**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG CHẤM CÔNG SỬ DỤNG KHUÔN MẶT CHO DOANH NGHIỆP**

|  |  |
| --- | --- |
| CBHD: | ThS. Nguyễn Trung Phú |
| Sinh viên: | Bùi Đức Lương |
| Mã sinh viên: | 2021600133 |

**Hà Nội, 2025**

**Lời cảm ơn**

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè đã giúp em hoàn thiện đồ án tốt nghiệp.

Trước tiên em xin gửi tới các thầy cô khoa Công nghệ thông tin trường Đại học Công Nghiệp Hà Nội lời chúc sức khỏe và lời cảm ơn chân thành. Với sự quan tâm, chỉ bảo của thầy cô đã tạo điều kiện cho em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp lần này.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Trung Phú đã trực tiếp tận tình hướng dẫn cũng như nhận xét và giúp đỡ em trong suốt quá trình hoàn thiện đồ án.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không tránh khỏi những thiếu sót. Em rất mong nhận được sự giúp đỡ của thầy cô và các bạn để hoàn thiện hơn đồ án của mình.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm  Sinh viên thực hiện |
|  |  |
|  |  |

**Mục lục**

[**Lời cảm ơn** 1](#_Toc198656382)

[Danh mục hình ảnh 5](#_Toc198656383)

[Lời mở đầu 7](#_Toc198656384)

[Lý do chọn đề tài 8](#_Toc198656385)

[Đặt vấn đề 9](#_Toc198656386)

[Ý nghĩa và cơ sở lý luận 10](#_Toc198656387)

[Mục tiêu và phạm vi đồ án 12](#_Toc198656388)

[CHƯƠNG 1 : TỔNG QUAN 14](#_Toc198656389)

[1.1. Tổng quan về các hình thức chấm công của doanh nghiệp. 14](#_Toc198656390)

[1.1.1. Chấm công bằng thẻ từ 14](#_Toc198656391)

[1.1.2. Chấm công bằng vân tay 15](#_Toc198656392)

[1.1.3. Chấm công bằng phần mềm GPS 15](#_Toc198656393)

[1.1.4. Chấm công sử dụng khuôn mặt 15](#_Toc198656394)

[1.2. Cơ sở lý thuyết về nhận diện và so sánh khuôn mặt 16](#_Toc198656395)

[1.2.1. Khái niệm và vai trò 16](#_Toc198656396)

[1.2.2. Quy trình nhận diện khuôn mặt 16](#_Toc198656397)

[1.2.3. Các phương pháp nhận diện khuôn mặt 17](#_Toc198656398)

[1.3. Lựa chọn công nghệ 17](#_Toc198656399)

[1.3.1. Python FastAPI 17](#_Toc198656400)

[1.3.2. Thư viện Face Recognition 18](#_Toc198656401)

[CHƯƠNG 2 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG 21](#_Toc198656402)

[2.1. Các tác nhân của hệ thống 21](#_Toc198656403)

[2.2. Hoạt động của hệ thống 21](#_Toc198656404)

[2.2.1. Hoạt động chấm công 21](#_Toc198656405)

[2.2.2. Hoạt động quản lý thông tin 21](#_Toc198656406)

[2.2.3. Hoạt động nhắn tin thời gian thực 22](#_Toc198656407)

[2.3. Các yêu cầu chức năng 22](#_Toc198656408)

[2.3.1. Phía nhân viên 22](#_Toc198656409)

[2.3.2. Phía trưởng phòng, giám đốc 22](#_Toc198656410)

[2.3.3. Phía admin 23](#_Toc198656411)

[2.4. Biểu đồ use case 24](#_Toc198656412)

[2.4.1. Biểu đồ use case tổng quát 24](#_Toc198656413)

[2.4.2. Biểu đồ phân rã use case cho nhân viên 25](#_Toc198656414)

[2.4.3. Biểu đồ phân rã use case cho trưởng phòng, giám đốc 27](#_Toc198656415)

[2.4.4. Biểu đồ phân rã use case cho admin 29](#_Toc198656416)

[2.5. Cơ sở dữ liệu 30](#_Toc198656417)

[2.6. Phân tích chi tiết use case 32](#_Toc198656418)

[2.6.1. Use case đăng nhập 32](#_Toc198656419)

[2.6.2. Use case đăng ký 33](#_Toc198656420)

[2.6.3. Use case chấm công 34](#_Toc198656421)

[2.6.4. Use case xem lịch sử chấm công 35](#_Toc198656422)

[2.6.5. Phân tích use case xem lịch sử ngày công 36](#_Toc198656423)

[2.6.6. Phân tích use case nhắn tin 37](#_Toc198656424)

[2.6.7. Phân tích use case tạo nhóm 38](#_Toc198656425)

[2.6.8. Phân tích use case rời nhóm 39](#_Toc198656426)

[2.6.9. Phân tích use case thêm người vào nhóm 40](#_Toc198656427)

[2.6.10. Phân tích use case quản lý thông tin cá nhân 41](#_Toc198656428)

[2.6.11. Phân tích use case xem lịch sử chấm công, ngày công của nhân viên 42](#_Toc198656429)

[2.6.12. Phân tích use case quản lý thông tin nhân viên 44](#_Toc198656430)

[2.7. Phân tích use case 45](#_Toc198656431)

[2.7.1. Use case đăng nhập 45](#_Toc198656432)

[2.7.2. Use case đăng ký 47](#_Toc198656433)

[2.7.3. Use case chấm công 49](#_Toc198656434)

[2.7.4. Use case xem lịch sử chấm công 51](#_Toc198656435)

[2.7.5. Use case xem lại lịch sử ngày công 53](#_Toc198656436)

[2.7.6. Use case nhắn tin thời gian thực 55](#_Toc198656437)

[2.7.7. Use case tạo nhóm chat 57](#_Toc198656438)

[2.7.8. Use case rời nhóm 59](#_Toc198656439)

[2.7.9. Use case thêm thành viên vào nhóm 61](#_Toc198656440)

[2.7.10. Use case cập nhật thông tin 63](#_Toc198656441)

[2.7.11. Use case xem lịch sử chấm công, ngày công của nhân viên 65](#_Toc198656442)

[2.7.12. Use case quản lý thông tin nhân viên 67](#_Toc198656443)

[CHƯƠNG 3 : TRIỂN KHAI HỆ THỐNG VÀ KIỂM THỬ 69](#_Toc198656444)

[3.1. Cài đặt chương trình 69](#_Toc198656445)

[3.1.1. Yêu cầu hệ thống 69](#_Toc198656446)

[3.1.2. Cài đặt chương trình 69](#_Toc198656447)

[3.2. Kiểm thử API ( Sử dụng Postman ) 72](#_Toc198656448)

[3.2.1. Kiểm thử API hệ thống so sánh khuôn mặt 72](#_Toc198656449)

[3.2.2. Kiểm thử API hệ thống quản lý chấm công và quản lý thông tin 73](#_Toc198656450)

[Kết luận 77](#_Toc198656451)

[Tài liệu tham khảo 78](#_Toc198656452)

# Danh mục hình ảnh

[Hình 2.1: Biểu đồ use case tổng quát 24](#_Toc198656453)

[Hình 2.2: Biểu đồ phân rã use case cho nhân viên 25](#_Toc198656454)

[Hình 2.3: Biểu đồ phân rã use case cho quản lí (trưởng phòng, giám đốc) 27](#_Toc198656455)

[Hình 2.4: Biểu đồ phân rã use case cho quản lí (trưởng phòng, giám đốc) 29](#_Toc198656456)

[Hình 2.5: Cơ sở dữ liệu 30](#_Toc198656457)

[Hình 2.6: Biểu đồ trình tự luồng cơ bản use case Đăng nhập 45](#_Toc198656458)

[Hình 2.7: Biểu đồ lớp phân tích use case Đăng nhập 46](#_Toc198656459)

[Hình 2.8: Biểu đồ trình tự luồng cơ bản use case Đăng ký 47](#_Toc198656460)

[Hình 2.9: Biểu đồ lớp phân tích use case Đăng ký 48](#_Toc198656461)

[Hình 2.10: Biểu đồ trình tự luồng cơ bản use case Chấm công 49](#_Toc198656462)

[Hình 2.11: Biểu đồ lớp phân tích use case Chấm công 50](#_Toc198656463)

[Hình 2.12: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử chấm công 51](#_Toc198656464)

[Hình 2.13: Biểu đồ lớp phân tích use case Xem lịch sử chấm công 52](#_Toc198656465)

[Hình 2.14: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử ngày công 53](#_Toc198656466)

[Hình 2.15: Biểu đồ lớp phân tích use case Xem lịch sử ngày công 54](#_Toc198656467)

[Hình 2.16: Biểu đồ trình tự luồng cơ bản use case Nhắn tin 55](#_Toc198656468)

[Hình 2.17: Biểu đồ lớp phân tích use case Nhắn tin 56](#_Toc198656469)

[Hình 2.18: Biểu đồ trình tự luồng cơ bản use case Tạo nhóm chat 57](#_Toc198656470)

[Hình 2.19: Biểu đồ lớp phân tích use case Tạo nhóm chat 58](#_Toc198656471)

[Hình 2.20: Biểu đồ trình tự luồng cơ bản use case Rời nhóm 59](#_Toc198656472)

[Hình 2.21: Biểu đồ lớp phân tích use case Rời nhóm 60](#_Toc198656473)

[Hình 2.22: Biểu đồ trình tự luồng cơ bản use case Thêm thành viên vào nhóm 61](#_Toc198656474)

[Hình 2.23: Biểu đồ lớp phân tích use case Thêm thành viên vào nhóm 62](#_Toc198656475)

[Hình 2.24: Biểu đồ trình tự luồng cơ bản use case Cập nhật thông tin 63](#_Toc198656476)

[Hình 2.25: Biểu đồ lớp phân tích use case Cập nhật thông tin 64](#_Toc198656477)

[Hình 2.26: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử chấm công, ngày công nhân viên 65](#_Toc198656478)

[Hình 2.27: Biểu đồ lớp phân tích use case Xem lịch sử chấm công, ngày công nhân viên 66](#_Toc198656479)

[Hình 2.28: Biểu đồ trình tự luồng cơ bản use case Quản lý thông tin nhân viên 67](#_Toc198656480)

[Hình 2.29: Biểu đồ trình tự luồng cơ bản use case Quản lý thông tin nhân viên 68](#_Toc198656481)

[Hình 3.1: Màn hình terminal chạy backend so sánh khuôn mặt thành công 70](#_Toc198656482)

[Hình 3.2: Màn hình terminal chạy backend quản lý chấm công và quản lý thông tin thành công 71](#_Toc198656483)

[Hình 3.3: Test postman API phát hiện khuôn mặt 72](#_Toc198656484)

[Hình 3.4: Test postman API so sánh khuôn mặt 72](#_Toc198656485)

[Hình 3.5: Test postman API đăng nhập 73](#_Toc198656486)

[Hình 3.6: Test postman API phát hiện đăng ký 73](#_Toc198656487)

[Hình 3.7: Test postman API chấm công 74](#_Toc198656488)

[Hình 3.8: Test postman API xem lịch sử chấm công 74](#_Toc198656489)

[Hình 3.9: Test postman API xem lịch sử ngày công 75](#_Toc198656490)

[Hình 3.10: Test postman API lấy tin nhắn 75](#_Toc198656491)

[Hình 3.11: Test postman API tạo nhóm 76](#_Toc198656492)

# Lời mở đầu

Trong kỷ nguyên công nghệ số, cùng với sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI) và thị giác máy tính, các doanh nghiệp đang ngày càng chú trọng đến việc ứng dụng công nghệ để tối ưu hoá quy trình quản lý và vận hành. Một trong những lĩnh vực được quan tâm hàng đầu là quản lý nhân sự, đặc biệt là việc chấm công – một công việc tưởng chừng đơn giản nhưng lại ảnh hưởng trực tiếp đến năng suất lao động, chi phí vận hành và sự minh bạch trong quản lý nhân viên.

Trong bối cảnh đó, hệ thống chấm công sử dụng công nghệ nhận diện khuôn mặt nổi lên như một giải pháp hiện đại, hiệu quả và phù hợp với xu hướng chuyển đổi số. Việc ứng dụng nhận diện khuôn mặt không chỉ giúp loại bỏ các hình thức chấm công truyền thống kém hiệu quả như dùng thẻ từ, giấy bút hay vân tay mà còn nâng cao độ chính xác, giảm gian lận, đồng thời mang lại trải nghiệm thuận tiện và chuyên nghiệp cho nhân viên.

Đề tài này nhằm mục tiêu giới thiệu và nghiên cứu việc xây dựng một ứng dụng chấm công sử dụng nhận diện khuôn mặt dành cho doanh nghiệp. Chúng ta sẽ cùng tìm hiểu những thách thức mà các phương pháp chấm công hiện tại đang gặp phải, đồng thời phân tích cách hệ thống này có thể khắc phục hạn chế, tối ưu hóa quy trình làm việc, từ việc nhận diện khuôn mặt, ghi nhận thời gian làm việc cho đến việc tổng hợp dữ liệu và báo cáo.

Trong các phần tiếp theo, chúng ta sẽ đi sâu vào tầm quan trọng của hệ thống chấm công hiện đại và những lợi ích thiết thực mà nó mang lại cho cả doanh nghiệp và người lao động.

# Lý do chọn đề tài

Ngày nay, công nghệ thông tin đóng vai trò then chốt trong công cuộc hiện đại hóa và phát triển của các quốc gia, là yếu tố cốt lõi thúc đẩy quá trình chuyển đổi số trong mọi lĩnh vực, đặc biệt là trong quản lý và vận hành doanh nghiệp. Bước vào thời đại công nghiệp 4.0, những thành tựu vượt bậc của trí tuệ nhân tạo (AI), nhận diện hình ảnh và học máy (machine learning) đang ngày càng được ứng dụng sâu rộng vào đời sống thực tiễn, từ sản xuất, kinh doanh đến các hoạt động quản trị nhân sự.

Trong bối cảnh các doanh nghiệp ngày càng mở rộng quy mô và gia tăng số lượng nhân viên, việc quản lý thời gian làm việc trở nên phức tạp và dễ xảy ra sai sót nếu vẫn áp dụng các hình thức chấm công thủ công như giấy tờ, thẻ từ hay máy vân tay. Những phương pháp này không chỉ thiếu tính linh hoạt mà còn tiềm ẩn nhiều rủi ro về gian lận, bảo mật và chi phí bảo trì thiết bị.

Trước nhu cầu cấp thiết đó, ứng dụng chấm công sử dụng nhận diện khuôn mặt trở thành một giải pháp tối ưu, vừa nâng cao hiệu quả quản lý, vừa đảm bảo tính chính xác, bảo mật và tiết kiệm thời gian cho cả doanh nghiệp và người lao động.

Với tư cách là một sinh viên khoa Công nghệ thông tin, em nhận thấy việc ứng dụng công nghệ nhận diện khuôn mặt vào hệ thống chấm công sẽ là một hướng đi thiết thực và phù hợp với xu thế chuyển đổi số hiện nay. Do đó, em quyết định chọn đề tài “Xây dựng ứng dụng chấm công sử dụng nhận diện khuôn mặt cho doanh nghiệp” để thực hiện đồ án của mình. Thông qua đề tài này, em mong muốn nghiên cứu và xây dựng một hệ thống hiện đại, mang tính ứng dụng cao, góp phần giải quyết bài toán thực tiễn trong quản lý nhân sự tại các doanh nghiệp hiện nay.

# Đặt vấn đề

Mức độ phát triển kinh tế – xã hội tại Việt Nam ngày càng cao kéo theo sự gia tăng nhu cầu về hiệu quả quản lý, đặc biệt là trong các doanh nghiệp vừa và lớn. Việc số hóa các hoạt động trong doanh nghiệp đang trở thành xu hướng tất yếu để nâng cao năng suất lao động, giảm thiểu chi phí vận hành và đảm bảo tính minh bạch trong công tác quản lý nhân sự. Trong đó, chấm công, một khâu quan trọng nhưng thường bị xem nhẹ lại là nơi phát sinh nhiều vấn đề như gian lận, sai sót hoặc thao tác rườm rà làm ảnh hưởng đến cả nhân viên lẫn bộ phận quản trị.

Hiện nay, phần lớn các doanh nghiệp vẫn sử dụng các hình thức chấm công truyền thống như ký tên thủ công, quẹt thẻ hoặc chấm công vân tay. Tuy nhiên, các hình thức này bộc lộ nhiều hạn chế: thiếu tính linh hoạt, dễ bị lợi dụng (như chấm công hộ), không đảm bảo vệ sinh (đặc biệt trong giai đoạn dịch bệnh), và gây tốn kém chi phí đầu tư bảo trì hệ thống.

Trong bối cảnh đó, công nghệ nhận diện khuôn mặt nổi lên như một giải pháp hiện đại và tối ưu. Không chỉ giúp nhận diện nhanh chóng và chính xác thời điểm làm việc của từng nhân viên, hệ thống còn giúp loại bỏ hoàn toàn việc gian lận chấm công, đồng thời mang lại trải nghiệm thuận tiện, không tiếp xúc và tự động hóa quy trình. Tuy nhiên, các hệ thống hiện tại thường khá đắt đỏ, thiếu tùy biến hoặc chưa được thiết kế phù hợp với quy mô và nhu cầu của doanh nghiệp vừa và nhỏ tại Việt Nam.

Từ những vấn đề trên, em nhận thấy việc xây dựng một ứng dụng chấm công sử dụng nhận diện khuôn mặt đơn giản, hiệu quả, chi phí hợp lý và dễ dàng triển khai tại doanh nghiệp là vô cùng cần thiết. Đây chính là lý do đồ án này được hình thành, với mong muốn thiết kế và triển khai một giải pháp chấm công thông minh, bắt kịp xu hướng chuyển đổi số, đồng thời giải quyết những tồn tại thực tế trong công tác quản lý nhân sự của doanh nghiệp hiện nay.

# Ý nghĩa và cơ sở lý luận

**Ý nghĩa của đề tài**

Nâng cao hiệu quả quản lý nhân sự: Hệ thống giúp doanh nghiệp tối ưu hóa quá trình chấm công nhờ vào công nghệ nhận diện khuôn mặt, đảm bảo tính chính xác và minh bạch trong việc ghi nhận thời gian làm việc của nhân viên.

Tự động hóa quy trình và tiết kiệm nguồn lực: Việc sử dụng nhận diện khuôn mặt loại bỏ các thao tác thủ công như quẹt thẻ hoặc ghi tay, giúp giảm thiểu sai sót và tiết kiệm thời gian cho cả nhân viên và người quản lý.

Tăng cường trải nghiệm người dùng: Ứng dụng cung cấp giao diện dễ sử dụng trên nền tảng di động, đồng thời hỗ trợ tính năng nhắn tin nhóm và cá nhân theo thời gian thực, tạo môi trường làm việc chuyên nghiệp và kết nối hiệu quả giữa các thành viên trong tổ chức.

Ứng dụng công nghệ tiên tiến vào thực tiễn: Đề tài là minh chứng rõ ràng cho việc ứng dụng các công nghệ hiện đại như nhận diện khuôn mặt (face recognition), Socket.IO, Android Kotlin… vào giải quyết bài toán thực tế trong doanh nghiệp.

Góp phần thúc đẩy chuyển đổi số: Việc triển khai hệ thống này không chỉ giúp doanh nghiệp nâng cao năng suất mà còn là bước tiến trong hành trình chuyển đổi số, phù hợp với xu hướng phát triển công nghệ 4.0 hiện nay.

**Cơ sở lý luận của đề tài**

**Quản lý chấm công trong doanh nghiệp:** Hệ thống dựa trên nguyên tắc cần thiết trong quản lý thời gian và nhân sự – đó là ghi nhận chính xác, minh bạch và nhanh chóng thời gian làm việc của nhân viên. Các phương pháp chấm công truyền thống hiện nay bộc lộ nhiều hạn chế, và việc ứng dụng công nghệ nhận diện khuôn mặt là hướng đi mới giúp giải quyết các tồn tại này.

**Công nghệ nhận diện khuôn mặt:** Đề tài áp dụng thư viện face\_recognition của Python để xử lý và so sánh khuôn mặt nhằm đảm bảo độ chính xác và an toàn trong quá trình chấm công. Đây là một trong những công nghệ sinh trắc học tiên tiến, có độ bảo mật cao và giảm thiểu tình trạng chấm công hộ.

**Công nghệ thời gian thực và giao tiếp:** Với việc tích hợp Socket.IO, hệ thống hỗ trợ nhắn tin nhóm và cá nhân theo thời gian thực, giúp tăng cường khả năng giao tiếp nội bộ trong doanh nghiệp, nâng cao tính tương tác và hỗ trợ xử lý công việc nhanh chóng. Kiến trúc hệ thống hiện đại và linh hoạt: Hệ thống được xây dựng trên nền tảng Android với Kotlin, kết nối tới backend Node.js và Python Fast API, kết hợp với cơ sở dữ liệu và API, cho phép mở rộng dễ dàng và tích hợp vào hệ thống quản lý hiện có của doanh nghiệp.

**Công nghệ thông tin và phát triển phần mềm:** Việc triển khai ứng dụng trên nền tảng di động giúp người dùng dễ dàng truy cập mọi lúc mọi nơi, đồng thời hệ thống backend với các công nghệ như FastAPI và Node.js đảm bảo tính ổn định, hiệu suất và dễ bảo trì.

# Mục tiêu và phạm vi đồ án

**Mục tiêu của đồ án**

* Tìm hiểu và ứng dụng các công nghệ hiện đại: Nghiên cứu, lựa chọn và triển khai các công nghệ như Android Kotlin, Python FastAPI, Node.js, Socket.IO, MongoDB nhằm xây dựng hệ thống chấm công có tính ổn định, bảo mật và mở rộng cao.
* Tìm hiểu cấu trúc và quy trình của hệ thống chấm công hiện đại: Nắm rõ quy trình hoạt động của một hệ thống chấm công sử dụng công nghệ nhận diện khuôn mặt, từ việc đăng ký khuôn mặt, chấm công, lưu trữ đến quản lý và thống kê dữ liệu.
* Phân tích và thiết kế hệ thống chấm công: Tiến hành phân tích yêu cầu thực tế của doanh nghiệp để thiết kế hệ thống bao gồm các thành phần chính như: chấm công bằng khuôn mặt, quản lý nhân sự, quản lý lịch sử chấm công, nhắn tin nội bộ theo thời gian thực.
* Xây dựng và triển khai hệ thống ứng dụng chấm công: Hoàn thiện hệ thống ứng dụng chấm công hoạt động đa nền tảng (mobile và web) sử dụng các công nghệ đã nghiên cứu, đảm bảo tính khả thi, thân thiện với người dùng và đáp ứng tốt yêu cầu nghiệp vụ thực tế của doanh nghiệp.

**Phạm vi đồ án**

* Nghiên cứu, khảo sát và thu thập yêu cầu thực tế từ các doanh nghiệp vừa và nhỏ đang sử dụng hoặc có nhu cầu xây dựng hệ thống chấm công.
* Xây dựng hệ thống bao gồm các chức năng: đăng ký khuôn mặt, chấm công bằng nhận diện khuôn mặt, quản lý thông tin nhân viên, thống kê giờ công, nhắn tin nhóm và cá nhân theo thời gian thực.
* Sử dụng Android Kotlin cho ứng dụng di động, Python FastAPI và Node.js cho backend, MongoDB cho lưu trữ dữ liệu, Socket.IO để truyền dữ liệu thời gian thực.
* Kiểm thử và triển khai hệ thống trong môi trường mạng nội bộ hoặc máy chủ demo, nhằm đánh giá khả năng hoạt động thực tế.

# : TỔNG QUAN

## Tổng quan về các hình thức chấm công của doanh nghiệp.

Trong bối cảnh nền kinh tế Việt Nam ngày càng phát triển và hội nhập sâu rộng, việc quản lý nguồn nhân lực hiệu quả đóng vai trò then chốt trong sự thành công của các doanh nghiệp. Một trong những yếu tố quan trọng trong quản lý nhân sự là hệ thống chấm công, công cụ giúp theo dõi và ghi nhận thời gian làm việc của nhân viên một cách chính xác và minh bạch. Tại Việt Nam, các doanh nghiệp đã và đang áp dụng nhiều hình thức chấm công khác nhau, từ truyền thống đến hiện đại. Các phương pháp phổ biến bao gồm: chấm công bằng thẻ từ, chấm công bằng vân tay, chấm công bằng phần mềm GPS, chấm công bằng nhận diện khuôn mặt. Mỗi phương pháp chấm công đều có những ưu nhược điểm riêng, và việc lựa chọn hình thức phù hợp phụ thuộc vào quy mô, nhu cầu và khả năng tài chính của từng doanh nghiệp. Trong xu thế chuyển đổi số hiện nay, việc áp dụng các công nghệ tiên tiến như trí tuệ nhân tạo (AI) trong hệ thống chấm công đang trở thành một hướng đi mới, hứa hẹn mang lại hiệu quả cao hơn trong quản lý nhân sự.

### Chấm công bằng thẻ từ

* Ưu điểm:

+ Tốc độ chấm công nhanh chóng

+ Dễ dàng triển khai và sử dụng

+ Chi phí đầu tư ban đầu thấp

+ Có thể tích hợp với phần mềm quản lý nhân sự

* Nhược điểm:

+ Dễ xảy ra tình trạng chấm công hộ nếu không có biện pháp kiểm soát.

+ Thẻ từ có thể mất, quên, hỏng gây bất tiện cho nhân viên.

+ Không phù hợp với môi trường có tính bảo mật cao.

### Chấm công bằng vân tay

* Ưu điểm:

+ Độ chính xác cao, khó gian lận chấm công

+ Không cần mang thẻ, giảm thiểu tình trạng quên mất

+ Chi phí vận hành thấp sau khi đầu tư thiết bị

+ Dễ dàng tích hợp với phần mềm quản lý nhân sự

* Nhược điểm:

+ Có thể gặp khó khăn khi chấm công trong các trường hợp tay ướt, bẩn, bị thương.v.v

+ Không phù hợp với môi trường làm việc nhiều bụi và yêu cầu vệ sinh cao

+ Thiết bị xuống cấp theo thời gian cần bảo trì định kỳ

### Chấm công bằng phần mềm GPS

* Ưu điểm

+ Chấm công linh hoạt, không cần thiết bị cố định, phù hợp với nhân viên làm việc ở nhiều địa điểm

+ Quản lý thời gian và vị trí làm việc hiệu quả

+ Tiết kiệm chi phí, thời gian

+ Tích hợp nhiều tính năng khác

* Nhược điểm:

+ Phụ thuộc vào thiết bị di động và kết nối internet

+ Nguy cơ gian lận vị trí chấm công

+ Độ chính xác GPS có thể bị ảnh hưởng.

### Chấm công sử dụng khuôn mặt

* Ưu điểm:

+ Độ chính xác cao, không thể gian lận

+ Khắc phục được hạn chế bởi các phương pháp thẻ từ và vân tay

+ Dễ dàng tích hợp với hệ thống quản lý nhân sự

* Nhược điểm:

+ Hiệu suất nhận diện có thể bị ảnh hưởng bởi các yếu tố ánh sáng, thay đổi ngoại hình

+ Cần đảm bảo quyền riêng tư và bảo mật dữ liệu khuôn mặt

## Cơ sở lý thuyết về nhận diện và so sánh khuôn mặt

### Khái niệm và vai trò

Nhận diện khuôn mặt (Face Recognition) là một lĩnh vực trong thị giác máy tính và trí tuệ nhân tạo, nhằm xác định hoặc xác minh danh tính của một cá nhân dựa trên hình ảnh hoặc video chứa khuôn mặt của họ. Công nghệ này được ứng dụng rộng rãi trong nhiều lĩnh vực như an ninh, giám sát, thanh toán điện tử, và đặc biệt là trong hệ thống chấm công tự động tại doanh nghiệp.

### Quy trình nhận diện khuôn mặt

Quá trình nhận diện khuôn mặt thường bao gồm các bước chính sau:

* Phát hiện khuôn mặt (Face Detection): Xác định vị trí của khuôn mặt trong hình ảnh hoặc video.
* Căn chỉnh khuôn mặt (Face Alignment): Điều chỉnh khuôn mặt để chuẩn hóa về kích thước, góc nhìn và ánh sáng, nhằm tăng độ chính xác trong các bước tiếp theo.
* Trích xuất đặc trưng (Feature Extraction): Rút trích các đặc điểm quan trọng của khuôn mặt như khoảng cách giữa các điểm đặc trưng (mắt, mũi, miệng) để tạo ra một vector đặc trưng đại diện cho khuôn mặt đó.
* So sánh và nhận diện (Face Matching): So sánh vector đặc trưng của khuôn mặt mới với các vector đã lưu trữ trong cơ sở dữ liệu để xác định hoặc xác minh danh tính.

### Các phương pháp nhận diện khuôn mặt

Các phương pháp nhận diện khuôn mặt có thể được phân loại thành hai nhóm chính:

* Phương pháp truyền thống: Sử dụng các kỹ thuật như phân tích thành phần chính (PCA), phân tích phân biệt tuyến tính (LDA), và mô hình Markov ẩn (HMM) để trích xuất và so sánh đặc trưng khuôn mặt.
* Phương pháp học sâu (Deep Learning): Sử dụng các mạng nơ-ron tích chập (CNN) để tự động học các đặc trưng khuôn mặt từ dữ liệu lớn, mang lại độ chính xác cao hơn trong điều kiện thực tế.

## Lựa chọn công nghệ

### Python FastAPI

FastAPI là một framework hiện đại và mạnh mẽ dành cho việc xây dựng các ứng dụng web và API trong Python. Được thiết kế để tối ưu hóa cả hiệu suất lẫn trải nghiệm lập trình viên, FastAPI giúp bạn xây dựng các dịch vụ API nhanh chóng, an toàn và dễ bảo trì. Đây là một công cụ mã nguồn mở được phát triển bởi Sebastián Ramírez, và hiện đang được cộng đồng phát triển mạnh mẽ và sử dụng rộng rãi trong các công ty lớn như Microsoft, Uber, Netflix, và nhiều tổ chức khác. Những điểm nổi bật của FastAPI: Hiệu năng vượt trội FastAPI được xây dựng trên Starlette (một framework ASGI) cho phần xử lý web và Pydantic cho việc kiểm tra dữ liệu và serial hóa. Nhờ sử dụng async/await, FastAPI hỗ trợ xử lý bất đồng bộ rất hiệu quả, giúp cải thiện hiệu suất và khả năng mở rộng, đặc biệt phù hợp với các ứng dụng có lượng người dùng lớn hoặc cần xử lý song song nhiều tác vụ. Tự động sinh tài liệu API FastAPI tự động sinh tài liệu API tương tác theo chuẩn OpenAPI (Swagger) và ReDoc. Điều này giúp lập trình viên dễ dàng kiểm thử và hiểu rõ các endpoint của hệ thống mà không cần viết tài liệu thủ công. Kiểm tra dữ liệu mạnh mẽ Thông qua Pydantic, FastAPI cho phép bạn xác thực và kiểm tra kiểu dữ liệu đầu vào của request một cách tự động, giúp phát hiện lỗi sớm và giảm thiểu rủi ro do dữ liệu không hợp lệ. Hỗ trợ đầy đủ type hints của Python FastAPI tận dụng tối đa khả năng khai báo kiểu dữ liệu (type hints) trong Python để giúp editor như VSCode hỗ trợ auto-complete, kiểm tra lỗi tĩnh và tài liệu hóa code tốt hơn. Thân thiện với lập trình viên Với thiết kế hướng developer, FastAPI giúp giảm thiểu boilerplate code (mã lặp lại) và tăng tốc độ phát triển. Dù là người mới học làm web hay lập trình viên backend dày dặn kinh nghiệm, FastAPI đều mang lại trải nghiệm dễ chịu và hiệu quả. Dễ tích hợp và mở rộng FastAPI hoạt động tốt với các thư viện phổ biến như SQLAlchemy, MongoDB, OAuth2, JWT, và có thể dễ dàng tích hợp với Docker, Kubernetes và CI/CD pipelines để triển khai thực tế.

### Thư viện Face Recognition

Thư viện Face Recognition là một thư viện mã nguồn mở mạnh mẽ được xây dựng trên nền tảng của thư viện dlib, vốn nổi tiếng trong lĩnh vực thị giác máy tính (computer vision) và học máy (machine learning). Đây là một trong những công cụ phổ biến nhất hiện nay trong việc nhận diện và so sánh khuôn mặt, nhờ vào tính dễ sử dụng, độ chính xác cao và khả năng triển khai linh hoạt. Thư viện này cho phép thực hiện nhiều tác vụ xử lý khuôn mặt như:

* Phát hiện khuôn mặt (face detection)
* Xác định các điểm đặc trưng trên khuôn mặt (face landmarks)
* Mã hóa khuôn mặt (face encoding)
* So sánh và nhận dạng khuôn mặt (face recognition & comparison)

Nguyên lý hoạt động:

- Phát hiện khuôn mặt (Face Detection) Bước đầu tiên trong quá trình nhận diện khuôn mặt là phát hiện vị trí các khuôn mặt trong ảnh. Face Recognition sử dụng mô hình HOG (Histogram of Oriented Gradients) hoặc CNN (Convolutional Neural Network) để tìm ra khuôn mặt trong ảnh. HOG là phương pháp nhẹ, nhanh, phù hợp cho thiết bị cấu hình vừa phải. CNN cho độ chính xác cao hơn nhưng đòi hỏi tài nguyên tính toán nhiều hơn.

- Xác định các điểm đặc trưng (Facial Landmarks Detection) Sau khi phát hiện được khuôn mặt, thư viện sẽ sử dụng mô hình học máy để xác định 68 điểm đặc trưng trên khuôn mặt như mắt, mũi, miệng, cằm,... Các điểm này giúp hiệu chỉnh (align) khuôn mặt về trạng thái chuẩn, giúp việc nhận diện chính xác hơn.

- Mã hóa khuôn mặt (Face Encoding) Tiếp theo, khuôn mặt được chuyển đổi thành một vector đặc trưng (128 chiều) thông qua mạng nơ-ron sâu. Vector này mang tính duy nhất đối với từng người – giống như một "dấu vân tay" của khuôn mặt – và là đầu vào cho bước so sánh.

- So sánh và nhận diện (Face Comparison) Khi cần xác định xem một khuôn mặt mới có khớp với khuôn mặt nào đã lưu trữ hay không, thư viện sẽ tính khoảng cách Euclidean giữa vector của khuôn mặt mới và vector của các khuôn mặt đã lưu. Nếu khoảng cách này nhỏ hơn một ngưỡng nhất định (thường là 0.6), thì hai khuôn mặt được coi là giống nhau.

* Ưu điểm của thư viện Face Recognition:

+ Dễ sử dụng: Chỉ cần vài dòng mã Python là đã có thể tích hợp tính năng nhận diện khuôn mặt.

+ Chính xác cao: Dựa trên mô hình ResNet được huấn luyện sẵn trên tập dữ liệu lớn.

+ Mã nguồn mở: Cho phép tùy chỉnh và triển khai dễ dàng trên các hệ thống nội bộ.

+ Tích hợp tốt với FastAPI: Phù hợp khi xây dựng các API nhận diện khuôn mặt cho hệ thống chấm công.

* Nhược điểm:

+ Phụ thuộc phần cứng:

+ Khi sử dụng phương pháp CNN, cần có GPU hoặc CPU mạnh để xử lý nhanh.

+ Không phù hợp cho dữ liệu quy mô cực lớn: Vì hiệu suất sẽ bị ảnh hưởng nếu không có cơ chế tối ưu (ví dụ như indexing hay pre-caching vector).

# : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Các tác nhân của hệ thống

* Tác nhân nhân viên: là những người nhân viên làm việc trong công ty
* Tác nhân trưởng phòng: là những người giữ chức vụ trưởng phòng các phòng ban trong công ty.
* Tác nhân giám đốc: là những người giữ chức vụ trong ban giám đốc của công ty.
* Tác nhân admin: là người quản lý thông tin hệ thống.

## Hoạt động của hệ thống

### Hoạt động chấm công

Tất cả người dùng có thể thực hiện chấm công bao gồm check-in, check-out, xem lại lịch sử chấm công theo ngày, xem lịch sử ngày công theo tháng và chi tiết ngày công.

### Hoạt động quản lý thông tin

* Nhân viên: Xem thông tin cá nhân, thay đổi mật khẩu, cập nhật thông tin cá nhân
* Trưởng phòng: Xem thông tin cá nhân, thay đổi mật khẩu, cập nhật thông tin cá nhân, cập nhật trạng thái, thông tin của nhân viên trong phòng ban mình quản lý.
* Giám đốc: Xem thông tin cá nhân, thay đổi mật khẩu, cập nhật thông tin cá nhân, cập nhật trạng thái, thông tin của nhân viên trong công ty.
* Admin: Quản lý thông tin nhân viên, xem thông tin chấm công, ngày công, xuất file báo cáo.

### Hoạt động nhắn tin thời gian thực

Người dùng có thể nhắn tin nhóm, nhắn tin riêng theo thời gian thực. Ngoài ra người dùng có thể xem người dùng online, tạo nhóm, thêm người vào nhóm, rời nhóm.

## Các yêu cầu chức năng

### Phía nhân viên

* Đăng ký, đăng nhập, đăng xuất
* Chấm công
* Quản lý lịch sử chấm công, ngày công
* Quản lý thông tin cá nhân
* Đổi mật khẩu
* Nhắn tin thời gian thực
* Tạo nhóm, rời nhóm, thêm người vào nhóm

### Phía trưởng phòng, giám đốc

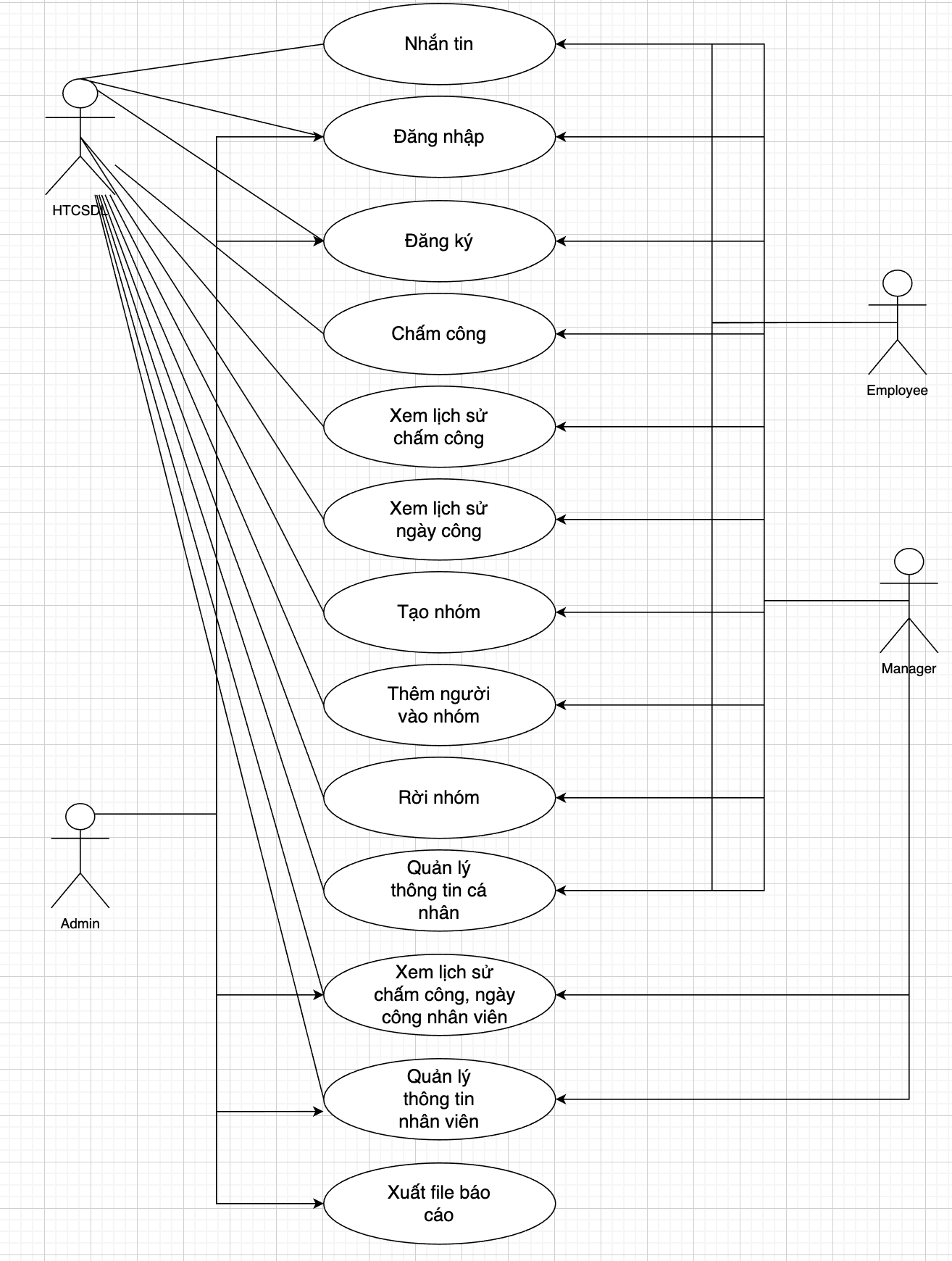
* Đăng ký, đăng nhập, đăng xuất
* Chấm công
* Quản lý lịch sử chấm công, ngày công
* Quản lý thông tin cá nhân
* Đổi mật khẩu
* Nhắn tin thời gian thực
* Tạo nhóm, rời nhóm, thêm người vào nhóm
* Quản lý thông tin, chấm công, ngày công của nhân viên trong phòng ban/ công ty

### Phía admin

* Đăng nhập, đăng ký, đăng xuất.
* Quản lý thông tin nhân viên.
* Xem lịch sử chấm công, ngày công.
* Xuất file báo cáo

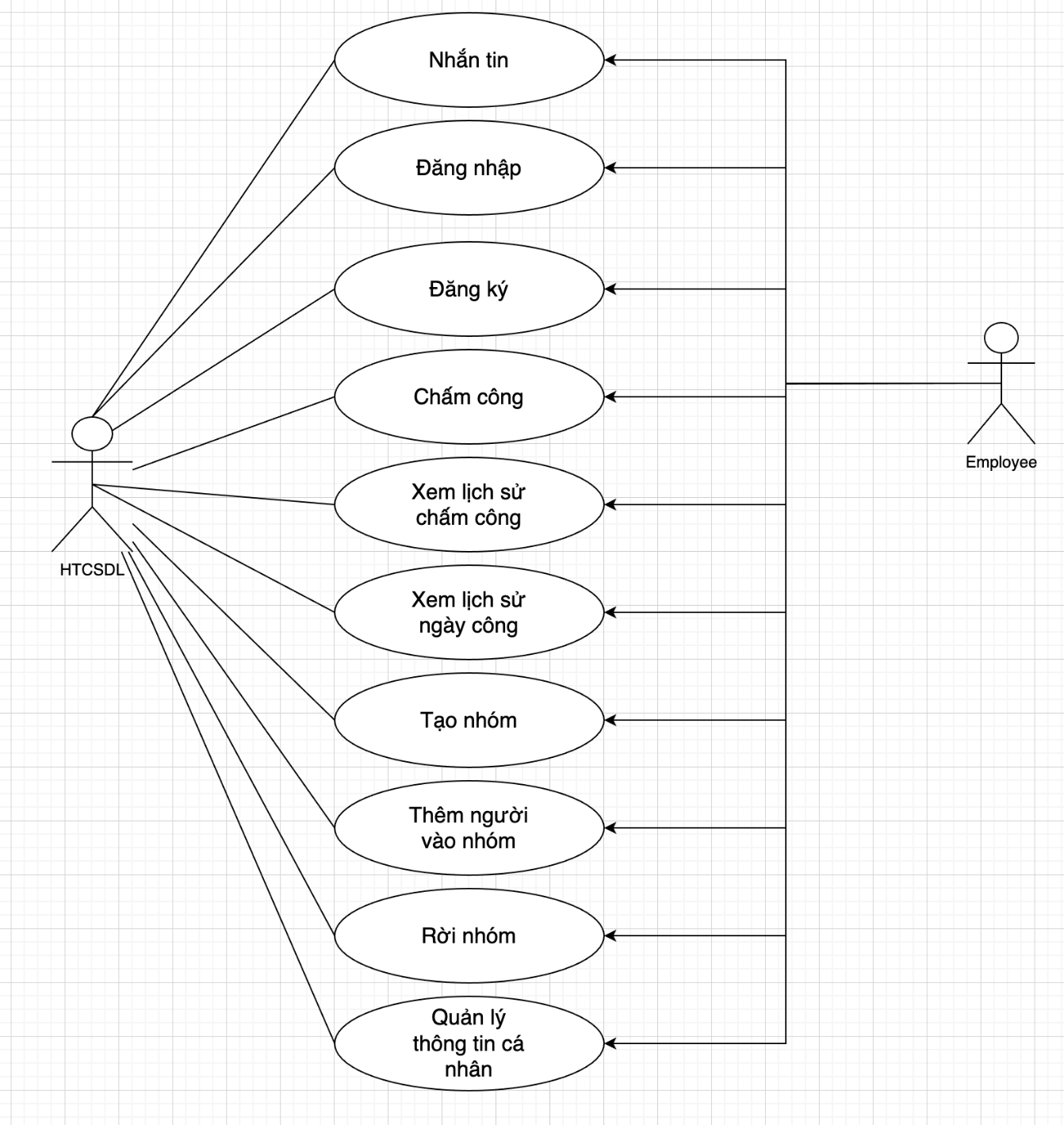
## Biểu đồ use case

### Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

### Biểu đồ phân rã use case cho nhân viên



Hình 2.2: Biểu đồ phân rã use case cho nhân viên

Use case đăng nhập cho phép nhân viên truy cập vào hệ thống bằng tài khoản đã đăng ký và được duyệt.

Use case đăng ký cho phép nhân viên đăng ký tài khoản để truy cập vào hệ thống bằng cách nhập các thông tin như họ tên, email, mật khẩu, số điện thoại, ngày sinh, địa chỉ, vị trí, phòng ban.

Use case chấm công cho phép nhân viên điểm danh chấm công (check in, check out) sử dụng khuôn mặt tại thời điểm chấm công.

Use case xem lịch sử chấm công cho phép nhân viên xem lại lịch sử những lần chấm công trước đó theo ngày hoặc xem tất cả.

Use case xem lịch sử ngày công cho phép nhân viên xem lại lịch sử chi tiết các ngày công trong tháng.

Use case nhắn tin cho phép nhân viên nhắn tin theo thời gian thực theo nhóm hoặc cá nhân.

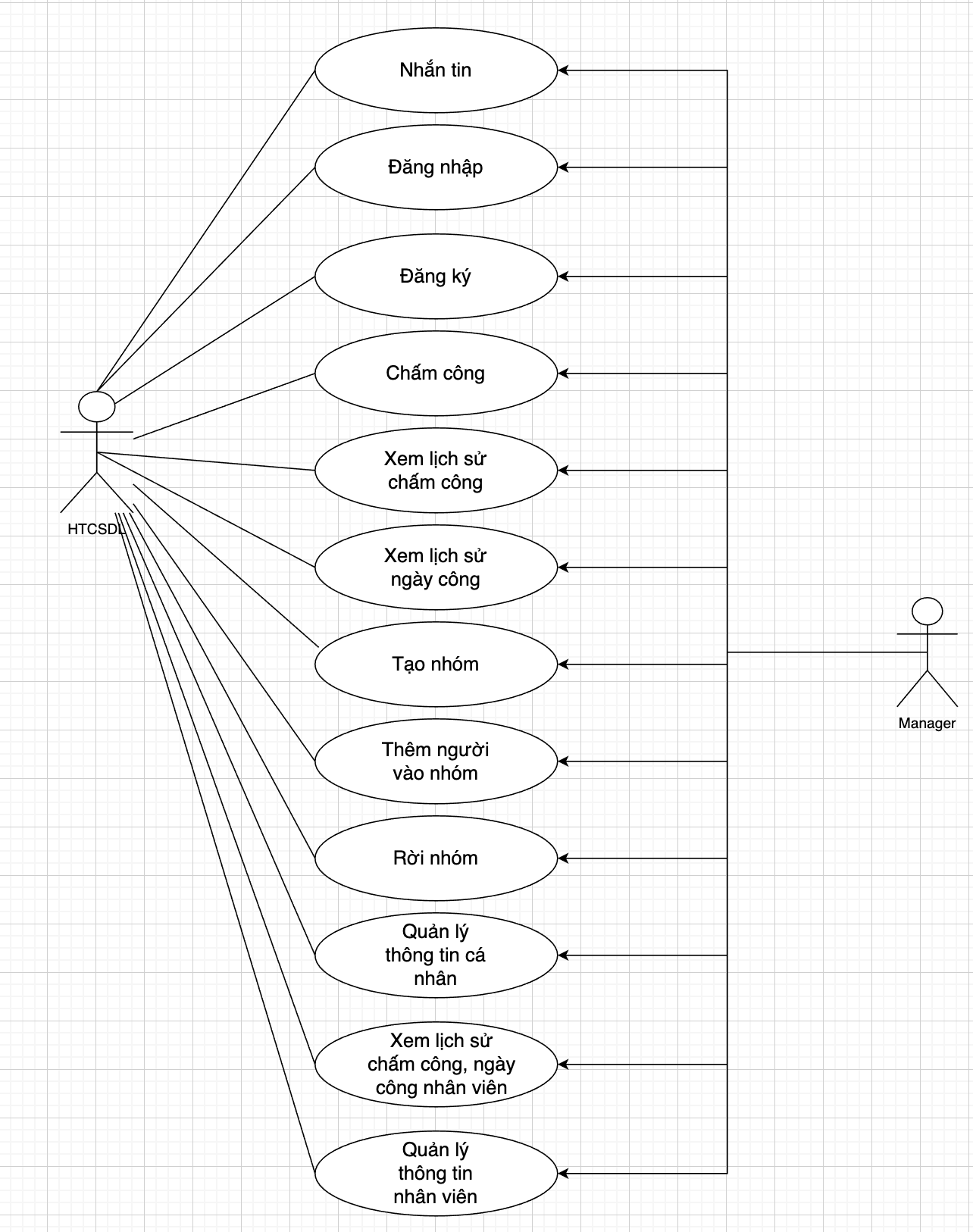
Use case tạo nhóm cho phép nhân viên tạo nhóm chat bằng cách tạo tên nhóm và chọn bất kì ai trong công ty.

Use case rời nhóm cho phép nhân viên rời nhóm chat khi nhân viên không muốn ở trong nhóm chat đó nữa.

Use case thêm người vào nhóm cho phép nhân viên thêm bất kì ai trong công ty vào nhóm chat.

Use case quản lý thông tin cho phép nhân viên xem, thay đổi thông tin cá nhân, mật khẩu của mình.

### Biểu đồ phân rã use case cho trưởng phòng, giám đốc



Hình 2.3: Biểu đồ phân rã use case cho quản lí (trưởng phòng, giám đốc)

Use case đăng nhập cho phép trưởng phòng, giám đốc truy cập vào hệ thống bằng tài khoản đã đăng ký và được duyệt.

Use case đăng ký cho phép trưởng phòng, giám đốc đăng ký tài khoản để truy cập vào hệ thống bằng cách nhập các thông tin như họ tên, email, mật khẩu, số điện thoại, ngày sinh, địa chỉ, vị trí, phòng ban.

Use case chấm công cho phép trưởng phòng, giám đốc điểm danh chấm công (check in, check out) sử dụng khuôn mặt tại thời điểm chấm công.

Use case xem lịch sử chấm công cho phép trưởng phòng, giám đốc xem lại lịch sử những lần chấm công trước đó theo ngày hoặc xem tất cả.

Use case xem lịch sử ngày công cho phép trưởng phòng, giám đốc xem lại lịch sử chi tiết các ngày công trong tháng.

Use case nhắn tin cho phép trưởng phòng, giám đốc nhắn tin theo thời gian thực theo nhóm hoặc cá nhân.

Use case tạo nhóm cho phép trưởng phòng, giám đốc tạo nhóm chat bằng cách tạo tên nhóm và chọn bất kì ai trong công ty.

Use case rời nhóm cho phép trưởng phòng, giám đốc rời nhóm chat khi trưởng phòng, giám đốc không muốn ở trong nhóm chat đó nữa.

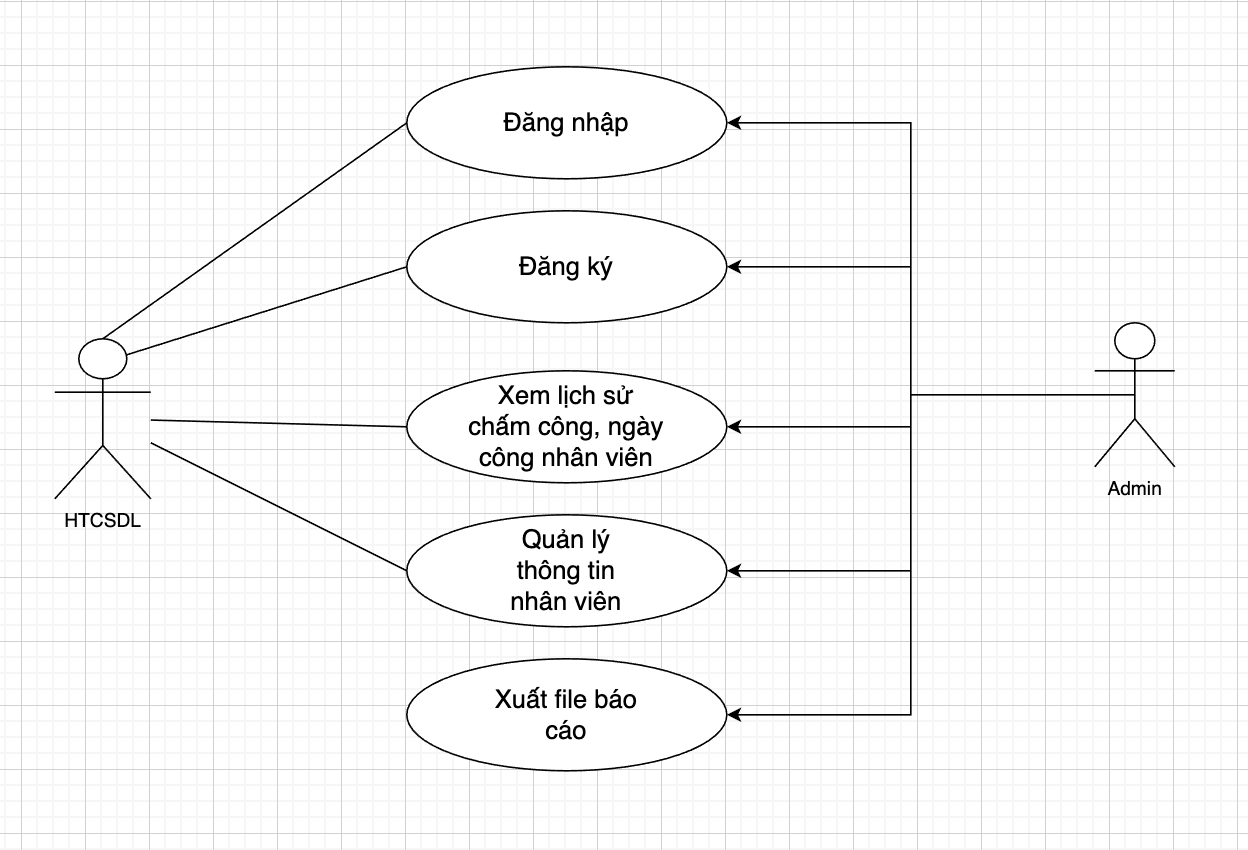
Use case thêm người vào nhóm cho phép trưởng phòng, giám đốc thêm bất kì ai trong công ty vào nhóm chat.

Use case quản lý thông tin cho phép trưởng phòng, giám đốc xem, thay đổi thông tin cá nhân, mật khẩu của mình.

Use case xem lịch sử chấm công, ngày công của nhân viên cho phép trưởng phòng, giám đốc xem lịch sử chấm công, ngày công của nhân viên trong phòng ban, toàn công ty.

Use case quản lý thông tin nhân viên cho phép trưởng phòng, giám đốc xem và chỉnh sửa thông tin, khóa, duyệt tài khoản của nhân viên trong phòng ban, toàn công ty.

### Biểu đồ phân rã use case cho admin



Hình 2.4: Biểu đồ phân rã use case cho quản lí (trưởng phòng, giám đốc)

Use case đăng nhập cho phép admin đăng nhập vào trang admin để quản lý hệ thống.

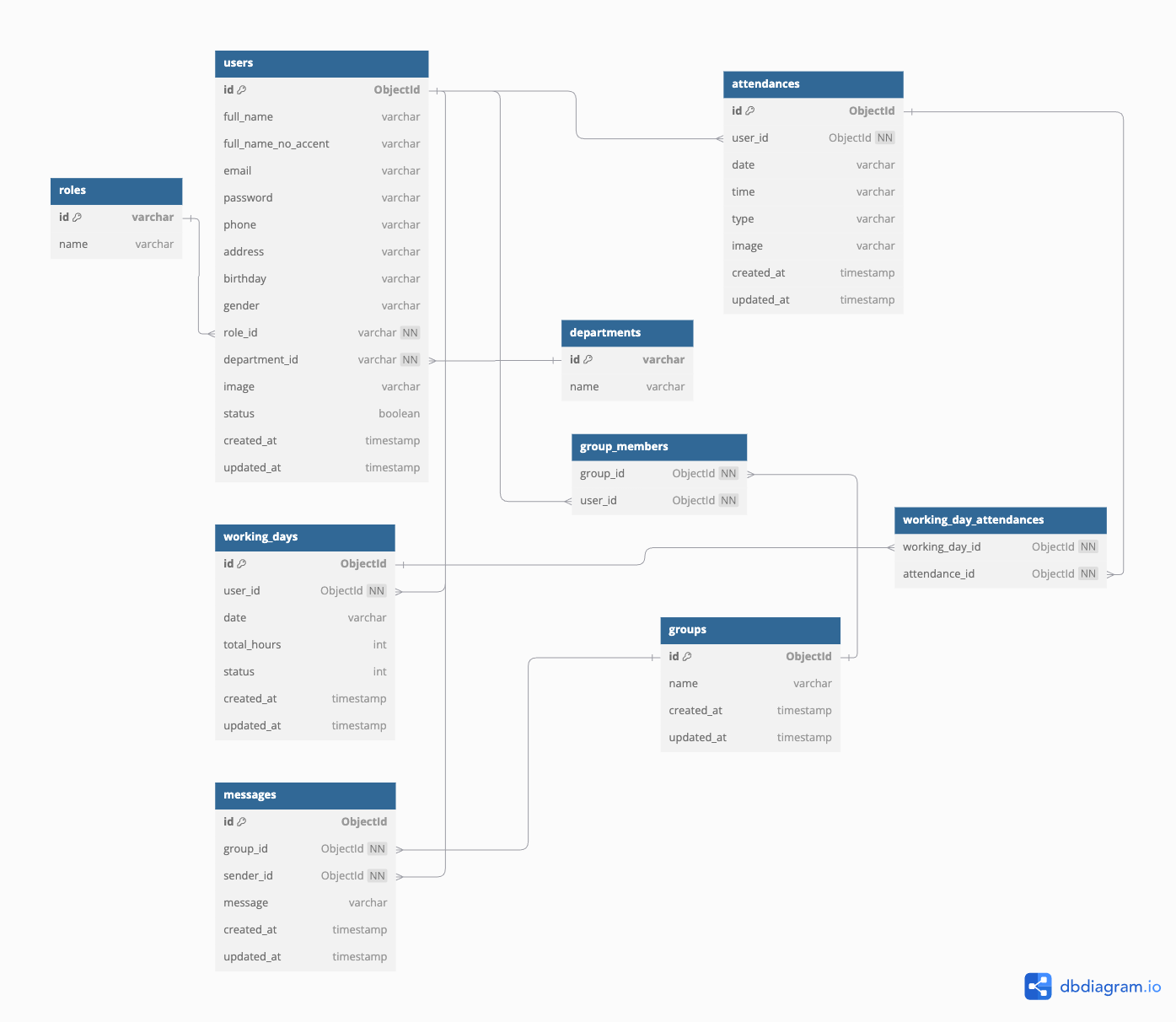
Use case đăng ký cho phép admin đăng ký tài khoản admin để truy cập vào trang admin.

Use case xem lịch sử chấm công, ngày công nhân viên cho phép admin xem toàn bộ lịch sử chấm công, ngày công của nhân viên toàn công ty.

Use case quản lý thông tin nhân viên cho phép admin thêm, xoá sửa thông tin toàn bộ nhân viên trong công ty.

Use case xuất file báo cáo cho phép admin xuất file báo cáo bao gồm danh sách nhân viên, danh sách ngày công, danh sách chấm công.

## Cơ sở dữ liệu



Hình 2.5: Cơ sở dữ liệu

Người dùng (Users): Lưu thông tin người dùng trong hệ thống, bao gồm các trường: mã người dùng, họ tên, họ tên không dấu, email, mật khẩu, số điện thoại, địa chỉ, ngày sinh, giới tính, mã vai trò, mã phòng ban, ảnh đại diện, trạng thái hoạt động, thời gian tạo và thời gian cập nhật. Một người dùng thuộc một phòng ban và một vai trò duy nhất. Một người dùng có thể thuộc nhiều nhóm, gửi nhiều tin nhắn, có nhiều ngày làm việc và điểm danh.

Vai trò (Roles): Chứa thông tin về vai trò của người dùng như quản trị viên, nhân viên,... Gồm mã vai trò và tên vai trò. Một vai trò có thể được gán cho nhiều người dùng.

Phòng ban (Departments): Lưu thông tin các phòng ban trong tổ chức, gồm mã phòng ban và tên phòng ban. Một phòng ban có thể có nhiều người dùng.

Nhóm (Groups): Lưu thông tin về các nhóm làm việc trong hệ thống, gồm mã nhóm, tên nhóm, thời gian tạo và thời gian cập nhật. Một nhóm có thể có nhiều thành viên và nhiều tin nhắn.

Thành viên nhóm (Group\_members): Bảng trung gian liên kết người dùng với nhóm. Gồm mã nhóm và mã người dùng. Một người dùng có thể tham gia nhiều nhóm.

Tin nhắn (Messages): Lưu các tin nhắn trong nhóm, gồm mã tin nhắn, mã nhóm, mã người gửi, nội dung tin nhắn, thời gian tạo và cập nhật. Mỗi tin nhắn thuộc một nhóm và được gửi bởi một người dùng.

Ngày làm việc (Working\_days): Lưu thông tin về ngày làm việc của người dùng, gồm mã ngày làm việc, mã người dùng, ngày, tổng số giờ làm, trạng thái, thời gian tạo và cập nhật. Một người dùng có thể có nhiều ngày làm việc.

Điểm danh (Attendances): Ghi lại thông tin điểm danh của người dùng gồm mã điểm danh, mã người dùng, ngày, thời gian, loại điểm danh (vào/ra), ảnh điểm danh, thời gian tạo và cập nhật. Một người dùng có thể có nhiều bản ghi điểm danh. Điểm danh trong ngày làm việc (Working\_day\_attendances): Bảng trung gian liên kết bản ghi ngày làm việc với bản ghi điểm danh, gồm mã ngày làm việc và mã điểm danh. Một ngày làm việc có thể liên kết với nhiều lần điểm danh.

## Phân tích chi tiết use case

### Use case đăng nhập

- Mô tả: Use case cho phép người dùng đăng nhập vào hệ thống

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng truy cập vào ứng dụng. Hệ thống hiển thị yêu cầu người dùng nhập email và mật khẩu.

2) Người dùng nhập email và mật khẩu sau đó nhấp vào nút Đăng nhập. Hệ thống xác nhận email và mật khẩu có hợp lệ không, nếu hợp lệ thì đăng nhập thành công hệ thống chuyển màn hình đến trang chủ. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản nếu người dùng không nhập tên hoặc mật khẩu, kiểu dữ liệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng cần nhập đúng thông tin để tiếp tục.

2) Tại bước 2 trong luồng cơ bản nếu kiểm tra tài khoản không có trong bảng User thì hệ thống sẽ hiển thị thông báo lỗi.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng đã có một tài khoản trong hệ thống.

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Use case đăng ký

- Mô tả: Use case cho phép người dùng đăng ký tài khoản truy cập ứng dụng

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp vào nút “Đăng ký”, ứng dụng di chuyển đến màn hình đăng ký

2) Người dùng nhập họ tên, email, mật khẩu, ngày sinh, địa chỉ, số điện thoại, ví trí, phòng ban, ngày sinh, giới tính sau đó nhấp vào nút “Đăng ký”. Nếu người dùng nhập thông tin hợp lệ thì hệ thống sẽ chuyển đến màn chụp ảnh đại diện.

3) Người dùng để khuôn mặt vào giữa màn hình rồi nhấn nút chụp. Hệ thống sẽ phân tích khuôn mặt nếu hợp lệ thì sẽ lưu lại khuôn mặt và di chuyển đến màn hình đăng nhập. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản nếu người dùng nhập thiếu hoặc không hợp lệ 1 trong các trường thì ứng dụng sẽ hiển thị thông báo lỗi cho người dùng, hoặc khi người dùng nhấn nút đăng ký, hệ thống sẽ báo lỗi nếu có.

2) Tại bước 3 trong luồng cơ bản, nếu như khuôn mặt không hợp lệ ví dụ như mờ, không rõ khuôn mặt,... hệ thống sẽ báo lỗi và người dùng phải chụp lại.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Use case chấm công

- Mô tả: Use case cho phép người dùng chấm công

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp vào nút “Chấm công” trên màn hình trang chủ, nếu vị trí hiện tại của bạn cách công ty 1 khoảng cách phù hợp thì sẽ di chuyển đến màn hình chấm công.

2) Người dùng đưa điện thoại để khuôn mặt ở chính giữa khung hình sau đó nhấn nút chụp, nếu khuôn mặt phù hợp thì sẽ hiển thị lịch sử chấm công trong ngày ở dưới. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 1 trong luồng cơ bản nếu người dùng đang ở vị trí xa hơn so với quy định thì sẽ không bấm được nút “Chấm công”.

2) Tại bước 2 nếu khuôn mặt không hợp lệ như không trùng khớp, không tìm thấy khuôn mặt thì ứng dụng sẽ hiển thị lỗi lên màn hình.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Use case xem lịch sử chấm công

- Mô tả: Use case cho phép người dùng xem lại lịch sử chấm công

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp icon “Lịch sử” trên thanh Navigation Bar và chọn Tab “Lịch sử chấm công”

2) Ứng dụng sẽ hiển thị tất cả lịch sử chấm công của người dùng. Use case kết thúc

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản, nếu người dùng ấn vào icon “ Lịch” rồi chọn ngày, hệ thống sẽ hiển thị tất cả lịch sử chấm công của người dùng trong ngày đã chọn.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case xem lịch sử ngày công

- Mô tả: Use case cho phép người dùng xem lại lịch sử ngày công

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp icon “Lịch sử” trên thanh Navigation Bar và chọn Tab “Lịch sử ngày công”

2) Ứng dụng sẽ hiển thị tất cả lịch sử ngày công của người dùng trong tháng hiện tại. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản, nếu người dùng chọn tháng và năm sau nó ấn vào nút “Tìm kiếm”, ứng dụng sẽ hiển thị tất cả lịch sử ngày công trong tháng mà người dùng đã chọn.

2) Tại bước 2 trong luồng cơ bản, nếu người dùng ấn vào 1 item ngày công, ứng dụng sẽ di chuyển đến màn hình chi tiết ngày công và hiển thị chi tiết ngày công. Use case kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case nhắn tin

- Mô tả: Use case cho phép người dùng nhắn tin theo thời gian thực

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng ở màn hình “Nhắn tin” và chọn 1 người hoặc 1 group nhắn tin

2) Người dùng nhập nội dung tin nhắn và nhấn nút gửi. Nếu hợp lệ thì tin nhắn mới sẽ hiển thị trên màn hình. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 1 trong luồng cơ bản, nếu người dùng muốn nhắn tin với 1 người mới thì có thể chọn vào icon danh sách, ứng dụng sẽ hiển thị tất cả nhân viên trong công ty kèm trạng thái online, offline để người dùng chọn nhắn tin.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case tạo nhóm

- Mô tả: Use case cho phép người dùng tạo nhóm chat

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp icon “+” ở góc màn hình nhắn tin.

2) Người dùng nhập tên nhóm, chọn thành viên trong nhóm, trên thanh tiêu đề sẽ hiển thị nút “Tạo”. Khi ấn vào ứng dụng sẽ tạo nhóm mới. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản, nếu người dùng không nhập tên nhóm hoặc chưa chọn bất kì thành viên nào thì sẽ không hiển thị nút “Tạo nhóm”

2) Tại bước 2 trong luồng cơ bản, nếu tên nhóm đã tồn tại, ứng dung sẽ hiển thị thông báo trên màn hình và người dùng phải nhập lại tên nhóm.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case rời nhóm

- Mô tả: Use case cho phép người dùng rời nhóm chat

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp icon “Rời nhóm” ở màn hình chi tiết nhóm

2) Ứng dụng sẽ hiển thị thông báo xác nhận. Người dùng ấn OK hệ thống sẽ xóa người dùng ra khỏi nhóm. Di chuyển đến màn hình Nhắn tin. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản, khi hiển thị thông báo xác nhận, người dùng ấn Hủy thì Use Case kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng cần ở trong nhóm

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case thêm người vào nhóm

- Mô tả: Use case cho phép người dùng thêm thành viên vào nhóm chat

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp icon “Thêm người” ở màn hình chi tiết nhóm. Ứng dụng sẽ di chuyển đến màn hình thêm thành viên.

2) Người dùng chọn thành viên và nhấn nút Thêm ở thanh tiêu đề. Nếu hợp lệ thì hệ thống sẽ thêm những thành viên vừa chọn vào nhóm. Di chuyển đến màn hình chi tiết nhóm. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản, nếu người dùng chưa chọn thành viên thì nút “Thêm” trên thanh tiêu đề sẽ không sáng.

2) Tại bước 2 trong luồng cơ bản, nếu người dùng chọn những thành viên đã tồn tại trong nhóm rồi nhấn Thêm, ứng dụng sẽ hiển thị thông báo và thêm những thành viên được chọn nhưng chưa ở trong nhóm. Use case kết thúc.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng cần ở trong nhóm

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case quản lý thông tin cá nhân

- Mô tả: Use case cho phép người dùng cập nhật thông tin cá nhân của mình.

- Tác nhân: Nhân viên, trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp vào mục “Cập nhật thông tin” ở màn hình “Thông tin nhân viên”. Người dùng chọn nút ”Cập nhật thông tin” để di chuyển đến màn “Cập nhật thông tin”. Người dùng nhập thông tin và mật khẩu rồi nhấn nút “Cập nhật”. Nếu thông tin hợp lệ hệ thống sẽ cập nhật thông tin cá nhân. Use case kết thúc.

2) Người dùng chọn mục “Thay đổi mật khẩu” ở màn “Thông tin cá nhân”. Người dùng nhập mật khẩu cũ và mật khẩu mới. Nếu hợp lệ thì hệ thống sẽ hiển thị thông báo thay đổi mật khẩu thành công cho người dùng. Di chuyển về màn hình “Thông tin cá nhân”. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 1 trong luồng cơ bản, nếu người dùng nhập thiếu thông tin hoặc sai mật khẩu hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc.

2) Tại bước 2 trong luồng cơ bản, nếu người dùng nhập sai mật khẩu cũ hoặc mật khẩu mới không hợp lệ hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng cần là trưởng phòng hoặc giám đốc

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case xem lịch sử chấm công, ngày công của nhân viên

- Mô tả: Use case cho phép trưởng phòng, giám đốc xem lịch sử chấm công của nhân viên trong phòng ban của mình, toàn công ty

- Tác nhân: Trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp vào mục “Quản lý chấm công” ở màn hình “Cá nhân”. Nếu người dùng là trưởng phòng thì sẽ hiển thị tất cả lịch sử chấm công của nhân viên trong phòng ban. Nếu người dùng là giám đốc sẽ hiển thị lịch sử chấm công của toàn công ty. Use case kết thúc.

2) Người dùng chọn mục “Quản lý ngày công” ở màn “Cá nhân”. Nếu người dùng là trưởng phòng thì sẽ hiển thị tất cả lịch sử ngày công của nhân viên trong phfong ban. Nếu người dùng là giám đốc sẽ hiển thị lịch sử ngày công của toàn công ty. Use case kết thúc.

\* Luồng rẽ nhánh

1) Tại bước 1 trong luồng cơ bản, nếu người dùng chọn phòng ban (nếu là giám đốc), nhập vào tên nhân viên và ngày hệ thống sẽ lọc lịch sử chấm công theo phòng ban, tên, ngày rồi hiển thị lên màn hình. Nếu người dùng là trưởng phòng sẽ không chọn được phòng ban.

2) Tại bước 2 trong luồng cơ bản, nếu người dùng chọn phòng ban (nếu là giám đốc), nhập vào tên nhân viên và ngày hệ thống sẽ lọc lịch sử ngày công theo phòng ban, tên, ngày rồi hiển thị lên màn hình. Nếu người dùng là trưởng phòng sẽ không chọn được phòng ban.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng cần là trưởng phòng hoặc giám đốc

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### Phân tích use case quản lý thông tin nhân viên

- Mô tả: Use case cho phép trưởng phòng, giám đốc quản lý thông tin nhân viên trong phòng ban hoặc toàn công ty.

- Tác nhân: Trưởng phòng, giám đốc

- Luồng sự kiện:

\* Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhấp vào mục “Quản lý nhân viên” ở màn hình “Cá nhân”. Nếu người dùng là trưởng phòng thì sẽ hiển thị tất cả danh sách nhân viên trong phòng ban. Nếu người dùng là giám đốc sẽ hiển thị các phòng ban, sau đó giám đốc chọn một phòng ban sẽ hiển thị tất cả nhân viên trong phòng ban đó.

2) Người dùng chọn khóa/ xác nhận để khóa và xác nhận tài khoản của nhân viên.

3) Người dùng chọn một nhân viên. Ứng dụng sẽ di chuyển đến màn “Cập nhật thông tin”. Người dùng sẽ cập nhật thông tin của nhân viên sau đó nhấn nút “ Cập nhật”. Nếu thông tin hợp lệ thì hệ thống sẽ cập nhật thông tin nhân viên đó.

\* Luồng rẽ nhánh

1) Tại bước 3 trong luồng cơ bản, nếu người dùng nhập thiếu thông tin hoặc sai mật khẩu thì ứng dụng sẽ hiển thị thông báo lỗi.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng cần là trưởng phòng hoặc giám đốc

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

## Phân tích use case

### Use case đăng nhập

#### Biểu đồ trình tự



Hình 2.6: Biểu đồ trình tự luồng cơ bản use case Đăng nhập

#### Biểu đồ lớp phân tích



Hình 2.7: Biểu đồ lớp phân tích use case Đăng nhập

### Use case đăng ký

#### Biểu đồ trình tự



Hình 2.8: Biểu đồ trình tự luồng cơ bản use case Đăng ký

#### Biểu đồ lớp phân tích



Hình 2.9: Biểu đồ lớp phân tích use case Đăng ký

### Use case chấm công

#### Biểu đồ trình tự



Hình 2.10: Biểu đồ trình tự luồng cơ bản use case Chấm công

#### Biểu đồ lớp phân tích



Hình 2.11: Biểu đồ lớp phân tích use case Chấm công

### Use case xem lịch sử chấm công

#### Biểu đồ trình tự



Hình 2.12: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử chấm công

#### Biểu đồ lớp phân tích



Hình 2.13: Biểu đồ lớp phân tích use case Xem lịch sử chấm công

### Use case xem lại lịch sử ngày công

#### Biểu đồ trình tự



Hình 2.14: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử ngày công

#### Biểu đồ lớp phân tích



Hình 2.15: Biểu đồ lớp phân tích use case Xem lịch sử ngày công

### Use case nhắn tin thời gian thực

#### Biểu đồ trình tự



Hình 2.16: Biểu đồ trình tự luồng cơ bản use case Nhắn tin

#### Biểu đồ lớp phân tích



Hình 2.17: Biểu đồ lớp phân tích use case Nhắn tin

### Use case tạo nhóm chat

#### Biểu đồ trình tự



Hình 2.18: Biểu đồ trình tự luồng cơ bản use case Tạo nhóm chat

#### Biểu đồ lớp phân tích



Hình 2.19: Biểu đồ lớp phân tích use case Tạo nhóm chat

### Use case rời nhóm

#### Biểu đồ trình tự



Hình 2.20: Biểu đồ trình tự luồng cơ bản use case Rời nhóm

#### Biểu đồ lớp phân tích



Hình 2.21: Biểu đồ lớp phân tích use case Rời nhóm

### Use case thêm thành viên vào nhóm

#### Biểu đồ trình tự



Hình 2.22: Biểu đồ trình tự luồng cơ bản use case Thêm thành viên vào nhóm

#### Biểu đồ lớp phân tích



Hình 2.23: Biểu đồ lớp phân tích use case Thêm thành viên vào nhóm

### Use case cập nhật thông tin

#### Biểu đồ trình tự



Hình 2.24: Biểu đồ trình tự luồng cơ bản use case Cập nhật thông tin

#### Biểu đồ lớp phân tích



Hình 2.25: Biểu đồ lớp phân tích use case Cập nhật thông tin

### Use case xem lịch sử chấm công, ngày công của nhân viên

#### Biểu đồ trình tự



Hình 2.26: Biểu đồ trình tự luồng cơ bản use case Xem lịch sử chấm công, ngày công nhân viên

#### Biểu đồ lớp phân tích



Hình 2.27: Biểu đồ lớp phân tích use case Xem lịch sử chấm công, ngày công nhân viên

### Use case quản lý thông tin nhân viên

#### Biểu đồ trình tự



Hình 2.28: Biểu đồ trình tự luồng cơ bản use case Quản lý thông tin nhân viên

#### Biểu đồ lớp phân tích



Hình 2.29: Biểu đồ trình tự luồng cơ bản use case Quản lý thông tin nhân viên

# : TRIỂN KHAI HỆ THỐNG VÀ KIỂM THỬ

## Cài đặt chương trình

### Yêu cầu hệ thống

Để có thể triển khai được chương trình thì yêu cầu máy tính cần phải cài đặt một số IDE và môi trường như sau:

* Visual Studio Code
* Android Studio
* Git
* NodeJs, Python
* MongoDB Compass

### Cài đặt chương trình

#### Cài đặt hệ thống backend so sánh khuôn mặt.

* Bước 1: Tạo folder để clone mã nguồn từ github

Người dùng mở cmd hoặc powershell gõ lệnh :

git clone <https://github.com/lbui2602/face-compare-service-python>

cd face-compare-service-python

* Bước 2: Tạo môi trường

Người dùng gõ lệnh

python -m venv venv  
venv\Scripts\activate

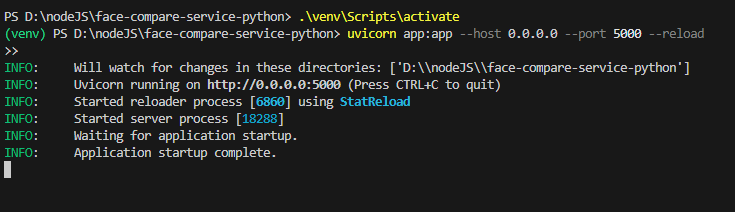
* Bước 3: Chạy lệnh sau để cài đặt thư viện

pip install -r require

* Bước 4: Chạy back end

uvicorn app:app --host 0.0.0.0 --port 5000 --reload

Nếu thành công thì terminal sẽ hiển thị



Hình 3.1: Màn hình terminal chạy backend so sánh khuôn mặt thành công

#### Cài đặt hệ thống backend quản lý chấm công và quản lý thông tin (NodeJs)

* Bước 1: Tạo folder để clone mã nguồn từ github

Người dùng mở cmd hoặc powershell gõ lệnh:

git clone <https://github.com/lbui2602/datn_backend>

cd datn\_backend

* Bước 2: Cài đặt thư viện

Người dùng gõ lệnh

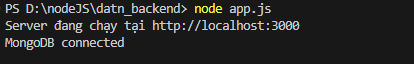
npm install

* Bước 3: Chạy back-end

Người dùng gõ lệnh

node app.js

Nếu thành công thì terminal sẽ hiển thị



Hình 3.2: Màn hình terminal chạy backend quản lý chấm công và quản lý thông tin thành công

#### Cài đặt ứng dụng bên phía client (Android Kotlin)

* Bước 1: Tạo folder clone mã nguồn từ github, mở cmd hoặc powershell

Người dùng gõ lệnh

git clone <https://github.com/lbui2602/DATN_Android>

Sau đó mở Android Studio và mở project DATN\_Android đã clone về

* Bước 2: Mở file string.xml để thay đổi GG\_API\_KEY = YOUR\_API\_KEY
* Bước 3: Người dùng lấy máy Android thật kết nối dây USB với máy tính chạy back-end. Sau đó mở cmd mới và cd đến thư mục platform-tools chứa file adb.exe  
  Ví dụ

cd C:\Users\Bui\_Duc\_Luong\AppData\Local\Android\Sdk\platform-tools

* Bước 4: Mở cổng port cho điện thoại

Người dùng gõ lệnh:

adb reverse tcp:3000 tcp:3000

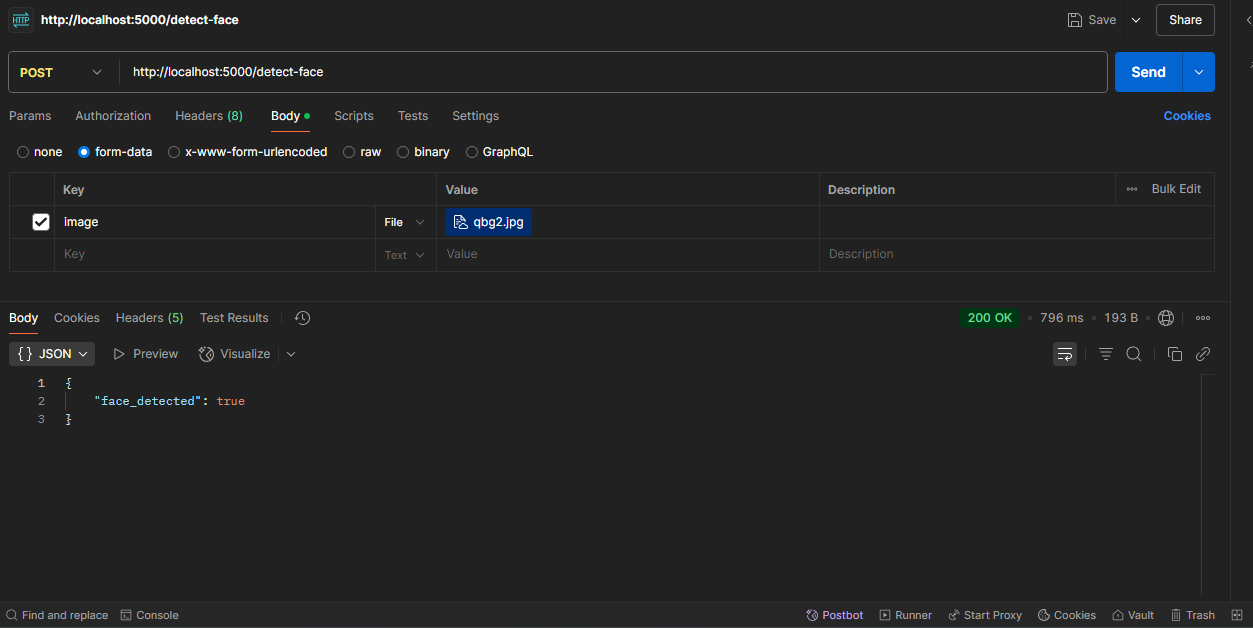
Nếu thành công sẽ không hiển thị gì hoặc hiển thị 3000

* Bước 5: Run app, người dùng bấm vào icon “Run App” trên Android Studio nếu thành công ứng dụng sẽ được cài đặt trên máy thật.

## Kiểm thử API ( Sử dụng Postman )

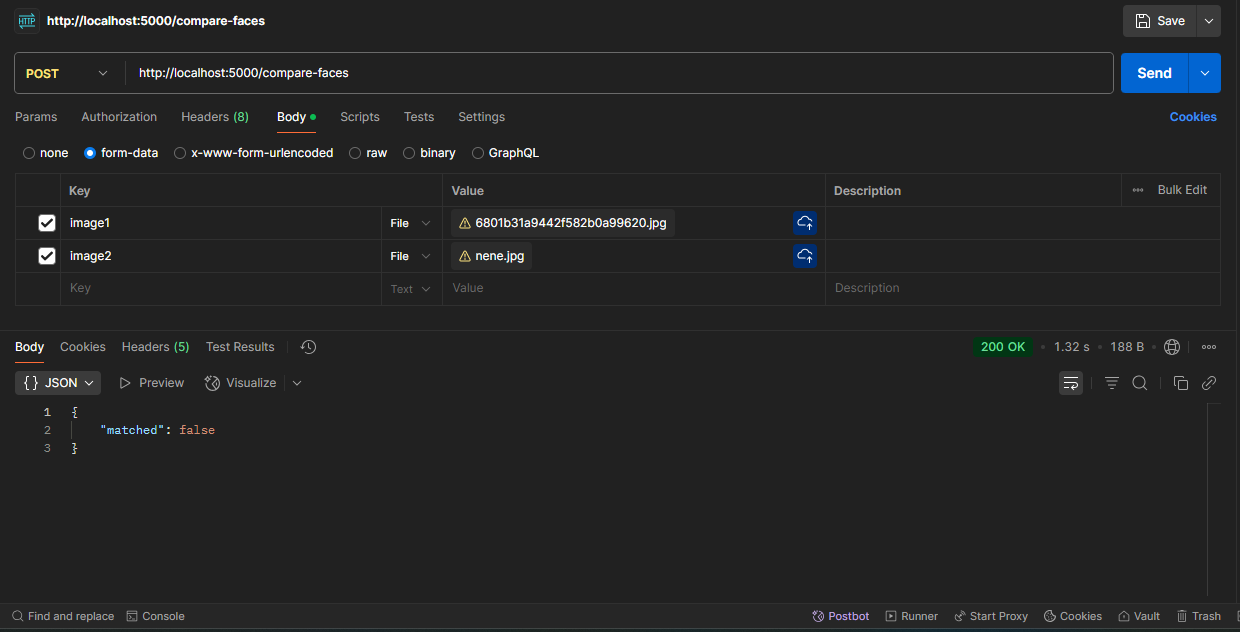
### Kiểm thử API hệ thống so sánh khuôn mặt

#### API phát hiện khuôn mặt



Hình 3.3: Test postman API phát hiện khuôn mặt

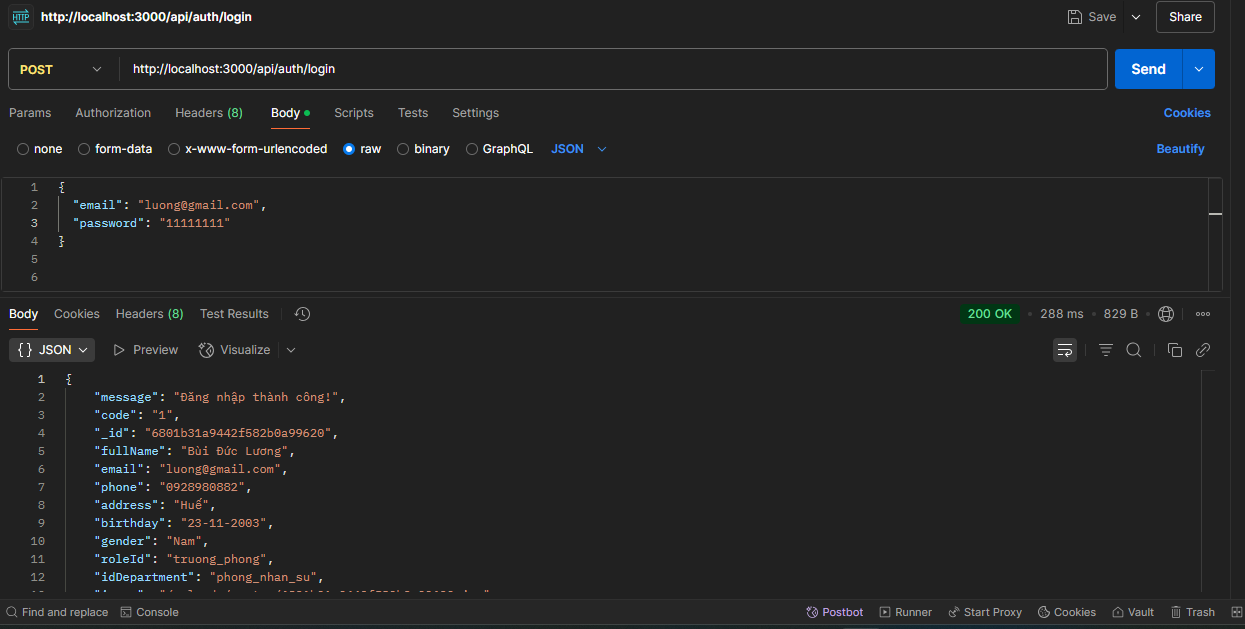
#### API so sánh khuôn mặt



Hình 3.4: Test postman API so sánh khuôn mặt

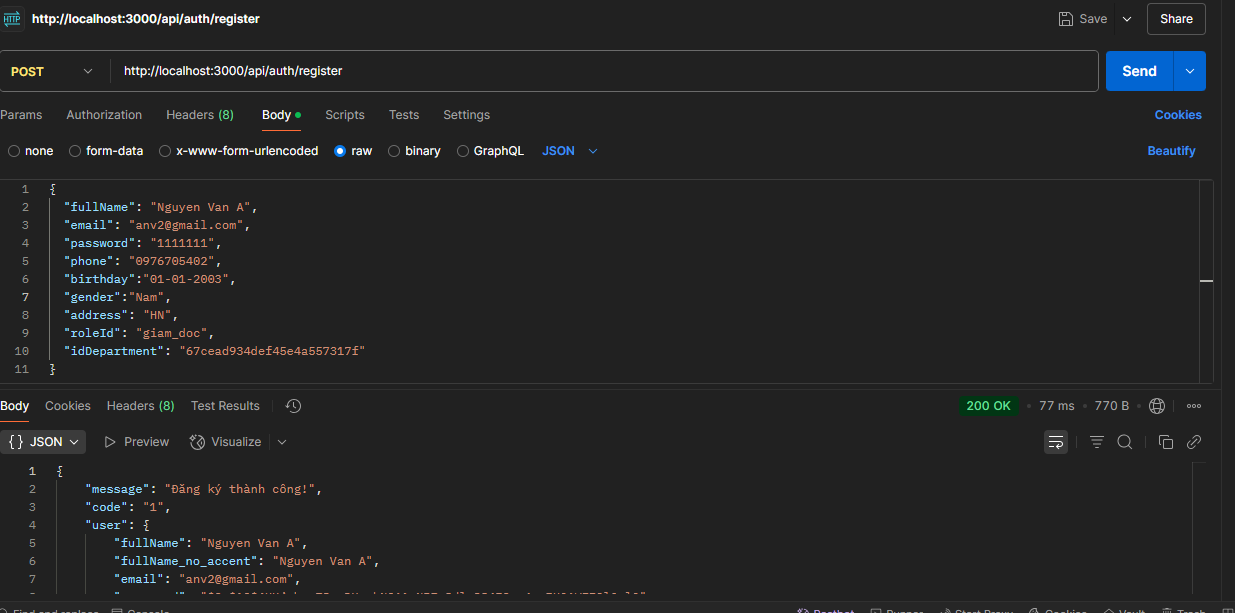
### Kiểm thử API hệ thống quản lý chấm công và quản lý thông tin

#### API đăng nhập



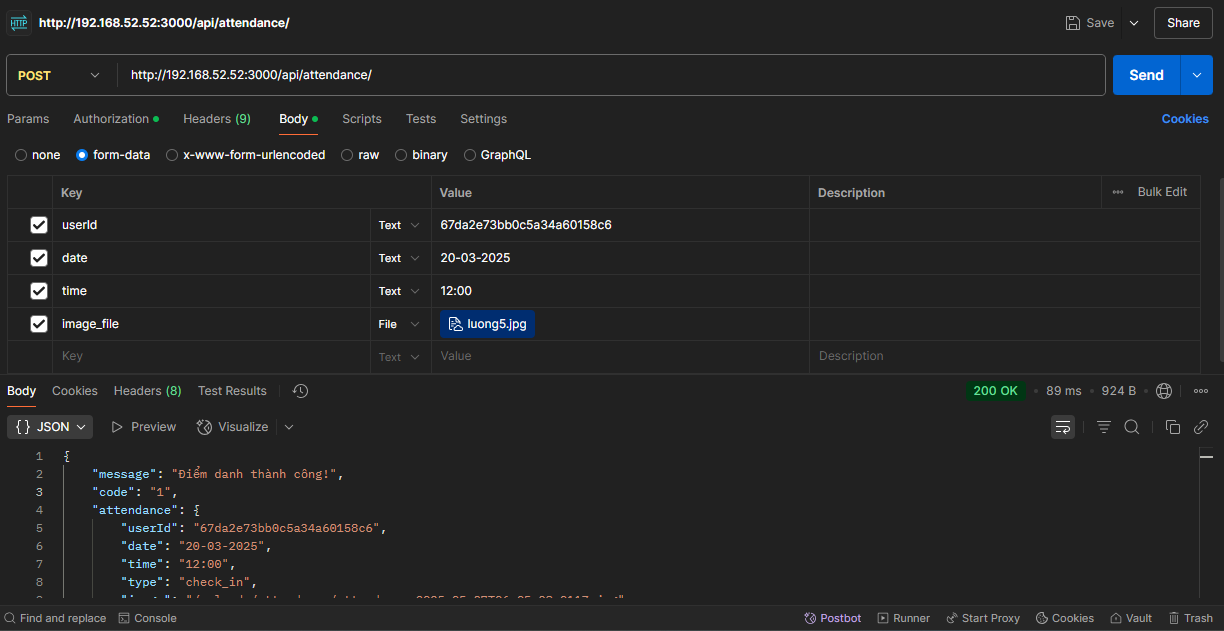
Hình 3.5: Test postman API đăng nhập

#### API đăng ký



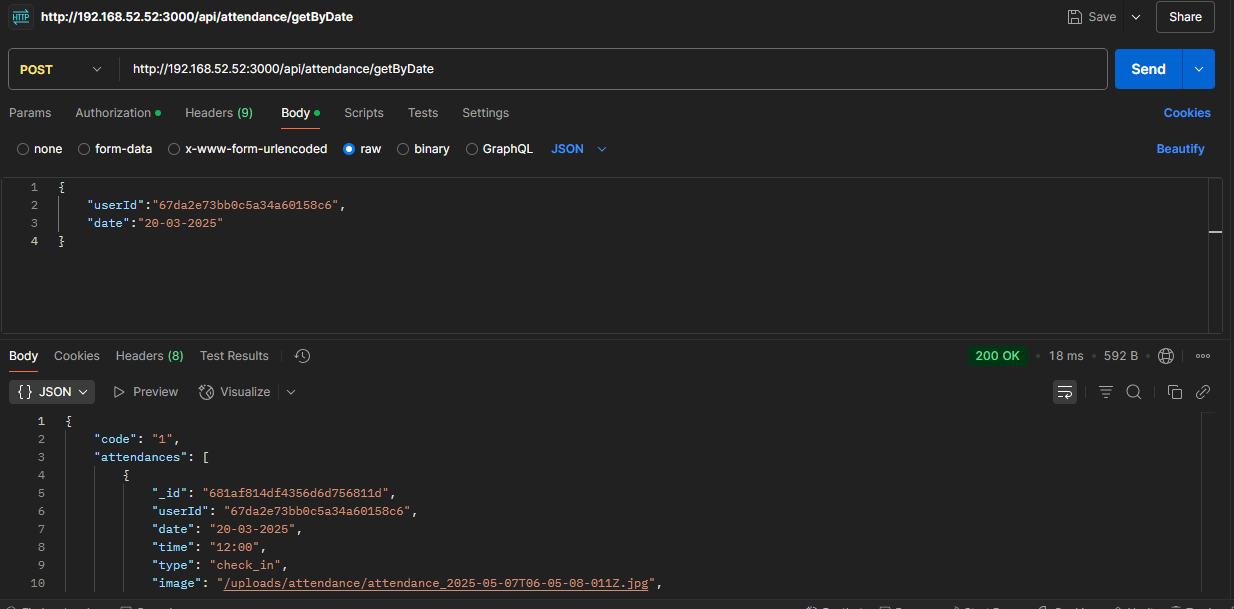
Hình 3.6: Test postman API phát hiện đăng ký

#### API chấm công



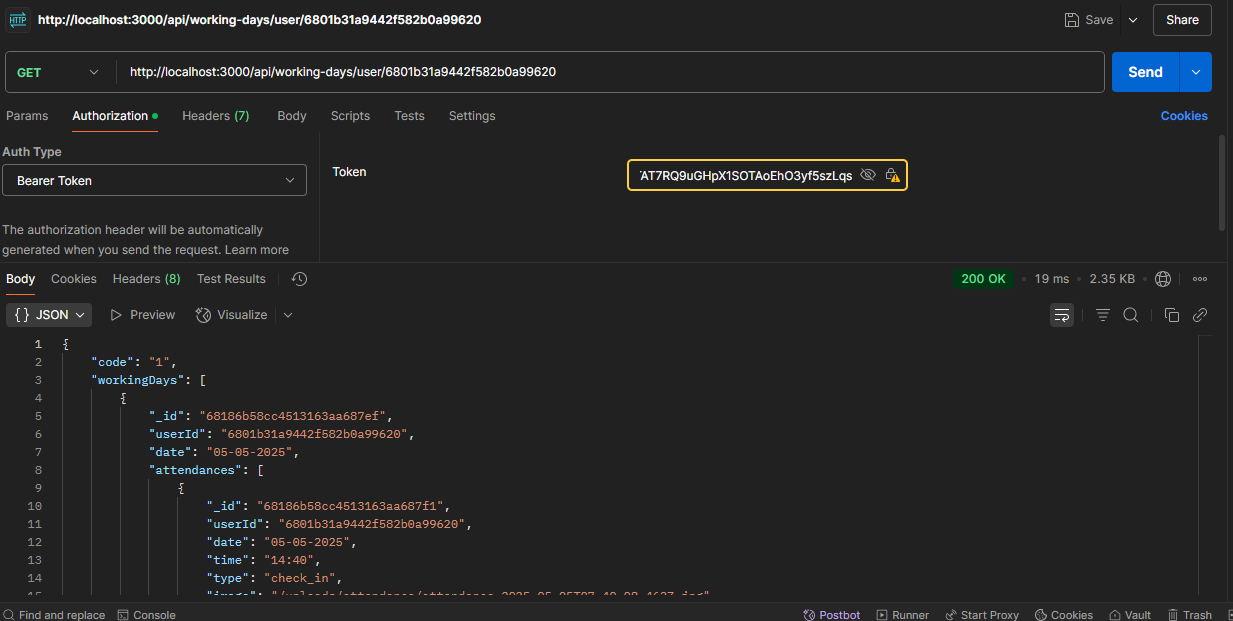
Hình 3.7: Test postman API chấm công

#### API xem lịch sử chấm công



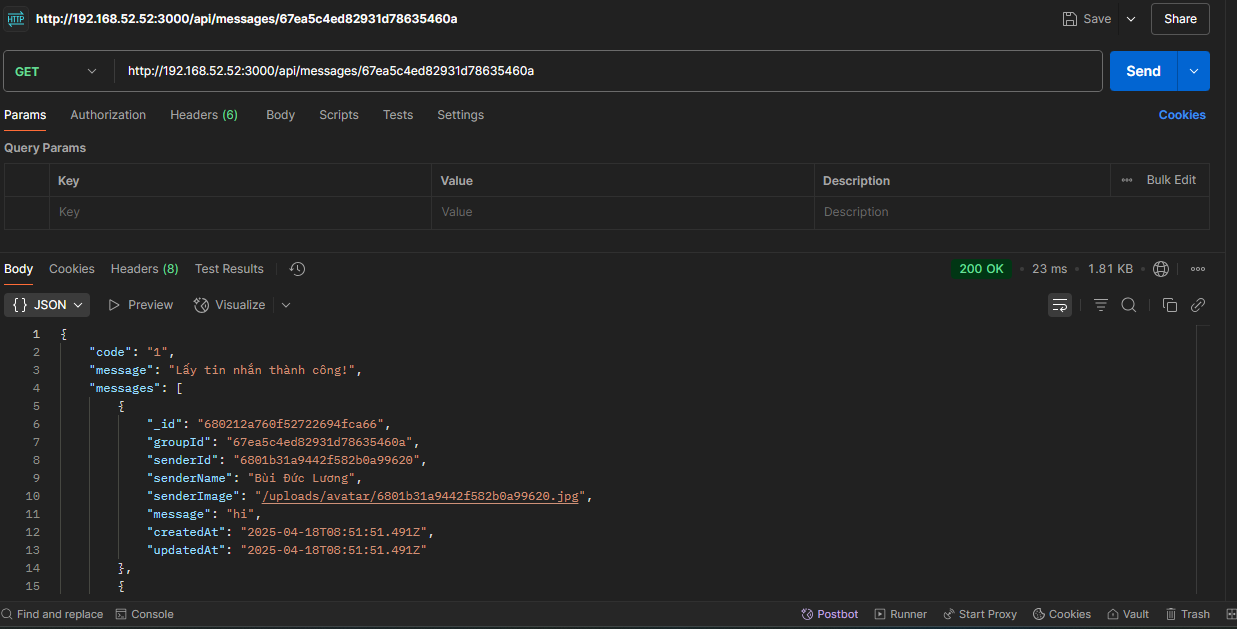
Hình 3.8: Test postman API xem lịch sử chấm công

#### API xem lịch sử ngày công



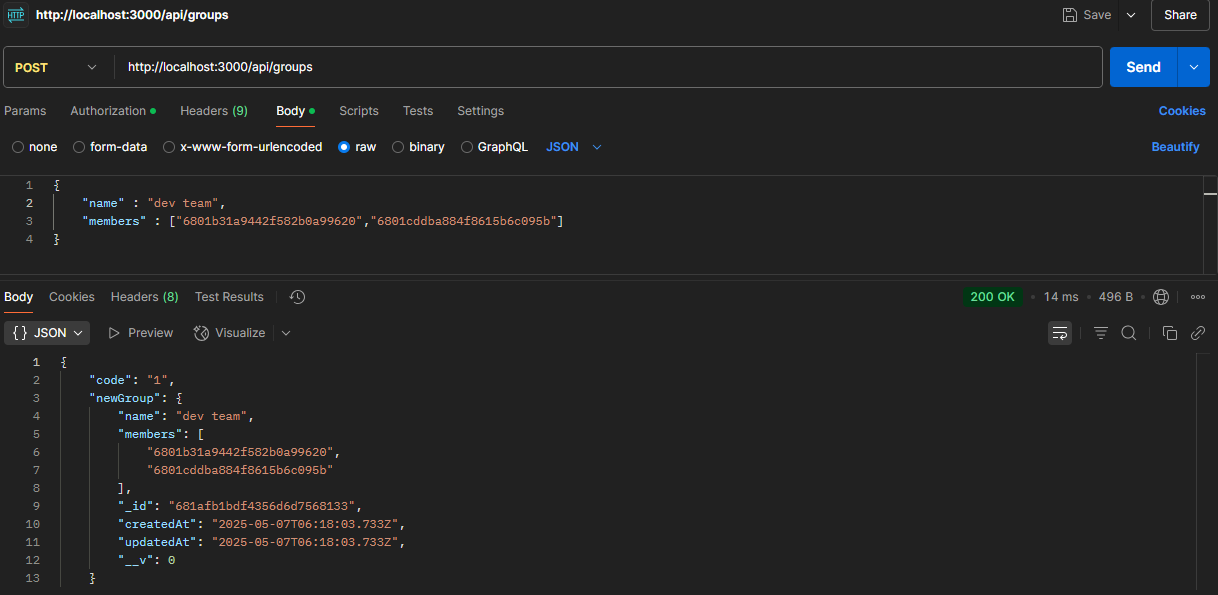
Hình 3.9: Test postman API xem lịch sử ngày công

#### API lấy tin nhắn



Hình 3.10: Test postman API lấy tin nhắn

#### API tạo nhóm



Hình 3.11: Test postman API tạo nhóm

# Kết luận

**Tóm tắt kết quả nghiên cứu:**

Đề tài “Xây dựng ứng dụng chấm công sử dụng khuôn mặt cho doanh nghiệp” của em đã hoàn thiện và đạt được những kết quả quan trọng. Đề tài giúp em tìm hiểu, nghiên cứu về các bài toán nhận diện, so sánh khuôn mặt, quy trình nghiệp vụ của hệ thống chấm công cho doanh nghiệp từ việc phân tích các chức năng, vẽ biểu đồ use case tổng quát, phân rã, biểu đồ trình tự các use case, thiết kế database, đặc tả từng chức năng.

Ứng dụng đã đạt được các yêu cầu cơ bản, đảm bảo các chức năng quan trọng như đăng nhập, đăng ký, chấm công, xem lịch sử, quản lý thông tin, nhắn tin theo thời gian thực. Giao diện ứng dụng đơn giản, dễ dùng, thân thiện với người dùng.

**Đề xuất hướng phát triển tương lai**

* Tăng cường bảo mật: Cần cải thiện cơ chế bảo mật hệ thống để ngăn chặn các rủi ro mất mát thông tin quan trọng của người dùng, bao gồm việc mã hóa dữ liệu và áp dụng các biện pháp bảo mật nâng cao.
* Cải thiện độ chính xác và tốc độ nhận diện và so sánh khuôn mặt.
* Tích hợp quản lý nhân sự xử lí các bài toán nghiệp vụ của quản lý nhân sự, như một app nội bộ. Xây dựng website cho admin để quản lý toàn bộ thông tin dữ liệu của ứng dụng.
* Tăng cường đề phòng gian lận chấm công: ngăn ngứa chấm công bằng hình ảnh, AI, ngăn chặn fake GPS để chấm công.
* Thêm chức năng quên mật khẩu để hệ thống hoàn chỉnh hơn.

# Tài liệu tham khảo

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[3] Trang tài liệu của NodeJs: <https://nodejs.org/docs/latest/api/>

[4] Trang tài liệu của FastAPI: [https://apidog.com/articles/what-is-fastapi/](about:blank)

[5] Trang tài liệu của Face Recognition:

<https://face-recognition.readthedocs.io/en/latest/index.html>

[6] Trang tài liệu của Android: [https://developer.android.com/develop](about:blank)

[7] Trang tài liệu của SocketIO: [https://socket.io/docs/v4/](about:blank)

[8] Trang tin tức về lập trình <https://viblo.asia>