giao diện người dùng. Thông qua đó, em không chỉ đáp ứng được nhu cầu quản lý nhân sự của doanh nghiệp, mà còn nâng cao kỹ năng lập trình web full-stack, rèn luyện năng lực phân tích và giải quyết các bài toán quản lý thông tin phức tạp.

Hơn nữa, với sự phát triển của công nghệ thông tin, việc xây dựng các hệ thống quản lý thông tin trên nền tảng web đang trở thành xu hướng tất yếu. Đề tài này không chỉ giúp em làm quen với các công nghệ web mới nhất mà còn có khả năng mở rộng và áp dụng vào thực tế ở các doanh nghiệp có quy mô lớn hơn trong tương lai. Đây chính là động lực quan trọng thúc đẩy em chọn đề tài này để nghiên cứu và phát triển.

**Mục tiêu của đề tài**

* + 1. Xây dựng một hệ thống quản lý nhân sự trực tuyến hiệu quả:
* Hệ thống cung cấp đầy đủ các tính năng cần thiết để quản lý thông tin nhân viên, bao gồm thông tin cá nhân, học vấn, kinh nghiệm làm việc, và lịch sử lương thưởng.
* Tạo ra một nền tảng dễ sử dụng, cho phép các đối tượng như Quản trị viên, quản lý nhân sự và nhân viên thực hiện các tác vụ quản lý liên quan đến chức vụ, phòng ban, dự án, và nghỉ phép.
  + 1. Tự động hóa các quy trình quản lý nhân sự:
* Tối ưu hóa quy trình nhập liệu, xử lý thông tin nhân viên nhằm giảm thiểu sai sót trong quá trình quản lý thủ công.
* Cung cấp tính năng duyệt hoặc từ chối các yêu cầu nghỉ phép và các thao tác quản lý lương, giúp quá trình xử lý diễn ra nhanh chóng, chính xác.
  + 1. Đảm bảo tính bảo mật và phân quyền:
* Xây dựng hệ thống bảo mật đăng nhập, chỉ cho phép các đối tượng có thẩm quyền truy cập và quản lý các thông tin phù hợp với vai trò của họ.
* Phân quyền rõ ràng giữa các đối tượng Quản trị viên, quản lý nhân sự và nhân viên, giúp đảm bảo tính an toàn và bảo mật của dữ liệu.
  + 1. Áp dụng các công nghệ hiện đại:
* Ứng dụng các công nghệ web tiên tiến như ReactJS để xây dựng giao diện người dùng linh hoạt và thân thiện.
* Sử dụng Nodejs và Express để xây dựng backend mạnh mẽ, kết hợp với MongoDB để quản lý cơ sở dữ liệu một cách hiệu quả, đảm bảo khả năng mở rộng và quản lý dữ liệu lớn.
  + 1. Tạo điều kiện cho các doanh nghiệp quản lý nhân sự từ xa:
* Hệ thống được thiết kế để hoạt động trực tuyến, giúp các doanh nghiệp dễ dàng truy cập, quản lý thông tin nhân sự mọi lúc, mọi nơi, từ đó nâng cao hiệu quả quản lý và khả năng phản hồi nhanh chóng.

**Giới hạn và phạm vi đề tài**

**Phạm vi đề tài**

Đề tài được phát triển nhằm đáp ứng nhu cầu quản lý nhân sự của các doanh nghiệp với các tính năng chính bao gồm:

1. Quản lý thông tin nhân viên:
   * + - Hệ thống cho phép lưu trữ và quản lý các thông tin cá nhân, học vấn, kinh nghiệm làm việc, và thông tin gia đình của từng nhân viên.
       - Mỗi nhân viên có quyền truy cập và chỉnh sửa thông tin cá nhân của mình.
2. Quản lý chức vụ, phòng ban, và dự án:
   * + - Quản trị viên và quản lý nhân sự có thể thực hiện các thao tác thêm, sửa, xóa dữ liệu liên quan đến các chức vụ, phòng ban và dự án trong công ty.
3. Quản lý lương và nghỉ phép:
   * + - Hệ thống hỗ trợ quản lý nhân sự trong việc cập nhật và quản lý lương cho nhân viên, cũng như duyệt hoặc từ chối các yêu cầu nghỉ phép của nhân viên.
4. Đăng nhập và phân quyền:
   * + - Hệ thống cho phép đăng nhập bằng tài khoản cá nhân và phân quyền theo vai trò (Quản trị viên, quản lý nhân sự, nhân viên) nhằm đảm bảo tính bảo mật và giới hạn quyền truy cập vào các chức năng phù hợp với từng vai trò.

**Giới hạn của đề tài**

Mặc dù hệ thống cung cấp đầy đủ các tính năng cơ bản cho việc quản lý nhân sự, do giới hạn về thời gian và phạm vi thực hiện, đề tài có một số giới hạn nhất định:

1. Chưa tích hợp các tính năng nâng cao như đánh giá hiệu suất:
   * Hệ thống hiện tại chỉ hỗ trợ các chức năng quản lý thông tin cơ bản như quản lý nhân viên, lương, nghỉ phép, phòng ban và dự án, nhưng chưa bao gồm các tính năng nâng cao như đánh giá hiệu suất làm việc hay phân tích dữ liệu.
2. Chưa tối ưu cho quy mô doanh nghiệp lớn:
   * Hệ thống được thiết kế cho doanh nghiệp có quy mô nhỏ và vừa, chưa có các tính năng phức tạp cho các doanh nghiệp quy mô lớn với hàng ngàn nhân viên. Các tính năng mở rộng như quản lý phúc lợi, bảo hiểm, và các yếu tố quản lý nhân sự phức tạp hơn chưa được tích hợp.
3. Giới hạn về tính năng báo cáo:
   * Hệ thống hiện chỉ hỗ trợ hiển thị dữ liệu cơ bản dưới dạng bảng, chưa phát triển các công cụ báo cáo trực quan, biểu đồ phân tích dữ liệu phức tạp phục vụ quản lý cấp cao.
4. Chưa hỗ trợ tích hợp với các hệ thống khác:
   * Do giới hạn về thời gian và phạm vi phát triển, hệ thống chưa được tích hợp với các hệ thống quản lý tài chính, kế toán, hay các phần mềm quản lý nhân sự khác mà doanh nghiệp có thể đang sử dụng.
5. Giới hạn về ngôn ngữ và đa nền tảng:
   * Hiện tại, hệ thống chỉ hỗ trợ ngôn ngữ tiếng Việt và chưa được tối ưu cho các ngôn ngữ khác hoặc các nền tảng di động, mặc dù vẫn có khả năng mở rộng trong tương lai.

**Kết quả dự kiến đạt được**

Sau khi hoàn thành Đề tài, em dự kiến sẽ đạt được các kết quả sau:

Xây dựng thành công hệ thống quản lý nhân viên trực tuyến: Hệ thống hoạt động ổn định, đáp ứng được các yêu cầu quản lý thông tin nhân sự, lương, phòng ban, dự án và nghỉ phép. Giao diện người dùng trực quan, dễ sử dụng, phù hợp với các đối tượng Quản trị viên, quản lý nhân sự, và Nhân viên.

Tự động hóa quy trình quản lý nhân sự: Thông tin nhân viên được lưu trữ và quản lý một cách hệ thống và chặt chẽ. Các quy trình như thêm mới, chỉnh sửa, xóa thông tin nhân viên, duyệt đơn nghỉ phép, và quản lý lương được tự động hóa, giúp giảm thiểu thời gian và công sức trong việc quản lý thủ công.

Hỗ trợ phân quyền và bảo mật dữ liệu: Hệ thống có khả năng phân quyền rõ ràng giữa các vai trò khác nhau. Chỉ những người có thẩm quyền mới có thể truy cập và chỉnh sửa thông tin quan trọng, đảm bảo tính bảo mật của dữ liệu.

Tăng cường tính minh bạch và hiệu quả quản lý: Thông tin nhân viên được cập nhật kịp thời và chính xác, giúp các nhà quản lý có thể dễ dàng theo dõi tình hình nhân sự trong công ty. Hệ thống cũng cung cấp tính năng ghi nhận và lưu trữ lịch sử các thao tác quản lý, đảm bảo tính minh bạch.

Nâng cao khả năng quản lý từ xa: Với việc xây dựng trên nền tảng web, hệ thống cho phép truy cập từ mọi nơi, tạo điều kiện thuận lợi cho doanh nghiệp trong việc quản lý nhân sự từ xa, đặc biệt trong bối cảnh công việc linh hoạt và làm việc từ xa ngày càng phổ biến.

Cơ sở để mở rộng và tích hợp trong tương lai: Hệ thống được thiết kế với kiến trúc mở, dễ dàng nâng cấp và tích hợp thêm các tính năng mới trong tương lai như quản lý đánh giá hiệu suất nhân viên, báo cáo phân tích dữ liệu, và tích hợp với các hệ thống quản lý khác (tài chính, kế toán).

# CHƯƠNG 1: KIẾN THỨC CƠ SỞ

## 1.1 Tổng quan về hệ thống quản lý nhân sự

### 1.1.1 Khái niệm hệ thống quản lý nhân sự là gì

Hệ thống quản lý nhân sự (Human Resource Management System – HRMS) là tập hợp các quy trình, công cụ và công nghệ hỗ trợ doanh nghiệp trong việc quản lý toàn bộ vòng đời của nhân viên, từ khi tuyển dụng, đào tạo, sử dụng, đánh giá cho đến khi nghỉ việc.

Đây có thể là hệ thống thủ công (sổ sách, biểu mẫu) hoặc hệ thống số hóa (phần mềm HRM/HRIS) nhằm tối ưu hóa việc quản trị con người.

Một hệ thống HRMS có thể tồn tại dưới hai hình thức chính:

* **Hệ thống thủ công**: Doanh nghiệp quản lý hồ sơ bằng sổ sách, giấy tờ hoặc các biểu mẫu Excel. Cách này phù hợp với quy mô nhỏ nhưng thường gây tốn thời gian, dễ sai sót, khó tra cứu và thiếu tính bảo mật.
* **Hệ thống số hóa (HRM/HRIS – Human Resource Information System)**: Đây là các phần mềm quản lý nhân sự hiện đại, cho phép tự động hóa nhiều nghiệp vụ nhân sự, tích hợp các chức năng chấm công – tính lương, quản lý hồ sơ tập trung, đánh giá KPI, quản lý phúc lợi, và cung cấp cổng thông tin cho nhân viên.

Việc ứng dụng HRMS trong doanh nghiệp mang lại nhiều lợi ích rõ rệt:

* **Tối ưu hóa hiệu quả quản lý nguồn nhân lực**: Giảm thiểu các công việc thủ công lặp lại, tăng tốc độ xử lý dữ liệu.
* **Nâng cao trải nghiệm của nhân viên**: Nhân viên có thể chủ động tra cứu bảng lương, lịch sử làm việc, hoặc gửi yêu cầu nghỉ phép trực tuyến.
* **Hỗ trợ ra quyết định quản trị**: Các báo cáo, thống kê từ HRMS giúp lãnh đạo có cái nhìn toàn diện về tình hình nhân sự, từ đó đưa ra chiến lược phù hợp.
* **Đảm bảo tính minh bạch và bảo mật**: Dữ liệu được lưu trữ tập trung, phân quyền chặt chẽ, giảm rủi ro thất lạc hoặc lộ thông tin.

### 1.1.2 Khảo sát hệ thống quản lý nhân sự

1. **Thực trạng khảo sát**

Hiện nay, nhiều doanh nghiệp vẫn quản lý nhân sự bằng Excel hoặc hồ sơ giấy tờ thủ công, dẫn đến khó khăn trong việc lưu trữ, tra cứu, tính toán và bảo mật thông tin.

Trên thị trường cũng đã có các phần mềm quản lý nhân sự hỗ trợ doanh nghiệp với nhiều tính năng khác nhau. Tuy nhiên, qua khảo sát và nghiên cứu, em nhận thấy một số chức năng nổi bật và thực sự cần thiết đối với doanh nghiệp Việt Nam hiện nay, cụ thể là:

* **Quản lý hồ sơ nhân viên tập trung**: Lưu trữ toàn bộ thông tin cá nhân, hợp đồng, lịch sử làm việc và quá trình khen thưởng/kỷ luật.
* **Quản lý chức vụ và phòng ban**: Theo dõi, sắp xếp cơ cấu tổ chức, chức danh và mối quan hệ giữa các phòng ban trong doanh nghiệp.
* **Quản lý khen thưởng – kỷ luật**: Ghi nhận, lưu trữ các quyết định khen thưởng hoặc kỷ luật đối với nhân viên.
* **Quản lý người dùng**: Hỗ trợ phân quyền, tạo tài khoản, đảm bảo an toàn và bảo mật dữ liệu hệ thống.
* **Quản lý đơn xin nghỉ phép, nghỉ việc**: Nhân viên có thể gửi đơn trực tuyến; cấp quản lý duyệt nhanh chóng và minh bạch.
* **Quản lý dự án**: Theo dõi dự án mà nhân viên tham gia, phân công công việc và đánh giá tiến độ.
* **Quản lý thông tin cá nhân – học vấn – kinh nghiệm làm việc – gia đình**: Lưu trữ thông tin chi tiết về trình độ học vấn, kỹ năng, kinh nghiệm nghề nghiệp và thông tin gia đình của nhân viên.
* **Quản lý chấm công – tính lương – lịch sử nhận lương**: Hệ thống tự động tổng hợp dữ liệu chấm công, tính lương chính xác và lưu vết lịch sử các kỳ trả lương.
* **Đánh giá hiệu suất làm việc (KPI/OKR)**: Cung cấp công cụ hỗ trợ theo dõi, đánh giá năng suất và kết quả công việc của nhân viên.
* **Cổng thông tin nhân viên (Self-Service)**: Cho phép nhân viên chủ động tra cứu thông tin cá nhân, bảng lương, lịch sử công tác, giảm tải cho bộ phận HR.

1. **Kết quả khảo sát**

Để đánh giá nhu cầu thực tế, em đã thực hiện khảo sát thử nghiệm với 10 doanh nghiệp vừa và nhỏ (thuộc các lĩnh vực thương mại, dịch vụ và sản xuất) cùng với 15 nhân viên/phòng nhân sự. Kết quả thu được như sau:

* 40% doanh nghiệp cho biết hiện vẫn sử dụng Excel hoặc hồ sơ giấy tờ thủ công để quản lý nhân sự.
* 30% doanh nghiệp đã sử dụng phần mềm nhân sự đơn lẻ, nhưng phản ánh rằng phần mềm còn thiếu các chức năng nâng cao như chấm công – tính lương tự động hoặc quản lý đơn nghỉ phép.
* 30% doanh nghiệp triển khai hệ thống ERP có module HRM, tuy nhiên chi phí đầu tư và bảo trì cao, khó phù hợp với quy mô nhỏ.

Về nhu cầu chức năng:

* 75% mong muốn có chức năng chấm công – tính lương tự động để giảm sai sót.
* 60% quan tâm đến quản lý đơn nghỉ phép, nghỉ việc trực tuyến.
* 55% cần cổng thông tin nhân viên (Self-service) để tra cứu thông tin cá nhân và bảng lương.
* 50% đánh giá cao tính năng quản lý khen thưởng – kỷ luật minh bạch.

**Kết luận:**

Qua khảo sát, có thể nhận thấy nhiều doanh nghiệp Việt Nam, đặc biệt là doanh nghiệp vừa và nhỏ, vẫn đang gặp khó khăn trong việc quản lý nhân sự. Các tính năng được quan tâm nhiều nhất là: chấm công – tính lương, quản lý hồ sơ tập trung, đơn nghỉ phép trực tuyến và cổng thông tin nhân viên. Đây là cơ sở quan trọng để định hướng xây dựng hệ thống quản lý nhân sự trong đề tài.

### 1.1.3 Phân tích yêu cầu chức năng

**Phân tích đối tượng**

* + - * Trong hệ thống quản lý nhân viên, các đối tượng chính tham gia vào hệ thống bao gồm Quản trị viên, quản lý nhân sự, và Nhân viên. Mỗi đối tượng có những nhiệm vụ và quyền hạn riêng, đảm bảo hệ thống hoạt động hiệu quả và phân quyền phù hợp. Dưới đây là phân tích chi tiết về vai trò của từng đối tượng:
      * Quản trị viên là người có quyền cao nhất trong hệ thống, chịu trách nhiệm quản lý tổng quan toàn bộ các hoạt động của hệ thống và đảm bảo tính an toàn, ổn định của hệ thống.
      * Quản lý nhân sự là người chịu trách nhiệm quản lý trực tiếp các thông tin liên quan đến nhân viên và các chính sách nhân sự trong công ty. Đây là vai trò quan trọng giúp đảm bảo hệ thống nhân sự vận hành suôn sẻ.
      * Nhân viên là đối tượng chính được quản lý trong hệ thống. Mỗi nhân viên có quyền truy cập vào hệ thống để theo dõi và quản lý thông tin cá nhân của mình, đồng thời tương tác với các chức năng khác liên quan đến công việc

**Phân tích chức năng**

**Quản trị viên:**

* + - * Đăng nhập: người dùng đăng nhập vào hệ thống bằng tài khoản được cấp sẵn để vào trang hệ thống với vai trò của họ
      * Quản lý chức vụ: Thực hiện các thao tác thêm mới, sửa đổi, hoặc xóa dữ liệu về các chức vụ trong công ty.
      * Quản lý khen thưởng, kỷ luật: Thêm, sửa, xóa các thông tin liên quan đến khen thưởng, kỷ luật của nhân viên, đảm bảo cập nhật đúng chuyên ngành cho từng vị trí.
      * Quản lý phòng ban: Theo dõi và quản lý các phòng ban trong công ty, thực hiện thêm, sửa hoặc xóa thông tin phòng ban.
      * Quản lý dự án: Theo dõi các dự án hiện tại của công ty, tạo mới, chỉnh sửa, hoặc xóa thông tin dự án.

**Quản lý nhân sự:**

* + - * Đăng nhập: người dùng đăng nhập vào hệ thống bằng tài khoản được cấp sẵn để vào trang hệ thống với vai trò của họ
      * Quản lý người dùng (nhân viên): Thực hiện các thao tác thêm, sửa, xóa thông tin của nhân viên trong hệ thống, bao gồm thông tin cá nhân, lương thưởng, khen thưởng, kỷ luật, và phòng ban.
      * Quản lý lương: Cập nhật và quản lý thông tin về mức lương của nhân viên, theo dõi sự thay đổi và điều chỉnh lương khi cần thiết.
      * Quản lý đơn xin nghỉ phép: Phê duyệt hoặc từ chối các yêu cầu nghỉ phép của nhân viên, lưu trữ và theo dõi các lịch sử nghỉ phép.
      * Quản lý thông tin công ty: Cập nhật thông tin về công ty, bao gồm địa chỉ, quy mô, chính sách, và các thông tin liên quan khác.
      * Quản lý chức vụ, khen thưởng, kỷ luật, phòng ban: Giống như Quản trị viên, quản lý nhân sự cũng có quyền thao tác với các dữ liệu này nhằm đảm bảo tổ chức nhân sự được phân bổ hợp lý.
      * Quản lý địa chỉ: Thêm, sửa, xóa thông tin về địa chỉ công ty, bao gồm quốc gia, khu vực, thành phố.

**Nhân viên:**

* + - * Quản lý thông tin cá nhân: Nhân viên có thể tự chỉnh sửa các thông tin cá nhân của mình trong hệ thống, bao gồm thông tin liên hệ, địa chỉ, và các dữ liệu cơ bản khác.
      * Quản lý học vấn: Thực hiện thêm, sửa, xóa các thông tin về quá trình học vấn và khen thưởng, kỷ luật của bản thân, giúp hồ sơ luôn được cập nhật chính xác.
      * Quản lý thông tin gia đình: Cập nhật các thông tin về gia đình của mình, giúp quản lý dễ dàng theo dõi tình trạng của nhân viên.
      * Quản lý kinh nghiệm làm việc: Nhân viên có thể thêm, sửa, hoặc xóa các thông tin về kinh nghiệm làm việc trước đây của mình.
      * Quản lý nghỉ phép: Nhân viên có thể tạo yêu cầu nghỉ phép, theo dõi trạng thái yêu cầu (đã được duyệt hoặc từ chối), và cập nhật thông tin nghỉ phép.

### 1.1.4 Phân tích yêu cầu phi chức năng

Yêu cầu phi chức năng là những yêu cầu liên quan đến hiệu suất, bảo mật, khả năng mở rộng và các yếu tố không trực tiếp liên quan đến chức năng của hệ thống. Đối với hệ thống quản lý nhân viên, các yêu cầu phi chức năng quan trọng bao gồm:

1. Hiệu suất (Performance)
   * + - Tốc độ xử lý nhanh: Hệ thống cần đảm bảo tốc độ phản hồi nhanh khi thực hiện các thao tác như đăng nhập, thêm, sửa, xóa thông tin nhân viên, và truy vấn dữ liệu từ cơ sở dữ liệu. Thời gian xử lý các yêu cầu của người dùng không quá 3 giây đối với hầu hết các thao tác.
       - Khả năng chịu tải: Hệ thống phải có khả năng xử lý đồng thời nhiều người dùng (Quản trị viên, quản lý nhân sự, nhân viên) truy cập và thao tác mà không gây ra hiện tượng treo hoặc chậm trễ. Hệ thống cần hỗ trợ ít nhất 100 người dùng đồng thời mà không ảnh hưởng đến hiệu suất.
2. Bảo mật (Security)
   * + - Xác thực và phân quyền người dùng: Hệ thống phải đảm bảo rằng mỗi người dùng chỉ có thể truy cập và thực hiện các thao tác theo đúng quyền hạn của họ (Quản trị viên, quản lý nhân sự, nhân viên). Mỗi người dùng phải có tài khoản riêng và hệ thống xác thực danh tính qua mật khẩu an toàn.
       - Mã hóa dữ liệu nhạy cảm: Các thông tin nhạy cảm như mật khẩu, thông tin cá nhân, lương của nhân viên phải được mã hóa và bảo mật, tránh rò rỉ thông tin khi hệ thống bị tấn công.
       - Chống tấn công: Hệ thống phải được bảo vệ khỏi các cuộc tấn công phổ biến như tấn công SQL Injection, Cross-Site Scripting (XSS), và Cross-Site Request Forgery (CSRF) nhằm đảm bảo an toàn cho dữ liệu.
3. Khả năng mở rộng (Scalability)
   * + - Khả năng mở rộng theo quy mô: Hệ thống phải được thiết kế với kiến trúc linh hoạt, cho phép mở rộng quy mô về số lượng người dùng và dữ liệu khi doanh nghiệp phát triển mà không cần thay đổi toàn bộ cấu trúc hệ thống.
       - Cơ sở dữ liệu linh hoạt: Hệ thống sử dụng cơ sở dữ liệu MongoDB, hỗ trợ dễ dàng mở rộng khi lượng dữ liệu tăng lên, đáp ứng nhu cầu lưu trữ và truy vấn dữ liệu lớn khi số lượng nhân viên trong doanh nghiệp gia tăng.
4. Tính khả dụng (Availability)
   * + - Khả năng hoạt động liên tục: Hệ thống phải đảm bảo hoạt động ổn định và liên tục, thời gian ngừng hoạt động (downtime) không quá 1% trong tổng thời gian hoạt động. Điều này rất quan trọng đối với các doanh nghiệp có nhu cầu truy cập thông tin nhân sự thường xuyên.
       - Phục hồi sau sự cố (Disaster Recovery): Hệ thống cần có cơ chế sao lưu dữ liệu định kỳ và phục hồi sau sự cố nhanh chóng, nhằm giảm thiểu mất mát dữ liệu khi xảy ra lỗi hoặc sự cố ngoài ý muốn.
5. Khả năng tương thích và dễ sử dụng (Usability & Compatibility)
   * + - Giao diện thân thiện: Hệ thống cần cung cấp giao diện người dùng dễ sử dụng, giúp người dùng ở các vai trò khác nhau (Quản trị viên, quản lý nhân sự, nhân viên) có thể thực hiện thao tác một cách nhanh chóng và dễ hiểu.
       - Tương thích đa nền tảng: Hệ thống phải hoạt động tốt trên nhiều trình duyệt web phổ biến (Google Chrome, Mozilla Firefox, Microsoft Edge) và các thiết bị có độ phân giải màn hình khác nhau, đảm bảo tính linh hoạt và trải nghiệm người dùng tốt.
6. Khả năng bảo trì và nâng cấp (Maintainability & Extensibility)
   * + - Dễ bảo trì: Hệ thống phải được xây dựng với cấu trúc mã nguồn rõ ràng, dễ hiểu và dễ bảo trì. Các tính năng, thành phần trong hệ thống cần được thiết kế module hóa, để có thể dễ dàng nâng cấp và sửa lỗi mà không ảnh hưởng đến các phần khác.
       - Dễ dàng nâng cấp: Khi có yêu cầu mở rộng hoặc thay đổi tính năng, hệ thống phải cho phép nâng cấp nhanh chóng mà không ảnh hưởng đến các chức năng hiện tại.

## Công nghệ sử dụng

### 1.2.1 React

Trong dự án quản lý nhân viên, em đã lựa chọn React là thư viện JavaScript chính để phát triển giao diện người dùng (UI). React được phát triển bởi Facebook và là một trong những thư viện phổ biến nhất hiện nay cho việc xây dựng các ứng dụng web tương tác và hiệu quả.

React giúp em tạo ra giao diện người dùng một cách nhanh chóng nhờ khả năng tái sử dụng các component. Các component trong React cho phép chia nhỏ giao diện thành các phần nhỏ hơn, mỗi phần có thể được phát triển và quản lý độc lập. Điều này không chỉ giúp mã nguồn của dự án trở nên dễ hiểu và bảo trì hơn, mà còn giúp việc phát triển tính năng mới được thực hiện nhanh chóng và hiệu quả hơn.

Ngoài ra, React sử dụng Virtual DOM (DOM ảo) để tối ưu hóa hiệu suất. Thay vì thực hiện thay đổi trực tiếp trên DOM thật của trình duyệt, React sẽ cập nhật trước trên Virtual DOM và chỉ thực hiện các thay đổi cần thiết trên DOM thật. Điều này giúp cải thiện tốc độ và hiệu suất của ứng dụng, đặc biệt khi có nhiều tương tác hoặc dữ liệu thay đổi liên tục.

* Tính năng chính:
  + - * + Component-based: React dựa trên các thành phần (components), giúp tách biệt giao diện thành những phần nhỏ dễ quản lý và tái sử dụng.
        + Virtual DOM: React sử dụng Virtual DOM để tối ưu hóa quá trình cập nhật giao diện, giúp tăng tốc độ xử lý và cải thiện trải nghiệm người dùng.
        + One-way Data Binding: Dữ liệu chỉ di chuyển một chiều, từ component gốc đến component con, giúp dễ dàng kiểm soát dòng dữ liệu trong ứng dụng.
* Ưu điểm:
  + Dễ bảo trì, tái sử dụng thành phần.
  + Thư viện lớn và cộng đồng đông đảo, có nhiều thư viện và tiện ích hỗ trợ.
  + Hiệu suất cao nhờ cơ chế Virtual DOM.

Ngoài việc sử dụng React để tạo giao diện thân thiện và dễ sử dụng, em còn kết hợp React với các thư viện khác như React Router để xử lý điều hướng giữa các trang, và Axios để tương tác với API của hệ thống phía server.

Với những ưu điểm về hiệu suất, khả năng mở rộng và quản lý component hiệu quả, React đã giúp em xây dựng một hệ thống quản lý nhân viên có tính tương tác cao, dễ sử dụng và có khả năng phát triển trong tương lai.

**1.2.2 Node.js**

Dự án sử dụng Node.js làm môi trường chạy phía server để xây dựng và phát triển API cho hệ thống. Node.js là một nền tảng chạy JavaScript trên môi trường máy chủ, giúp cho việc xây dựng các ứng dụng server-side trở nên linh hoạt, hiệu quả và đồng bộ hơn.

Một trong những lý do quan trọng khiến nhóm lựa chọn **Node.js** cho dự án là khả năng xử lý **non-blocking I/O** (xử lý bất đồng bộ). Cơ chế này cho phép hệ thống có thể tiếp nhận và quản lý đồng thời nhiều yêu cầu khác nhau mà không bị “tắc nghẽn” khi thực hiện các tác vụ tốn thời gian như truy vấn cơ sở dữ liệu hoặc đọc/ghi tệp. Trong bối cảnh hệ thống quản lý nhân sự cần đáp ứng nhiều chức năng như đăng nhập, cập nhật hồ sơ, hay truy vấn dữ liệu nhân viên, tính bất đồng bộ của Node.js giúp tối ưu hóa hiệu suất và rút ngắn thời gian phản hồi của hệ thống.

Bên cạnh đó, Node.js được phát triển dựa trên **V8 Engine của Google**, mang lại khả năng xử lý nhanh chóng các tác vụ JavaScript và tận dụng tối đa tài nguyên phần cứng. Đây là yếu tố quan trọng khi hệ thống cần xử lý khối lượng lớn dữ liệu liên quan đến nhân viên, dự án, tiền lương hay nghỉ phép.

Trong dự án này, Node.js được kết hợp với **Express.js**, một framework phổ biến và mạnh mẽ, giúp đơn giản hóa việc xây dựng các **API RESTful**. Các API này đóng vai trò trung gian, chịu trách nhiệm trao đổi dữ liệu giữa **client (React)** và **server (Node.js/Express)**, từ đó đảm bảo tính liền mạch và hiệu quả trong quá trình vận hành hệ thống

* Tính năng chính:
  + - * + Asynchronous I/O: Node.js sử dụng mô hình xử lý bất đồng bộ, giúp cải thiện hiệu suất cho các ứng dụng I/O-intensive (ứng dụng có tác vụ đọc ghi nhiều).
        + Single-threaded Event Loop: Node.js sử dụng mô hình luồng đơn (single-thread) nhưng vẫn có khả năng xử lý nhiều yêu cầu cùng lúc nhờ event loop.
* Ưu điểm:
  + - * + Hiệu suất cao cho các ứng dụng thời gian thực, như chat, game trực tuyến.
        + Cộng đồng lớn, có nhiều module và thư viện (npm - Node Package Manager).
        + Chạy JavaScript trên cả client và server, giúp dễ dàng phát triển ứng dụng full-stack.

Một ưu điểm khác của Node.js là khả năng dễ dàng mở rộng hệ thống. Nhờ cộng đồng lớn và hệ sinh thái phong phú, dự án đã tận dụng các module và thư viện hỗ trợ trong Node.js như Mongoose để tương tác với cơ sở dữ liệu MongoDB và jsonwebtoken để xử lý xác thực bằng token (JWT).

Với sự lựa chọn Node.js, em đã có thể phát triển một hệ thống có hiệu suất cao, ổn định và linh hoạt, đáp ứng tốt các yêu cầu về tính tương tác và khả năng mở rộng trong tương lai.

**1.2.3 Express.js**

Express.js là một web framework nhẹ và mạnh mẽ cho Node.js, giúp đơn giản hóa quá trình phát triển các ứng dụng web và API.

Một trong những ưu điểm nổi bật của Express.js là sự đơn giản và linh hoạt trong việc thiết lập và quản lý các route (đường dẫn). Express.js cho phép em dễ dàng định nghĩa các endpoint để nhận và xử lý các yêu cầu từ client (React), đồng thời gửi lại phản hồi với dữ liệu tương ứng. Cụ thể, trong dự án này, Express.js chịu trách nhiệm xây dựng các API RESTful, giúp tương tác giữa client và server diễn ra mượt mà.

* Tính năng chính:
  + - * + Middleware: Express hỗ trợ khái niệm middleware, giúp dễ dàng thêm các chức năng vào luồng xử lý yêu cầu, như xử lý xác thực, phân tích dữ liệu từ request, logging.
        + Routing: Express cung cấp hệ thống định tuyến đơn giản, cho phép định nghĩa các tuyến đường xử lý các yêu cầu HTTP như GET, POST, PUT, DELETE.
* Ưu điểm:
  + - * + Rất nhẹ và linh hoạt, không ràng buộc với cấu trúc cố định.
        + Dễ học và sử dụng với các tính năng mạnh mẽ.
        + Kết hợp hoàn hảo với Nodejs cho các ứng dụng backend hiệu quả.

Ngoài ra, Expressjs cho phép dễ dàng tích hợp với các middleware – các hàm trung gian giúp xử lý các tác vụ như xác thực người dùng, log các yêu cầu HTTP, xử lý lỗi và bảo mật. Em đã sử dụng một số middleware như:

* + - * body-parser: để phân tích các yêu cầu POST từ client và chuyển đổi dữ liệu thành định dạng JSON.
      * jsonwebtoken: để xác thực và bảo mật bằng cách sử dụng JSON Web Token (JWT).
      * cors: để xử lý vấn đề CORS (Cross-Origin Resource Sharing), cho phép client và server giao tiếp an toàn qua các domain khác nhau.

Express.js cũng giúp hệ thống trở nên dễ dàng mở rộng và bảo trì. Với cấu trúc rõ ràng và module hóa, Em có thể dễ dàng thêm các tính năng mới hoặc chỉnh sửa các chức năng hiện tại mà không ảnh hưởng đến các phần khác của hệ thống.

Nhờ có Express.js, việc xây dựng các API cho hệ thống quản lý nhân viên trở nên nhanh chóng, dễ dàng, đồng thời đảm bảo hiệu suất và tính bảo mật cao.

**1.2.4 MongoDB**

Trong dự án quản lý nhân viên, em đã sử dụng MongoDB làm hệ quản trị cơ sở dữ liệu chính. MongoDB là một cơ sở dữ liệu NoSQL, sử dụng mô hình lưu trữ dữ liệu dưới dạng document (tài liệu) trong các collection (bộ sưu tập). Điều này cho phép lưu trữ dữ liệu một cách linh hoạt và dễ mở rộng.

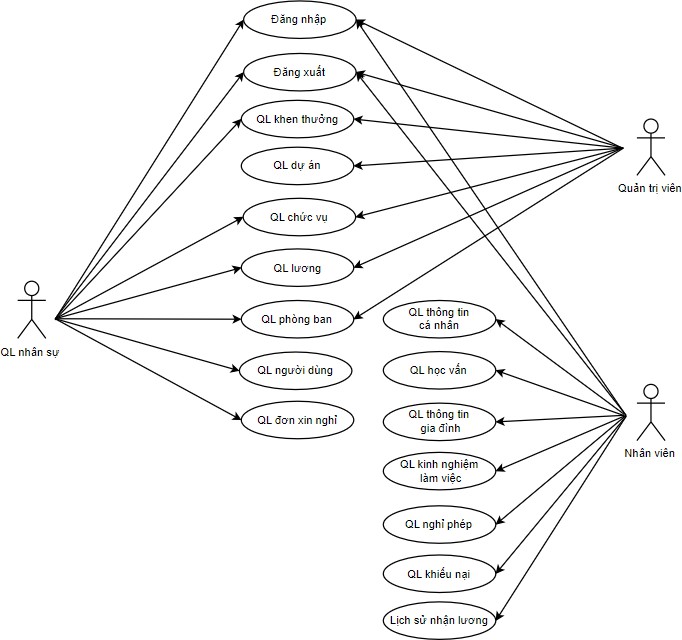
Một trong những lý do chính mà em chọn MongoDB cho dự án này là khả năng quản lý các dữ liệu có cấu trúc phức tạp như thông tin nhân viên, dự án, phòng ban, chức vụ, và lương. Mô hình document của MongoDB cho phép lưu trữ dữ liệu theo dạng JSON (hoặc BSON), giúp việc quản lý và truy vấn dữ liệu trở nên dễ dàng hơn so với các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.

* Tính năng chính:
  + - * + Cấu trúc linh hoạt: MongoDB không yêu cầu schema cứng nhắc, cho phép em dễ dàng thêm các thuộc tính mới hoặc điều chỉnh cấu trúc dữ liệu mà không cần phải thay đổi toàn bộ hệ thống. Điều này rất hữu ích khi quản lý các thông tin thay đổi thường xuyên như chức vụ, khen thưởng, kỷ luật và thông tin cá nhân của nhân viên.
        + Hiệu suất cao với khối lượng dữ liệu lớn: MongoDB hỗ trợ khả năng mở rộng theo chiều ngang, giúp lưu trữ và xử lý khối lượng lớn dữ liệu một cách hiệu quả. Điều này rất phù hợp với các ứng dụng có nhiều dữ liệu liên quan đến nhân sự và dự án.
        + Tích hợp dễ dàng với Node.js và Express.js: Với sự hỗ trợ mạnh mẽ từ thư viện Mongoose, MongoDB có thể dễ dàng tương tác với Node.js và Express.js. Mongoose giúp tạo mô hình dữ liệu (model) và thực hiện các thao tác như truy vấn, thêm, sửa, xóa dữ liệu một cách đơn giản và nhanh chóng. Nhờ đó, em có thể triển khai các chức năng như quản lý nhân viên, quản lý dự án và nghỉ phép một cách hiệu quả.
        + Khả năng mở rộng: MongoDB cho phép mở rộng hệ thống theo nhu cầu mà không ảnh hưởng đến các thành phần khác. Điều này đảm bảo hệ thống quản lý nhân viên có thể phát triển để đáp ứng nhu cầu ngày càng tăng của công ty.
* Ưu điểm:
  + - * + Phù hợp cho ứng dụng với dữ liệu lớn và yêu cầu truy vấn phức tạp.
        + Dễ mở rộng và tích hợp với các công nghệ hiện đại.
        + Linh hoạt trong việc lưu trữ dữ liệu có cấu trúc khác nhau trong cùng một collection

Với những ưu điểm về hiệu suất, khả năng mở rộng và tính linh hoạt trong quản lý dữ liệu, MongoDB đã giúp em xây dựng hệ thống quản lý nhân viên có tính tương tác cao, dễ quản lý và dễ bảo trì trong tương lai.

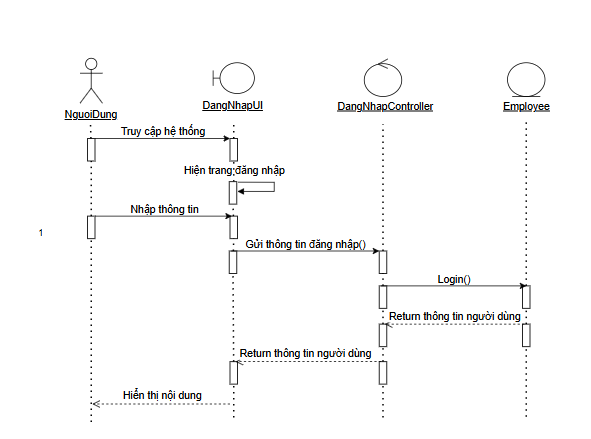
# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích use case

****Hình 2.1 Use case tổng quan

## 2.2 Đặc tả usecase

### 2.2.1 Use case đăng nhập



Hình 2.2 Biểu đồ trình tự Use case đăng nhập

**Mô tả vắn tắt:** Cho phép người dùng (quản trị viên, quản lý nhân sự, nhân viên) đăng nhập vào hệ thống để sử dụng các chức năng tương ứng với quyền của họ.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Người dùng truy cập trang đăng nhập của hệ thống.
2. Hệ thống yêu cầu nhập email và mật khẩu.
3. Người dùng nhập thông tin và nhấn nút “Đăng nhập”.
4. Hệ thống kiểm tra thông tin trong bảng employee.
5. Nếu hợp lệ, hệ thống xác định quyền của người dùng và chuyển hướng đến trang chủ hoặc dashboard phù hợp.
6. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Nhập thiếu hoặc sai thông tin → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

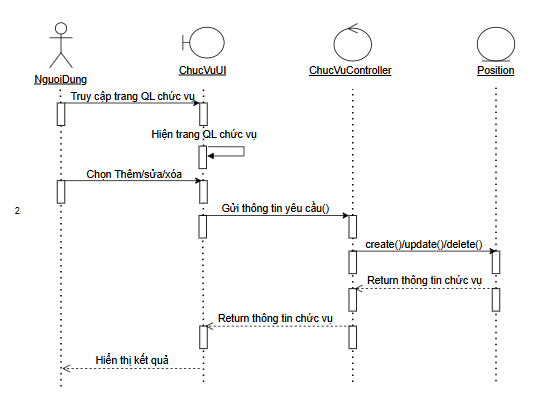
**Tiền điều kiện:** Người dùng đã được cấp tài khoản trong hệ thống.

**Hậu điều kiện:** Người dùng đăng nhập thành công và được điều hướng đến giao diện phù hợp với quyền hạn.

**Yêu cầu đặc biệt:** Đăng nhập phải đúng thông tin (tên đăng nhập, mật khẩu).

**Điểm mở rộng:** Không có.

### 2.2.2 Use case quản lý chức vụ

****

Hình 2.3 Biểu đồ Use case quản lý chức vụ

**Mô tả vắn tắt:** Cho phép quản trị viên và quản lý nhân sự xem, thêm, sửa, xóa thông tin về các chức vụ trong công ty.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Người dùng truy cập vào hệ thống và chọn chức năng “Quản lý chức vụ”.
2. Hệ thống hiển thị danh sách các chức vụ hiện có.
3. Người dùng thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa chức vụ.
4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
5. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Người dùng nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

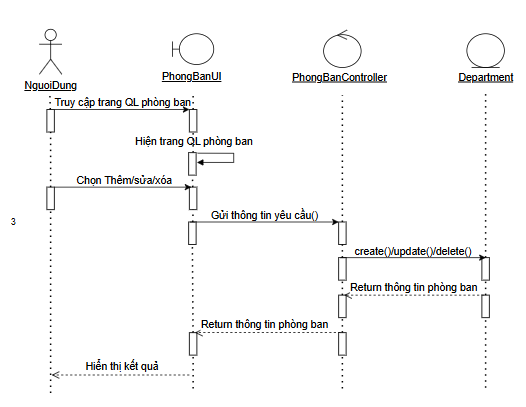
**Tiền điều kiện:** Người dùng đã đăng nhập và có quyền quản trị hoặc quản lý nhân sự.

**Hậu điều kiện:** Thông tin về chức vụ được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt:** Thông tin nhập vào phải hợp lệ (ví dụ: tên chức vụ không để trống, không trùng lặp).

**Điểm mở rộng:** Không có.

### 2.2.3 Use case quản lý phòng ban

****

Hình 2.4 Biểu đồ Use case quản lý phòng ban

**Mô tả vắn tắt:** Cho phép quản trị viên hoặc quản lý nhân sự xem, thêm, sửa, xóa thông tin về các phòng ban trong công ty.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Người dùng truy cập vào hệ thống và chọn chức năng “Quản lý phòng ban”.
2. Hệ thống hiển thị danh sách các phòng ban hiện có.
3. Người dùng thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa phòng ban.
4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
5. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Người dùng nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Người dùng đã đăng nhập và có quyền quản trị hoặc quản lý nhân sự.

**Hậu điều kiện:** Thông tin phòng ban được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt:** Thông tin nhập vào phải hợp lệ (ví dụ: tên phòng ban không để trống, không trùng lặp).

**Điểm mở rộng:** Không có.

### 2.2.4 Use case quản lý khen thưởng, kỷ luật

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2.5 Biểu đồ Use case quản lý khen thưởng, kỷ luật

**Mô tả vắn tắt:** Cho phép quản trị viên và quản lý nhân sự xem, thêm, sửa, xóa thông tin về các khen thưởng, kỷ luật trong công ty.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Người dùng đăng nhập vào hệ thống và chọn chức năng “Quản lý khen thưởng, kỷ luật”.
2. Hệ thống hiển thị danh sách khen thưởng, kỷ luật hiện có.
3. Người dùng thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin khen thưởng, kỷ luật.
4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
5. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Người dùng nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

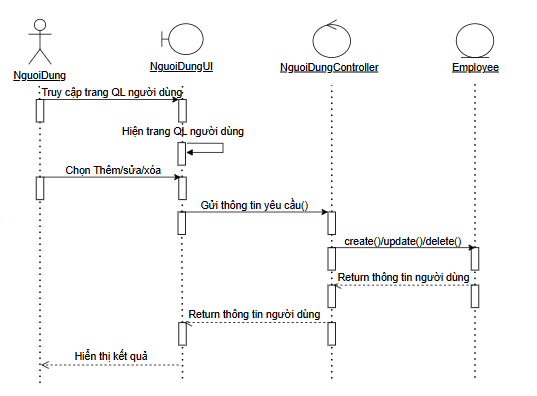
**Tiền điều kiện:** Người dùng đã đăng nhập và có quyền quản trị hoặc quản lý nhân sự.

**Hậu điều kiện:** Thông tin khen thưởng, kỷ luật được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt:** Thông tin nhập vào phải hợp lệ (ví dụ: ngày tháng, nội dung khen thưởng/kỷ luật không để trống).

**Điểm mở rộng:** Không có.

### 2.2.5 Use case quản lý người dùng

****

Hình 2.6 Biểu đồ Use case quản lý người dùng

**Mô tả vắn tắt:** Cho phép quản lý nhân sự xem, thêm, sửa và khóa tài khoản người dùng trong công ty. Chức năng khóa chỉ áp dụng đối với tài khoản nhân viên. Ngoài ra, có thể tìm kiếm người dùng thông qua mã nhân viên hoặc tên nhân viên

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Quản lý nhân sự đăng nhập vào hệ thống và chọn chức năng “Quản lý người dùng”.
2. Hệ thống hiển thị danh sách tài khoản người dùng hiện có.
3. Quản lý nhân sự thực hiện thao tác thêm mới, chỉnh sửa hoặc khóa tài khoản nhân viên.
4. Quản lý nhân sự có thể tìm kiếm người dùng theo mã nhân viên hoặc tên nhân viên.
5. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
6. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Người dùng nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Quản lý nhân sự cố gắng khóa tài khoản của đối tượng không phải nhân viên → hiển thị thông báo từ chối.
3. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Quản lý nhân sự đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin người dùng được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt:** Dữ liệu nhập vào phải hợp lệ (ví dụ: mã nhân viên duy nhất, tên không để trống).

**Điểm mở rộng:** Không có.

**2.2.6 Use case quản lý lương của quản lý**

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2.7 Biểu đồ Use case quản lý lương của quản lý

**Mô tả vắn tắt:** Cho phép quản lý nhân sự và quản trị viên quản lý thông tin về lương của nhân viên. Người dùng có thể xem, thêm, chỉnh sửa, xóa thông tin lương, đồng thời tìm kiếm thông tin theo mã nhân viên.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Người dùng đăng nhập vào hệ thống và chọn chức năng “Quản lý lương”.
2. Hệ thống hiển thị danh sách thông tin lương hiện có.
3. Người dùng thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin lương của nhân viên.
4. Người dùng có thể tìm kiếm thông tin lương theo mã nhân viên.
5. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
6. Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Người dùng nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Người dùng đã đăng nhập và có quyền quản lý nhân sự hoặc quản trị hệ thống.

**Hậu điều kiện:** Thông tin lương của nhân viên được cập nhật thành công trong hệ thống

**Yêu cầu đặc biệt**: Thông tin lương phải hợp lệ (ví dụ: không để trống, số tiền là giá trị số).  
**Điểm mở rộng**: Không có

### 2.2.7 Use case quản lý đơn xin nghỉ phép

**A diagram of a company

AI-generated content may be incorrect.**

Hình 2.8 Biểu đồ Use case quản lý đơn xin nghỉ phép

**Mô tả vắn tắt:** Cho phép quản lý nhân sự quản lý các đơn xin nghỉ phép của nhân viên, bao gồm duyệt, từ chối và tìm kiếm đơn theo mã nhân viên.

**Luồng sự kiện**

* **Luồng cơ bản:**

1. Quản lý nhân sự đăng nhập vào hệ thống và chọn chức năng “Quản lý đơn xin nghỉ”.
2. Hệ thống hiển thị danh sách các đơn xin nghỉ của nhân viên.
3. Quản lý nhân sự thực hiện thao tác duyệt hoặc từ chối đơn xin nghỉ.
4. Hệ thống ghi nhận kết quả và cập nhật trạng thái đơn trong cơ sở dữ liệu.
5. Use case kết thúc.
6. **Luồng rẽ nhánh:**
7. Quản lý nhân sự tìm kiếm đơn xin nghỉ theo mã nhân viên → hệ thống hiển thị kết quả phù hợp.
8. Nếu dữ liệu không tồn tại hoặc sai mã nhân viên → hiển thị thông báo lỗi.
9. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc

**Tiền điều kiện:** Quản lý nhân sự đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Đơn xin nghỉ của nhân viên được duyệt hoặc từ chối và trạng thái được cập nhật thành công.

**Yêu cầu đặc biệt**: Thông tin xử lý phải chính xác và ghi nhận đầy đủ lịch sử duyệt/từ chối.  
**Điểm mở rộng**: Không có

### 2.2.8 Use case quản lý dự án

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2.9 Biểu đồ Use case quản lý dự án

**Mô tả vắn tắt:** Cho phép quản trị viên quản lý thông tin về các dự án của công ty, bao gồm thêm, sửa, xóa và theo dõi danh sách dự án.

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Quản trị viên đăng nhập vào hệ thống và chọn chức năng “Quản lý dự án”.
  2. Hệ thống hiển thị danh sách các dự án hiện có.
  3. Quản trị viên thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin dự án.
  4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và làm mới danh sách.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Quản trị viên nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
  2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Quản trị viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin dự án được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt**: Thông tin dự án phải hợp lệ (ví dụ: tên dự án không để trống, ngày bắt đầu ≤ ngày kết thúc).  
**Điểm mở rộng**: Không có

### 2.2.9 Use case quản lý thông tin cá nhân

**A diagram of a computer program

AI-generated content may be incorrect.**

Hình 2.10 Biểu đồ Use case quản lý thông tin cá nhân

**Mô tả vắn tắt:** Cho phép nhân viên quản lý và chỉnh sửa thông tin cá nhân của mình, đồng thời có thể xem thông tin liên quan đến lương

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Quản lý thông tin cá nhân”.
  2. Hệ thống hiển thị thông tin cá nhân và thông tin lương của nhân viên.
  3. Nhân viên thực hiện chỉnh sửa, cập nhật thông tin cá nhân.
  4. Hệ thống kiểm tra dữ liệu và lưu lại các thay đổi vào cơ sở dữ liệu.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nhân viên nhập thiếu hoặc sai định dạng thông tin (ví dụ: email sai định dạng) → hiển thị thông báo lỗi.
  2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin cá nhân của nhân viên được cập nhật thành công trong hệ thống..

**Yêu cầu đặc biệt**: Dữ liệu cập nhật phải hợp lệ (ví dụ: số điện thoại, email đúng định dạng).

**Điểm mở rộng**: Không có

### 2.2.10 Use case quản lý học vấn

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2.11 Biểu đồ Use case quản lý học vấn

**Mô tả vắn tắt:** Cho phép nhân viên quản lý thông tin học vấn của mình, bao gồm thêm mới, chỉnh sửa và xóa

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Quản lý học vấn”.
  2. Hệ thống hiển thị danh sách thông tin học vấn hiện có.
  3. Nhân viên thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin học vấn.
  4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và hiển thị danh sách đã được cập nhật.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nhân viên nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
  2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin học vấn của nhân viên được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt**: Thông tin học vấn phải hợp lệ (ví dụ: tên trường, chuyên ngành, thời gian học không để trống).

**Điểm mở rộng**: Không có

### 2.2.11 Use case quản lý thông tin gia đình

**A diagram of a company

AI-generated content may be incorrect.**

Hình 2.12 Biểu đồ Use case quản lý thông tin gia đình

**Mô tả vắn tắt:** Cho phép nhân viên quản lý thông tin gia đình của mình, bao gồm thêm mới, chỉnh sửa và xóa.

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Quản lý thông tin gia đình”.
  2. Hệ thống hiển thị danh sách thông tin gia đình hiện có.
  3. Nhân viên thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin gia đình.
  4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và hiển thị danh sách đã được cập nhật.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**

1. Nhân viên nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin gia đình của nhân viên được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt**: Thông tin nhập vào phải hợp lệ (ví dụ: quan hệ gia đình, họ tên, ngày sinh không để trống).

**Điểm mở rộng**: Không có

### 2.2.12 Use case quản lý kinh nghiệm làm việc

**A diagram of a work flow

AI-generated content may be incorrect.**

Hình 2.13 Biểu đồ Use case quản lý kinh nghiệm làm việc

**Mô tả vắn tắt:** Cho phép nhân viên quản lý thông tin về kinh nghiệm làm việc của mình, bao gồm thêm mới, chỉnh sửa và xóa.

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Quản lý kinh nghiệm làm việc”.
  2. Hệ thống hiển thị danh sách kinh nghiệm làm việc hiện có.
  3. Nhân viên thực hiện thao tác thêm mới, chỉnh sửa hoặc xóa thông tin kinh nghiệm làm việc.
  4. Hệ thống cập nhật dữ liệu trong cơ sở dữ liệu và hiển thị danh sách đã được cập nhật.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nhân viên nhập thiếu hoặc sai thông tin khi thêm/sửa → hiển thị thông báo lỗi.
  2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin kinh nghiệm làm việc của nhân viên được cập nhật thành công trong hệ thống.

**Yêu cầu đặc biệt**: Thông tin phải hợp lệ (ví dụ: tên công ty, chức danh, thời gian làm việc không để trống).

**Điểm mở rộng**: Không có

### 2.2.13 Use case quản lý nghỉ phép

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2.14 Biểu đồ Use case quản lý nghỉ phép

**Mô tả vắn tắt:** Cho phép nhân viên tạo đơn xin nghỉ phép và gửi đến quản lý nhân sự để xét duyệt.

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Tạo đơn xin nghỉ”.
  2. Hệ thống hiển thị form nhập thông tin đơn xin nghỉ.
  3. Nhân viên điền đầy đủ thông tin vào đơn (thời gian nghỉ, lý do, số ngày nghỉ...).
  4. Nhân viên nhấn nút “Gửi đơn”.
  5. Hệ thống lưu đơn và chuyển trạng thái đơn sang “Chờ duyệt” để gửi đến quản lý nhân sự.
  6. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nhân viên nhập thiếu hoặc sai định dạng thông tin (ví dụ: ngày nghỉ không hợp lệ) → hiển thị thông báo lỗi.
  2. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Đơn xin nghỉ của nhân viên được tạo thành công và chờ xét duyệt từ quản lý nhân sự.

**Yêu cầu đặc biệt**: Thông tin nhập vào phải hợp lệ (ngày nghỉ phải lớn hơn hoặc bằng ngày hiện tại, số ngày nghỉ không vượt quá quy định)

**Điểm mở rộng**: Không có

### 2.2.14 Use case lịch sử nhận lương

**A diagram of a company

AI-generated content may be incorrect.**

Hình 2.15 Biểu đồ Use case lịch sử nhận lương

**Mô tả vắn tắt:** Cho phép nhân viên xem thông tin lương và thống kê lương của mình theo từng tháng/năm..

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Lịch sử nhận lương”.
  2. Hệ thống hiển thị danh sách các lần nhận lương của nhân viên.
  3. Nhân viên chọn khoảng thời gian (tháng/năm) muốn xem.
  4. Hệ thống lọc và hiển thị thông tin lương tương ứng với thời gian đã chọn.
  5. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nếu nhân viên không chọn thời gian → hệ thống mặc định hiển thị thông tin lương gần nhất.
  2. Nếu không có dữ liệu lương trong khoảng thời gian được chọn → hiển thị thông báo “Không có dữ liệu”.
  3. Kết nối cơ sở dữ liệu bị lỗi → hiển thị lỗi và kết thúc.

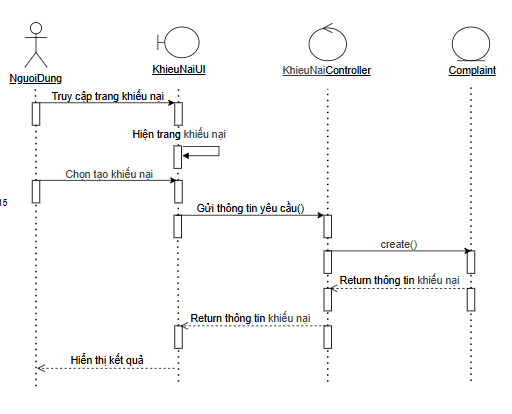
**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Thông tin lương được hiển thị đầy đủ và chính xác theo thời gian mà nhân viên lựa chọn.

**Yêu cầu đặc biệt**: Dữ liệu lương phải được cập nhật đúng và đồng bộ từ hệ thống tính lương.

**Điểm mở rộng**: Không có

### 2.2.15 Use case quản lý khiếu nại

****

Hình 2.16 Use case quản lý khiếu nại

**Mô tả vắn tắt:** Cho phép nhân viên gửi khiếu nại về các vấn đề trong công ty đến quản trị viên.

**Luồng sự kiện**

* **Luồng cơ bản:**
  1. Nhân viên đăng nhập vào hệ thống và chọn chức năng “Khiếu nại”.
  2. Hệ thống hiển thị danh sách các khiếu nại trước đây của nhân viên (nếu có).
  3. Nhân viên chọn “Thêm khiếu nại mới”.
  4. Hệ thống hiển thị form nhập thông tin khiếu nại.
  5. Nhân viên nhập nội dung khiếu nại và xác nhận gửi.
  6. Hệ thống lưu thông tin, cập nhật danh sách khiếu nại và gửi khiếu nại đến quản trị viên.
  7. Use case kết thúc.
* **Luồng rẽ nhánh:**
  1. Nếu nhân viên nhập thiếu thông tin bắt buộc → hiển thị thông báo lỗi và yêu cầu nhập lại.
  2. Nếu hệ thống không thể gửi đến quản trị viên (lỗi kết nối, lỗi dữ liệu) → hiển thị thông báo lỗi và kết thúc.

**Tiền điều kiện:** Nhân viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:** Khiếu nại của nhân viên được lưu thành công, hiển thị trong danh sách khiếu nại và gửi đến quản trị viên.

**Yêu cầu đặc biệt**: Không có

**Điểm mở rộng**: Không có

## A screenshot of a computer AI-generated content may be incorrect.2.3 Thiết kế cơ sở dữ liệu

Hình 2.17 Thiết kế CSDL trong mongoDB

Mô tả cơ sở dữ liệu

* + - * cities: Lưu trữ thông tin về các thành phố.
      * companies: Lưu trữ thông tin về công ty.
      * countries: Lưu trữ thông tin về các quốc gia.
      * departments: Lưu trữ thông tin về các phòng ban trong công ty.
      * educations: Lưu trữ thông tin về trình độ học vấn của nhân viên.
      * employees: Lưu trữ thông tin về nhân viên.
      * familyinfos: Lưu trữ thông tin về gia đình của nhân viên.
      * identitycounters: Có thể lưu trữ các số đếm tự động, giúp quản lý ID hoặc mã định danh trong hệ thống.
      * leaveapplications: Lưu trữ các đơn xin nghỉ phép của nhân viên.
      * projects: Lưu trữ thông tin về các dự án của công ty.
      * roles: Lưu trữ thông tin về các chức vụ của người dùng trong hệ thống.
      * salaries: Lưu trữ thông tin về lương của nhân viên.
      * states: Lưu trữ thông tin về các khu vực trong quốc gia.
      * workexperiences: Lưu trữ thông tin về kinh nghiệm làm việc của nhân viên.

# CHƯƠNG 3: THỰC NGHIỆM

## 3.1 Quy trình kiểm thử

1. Lập kế hoạch kiểm thử (Test Planning)

* Phạm vi kiểm thử: Các chức năng chính của hệ thống (Đăng nhập, Quản lý nhân viên, Quản lý lương, Quản lý nghỉ phép).
* Mục tiêu kiểm thử: Đảm bảo hệ thống hoạt động đúng theo yêu cầu, xử lý chính xác các trường hợp dữ liệu hợp lệ và hiển thị thông báo phù hợp khi dữ liệu không hợp lệ.
* Môi trường kiểm thử:
  + - Hệ điều hành: Windows 10
    - Trình duyệt: Google Chrome
    - Backend: Node.js + Express
    - CSDL: MongoDB Atlas
    - Công cụ hỗ trợ: Postman (kiểm thử API)

1. Thiết kế ca kiểm thử (Test Case Design)

* Dựa trên yêu cầu phần mềm, các ca kiểm thử hộp đen được xây dựng.
* Mỗi ca kiểm thử bao gồm: chức năng, đầu vào, đầu ra mong muốn, kết quả thực tế.

1. Thực thi kiểm thử (Test Execution)

* Các ca kiểm thử được thực hiện trên giao diện web và API.
* Ghi nhận kết quả Pass/Fail.

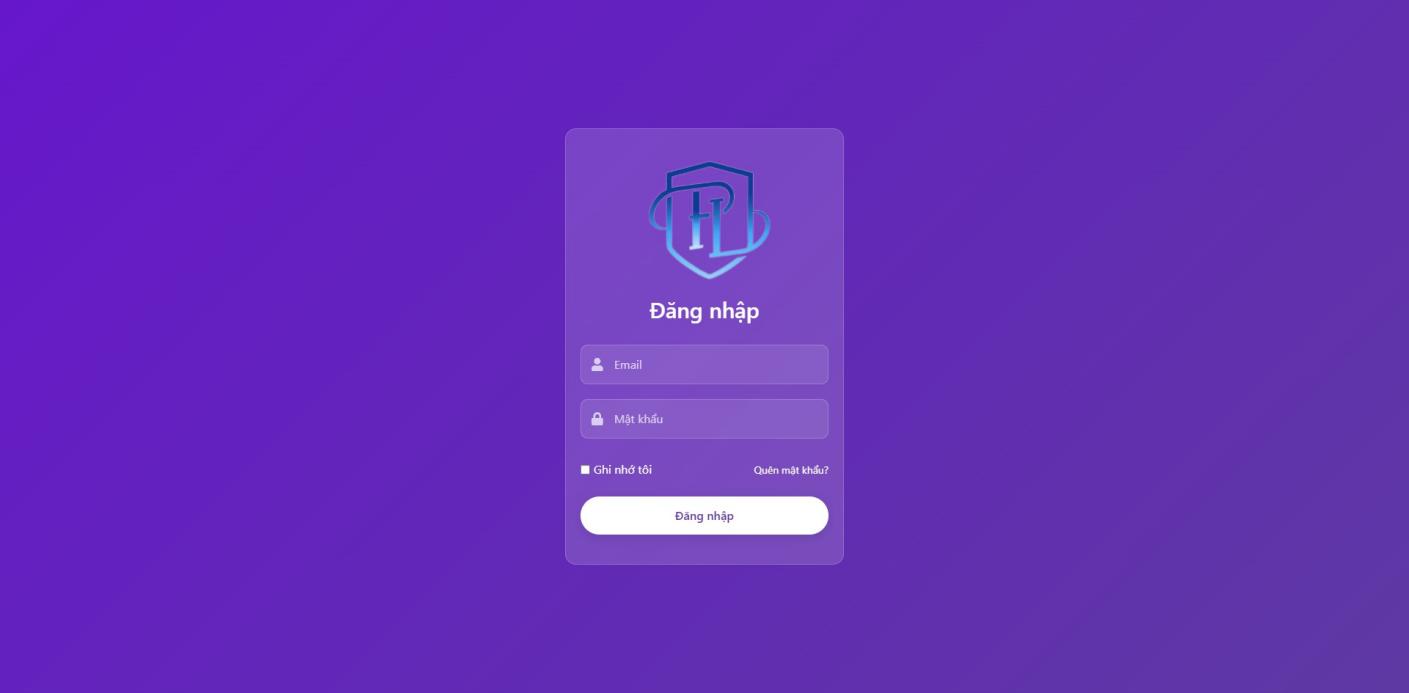
1. Báo cáo kiểm thử (Test Report)
   * Tổng hợp kết quả kiểm thử, đưa ra tỷ lệ Pass/Fail.
   * Đánh giá mức độ đáp ứng của hệ thống so với yêu cầu đặt ra.

## 3.2 Kết quả kiểm thử

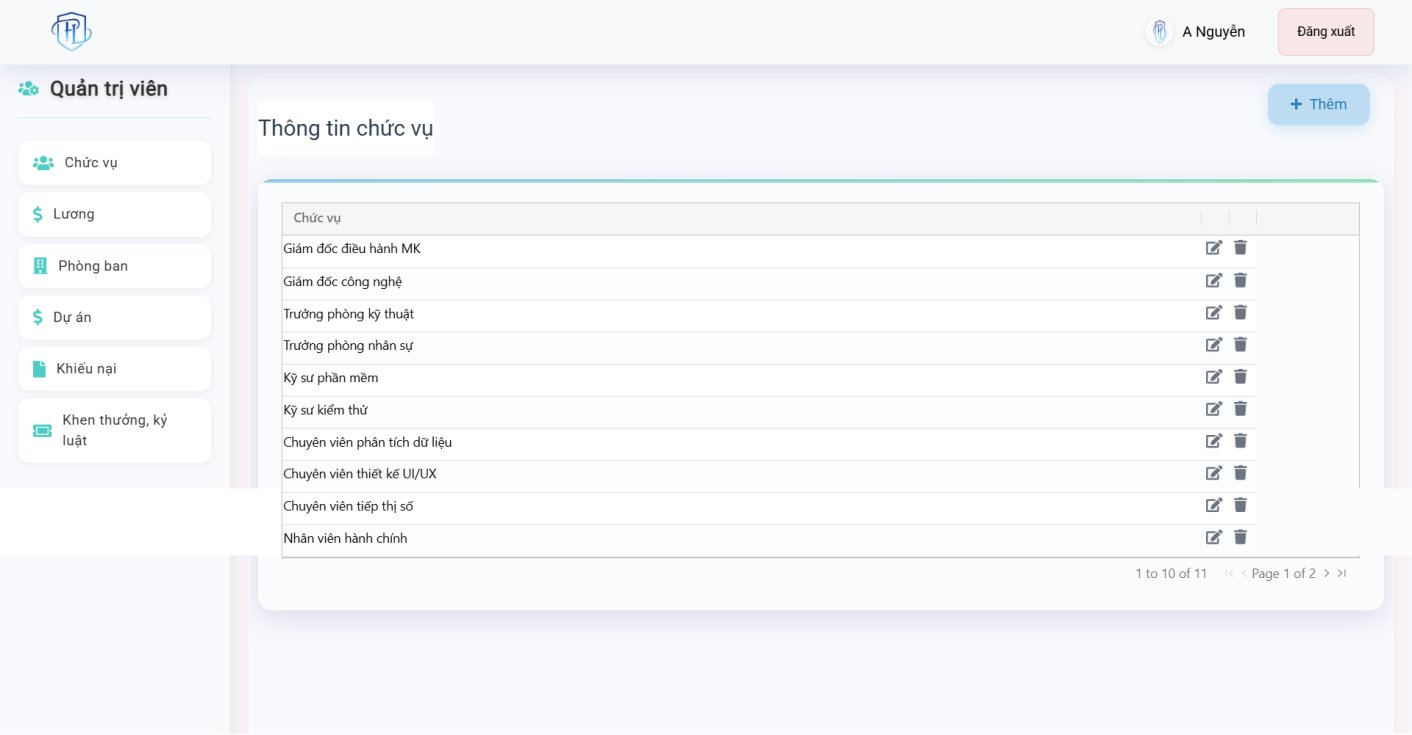
Bảng 3.1 Kiểm thử hộp đen

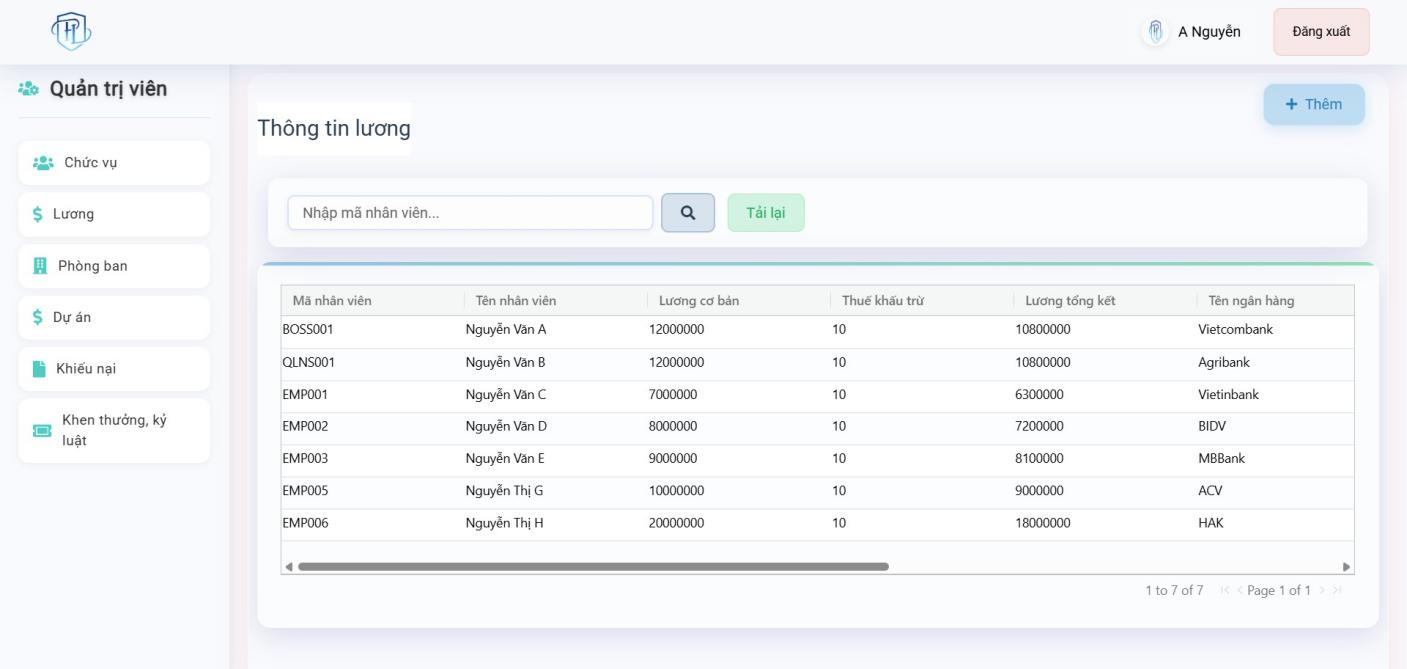
|  |  |  |  |
| --- | --- | --- | --- |
| Chức năng | Đầu vào | Đầu ra mong muốn | Kết quả |
| Đăng nhập (Admin) | Tên đăng nhập, mật khẩu đúng | Trang quản lý chính của Quản trị viên | Pass |
| Đăng nhập (Admin) | Tên đăng nhập, mật khẩu sai | Thông báo lỗi: Tên người dùng hoặc mật khẩu không hợp lệ | Pass |
| Thêm nhân viên | Thông tin nhân viên (Họ tên, địa chỉ, chức vụ,...) | Thông báo thành công, nhân viên được thêm vào hệ thống | Pass |
| Thêm nhân viên | Thiếu thông tin bắt buộc (Họ tên) | Thông báo lỗi: "Họ tên là bắt buộc" | Pass |
| Sửa thông tin nhân viên | ID nhân viên không tồn tại | Thông báo lỗi: "Nhân viên không tồn tại" | Pass |
| Khóa nhân viên | ID nhân viên | Thông báo thành công, nhân viên đã bị xóa | Pass |
| Khóa nhân viên | ID nhân viên không tồn tại | Thông báo lỗi: "Nhân viên không tồn tại" | Pass |
| Quản lý lương | ID nhân viên, lương mới | Thông báo thành công, lương của nhân viên được cập nhật | Pass |
| Quản lý lương | ID nhân viên không tồn tại | Thông báo lỗi: "Nhân viên không tồn tại" | Pass |

## 3.3 Kết quả

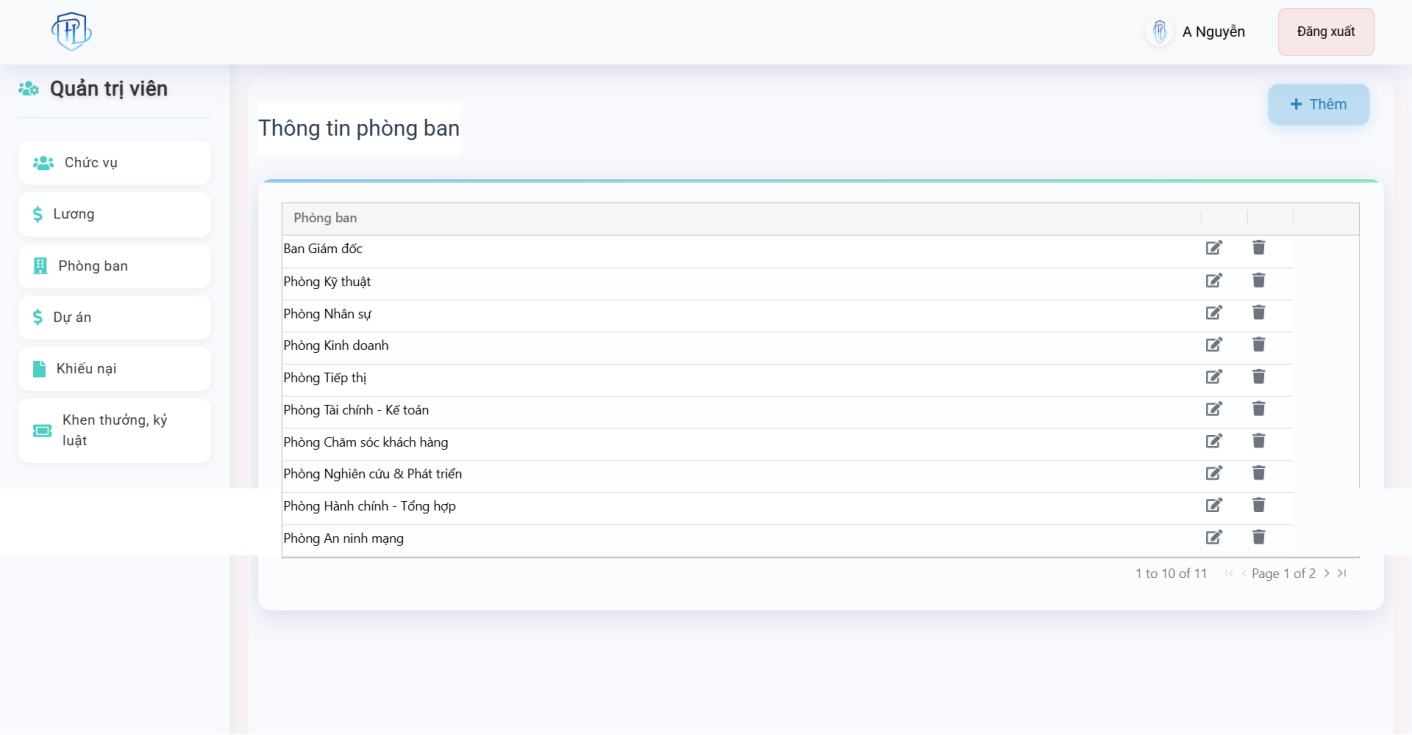
****

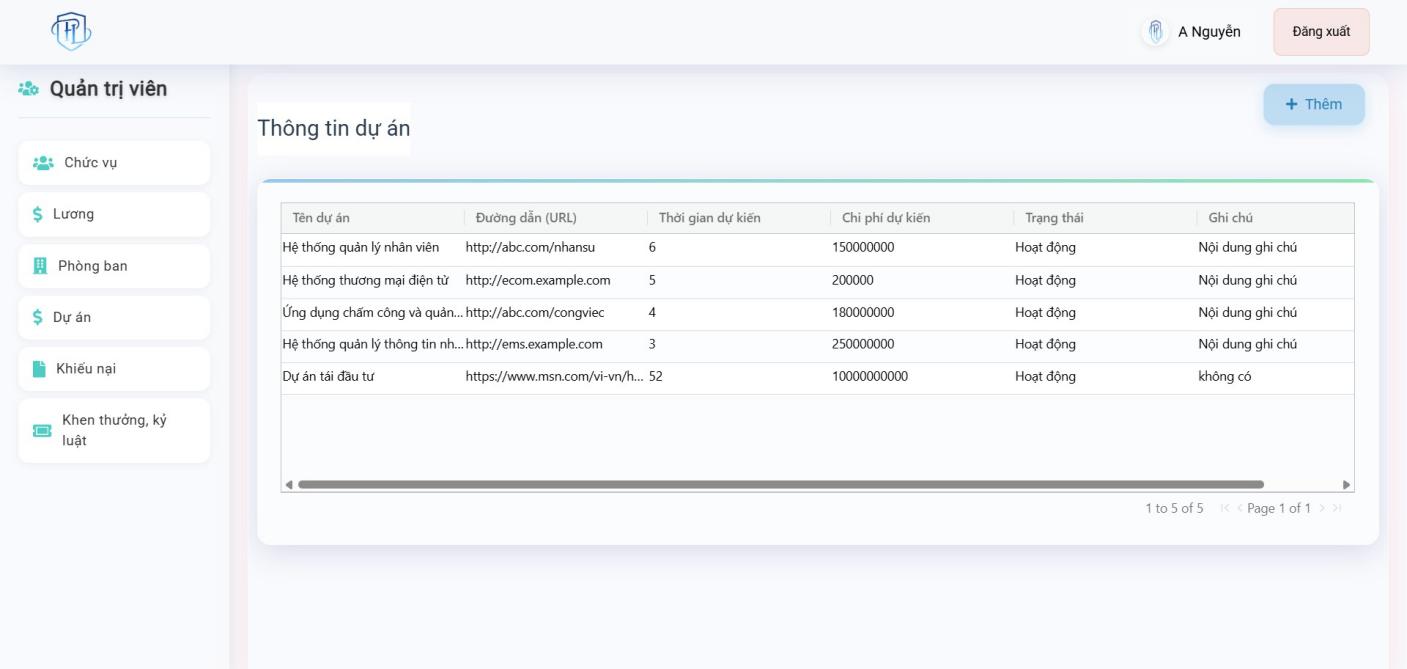
Hình 3.1: Giao diện đăng nhập



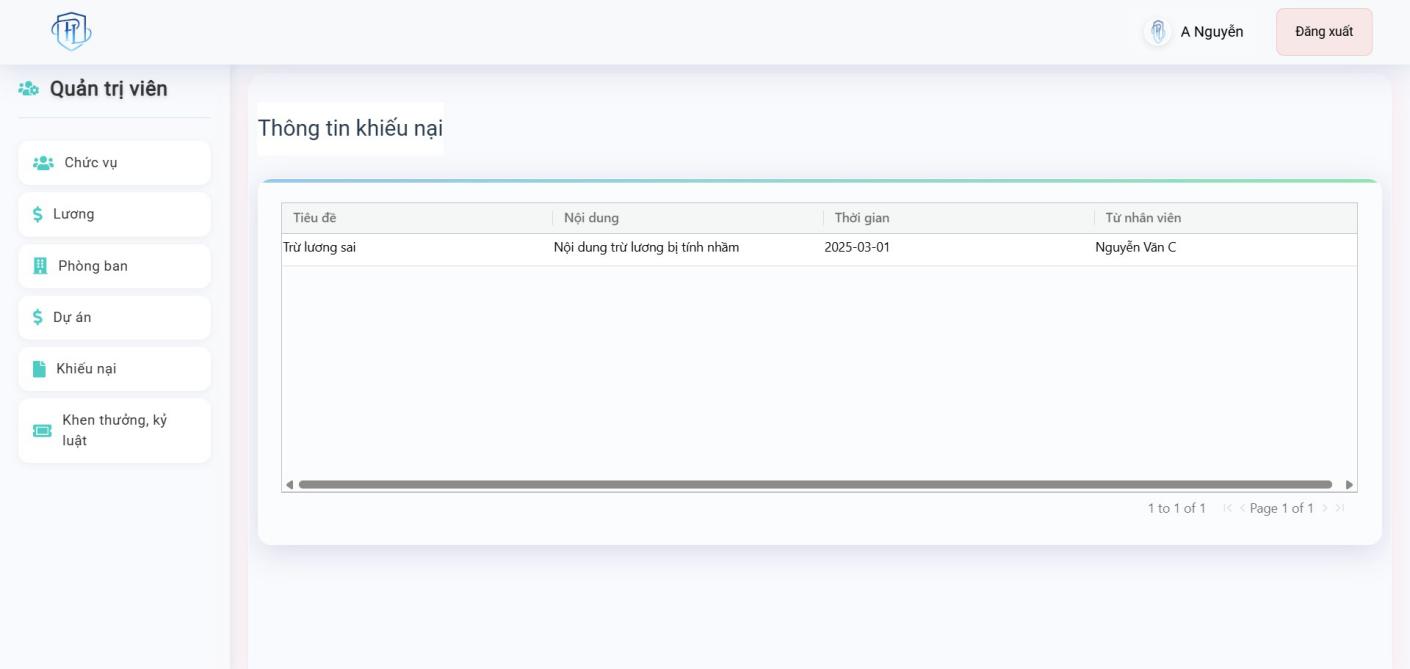
Hình 3.2: Giao diện quản lý chức vụ

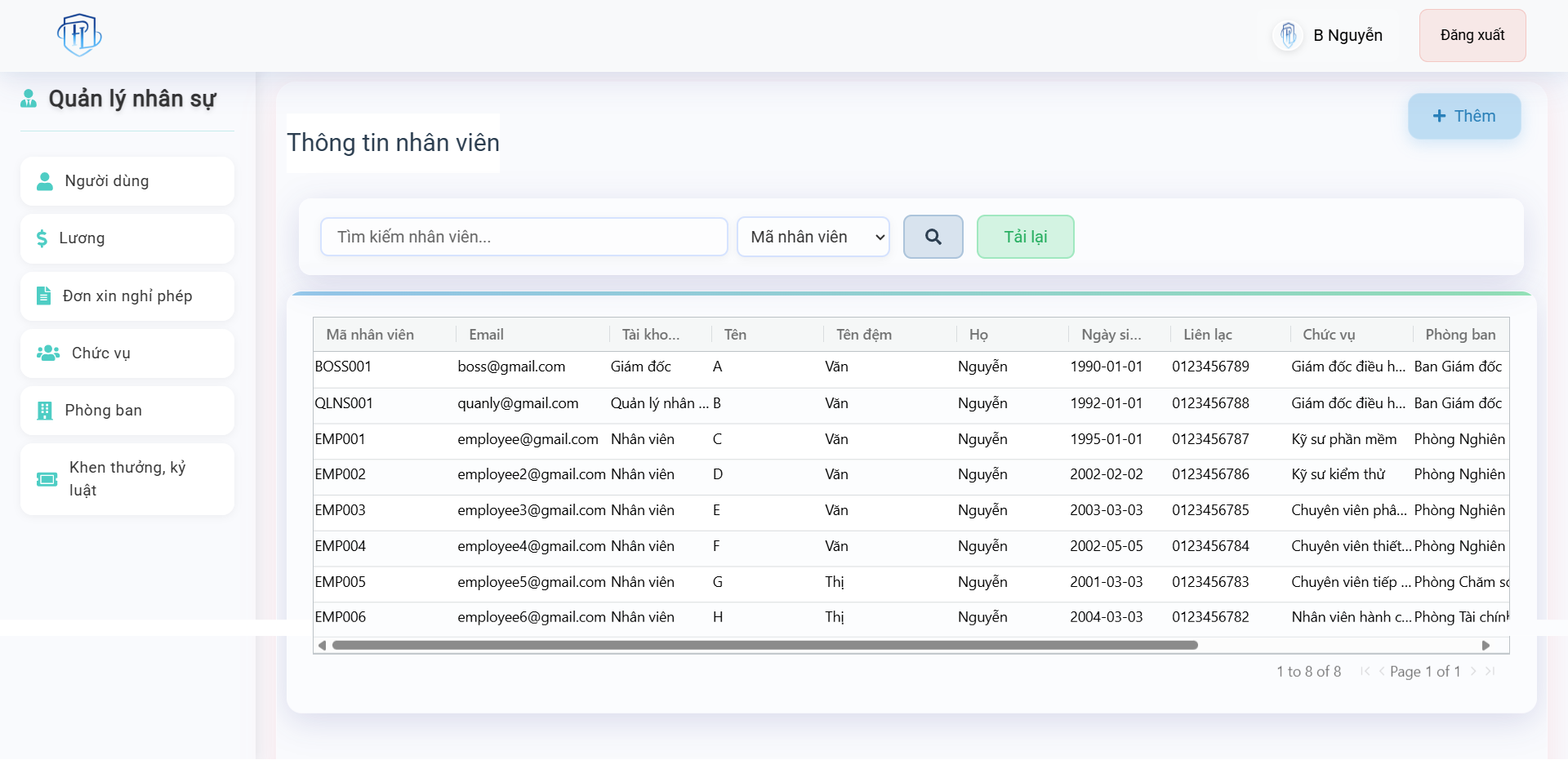
Hình 3.3: Giao diện quản lý lương



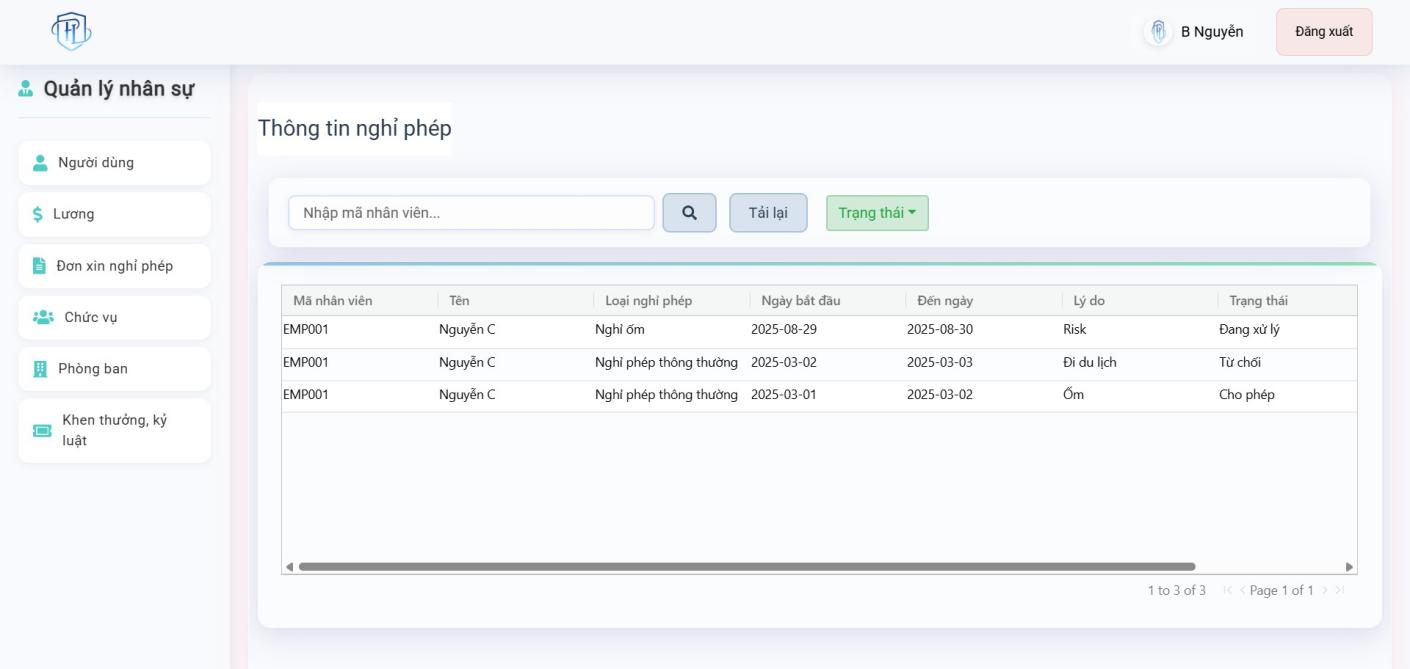
Hình 3.4: Giao diện quản lý phòng ban

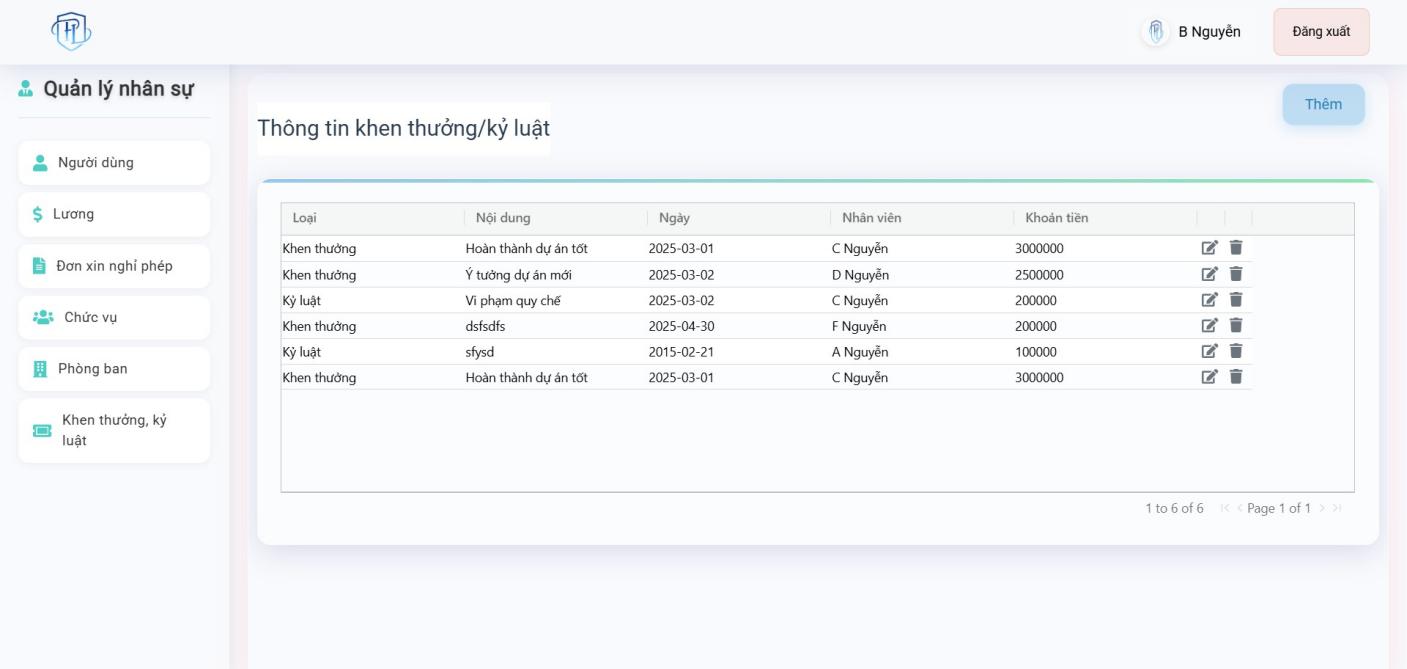
Hình 3.5: Giao diện quản lý dự án



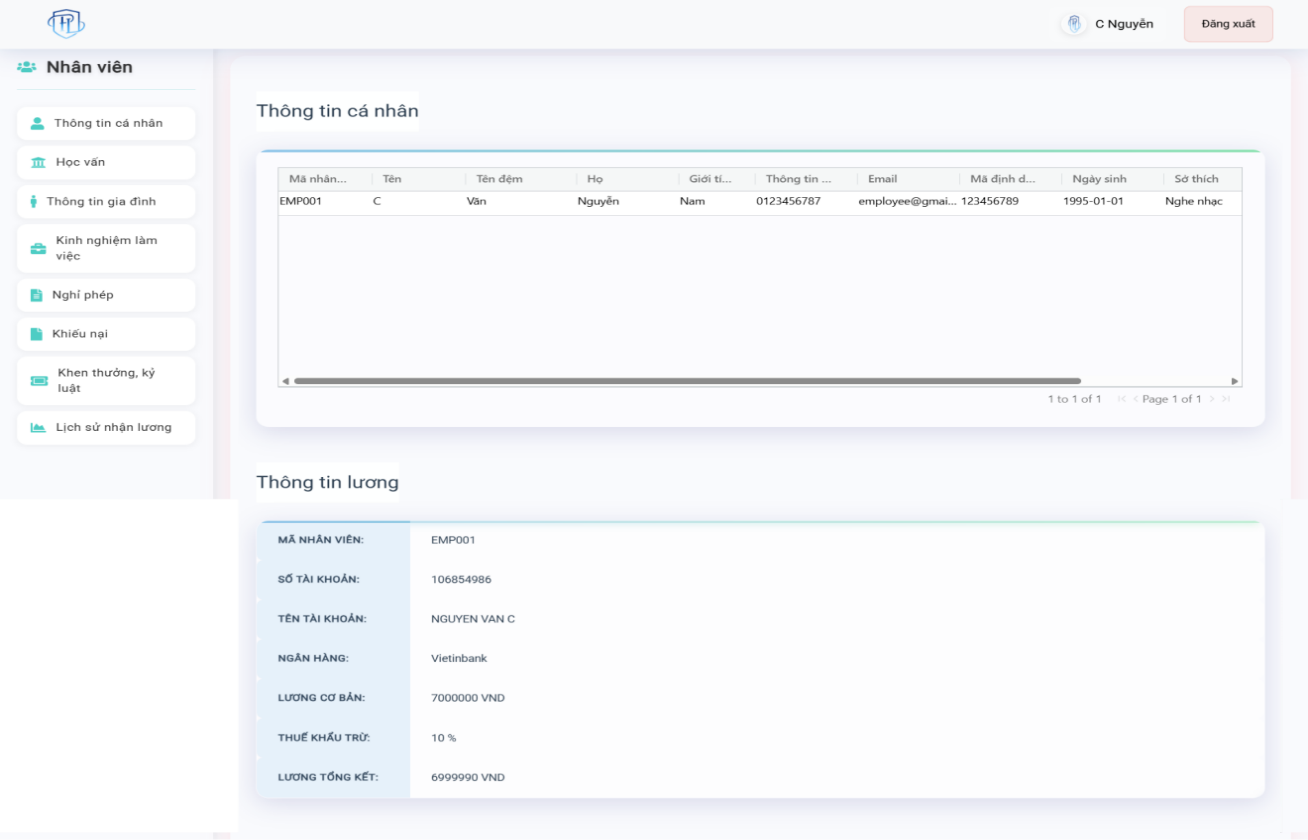
Hình 3.6: Giao diện quản lý khiếu nại

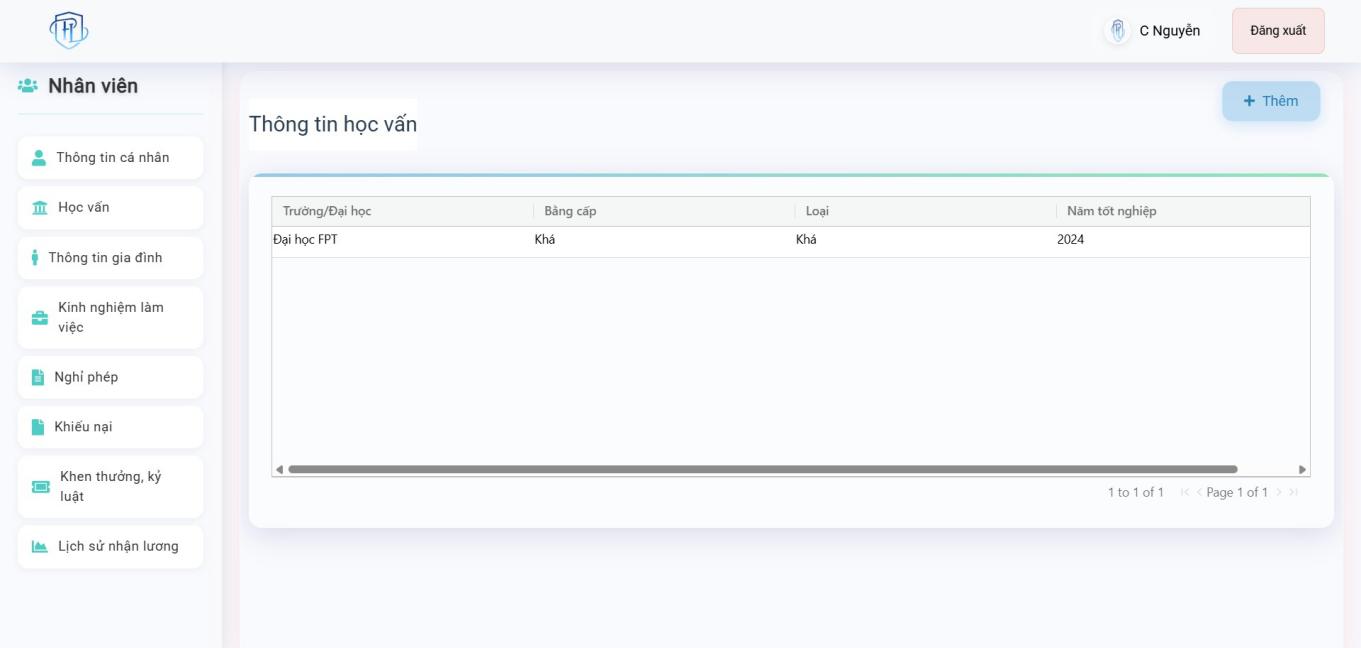
Hình 3.7: Giao diện quản lý người dùng



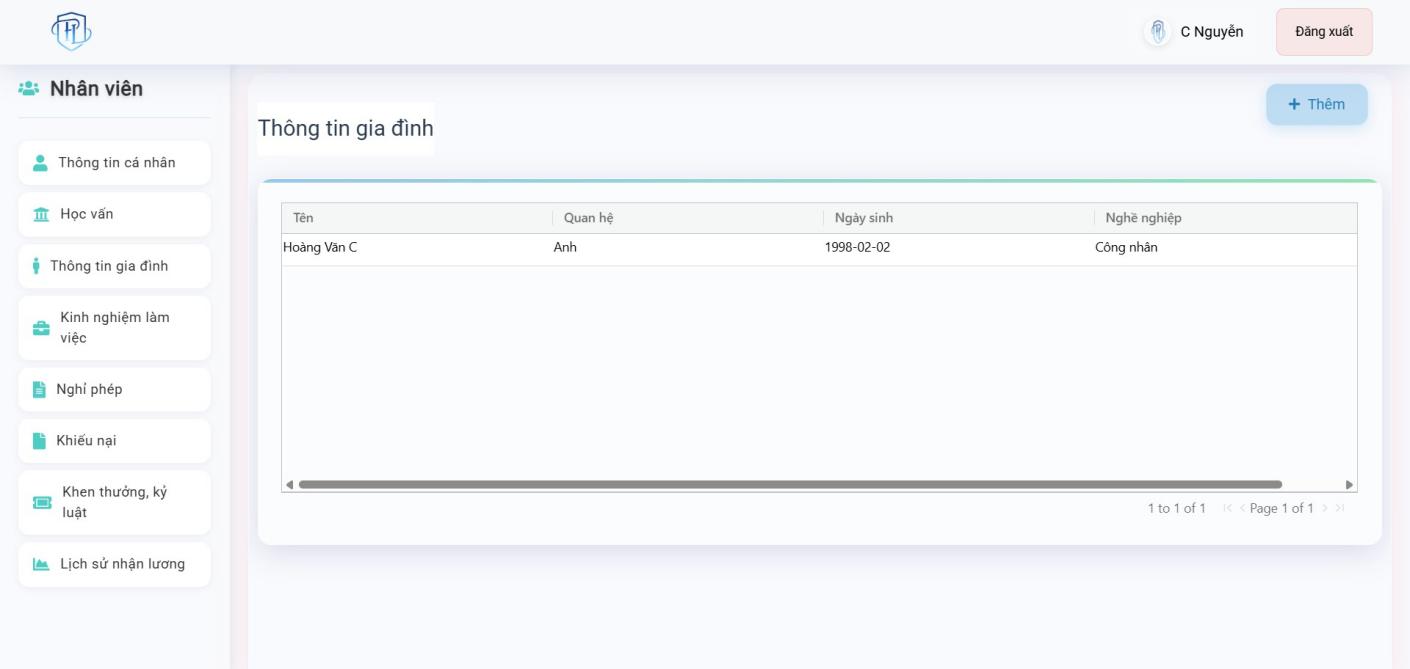
Hình 3.8: Giao diện quản lý đơn xin nghỉ phép

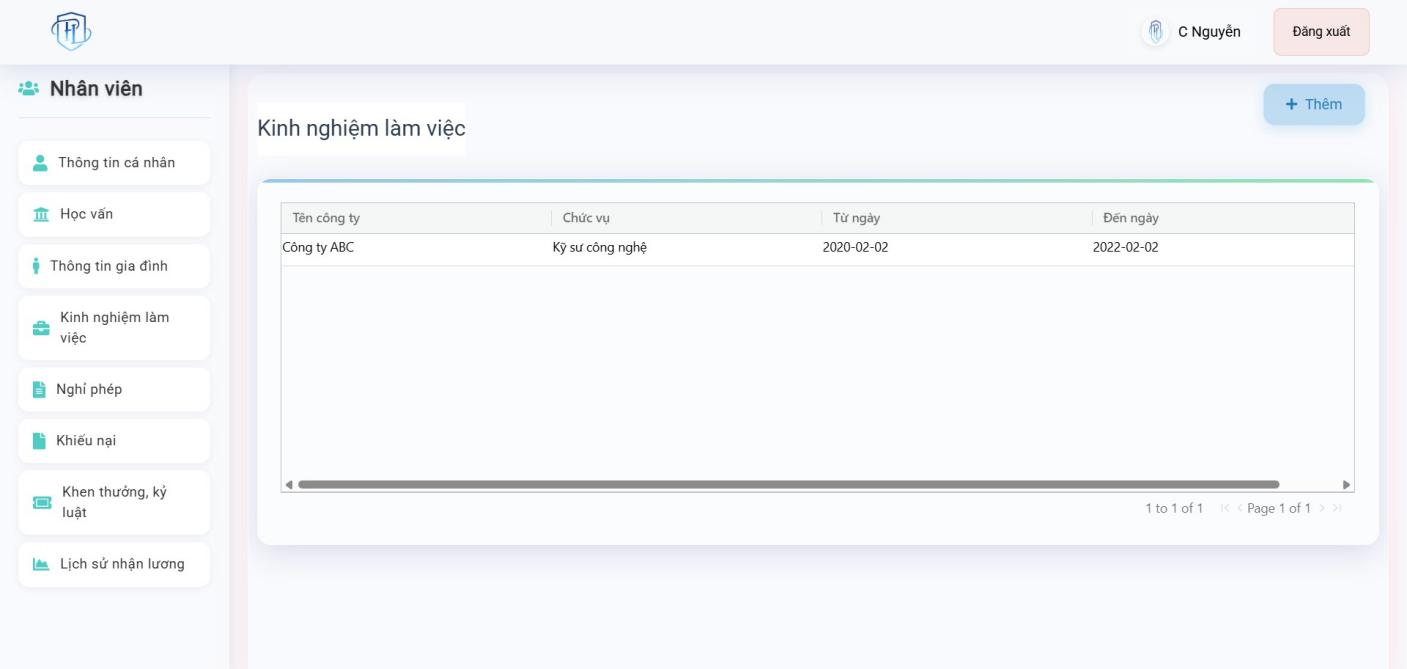
Hình 3.9: Giao diện quản lý khen thưởng, kỷ luật



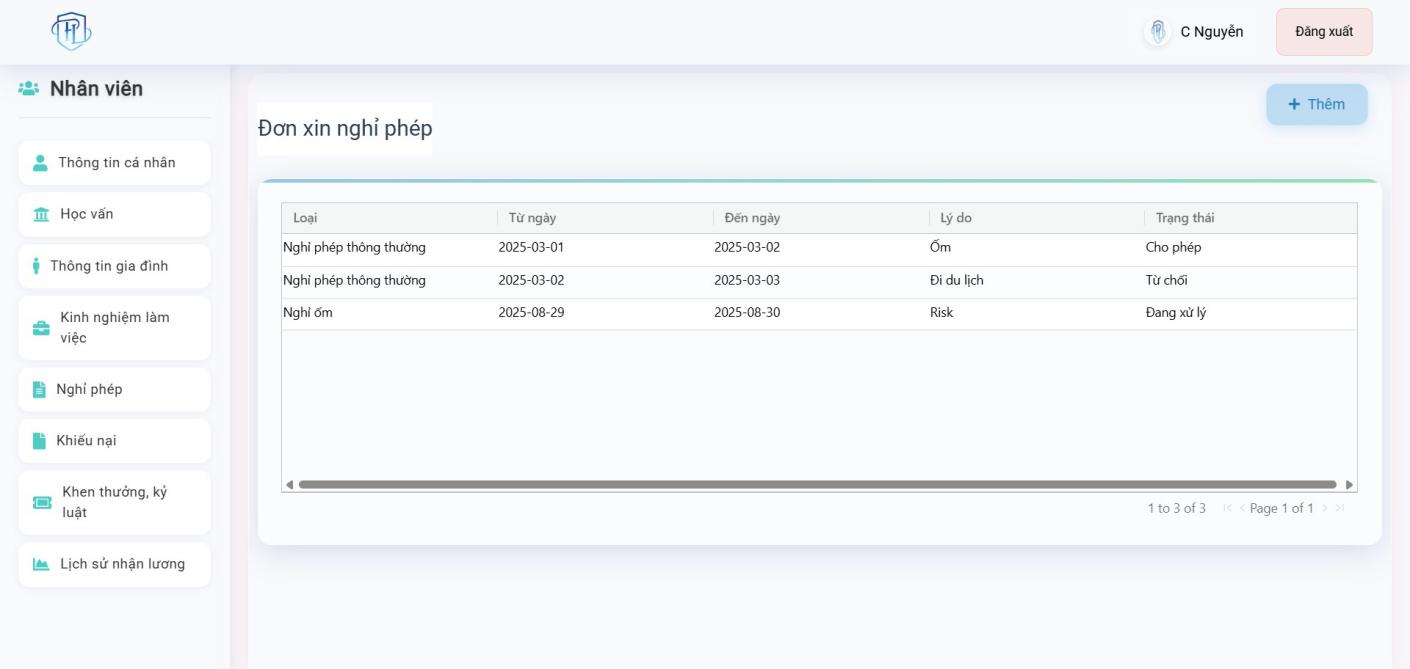
Hình 3.10: Giao diện quản lý thông tin cá nhân

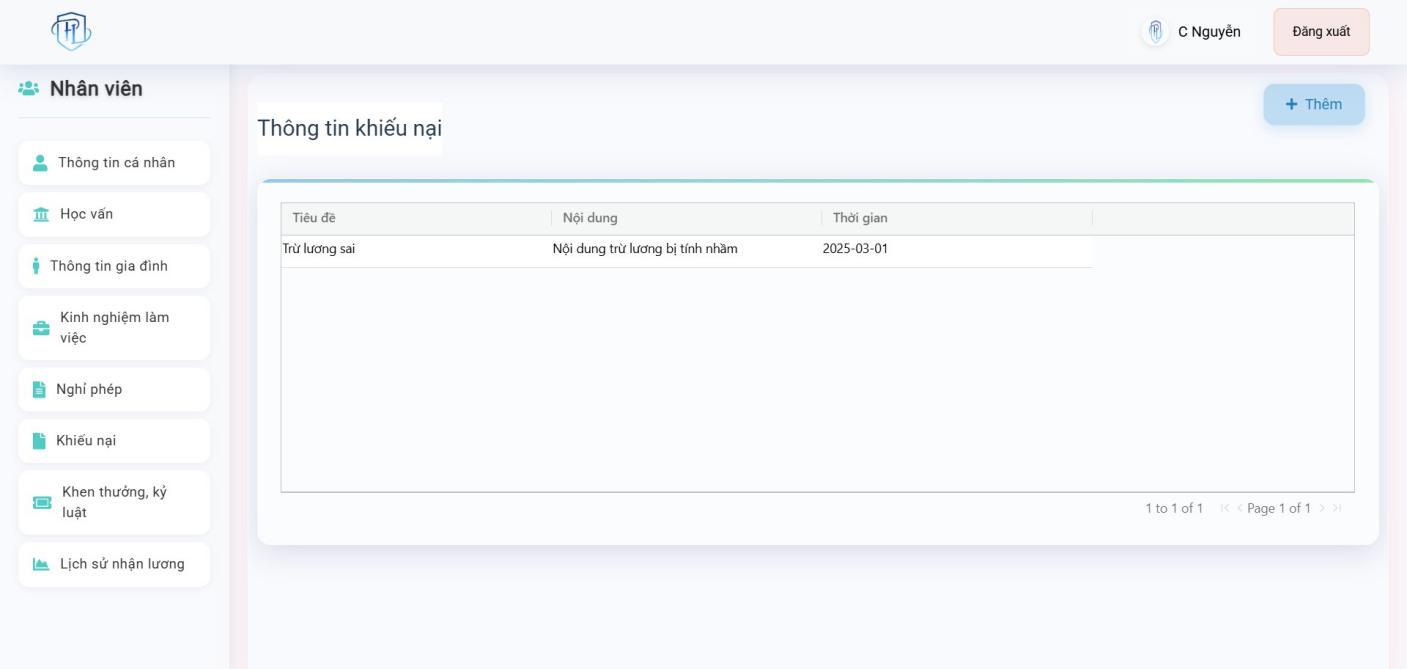
Hình 3.11: Giao diện quản lý học vấn



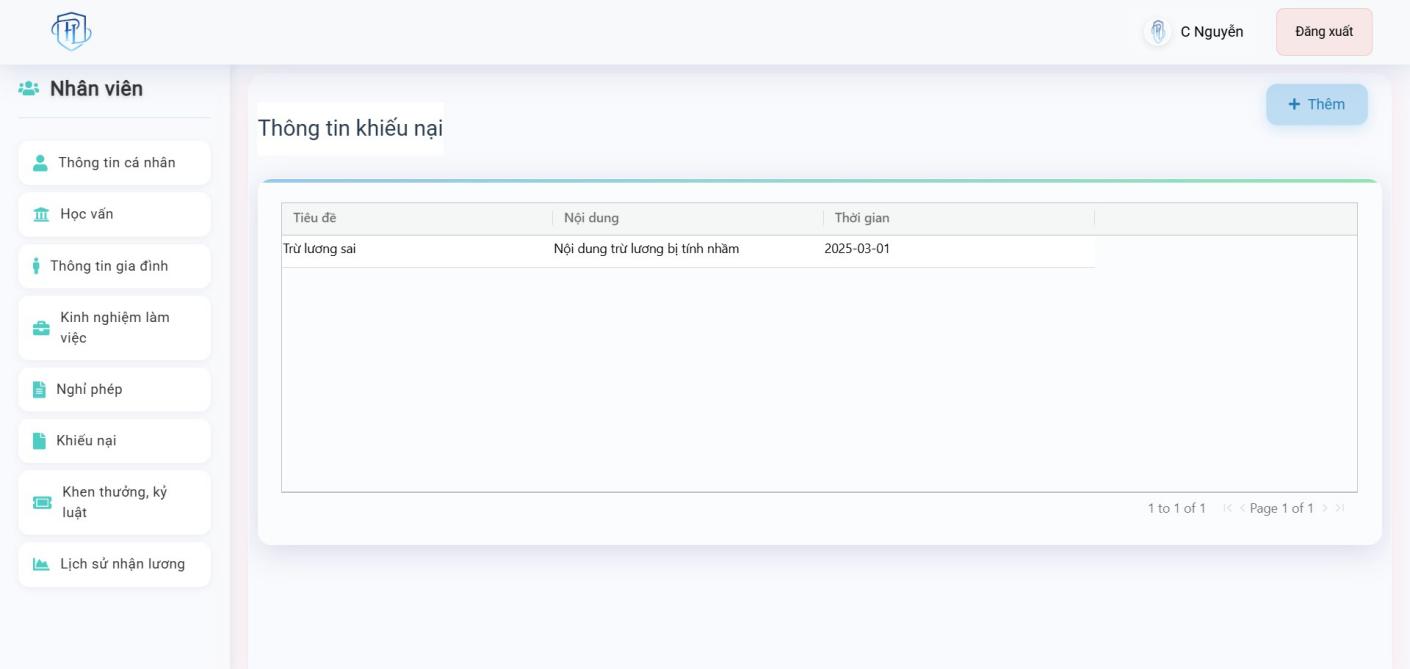
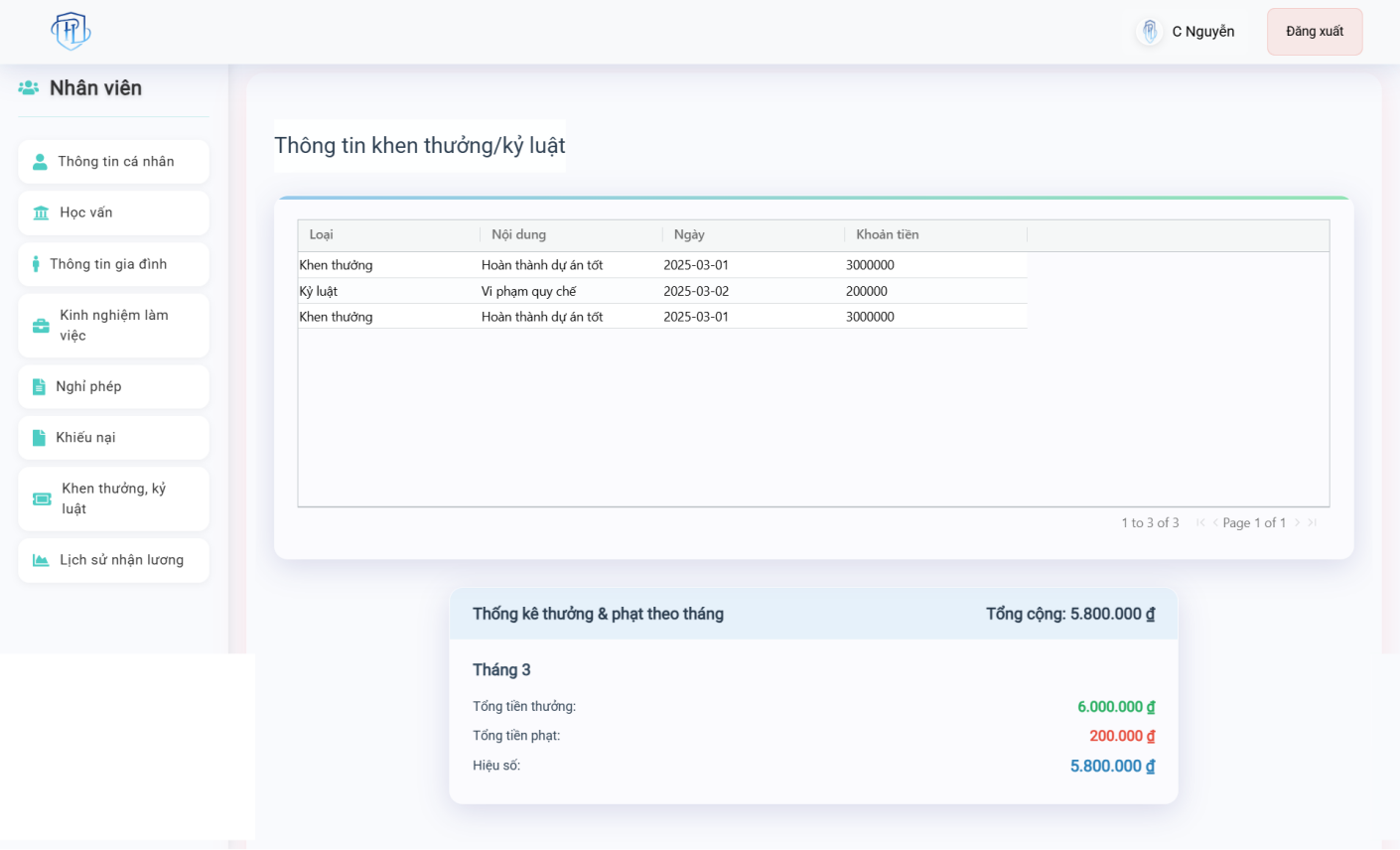
Hình 3.12 Giao diện quản lý thông tin gia đình

Hình 3.13 Giao diện quản lý thông tin kinh nghiệm làm việc

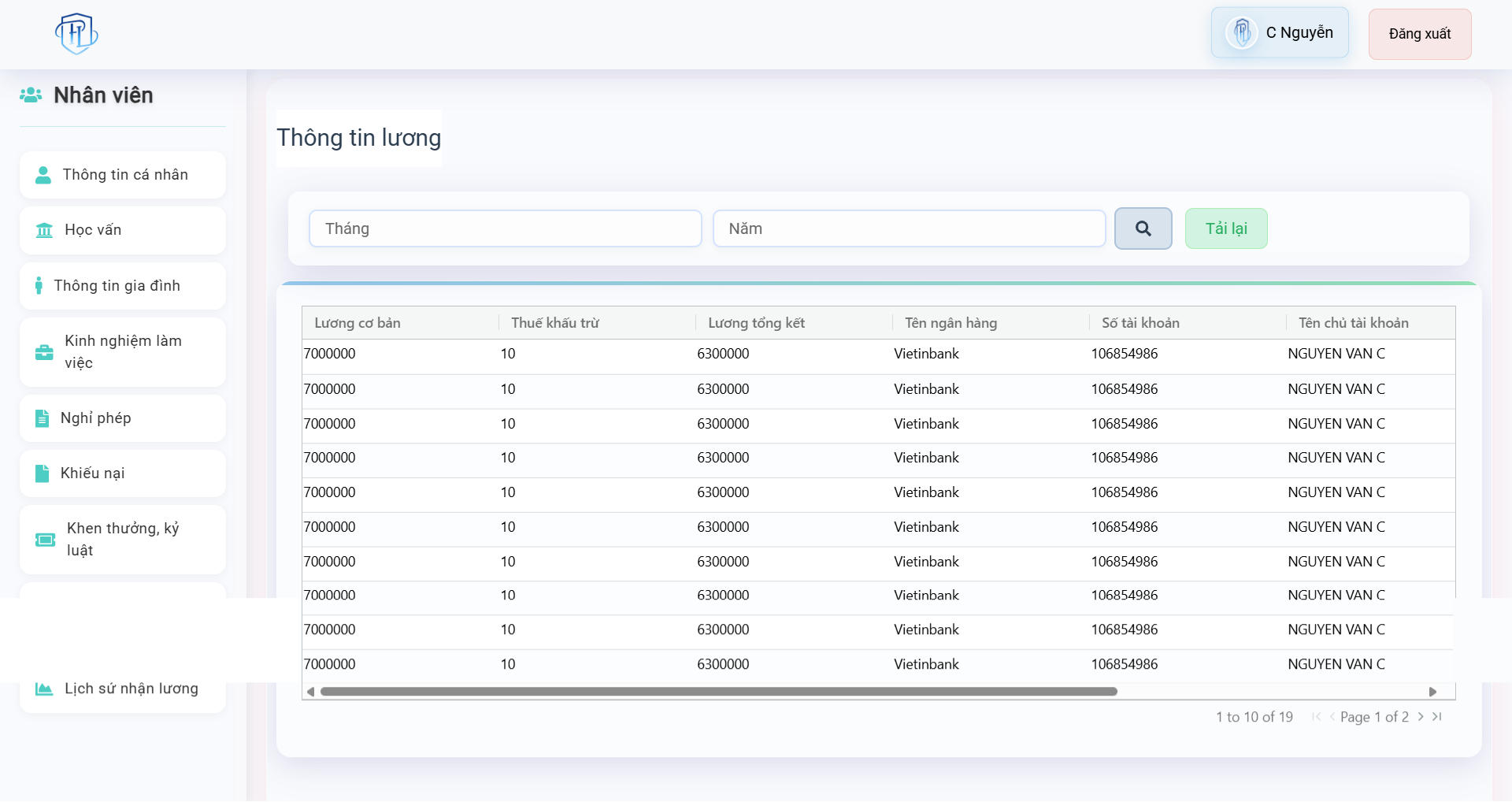


Hình 3.14 Giao diện tạo đơn xin nghỉ phép

Hình 3.15 Giao diện tạo đơn nghỉ phép

Hình 3.16 Giao diện tạo đơn khiếu nại

Hình 3.17 Giao diện thông tin khen thưởng/kỷ luật



Hình 3.18 Giao diện lịch sử nhận lương

# KẾT LUẬN

**Những gì đạt được**

Xây dựng hệ thống quản lý nhân viên hoàn chỉnh: Hệ thống đã được phát triển với đầy đủ chức năng cần thiết, bao gồm quản lý thông tin nhân viên, quản lý lương, đơn xin nghỉ phép, và phân quyền người dùng. Điều này giúp cho các nhà quản lý có thể theo dõi và quản lý nhân sự một cách hiệu quả hơn.

Nâng cao tính bảo mật và an toàn dữ liệu: Hệ thống đã được thiết kế với các biện pháp bảo mật cần thiết, bao gồm xác thực người dùng, phân quyền rõ ràng và mã hóa dữ liệu nhạy cảm. Điều này giúp đảm bảo rằng thông tin cá nhân và dữ liệu lương của nhân viên được bảo vệ an toàn trước các mối đe dọa từ bên ngoài.

**Những khó khăn**

Tải trọng hệ thống: Khi thử nghiệm với số lượng người dùng đồng thời lớn, hiệu suất của hệ thống có xu hướng giảm, dẫn đến thời gian phản hồi kéo dài. Điều này đã gây ra một số khó khăn trong việc đảm bảo rằng tất cả người dùng có thể truy cập hệ thống mà không gặp phải hiện tượng chậm trễ.

Tối ưu hóa truy vấn cơ sở dữ liệu: Việc tối ưu hóa các truy vấn đến cơ sở dữ liệu MongoDB để giảm thiểu thời gian truy xuất dữ liệu cũng là một thách thức. Một số truy vấn phức tạp đã yêu cầu nhiều tài nguyên và thời gian xử lý.

Giao diện người dùng: Mặc dù đã cố gắng thiết kế giao diện thân thiện, một số người dùng vẫn cảm thấy khó khăn trong việc sử dụng một số chức năng nhất định. Việc thu thập phản hồi từ người dùng để cải thiện giao diện và các bước thao tác là cần thiết nhưng gặp phải một số hạn chế về thời gian.

Hướng dẫn sử dụng: Việc cung cấp hướng dẫn đầy đủ và dễ hiểu cho tất cả các chức năng trong hệ thống cũng là một thách thức. Không phải tất cả người dùng đều có kinh nghiệm sử dụng công nghệ, do đó, việc giải thích rõ ràng và đơn giản là rất quan trọng.

Kiến trúc hệ thống: Mặc dù hệ thống được thiết kế với khả năng mở rộng, nhưng một số thành phần của kiến trúc đã gặp khó khăn trong việc tích hợp các tính năng mới mà không làm ảnh hưởng đến các chức năng hiện tại. Điều này đã yêu cầu em phải xem xét lại một số quyết định thiết kế ban đầu.

Quản lý dữ liệu lớn: Khi lượng dữ liệu ngày càng tăng, việc lưu trữ và quản lý dữ liệu hiệu quả trở thành một thách thức. Cần phải có các chiến lược tối ưu hóa để đảm bảo rằng cơ sở dữ liệu có thể hoạt động hiệu quả với khối lượng dữ liệu lớn trong tương lai.

**Hướng phát triển**

Tối ưu hóa truy vấn: em sẽ tiếp tục nghiên cứu và tối ưu hóa các truy vấn cơ sở dữ liệu để đảm bảo tốc độ xử lý nhanh chóng, đặc biệt khi có nhiều người dùng truy cập đồng thời. Việc sử dụng các chỉ mục phù hợp và kỹ thuật phân tán dữ liệu sẽ giúp cải thiện hiệu suất tổng thể của hệ thống.

Nâng cấp hạ tầng: Đầu tư vào hạ tầng máy chủ và các dịch vụ đám mây để đáp ứng tốt hơn với khối lượng người dùng và dữ liệu lớn sẽ giúp cải thiện hiệu suất hoạt động của hệ thống.

Cải thiện giao diện: em sẽ thu thập phản hồi từ người dùng để thiết kế lại giao diện người dùng sao cho thân thiện và trực quan hơn. Việc thử nghiệm A/B với các phiên bản giao diện khác nhau có thể giúp tìm ra phiên bản tốt nhất cho người dùng.

Cung cấp hướng dẫn và hỗ trợ: Tạo các tài liệu hướng dẫn sử dụng chi tiết và video hướng dẫn để giúp người dùng dễ dàng tiếp cận và sử dụng hệ thống. Ngoài ra, việc thiết lập một hệ thống hỗ trợ trực tuyến để giải đáp thắc mắc cũng rất cần thiết.

Thêm các tính năng mới: em dự kiến phát triển thêm các tính năng như quản lý đánh giá hiệu suất nhân viên, báo cáo phân tích dữ liệu và tích hợp các công cụ hỗ trợ như biểu đồ và thống kê để giúp nhà quản lý có cái nhìn tổng quát hơn về nhân sự.

Tích hợp với các hệ thống khác: Xem xét tích hợp hệ thống với các phần mềm quản lý khác trong doanh nghiệp như hệ thống kế toán, quản lý dự án để tăng cường tính đồng bộ và giảm thiểu công việc nhập liệu thủ công.

Cập nhật các biện pháp bảo mật: Thường xuyên rà soát và cập nhật các biện pháp bảo mật để đối phó với các mối đe dọa mới từ bên ngoài. Việc tổ chức đào tạo về bảo mật cho người dùng cũng là một phần quan trọng trong việc đảm bảo an toàn thông tin.

Tuân thủ quy định về bảo vệ dữ liệu: Thực hiện các biện pháp cần thiết để tuân thủ các quy định và luật pháp về bảo vệ dữ liệu cá nhân như GDPR, nhằm bảo vệ quyền lợi cho người dùng và tránh rủi ro pháp lý.

Xây dựng cộng đồng hỗ trợ: Tạo dựng một cộng đồng người dùng nơi họ có thể trao đổi ý kiến, chia sẻ kinh nghiệm và hỗ trợ lẫn nhau trong quá trình sử dụng hệ thống. Điều này không chỉ giúp tăng cường sự tương tác mà còn tạo điều kiện cho việc thu thập phản hồi hữu ích cho việc cải tiến hệ thống.

Tổ chức các buổi đào tạo: Định kỳ tổ chức các buổi đào tạo hoặc hội thảo về cách sử dụng hệ thống, giúp người dùng nắm rõ các chức năng và tối ưu hóa quy trình làm việc của mình.

Thiết lập hệ thống đánh giá: Thiết lập một hệ thống đánh giá và theo dõi định kỳ để xác định những điểm mạnh và điểm yếu của hệ thống. Việc này sẽ giúp em có cái nhìn rõ ràng hơn về hiệu suất hoạt động và nhu cầu của người dùng.

Lập kế hoạch cải tiến: Dựa trên kết quả đánh giá, xây dựng kế hoạch cải tiến và phát triển hệ thống trong tương lai, đảm bảo rằng hệ thống luôn đáp ứng tốt nhất nhu cầu của doanh nghiệp.

# TÀI LIỆU THAM KHẢO

[1] Alex Banks & Eve Porcello, Learning React, NXB O'Reilly Media.

[2] Joan Ayebola, “Cách làm việc với API RESTful trong React”, <https://www.freecodecamp.org/news/how-work-with-restful-apis-in-react-simplified-steps-and-practical-examples/>

[3] Kayode Adeniyi, “Hướng dẫn tối ưu hóa API với Node.js”, <https://www.freecodecamp.org/news/how-to-optimize-nodejs-apis/>

[4] Mario Casciaro & Luciano Mammino, Node.js Design Patterns, NXB O'Reilly Media.

[5] Shannon Bradshaw, Eoin Brazil & Kristina Chodorow, MongoDB: The Definitive Guide, NXB O'Reilly Media.