

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng

Người hướng dẫn: TS. Lê Văn Minh

Sinh viên thực hiện: **Phạm Minh Huy**

Mã số sinh viên: **2151220024**

Lớp: **21CT1**

Đà Nẵng, 01/2026

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

***Xây dựng website quản lý nhân viên và ca
làm việc của nhân viên cho chuỗi cửa hàng***

Người hướng dẫn: **TS. Lê Văn Minh**

Sinh viên thực hiện: **Phạm Minh Huy**

Mã số sinh viên: **2151220024**

Lớp: **21CT1**

Đà Nẵng, 01/2026

TÓM TẮT

Tên đề tài: ... Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng.

Sinh viên thực hiện: Phạm Minh Huy

Mã số SV: 2151220024 Lớp: 21CT1

Đề tài xây dựng hệ thống website quản lý nhân viên và ca làm việc cho chuỗi cửa hàng nhằm hỗ trợ quá trình quản lý nhân viên và ca làm việc, phân công ca làm, chấm công và theo dõi hiệu suất làm việc một cách hiệu quả. Hệ thống được phát triển với hai phần chính: backend sử dụng Node.js và MongoDB để xử lý dữ liệu, xác thực người dùng, quản lý nhân viên, lịch làm việc, chấm công, đổi ca và thống kê; frontend sử dụng ReactJS để xây dựng giao diện trực quan, thân thiện và dễ sử dụng cho quản lý và nhân viên.

Hệ thống cho phép quản lý thông tin nhân viên, phân công và điều chỉnh ca làm việc, hỗ trợ đăng ký đổi ca, duyệt yêu cầu đổi ca, thực hiện check-in/check-out, theo dõi thời gian làm việc thực tế và quản lý lịch sử làm việc của nhân viên. Đồng thời, hệ thống cung cấp các chức năng thống kê số giờ làm, tình trạng đi làm, hiệu suất làm việc và báo cáo phục vụ công tác quản lý nhân viên và ca làm việc trong chuỗi cửa hàng.

Dự án góp phần tự động hóa quy trình quản lý nhân viên, giảm thiểu sai sót trong việc phân công và theo dõi ca làm, nâng cao hiệu quả quản lý cũng như trải nghiệm làm việc của nhân viên. Hệ thống đảm bảo tính bảo mật thông tin, khả năng mở rộng và dễ dàng tích hợp thêm các chức năng quản lý nhân viên và ca làm việc trong tương lai, phù hợp với xu hướng chuyển đổi số trong hoạt động quản lý doanh nghiệp bán lẻ hiện nay.

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý nhân viên và ca làm việc vào hoạt động vận hành doanh nghiệp đã trở thành nhu cầu thiết yếu. Đặc biệt đối với các chuỗi cửa hàng bán lẻ, việc quản lý nhân viên, phân công ca làm việc, theo dõi thời gian làm việc và xử lý các yêu cầu liên quan đến lịch làm việc bằng phương pháp thủ công thường gây mất nhiều thời gian, dễ xảy ra sai sót và khó kiểm soát. Vì vậy, việc xây dựng một hệ thống quản lý nhân viên và ca làm việc trên nền tảng web là một giải pháp cần thiết, góp phần nâng cao hiệu quả quản lý.

Đề tài “Xây dựng website quản lý nhân viên và ca làm việc cho chuỗi cửa hàng” tập trung phát triển một hệ thống quản lý nhân viên và ca làm việc hoàn chỉnh với các chức năng như quản lý thông tin nhân viên, phân công ca làm việc, đăng ký và duyệt yêu cầu đổi ca, chấm công thông qua check-in/check-out, theo dõi thời gian làm việc thực tế, quản lý lịch sử làm việc và hỗ trợ thống kê dữ liệu phục vụ công tác quản lý. Hệ thống được xây dựng với giao diện trực quan, thân thiện, giúp cả quản lý và nhân viên dễ dàng sử dụng trong quá trình làm việc.

Quá trình thực hiện đề tài đã giúp em củng cố và nâng cao kiến thức về phát triển ứng dụng web hiện đại, hiểu rõ hơn về quy trình xây dựng hệ thống thực tế từ khâu phân tích yêu cầu, thiết kế cơ sở dữ liệu, phát triển giao diện người dùng đến triển khai các chức năng nghiệp vụ. Đồng thời, em cũng có cơ hội rèn luyện kỹ năng lập trình, tư duy giải quyết vấn đề và làm việc với các công nghệ phổ biến như ReactJS, Node.js và MongoDB.

Trong suốt thời gian thực hiện đồ án, em đã nhận được sự hướng dẫn tận tình của thầy Lê Văn Minh. Thầy đã luôn hỗ trợ, định hướng và truyền đạt những kiến thức quý báu, giúp em vượt qua những khó khăn trong quá trình nghiên cứu và hoàn thành đề tài. Em xin gửi lời cảm ơn chân thành đến thầy, cùng gia đình và bạn bè đã luôn động viên, giúp đỡ và tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và thực hiện đồ án.

Mặc dù đã nỗ lực hoàn thành đề tài với tinh thần học hỏi và trách nhiệm cao nhất, nhưng do kiến thức và kinh nghiệm thực tế còn hạn chế nên báo cáo khó tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý thầy cô để có thể hoàn thiện hơn trong tương lai.

Em hy vọng rằng hệ thống được xây dựng trong đề tài này sẽ góp phần hỗ trợ hiệu quả công tác quản lý nhân viên và ca làm việc tại các chuỗi cửa hàng, đồng thời là nền tảng để tiếp tục nghiên cứu, mở rộng và phát triển thêm nhiều chức năng hữu ích trong tương lai.

CAM ĐOAN

Em xin cam đoan đồ án “Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng (sử dụng ReactJS, Node.js và MongoDB).” là kết quả của quá trình học tập, nghiên cứu và thực hiện của bản thân dưới sự hướng dẫn của thầy Lê Văn Minh. Các số liệu, kết quả và nội dung trình bày trong báo cáo đều trung thực. Em hoàn toàn chịu trách nhiệm về tính trung thực của đồ án này.

Sinh viên thực hiện

Phạm Minh Huy

MỤC LỤC

TÓM TẮT.....	3
LỜI NÓI ĐẦU.....	i
CAM ĐOAN.....	ii
MỤC LỤC.....	iii
DANH SÁCH CÁC BẢNG, HÌNH VẼ.....	v
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	vii
MỞ ĐẦU.....	1
Chương 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT VÀ ĐỀ TÀI.....	2
1.1. Tổng quan về cơ sở lý thuyết.....	2
1.1.1. Ngôn ngữ lập trình ReactJS và Nodejs.....	2
1.1.2. Cơ sở lý thuyết về MongoDB.....	3
1.1.3. Công cụ thiết kế (Draw.io, UML).....	5
1.1.4. RESTful API.....	7
1.2. Mô tả các chức năng chính của hệ thống.....	7
1.2.1. Phân tích các tính năng kỹ thuật nổi bật.....	7
1.2.2. Phát biểu bài toán.....	10
1.3. Danh sách người sử dụng hệ thống.....	10
1.3.1. Người quản lý (Admin):.....	10
1.3.2. Nhân viên (User):.....	11
1.4. Các yêu cầu của người dùng.....	12
Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	14
2.1. Phân tích yêu cầu hệ thống.....	14
2.1.1. Yêu cầu chức năng.....	14
2.1.2. Yêu cầu phi chức năng.....	15
2.1.3. Các ràng buộc và giả định.....	15
2.2. Mô hình hóa hệ thống.....	16
2.2.1. Sơ đồ ngữ cảnh.....	16
2.2.2. Sơ đồ luồng dữ liệu (DFD).....	18
2.3. Thiết kế hệ thống.....	19
2.3.1. Thiết kế kiến trúc tổng thể.....	19
2.3.2. Thiết kế cơ sở dữ liệu (ERD, bảng dữ liệu).....	20
2.3.3. Nội dung thiết kế giao diện người dùng (UI/UX).....	22
2.4. Sơ đồ hoạt động và các sơ đồ minh họa.....	23

2.4.1. Sơ đồ Use Case.....	23
2.4.2. Sơ đồ hoạt động của các chức năng chính.....	26
2.4.3. Sơ đồ trình tự (Sequence Diagram).....	27
2.4.4. Sơ đồ lớp (Class Diagram).....	37
2.5. Tổng kết chương.....	38
Chương 3: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG.....	40
3.1. Tổng quan về xây dựng và triển khai hệ thống.....	40
3.2. Môi trường và công cụ phát triển.....	40
3.2.1. Ngôn ngữ lập trình và Framework.....	40
3.2.2. Cơ sở dữ liệu.....	41
3.2.3. Công cụ hỗ trợ phát triển.....	41
3.2.4. Thư viện dùng để tích hợp.....	41
3.3. Quá trình xây dựng hệ thống.....	41
3.3.1. Khởi tạo dự án và cấu trúc thư mục.....	41
3.3.2. Thiết kế và xây dựng cơ sở dữ liệu.....	42
3.3.3. Phát triển BackEnd.....	46
3.3.4. Phát triển Frontend.....	47
3.3.5. Tích hợp các chức năng và dịch vụ.....	49
3.4. Xây dựng các chức năng chính.....	50
3.4.1. Đăng kí/Đăng nhập.....	50
3.4.2. Giao diện nhân viên.....	51
3.4.3. Giao diện Admin.....	53
3.5. Kiểm thử và đánh giá hệ thống.....	55
3.5.1. Mục tiêu kiểm thử.....	55
3.5.2. Quy trình và phương pháp kiểm thử.....	55
3.5.3. Kết quả kiểm thử.....	56
3.5.4. Đánh giá tổng hệ thống.....	57
3.6. Tổng kết chương.....	57
KẾT LUẬN.....	58
TÀI LIỆU THAM KHẢO.....	61

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 1.1: Tính năng kỹ thuật của hệ thống “ Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng (sử dụng ReactJS, Node.js và MongoDB).”	8
Bảng 1.2: Tính năng quản lý ca làm việc	8
Bảng 1.3: Tính năng quản lý yêu cầu đổi ca	9
Bảng 1.4: Tính năng quản lý người dùng.....	9
Bảng 1.5: Tính năng thống kê và báo cáo	10
Bảng 2.1: Bảng User dữ liệu của hệ thống.....	22
Hình 1.1: Hình ảnh ReactJS.....	3
Hình 1.2: Hình ảnh Node js.....	3
Hình 1.3: Hình ảnh của database MongoDB	4
Hình 1.4: Hình ảnh minh hoạ lưu trữ của MongoDB.....	4
Hình 1.5: Hình ảnh xử lý dữ liệu của MongoDB.....	5
Hình 1.6: Tính năng mở rộng của MongoDB.....	5
Hình 1.7: Hình ảnh giao diện Draw.io.....	6
Hình 1.8: Hình ảnh sơ đồ mẫu của UML Diagrams	6
Hình 1.9: Hình ảnh RESTful API.....	7
Hình 2.1: Sơ đồ DFD ngữ cảnh.....	16
Hình 2.2: Sơ đồ luồng dữ liệu DFD.....	17
Hình 2.3: Hình ảnh thiết kế hệ thống tổng thể	18
Hình 2.4: Sơ đồ ERD hệ thống.....	19
Hình 2.5: Sơ đồ Use Case tổng quát của nhân viên.....	22
Hình 2.6: Sơ đồ Use Case quản lý (Admin)	23
<i>Hình 2.7: Sơ đồ usecase tổng quát</i>	<i>24</i>
Hình 2.8: Sơ đồ hoạt động các chức năng chính (Activity Diagram).....	25
Hình 2.9: Sơ đồ trình tự đăng nhập.....	27
Hình 2.10: Sơ đồ trình tự yêu cầu đổi ca (nhân viên).....	28
Hình 2.11: Sơ đồ trình tự trạng thái đổi ca (nhân viên).....	29
Hình 2.12: Sơ đồ trình tự xem thông tin cá nhân.....	30
Hình 2.13: Sơ đồ trình tự sửa thông tin cá nhân.....	31
Hình 2.14: Sơ đồ trình tự đăng kí ca làm việc.....	32
Hình 2.15: Sơ đồ trình tự xem ca làm việc.....	33
Hình 2.16: Sơ đồ trình tự duyệt yêu cầu đổi ca.....	34

Hình 2.17: Sơ đồ trình tự tạo lịch ca làm việc	35
Hình 2.18: Sơ đồ Lớp Class Diagram tổng quát.....	36
Hình 3.1: Bố cục giao diện Backend và Frontend.....	41
Hình 3.2: Các Collection chính của hệ thống trong CSDL MongoDB.....	42
Hình 3.3: Các thuộc tính của đối tượng người dùng (Users)	42
Hình 3.4: Các thuộc tính của nhân viên.....	43
Hình 3.5: Bố cục của thư mục Models trong Backend.....	45
Hình 3.6: Các API RESTful cho các chức năng trong mục Router	45
Hình 3.7: Bố cục admin.....	46
Hình 3.8: Bố cục các employee trong Frontend.....	47
Hình 3.9: Bố cục các components trong Frontend.....	48
Hình 3.10: Sơ đồ giao tiếp của Axios trong hệ thống.....	49
Hình 3.11: Giao diện đăng kí.....	49
Hình 3.12: Giao diện đăng nhập	50
Hình 3.13: Giao diện thông tin cá nhân nhân viên.....	50
Hình 3.14: Giao diện đăng kí ca làm việc của nhân viên.....	51
Hình 3.15: Giao diện lịch làm việc của nhân viên	51
Hình 3.16: Giao diện yêu cầu đổi ca.....	52
Hình 3.17: Giao diện tổng quan hệ thống.....	52
Hình 3.18: Giao diện quản lý nhân viên.....	53
Hình 3.19: Giao diện quản lý yêu cầu đổi ca.	53
Hình 3.20: Giao diện tạo ca làm cho nhân viên.....	54

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÝ HIỆU:

- DFD (Data Flow Diagram): Sơ đồ luồng dữ liệu, dùng để mô tả luồng thông tin giữa các tác nhân và hệ thống.
- BFD (Business Function Diagram): Sơ đồ các chức năng chính, dùng để phân tích và phân cấp các chức năng nghiệp vụ của hệ thống.
- ERD (Entity-Relationship Diagram): Sơ đồ quan hệ thực thể, dùng để thiết kế cơ sở dữ liệu, mô tả mối quan hệ giữa các thực thể.
- 3NF (Third Normal Form): Chuẩn hóa cơ sở dữ liệu dạng thứ 3, loại bỏ dư thừa dữ liệu và đảm bảo tính nhất quán.
- UI/UX (User Interface/User Experience): Giao diện người dùng/Trải nghiệm người dùng, ký hiệu dùng để chỉ thiết kế giao diện và tương tác.
- → (Arrow (Mũi tên)): Ký hiệu chỉ hướng luồng dữ liệu hoặc quy trình trong sơ đồ DFD, Activity Diagram.
- [] (Bracket (Ngoặc vuông)): Ký hiệu dùng trong sơ đồ Use Case để chỉ điều kiện hoặc ghi chú (ví dụ: trạng thái đơn hàng).
- () (Parenthesis (Ngoặc đơn)): Ký hiệu dùng trong sơ đồ Sequence để chỉ tham số hoặc thông điệp (ví dụ: yêu cầu API).

CHỮ VIẾT TẮT:

- ReactJS (React JavaScript): Thư viện JavaScript dùng xây dựng giao diện frontend, lập lại trong phần công nghệ và triển khai.
- NodeJS (Node JavaScript): Môi trường runtime JavaScript dùng để phát triển backend và xây dựng API.
- MongoDB (Mongo Database): Hệ quản trị cơ sở dữ liệu NoSQL dùng lưu trữ dữ liệu nhân viên, ca làm việc và tài khoản người dùng.
- UML (Unified Modeling Language): Ngôn ngữ mô hình hóa thống nhất, dùng thiết kế sơ đồ Use Case, Sequence, Class Diagram.
- API (Application Programming Interface): Giao diện lập trình ứng dụng, đặc biệt RESTful API để kết nối frontend-backend.

- RESTful (Representational State Transfer): Kiến trúc API dựa trên HTTP, dùng trao đổi dữ liệu JSON giữa các thành phần hệ thống.
- VS Code (Visual Studio Code): Môi trường phát triển mã nguồn, dùng lập trình Node.js và ReactJS.
- Git (Global Information Tracker): Hệ thống kiểm soát phiên bản mã nguồn, dùng quản lý dự án.
- Axios (Axios HTTP Client): Thư viện gửi yêu cầu HTTP từ frontend đến backend.
- Redux (Redux State Management): Thư viện quản lý trạng thái ứng dụng React, dùng đồng bộ dữ liệu giao diện.
- TS. (Tiến sĩ): Chữ viết tắt học vị của người hướng dẫn (TS. Lê Văn Minh).
- SV (Sinh viên): Chữ viết tắt chỉ sinh viên thực hiện (mã SV: 2151220024).

MỞ ĐẦU

Trong bối cảnh cuộc Cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ thông tin ngày càng đóng vai trò quan trọng trong hoạt động quản lý và vận hành doanh nghiệp. Đặc biệt đối với các chuỗi cửa hàng bán lẻ, việc quản lý nhân viên, phân công ca làm việc, theo dõi thời gian làm việc và xử lý các yêu cầu liên quan đến lịch trình làm việc bằng phương pháp thủ công thường gặp nhiều khó khăn, tốn thời gian và dễ phát sinh sai sót. Vì vậy, việc xây dựng một hệ thống quản lý nhân viên và ca làm việc trên nền tảng web là một giải pháp cần thiết nhằm nâng cao hiệu quả quản lý, tối ưu hóa nguồn nhân lực và đáp ứng yêu cầu chuyển đổi số trong doanh nghiệp hiện nay.

Đề tài “Xây dựng website quản lý nhân viên và ca làm việc cho chuỗi cửa hàng” được thực hiện nhằm xây dựng một hệ thống hỗ trợ quản lý thông tin nhân viên, phân công ca làm việc, quản lý lịch trình làm việc, đăng ký và duyệt yêu cầu đổi ca, chấm công thông qua chức năng check-in/check-out, đồng thời hỗ trợ theo dõi và thống kê thời gian làm việc của nhân viên. Hệ thống được thiết kế với giao diện trực quan, thân thiện và dễ sử dụng, đáp ứng nhu cầu của cả người quản lý và nhân viên trong quá trình sử dụng.

Mục tiêu của đề tài là xây dựng một website quản lý nhân viên và ca làm việc và ca làm việc hiệu quả, giúp tự động hóa các nghiệp vụ quản lý, giảm thiểu sai sót trong quá trình phân công và theo dõi nhân viên, đồng thời nâng cao khả năng giám sát, đánh giá hiệu suất làm việc. Bên cạnh đó, hệ thống còn hướng đến việc đảm bảo tính bảo mật dữ liệu, khả năng mở rộng và dễ dàng nâng cấp thêm các chức năng mới trong tương lai.

Phạm vi nghiên cứu của đề tài tập trung vào các chức năng chính như quản lý thông tin nhân viên, quản lý ca làm việc, phân công lịch làm, đăng ký đổi ca, xét duyệt yêu cầu đổi ca, chấm công, theo dõi thời gian làm việc thực tế và thống kê dữ liệu phục vụ công tác quản lý nhân viên và ca làm việc. Đối tượng nghiên cứu là các chuỗi cửa hàng bán lẻ có nhu cầu ứng dụng công nghệ vào công tác quản lý nhân viên và lịch làm việc.

Phương pháp nghiên cứu được sử dụng bao gồm khảo sát thực tế nhu cầu quản lý nhân viên và ca làm việc tại các cửa hàng, phân tích yêu cầu hệ thống, thiết kế cơ sở dữ liệu, xây dựng giao diện người dùng, phát triển các chức năng nghiệp vụ, kiểm thử và đánh giá hệ thống. Trong quá trình thực hiện đề tài, người thực hiện đã vận dụng các

kiến thức về lập trình web, cơ sở dữ liệu, phân tích thiết kế hệ thống và các công nghệ hiện đại như ReactJS, Node.js và MongoDB để xây dựng một sản phẩm có tính ứng dụng thực tiễn cao.

Cấu trúc đồ án gồm các phần: tổng quan đề tài, cơ sở lý thuyết, phân tích và thiết kế hệ thống, triển khai và kiểm thử hệ thống, đánh giá kết quả đạt được và đề xuất hướng phát triển trong tương lai. Nội dung báo cáo được trình bày theo trình tự logic và khoa học nhằm làm rõ quá trình nghiên cứu, thiết kế, xây dựng và ứng dụng hệ thống quản lý nhân viên và ca làm việc trong thực tế.

Chương 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT VÀ ĐỀ TÀI

1.1. Tổng quan về cơ sở lý thuyết

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý trên nền tảng web đã trở thành xu hướng tất yếu trong quá trình chuyển đổi số của doanh nghiệp. Đặc biệt đối với các chuỗi cửa hàng, việc quản lý nhân viên, phân công ca làm việc, theo dõi thời gian làm việc và xử lý các yêu cầu liên quan đến lịch làm việc bằng phương pháp thủ công thường gặp nhiều khó khăn, dễ xảy ra sai sót và tốn nhiều thời gian.

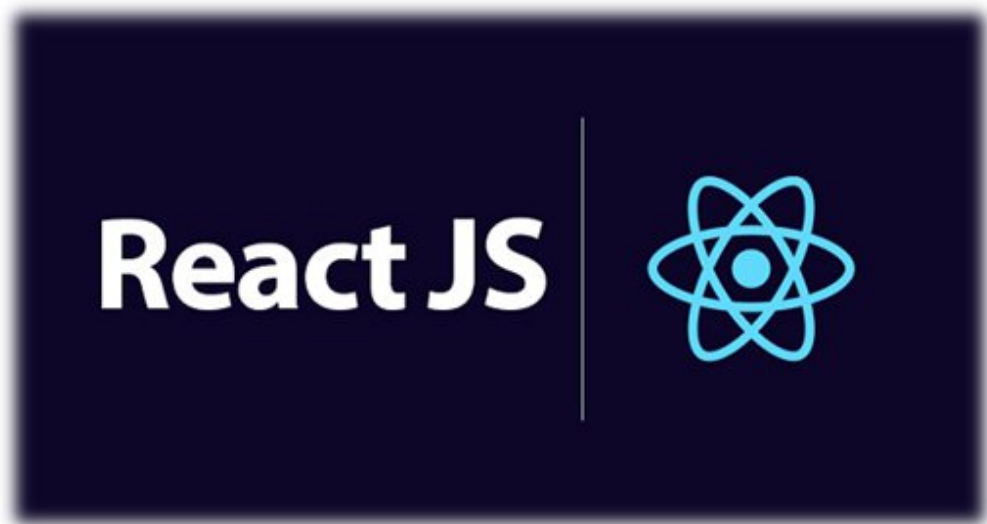
Việc xây dựng website quản lý nhân viên và ca làm việc giúp doanh nghiệp tối ưu hóa quy trình quản lý nhân viên và ca làm việc, nâng cao hiệu quả điều hành và hỗ trợ người quản lý theo dõi hoạt động của nhân viên một cách chính xác và thuận tiện hơn. Đồng thời, hệ thống còn giúp nhân viên dễ dàng theo dõi lịch làm việc, đăng ký ca làm, gửi yêu cầu đổi ca và cập nhật thông tin cá nhân mọi lúc, mọi nơi thông qua môi trường trực tuyến.

Các hệ thống quản lý hiện nay thường được xây dựng trên nền tảng web và sử dụng các công nghệ hiện đại như ReactJS, Node.js và MongoDB nhằm đảm bảo tính linh hoạt, khả năng mở rộng, hiệu năng xử lý cao và tính bảo mật của dữ liệu. Việc nghiên cứu và phát triển website quản lý nhân viên và ca làm việc là một hướng đi phù hợp với nhu cầu thực tế của các chuỗi cửa hàng, góp phần nâng cao hiệu quả quản lý nhân viên và ca làm việc và tối ưu hóa hoạt động vận hành của doanh nghiệp.

1.1.1. Ngôn ngữ lập trình ReactJS và Nodejs

Đề tài "Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng sử dụng ReactJS, Node.js và MongoDB" sử dụng ReactJS và Node.js là hai công nghệ phổ biến trong phát triển ứng dụng web hiện đại, giúp xây dựng hệ thống có khả năng hoạt động ổn định, hiệu quả và dễ mở rộng.

- *ReactJS*: Là thư viện JavaScript dùng để xây dựng giao diện người dùng. ReactJS cho phép tạo ra các thành phần giao diện động, tái sử dụng được và mang lại trải nghiệm mượt mà cho người dùng.



Hình 1.1: Hình ảnh ReactJS

- *Node.js*: Là môi trường chạy JavaScript trên server, hỗ trợ phát triển backend và xử lý các yêu cầu từ phía client. Node.js kết hợp với MongoDB giúp quản lý dữ liệu nhanh chóng và ổn định. Bằng việc kết hợp ReactJS và Node.js, hệ thống sẽ có khả năng xử lý yêu cầu người dùng và hiển thị thông tin một cách nhanh chóng, nâng cao trải nghiệm người dùng.



Hình 1.2: Hình ảnh Node js.

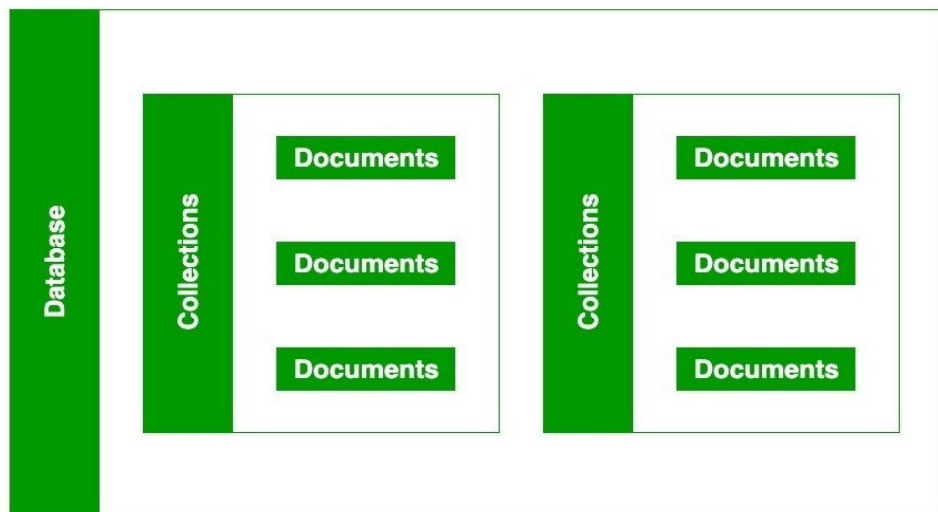
1.1.2. Cơ sở lý thuyết về MongoDB



Hình 1.3: Hình ảnh của database MongoDB

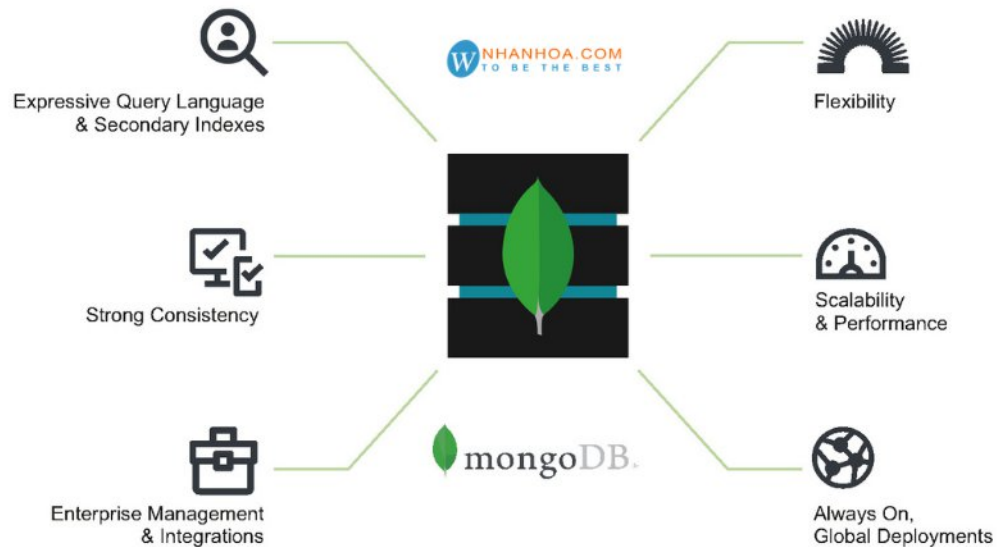
MongoDB là hệ quản trị cơ sở dữ liệu NoSQL mạnh mẽ, phù hợp với hệ thống quản lý dữ liệu phi cấu trúc và cho phép lưu trữ, truy xuất dữ liệu nhanh chóng. Các đặc điểm chính của MongoDB trong hệ thống này bao gồm:

- *Lưu trữ dạng document*: MongoDB lưu trữ dữ liệu dưới dạng JSON-like document, giúp tổ chức dữ liệu linh hoạt và mở rộng dễ dàng.



Hình 1.4: Hình ảnh minh họa lưu trữ của MongoDB

- *Khả năng xử lý dữ liệu lớn*: MongoDB được tối ưu hóa để xử lý một lượng lớn dữ liệu, phù hợp cho các trang web có lượng người dùng truy cập cao.



Hình 1.5: Hình ảnh xử lý dữ liệu của MongoDB

- *Tính năng mở rộng:* MongoDB cho phép mở rộng hệ thống theo chiều ngang (horizontal scaling), giúp duy trì hiệu suất hệ thống khi có nhiều người dùng. MongoDB sẽ lưu trữ thông tin, đồng thời hỗ trợ truy vấn dữ liệu nhanh chóng.

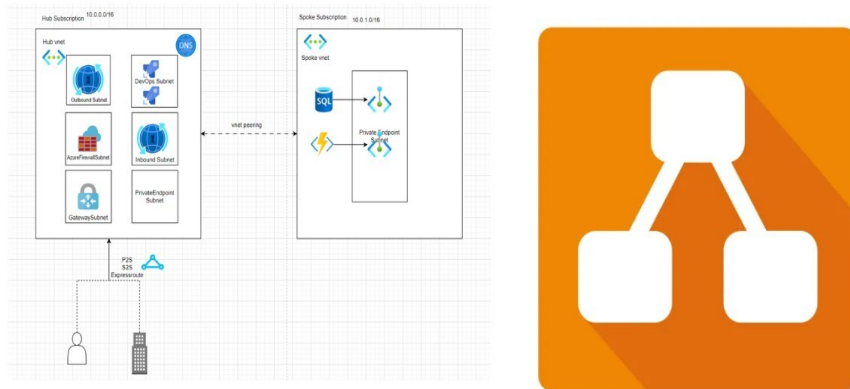


Hình 1.6: Tính năng mở rộng của MongoDB

1.1.3. Công cụ thiết kế (Draw.io, UML)

Việc sử dụng các công cụ thiết kế giúp tổ chức và trực quan hóa hệ thống một cách hiệu quả. Trong đề tài này:

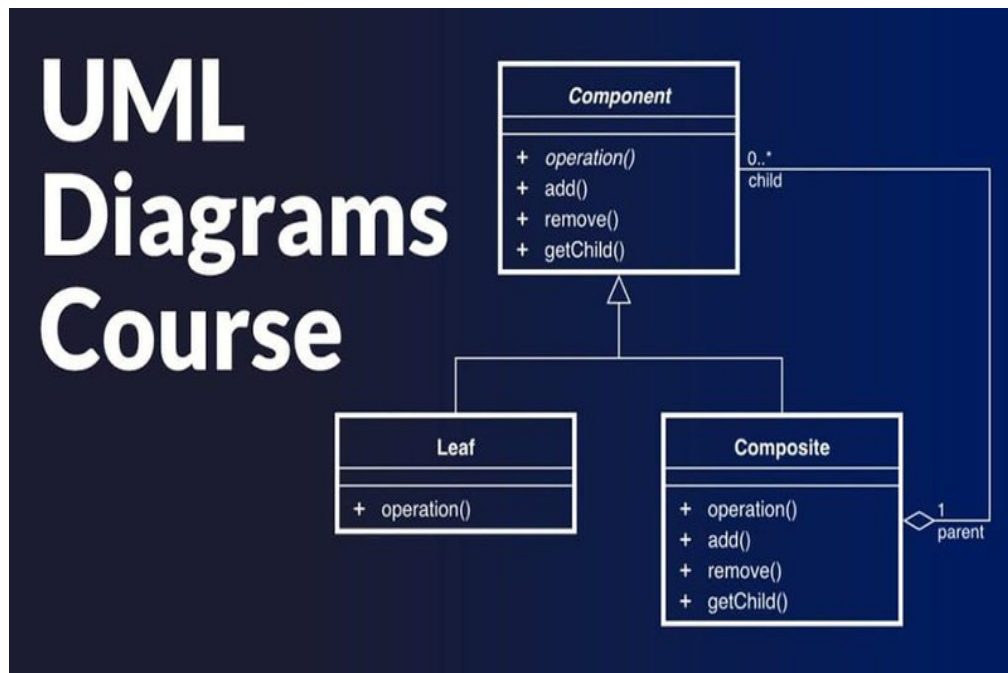
- *Draw.io:* Công cụ thiết kế trực tuyến giúp xây dựng các sơ đồ luồng dữ liệu, biểu đồ quan hệ thực thể và các sơ đồ khác để mô tả mô hình hệ thống.



Get started with draw.io

Hình 1.7: Hình ảnh giao diện Draw.io

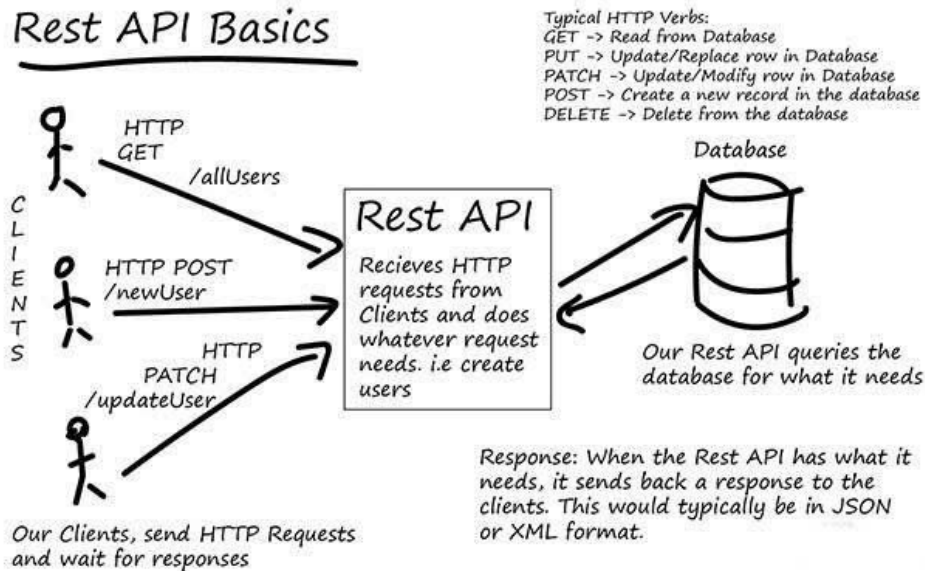
- *UML (Unified Modeling Language)*: Là ngôn ngữ mô hình hóa trực quan giúp thiết kế các biểu đồ lớp (class diagram), biểu đồ trình tự (sequence diagram) và biểu đồ use case. UML hỗ trợ trong việc phân tích và thiết kế hệ thống trước khi triển khai, giúp các thành phần hệ thống hoạt động mạch lạc và nhất quán. Các công cụ này sẽ được áp dụng trong suốt quá trình phát triển hệ thống, từ khâu thiết kế ban đầu đến khâu kiểm thử và triển khai, đảm bảo tính logic và chính xác của các thành phần trong hệ thống.



Hình 1.8: Hình ảnh sơ đồ mẫu của UML Diagrams

1.1.4. RESTful API

RESTful API (Representational State Transfer Application Programming Interface) là một kiểu kiến trúc phần mềm dùng để xây dựng các dịch vụ web, cho phép các hệ thống giao tiếp với nhau thông qua giao thức HTTP. REST tuân thủ các nguyên tắc như: sử dụng các phương thức HTTP (GET, POST, PUT, DELETE), tài nguyên được xác định bằng URL, dữ liệu trao đổi thường ở dạng JSON hoặc XML.



Hình 1.9: Hình ảnh RESTful API

Trong dự án website quản lý ca RESTful API đóng vai trò cầu nối giữa frontend (ReactJS) và backend (NodeJS). Frontend gửi các yêu cầu HTTP đến các endpoint của backend để thực hiện các chức năng như lấy danh sách nhân viên, tạo ca làm việc, đăng nhập, đăng ký, ... Backend nhận yêu cầu, xử lý dữ liệu và trả về kết quả cho frontend.

Việc sử dụng RESTful API giúp hệ thống dễ dàng mở rộng, bảo trì, tích hợp với các dịch vụ khác và đảm bảo tính nhất quán trong giao tiếp giữa các thành phần của ứng dụng.

1.2. Mô tả các chức năng chính của hệ thống

1.2.1. Phân tích các tính năng kỹ thuật nổi bật

“Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng.” được xây dựng với nhiều tính năng kỹ thuật hiện đại, đáp ứng nhu cầu quản lý nhân viên và ca làm việc và ca làm việc hiệu quả..

Bảng 1.1: Tính năng kỹ thuật của hệ thống “ Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng.”

Tính năng	Mô tả chức năng	Công nghệ sử dụng
Quản lý nhân viên	Thêm, sửa, xóa, tìm kiếm nhân viên	Nodejs, MongoDB
Quản lý hồ sơ cá nhân	Cập nhật thông tin cá nhân của nhân viên	Nodejs, MongoDB

Mô tả:

- *Quản lý nhân viên:* Cho phép thêm mới, chỉnh sửa, xóa và tìm kiếm nhân viên
- *Quản lý hồ sơ cá nhân:* Cho phép nhân viên xem và cập nhật thông tin cá nhân như họ tên, số điện thoại, email và địa chỉ....

Bảng 1.2: Tính năng quản lý ca làm việc

Tính năng	Mô tả chức năng	Công nghệ sử dụng
Quản lý ca làm việc	Tạo, sửa, xóa ca làm việc	Node.js, MongoDB
Xem lịch làm việc	Hiển thị lịch theo ngày, tuần, tháng	ReactJS, Node.js
Đăng ký ca làm việc	Nhân viên đăng ký ca làm việc	ReactJS, Node.js

Mô tả:

- *Quản lý ca làm việc:* Hỗ trợ quản lý tạo mới, chỉnh sửa, xóa và phân công các ca làm việc cho nhân viên. Chức năng này giúp tối ưu việc sắp xếp lịch làm việc và phân bổ nhân sự hợp lý.
- *Đăng ký ca làm việc:* Cho phép nhân viên lựa chọn và đăng ký các ca làm việc phù hợp với thời gian của mình. Hệ thống ghi nhận và cập nhật trạng thái đăng ký ca.
- *Xem lịch làm việc:* Cho phép nhân viên theo dõi lịch làm việc theo ngày, tuần hoặc tháng, giúp chủ động sắp xếp thời gian làm việc hiệu quả hơn.

Bảng 1.3: Tính năng quản lý yêu cầu đổi ca

Tính năng	Mô tả chức năng	Công nghệ sử dụng
Gửi yêu cầu đổi ca	Nhân viên gửi yêu cầu đổi ca	ReactJS, Node.js
Duyệt yêu cầu đổi ca	Quản lý duyệt hoặc từ chối yêu cầu	Node.js, MongoDB
Theo dõi trạng thái	Xem trạng thái xử lý yêu cầu đổi ca	ReactJS, MongoDB

Mô tả:

- *Gửi yêu cầu đổi ca:* Cho phép nhân viên gửi yêu cầu thay đổi ca làm việc khi có nhu cầu.
- *Duyệt yêu cầu đổi ca:* Cho phép quản lý xem xét, phê duyệt hoặc từ chối yêu cầu đổi ca.
- *Theo dõi trạng thái:* Giúp nhân viên theo dõi trạng thái xử lý của yêu cầu đổi ca như chờ duyệt, đã duyệt hoặc từ chối.

Bảng 1.4: Tính năng quản lý người dùng

Tính năng	Mô tả chức năng	Công nghệ sử dụng
Đăng nhập	Xác thực người dùng	JWT, Bcrypt
Phân quyền	Phân quyền nhân viên và quản lý	JWT
Đăng xuất	Kết thúc phiên làm việc	JWT

Mô tả:

- *Đăng nhập:* Cho phép người dùng xác thực tài khoản để truy cập vào hệ thống. Hệ thống kiểm tra thông tin đăng nhập và xác định quyền truy cập tương ứng với vai trò của người dùng.
- *Phân quyền:* Hỗ trợ phân quyền giữa quản lý và nhân viên nhằm đảm bảo mỗi người dùng chỉ được sử dụng các chức năng phù hợp với nhiệm vụ của mình.
- *Đăng xuất:* Cho phép người dùng kết thúc phiên làm việc và đăng xuất khỏi hệ thống một cách an toàn, giúp bảo vệ thông tin tài khoản và dữ liệu cá nhân.

Bảng 1.5: Tính năng thống kê và báo cáo

Tính năng	Mô tả chức năng	Công nghệ sử dụng
Thống kê nhân viên	Thống kê số lượng nhân viên	MongoDB Aggregation
Thống kê ca làm việc	Thống kê số lượng ca làm việc	MongoDB Aggregation
Thống kê yêu cầu đổi ca	Thống kê các yêu cầu đổi ca	MongoDB Aggregation

Mô tả:

- *Thống kê nhân viên:* Hỗ trợ tổng hợp và hiển thị số lượng nhân viên hiện có trong hệ thống, giúp quản lý dễ dàng theo dõi tình hình nhân sự của cửa hàng.
- *Thống kê ca làm việc:* Cung cấp thông tin về số lượng ca làm việc đã được tạo và phân công, hỗ trợ quản lý trong việc sắp xếp và điều phối nhân viên hợp lý.
- *Thống kê yêu cầu đổi ca:* Tổng hợp các yêu cầu đổi ca của nhân viên theo từng trạng thái như đã duyệt, từ chối hoặc đang chờ xử lý, giúp quản lý theo dõi và đánh giá tình hình thay đổi lịch làm việc.

1.2.2. Phát biểu bài toán

Bài toán đặt ra là xây dựng một “Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng (sử dụng ReactJS, Node.js và MongoDB).”, giúp tự động hóa các quy trình, quản lý nhân viên, quản lý ca, người dùng. Hệ thống cần đảm bảo tính bảo mật, dễ sử dụng, hỗ trợ quản lý trong việc quản lý nhân viên, ca làm việc và nâng cao hiệu quả vận hành của chuỗi cửa hàng.

1.3. Danh sách người sử dụng hệ thống

1.3.1. Người quản lý (Admin):

- *Quản lý nhân viên:* Thực hiện các chức năng thêm mới nhân viên, chỉnh sửa thông tin nhân viên, xóa hoặc vô hiệu hóa tài khoản nhân viên nghỉ việc, cập nhật thông tin cá nhân, vị trí công việc, chuyên môn và trạng thái làm việc của nhân viên trong hệ thống.
- *Quản lý lịch trình và ca làm việc:* Hỗ trợ tạo lịch làm việc, phân công ca làm cho nhân viên theo từng ngày hoặc từng tuần, cập nhật và điều chỉnh lịch làm việc khi có

thay đổi, theo dõi trạng thái ca làm như chờ check-in, đang làm việc và đã hoàn thành.-
Quản lý người dùng: Phân quyền truy cập cho các tài khoản, khóa hoặc mở tài khoản người dùng khi cần thiết, xem thông tin chi tiết của từng nhân viên để hỗ trợ và xử lý các vấn đề phát sinh.

- *Quản lý tài khoản và phân quyền*: Thực hiện duyệt tài khoản đăng ký của nhân viên, phân quyền truy cập theo từng vai trò (quản lý, nhân viên), khóa hoặc mở khóa tài khoản khi cần thiết, đảm bảo tính bảo mật và kiểm soát truy cập hệ thống.

- *Báo cáo và thống kê*: Thống kê số lượng nhân viên, số ca làm việc đã hoàn thành, số lần đổi ca, số giờ làm việc của nhân viên, tình trạng đi làm và các yêu cầu đổi ca. Hệ thống hỗ trợ hiển thị báo cáo trực quan giúp người quản lý dễ dàng theo dõi, đánh giá hiệu quả làm việc và đưa ra các quyết định quản lý phù hợp.

1.3.2. Nhân viên (User):

- *Đăng ký tài khoản*: Cho phép nhân viên tạo tài khoản mới bằng cách cung cấp các thông tin cá nhân cần thiết như họ tên, email, số điện thoại và mật khẩu. Sau khi đăng ký, tài khoản sẽ được gửi đến quản trị viên để xét duyệt trước khi được kích hoạt.

- *Đăng nhập hệ thống*: Truy cập vào hệ thống bằng tài khoản đã đăng ký để sử dụng các chức năng nâng cao.

- *Quản lý thông tin cá nhân*: Cho phép nhân viên xem và cập nhật các thông tin cá nhân như họ tên, số điện thoại, địa chỉ, email, ảnh đại diện và các thông tin liên quan khác. Hệ thống đảm bảo lưu trữ và cập nhật dữ liệu một cách an toàn và chính xác.

- *Đăng ký ca làm việc*: Nhân viên có thể xem danh sách các ca làm việc còn trống và thực hiện đăng ký nhận ca phù hợp với lịch trình cá nhân của mình. Sau khi được quản lý phê duyệt, ca làm sẽ được cập nhật vào lịch làm việc của nhân viên.

- *Xem lịch làm việc*: Cho phép nhân viên theo dõi lịch làm việc được phân công theo ngày, tuần hoặc tháng. Hệ thống hiển thị đầy đủ thông tin về thời gian làm việc, vị trí công việc và trạng thái ca làm.

- *Gửi yêu cầu đổi ca*: Trong trường hợp không thể tham gia ca làm đã được phân công, nhân viên có thể gửi yêu cầu đổi ca hoặc nhường ca cho nhân viên khác. Yêu cầu sẽ được chuyển đến quản lý để xem xét và phê duyệt.

- *Lịch sử đổi ca*: Cho phép nhân viên theo dõi và tra cứu toàn bộ lịch sử các yêu cầu đổi ca đã gửi trong quá trình làm việc. Hệ thống hiển thị chi tiết thông tin của từng yêu cầu như ca làm hiện tại, ca muốn đổi, lý do đổi ca, thời gian gửi yêu cầu, người được đề xuất đổi ca (nếu có) và trạng thái xử lý của yêu cầu (chờ duyệt, đã duyệt hoặc từ chối). Chức năng này giúp nhân viên dễ dàng theo dõi tiến trình xử lý yêu cầu, đồng thời hỗ trợ quản lý trong việc kiểm tra và đối chiếu lịch sử thay đổi ca làm việc.

- *Theo dõi lịch sử làm việc*: Cho phép nhân viên xem lại lịch sử các ca làm đã hoàn thành, thời gian check-in/check-out, tổng số giờ làm việc và các thông tin liên quan đến quá trình làm việc của bản thân.

1.4. Các yêu cầu của người dùng

- *Giao diện thân thiện, dễ sử dụng*: Hệ thống cần có giao diện trực quan, bố cục hợp lý, dễ thao tác trên cả máy tính và thiết bị di động, giúp người dùng tiếp cận các chức năng nhanh chóng.

- *Quản lý lịch làm việc hiệu quả*: Hệ thống phải hỗ trợ phân công, theo dõi và quản lý lịch làm việc của nhân viên một cách chính xác, giúp người quản lý dễ dàng sắp xếp ca làm và nhân viên thuận tiện theo dõi lịch trình của mình

- *Quản lý yêu cầu đổi ca rõ ràng*: Các yêu cầu đổi ca phải được hiển thị đầy đủ thông tin, bao gồm ca hiện tại, ca muốn đổi, lý do đổi ca và trạng thái xử lý. Hệ thống cần cập nhật liên tục trạng thái yêu cầu (chờ duyệt, đã duyệt, từ chối) để nhân viên và quản lý dễ dàng theo dõi.

- *Bảo mật thông tin*: Hệ thống phải đảm bảo an toàn cho dữ liệu cá nhân của nhân viên, thông tin tài khoản và dữ liệu quản lý. Việc xác thực đăng nhập và phân quyền truy cập cần được thực hiện chặt chẽ nhằm hạn chế các truy cập trái phép.

- *Chấm công chính xác*: Hệ thống phải hỗ trợ chức năng check-in và check-out, ghi nhận thời gian làm việc thực tế của nhân viên một cách chính xác, hỗ trợ theo dõi giờ làm việc, tình trạng đi muộn hoặc về sớm và phục vụ công tác thống kê sau này.

- *Quản lý tài khoản và phân quyền*: Quản trị viên cần có quyền duyệt tài khoản đăng ký của nhân viên, phân quyền sử dụng hệ thống, khóa hoặc mở khóa tài khoản khi cần thiết nhằm đảm bảo tính bảo mật và kiểm soát hoạt động của hệ thống.

Quản lý nhân viên toàn diện: Quản trị viên cần có khả năng quản lý toàn bộ thông tin nhân viên, lịch làm việc, yêu cầu đổi ca, chấm công và các báo cáo thống kê nhằm nâng cao hiệu quả quản lý nhân viên và ca làm việc trong chuỗi cửa hàng.

Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu hệ thống

Phân tích yêu cầu hệ thống là bước quan trọng nhằm xác định các chức năng cần thiết, các tiêu chí phi chức năng và những ràng buộc, giả định trong quá trình Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng. Việc xác định đúng và đủ các yêu cầu sẽ giúp hệ thống đáp ứng tốt nhu cầu thực tế.

2.1.1. Yêu cầu chức năng

- Hệ thống cho phép quản trị viên đăng nhập, quản lý thông tin nhân viên, thêm mới, chỉnh sửa, xóa hoặc cập nhật thông tin cá nhân, chức vụ và trạng thái làm việc của nhân viên.

- Quản trị viên có thể tạo và quản lý lịch làm việc, phân công ca làm cho nhân viên theo ngày hoặc tuần, theo dõi trạng thái ca làm và điều chỉnh lịch trình khi cần thiết.

- Chức năng quản lý tài khoản cho phép quản trị viên duyệt tài khoản đăng ký, phân quyền người dùng, khóa hoặc mở khóa tài khoản và xem thông tin chi tiết của từng nhân viên.

- Nhân viên có thể đăng ký tài khoản, đăng nhập hệ thống, cập nhật thông tin cá nhân, xem lịch làm việc, đăng ký nhận ca làm việc và theo dõi các ca làm đã được phân công.

- Hệ thống hỗ trợ chức năng gửi yêu cầu đổi ca, cho phép nhân viên đăng ký đổi ca hoặc nhường ca làm việc. Quản trị viên có thể xem, xét duyệt hoặc từ chối các yêu cầu đổi ca và cập nhật lịch làm việc tương ứng.

- Hệ thống cung cấp chức năng check-in và check-out để ghi nhận thời gian làm việc thực tế của nhân viên, hỗ trợ theo dõi giờ làm việc, tình trạng đi muộn, về sớm và lịch sử làm việc.

- Hệ thống cung cấp các chức năng thống kê và báo cáo như số lượng nhân viên, số ca làm việc đã hoàn thành, số yêu cầu đổi ca, tổng thời gian làm việc của nhân viên và các dữ liệu hỗ trợ công tác quản lý nhân viên và ca làm việc.

- Nhân viên có thể theo dõi lịch sử đổi ca, lịch sử làm việc, trạng thái các yêu cầu đã gửi và quá trình xử lý của quản trị viên một cách dễ dàng và trực quan.

2.1.2. Yêu cầu phi chức năng

- Giao diện website phải thân thiện, dễ sử dụng.
- Hệ thống phải đảm bảo tính bảo mật đối với thông tin cá nhân, tài khoản người dùng và dữ liệu quản lý nhân viên và ca làm việc. Các chức năng xác thực đăng nhập, phân quyền truy cập và mã hóa mật khẩu cần được triển khai nhằm hạn chế các truy cập trái phép
- Thời gian phản hồi của hệ thống phải nhanh, đảm bảo việc truy cập dữ liệu nhân viên, lịch làm việc, yêu cầu đổi ca và các chức năng quản lý diễn ra ổn định, không gây ảnh hưởng đến trải nghiệm người dùng.
- Hệ thống cần có khả năng mở rộng và nâng cấp trong tương lai, cho phép bổ sung thêm các chức năng như tính lương tự động, đánh giá hiệu suất làm việc, quản lý nghỉ phép hoặc tích hợp với các hệ thống quản lý khác.
- Dữ liệu của hệ thống phải được lưu trữ an toàn trên cơ sở dữ liệu MongoDB, có cơ chế sao lưu và phục hồi dữ liệu nhằm hạn chế rủi ro mất mát thông tin khi xảy ra sự cố.
- Website phải đảm bảo tính ổn định trong quá trình vận hành, hạn chế tối đa các lỗi phát sinh, đồng thời duy trì khả năng hoạt động liên tục để phục vụ nhu cầu quản lý nhân viên và ca làm việc.
- Hệ thống phải hỗ trợ truy cập trên nhiều thiết bị khác nhau như máy tính, máy tính bảng và điện thoại thông minh, đảm bảo giao diện hiển thị phù hợp và dễ sử dụng trên từng loại thiết bị.
- Dữ liệu về lịch làm việc, yêu cầu đổi ca, check-in/check-out và thống kê phải được cập nhật chính xác và đồng bộ, đảm bảo tính nhất quán trong quá trình quản lý và theo dõi hoạt động của nhân viên.

2.1.3. Các ràng buộc và giả định

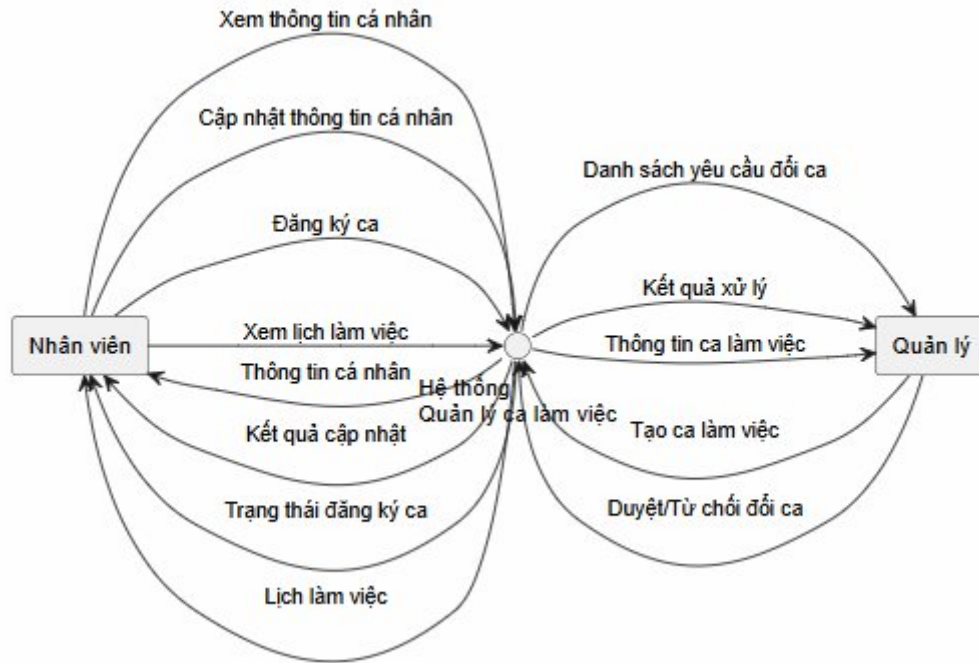
- Hệ thống được xây dựng dựa trên nền tảng web, sử dụng các công nghệ như Node.js, ReactJS, MongoDB.
- Người dùng cần có kết nối Internet ổn định để truy cập và sử dụng các chức năng của hệ thống như đăng nhập, xem lịch làm việc, đăng ký ca làm việc, gửi yêu cầu đổi ca và thực hiện chấm công.

- Hệ thống giả định rằng mỗi nhân viên sở hữu một tài khoản duy nhất và được quản trị viên phê duyệt trước khi sử dụng các chức năng của hệ thống.
- Việc phân công ca làm việc, phê duyệt yêu cầu đổi ca và quản lý thông tin nhân viên được thực hiện bởi quản trị viên hoặc người quản lý có quyền truy cập tương ứng.
- Dữ liệu nhân viên, lịch làm việc, yêu cầu đổi ca, lịch sử làm việc và thông tin tài khoản được lưu trữ tập trung trên cơ sở dữ liệu MongoDB nhằm đảm bảo tính nhất quán, bảo mật và dễ dàng quản lý.
- Hệ thống giả định rằng nhân viên thực hiện check-in và check-out đúng quy trình để đảm bảo tính chính xác của dữ liệu chấm công và thời gian làm việc thực tế.
- Trong trường hợp mất kết nối mạng hoặc xảy ra sự cố hệ thống, một số chức năng có thể bị gián đoạn tạm thời cho đến khi kết nối được khôi phục hoặc hệ thống hoạt động ổn định trở lại.
- Hệ thống được thiết kế để phục vụ nhu cầu quản lý nhân viên và ca làm việc cho các chuỗi cửa hàng có quy mô nhỏ và vừa, đồng thời có khả năng mở rộng để đáp ứng nhu cầu phát triển trong tương lai.

2.2. Mô hình hóa hệ thống

Trong quá trình phân tích và thiết kế hệ thống, việc mô hình hóa giúp xác định rõ các thành phần, chức năng và luồng dữ liệu giữa các tác nhân với hệ thống. Các sơ đồ như sơ đồ ngữ cảnh, sơ đồ luồng dữ liệu (DFD) và sơ đồ chức năng chính là công cụ trực quan hỗ trợ quá trình xây dựng hệ thống một cách logic, khoa học và phù hợp với yêu cầu thực tế của chuỗi cửa hàng.

2.2.1. Sơ đồ ngữ cảnh



Hình 2.1: Sơ đồ DFD ngữ cảnh

Mô tả sơ đồ hệ thống:

Sơ đồ ngữ cảnh dưới đây mô tả tổng quan các tác nhân bên ngoài và các luồng thông tin trao đổi giữa các tác nhân với hệ thống quản lý ca làm việc.

- Nhân viên:

- + Có thể xem thông tin cá nhân của mình trên hệ thống.
- + Có thể cập nhật thông tin cá nhân khi có thay đổi.
- + Có thể đăng ký ca làm việc theo lịch được cung cấp.
- + Có thể xem lịch làm việc theo ngày, tuần hoặc tháng.
- + Khi thực hiện các chức năng trên, hệ thống sẽ trả về thông tin cá nhân, kết quả cập nhật, trạng thái đăng ký ca và lịch làm việc tương ứng.

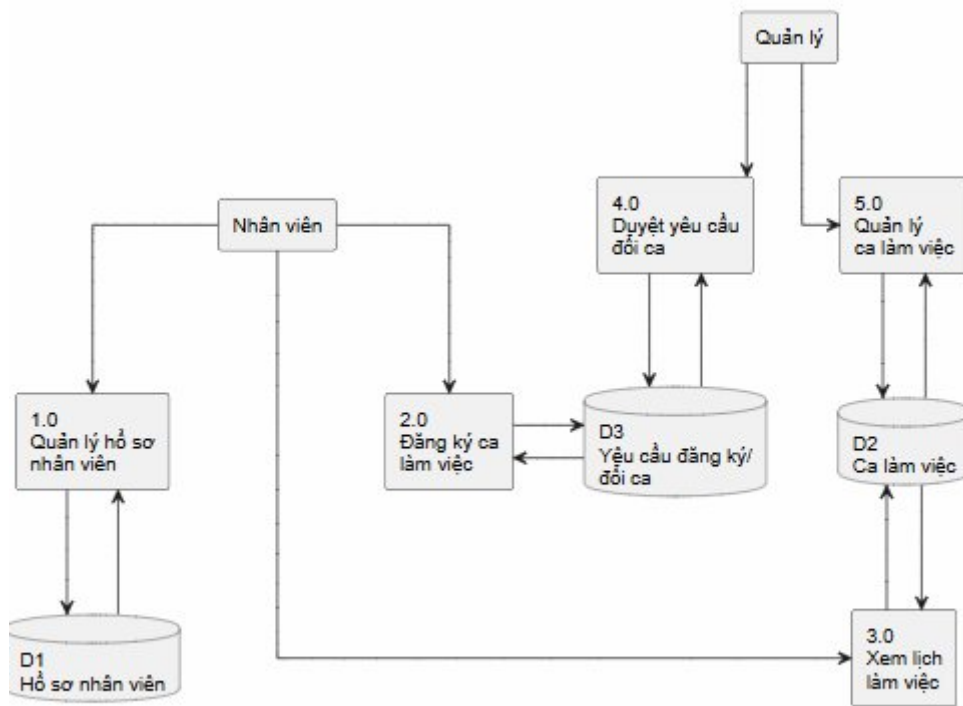
- Quản lý:

- + Có quyền tạo mới các ca làm việc và phân công lịch làm việc cho nhân viên.
- + Có thể xem danh sách các yêu cầu đổi ca từ nhân viên.
- + Có quyền duyệt hoặc từ chối các yêu cầu đổi ca.
- + Hệ thống sẽ cung cấp danh sách yêu cầu đổi ca, thông tin các ca làm việc và kết quả xử lý yêu cầu cho quản lý.

- Người quản lý (Admin):

- + Có quyền quản lý nhân viên, quản lý tài khoản, quản lý ca làm việc và xem các báo cáo thống kê.
- + Hệ thống sẽ trả về kết quả các thao tác quản lý và các báo cáo thống kê cho quản trị viên.

2.2.2. Sơ đồ luồng dữ liệu (DFD)



Hình 2.2: Sơ đồ luồng dữ liệu DFD

Mô tả hệ thống sơ đồ luồng dữ liệu:

Nhân viên: Thực hiện các chức năng xem thông tin cá nhân, cập nhật thông tin cá nhân, đăng ký ca làm việc và xem lịch làm việc. Hệ thống sẽ phản hồi lại thông tin cá nhân, kết quả cập nhật, trạng thái đăng ký ca và lịch làm việc tương ứng cho nhân viên.

Quản lý: Thực hiện các chức năng tạo ca làm việc, xem danh sách yêu cầu đổi ca và duyệt hoặc từ chối các yêu cầu đổi ca của nhân viên. Hệ thống sẽ trả về kết quả xử lý yêu cầu đổi ca, thông tin ca làm việc và các thông báo liên quan cho quản lý.

Hệ thống kết nối với ba kho dữ liệu chính:

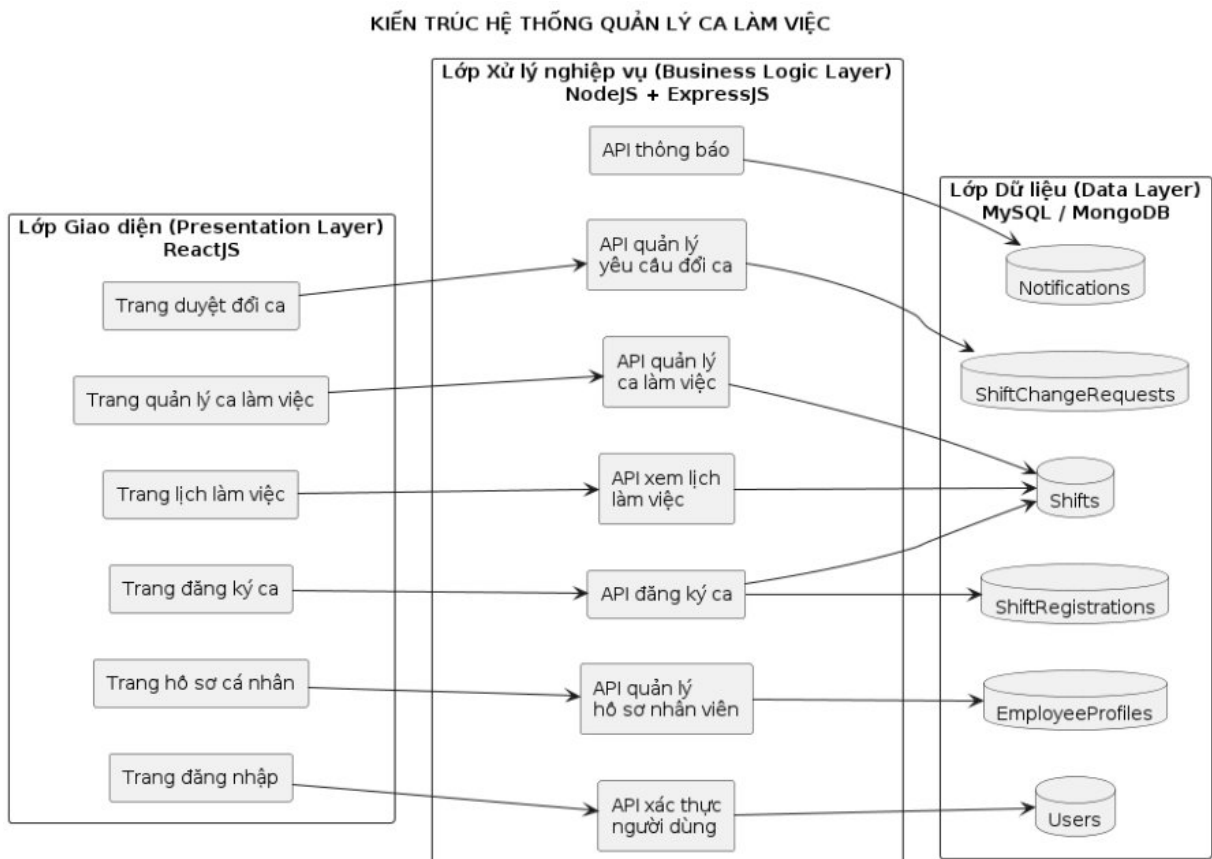
- D1 – *Hồ sơ nhân viên*: Lưu trữ thông tin cá nhân của nhân viên, phục vụ cho việc xem và cập nhật hồ sơ cá nhân.

- D2 – *Ca làm việc*: Lưu trữ thông tin về các ca làm việc, phục vụ cho việc tạo ca, cập nhật ca và tra cứu lịch làm việc của nhân viên.

- D3 – *Yêu cầu đăng ký/đổi ca*: Lưu trữ các yêu cầu đăng ký ca và đổi ca của nhân viên, phục vụ cho việc xét duyệt và cập nhật trạng thái yêu cầu.

2.3. Thiết kế hệ thống

2.3.1. Thiết kế kiến trúc tổng thể.



Hình 2.3: Hình ảnh thiết kế hệ thống tổng thể

Hệ thống quản lý ca làm việc được xây dựng theo mô hình kiến trúc ba lớp (Three-tier Architecture), bao gồm:

Lớp giao diện người dùng (Presentation Layer):

Được phát triển bằng ReactJS, đảm nhận vai trò hiển thị thông tin và tiếp nhận các thao tác từ người dùng là nhân viên và quản lý. Giao diện cung cấp các chức năng như đăng nhập, quản lý hồ sơ cá nhân, đăng ký ca làm việc, xem lịch làm việc, duyệt yêu cầu đổi ca và quản lý ca làm việc. Các yêu cầu từ người dùng được gửi đến hệ thống thông

qua các API RESTful. Giao diện được thiết kế trực quan, dễ sử dụng và hỗ trợ trên nhiều loại thiết bị.

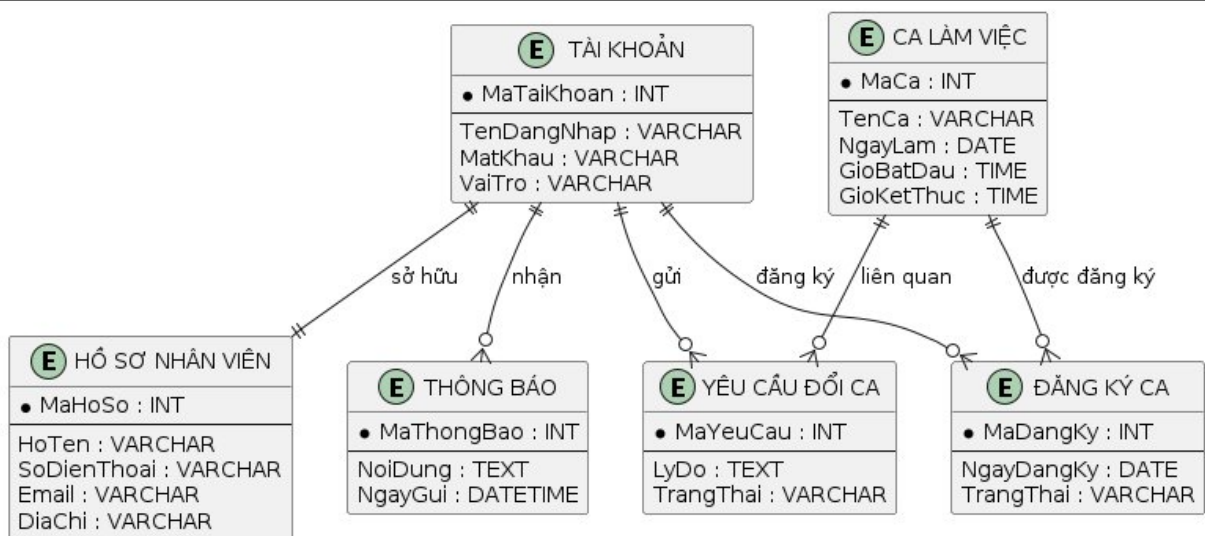
Lớp xử lý nghiệp vụ (Business Logic Layer):

Được xây dựng bằng Node.js và ExpressJS, chịu trách nhiệm xử lý toàn bộ nghiệp vụ của hệ thống. Lớp này cung cấp các API phục vụ cho việc xác thực người dùng, quản lý hồ sơ nhân viên, đăng ký ca làm việc, xem lịch làm việc, quản lý yêu cầu đổi ca, quản lý ca làm việc và gửi thông báo. Backend tiếp nhận yêu cầu từ giao diện, xử lý dữ liệu theo quy tắc nghiệp vụ và trả kết quả về cho người dùng.

Lớp cơ sở dữ liệu (Data Layer):

Sử dụng MongoDB để lưu trữ dữ liệu của hệ thống. Cơ sở dữ liệu bao gồm các bảng/thực thể chính như Users, EmployeeProfiles, Shifts, ShiftRegistrations, ShiftChangeRequests và Notifications. Dữ liệu được tổ chức hợp lý nhằm đảm bảo tính nhất quán, hỗ trợ truy vấn nhanh chóng và thuận tiện cho việc mở rộng hệ thống trong tương lai.

2.3.2. Thiết kế cơ sở dữ liệu (ERD, bảng dữ liệu)



Hình 2.4: Sơ đồ ERD hệ thống.

Mô tả sơ đồ ERD hệ thống quản lý ca làm việc

- *Tài khoản (TaiKhoan):* Lưu thông tin đăng nhập của người dùng trong hệ thống, bao gồm nhân viên và quản lý. Các thuộc tính chính gồm: ma_tai_khoan, ten_dang_nhap, mat_khau, vai_tro. Mỗi tài khoản tương ứng với một hồ sơ nhân viên và có thể thực hiện nhiều thao tác trong hệ thống.

- *Hồ sơ nhân viên (HoSoNhanVien)*: Lưu thông tin cá nhân của nhân viên, gồm: ma_ho_so, ho_ten, so_dien_thoai, email, dia_chi. Mỗi hồ sơ nhân viên thuộc về một tài khoản.
- *Ca làm việc (CaLamViec)*: Lưu thông tin các ca làm việc do quản lý tạo, gồm: ma_ca, ten_ca, ngay_lam, gio_bat_dau, gio_ket_thuc, trang_thai. Một ca làm việc có thể được nhiều nhân viên đăng ký hoặc liên quan đến nhiều yêu cầu đổi ca.
- *Đăng ký ca (DangKyCa)*: Lưu thông tin đăng ký ca làm việc của nhân viên, gồm: ma_dang_ky, ma_tai_khoan, ma_ca, ngay_dang_ky, trang_thai. Mỗi bản ghi thể hiện một lần đăng ký ca của nhân viên đối với một ca làm việc cụ thể.
- *Yêu cầu đổi ca (YeuCauDoiCa)*: Lưu các yêu cầu đổi ca của nhân viên, gồm: ma_yeu_cau, ma_tai_khoan, ma_ca, ly_do, trang_thai, ngay_gui. Mỗi yêu cầu đổi ca liên kết với một nhân viên và một ca làm việc.
- *Thông báo (ThongBao)*: Lưu các thông báo được gửi đến người dùng, gồm: ma_thong_bao, ma_tai_khoan, noi_dung, ngay_gui, trang_thai. Một tài khoản có thể nhận nhiều thông báo từ hệ thống.

Các mối quan hệ giữa các thực thể

- + Một tài khoản có một hồ sơ nhân viên (quan hệ 1-1).
- + Một tài khoản có thể đăng ký nhiều ca làm việc thông qua thực thể Đăng ký ca (quan hệ 1-N).
- + Một ca làm việc có thể được nhiều nhân viên đăng ký (quan hệ 1-N).
- + Một tài khoản có thể gửi nhiều yêu cầu đổi ca (quan hệ 1-N).
- + Một ca làm việc có thể liên kết với nhiều yêu cầu đổi ca (quan hệ 1-N).
- + Một tài khoản có thể nhận nhiều thông báo từ hệ thống (quan hệ 1-N).

Bảng 2.1: Bảng Người dùng (User)

Tên trường	Kiểu dữ liệu	Mô tả
user_id	String	Khóa chính
name	String	Tên người dùng
email	String	Email
password	String	Mật khẩu
phone	String	Số điện thoại
role	String	Vai trò (admin/user)

Mô tả bảng User (Người dùng):

- Lưu trữ thông tin tài khoản của tất cả người dùng hệ thống, bao gồm nhân viên và quản lý
- Các trường chính: user_id, name, email, password, phone, role.
- Dùng để xác thực đăng nhập, phân quyền và quản lý tài khoản.

2.3.3. Nội dung thiết kế giao diện người dùng (UI/UX)

Giao diện người dùng của hệ thống quản lý ca làm việc được thiết kế theo tiêu chí thân thiện, trực quan và dễ sử dụng đối với cả nhân viên và quản lý. Màu sắc, bố cục và các thành phần giao diện được sắp xếp hợp lý nhằm giúp người dùng thao tác nhanh chóng, thuận tiện và nâng cao hiệu quả quản lý ca làm việc.

- Trang đăng nhập:

Cho phép người dùng đăng nhập vào hệ thống bằng tài khoản được cấp. Hệ thống thực hiện xác thực thông tin và phân quyền truy cập theo vai trò nhân viên hoặc quản lý.

- Trang hồ sơ cá nhân:

Cho phép nhân viên xem và cập nhật các thông tin cá nhân như họ tên, số điện thoại, email và địa chỉ. Người dùng cũng có thể thay đổi mật khẩu tài khoản khi cần thiết.

- Trang đăng ký ca làm việc:

Hiển thị danh sách các ca làm việc hiện có để nhân viên lựa chọn và đăng ký. Hệ thống hỗ trợ theo dõi trạng thái đăng ký ca và hiển thị các thông báo liên quan.

- Trang lịch làm việc:

Cho phép nhân viên xem lịch làm việc theo ngày, tuần hoặc tháng. Giao diện hiển thị trực quan các ca làm việc đã được phân công hoặc đăng ký thành công.

- *Trang yêu cầu đổi ca:*

Cho phép nhân viên gửi yêu cầu đổi ca làm việc và theo dõi trạng thái xử lý yêu cầu. Quản lý có thể xem danh sách yêu cầu đổi ca và thực hiện phê duyệt hoặc từ chối trực tiếp trên giao diện.

- *Trang quản lý ca làm việc:*

Dành cho quản lý, hỗ trợ tạo mới, chỉnh sửa và cập nhật thông tin các ca làm việc. Quản lý có thể phân công ca làm việc cho nhân viên và theo dõi tình trạng các ca trong hệ thống.

- *Trang quản trị hệ thống:*

Dành cho quản lý, cung cấp các chức năng quản lý nhân viên, quản lý ca làm việc, theo dõi các yêu cầu đổi ca và xem báo cáo thống kê liên quan đến hoạt động làm việc của nhân viên.

2.4. Sơ đồ hoạt động và các sơ đồ minh họa

2.4.1. Sơ đồ Use Case.

- *Sơ đồ Use Case tổng quát của nhân viên:*



Hình 2.5: Sơ đồ Use Case tổng quát của nhân viên

Mô tả sơ đồ Use Case tổng quát của nhân viên:

- + Đăng ký tài khoản, đăng nhập và đăng xuất hệ thống.
- + Xem và cập nhật thông tin cá nhân.
- + Xem lịch làm việc được phân công theo ngày hoặc tuần.
- + Đăng ký nhận ca làm việc còn trống.
- + Gửi yêu cầu đổi ca hoặc nhường ca làm việc khi có nhu cầu.
- + Theo dõi trạng thái xử lý các yêu cầu đổi ca.
- + Xem lịch sử đổi ca đã thực hiện.
- + Xem lịch sử làm việc, tổng thời gian làm việc và các thông tin liên quan đến quá trình làm việc của bản thân.

- Sơ đồ Use Case người quản lý (Admin):



Hình 2.6: Sơ đồ Use Case quản lý (Admin)

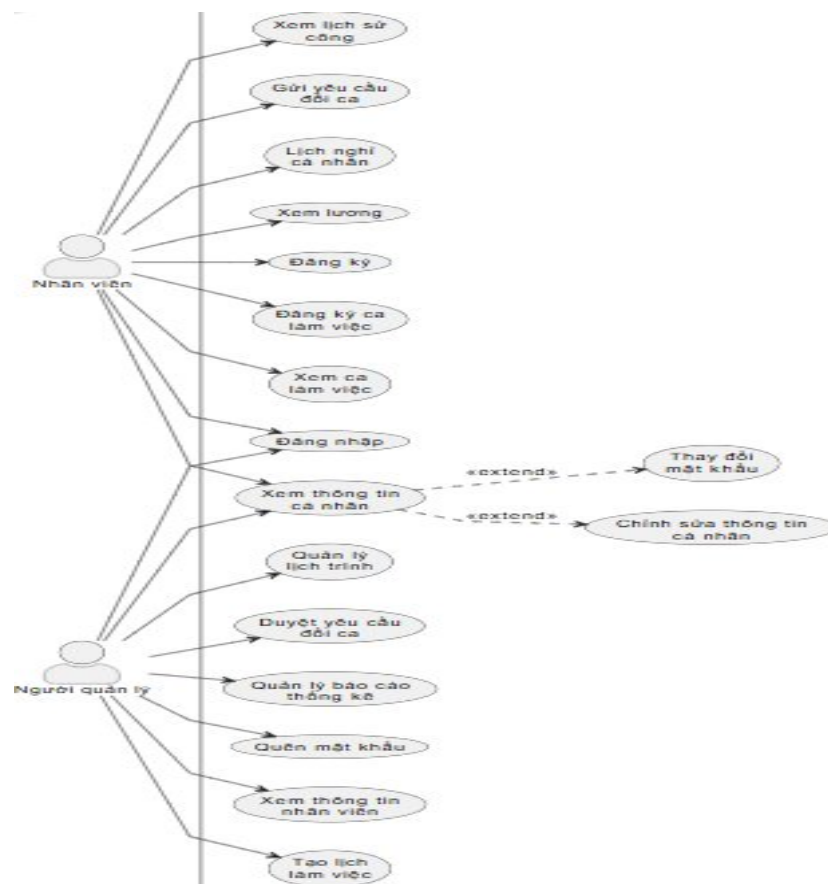
Mô tả:

- + Đăng nhập và đăng xuất hệ thống.
- + Quản lý thông tin cá nhân, chỉnh sửa thông tin và thay đổi mật khẩu.

- + Xem danh sách và thông tin chi tiết của nhân viên.
- + Duyệt hoặc từ chối tài khoản đăng ký của nhân viên.
- + Quản lý lịch trình và ca làm việc của nhân viên.
- + Duyệt hoặc từ chối các yêu cầu đổi ca của nhân viên.
- + Theo dõi trạng thái làm việc của nhân viên thông qua chức năng check-in/check-out.

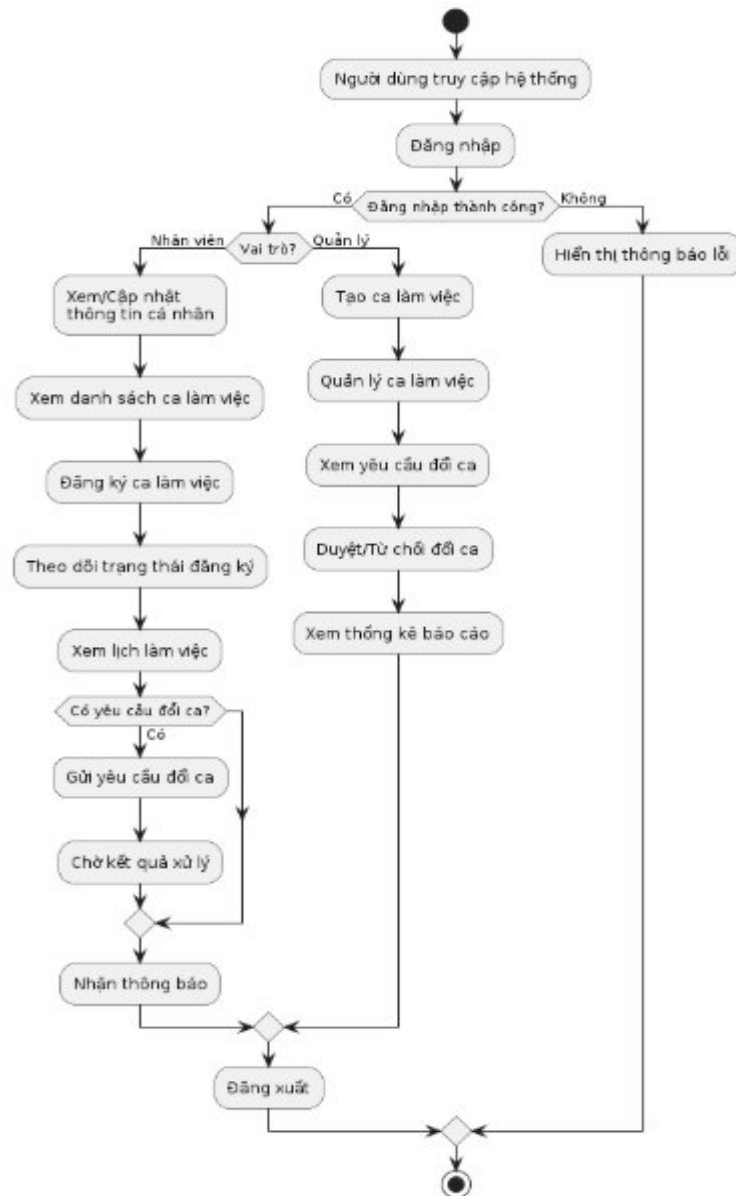
- + Xem báo cáo thống kê số lượng nhân viên.

- Sơ đồ Use Case tổng quát:



Hình 2.7: Sơ đồ usecase tổng quát

2.4.2. Sơ đồ hoạt động của các chức năng chính



Hình 2.8: Sơ đồ hoạt động các chức năng chính (Activity Diagram)

Mô tả sơ đồ:

- *Người dùng truy cập hệ thống*

+ Bắt đầu quy trình khi người dùng truy cập vào hệ thống quản lý ca làm việc.

- *Đăng nhập bằng tài khoản*

+ Người dùng thực hiện đăng nhập bằng tài khoản được cấp.

+ Nếu đăng nhập thất bại, hệ thống hiển thị thông báo lỗi và yêu cầu đăng nhập lại.

+ Nếu đăng nhập thành công, người dùng tiếp tục sử dụng các chức năng của hệ thống theo quyền được phân công.

- Nhân viên sử dụng hệ thống

- + Nhân viên có thể xem thông tin cá nhân.
- + Nhân viên có thể cập nhật thông tin cá nhân khi cần thiết.
- + Nhân viên xem danh sách các ca làm việc hiện có.
- + Nhân viên lựa chọn và gửi yêu cầu đăng ký ca làm việc.
- + Hệ thống tiếp nhận yêu cầu và cập nhật trạng thái đăng ký ca.
- + Nhân viên có thể xem lịch làm việc theo ngày, tuần hoặc tháng.

- Yêu cầu đổi ca

- + Khi có nhu cầu thay đổi lịch làm việc, nhân viên gửi yêu cầu đổi ca.
- + Hệ thống ghi nhận yêu cầu và chuyển đến quản lý để xem xét xử lý.

- Quản lý xử lý yêu cầu đổi ca

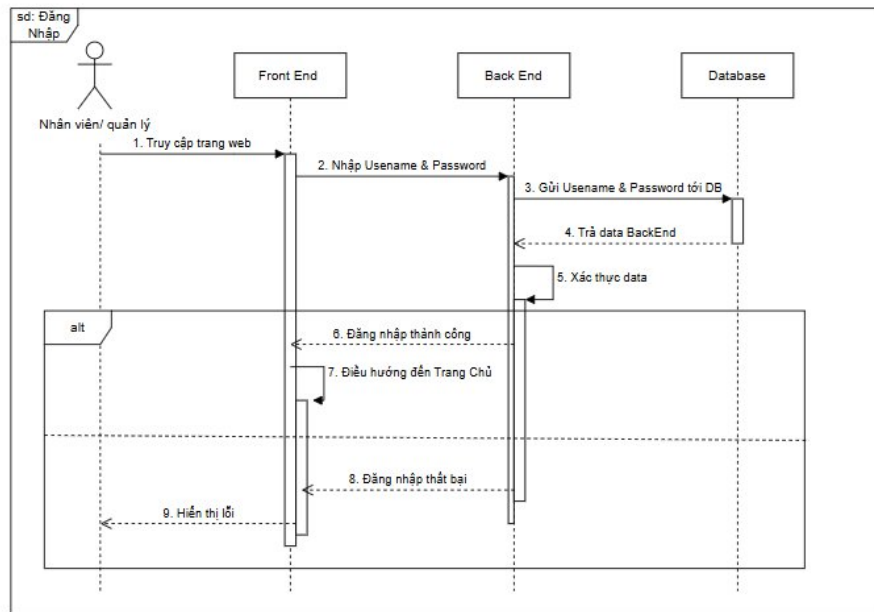
- + Quản lý đăng nhập vào hệ thống.
- + Quản lý xem danh sách các yêu cầu đổi ca của nhân viên.
- + Hệ thống cập nhật trạng thái yêu cầu và gửi thông báo kết quả cho nhân viên.

- Quản lý ca làm việc

- + Quản lý tạo mới ca làm việc.
- + Quản lý cập nhật hoặc chỉnh sửa thông tin ca làm việc khi cần thiết.
- + Hệ thống lưu trữ và cập nhật dữ liệu ca làm việc.

2.4.3. Sơ đồ trình tự (Sequence Diagram)

+ Sơ đồ trình tự đăng nhập:

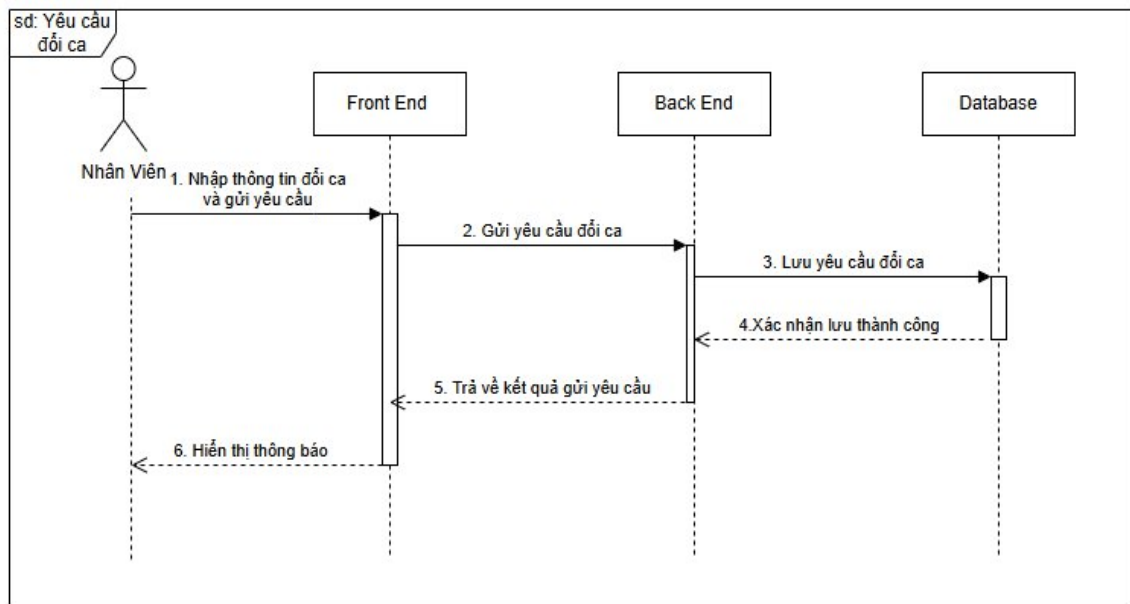


Hình 2.9: Sơ đồ trình tự đăng nhập

Mô tả quy trình sơ đồ trình tự đăng nhập:

- Nhân viên hoặc quản lý truy cập vào trang đăng nhập của hệ thống thông qua giao diện người dùng (Frontend).
- Người dùng nhập tên đăng nhập và mật khẩu, sau đó gửi yêu cầu đăng nhập đến hệ thống.
- Giao diện (Frontend) tiếp nhận thông tin đăng nhập và gửi dữ liệu tài khoản đến Server (Backend) để xử lý.
- Server (Backend) nhận thông tin đăng nhập và thực hiện truy vấn cơ sở dữ liệu (MongoDB) để tìm kiếm tài khoản tương ứng.
- Database (MongoDB) trả về thông tin tài khoản cho Server (Backend).
- Server (Backend) tiến hành xác thực tên đăng nhập, mật khẩu và kiểm tra trạng thái tài khoản.
- Nếu thông tin đăng nhập không hợp lệ hoặc tài khoản chưa được kích hoạt, Server (Backend) trả về thông báo lỗi.
- Giao diện (Frontend) hiển thị thông báo lỗi để người dùng kiểm tra và thực hiện đăng nhập lại.

+ Sơ đồ trình tự yêu cầu đổi ca:



Hình 2.10: Sơ đồ trình tự yêu cầu đổi ca (nhân viên)

Mô tả quy trình sơ đồ trình tự yêu cầu đổi ca:

Nhân viên truy cập chức năng theo dõi trạng thái yêu cầu đổi ca trên hệ thống.

Giao diện người dùng (Frontend) tiếp nhận yêu cầu xem trạng thái và gửi yêu cầu đến Server (Backend).

Server (Backend) nhận yêu cầu và thực hiện truy vấn cơ sở dữ liệu (MongoDB) để lấy danh sách các yêu cầu đổi ca của nhân viên.

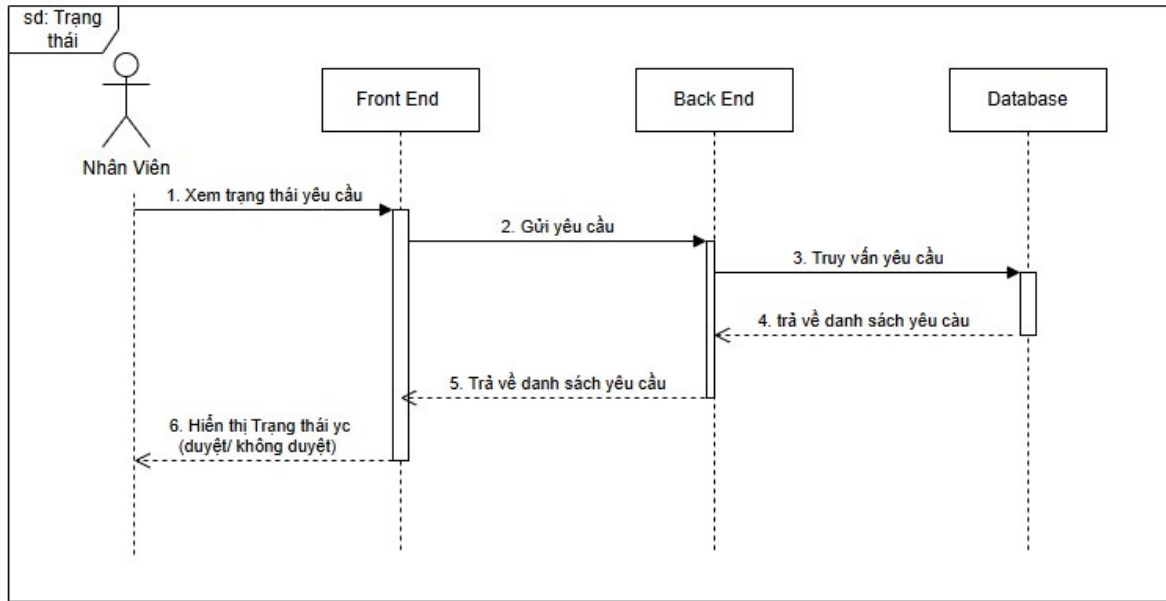
Database (MongoDB) trả về dữ liệu danh sách yêu cầu đổi ca cùng trạng thái xử lý cho Server (Backend).

Server (Backend) xử lý dữ liệu và gửi kết quả về cho Giao diện người dùng (Frontend).

Giao diện (Frontend) hiển thị danh sách yêu cầu đổi ca cùng trạng thái tương ứng như: Chờ duyệt, Đã duyệt hoặc Từ chối để nhân viên theo dõi.

Nhân viên có thể xem chi tiết từng yêu cầu, bao gồm thông tin ca làm việc, lý do đổi ca, thời gian gửi yêu cầu và kết quả xử lý từ người quản lý.

+ Sơ đồ trình tự trạng thái đổi ca.

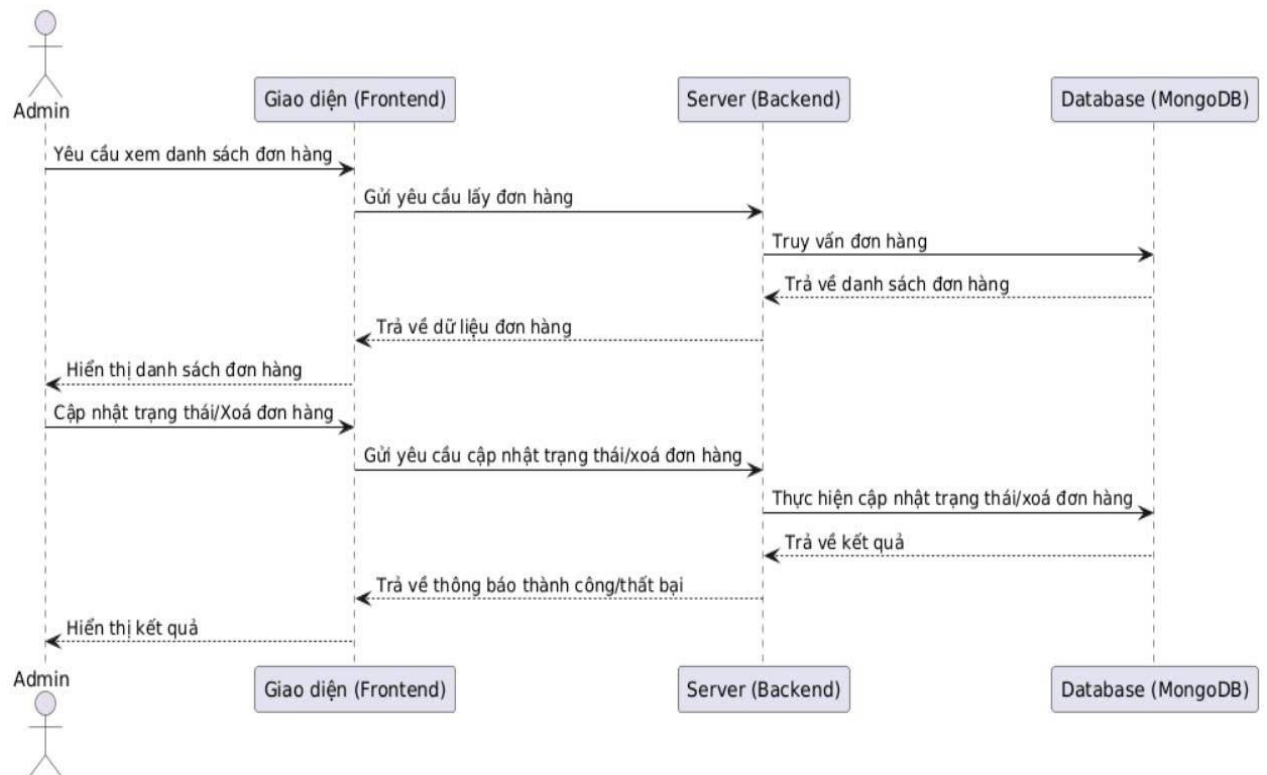


Hình 2.11: Sơ đồ trình tự trạng thái đổi ca (nhân viên)

Mô tả quy trình sơ đồ trình tự trạng thái đổi ca:

- Admin gửi yêu cầu xem danh sách người dùng trên giao diện người dùng (Frontend).
- Giao diện (Frontend) tiếp nhận yêu cầu và gửi yêu cầu lấy danh sách người dùng đến Server (Backend).
- Server (Backend) truy vấn dữ liệu người dùng từ Database (MongoDB) và nhận về danh sách người dùng.
- Server (Backend) trả về dữ liệu người dùng cho Giao diện (Frontend) để hiển thị cho admin.
- Khi admin thực hiện thao tác cấp quyền cho người dùng, Giao diện (Frontend) gửi yêu cầu cập nhật quyền đến Server (Backend).
- Server (Backend) thực hiện cập nhật quyền người dùng trên Database (MongoDB) và nhận kết quả trả về.
- Server (Backend) trả về thông báo thành công hoặc thất bại cho Giao diện (Frontend), sau đó giao diện hiển thị kết quả cho admin.

+ Sơ đồ trình tự xem thông tin cá nhân:



Hình 2.12: Sơ đồ trình tự xem thông tin cá nhân

Mô tả quy trình sơ đồ trình tự xem thông tin cá nhân:

- Nhân viên truy cập chức năng xem thông tin cá nhân trên hệ thống.

Giao diện người dùng (Frontend) tiếp nhận yêu cầu và gửi yêu cầu lấy thông tin cá nhân đến Server (Backend).

Server (Backend) nhận yêu cầu và thực hiện truy vấn cơ sở dữ liệu (MongoDB) để lấy thông tin của nhân viên.

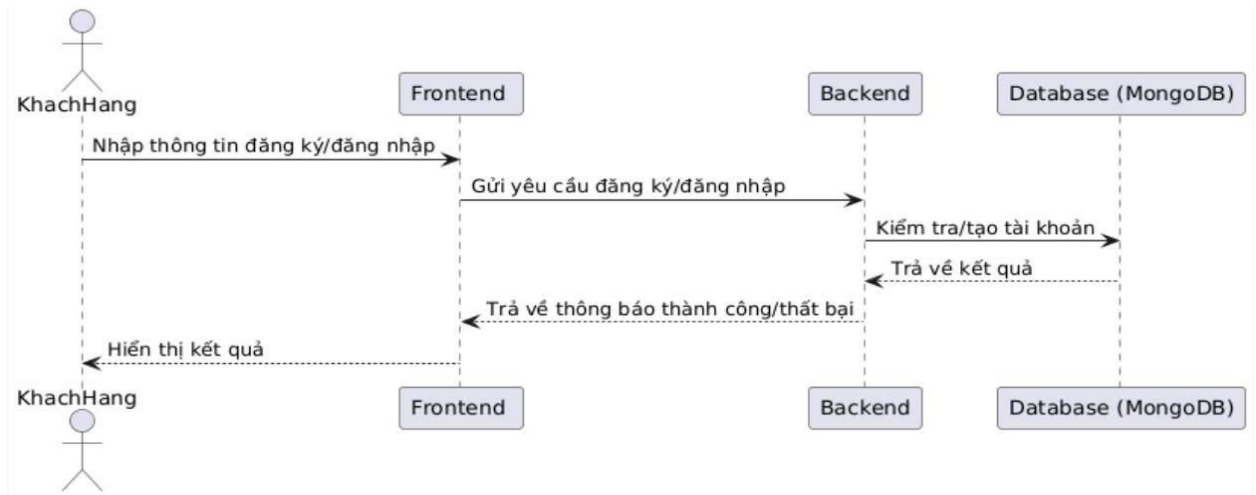
Database (MongoDB) trả về dữ liệu thông tin cá nhân cho Server (Backend).

Server (Backend) xử lý dữ liệu và gửi kết quả về cho Giao diện người dùng (Frontend).

Giao diện (Frontend) hiển thị đầy đủ thông tin cá nhân của nhân viên như họ tên, mã nhân viên, email, số điện thoại, chức vụ và các thông tin liên quan khác.

Nhân viên có thể xem lại thông tin cá nhân của mình để kiểm tra tính chính xác hoặc thực hiện cập nhật khi cần thiết.

+ Sơ đồ trình tự sửa thông tin cá nhân:

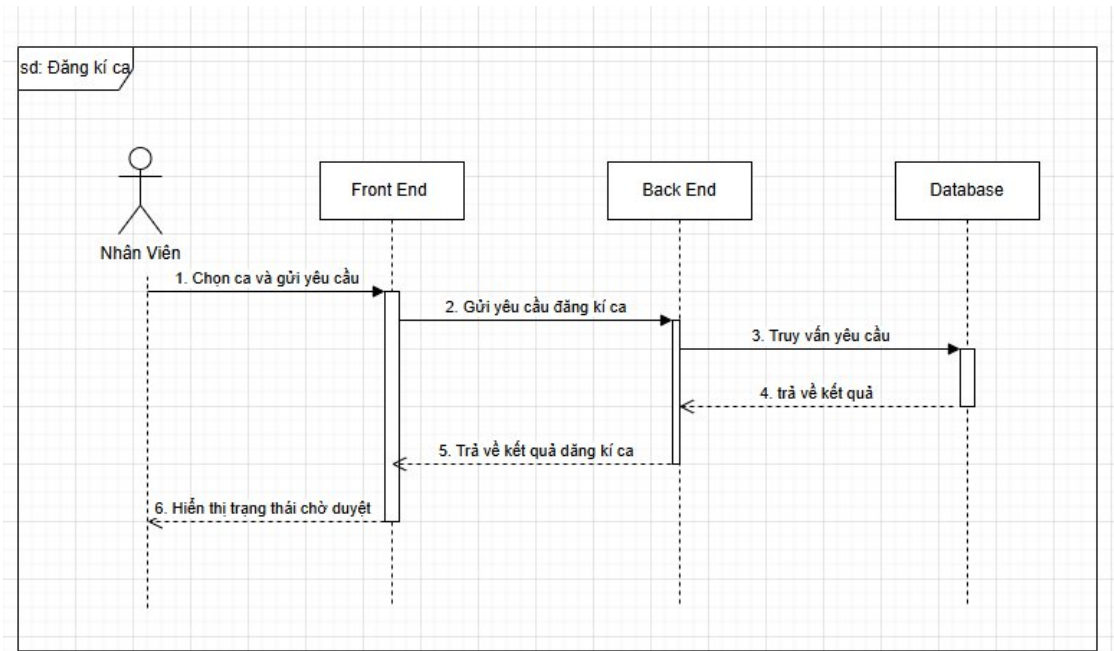


Hình 2.13: Sơ đồ trình tự sửa thông tin cá nhân

Mô tả quy trình sơ đồ trình tự sửa thông tin cá nhân:

- Nhân viên truy cập chức năng chỉnh sửa thông tin cá nhân trên hệ thống.
- Nhân viên cập nhật các thông tin cần thay đổi như họ tên, số điện thoại, email hoặc các thông tin cá nhân khác và gửi yêu cầu cập nhật.
- Giao diện người dùng (Frontend) tiếp nhận dữ liệu cập nhật và gửi yêu cầu đến Server (Backend).
- Server (Backend) nhận yêu cầu, kiểm tra tính hợp lệ của dữ liệu và thực hiện cập nhật thông tin vào cơ sở dữ liệu (MongoDB).
- Database (MongoDB) lưu các thông tin mới và trả kết quả cập nhật về cho Server (Backend).
- Server (Backend) xử lý kết quả và gửi phản hồi thành công cho Giao diện người dùng (Frontend).
- Giao diện (Frontend) hiển thị thông báo cập nhật thành công cho nhân viên.
- Sau khi cập nhật, các thông tin cá nhân mới của nhân viên sẽ được lưu trong hệ thống và được sử dụng cho các chức năng liên quan.

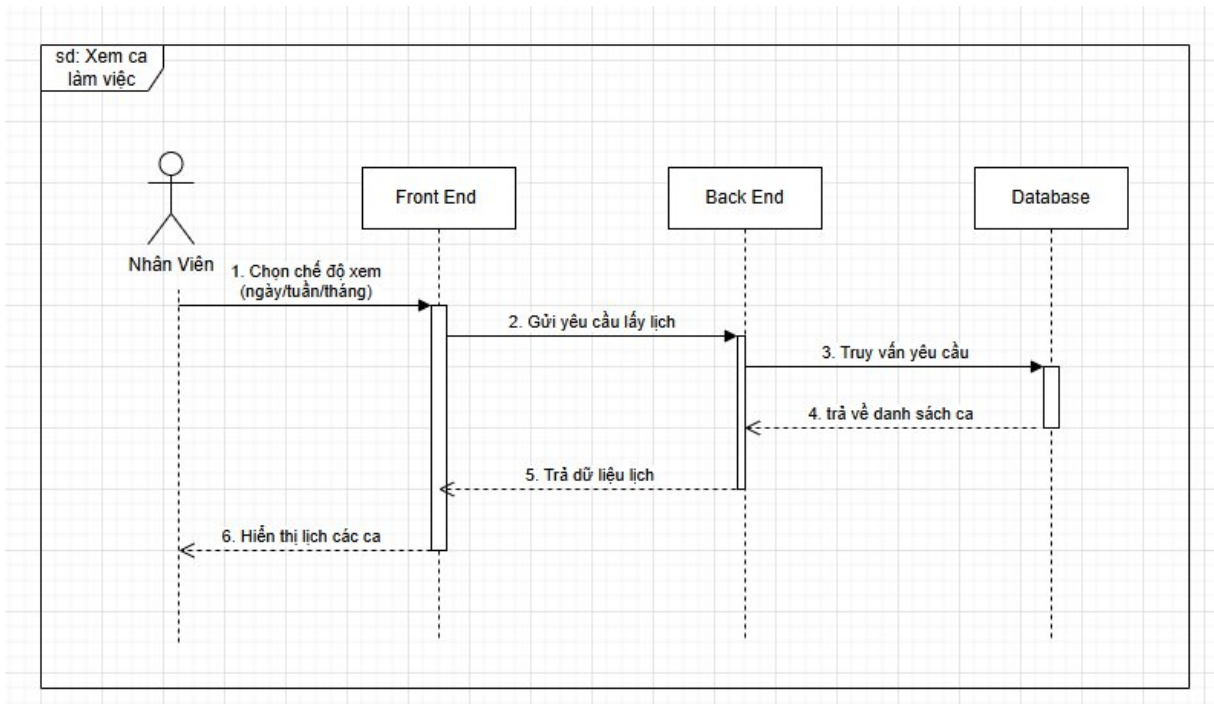
+ Sơ đồ trình tự đăng kí ca làm việc:



Hình 2.14: Sơ đồ trình tự đăng kí ca làm việc

Mô tả quy trình sơ đồ trình tự đăng kí ca làm việc:

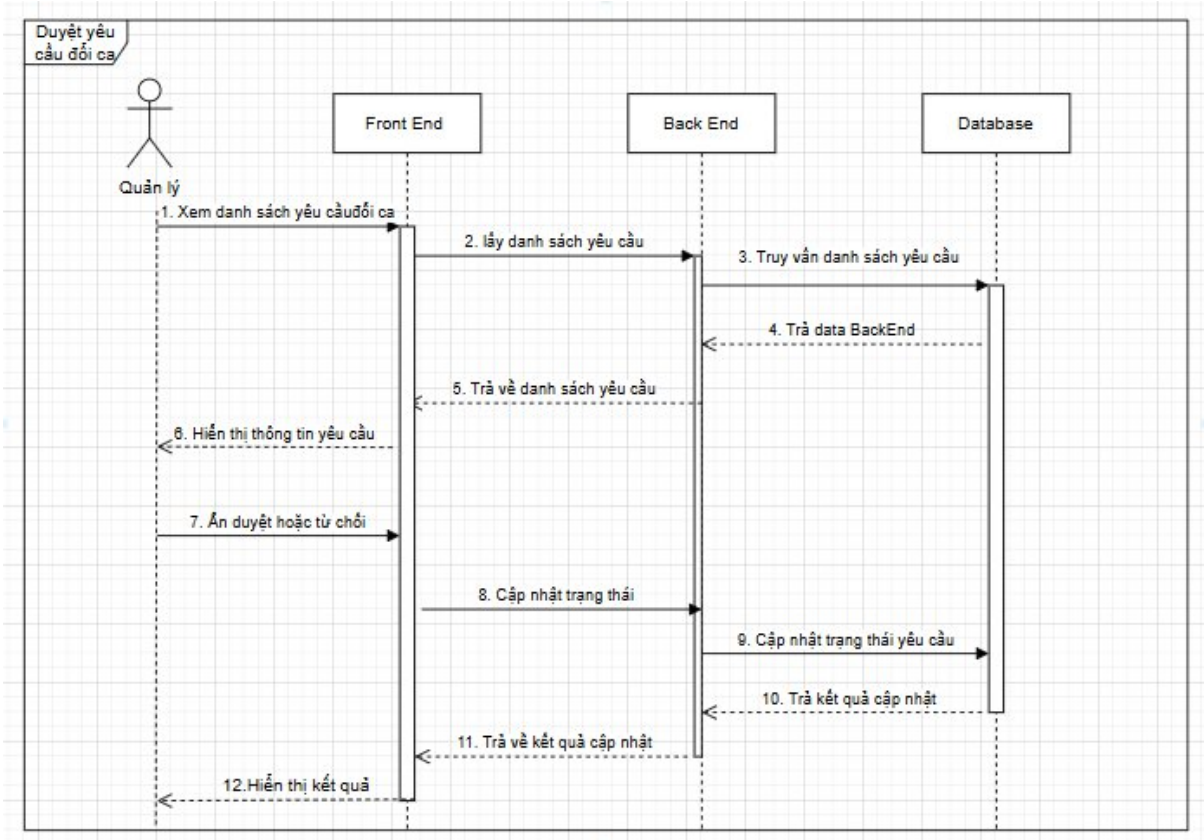
- Nhân viên truy cập chức năng đăng ký ca làm việc trên hệ thống.
 - Nhân viên lựa chọn ca làm việc phù hợp và gửi yêu cầu đăng ký ca.
 - Giao diện người dùng (Frontend) tiếp nhận yêu cầu đăng ký ca và gửi dữ liệu đến Server (Backend).
 - Server (Backend) nhận yêu cầu, kiểm tra tính hợp lệ của ca làm việc và thực hiện lưu thông tin đăng ký vào cơ sở dữ liệu (MongoDB).
 - Database (MongoDB) xử lý yêu cầu, lưu dữ liệu đăng ký ca và trả kết quả về cho Server (Backend).
 - Server (Backend) nhận kết quả từ cơ sở dữ liệu và gửi phản hồi về Giao diện người dùng (Frontend).
 - Giao diện (Frontend) hiển thị trạng thái đăng ký ca thành công và thông báo yêu cầu đang chờ quản lý xét duyệt.
 - Sau khi quản lý xét duyệt, trạng thái ca làm việc sẽ được cập nhật và nhân viên có thể theo dõi kết quả trong hệ thống.
- + *Sơ đồ trình tự xem ca làm việc:*



Hình 2.15: Sơ đồ trình tự xem ca làm việc

Mô tả quy trình sơ đồ trình tự xem ca làm việc:

- Nhân viên truy cập chức năng xem lịch làm việc trên hệ thống.
 - Nhân viên lựa chọn chế độ xem lịch theo ngày, tuần hoặc tháng tùy theo nhu cầu sử dụng.
 - Giao diện người dùng (Frontend) tiếp nhận yêu cầu và gửi yêu cầu lấy dữ liệu lịch làm việc đến Server (Backend).
 - Server (Backend) nhận yêu cầu và thực hiện truy vấn cơ sở dữ liệu (MongoDB) để lấy danh sách các ca làm việc của nhân viên.
 - Database (MongoDB) trả về dữ liệu lịch làm việc cho Server (Backend).
 - Server (Backend) xử lý dữ liệu và gửi kết quả về cho Giao diện người dùng (Frontend).
 - Giao diện (Frontend) hiển thị danh sách các ca làm việc theo chế độ xem đã được nhân viên lựa chọn.
 - Nhân viên có thể theo dõi thông tin chi tiết của từng ca làm việc như ngày làm việc, thời gian bắt đầu, thời gian kết thúc, vị trí làm việc và trạng thái ca làm.
- + Sơ đồ trình tự duyệt yêu cầu đổi ca



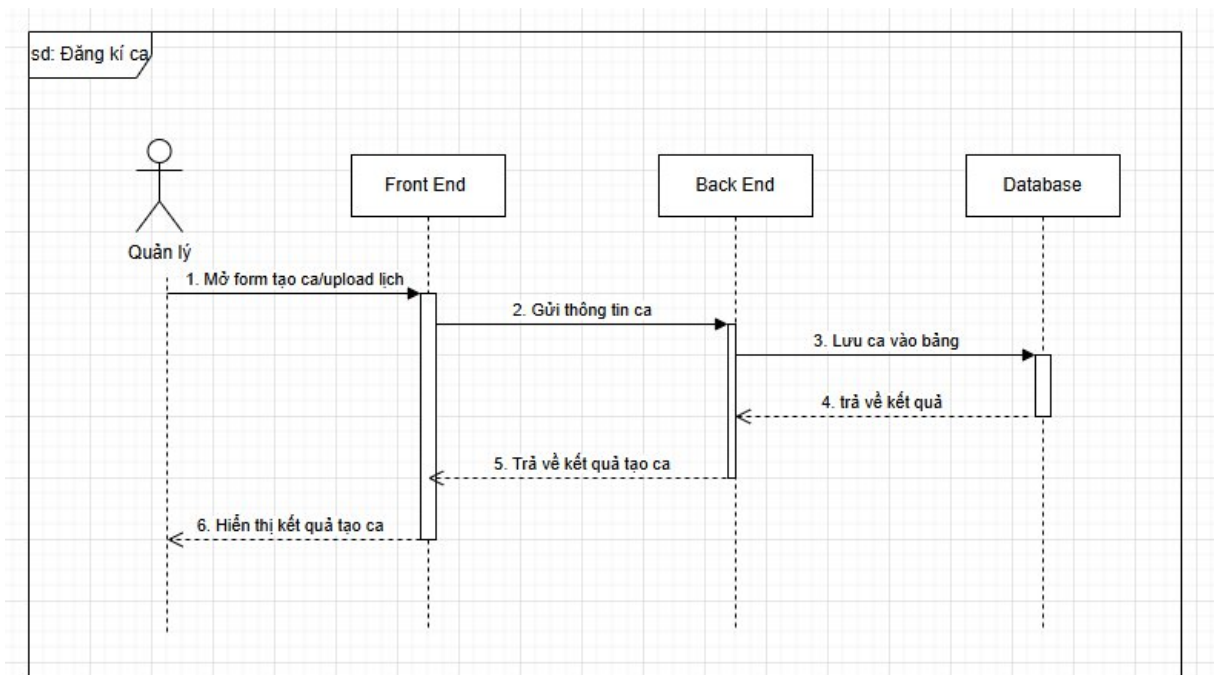
Hình 2.16: Sơ đồ trình tự duyệt yêu cầu đổi ca

Mô tả quy trình sơ đồ trình tự duyệt yêu cầu đổi ca:

- Quản lý truy cập chức năng quản lý yêu cầu đổi ca trên hệ thống.
- Quản lý yêu cầu xem danh sách các yêu cầu đổi ca đang chờ xử lý.
- Giao diện người dùng (Frontend) gửi yêu cầu lấy danh sách yêu cầu đổi ca đến Server (Backend).
- Server (Backend) thực hiện truy vấn cơ sở dữ liệu (MongoDB) để lấy danh sách các yêu cầu đổi ca.
- Database (MongoDB) trả về dữ liệu danh sách yêu cầu đổi ca cho Server (Backend).
- Server (Backend) xử lý dữ liệu và gửi kết quả về cho Giao diện người dùng (Frontend).
- Giao diện (Frontend) hiển thị danh sách các yêu cầu đổi ca cùng các thông tin liên quan như nhân viên gửi yêu cầu, ca hiện tại, ca muốn đổi, lý do đổi ca và trạng thái xử lý.

- Sau khi xem xét, quản lý thực hiện thao tác phê duyệt hoặc từ chối yêu cầu đổi ca.
- Giao diện (Frontend) gửi yêu cầu cập nhật trạng thái đến Server (Backend).
- Server (Backend) xử lý yêu cầu và cập nhật trạng thái yêu cầu đổi ca trong cơ sở dữ liệu (MongoDB).
- Database (MongoDB) lưu kết quả cập nhật và trả phản hồi cho Server (Backend).
- Server (Backend) gửi kết quả xử lý về cho Giao diện người dùng (Frontend).
- Giao diện (Frontend) hiển thị thông báo kết quả xử lý thành công và cập nhật trạng thái mới của yêu cầu đổi ca.
- Trong trường hợp yêu cầu được phê duyệt, hệ thống sẽ cập nhật thông tin ca làm việc tương ứng theo quy định của hệ thống.

+ Sơ đồ trình tự tạo lịch ca làm việc:



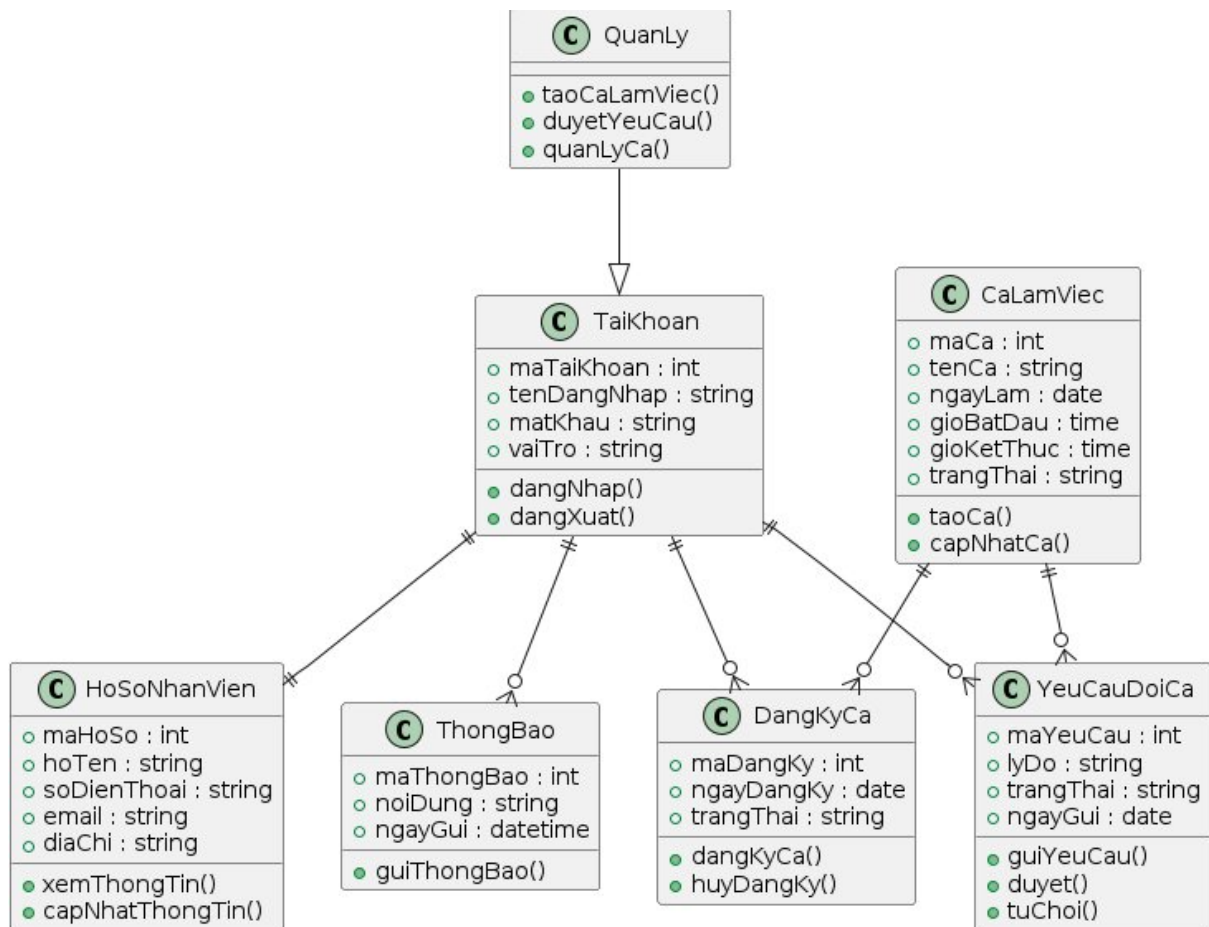
Hình 2.17: Sơ đồ trình tự tạo lịch ca làm việc

Mô tả quy trình sơ đồ trình tự tạo lịch ca làm việc:

- Quản lý truy cập chức năng tạo ca làm việc hoặc tải lên lịch làm việc trên hệ thống.
- Quản lý nhập các thông tin cần thiết của ca làm việc như ngày làm việc, thời gian bắt đầu, thời gian kết thúc, vị trí công việc và nhân viên được phân công.

- Giao diện người dùng (Frontend) tiếp nhận dữ liệu và gửi thông tin ca làm việc đến Server (Backend).
- Server (Backend) nhận yêu cầu, kiểm tra tính hợp lệ của dữ liệu và thực hiện lưu thông tin ca làm việc vào cơ sở dữ liệu (MongoDB).
- Database (MongoDB) lưu dữ liệu ca làm việc và trả kết quả xử lý cho Server (Backend).
- Server (Backend) nhận phản hồi từ cơ sở dữ liệu và gửi kết quả tạo ca làm việc về cho Giao diện người dùng (Frontend).
- Giao diện (Frontend) hiển thị thông báo tạo ca làm việc thành công cho quản lý.
- Sau khi được tạo thành công, ca làm việc sẽ được cập nhật vào hệ thống và hiển thị trên lịch làm việc để nhân viên có thể theo dõi hoặc đăng ký theo quy định của hệ thống.

2.4.4. Sơ đồ lớp (Class Diagram)



Hình 2.18: Sơ đồ Lớp Class Diagram tổng quát

Mô tả sơ đồ Class Diagram tổng quát:

- *Lớp TàiKhoản (TaiKhoan)*: Đại diện cho tài khoản người dùng trong hệ thống, bao gồm nhân viên và quản lý. Lớp chứa các thuộc tính như maTaiKhoan, tenDangNhap, matKhau, vaiTro và các phương thức đăng nhập, đăng xuất.

- *Lớp HồSơNhânViên (HoSoNhanVien)*: Lưu trữ thông tin cá nhân của nhân viên, gồm các thuộc tính như maHoSo, hoTen, soDienThoai, email, diaChi. Lớp cung cấp các chức năng xem và cập nhật thông tin cá nhân.

Lớp QuảnLý (QuanLy): Là lớp kế thừa từ lớp TàiKhoản, đại diện cho người quản lý hệ thống. Lớp này có các chức năng tạo ca làm việc, quản lý ca làm việc và duyệt hoặc từ chối các yêu cầu đổi ca của nhân viên.

- *Lớp CaLàmViệc (CaLamViec)*: Lưu thông tin đăng ký ca làm việc của nhân viên, gồm các thuộc tính maDangKy, ngayDangKy, trangThai. Lớp hỗ trợ các chức năng đăng ký và hủy đăng ký ca làm việc.

- *Lớp YêuCầuĐổiCa (YeuCauDoiCa)*: Lưu trữ các yêu cầu đổi ca do nhân viên gửi, bao gồm maYeuCau, lyDo, trangThai, ngayGui. Lớp cung cấp các chức năng gửi yêu cầu đổi ca, duyệt hoặc từ chối yêu cầu.

2.5. Tổng kết chương

Qua chương 2, em đã tiến hành phân tích và thiết kế hệ thống website quản lý nhân viên và ca làm việc cho chuỗi cửa hàng. Trong quá trình thực hiện, em đã xác định rõ các yêu cầu chức năng, yêu cầu phi chức năng, các ràng buộc và phạm vi hoạt động của hệ thống nhằm đáp ứng nhu cầu quản lý nhân viên và lịch làm việc một cách hiệu quả.

Các mô hình phân tích và thiết kế như sơ đồ ngữ cảnh, sơ đồ luồng dữ liệu (DFD), sơ đồ chức năng (BFD), sơ đồ thực thể liên kết (ERD), sơ đồ hoạt động (Activity Diagram), sơ đồ trình tự (Sequence Diagram) và sơ đồ lớp (Class Diagram) đã giúp mô tả rõ cấu trúc hệ thống, luồng xử lý dữ liệu, mối quan hệ giữa các thành phần cũng như các chức năng nghiệp vụ của hệ thống.

Bên cạnh đó, kiến trúc hệ thống được thiết kế theo mô hình ba lớp (Three-tier Architecture) với ReactJS đảm nhiệm phần giao diện người dùng, Node.js và ExpressJS đảm nhiệm xử lý nghiệp vụ phía máy chủ, cùng MongoDB đảm nhiệm lưu trữ và quản

lý dữ liệu. Cách tiếp cận này giúp hệ thống có tính linh hoạt, dễ bảo trì, dễ mở rộng và đáp ứng tốt nhu cầu phát triển trong tương lai.

Những nội dung phân tích và thiết kế được trình bày trong chương này là cơ sở quan trọng cho quá trình xây dựng và triển khai hệ thống ở các chương tiếp theo. Đồng thời, các mô hình thiết kế và kiến trúc hệ thống cũng góp phần đảm bảo việc phát triển website quản lý nhân viên và ca làm việc được thực hiện một cách khoa học, hiệu quả và phù hợp với yêu cầu thực tế của chuỗi cửa hàng.

Chương 3: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG

3.1. Tổng quan về xây dựng và triển khai hệ thống

Sau khi hoàn thành các bước phân tích và thiết kế, em tiến hành xây dựng và triển khai hệ thống quản lý nhân viên và ca làm việc cho chuỗi cửa hàng dựa trên các yêu cầu đã đề ra. Quá trình này bao gồm việc hiện thực hóa các chức năng quản lý nhân viên, quản lý lịch làm việc, đăng ký ca làm việc, yêu cầu đổi ca, chấm công và thống kê báo cáo.

Trong quá trình phát triển, em đặc biệt chú trọng đến tính ổn định, bảo mật và khả năng sử dụng của hệ thống. Các công nghệ như ReactJS, Node.js và MongoDB được lựa chọn nhằm đáp ứng tốt yêu cầu về hiệu năng, khả năng mở rộng và quản lý dữ liệu tập trung. Giao diện hệ thống được thiết kế thân thiện, trực quan giúp người dùng dễ dàng thao tác và sử dụng.

Việc triển khai hệ thống được thực hiện theo từng giai đoạn bao gồm: thiết kế và xây dựng cơ sở dữ liệu, phát triển các chức năng phía máy chủ (Backend), xây dựng giao diện người dùng (Frontend), tích hợp các chức năng quản lý nhân viên và ca làm việc, kiểm thử hệ thống và hoàn thiện.

Kết quả triển khai là một hệ thống quản lý nhân viên hoàn chỉnh, hỗ trợ quản lý thông tin nhân viên, quản lý lịch làm việc, đăng ký và duyệt ca làm việc, xử lý yêu cầu đổi ca, theo dõi check-in/check-out và cung cấp các báo cáo thống kê phục vụ công tác quản lý nhân viên và ca làm việc. Hệ thống góp phần nâng cao hiệu quả quản lý, giảm thiểu sai sót trong quá trình phân công công việc và hỗ trợ quá trình chuyển đổi số trong hoạt động quản lý nhân viên và ca làm việc của chuỗi cửa hàng.

3.2. Môi trường và công cụ phát triển.

3.2.1. Ngôn ngữ lập trình và Framework

- Phần backend của hệ thống được phát triển bằng Node.js kết hợp với Express. Node.js là nền tảng mạnh mẽ, hỗ trợ xây dựng các ứng dụng web hiệu năng cao, dễ mở rộng. Express giúp tổ chức các API RESTful một cách rõ ràng, thuận tiện cho việc phát triển và bảo trì.

- Phần frontend sử dụng ReactJS, một thư viện JavaScript phổ biến cho việc xây dựng giao diện người dùng động, thân thiện và dễ dàng quản lý trạng thái ứng dụng. ReactJS giúp tối ưu hóa trải nghiệm người dùng và tăng tốc độ phát triển giao diện.

3.2.2. Cơ sở dữ liệu

- MongoDB được lựa chọn làm hệ quản trị cơ sở dữ liệu. Đây là hệ quản trị cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng tài liệu (document), phù hợp với các ứng dụng web hiện đại nhờ khả năng mở rộng linh hoạt, truy vấn nhanh và dễ dàng tích hợp với Node.js.

3.2.3. Công cụ hỗ trợ phát triển

- Visual Studio Code (VS Code) là môi trường lập trình chính, cung cấp nhiều tiện ích mở rộng hỗ trợ lập trình JavaScript, Node.js, ReactJS, kiểm tra lỗi, tự động hoàn thành mã nguồn và quản lý dự án hiệu quả.

- Git được sử dụng để quản lý mã nguồn, giúp kiểm soát phiên bản, phối hợp làm việc nhóm và đảm bảo an toàn cho quá trình phát triển phần mềm.

3.2.4. Thư viện dùng để tích hợp

- Axios được sử dụng để giao tiếp giữa frontend và backend thông qua các API.

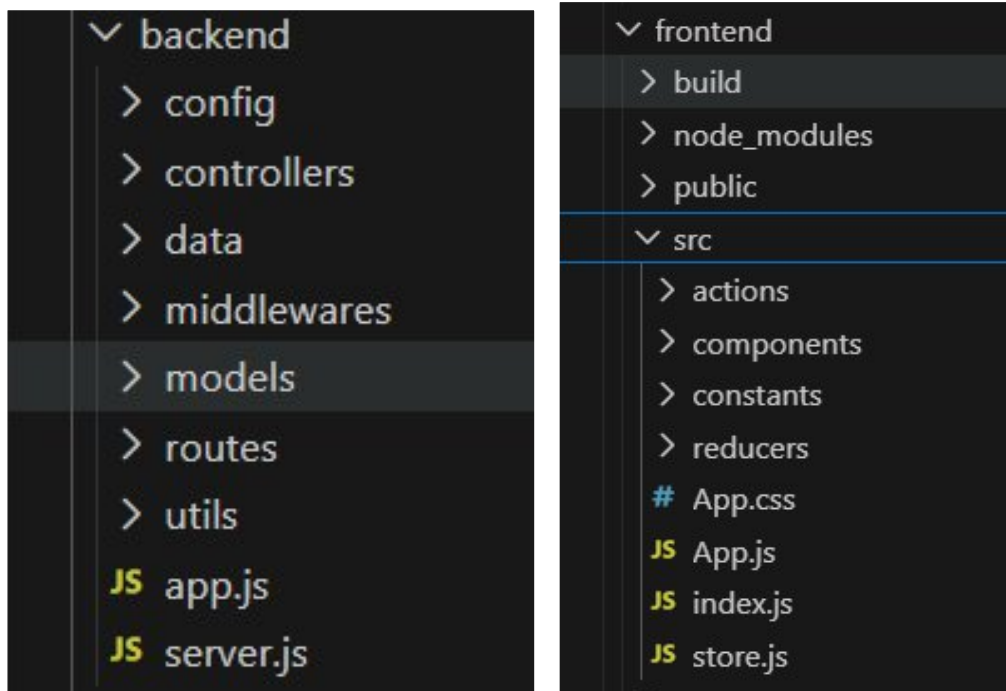
- Redux hỗ trợ quản lý trạng thái ứng dụng trên frontend, giúp dữ liệu nhất quán giữa các thành phần giao diện.

- Chart.js được tích hợp để hiển thị các biểu đồ thống kê doanh thu, số lượng ca làm việc, lịch làm việc..

3.3. Quá trình xây dựng hệ thống

3.3.1. Khởi tạo dự án và cấu trúc thư mục

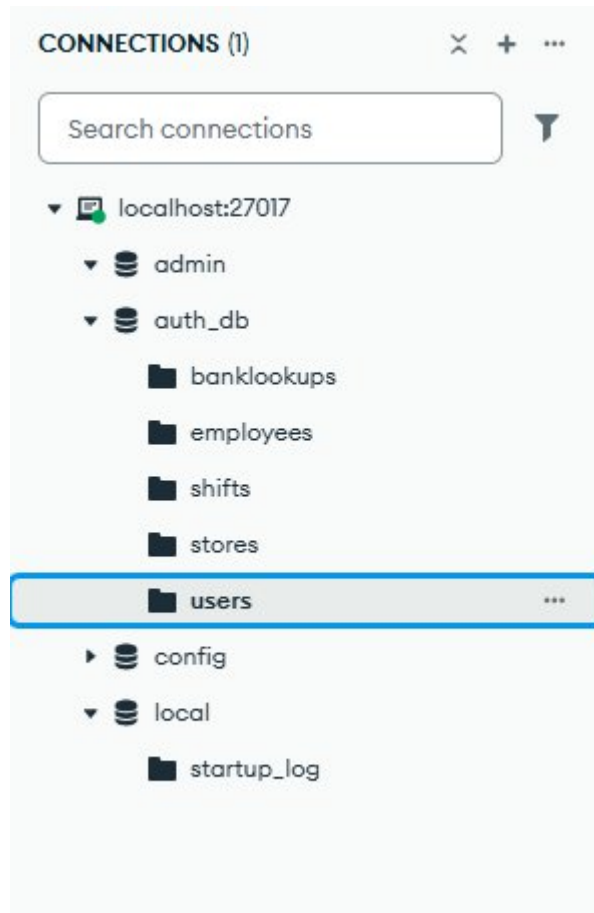
Thực hiện tạo hai thư mục chính là backend và frontend. Thư mục backend chứa mã nguồn *Node.js/Express*, các controller, model, route, middleware, cấu hình cơ sở dữ liệu và các tiện ích. Thư mục frontend chứa mã nguồn ReactJS, các component, actions, reducers, store, và các file cấu hình giao diện.



Hình 3.1: Bố cục giao diện Backend và Frontend

3.3.2. Thiết kế và xây dựng cơ sở dữ liệu

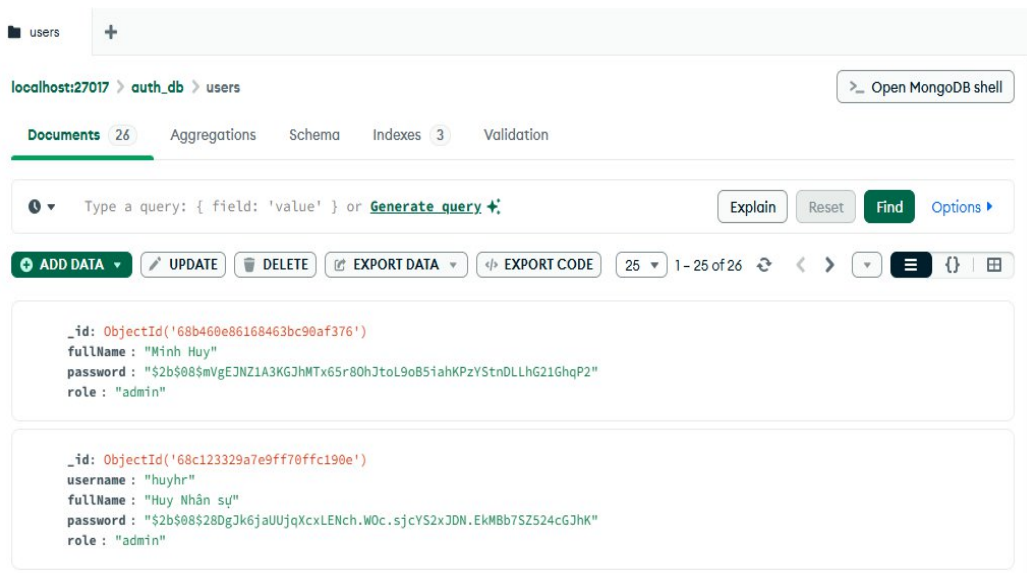
Hệ thống đã sử dụng MongoDB để lưu trữ dữ liệu. Các collection chính gồm: users, employees, shifts, stores.



Hình 3.2: Các Collection chính của hệ thống trong CSDL MongoDB

Chi tiết và các thuộc tính trong các Collection chính:

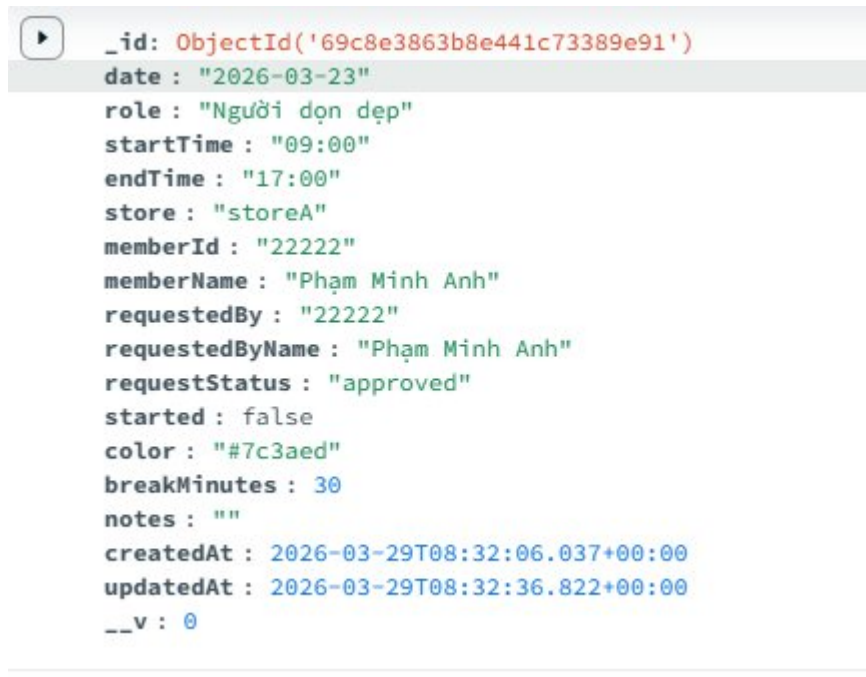
- User:



Hình 3.3: Các thuộc tính của đối tượng người dùng (Users)

Mô tả thuộc tính của đối tượng người dùng trong hệ thống:

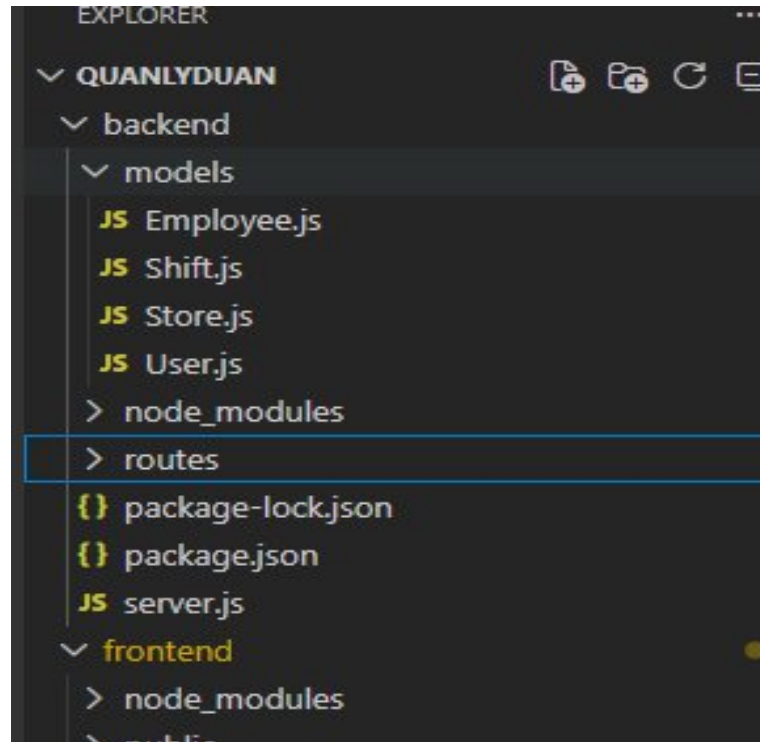
- + *_id*: Mã định danh duy nhất của người dùng.
- + *role*: Vai trò của người dùng trong hệ thống, ví dụ: "admin" , dùng để phân quyền truy cập và thao tác.
- + *name*: Họ và tên đầy đủ của người dùng, phục vụ cho việc hiển thị và quản lý thông tin cá nhân.
- + *email*: Địa chỉ email của người dùng, dùng để đăng nhập và liên hệ.
- + *password*: Mật khẩu của người dùng đã được mã hóa (băm), đảm bảo an toàn bảo mật thông tin đăng nhập.
- + *avatar*: Thông tin về ảnh đại diện của người dùng, bao gồm:
 - public_id*: Mã định danh ảnh trên hệ thống lưu trữ (ví dụ Cloudinary).
 - url*: Đường dẫn truy cập trực tiếp đến ảnh đại diện.
 - createdAt*: Thời gian tài khoản được tạo, giúp theo dõi lịch sử đăng ký của người dùng.
- *employees*: Lưu trữ thông tin nhân viên như họ tên, email, số điện thoại, chức vụ, trạng thái làm việc và các thông tin liên quan đến quản lý nhân viên và ca làm việc.
- *shifts*: Lưu trữ dữ liệu về các ca làm việc, bao gồm thời gian bắt đầu, thời gian kết thúc, vị trí làm việc, trạng thái ca và thông tin nhân viên được phân công.



Hình 3.4: Các thuộc tính của nhân viên

Mô tả thuộc tính

- + *_id*: Mã định danh duy nhất của ca làm việc trong hệ thống.
 - + *date*: Ngày diễn ra ca làm việc.
 - + *role*: Vị trí hoặc chức vụ được phân công trong ca làm việc (ví dụ: Người dọn dẹp, Bảo vệ, Lễ tân,...).
 - + *startTime*: Thời gian bắt đầu ca làm việc.
 - + *endTime*: Thời gian kết thúc ca làm việc.
 - + *store*: Mã hoặc tên của hàng nơi nhân viên thực hiện ca làm việc.
 - + *memberId*: Mã định danh của nhân viên được phân công hoặc được duyệt thực hiện ca làm việc.
 - + *memberName*: Họ và tên của nhân viên được phân công vào ca làm việc.
 - + *requestedBy*: Mã định danh của nhân viên gửi yêu cầu đăng ký hoặc nhận ca làm việc.
 - + *requestedByName*: Họ và tên của nhân viên gửi yêu cầu đăng ký ca làm việc.
 - + *requestStatus*: Trạng thái xử lý của ca làm việc hoặc yêu cầu đăng ký ca (ví dụ: *pending*, *approved*, *rejected*, *waiting_checkin*, *completed*).
 - + *started*: Trạng thái bắt đầu ca làm việc, cho biết nhân viên đã thực hiện check-in hay chưa.
 - + *color*: Mã màu được sử dụng để hiển thị ca làm việc trên giao diện lịch trình.
 - + *breakMinutes*: Thời gian nghỉ giữa ca (đơn vị: phút).
 - =+ *notes*: Ghi chú bổ sung liên quan đến ca làm việc.
 - + *createdAt*: Thời điểm ca làm việc được tạo trong hệ thống.*updatedAt*: Thời điểm thông tin ca làm việc được cập nhật gần nhất.
 - + *__v*: Trường phiên bản dữ liệu do MongoDB/Mongoose tự động quản lý. Các schema được định nghĩa trong thư mục *backend/models* như *user.js* ,v.v. Dữ liệu mẫu được lưu trong các file JSON để phục vụ kiểm thử.
- Bộ cục Modal backend

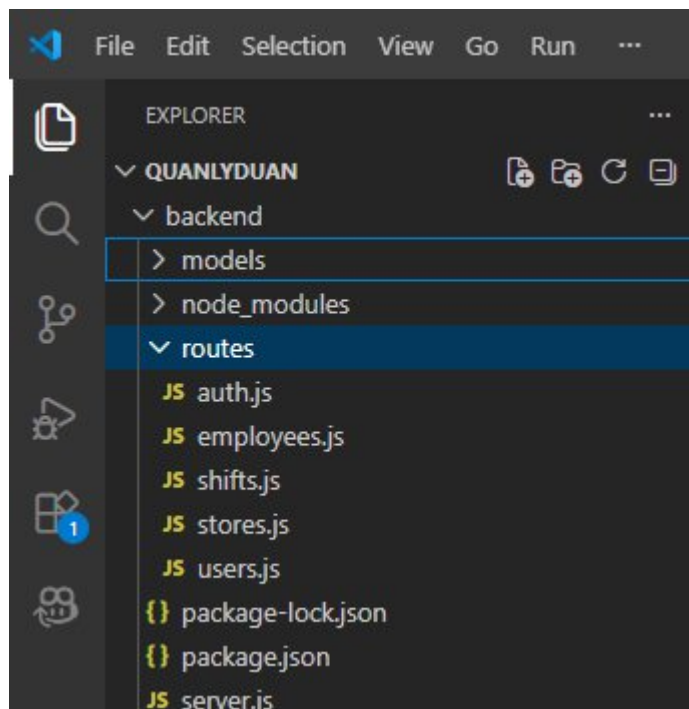


Hình 3.5: Bố cục của thư mục Models trong Backend

3.3.3. Phát triển BackEnd

Tiếp theo là xây dựng các API RESTful cho các chức năng: đăng ký, đăng nhập, quản lý người dùng, quản lý nhân viên.

- Các *controller* xử lý logic trong *backend/router*:



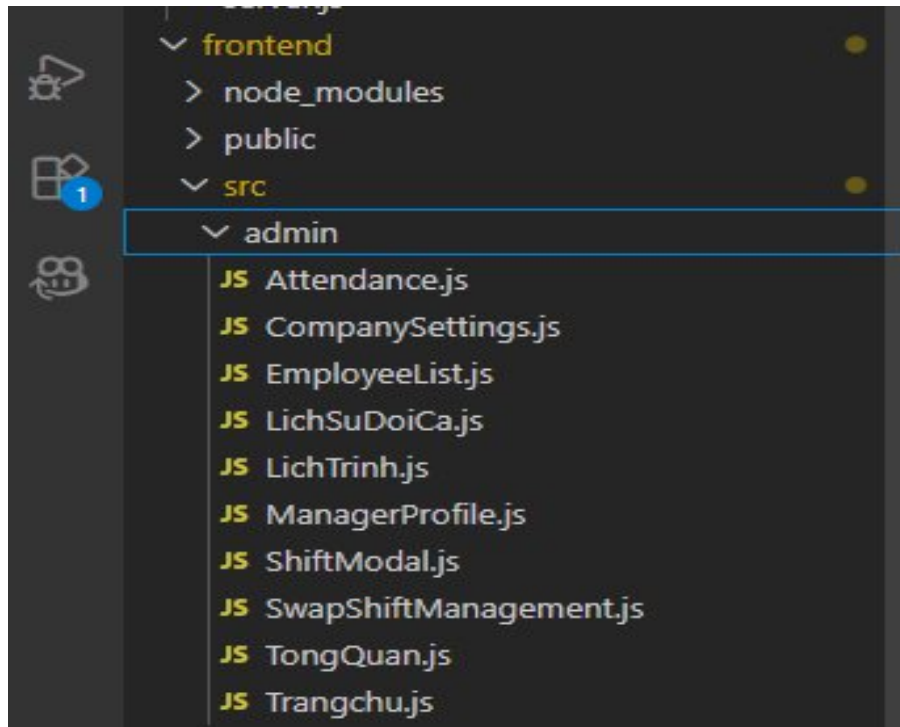
Hình 3.6: Các API RESTful cho các chức năng trong mục Router

3.3.4. Phát triển Frontend

Giao diện người dùng được xây dựng bằng ReactJS theo mô hình Component-Based Architecture, trong đó các chức năng của hệ thống được chia thành các thư mục riêng biệt nhằm dễ dàng quản lý, bảo trì và mở rộng trong tương lai

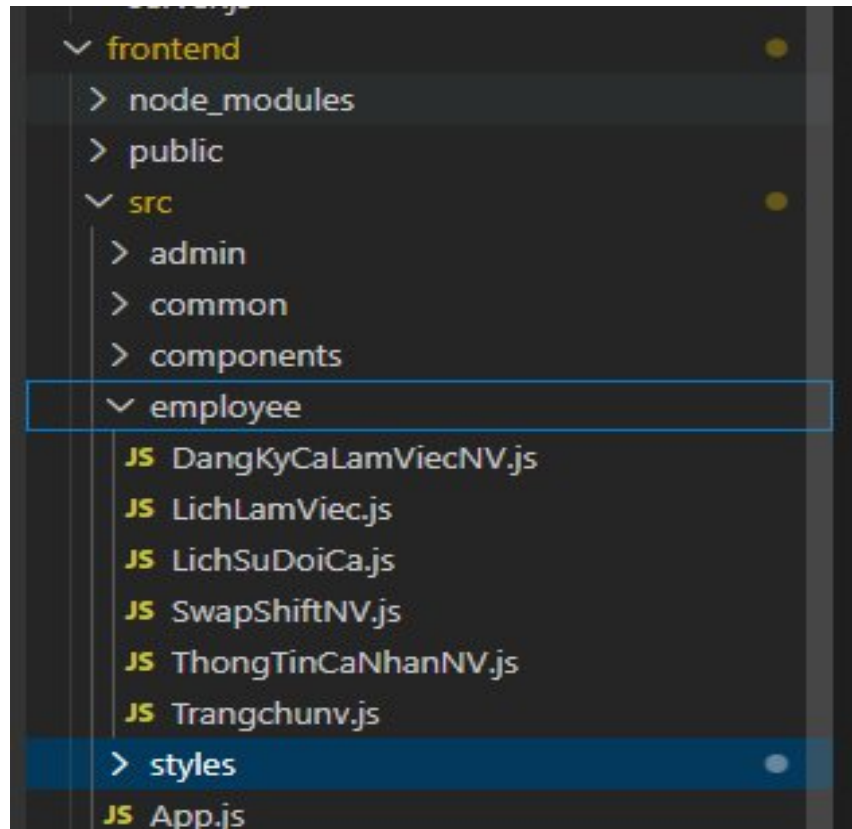
Cấu trúc thư mục Frontend được tổ chức như sau:

admin/: Chứa các giao diện dành cho quản trị viên như quản lý nhân viên, quản lý lịch trình, duyệt tài khoản, duyệt yêu cầu đổi ca, chấm công và thống kê báo cáo.



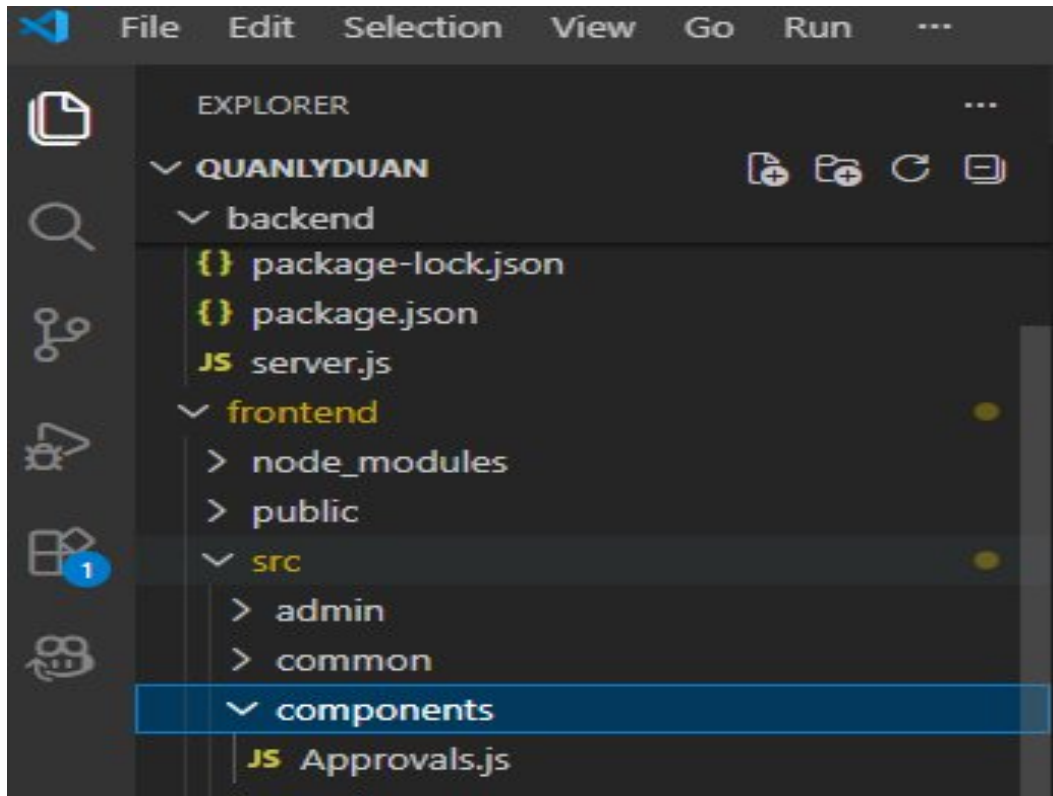
Hình 3.7: Bố cục admin

employee/: Chứa các giao diện dành cho nhân viên như đăng ký tài khoản, đăng nhập, xem lịch làm việc, đăng ký ca làm việc, gửi yêu cầu đổi ca, theo dõi trạng thái yêu cầu, lịch sử đổi ca và cập nhật thông tin cá nhân.



Hình 3.8: Bố cục các employee trong Frontend

components/: Chứa các component dùng chung được tái sử dụng trong nhiều chức năng khác nhau như modal, bảng dữ liệu, form nhập liệu, thanh điều hướng và các thành phần giao diện hỗ trợ.



Hình 3.9: Bố cục các components trong Frontend

3.3.5. Tích hợp các chức năng và dịch vụ

Hệ thống sử dụng Axios để giao tiếp giữa frontend và backend. Các chức năng như tìm kiếm nhân viên, phân trang, quản lý trạng thái đổi ca, thống kê doanh thu đều được hiện thực hóa rõ ràng.

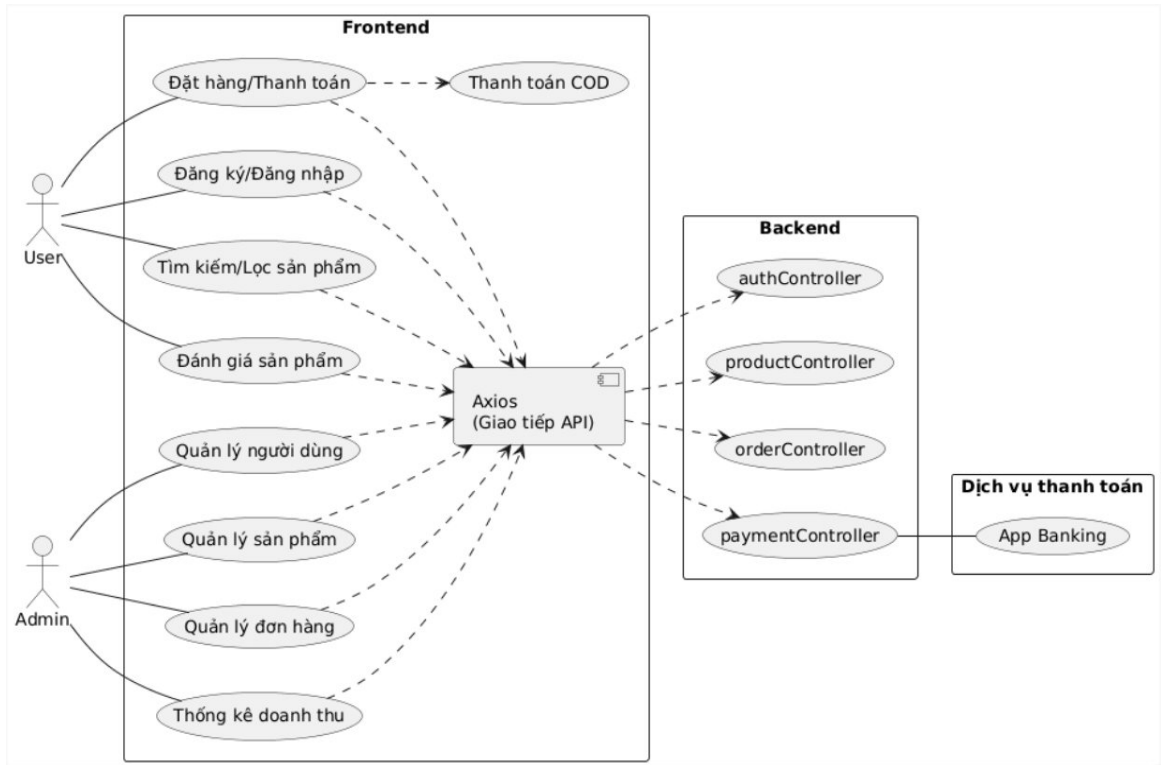
- Giao tiếp giữa Backend với Frontend

- + Được thực hiện qua thư viện Axios ở phía frontend (thường trong các file actions hoặc services của React).

- + Các request này sẽ gọi đến các API RESTful do backend xây dựng (các controller trong backend)

- Các chức năng nghiệp vụ khác

- + Được hiện thực hóa ở cả frontend (giao diện, gọi API) và backend.

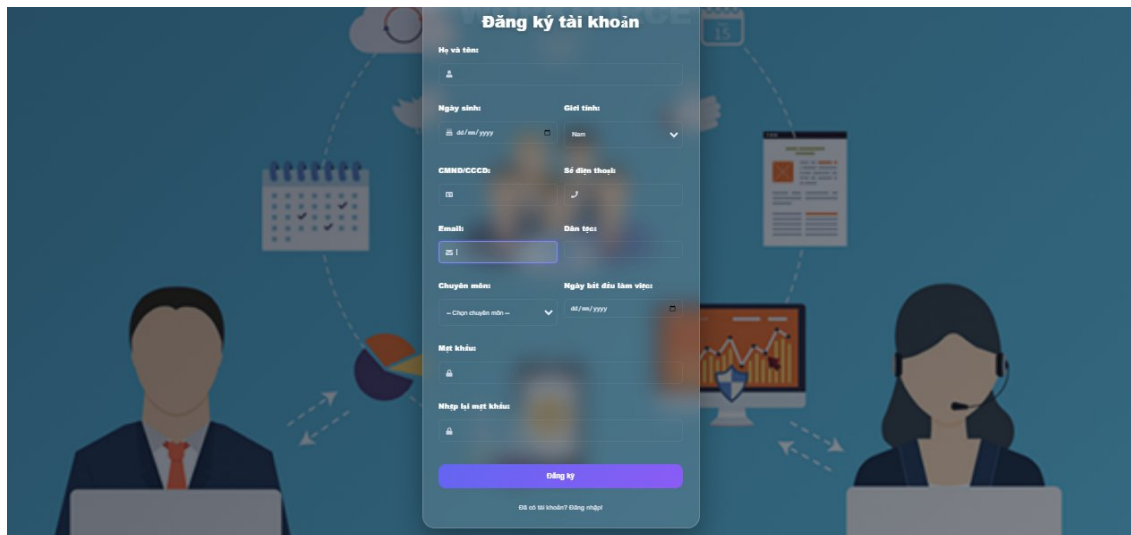


Hình 3.10: Sơ đồ giao tiếp của Axios trong hệ thống

3.4. Xây dựng các chức năng chính

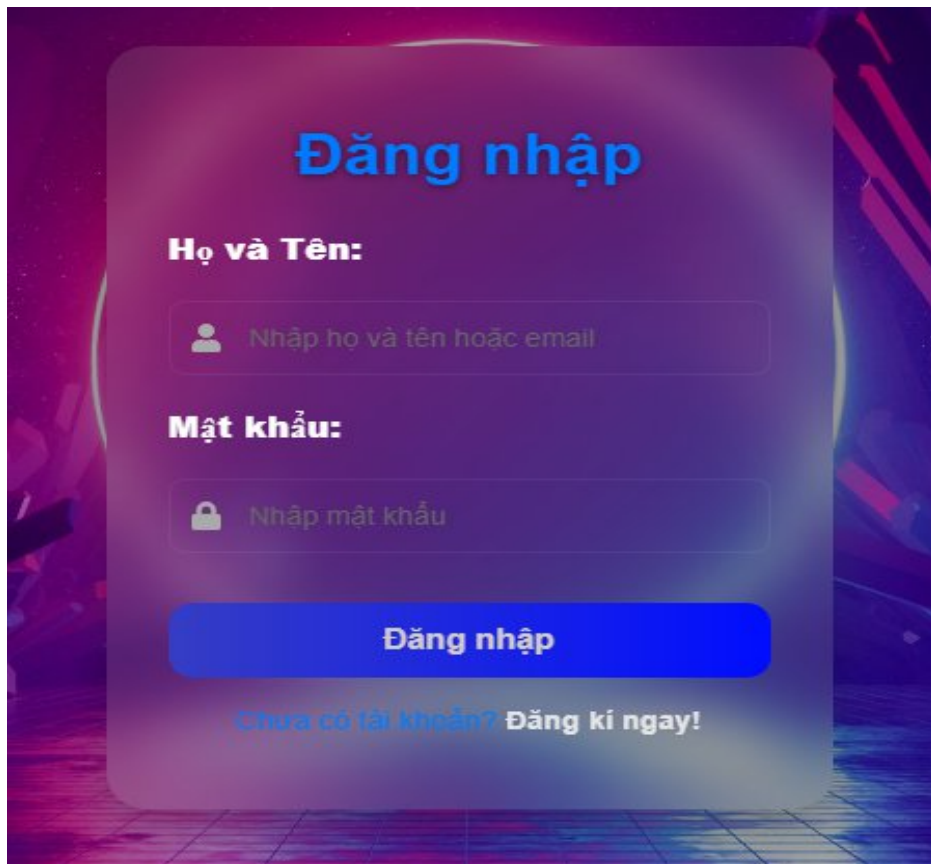
3.4.1. Đăng kí/Đăng nhập

- Đăng kí:



Hình 3.11: Giao diện đăng kí.

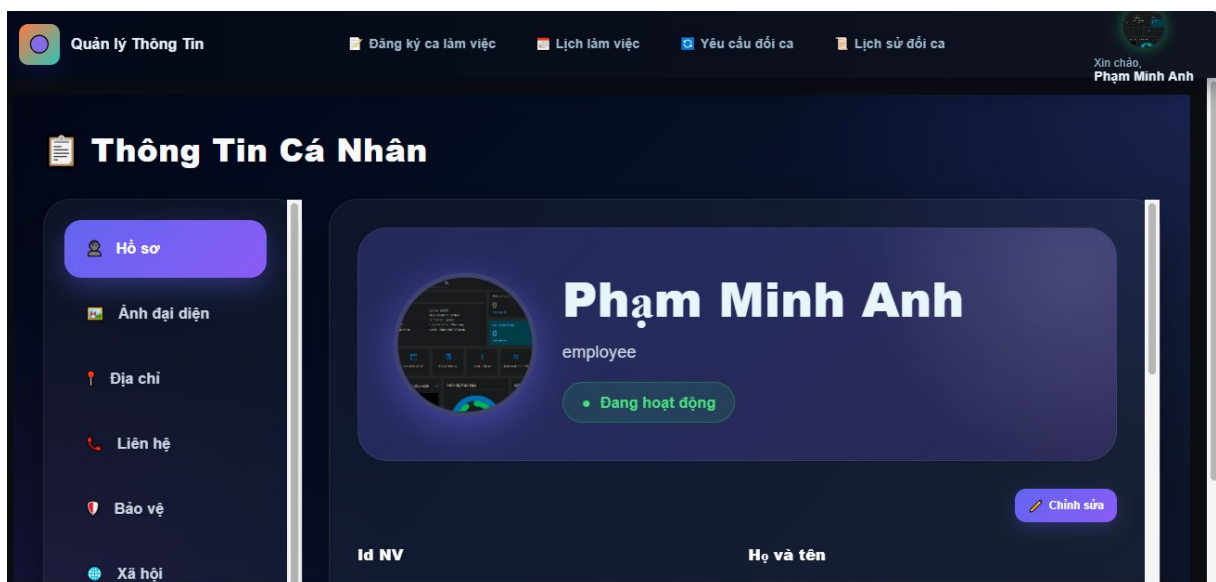
- Đăng nhập:



Hình 3.12: Giao diện đăng nhập

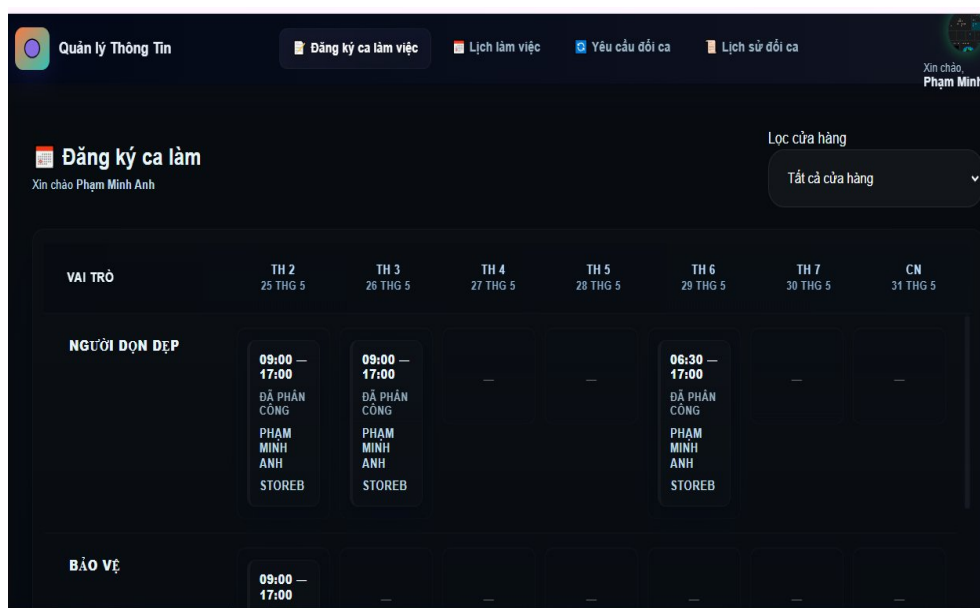
3.4.2. Giao diện nhân viên

- Thông tin cá nhân nhân viên:



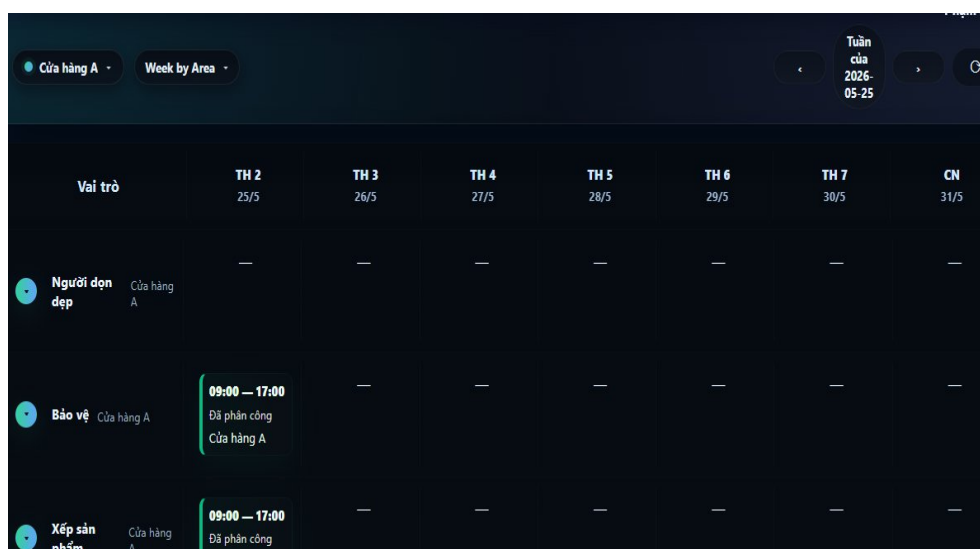
Hình 3.13: Giao diện thông tin cá nhân nhân viên

- Đăng kí ca làm việc của nhân viên



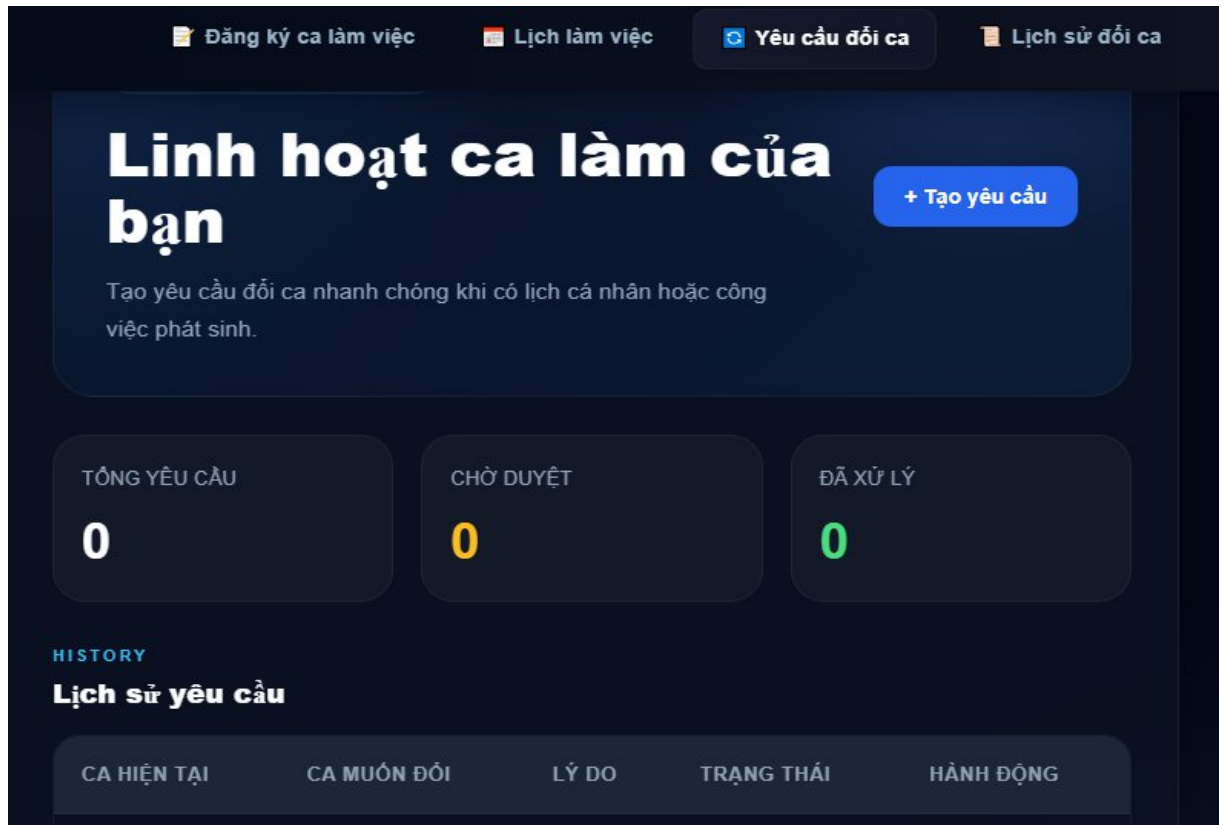
Hình 3.14: Giao diện đăng kí ca làm việc của nhân viên

- Lịch làm việc của nhân viên



Hình 3.15: Giao diện lịch làm việc của nhân viên

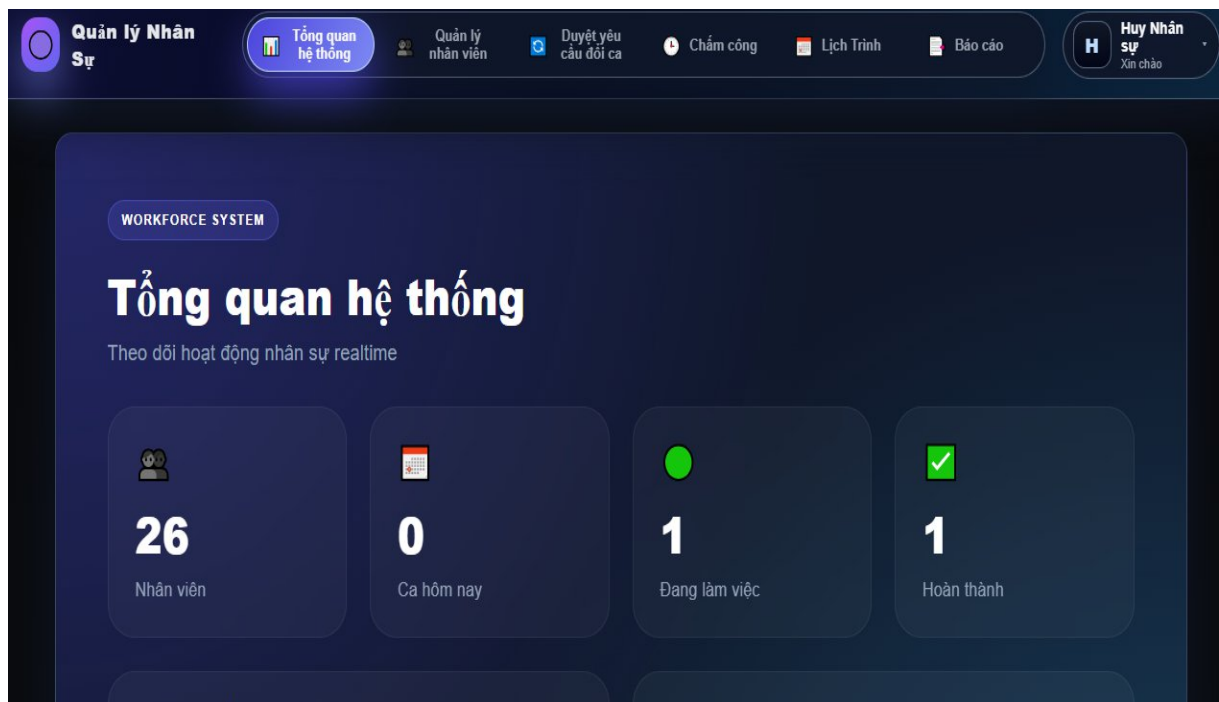
- Yêu cầu đổi ca của nhân viên



Hình 3.16: Giao diện yêu cầu đổi ca

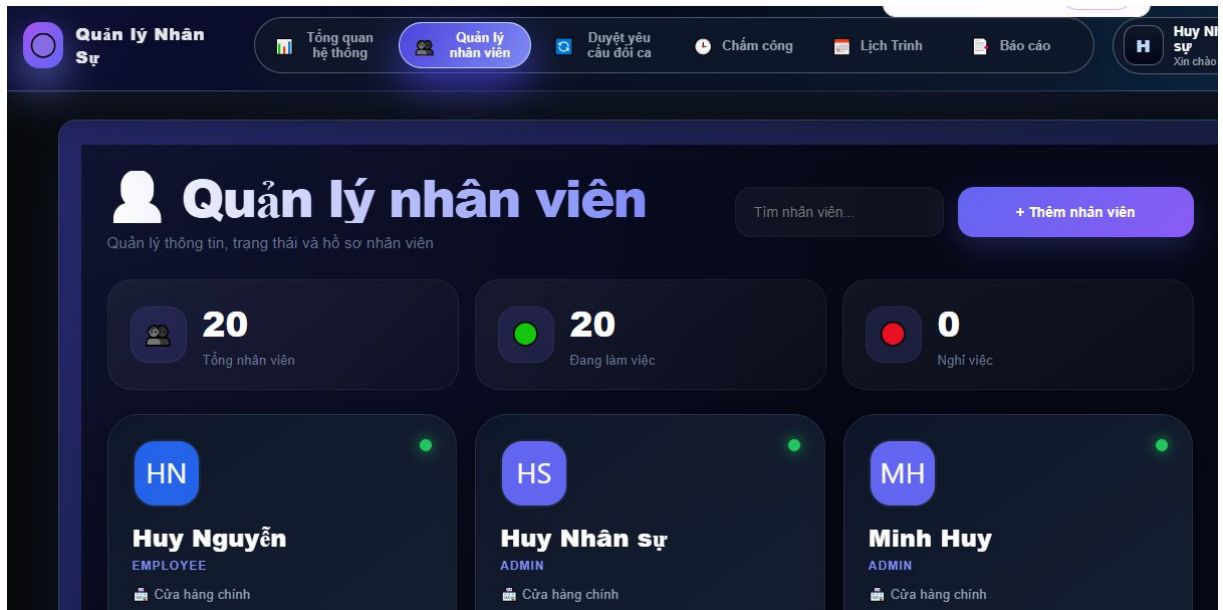
3.4.3. Giao diện Admin

- Tổng quan hệ thống:



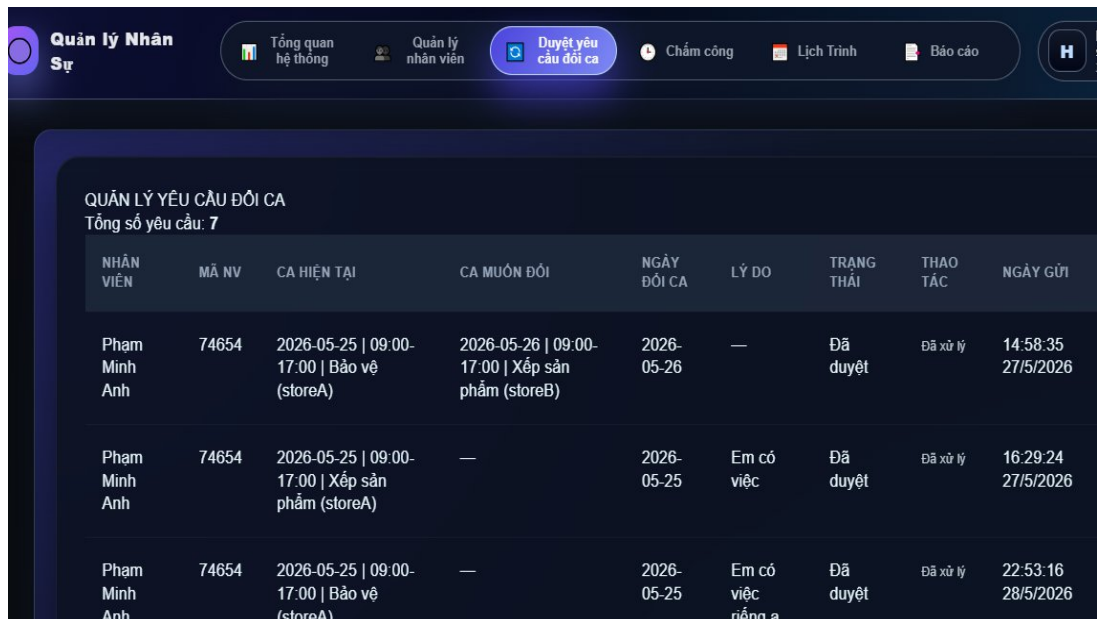
Hình 3.17: Giao diện tổng quan hệ thống

- Thêm/sửa/xoá nhân viên:



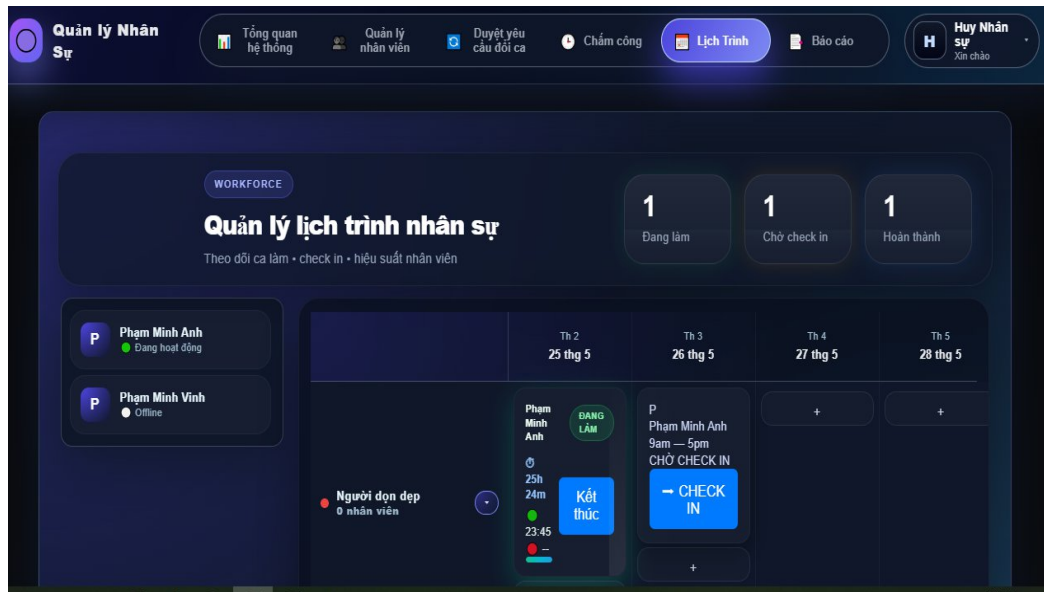
Hình 3.18: Giao diện quản lý nhân viên

- Duyệt yêu cầu đổi ca của nhân viên



Hình 3.19: Giao diện quản lý yêu cầu đổi ca.

- Lịch và tạo ca làm cho nhân viên



Hình 3.20: Giao diện tạo ca làm cho nhân viên

3.5. Kiểm thử và đánh giá hệ thống

3.5.1. Mục tiêu kiểm thử

Mục tiêu của quá trình kiểm thử là đảm bảo website quản lý nhân viên và ca làm việc hoạt động đúng theo các yêu cầu đã được phân tích và thiết kế. Quá trình kiểm thử nhằm xác minh tính chính xác của các chức năng, phát hiện và khắc phục các lỗi có thể ảnh hưởng đến hoạt động của hệ thống trước khi đưa vào sử dụng thực tế.

Việc kiểm thử tập trung đánh giá các chức năng chính như đăng nhập và phân quyền người dùng, quản lý hồ sơ nhân viên, đăng ký ca làm việc, xem lịch làm việc, quản lý ca làm việc, gửi và xử lý yêu cầu đổi ca, cũng như hệ thống thông báo. Bên cạnh đó, kiểm thử còn giúp đánh giá khả năng tương tác giữa giao diện ReactJS, máy chủ Node.js và cơ sở dữ liệu MongoDB nhằm đảm bảo dữ liệu được xử lý và đồng bộ chính xác.

Ngoài kiểm thử chức năng, hệ thống cũng được đánh giá về các yêu cầu phi chức năng như hiệu năng xử lý, tính ổn định, khả năng mở rộng và mức độ bảo mật trong quá trình xác thực người dùng và quản lý dữ liệu. Thông qua quá trình kiểm thử, hệ thống được hoàn thiện hơn, nâng cao độ tin cậy và đáp ứng tốt nhu cầu quản lý nhân viên và ca làm việc cho chuỗi cửa hàng.

3.5.2. Quy trình và phương pháp kiểm thử

Quy trình kiểm thử hệ thống được thực hiện theo các bước sau nhằm đảm bảo hệ thống hoạt động ổn định, chính xác và đáp ứng các yêu cầu đã đề ra:

Kiểm thử chức năng (Functional Testing): Thực hiện kiểm tra các chức năng chính của hệ thống như đăng nhập và phân quyền người dùng, quản lý hồ sơ nhân viên, đăng ký ca làm việc, xem lịch làm việc, quản lý ca làm việc, gửi yêu cầu đổi ca, duyệt hoặc từ chối yêu cầu đổi ca và hệ thống thông báo. Quá trình kiểm thử nhằm đảm bảo các chức năng hoạt động đúng theo yêu cầu nghiệp vụ và dữ liệu được xử lý chính xác.

Kiểm thử giao diện người dùng (UI Testing): Đánh giá tính trực quan, thân thiện và dễ sử dụng của giao diện người dùng. Kiểm tra khả năng hiển thị và tương thích của hệ thống trên nhiều trình duyệt và thiết bị khác nhau nhằm đảm bảo trải nghiệm người dùng được nhất quán.

Kiểm thử hiệu năng (Performance Testing): Đánh giá tốc độ phản hồi của hệ thống trong quá trình xử lý các thao tác như đăng nhập, đăng ký ca làm việc, truy xuất lịch làm việc và quản lý dữ liệu nhân viên. Đồng thời kiểm tra khả năng xử lý đồng thời nhiều yêu cầu từ người dùng nhằm đảm bảo hệ thống hoạt động ổn định trong môi trường thực tế.

Kiểm thử bảo mật (Security Testing): Kiểm tra các lỗ hổng bảo mật, xác thực và phân quyền người dùng.

3.5.3. Kết quả kiểm thử

Sau quá trình kiểm thử, hệ thống đạt được các kết quả như sau:

- Các chức năng chính của hệ thống như đăng nhập, phân quyền người dùng, quản lý hồ sơ nhân viên, đăng ký ca làm việc, xem lịch làm việc, quản lý ca làm việc, gửi và xử lý yêu cầu đổi ca đều hoạt động ổn định và đáp ứng đúng yêu cầu nghiệp vụ đã đặt ra.
- Giao diện người dùng được thiết kế trực quan, dễ sử dụng và hiển thị tốt trên các trình duyệt phổ biến cũng như các thiết bị có kích thước màn hình khác nhau, đảm bảo trải nghiệm người dùng.
- Hệ thống có thời gian phản hồi nhanh trong quá trình xử lý các thao tác như đăng nhập, truy xuất dữ liệu nhân viên, đăng ký ca làm việc và cập nhật lịch

làm việc. Hệ thống có khả năng đáp ứng tốt với số lượng người dùng truy cập đồng thời ở mức trung bình.

- Dữ liệu được đồng bộ chính xác giữa giao diện ReactJS, máy chủ Node.js và cơ sở dữ liệu MongoDB, đảm bảo tính nhất quán và toàn vẹn dữ liệu.

Trong quá trình kiểm thử, không phát hiện các lỗ hổng bảo mật nghiêm trọng. Cơ chế xác thực và phân quyền giữa nhân viên và quản lý hoạt động đúng theo thiết kế, giúp bảo vệ dữ liệu và hạn chế truy cập trái phép.

3.5.4. Đánh giá tổng hệ thống

Qua quá trình kiểm thử và đánh giá, có thể nhận thấy hệ thống đã đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng đã đề ra. Các chức năng quản lý nhân viên, quản lý ca làm việc, đăng ký ca, xử lý yêu cầu đổi ca và quản lý thông tin cá nhân đều hoạt động ổn định, hỗ trợ hiệu quả cho công tác quản lý nhân viên và ca làm việc tại chuỗi cửa hàng.

Kiến trúc hệ thống sử dụng ReactJS, Node.js và MongoDB giúp đảm bảo khả năng mở rộng, dễ bảo trì và thuận tiện trong quá trình phát triển các tính năng mới trong tương lai. Giao diện thân thiện, dễ sử dụng giúp người dùng thao tác nhanh chóng và giảm thiểu sai sót trong quá trình sử dụng.

Mặc dù hệ thống đã đáp ứng tốt các yêu cầu hiện tại, nhưng trong quá trình triển khai thực tế vẫn cần tiếp tục theo dõi, kiểm thử và tối ưu hiệu năng để đáp ứng tốt hơn khi số lượng nhân viên, dữ liệu và lượng truy cập tăng lên. Ngoài ra, có thể nghiên cứu bổ sung các tính năng như thống kê nâng cao, thông báo thời gian thực và tích hợp chấm công để nâng cao hiệu quả quản lý trong tương lai.

3.6. Tổng kết chương.

Trong chương này, em đã trình bày quá trình xây dựng và triển khai website quản lý nhân viên và ca làm việc cho chuỗi cửa hàng dựa trên các nội dung phân tích và thiết kế đã được thực hiện ở chương trước. Hệ thống được phát triển theo kiến trúc ba lớp sử dụng ReactJS cho giao diện người dùng, Node.js và ExpressJS cho xử lý nghiệp vụ phía máy chủ, cùng MongoDB để lưu trữ và quản lý dữ liệu.

Bên cạnh đó, các chức năng chính của hệ thống như đăng nhập và phân quyền người dùng, quản lý hồ sơ nhân viên, đăng ký ca làm việc, xem lịch làm việc, quản lý

ca làm việc, gửi và xử lý yêu cầu đổi ca đã được triển khai và kiểm thử. Quá trình kiểm thử được thực hiện trên nhiều khía cạnh bao gồm chức năng, giao diện, hiệu năng và bảo mật nhằm đảm bảo hệ thống đáp ứng các yêu cầu đã đề ra.

Kết quả kiểm thử cho thấy hệ thống hoạt động ổn định, các chức năng được thực hiện chính xác theo yêu cầu nghiệp vụ, giao diện thân thiện và dễ sử dụng. Đồng thời, hệ thống đảm bảo khả năng lưu trữ và xử lý dữ liệu hiệu quả, đáp ứng nhu cầu quản lý nhân viên và ca làm việc trong môi trường chuỗi cửa hàng.

Tuy nhiên, để hệ thống vận hành hiệu quả hơn trong thực tế và đáp ứng nhu cầu phát triển lâu dài, cần tiếp tục theo dõi, kiểm thử và tối ưu hóa hiệu năng khi số lượng người dùng và dữ liệu tăng lên. Ngoài ra, hệ thống có thể được mở rộng thêm các chức năng như thống kê nâng cao, thông báo thời gian thực, tích hợp chấm công và báo cáo quản lý nhằm nâng cao hiệu quả quản trị nhân sự trong tương lai.

KẾT LUẬN

Kết luận:

Sau quá trình nghiên cứu, phân tích, thiết kế và xây dựng, đề tài “Xây dựng website quản lý nhân viên và ca làm việc của nhân viên cho chuỗi cửa hàng sử dụng ReactJS, Node.js và MongoDB” đã hoàn thành các mục tiêu đề ra. Hệ thống cung cấp các chức năng cơ bản phục vụ cho công tác quản lý nhân viên và ca làm việc và ca làm việc như quản lý thông tin nhân viên, đăng nhập và phân quyền người dùng, đăng ký ca làm việc, xem lịch làm việc, quản lý ca làm việc, gửi và xử lý yêu cầu đổi ca.

Việc áp dụng ReactJS trong xây dựng giao diện người dùng, Node.js và ExpressJS trong xử lý nghiệp vụ cùng MongoDB trong lưu trữ dữ liệu đã giúp hệ thống hoạt động ổn định, dễ mở rộng và thuận tiện trong quá trình phát triển. Hệ thống góp phần hỗ trợ các chuỗi cửa hàng quản lý nhân viên và lịch làm việc hiệu quả hơn, giảm thiểu các thao tác thủ công và nâng cao khả năng quản lý tập trung.

Kết quả kiểm thử cho thấy các chức năng của hệ thống hoạt động đúng theo yêu cầu, giao diện thân thiện, dễ sử dụng và đáp ứng được các nhu cầu cơ bản của người dùng trong môi trường thực tế.

Hạn chế:

Mặc dù đã đáp ứng được các chức năng chính của hệ thống quản lý nhân viên và ca làm việc, đề tài vẫn còn một số hạn chế như:

- Chưa tối ưu hoàn toàn hiệu năng khi số lượng nhân viên và dữ liệu tăng lên ở quy mô lớn.
- Chức năng thống kê và báo cáo còn ở mức cơ bản, chưa cung cấp nhiều biểu đồ và phân tích chuyên sâu.
- Hệ thống thông báo hiện tại chưa hỗ trợ cập nhật thời gian thực (real-time).
- Chưa tích hợp chức năng chấm công và theo dõi thời gian làm việc thực tế của nhân viên.
- Việc kiểm thử chủ yếu được thực hiện thủ công, chưa áp dụng các công cụ kiểm thử tự động để nâng cao chất lượng phần mềm.
- Các cơ chế bảo mật vẫn còn ở mức cơ bản và cần được nâng cao hơn khi triển khai trong môi trường thực tế.

Hướng phát triển:

Trong tương lai, hệ thống có thể được nghiên cứu và phát triển thêm các chức năng sau:

Tích hợp chức năng chấm công bằng mã QR, GPS hoặc nhận diện khuôn mặt.

Xây dựng hệ thống thông báo thời gian thực thông qua WebSocket hoặc Socket.IO.

Bổ sung các chức năng thống kê và báo cáo nâng cao dưới dạng biểu đồ trực quan.

Phát triển ứng dụng di động trên nền tảng Android và iOS để hỗ trợ nhân viên thao tác thuận tiện hơn.

Tích hợp chức năng quản lý lương, thưởng và tính công dựa trên lịch làm việc thực tế.

Nâng cao khả năng bảo mật bằng xác thực hai lớp (Two-Factor Authentication), mã hóa dữ liệu và các cơ chế bảo vệ hiện đại.

Tối ưu hiệu năng hệ thống để đáp ứng số lượng lớn người dùng đồng thời.

Áp dụng các công cụ kiểm thử tự động và quy trình CI/CD nhằm nâng cao chất lượng phần mềm và hỗ trợ triển khai hệ thống hiệu quả hơn.

Với những kết quả đạt được, đề tài đã xây dựng được nền tảng cơ bản cho một hệ thống quản lý nhân viên và ca làm việc phục vụ cho chuỗi cửa hàng, đồng thời tạo tiền đề cho việc mở rộng và phát triển các tính năng nâng cao trong tương lai.

TÀI LIỆU THAM KHẢO

1. Nguyễn Thanh Bình, “Giáo trình Phát triển ứng dụng Web với Node.js và React,” NXB Bách Khoa Hà Nội, 2021.
<https://www.nxbkhn.vn/>
2. Nguyễn Văn Hiếu, “Lập trình Web với JavaScript, HTML5 và CSS3,” NXB Thông tin và Truyền thông, 2020.
<https://www.nxbtttt.vn/>
3. Nguyễn Văn Tuấn, “An toàn và bảo mật hệ thống thông tin,” NXB Khoa học và Kỹ thuật, 2019.
<https://www.nxbkhkt.com.vn/>
4. MDN Web Docs, “Express.js Guide,” Mozilla, 2025.
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs
5. ReactJS, “Getting Started – React,” Meta, 2025.
<https://react.dev/learn>
6. OWASP Foundation, “OWASP Top Ten Web Application Security Risks,” 2025.
<https://owasp.org/www-project-top-ten/>