|  |  |
| --- | --- |
|  |  |
| BÙI THỊ MINH NGUYỆT | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC |
| NGÀNH CÔNG NGHỆ THÔNG TIN |
| **XÂY DỰNG WEBAPP CHẤM CÔNG TỰ ĐỘNG TÍCH HỢP AI CHO DOANH NGHIỆP** |
|  |
|  |
| **CBHD: *TS. Đặng Trọng Hợp*** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên: Bùi Thị Minh Nguyệt** |
| **Mã số sinh viên: *2020605647*** |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |
|  |

**MỤC LỤC**

[**LỜI CẢM ƠN 4**](#_Toc167471653)

[**DANH MỤC HÌNH ẢNH 5**](#_Toc167471654)

[**DANH MỤC CÁC CHỮ VIẾT TẮT 6**](#_Toc167471655)

[**CHƯƠNG 1. ĐẶT VẤN ĐỀ 7**](#_Toc167471656)

[1.1. Tên đề tài 7](#_Toc167471657)

[1.2. Lý do chọn đề tài 7](#_Toc167471658)

[1.3. Mục tiêu nghiên cứu 7](#_Toc167471659)

[1.4. Đối tượng phạm vị 8](#_Toc167471660)

[1.5. Kết quả đạt được 8](#_Toc167471661)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG 8**](#_Toc167471662)

[2.1. Cơ sở lý thuyết 8](#_Toc167471663)

[2.1.1. Web Application 8](#_Toc167471664)

[2.1.2. HTML 9](#_Toc167471665)

[2.1.3. VueJS 10](#_Toc167471666)

[2.1.4. Java Spring boot 13](#_Toc167471667)

[2.1.5. PostgreSQL 13](#_Toc167471668)

[2.2. Công cụ sử dụng 14](#_Toc167471669)

[2.2.1. Các công cụ sử dụng cho Website 14](#_Toc167471670)

[2.2.2 Các công cụ sử dụng cho bài toán AI check in bằng khuôn mặt 15](#_Toc167471671)

[**CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 23**](#_Toc167471672)

[3.1. Phân tích hệ thống 23](#_Toc167471673)

[3.1.1. Vai trò người dùng 23](#_Toc167471674)

[3.1.2. Yêu cầu chức năng 23](#_Toc167471675)

[3.1.3. Yêu cầu phi chức năng 24](#_Toc167471676)

[3.2. Biểu đồ use case tổng quát 25](#_Toc167471677)

[3.3. Biểu đồ thực thể liên kết 26](#_Toc167471678)

[3.4. Đặc tả use case 26](#_Toc167471679)

[3.4.1. Use case Đăng nhập 26](#_Toc167471680)

[3.4.2. Use case đăng xuất 28](#_Toc167471681)

[3.4.3. Use case Đăng ký vắng mặt 29](#_Toc167471682)

[3.4.4. Use case Cập nhật thông tin tài khoản 31](#_Toc167471683)

[3.4.5. Use case Chấm công bằng khuôn mặt 32](#_Toc167471684)

[3.4.6. Use case Chấm công thủ công 33](#_Toc167471685)

[3.4.7. Use case Xem danh sách vi phạm 34](#_Toc167471686)

[3.4.8. Use case Quản lý tài khoản 35](#_Toc167471687)

[3.4.9. Use case Quản lý phòng ban 38](#_Toc167471688)

[**3.5. THIẾT KẾ HỆ THỐNG 41**](#_Toc167471689)

[3.5.1. Biểu đồ triển khai hệ thống 41](#_Toc167471690)

[3.5.2. Thiết kế cơ sở dữ liệu 41](#_Toc167471691)

[3.5.3. Biểu đồ hoạt động bài toán AI check in khuôn mặt 42](#_Toc167471692)

[**CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 44**](#_Toc167471693)

[4.1. Giao diện các màn hình trên hệ thống 44](#_Toc167471694)

[4.2. Kiểm thử hệ thống 51](#_Toc167471695)

[**KẾT LUẬN 53**](#_Toc167471696)

[**TÀI LIỆU THAM KHẢO 55**](#_Toc167471697)

# LỜI CẢM ƠN

Trong thời đại 4.0, việc ứng dụng công nghệ thông tin vào quản lý doanh nghiệp đã phát triển mạnh ở các nước tiên tiến trên thế giới. Trong thị trường cạnh tranh ngày càng khốc liệt cùng với sự tiến bộ không ngừng của trí tuệ nhân tạo và các công nghệ liên quan, việc tối ưu hóa quá trình quản lý nhân sự và nâng cao hiệu suất làm việc của nhân viên trở nên cực kỳ quan trọng. Trước tình hình đó, vấn đề áp dụng công nghệ để quản lý dường như đã thay thế rất nhiều cho các phương thức thủ công truyền thống...

Nhu cầu quản lý nhân sự đang là một trong những nhu cầu hết sức thiết yếu trong bối cảnh các doanh nghiệp, đặc biệt là nhu cầu về việc quản lý và sự minh bạch trong quá trình chấm công cho nhân viên trong doanh nghiệp.Việc xây dựng webapp quản lý chấm công góp phần giúp các doanh nghiệp tối ưu hóa quá trình chấm công, giảm thiều sai sót, tăng cường sự chính xác trong quản lý thời gian làm việc cho nhân viên.

Sau khi tìm hiểu cẩn thận và dưới sự hướng dẫn của thầy Đặng Trọng Hợp, em đã phân tích, thiết kế và xây dựng nên Webapp quản lý chấm công tích hợp AI sử dụng VueJS, Java và PostgreSQL để giải quyết vấn đề trên. Đó cũng là lý do chính mà em chọn đề tài này.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội, thầy giáo hướng dẫn đề tài – Tiến Sĩ Đặng Trọng Hợp – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành đồ án.

***Hà Nội, Ngày 20 tháng 05 năm 2024***

# DANH MỤC HÌNH ẢNH

[Hình 2.1 HTML 9](#_Toc167366017)

[Hình 2.2 VueJS 10](#_Toc167366018)

[Hình 2.3 Cấu trúc thư mục VueJS 11](#_Toc167366019)

[Hình 2.4 Java 13](#_Toc167366020)

[Hình 2.5 PostgreSQL 14](#_Toc167366021)

[Hình 2.6 Artificial Intelligence 15](#_Toc167366022)

[Hình 2.7. Logo Anaconda 18](#_Toc167366023)

[Hình 2.8. Logo python 20](#_Toc167366024)

[Hình 2.9. Logo Pytorch 22](#_Toc167366025)

[Hình 2.10. Logo FastAPI 24](#_Toc167366026)

[Hình 3.1. Sơ đồ use case tổng quát 28](#_Toc167366027)

[Hình 3.2. Biểu đồ thực thể liên kết 29](#_Toc167366028)

[Hình 3.3. Biều đồ trình tự use case Đăng nhập 31](#_Toc167366029)

[Hình 3.4. Biểu đồ trình tự usecase đăng xuất 32](#_Toc167366030)

[Hình 3.5. Biều đồ trình tự usecase đăng ký vắng mặt 33](#_Toc167366031)

[Hình 3.6. Biều đồ trình tự usecase cập nhật thông tin tài khoản 35](#_Toc167366032)

[Hình 3.7. Biểu đồ trình tự usecsase chấm công khuôn mặt 36](#_Toc167366033)

[Hình 3.8. Biểu đồ trình tự usecase chấm công thủ công 37](#_Toc167366034)

[Hình 3.9. Biểu đồ trình tự usecase xem danh sách vi phạm 38](#_Toc167366035)

[Hình 3.10. Biểu đồ trình tự usecase quản lý tài khoản 41](#_Toc167366036)

[Hình 3.11. Biểu đồ trình tự usecase quản lý phòng ban 43](#_Toc167366037)

[Hình 3.12. Biểu đồ triển trai hệ thống 44](#_Toc167366038)

[Hình 3.13. Biểu đồ cấu trúc cơ sở dữ liệu 45](#_Toc167366039)

[Hình 3.14. Sơ đồ hoạt động bài toán AI check in khuôn mặt 46](#_Toc167366040)

[Hình 4.1. Giao diện màn hình đăng nhập 47](#_Toc167366041)

[Hình 4.2. Giao diện màn hình dashboard 48](#_Toc167366042)

[Hình 4.3. Giao diện màn hình Đăng ký vắng mặt 49](#_Toc167366043)

[Hình 4.4. Giao diện màn hình Dữ liệu chấm công 49](#_Toc167366044)

[Hình 4.5. Giao diện màn hình Quản lý tài khoản 50](#_Toc167366045)

[Hình 4.6.Giao diện màn hình Quản lý phòng ban 50](#_Toc167366046)

[Hình 4.7. Giao diện màn hình Profile 51](#_Toc167366047)

[Hình 4.8. Giao diện màn hình Phê duyệt 51](#_Toc167366048)

[Hình 4.9. Giao diện màn hình thêm mới tài khoản 52](#_Toc167366049)

[Hình 4.10. Giao diện màn hình Thêm mới phòng ban 53](#_Toc167366050)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| API | Application Programming Interface |
| OOP | Object Oriented Programming |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| HTTP | HyperText Transfer Protocol |
| SPA | Single Page Application |
| SEO | Search Engine Optimization |
| SQL | Structured Query Language |
| NoSQL | Not Only SQL |
| OQL | Object Query Language |

# CHƯƠNG 1. ĐẶT VẤN ĐỀ

## 1.1. Tên đề tài

Xây dựng ứng dụng web chấm công tự động tích hợp AI cho doanh nghiệp

## 1.2. Lý do chọn đề tài

Nhu cầu quản lý nhân sự đang là một trong những nhu cầu hết sức thiết yếu trong bối cảnh các doanh nghiệp, đặc biệt là nhu cầu về việc quản lý và sự minh bạch trong quá trình chấm công cho nhân viên trong doanh nghiệp.Việc xây dựng webapp quản lý chấm công góp phần giúp các doanh nghiệp tối ưu hóa quá trình chấm công, giảm thiều sai sót, tăng cường sự chính xác trong quản lý thời gian làm việc cho nhân viên. Vậy nên em chọn đề tài ˮ***Xây dựng ứng dụng web chấm công tự động tích hợp AI cho doanh nghiệp*** ˮ với mục tiêu tối ưu hóa quá trình quản lý nhân sự và nâng cao hiệu suất làm việc của nhân viên.

Em rất mong nhận được những lời nhận xét, đánh giá của thầy cô để bài báo cáo trở nên hoàn thiện và kỹ năng lập trình của bản thân được phát triển xa hơn nữa trong tương lai.

## 1.3. Mục tiêu nghiên cứu

Mục tiêu chính của nghiên cứu bao gồm: phân tích yêu cầu hệ thống và xác định các hạn chế của quy trình chấm công hiện tại; thiết kế kiến trúc hệ thống web, tích hợp AI để tối ưu hóa chấm công như nhận diện khuôn mặt và phát hiện gian lận; xây dựng cơ sở dữ liệu an toàn và bảo mật; phát triển, kiểm thử và triển khai hệ thống trong môi trường thực tế; đồng thời thu thập phản hồi để nâng cấp và bảo trì hệ thống liên tục. Kết quả nghiên cứu sẽ được đánh giá dựa trên độ chính xác, hiệu suất và mức độ hài lòng của người dùng, đồng thời đề xuất các hướng phát triển tiếp theo. Nghiên cứu này hướng tới việc cải thiện quy trình quản lý chấm công, mang lại nhiều lợi ích cho doanh nghiệp.

## 1.4. Đối tượng phạm vị

Các doanh nghiệp lớn nhỏ trong thị trường Việt Name

## 1.5. Kết quả đạt được

Web bao gồm những mục chính và chức năng chính:

* Các chức năng truy cập và hệ thống: Đăng nhập, đăng xuất
* Các chức năng liên quan đến quy trình chấm công: Chấm công bằng AI, chấm công thủ công, quản lý chấm công, đăng ký vắng mặt, phê duyệt, danh sách vi phạm.
* Phân quyền : Người quản trị, Trưởng phòng, Nhân viên.
* Các chức năng người quản trị: quản lý tài khoản, quản lý phòng ban

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

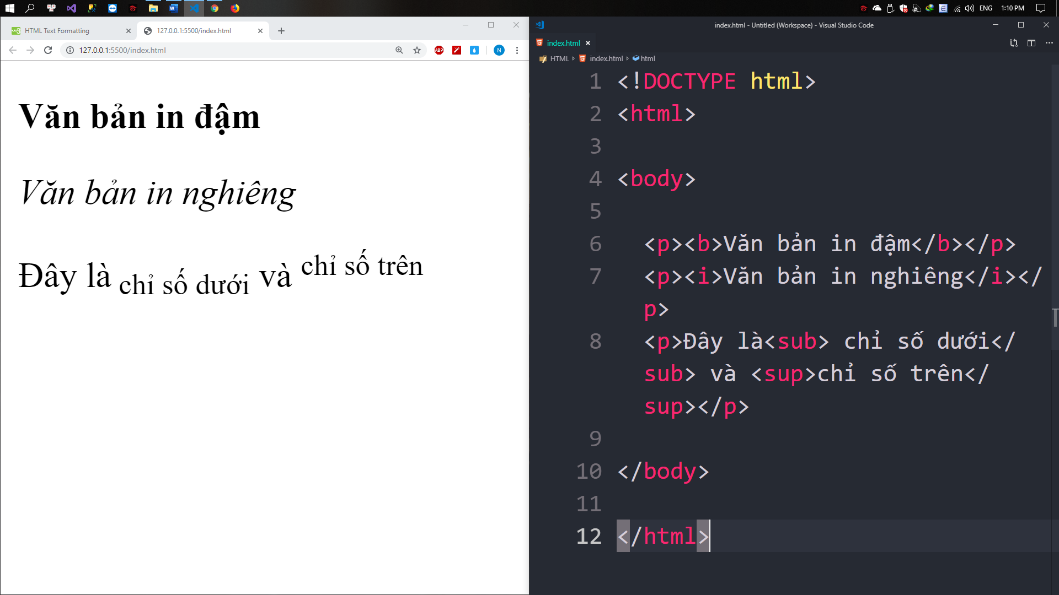
## 2.1. Cơ sở lý thuyết

### 2.1.1. Web Application

Web app là một ứng dụng phần mềm được phát triển để chạy trên môi trường web. Điều này có nghĩa là người dùng có thể truy cập và sử dụng nó thông qua trình duyệt web mà không cần phải tải về và cài đặt trực tiếp lên thiết bị của họ. WebApp thường được xây dựng bằng các ngôn ngữ lập trình web như HTML, CSS và JavaScript, cũng như các framework và thư viện phổ biến như Angular, React và Vue.js. Các xử lý khác liên quan đến backend như database sẽ được thực hiện phía trên server. Với một số công nghệ như VueJS, một trang web có thể giống y hệt một ứng dụng di động thật sự.

* Ưu điểm:
* Chỉ cần có kiến thức về web là viết được
* Viết một lần, chạy được trên mọi hệ điều hành
* Người dùng không cần phải cài app, có thể vào thẳng trang web
* Nhược điểm
* Với một số máy đời cũ, Web App sẽ bị bể giao diện, hiển thị sai, hoặc javascript không chạy.
* Performance chậm

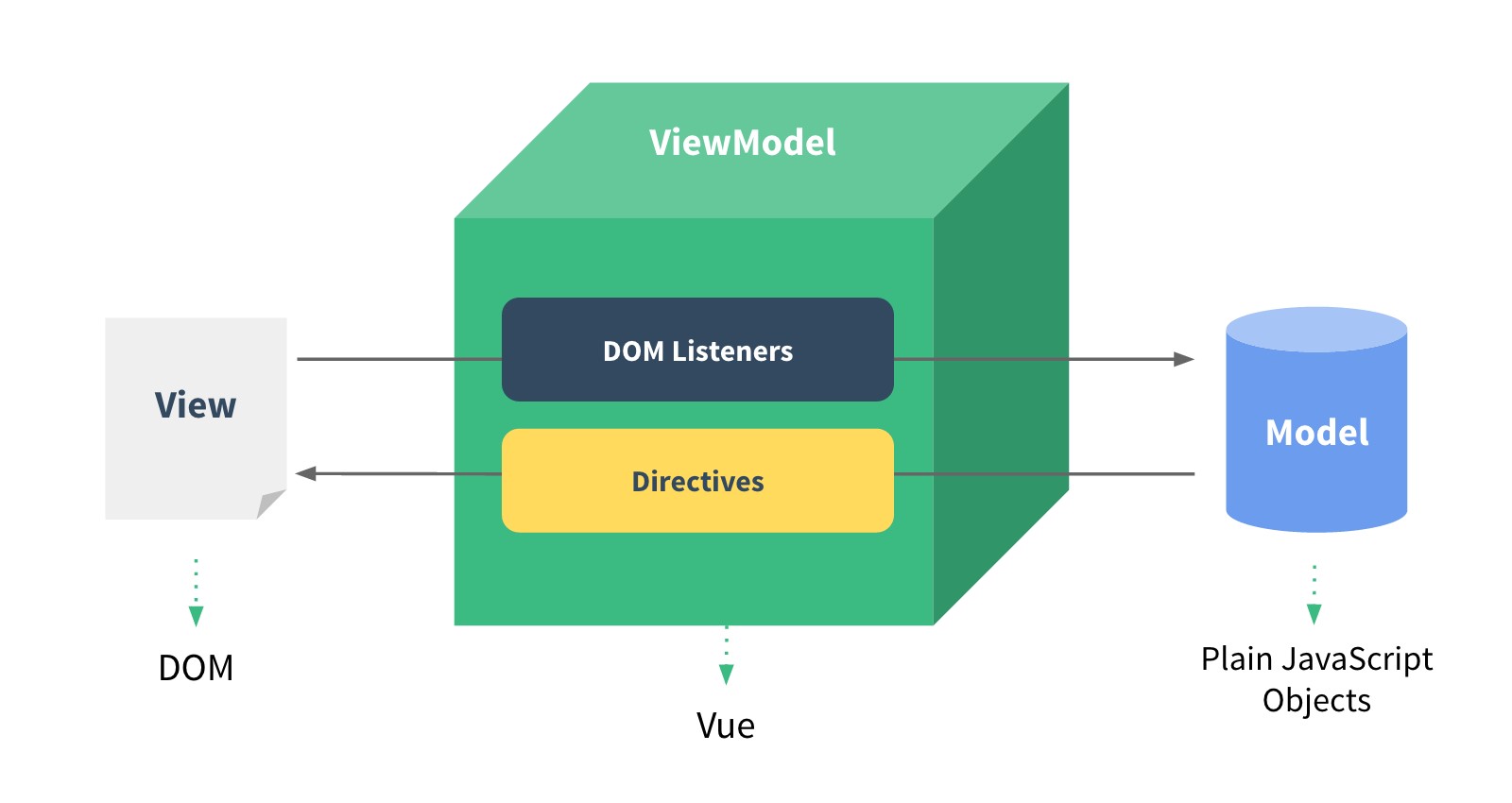
### 2.1.2. HTML



Hình 2.1 HTML

* HTML là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo ra các trang web. HTML được sử dụng để định dạng và hiển thị nội dung trên trang web, bao gồm văn bản, hình ảnh, video, âm thanh và các phần tử khác.
* HTML được sử dụng để tạo ra các trang web tĩnh và động. Trang web tĩnh là các trang web không thay đổi nội dung của chúng, trong khi trang web động có thể thay đổi nội dung của chúng dựa trên hành động của người dùng.
* HTML được sử dụng cùng với CSS và JavaScript để tạo ra các trang web đẹp và tương tác với người dùng. CSS được sử dụng để định dạng và trang trí trang web, trong khi JavaScript được sử dụng để tạo ra các hiệu ứng tương tác và động.
* HTML là một phần quan trọng của lập trình web và là một trong những kỹ năng cơ bản mà một lập trình viên web cần phải có khi có mong muốn phát triển mảng lập trình web

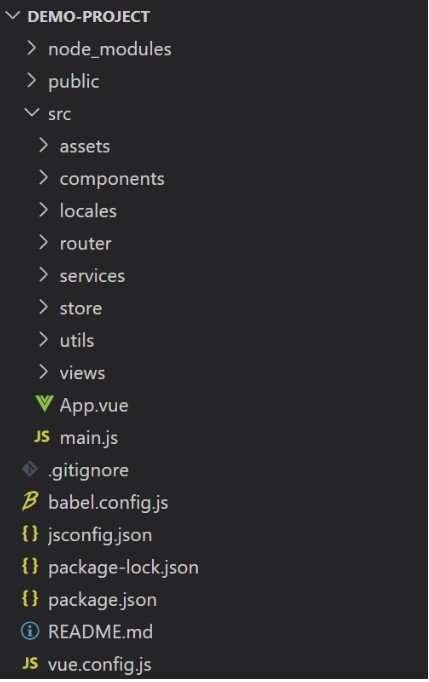
### 2.1.3. VueJS



Hình 2.2 VueJS

Vue.js là một thư viện của Javascript để xây dựng giao diện cho người dùng, nó hoạt động khá linh hoạt theo mô hình MVVM Model View  View Model). Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ dàng chọn và tích hợp với các thư viện khác hoặc các dự án hiện có. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ. Vue cũng đáp ứng được nhu cầu xây dựng ứng dụng một trang SPA (single-page applications) với độ phức tạp cao hơn nhiều.

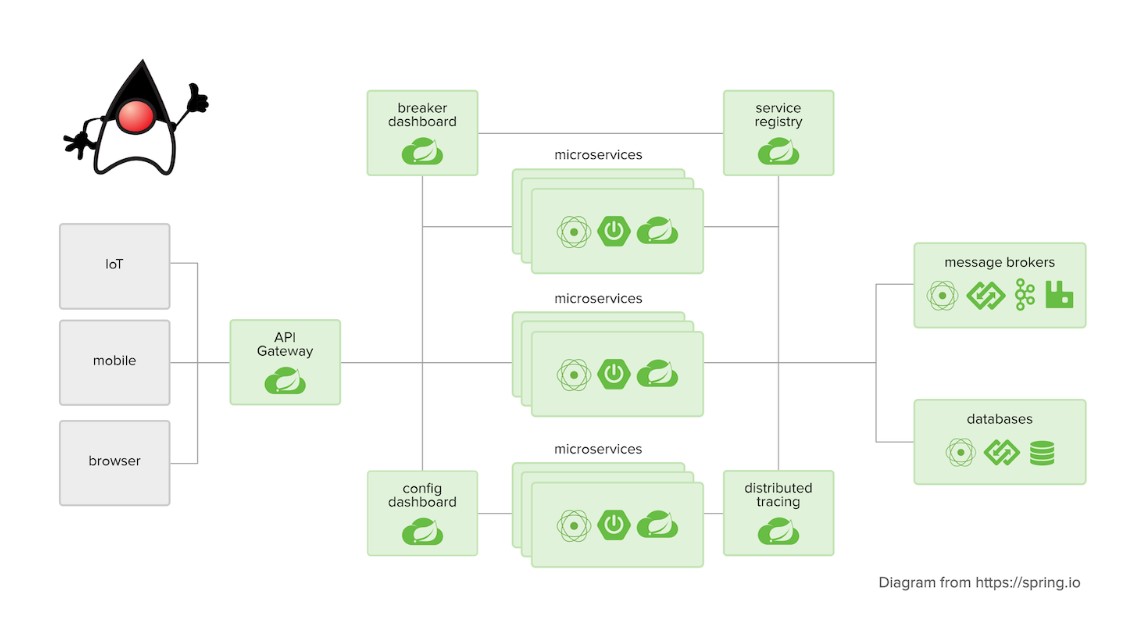
* Ưu điểm:
* Dễ dàng học và áp dụng vào các dự án
* Kho thư viện lớn, hỗ trợ xây dựng giao diện người nhanh chóng
* Dung lượng tải thấp, giúp tốc độ tải trang nhanh hơn
* Trang web nhẹ, tốc độ xử lý cực kỳ nhanh bởi được Render, xử lý bằng Javascript
* Có khả năng xử lý các render thuộc server thành file Javascript tĩnh
* Cấu trúc thư mục src của 1 dự án VueJS



Hình 2.3 Cấu trúc thư mục VueJS

* **node\_modules:** Đây là thư mục chứa tất cả các thư viện cần để xây dựng Vue.
* **public:** Đây là thư mục chứa các static assets mà bạn không muốn chạy thông qua webpack
* **index.html:** Với ứng dụng SPA Single Page Application) thì có 1 trang duy nhất. Sau đó, nội dung của trang bị thay đổi mà không phải tải lại trang, thì đây chính là trang duy nhất đó.
* **src:** Đây là thư mục chứa mã nguồn của dự án. Bên trong đó gồm các folder:
* **assets:** Dùng để bất kỳ tài nguyên nào được nhập vào trong các components của bạn
* **components:** Là thư mục chứa các UI components của dự án
* **locales:** Để lưu trữ file ngôn ngữ dạng \*.json (thường sử dụng Vue-i18n) **router:** Là nơi sẽ viết các routes và kết nối chúng với Components **services:** Là thư mục chứa các hàm call API
* **store (optional):** Các hằng số Vuex trong các đột biến-type.js, các mô đun VueX trong các mô đun thư mục con (mà sau đó được nạp trong index.js).
* **utils:** Chứa các hàm sử dụng chung trong một số components
* **views:** Để dự án dễ dàng hơn, nên chia tách các thành phần được định tuyến và đặt chúng trong thư mục này.
* **App.vue:** Đây là entry point component. Là file khởi tạo tất cả các component khác.
* **main.js:** Entry point file để mount App.vue. Đây chính là file render ra App.vue component.

### 2.1.4. Java Spring boot

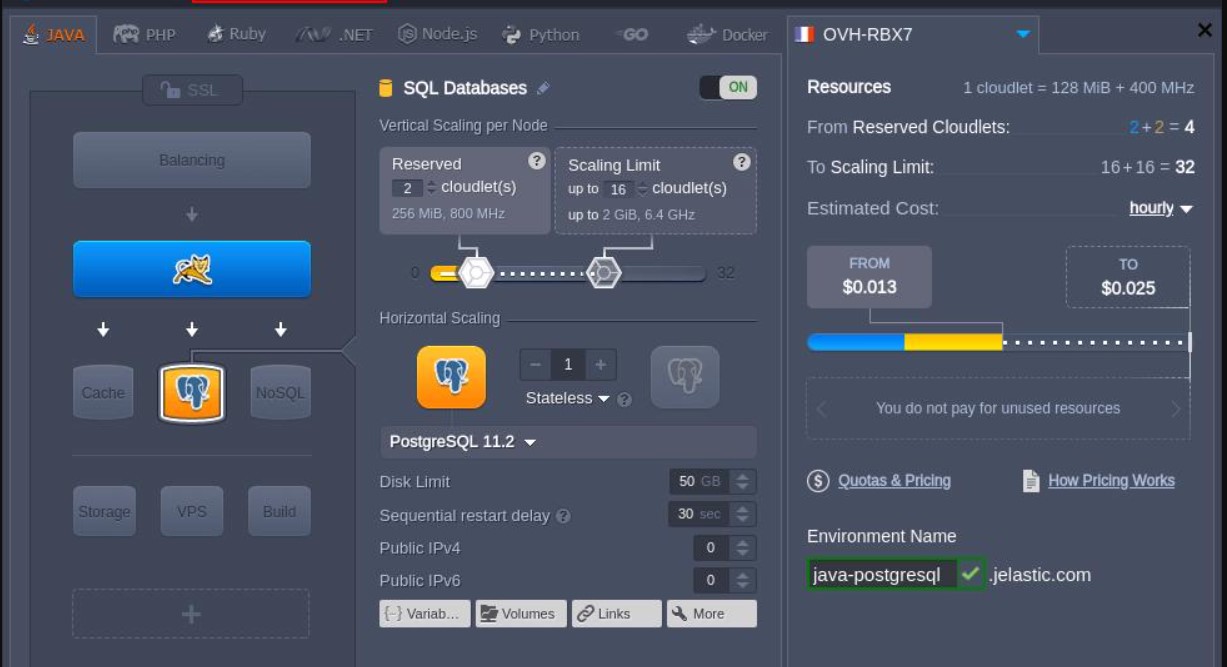


Hình 2.4 Java

### 2.1.5. PostgreSQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và nguồn mở RDBMS tiên tiến nhất hiện nay. khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Nó được thiết kế để xử lý một loạt các khối lượng công việc lớn, từ các máy tính cá nhân đến kho dữ liệu hoặc dịch vụ Web có nhiều người dùng đồng thời.

Một vài tính năng cơ bản của PostgreSQL:



Hình 2.5 PostgreSQL

* **Cung cấp nhiều kiểu dữ liệu**: PostgreSQL cung cấp đa dạng kiểu dữ liệu như nguyên hàm, cấu trúc, hình học,..
* **Bảo đảm toàn vẹn dữ liệu**: Dữ liệu trong PostgreSQL luôn được đảm bảo tính toàn vẹn.
* **Tính năng thiết lập linh hoạt**: PostgreSQL cho phép người dùng thiết lập danh mục từ cơ bản đến nâng cao, tối ưu ưu hóa tốc độ truy cập, hỗ trợ thống kê trên nhiều cột,.. Cùng với đó là vô số thiết lập khác.
* **Chức năng bảo mật**: PostgreSQL hỗ trợ xây dựng hàng rào bảo mật, xác thực mạnh.
* **Khả năng mở rộng**: Người dùng có thể thực hiện mở rộng hệ thống qua các phức thức lưu trữ, kết nối cơ sở dữ liệu.
* **Chức năng tìm kiếm văn bản**: PostgreSQL cung cấp tính năng tìm kiếm văn bản đầy đủ, hệ thống hóa ký tự theo cách khoa học.

## 2.2. Công cụ sử dụng

### 2.2.1. Các công cụ sử dụng cho Website

* Visual Studio Code: Phục vụ mục đích code VueJS
* PostgreSQL: Cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* Postman: Công cụ để thử nghiệm API.
* IntelliJ: Phục vụ mục đích code Java
* Chrome DevTools: gỡ lỗi khi phát triển web.
* Drawio: thiết kế hệ thống.
* Git, Github: Quản lý mã nguồn

### 2.2.2 Các công cụ sử dụng cho bài toán AI check in bằng khuôn mặt

**Môi trường Anaconda.**

Anaconda là một môi trường phát triển và quản lý gói phần mềm mã nguồn mở, được sử dụng rộng rãi trong việc phân tích dữ liệu và phát triển các ứng dụng khoa học dữ liệu. Nó cung cấp một bộ công cụ linh hoạt và mạnh mẽ để làm việc với Python và các ngôn ngữ khác như R và Julia.



Hình 2.7. Logo Anaconda

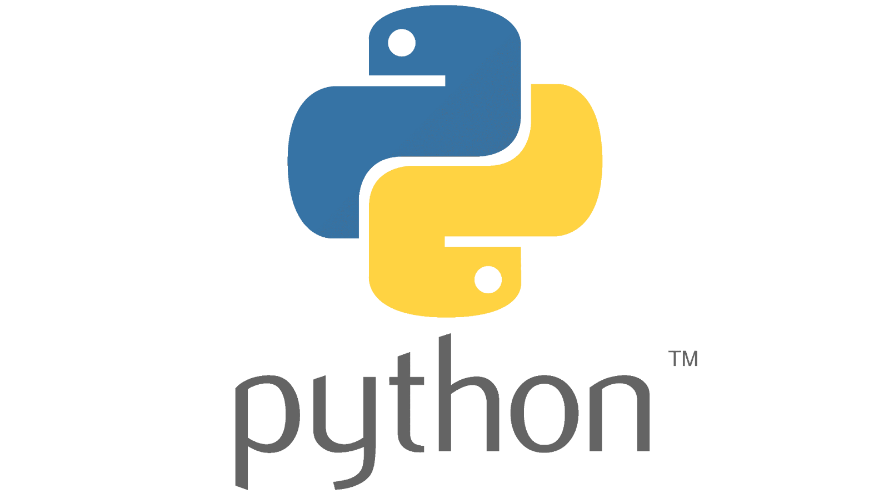
Dưới đây là một số đặc điểm chính của môi trường Anaconda:

* Quản lý gói phần mềm: Anaconda cung cấp một hệ thống quản lý gói phần mềm mạnh mẽ cho Python và các ngôn ngữ khác. Người dùng có thể dễ dàng cài đặt, cập nhật và xóa bỏ các gói phần mềm từ một kho lưu trữ trực tuyến.
* Môi trường ảo: Anaconda cho phép bạn tạo và quản lý các môi trường ảo, giúp tách biệt các phiên bản và phụ thuộc của các gói phần mềm. Điều này cho phép bạn cài đặt các phiên bản khác nhau của các gói và thử nghiệm ứng dụng mà không ảnh hưởng đến môi trường hệ thống chính.
* Công cụ tích hợp: Anaconda đi kèm với một số công cụ tích hợp hữu ích như Jupyter Notebook, JupyterLab và Spyder. Điều này cho phép bạn làm việc trực tiếp trong môi trường phát triển tích hợp, cung cấp khả năng tương tác và khám phá dữ liệu thuận tiện.
* Hỗ trợ đa nền tảng: Anaconda hỗ trợ đa nền tảng, cho phép bạn sử dụng môi trường và các công cụ của nó trên các hệ điều hành phổ biến như Windows, macOS và Linux.

Conda giúp chúng ta quản lý thư viện một cách dễ dàng hơn, kiểm tra các phiên bản sao cho phù hợp và có thể tách mỗi project là một môi trường hoàn toàn độc lập với nhau.

**Ngôn ngữ Python**

Python là một ngôn ngữ lập trình thông dịch, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Nó có cú pháp đơn giản, dễ đọc và dễ hiểu, làm cho việc viết mã Python trở nên dễ dàng cho cả người mới học lập trình và những người có kinh nghiệm.



Hình 2.8. Logo python

Dưới đây là một số đặc điểm và tính năng chính của Python:

* Đa mục đích: Python là một ngôn ngữ đa mục đích, có thể được sử dụng để phát triển ứng dụng web, ứng dụng di động, phân tích dữ liệu, máy học (machine learning), trí tuệ nhân tạo (artificial intelligence), và nhiều lĩnh vực lập trình khác.
* Cú pháp đơn giản: Python có cú pháp rõ ràng và đơn giản, với việc sử dụng khoảng trắng để định dạng mã nguồn. Điều này làm cho mã Python dễ đọc và dễ hiểu, giúp tăng tính tương tác và sáng tạo của các nhà lập trình.
* Hỗ trợ thư viện phong phú: Python đi kèm với một số thư viện và module chuẩn mạnh mẽ, cho phép bạn thực hiện nhiều tác vụ thông qua các hàm sẵn có mà không cần phải viết lại từ đầu. Ngoài ra, Python còn có cộng đồng lớn và năng động, sản sinh ra nhiều thư viện bên thứ ba phong phú và mạnh mẽ, từ đó hỗ trợ cho nhiều lĩnh vực và ứng dụng khác nhau.
* Hỗ trợ đa nền tảng: Python là ngôn ngữ đa nền tảng, có thể chạy trên các hệ điều hành phổ biến như Windows, macOS và Linux. Điều này cho phép bạn phát triển ứng dụng Python một cách linh hoạt trên nhiều môi trường.
* Mạnh trong phân tích dữ liệu và máy học: Python là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực phân tích dữ liệu và máy học. Có các thư viện mạnh mẽ như NumPy, Pandas, Matplotlib, scikit-learn và TensorFlow hỗ trợ việc xử lý dữ liệu, phân tích, và xây dựng mô hình máy học.
* Học và sử dụng dễ dàng: Python là một ngôn ngữ lập trình rất dễ học và sử dụng. Cú pháp đơn giản và rõ ràng giúp người mới học nắm bắt nhanh chóng, và cộng đồng Python rất hướng dẫn và hỗ trợ. Ngoài ra, Python cũng có tài liệu phong phú và rất nhiều nguồn tài nguyên trực tuyến, giúp bạn giải quyết vấn đề và tìm kiếm thông tin dễ dàng.
* Tích hợp và mở rộng: Python có khả năng tích hợp với các ngôn ngữ lập trình khác như C/C++, Java, và .NET. Bạn có thể sử dụng Python để viết các phần mở rộng cho ứng dụng sử dụng các ngôn ngữ khác hoặc sử dụng các thư viện có sẵn để tận dụng các tính năng mạnh mẽ từ các ngôn ngữ khác.
* Độ tin cậy và ổn định: Python là một ngôn ngữ ổn định và đáng tin cậy. Nó có cộng đồng lớn và sự hỗ trợ liên tục, giúp người dùng giải quyết các vấn đề và tìm kiếm giải pháp khi gặp lỗi.

Tổng quan, Python là một ngôn ngữ lập trình phổ biến và mạnh mẽ, với cú pháp đơn giản, hỗ trợ thư viện phong phú, và rất dễ học và sử dụng. Nó đã trở thành một công cụ quan trọng trong nhiều lĩnh vực, từ phân tích dữ liệu đến phát triển web và trí tuệ nhân tạo.

**Thư viện Pytorch**

Hình 2.9. Logo Pytorch

PyTorch là một thư viện mã nguồn mở cho việc xây dựng và huấn luyện mô hình học sâu. Nó được phát triển bởi Facebook AI Research và hỗ trợ các tính năng quan trọng để triển khai các mô hình học máy và trí tuệ nhân tạo.

Dưới đây là một tổng quan về các tính năng chính của PyTorch:

* Dynamic Computation Graph: PyTorch sử dụng đồ thị tính toán động, cho phép người dùng định nghĩa và thay đổi các phép tính trong quá trình chạy mà không cần xây dựng lại đồ thị từ đầu. Điều này tạo điều kiện thuận lợi cho việc debug và tạo ra các mô hình phức tạp.
* Tích hợp với Python: PyTorch được thiết kế để tương tác mượt mà với Python và các thư viện phổ biến khác như NumPy. Điều này giúp người dùng dễ dàng sử dụng và tích hợp PyTorch vào quy trình làm việc hiện có của mình.
* Cung cấp các lớp và hàm cơ bản: PyTorch cung cấp một loạt các lớp và hàm cơ bản để xây dựng mô hình học sâu, bao gồm các lớp và hàm cho việc tạo mạng nơ-ron, hàm kích hoạt, lớp tích chập, lớp pooling, và nhiều hơn nữa. Điều này giúp tiết kiệm thời gian và công sức trong việc triển khai mô hình.
* Hỗ trợ GPU: PyTorch hỗ trợ tính toán trên GPU, cho phép tăng tốc độ huấn luyện và triển khai mô hình lên GPU. Người dùng có thể dễ dàng di chuyển dữ liệu và mô hình giữa CPU và GPU sử dụng PyTorch.

PyTorch cung cấp một giao diện trực quan và dễ sử dụng cho việc xây dựng và huấn luyện mô hình học sâu. Nó đã trở thành một trong những thư viện phổ biến và được ưa chuộng trong cộng đồng nghiên cứu và ứng dụng học sâu. Điểm mạnh của PyTorch bao gồm:

* Tính linh hoạt: PyTorch cho phép người dùng tạo và tùy chỉnh các mô hình học sâu theo ý muốn. Người dùng có thể dễ dàng xác định các lớp, hàm kích hoạt, hàm mất mát và tối ưu hóa, và thậm chí tạo ra các kiến trúc mô hình phức tạp.
* Hỗ trợ tính toán tự động (Autograd): PyTorch tự động tính đạo hàm cho các phép tính trong mô hình, giúp người dùng dễ dàng thực hiện lan truyền ngược và cập nhật trọng số mô hình. Điều này giúp rút ngắn quá trình triển khai mô hình và tối ưu hóa.
* Giao diện dễ sử dụng: PyTorch có cú pháp đơn giản và trực quan, giúp người dùng dễ dàng hiểu và sử dụng các hàm và lớp của thư viện. Nó cũng cung cấp nhiều hàm tiện ích và công cụ hỗ trợ cho việc xử lý dữ liệu và triển khai mô hình.
* Tích hợp với các thư viện khác: PyTorch được tích hợp tốt với các thư viện phổ biến khác trong hệ sinh thái Python như NumPy và SciPy. Điều này giúp người dùng dễ dàng làm việc với dữ liệu số, thực hiện các phép tính mảng và tận dụng các công cụ phân tích dữ liệu mạnh mẽ.
* Hỗ trợ đa nền tảng: PyTorch có thể hoạt động trên nhiều nền tảng khác nhau, bao gồm Linux, Windows và macOS. Nó cũng hỗ trợ các kiến trúc máy tính khác nhau như CPU và GPU.

Tổng quan, PyTorch là một thư viện mạnh mẽ và linh hoạt cho việc xây dựng và huấn luyện mô hình học sâu. Nó đã thu hút sự quan tâm và đóng góp của một cộng đồng lớn, và tiếp tục phát triển và cải tiến để đáp ứng các yêu cầu của người dùng.

**Thư viện FastAPI**

FastAPI là một framework phát triển ứng dụng web nhanh và dễ sử dụng, được xây dựng trên nền tảng Starlette và hỗ trợ Python 3.7+.



Hình 2.10. Logo FastAPI

Dưới đây là một tổng quan về một số tính năng chính của FastAPI:

* Tích hợp dễ dàng: FastAPI tích hợp tốt với các thư viện và công cụ phổ biến như Pydantic, SQLAlchemy, Tortoise ORM, OAuth, etc. Điều này giúp việc phát triển ứng dụng web trở nên dễ dàng và tiện lợi.
* Hiệu suất cao: FastAPI được xây dựng với hiệu suất cao như giao thức ASGI và asyncio. Điều này cho phép xử lý các yêu cầu đồng thời một cách hiệu quả và đáp ứng nhanh chóng.
* Validation và Serialization tự động: FastAPI sử dụng Pydantic, một thư viện validation và serialization mạnh mẽ, để kiểm tra và xử lý dữ liệu đầu vào và đầu ra của API tự động. Việc này giúp giảm lỗi và tăng tính đáng tin cậy của ứng dụng.
* Tài liệu tự động và giao diện API tương tác: FastAPI tự động tạo tài liệu API từ các chú thích trong mã nguồn và cung cấp giao diện tương tác để thử nghiệm và tài liệu hóa API. Điều này giúp giảm thời gian phát triển và cung cấp một cách dễ dàng để khám phá và sử dụng API.
* Hỗ trợ WebSockets: FastAPI hỗ trợ giao thức WebSockets, cho phép xây dựng ứng dụng thời gian thực và truyền tải dữ liệu hai chiều giữa máy khách và máy chủ.
* Cấu hình đơn giản: FastAPI sử dụng Python Typing để kiểm tra kiểu dữ liệu đầu vào và đầu ra, giúp tạo ra mã nguồn rõ ràng và dễ đọc. Cấu hình đơn giản và ngắn gọn giúp làm tăng tính bảo mật và khả năng bảo trì của ứng dụng.

Với những tính năng mạnh mẽ này, FastAPI đã trở thành một lựa chọn phổ biến cho việc phát triển các ứng dụng web Python nhanh chóng và hiệu quả.

# Chương 3. Phân tích thiết kế hệ thống

## 3.1. Phân tích hệ thống

### 3.1.1. Vai trò người dùng

* Hệ thống chia làm 3 nhóm người dùng chính
* Người quản trị
* Trưởng phòng
* Nhân viên
* Mô tả nhóm người dùng:
* Người quản trị: Là người quản lí hệ thống và được truy cập cập vào chức năng quản lý chấm công, chấm công tự động bằng khuôn mặt, quản lý tài khoản, quản lý phòng ban
* Trường phòng: Là người được truy cập vào các chức năng chấm công thủ công, đăng ký vắng mặt, phê duyệt
* Nhân viên: Là ngươi được truy cập vào các chức năng đăng ký vắng mặt, xem lịch sử chấm công

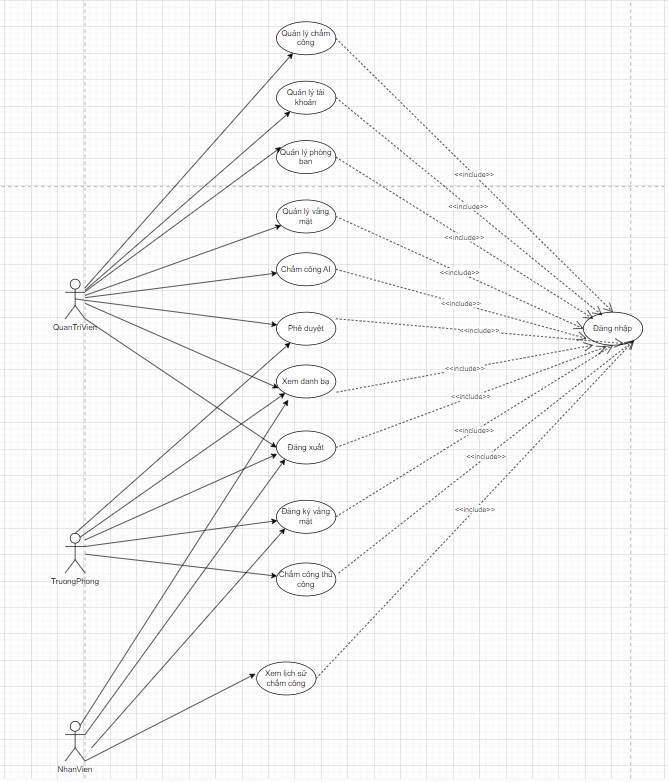
### 3.1.2. Yêu cầu chức năng

* Người quản trị
* Đăng nhập, đăng xuất
* Chấm công khuôn mặt
* Quản lý tài khoản
* Quản lý phòng ban
* Phê duyệt
* Trưởng phòng
* Đăng nhập, đăng xuất
* Đăng ký vắng mặt, phê duyệt
* Chấm công thủ công
* Nhân viên
* Đăng nhập, đăng xuất
* Đăng ký vắng mặt
* Xem lịch sử chấm công

### 3.1.3. Yêu cầu phi chức năng

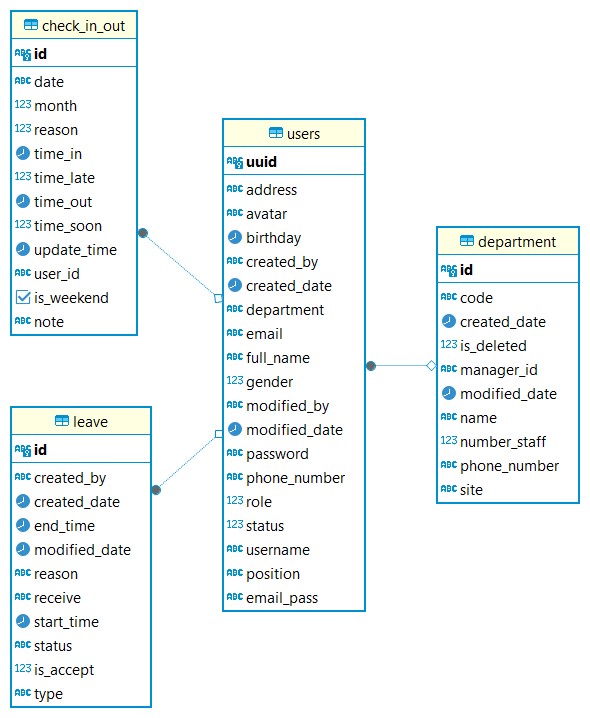
Thiết kế giao diện đơn giản: cung cấp giao diện dễ sử dụng và tối ưu hóa trải nghiệm người dùng.

## 3.2. Biểu đồ use case tổng quát



Hình 3.1. Sơ đồ use case tổng quát

## 3.3. Biểu đồ thực thể liên kết



Hình 3.2. Biểu đồ thực thể liên kết

## 3.4. Đặc tả use case

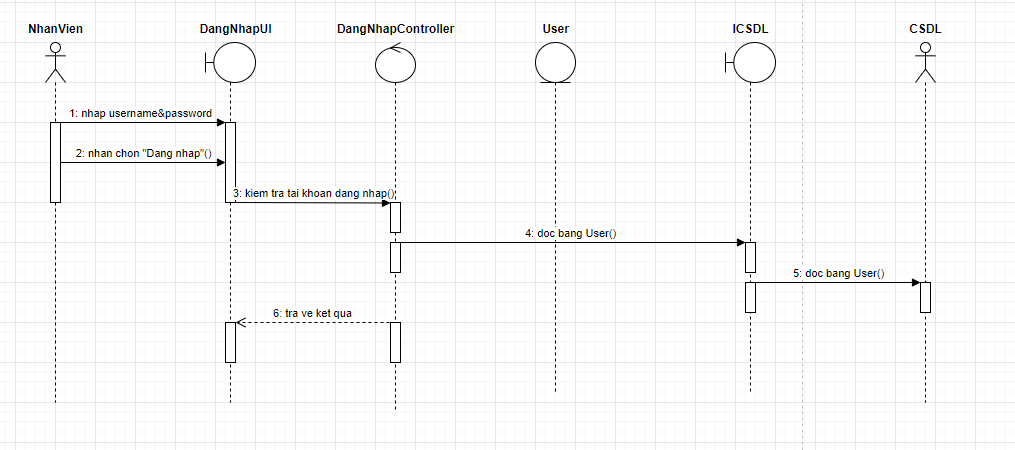
### 3.4.1. Use case Đăng nhập

1. Đặc tả use case

Use case cho phép người dùng hệ thống có thể đăng nhập tài khoản vào hệ thống

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi người dùng nhấn chọn “Đăng nhậpˮ
* Hệ thống yêu cầu người dùng nhập vào tên đăng nhập, mật khẩu.
* Người dùng nhập đầy đủ thông tin yêu cầu và nhấp chọn “Đăng nhậpˮ.
* Hệ thống kiểm tra thông tin người dùng trong bảng USER và chuyển vào giao diện màn hình chính, use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Tên đăng nhập hoặc mật khẩu không chính xác. Tại bước 3 luồng cơ bản, nếu người dùng nhập sai thông tin tài khoản, hệ thống sẽ hiển thị thông báo đăng nhập không thành công.
* Người dùng nhấp nút back, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: màn hình chuyển hướng vào trang chủ
* Điểm mở rộng: không có.

1. Biểu đồ trình tự



Hình 3.3. Biều đồ trình tự use case Đăng nhập

### 3.4.2. Use case đăng xuất

1. Đặc tả use case

Use case cho phép người dùng đăng xuất khỏi hệ thống một cách an toàn

* Luồng sự kiện
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng click vào nút Đăng xuất, hệ thống sẽ xoá phiên hoạt động và chuyển hướng đến màn hình đăng nhập.
* Use case kết thúc
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: màn hình chuyển hướng vào trang chủ
* Điểm mở rộng: không có.

1. Biều đồ trình tự

A picture containing diagram

Description automatically generated

Hình 3.4. Biểu đồ trình tự usecase đăng xuất

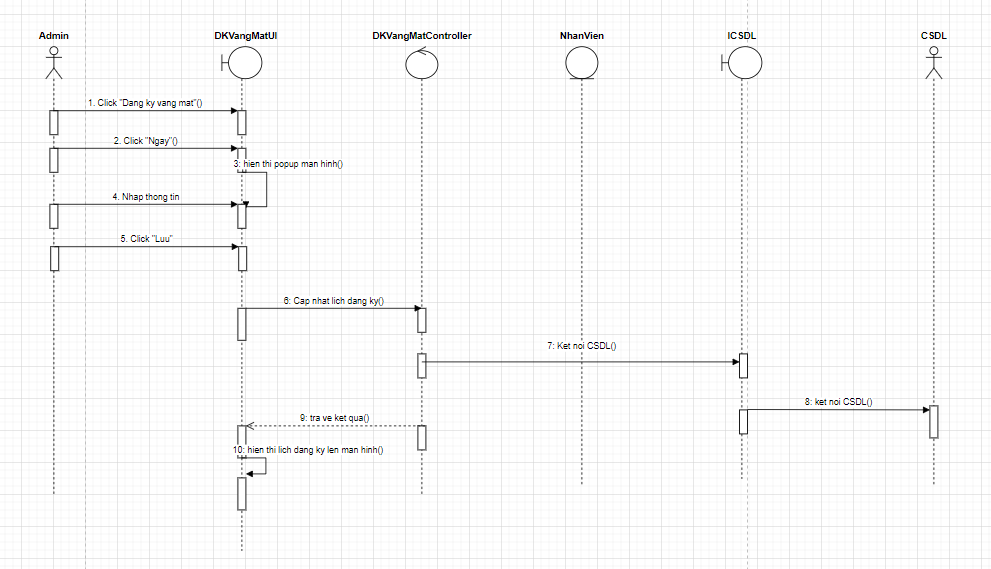
### 3.4.3. Use case Đăng ký vắng mặt

1. Đặc tả use case

Use case cho phép người dùng đăng xuất khỏi hệ thống một cách an toàn

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi người dùng nhấn chọn Ngày trong Schedule.
* Hệ thống hiển thị Form đăng ký lên màn hình
* Người dùng chọn trạng thái nghỉ, thời gian bắt đầu, kết thúc, nhập lý do nghỉ và chọn người duyệt là cấp trên.
* Người dùng click button “Lưuˮ
* Hệ thống kiểm tra và lưu thông tin mới cập nhật hiển thị thông tin nghỉ phép đã đăng ký lên màn hình danh sách Schedule.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện:
* Thông tin nghỉ phép của người dùng đã được lưu và hiển thị trong danh sách Schedule.
* Yêu cầu nghỉ phép được gửi đến cấp trên để phê duyệt.
* Điểm mở rộng: không có.

1. Biểu đồ trình tự



Hình 3.5. Biều đồ trình tự usecase đăng ký vắng mặt

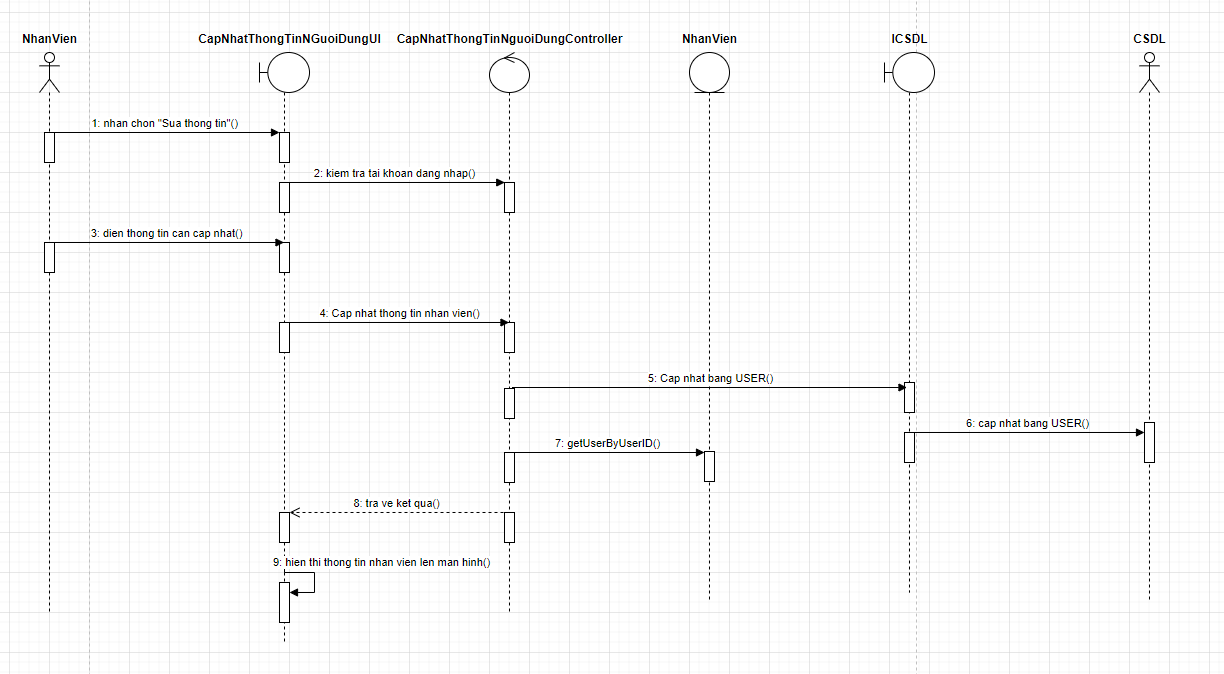
### 3.4.4. Use case Cập nhật thông tin tài khoản

1. Đặc tả use case

Use case cho phép người dùng cập nhật thông tin cá nhân trong tài khoản của mình thông qua giao diện màn hình

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi người dùng truy cập vào trang "Profile".
* Người dùng click button “Cập nhậtˮ
* Hệ thống hiển thị Form cập nhật thông tin tài khoản với các trường thông tin hiện tại của người dùng.
* Người dùng sửa đổi các thông tin cá nhân như họ tên, email, số điện thoại, địa chỉ,
* Người dùng click button “Lưuˮ
* Hệ thống kiểm tra tính hợp lệ của thông tin mới.
* Hệ thống cập nhật thông tin mới vào cơ sở dữ liệu và hiển thị thông báo "Cập nhật thành công".
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: Thông tin tài khoản của người dùng đã được cập nhật và lưu trữ trong cơ sở dữ liệu.

1. Biều đồ trình tự



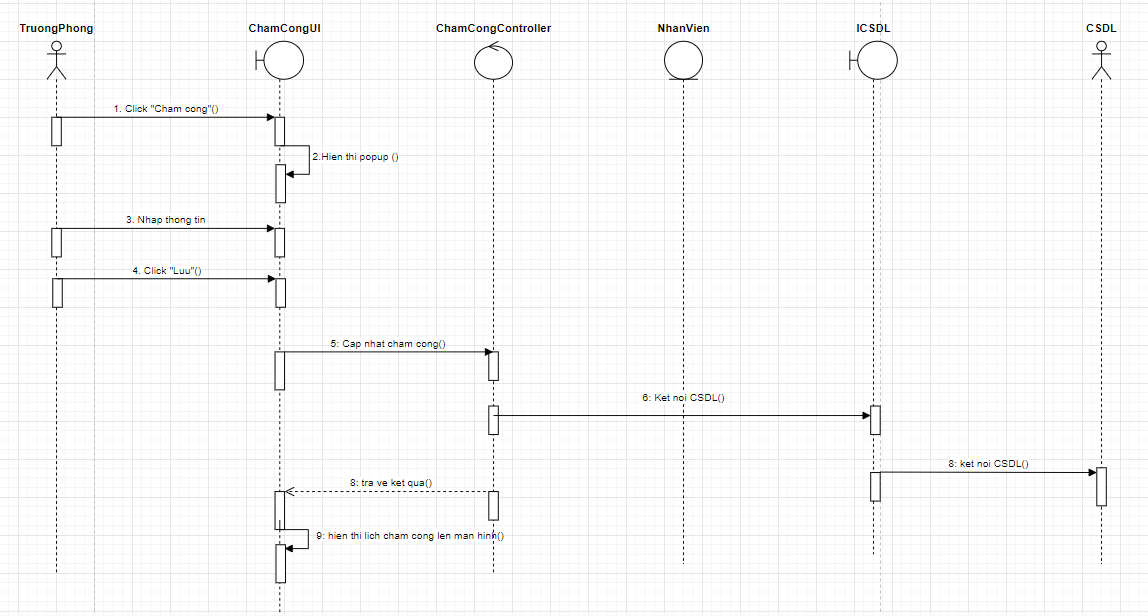
Hình 3.6. Biều đồ trình tự usecase cập nhật thông tin tài khoản

### 3.4.5. Use case Chấm công bằng khuôn mặt

1. Đặc tả use case

Use case cho nhân viên chấm công bằng khuôn mặt

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi Admin click vào trang chấm công bằng khuôn mặt.
* Hệ thống hiển thị camera lên màn hình
* Người dùng đưa mặt vào camera
* Hệ thống kiểm tra dữ liệu của người dùng.
* Hệ thống hiển thị thông báo "Check in thành công".
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: màn hình chuyển hướng vào trang chủ
* Điểm mở rộng: không có.



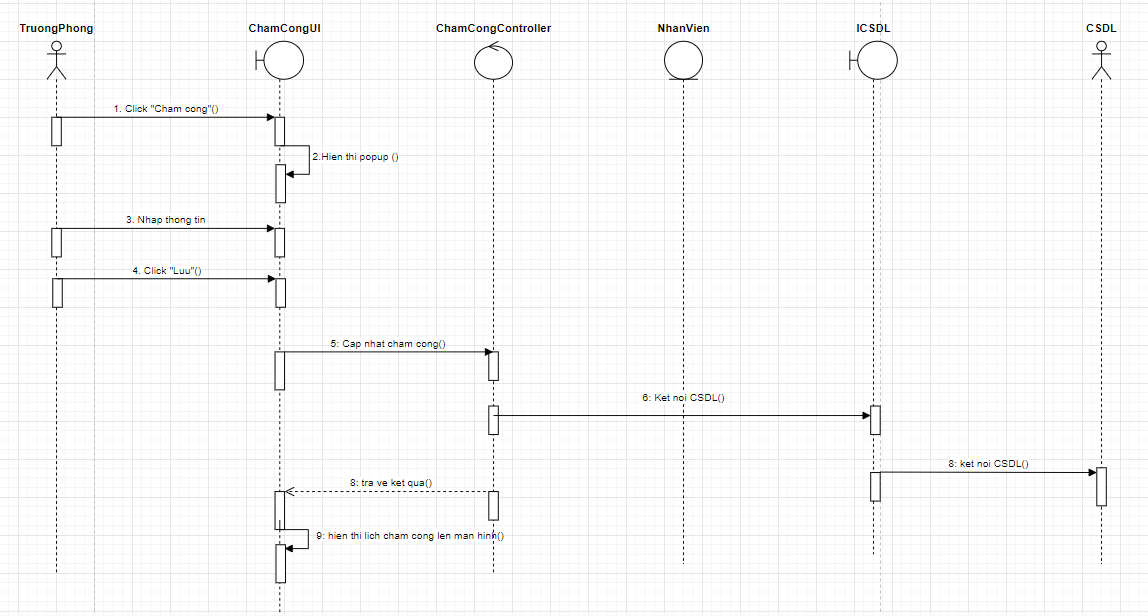
Hình 3.7. Biểu đồ trình tự usecsase chấm công khuôn mặt

### 3.4.6. Use case Chấm công thủ công

1. Đặc tả use case

Use case cho phép trưởng phòng chấm công cho nhân viên trên màn hình giao diện

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi Admin/Trưởng phòng click vào chấm công thủ công trên thanh menu.
* Admin/Trưởng phòng tìm nhân viên muốn chấm công trên hệ thống
* Hệ thống hiển thị dữ liệu chấm công của nhân viên Admin/Trưởng phòng click button “Chấm côngˮ.
* Hệ thống hiển thị Form timeout .
* Admin/Trưởng phòng chọn thời gian check in check out cho nhân viên
* Hệ thống update dữ liệu check in check out nhân viên lên màn hình Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: màn hình chuyển hướng vào trang chủ
* Điểm mở rộng: không có.



Hình 3.8. Biểu đồ trình tự usecase chấm công thủ công

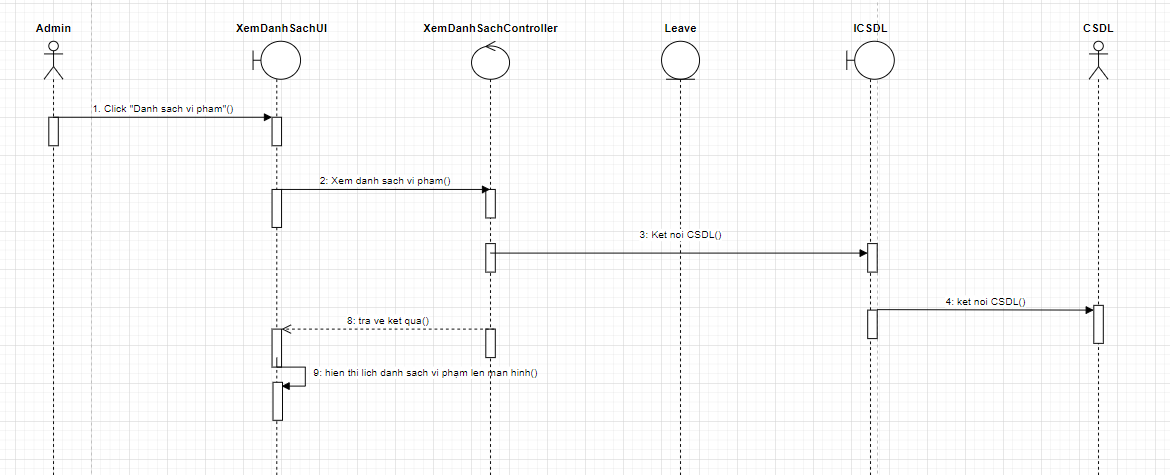
### 3.4.7. Use case Xem danh sách vi phạm

1. Đặc tả use case

Use case cho phép quản trị viên xem danh sách nhân viên vi phạm thông qua giao diện màn hình.

* Luồng sự kiện
* Luồng cơ bản:
* Use case bắt đầu khi Admin click vào “Danh sách vi phạmˮ trên thanh menu.
* Hệ thống hiển thị dữ liệu của nhân viên vi phạm bao gồm số phút đi muộn, số phút về sớm Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống.
* Hậu điều kiện: màn hình chuyển hướng vào trang chủ
* Điểm mở rộng: không có.

1. Biểu đồ trình tự



Hình 3.9. Biểu đồ trình tự usecase xem danh sách vi phạm

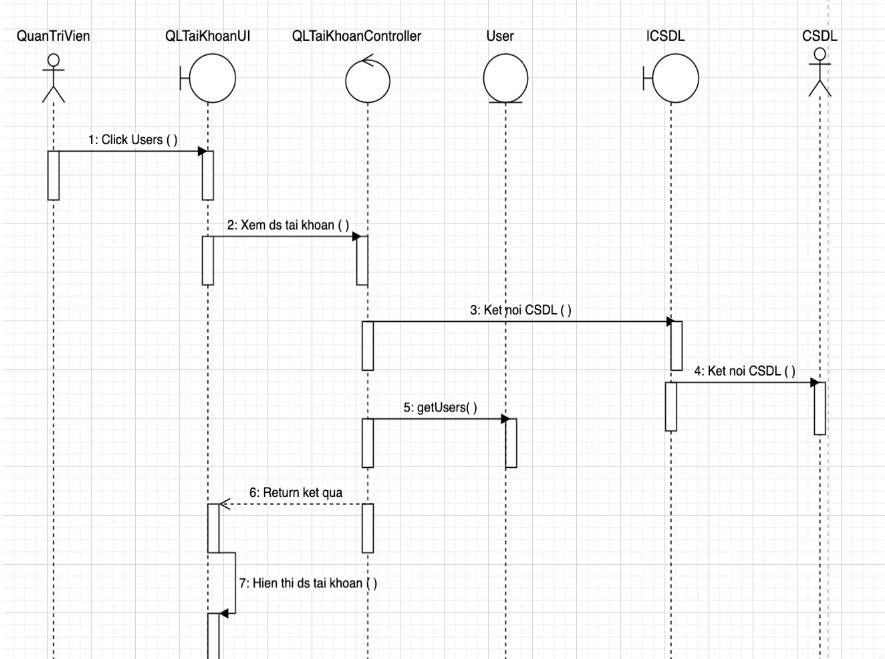
### 3.4.8. Use case Quản lý tài khoản

1. Đặc tả use case

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các người dùng trong bảng User.

* Luồng sự kiện
* Luồng cơ bản:
* Use case này bắt đầu khi người quản trị click vào nút Quản lý tài khoản trên thanh sidebar.
* Hệ thống hiển thị danh sách các tài khoản hiện có với các tùy chọn: Tạo mới, Cập nhật, Xem chi tiết, và Xóa tài khoản.
* Tạo mới tài khoản:
* Quản trị viên nhấn vào nút "Tạo mới".
* Hệ thống hiển thị Form tạo mới tài khoản
* Quản trị viên nhập các thông tin cần thiết như họ tên, email, tên đăng nhập, mật khẩu, quyền truy cập, v.v.
* Quản trị viên click button “Lưuˮ
* Hệ thống kiểm tra tính hợp lệ của thông tin.
* Quản trị viên quay lại trang "Quản lý tài khoản" Cập nhật tài khoản:
* Quản trị viên nhấn vào nút "Sửa" bên cạnh tài khoản cần chỉnh sửa.
* Hệ thống hiển thị Form cập nhật với thông tin hiện tại của tài khoản.
* Quản trị viên chỉnh sửa các thông tin cần thiết.
* Quản trị viên click button “Lưuˮ.
* Hệ thống kiểm tra tính hợp lệ của thông tin
* Quản trị viên quay lại trang "Quản lý tài khoản".
* Xem chi tiết tài khoản:
* Quản trị viên nhấn vào nút "Xem" bên cạnh tài khoản muốn xem.
* Hệ thống hiển thị thông tin chi tiết của tài khoản.
* Quản trị viên quay lại trang "Quản lý tài khoản".
* Xóa tài khoản:
* Quản trị viên nhấn vào nút "Xóa" bên cạnh tài khoản muốn xóa.
* Hệ thống hiển thị hộp thoại xác nhận xóa tài khoản.
* Quản trị viên xác nhận xóa tài khoản.
* Hệ thống xóa tài khoản khỏi cơ sở dữ liệu và hiển thị thông báo "Xóa tài khoản thành công".
* Quản trị viên quay lại trang "Quản lý tài khoản".
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Đăng nhập bằng role Admin vào hệ thống
* Hậu điều kiện: Các thay đổi về tài khoản đã được lưu trữ thành công trong cơ sở dữ liệu.
* Điểm mở rộng: không có.

1. Biểu đồ trình tự



Hình 3.10. Biểu đồ trình tự usecase quản lý tài khoản

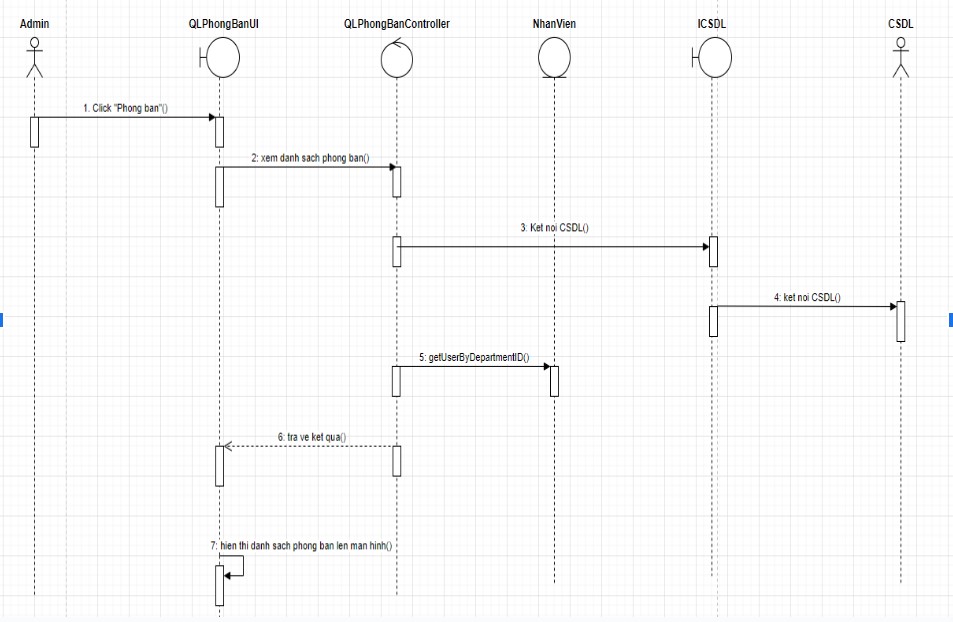
### 3.4.9. Use case Quản lý phòng ban

1. Đặc tả use case

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các phòng ban trong bảng PhongBan.

* Luồng sự kiện
* Luồng cơ bản:
* Use case này bắt đầu khi người quản trị click vào nút Quản lý phòng ban trên thanh sidebar.
* Hệ thống hiển thị danh sách các phòng ban hiện có với các tùy chọn: Tạo mới, Cập nhật, Xem chi tiết, và Xóa tài khoản.
* Tạo mới phòng ban:
* Quản trị viên nhấn vào nút "Tạo mới".
* Hệ thống hiển thị Form tạo mới phòng ban
* Quản trị viên nhập các thông tin cần thiết như tên phòng ban, vị trí, mã phòng ban,số điện thoại .
* Quản trị viên click button “Lưuˮ
* Hệ thống kiểm tra tính hợp lệ của thông tin.
* Quản trị viên quay lại trang "Quản lý phòng ban" Cập nhật phòng ban:
* Quản trị viên nhấn vào nút "Sửa" bên cạnh phòng ban cần chỉnh sửa.
* Hệ thống hiển thị Form cập nhật với thông tin hiện tại của phòng ban.
* Quản trị viên chỉnh sửa các thông tin cần thiết.
* Quản trị viên click button “Lưuˮ.
* Hệ thống kiểm tra tính hợp lệ của thông tin
* Quản trị viên quay lại trang "Quản lý phòng ban".
* Xem chi tiết phòng ban:
* Quản trị viên nhấn vào nút "Xem" bên cạnh phòng ban muốn xem.
* Hệ thống hiển thị thông tin chi tiết của phòng ban.
* Quản trị viên quay lại trang "Quản lý phòng ban".
* Xóa tài khoản:
* Quản trị viên nhấn vào nút "Xóa" bên cạnh phòng ban muốn xóa.
* Hệ thống hiển thị hộp thoại xác nhận xóa phòng ban.
* Quản trị viên xác nhận xóa phòng ban.
* Hệ thống xóa tài khoản khỏi cơ sở dữ liệu và hiển thị thông báo "Xóa phòng ban thành công".
* Quản trị viên quay lại trang "Quản lý phòng ban".
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt: không có.
* Tiền điều kiện: Đăng nhập bằng role Admin vào hệ thống
* Hậu điều kiện: Các thay đổi về phòng ban đã được lưu trữ thành công trong cơ sở dữ liệu.
* Điểm mở rộng: không có.

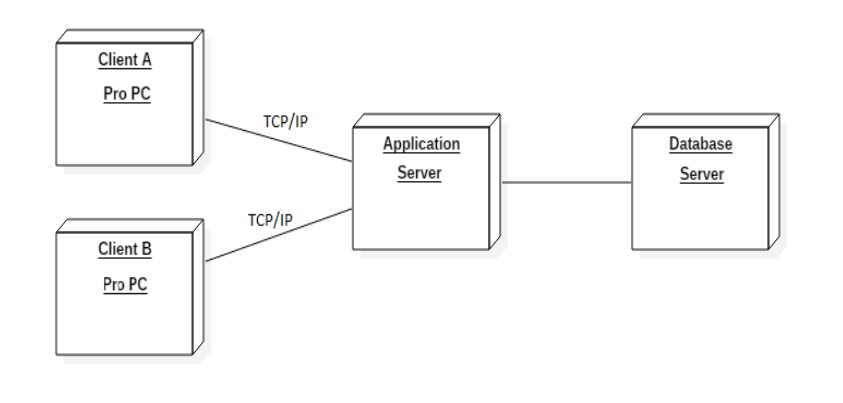
1. Biểu đồ trình tự



Hình 3.11. Biểu đồ trình tự usecase quản lý phòng ban

## 3.5. Thiết kế hệ thống

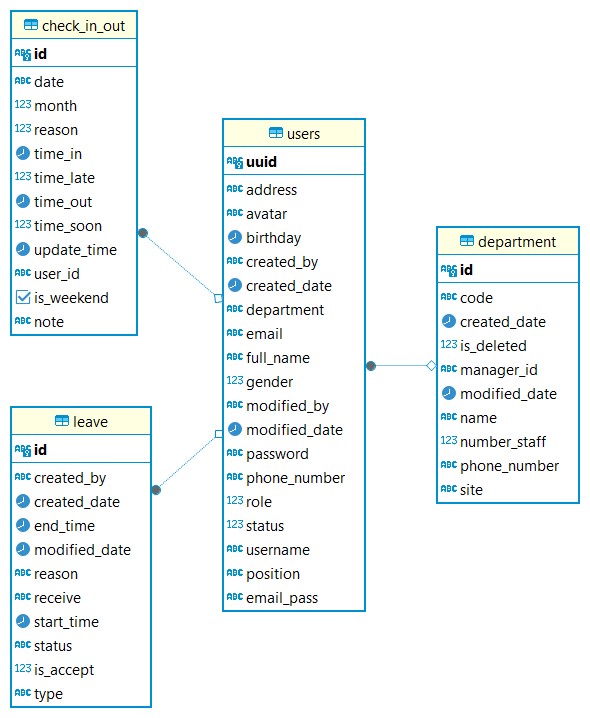
### 3.5.1. Biểu đồ triển khai hệ thống



Hình 3.12. Biểu đồ triển trai hệ thống

### 3.5.2. Thiết kế cơ sở dữ liệu

Biểu đồ mô tả cấu trúc cơ sở dữ liệu



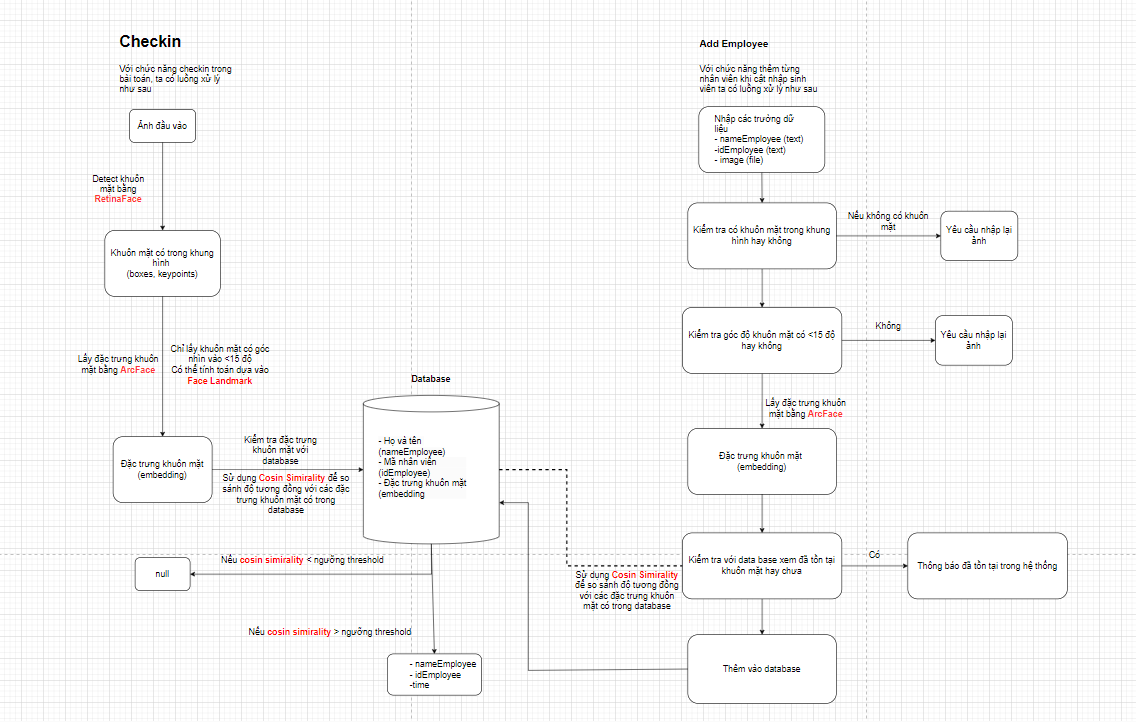
Hình 3.13. Biểu đồ cấu trúc cơ sở dữ liệu

### 3.5.3. Biểu đồ hoạt động bài toán AI check in khuôn mặt

Với chức năng như nhận dạng khuôn mặt (tự động checkin):

* Sau khi thu được hình ảnh từ camera 🡪 hình ảnh được truyền vào server AI để tiến hành nhận dạng
* Hình ảnh sau khi được truyền vào server AI sẽ được đưa qua mô hình RetinaFace để tiến hành nhận diện khuôn mặt, tại nước này ta sẽ thu được các bounding boxes của khuôn mặt có trong hình 🡪 Bước này ta sẽ xử lý khuôn mặt có kích thước to nhất (đứng gần camera nhất)

Sau khi thu được bounding box và keypoint của khuôn mặt thì ta tiến hành xác thực khuôn mặt, dựa vào keypint ta sẽ tính toán được góc độ của khuôn mặt. Sau khi xác thực là có khuôn mặt thỏa mãn thì tiếp tục đưa 2 output này vào trong mô hình trích xuất đặc trưng là ArcFace. Ở bước này thì ta sẽ thu được 1 vector embedding tương ứng của khuôn mặt đã xác định là đang check in.



Hình 3.14. Sơ đồ hoạt động bài toán AI check in khuôn mặt

* Hình ảnh khuôn mặt đã được trích xuất lấy đặc trưng và biểu diễn bởi 1 vector embedding, ta tiến hành so sánh với các vector đặc trưng có trong database (ở đây là tập hợp các đặc trưng của hình ảnh khuôn mặt nhân viên đã đăng kí trước đó)
* Áp dụng công thức cosin simirality để so sánh lần lượt các đặc trưng để xét xem đặc trưng khuôn mặt đầu vào có trùng khớp với khuôn mặt nào trong database không. Ở bước này ta sẽ xác định một ngưỡng cụ thể để đánh giá độ trùng khớp của đặc trưng khuôn mặt đầu vào với các đặc trưng khuôn mặt đã có trong database
* Nếu không 🡪 Trả ra null 🡪 Xác định người này không phải là nhân viên của công ty
* Nếu có 🡪 Tuy xuất thông tin của nhân viên đó và trả ra thông tin check in 🡪 Xác định nhân viên này đã check in

Với chức năng đăng kí khuôn mặt nhân viên:

* Ta cần nhập thông tin của nhân viên đó đính kèm hình ảnh chứa khuôn mặt
* Sau khi có yêu cầu thêm khuôn mặt mới, hệ thống AI sẽ kiểm tra lại hình ảnh xem đã thỏa mãn điều kiện của hệ thống hay chưa, với các điều kiện như hình ảnh chưa khuôn mặt và khuôn mặt phải đạt góc độ nhỏ hơn 15 độ, bên cạnh đó hệ thống sẽ kiểm tra xem khuôn mặt đăng kí mới đã tồn tại trong hệ thống hay chưa, nếu đã có khuôn mặt trong hệ thống thì không thể đăng kí. Ngược lại, nếu khuôn mặt chưa có trong hệ thống thì hoàn tất đăng kí

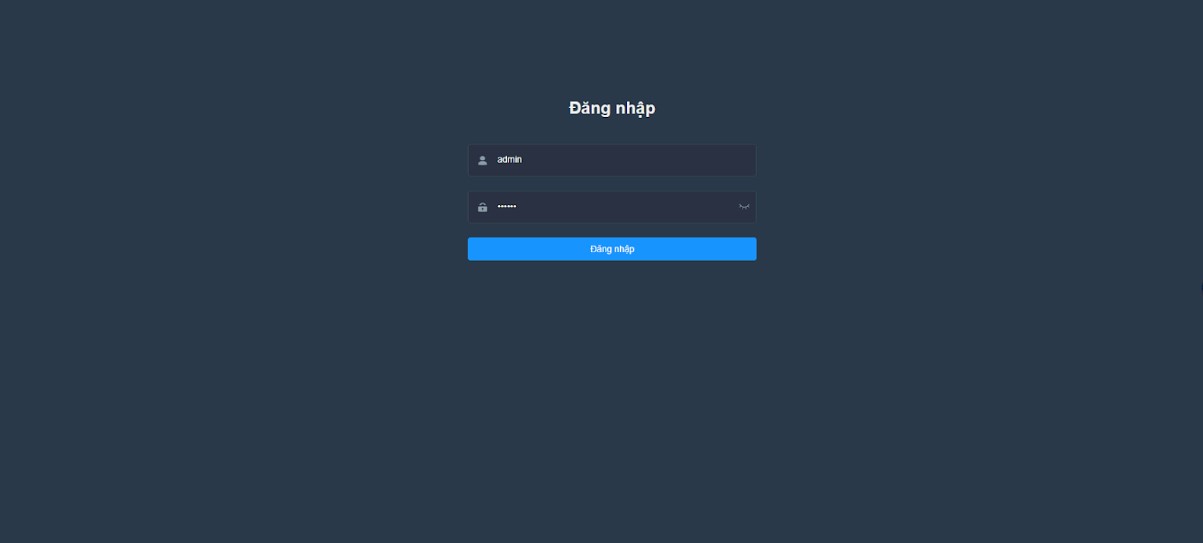
Với việc hệ thống AI nhận dạng khuôn mặt được hoạt động xoay quanh các vector đặc trưng của khuôn mặt được xử lý bởi các mô hình như RetinaFace, ArcFace, FaceLandmark và thuật toán so sánh vector là cosin simirality. Hệ thống hoàn toàn có thể hoạt động với chỉ cần đăng kí 1 hình ảnh duy nhất rõ nét khuôn mặt thì đặc trưng thu được càng rõ rang và dễ phân biệt.

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

## 4.1. Giao diện các màn hình trên hệ thống

Giao diện đăng ký được minh họa bằng ảnh dưới đây

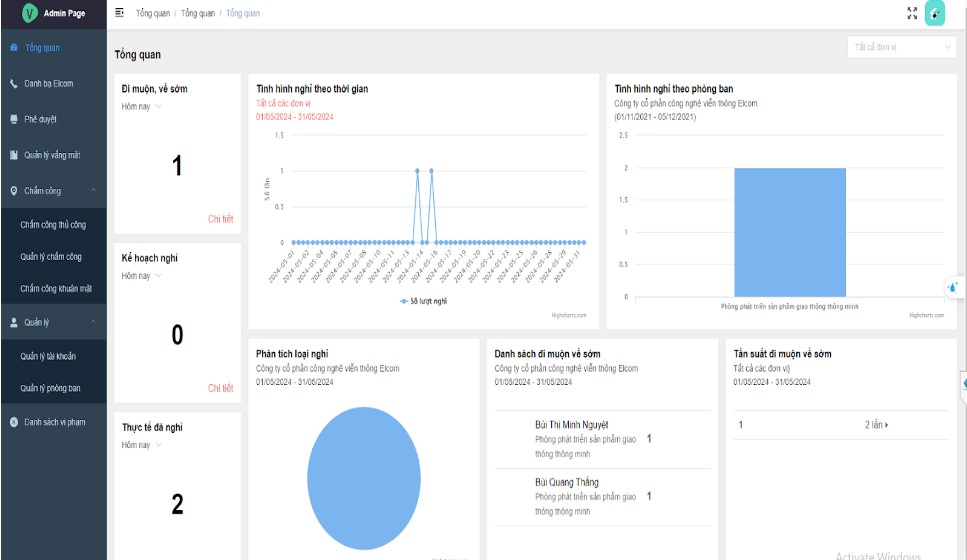
* Chức năng cho phép người dùng đăng nhập vào hệ thống



Hình 4.1. Giao diện màn hình đăng nhập

Giao diện trang chủ được minh họa bằng ảnh dưới đây

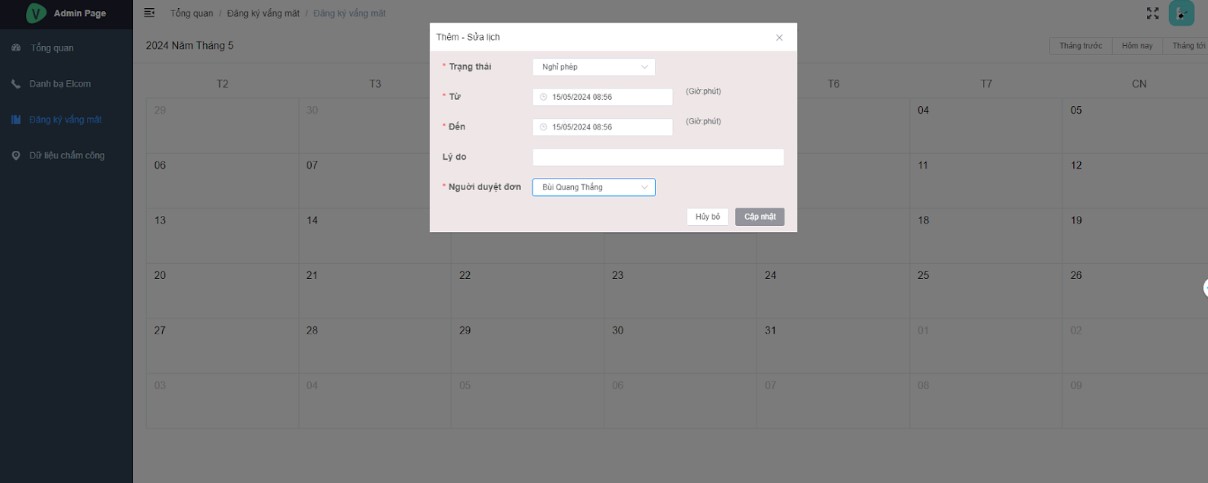
* Màn hình thống kê tình hình nghỉ , danh sách đi muộn về sớm, tần suất đi muộn về sớm. Tại đây người quản trị có thể xem thông tin tổng quát của từng loại.



Hình 4.2. Giao diện màn hình dashboard

Giao diện đăng ký vắng mặt được minh họa bằng ảnh dưới đây

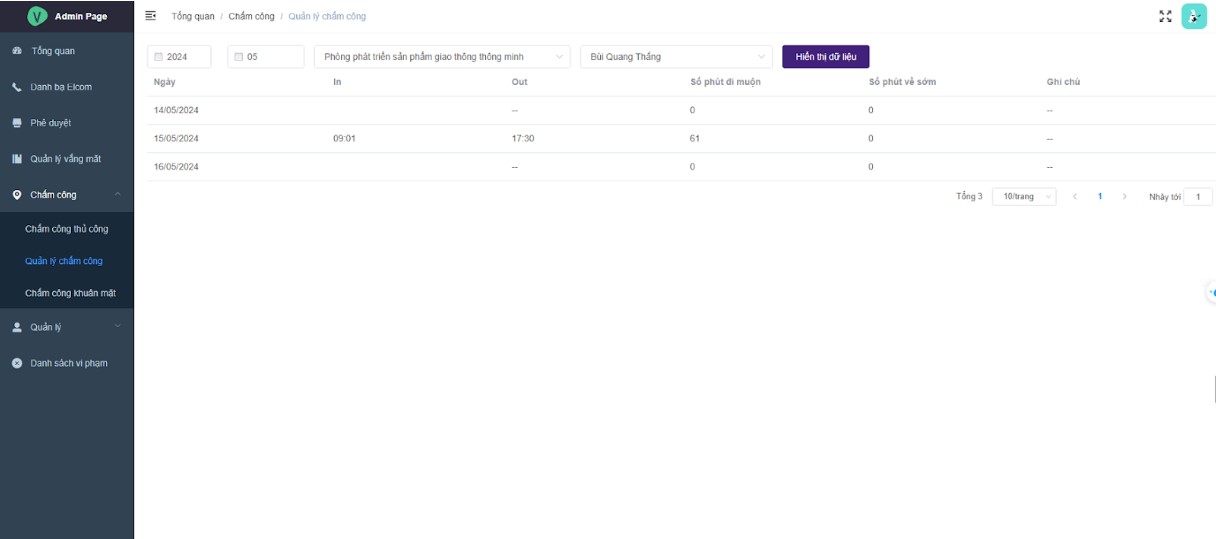
* Tại màn hình, trưởng phòng và nhân viên có thể đăng ký xin nghỉ phép, khi chọn ngày muốn xin nghỉ, hệ thống hiển thị popup gồm các thông tin như : trạng thái, thời gian, lý do và người duyệt đơn.



Hình 4.3. Giao diện màn hình Đăng ký vắng mặt

Giao diện Dữ liệu chấm công được minh họa bằng ảnh dưới đây

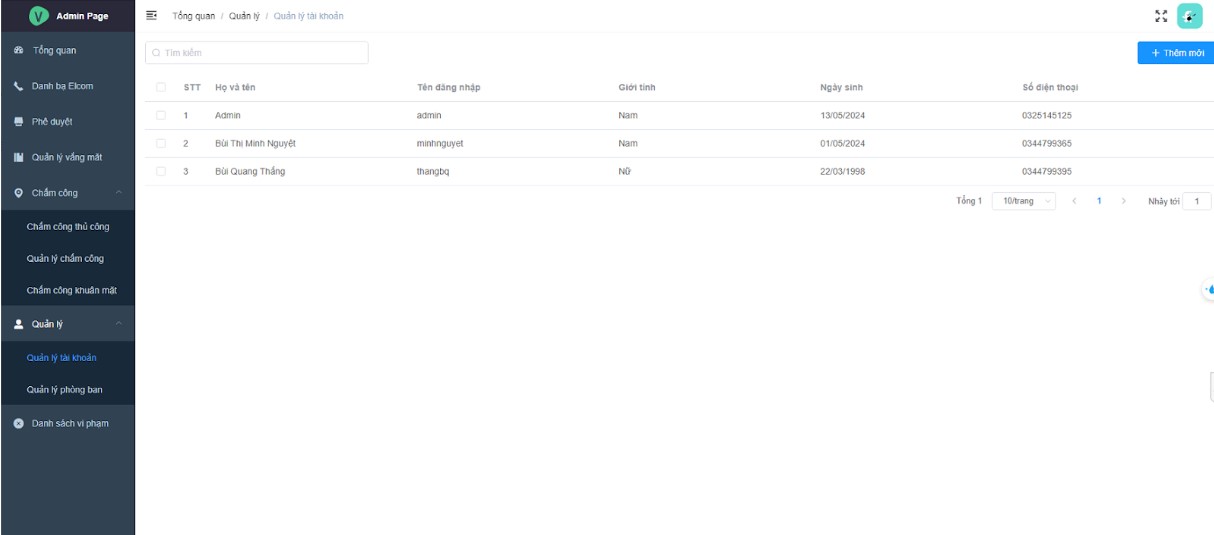
* Tại màn hình, sau khi chấm công thành công, dữ liệu chấm công của nhân viên sẽ được hiển thị đầy đủ thời gian checkin, checkout , số phút đi muộn, số phút về sớm. Ngoài ra khi đăng nhập với role Trưởng phòng/admin thì có thể xem thông tin dữ liệu chấm công của nhân viên cấp dưới



Hình 4.4. Giao diện màn hình Dữ liệu chấm công

Giao diện Quản lý tài khoản được minh họa bằng ảnh dưới đây

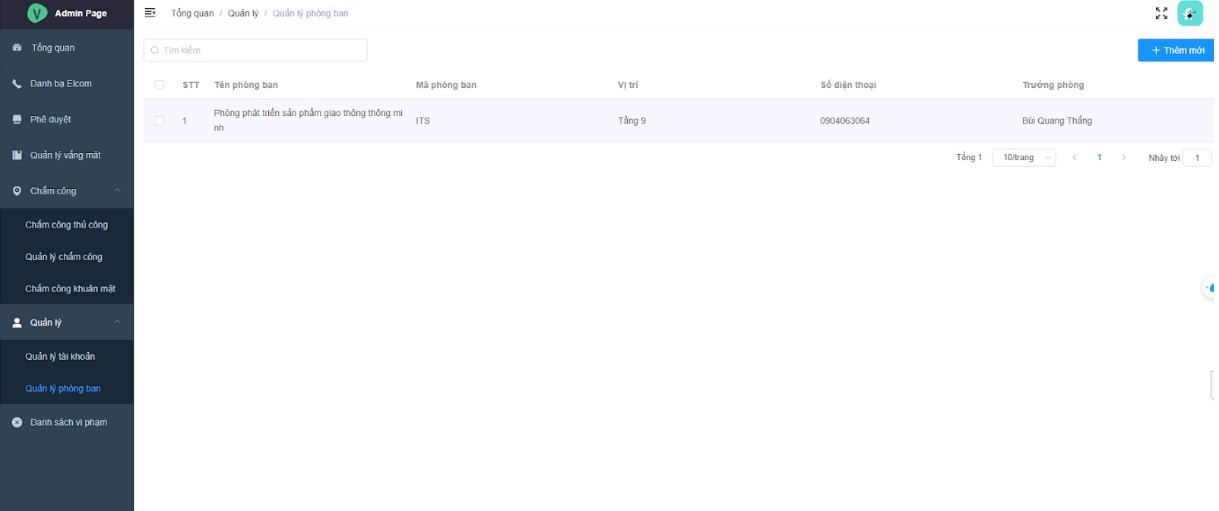
* Màn hình bao gồm các thông tin tài khoản của nhân viên, admin có quyền thêm mới, sửa, xóa nhân viên



Hình 4.5. Giao diện màn hình Quản lý tài khoản

Giao diện Quản lý phòng ban được minh họa bằng ảnh dưới đây

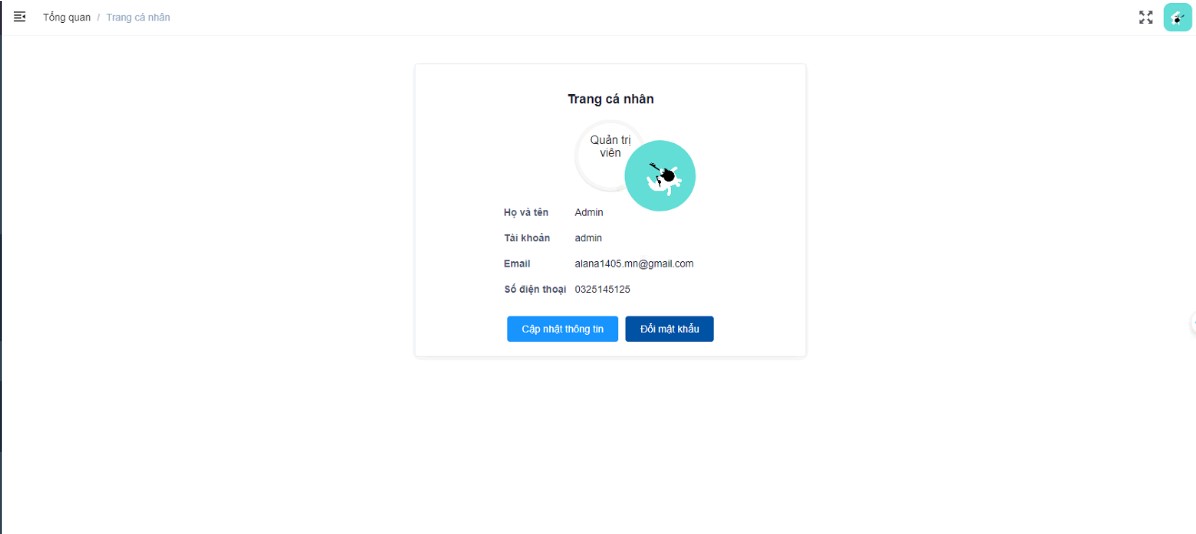
* Màn hình bao gồm các thông tin các phòng ban, admin có quyền thêm mới, sửa, xóa phòng ban



Hình 4.6.Giao diện màn hình Quản lý phòng ban

Giao diện Profile được minh họa bằng ảnh dưới đây

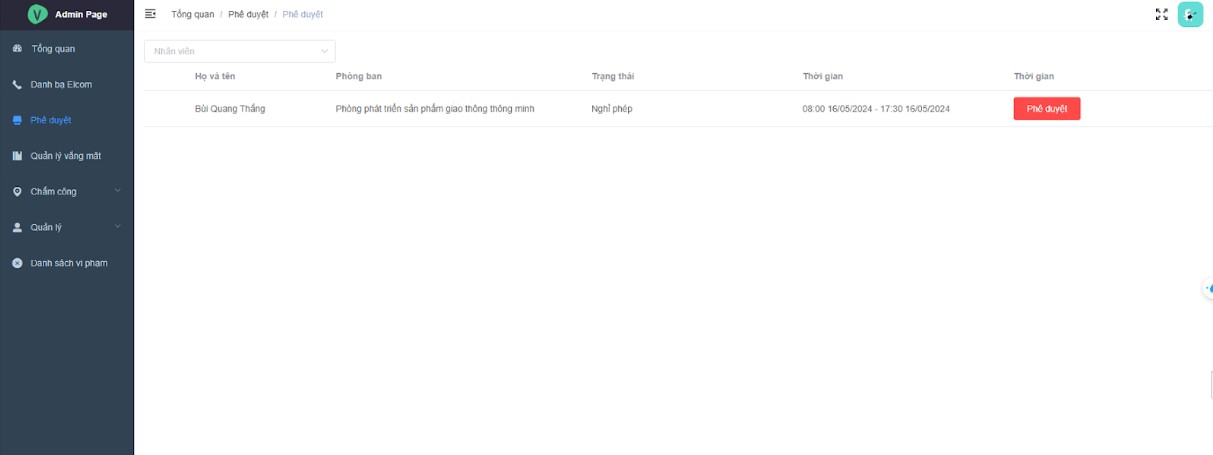
* Màn hình gồm thông tin cá nhân của nhân viên, nhân viên có quyền cập nhật thông tin hoặc đổi mật khẩu



Hình 4.7. Giao diện màn hình Profile

Giao diện Phê duyệt được minh họa bằng ảnh dưới đây

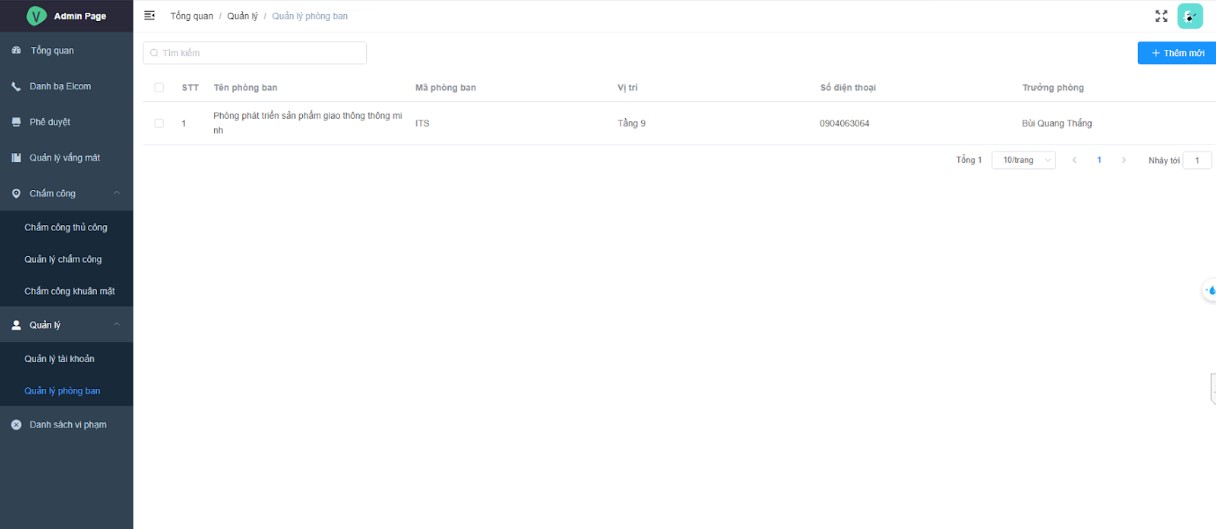
* Tại màn hình , trường phòng/admin sẽ có quyền phê duyệt đơn xin nghỉ phép của nhân viên cấp dưới



Hình 4.8. Giao diện màn hình Phê duyệt

Giao diện Thêm mới tài khoản được minh họa bằng ảnh dưới đây

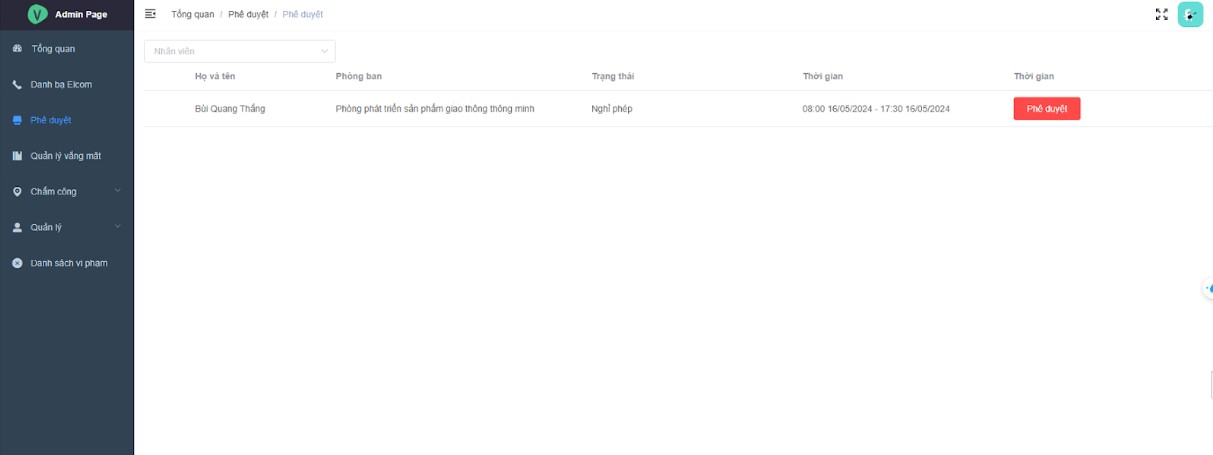
* Màn hình thêm mới tài khoản hiển thị khi người dùng click button “Thêm mới”, popup thêm mới hiển thị , sau khi thêm mới thành công bản ghi được update vào danh sách



Hình 4.9. Giao diện màn hình thêm mới tài khoản

Giao diện Thêm mới phòng ban được minh họa bằng ảnh dưới đây

* Màn hình thêm mới phòng ban hiển thị khi người dùng click button “Thêm mới”, popup thêm mới hiển thị , sau khi thêm mới thành công bản ghi được update vào danh sách



Hình 4.10. Giao diện màn hình Thêm mới phòng ban

## 4.2. Kiểm thử hệ thống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mục đích kiểm thử | Đầu vào | Đầu ra | Kết quả |
| 1 | Đăng nhập | Nhập dữ liệu hợp lệ | Chuyển hướng sang trang dashboard | Pass |
| 2 | Đăng xuất | Click nút “Đăng xuấtˮ | Chuyển hướng trang đăng nhập | Pass |
| 3 | Đăng ký vắng mặt | Nhập dữ liệu hợp lệ | Hệ thống lưu vào csdl và hiển thị lên màn hình đăng ký vắng mặt | Pass |
| 4 | Thêm mới tài khoản | Nhập dữ liệu hợp lệ | Hiển thị thông tin tài khoản mới tại màn hình danh sách tài khoản | Pass |
| 5 | Update tài khoản | Nhập dữ liệu hợp lệ. | Hiển thị thông tin tài khoản mới cập nhật tại  màn hình danh sách tài khoản | Pass |
| 6 | Xóa tài khoản | Click “Xóaˮ tại màn hình danh sách người dùng + click nút “Xác nhậnˮ để xác nhận xoá | Hệ thống hiển thị dialog xác nhận xoá Hệ thống hiển thị thông báo xoá thành công | Pass |
| 7 | Thêm mới phòng ban | Nhập dữ liệu hợp lệ | Hiển thị thông tin phòng ban mới tại màn | Pass |
|  |  |  | hình danh sách tài khoản |  |
| 8 | Update phòng ban | Nhập dữ liệu hợp lệ | Hiển thị thông tin phòng ban mới cập nhật tại màn hình danh sách phòng ban | Pass |
| 9 | Chấm công thủ công | Đăng nhập với role  Trường phòng Click “Chấm côngˮ tại màn hình chấm công thủ công | Hiển thị thời gian check in check out của nhân viên lên màn hình | Pass |
|  |  |  |  |  |

# KẾT LUẬN

Đề tài **“Xây dựng web chấm công tự động tích hợp AI cho doanh nghiệpˮ** xuất phát từ những nhu cầu thực tế mà ngày nay trong việc quản lý nhân sự cần có. Những phương pháp chấm công truyền thống dần lỗi thời và thay vào đó việc tích hợp việc quản lý chấm công vào website khắc phục được những tồn đọng khó khăn mà phương pháp thủ công đang gặp phải.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức cũng như các công nghệ mới.

**Những kết quả đạt được:**

**Về công nghệ:**

* Hiểu được quá trình thiết kế một webapp thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, viết chương trình, kiểm thử.
* Xây dựng webapp đáp ứng các nhu cầu cơ bản trong quản lý nhân viên
* phục vụ cho các doanh nghiệp
* Hiểu được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript.
* Hiểu được các kiến thức về framework VueJS, Spring boot trong lập trình web.
* Hiểu được các kiến thức nền tảng của các công cụ: Visual Studio Code, Postman, Git, Github.

**Về cài đặt chương trình:**

* Thực hiện các chức năng về xác thực và phân quyền cho nhân viên và quản trị hệ thống.
* Cho phép nhân viên có thể theo dõi lịch sử chấm công.
* Trang quản trị đầy đủ chức năng quản lý: quản lý tài khoản, dữ liệu chấm công

**Hướng phát triển:**

* Tích hợp với các hệ thống khác như quản lý nhân sự, tính lương.
* Phát triển ứng dụng di động cho phép chấm công từ xa.

# TÀI LIỆU THAM KHẢO

1. *Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng,* Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.*
2. *Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, Nhập môn công nghệ phần mềm, NXB Đại học Công nghiệp Hà Nội*
3. *Giáo trình thiết kế web*, Trường đại học Công Nghiệp Hà Nội.
4. Trang tin tức về lập trình [https://viblo.asia](https://viblo.asia/)
5. Kiểm thử hệ thống : https://docs.google.com/spreadsheets/d/1IfsMi0RfuLjPDpGimF3udwgrHaAc1gJH/edit?usp=sharing&ouid=103264911356642038694&rtpof=true&sd=true
6. Hướng dẫn sử dung [https://docs.google.com/document/d/1HioOlQPIO SEEXoZDgyl0kYvvvB\_9VCWGlDGag1Ghjc/edit?usp=sharing](https://docs.google.com/document/d/1HioOlQPIO-SEEXoZDgyl0kYvvvB_9VCWGlDGag1Ghjc/edit?usp=sharing)