

TRƯỜNG ĐẠI HỌC QUY NHƠN  
KHOA CÔNG NGHỆ THÔNG TIN

HUỲNH TẤN PHÁT

**TÌM HIỂU CÔNG NGHỆ NHẬN DẠNG  
KHUÔN MẶT ỨNG DỤNG VÀO CHẤM CÔNG  
TỰ ĐỘNG**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

QUY NHƠN - 2021

TRƯỜNG ĐẠI HỌC QUY NHƠN  
KHOA CÔNG NGHỆ THÔNG TIN

---

HUỲNH TẤN PHÁT

**TÌM HIỂU CÔNG NGHỆ NHẬN DẠNG  
KHUÔN MẶT ỨNG DỤNG VÀO CHẤM CÔNG  
TỰ ĐỘNG**

**Ngành:** Công nghệ thông tin

**Mã số sinh viên:** 4051050166

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

**NGƯỜI HƯỚNG DẪN:**

*ThS. Nguyễn Thị Loan*

**QUY NHƠN – 2021**

## LỜI CẢM ƠN

Trong suốt thời gian thực hiện đồ án tốt nghiệp không chỉ giúp tôi trang bị thêm nhiều kiến thức chuyên ngành phục vụ cho quá trình học tập, làm việc, mà còn là khoảng thời gian giúp tôi hình thành thêm nhiều kỹ năng hữu ích trong công việc và đời sống.

Trong quá trình học tập nói chung và thực hiện khóa luận nói riêng, tôi đã có cơ hội được học hỏi, nghiên cứu và trau dồi kiến thức thuộc lĩnh vực mình đam mê. Tôi xin gửi lời cảm ơn chân thành tới các thầy, cô khoa Công nghệ thông tin của Trường Đại học Quy Nhơn đã tận tình hướng dẫn, truyền đạt kiến thức cho tôi trong suốt thời gian học tập tại Trường, đến các cán bộ văn phòng Khoa đã tạo điều kiện tốt nhất cho tôi hoàn thành các thủ tục trong quá trình làm khóa luận.

Bằng sự biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành nhất đến Th.S Nguyễn Thị Loan, vì sự nhiệt tình của Cô trong hướng dẫn cũng như giúp đỡ tôi chỉnh sửa, hoàn thiện khóa luận. Sự tận tâm của Cô là động lực để tôi hoàn thành tốt khóa luận này.

Tôi xin chân thành cảm ơn./.

Sinh viên thực hiện

# MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC .....	ii
DANH MỤC CHỮ VIẾT TẮT.....	iv
DANH MỤC BẢNG .....	v
DANH MỤC HÌNH .....	vi
MỞ ĐẦU .....	vii
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	1
1.1. Tổng quan về nhận dạng khuôn mặt.....	1
1.1.1. Khái niệm hệ thống nhận dạng khuôn mặt .....	1
1.1.2. Một số thuật toán nhận dạng khuôn mặt .....	1
1.1.3. Lịch sử và ứng dụng của hệ thống nhận dạng khuôn mặt.....	2
1.2. Tổng quan hệ thống nhận dạng khuôn mặt.....	4
1.2.1. Nhận dạng hình ảnh .....	4
1.2.2. Các bước của quá trình nhận dạng .....	4
1.2.3. Một số kỹ thuật xử lý hình ảnh .....	5
1.2.4. Kiến trúc chung của một hệ thống nhận dạng khuôn mặt.....	8
1.3. Thư viện OpenCV .....	9
1.3.1. Giới thiệu.....	9
1.3.2. Ứng dụng thực tế.....	9
1.3.3. Các thao tác xử lý ảnh cơ bản .....	9
1.4. Thư viện Face Recognition .....	10
1.4.1. Giới thiệu.....	10
1.4.2. Các thao tác xử lý ảnh cơ bản .....	11
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG NHẬN DẠNG KHUÔN MẶT ỨNG DỤNG VÀO CHẤM CÔNG TỰ ĐỘNG .....	13
2.1. Mô tả bài toán .....	13
2.2. Mô hình hoạt động chung của hệ thống.....	13
2.3. Phân tích và thiết kế hệ thống chấm công .....	14
2.3.1. Biểu đồ Usecase .....	15
2.3.2. Biểu đồ tuần tự .....	24
2.3.3. Các thực thể.....	25
2.3.4. Kiến trúc phân tầng .....	25

2.3.5. Biểu đồ triển khai .....	25
2.4. Phân tích và thiết kế cơ sở dữ liệu .....	26
2.5. Thiết kế các màn hình chức năng .....	29
CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI ỨNG DỤNG .....	35
3.1. Các chức năng của hệ thống .....	35
3.2. Lựa chọn công nghệ cài đặt .....	35
3.2.1. Thư viện Ant Design of Angular .....	35
3.2.2. Thư viện PyQt5 .....	36
3.2.3. Spring Boot Framework .....	36
3.2.4. Thư viện OpenCV .....	37
3.2.5. Thư viện face-recognition .....	37
3.2.6. Xampp .....	37
3.3. Yêu cầu hệ thống .....	37
3.3.1. Phần cứng (Hardware) .....	37
3.3.2. Phần mềm (Software) .....	37
3.4. Xây dựng ứng dụng .....	37
3.5. Triển khai ứng dụng .....	38
KẾT LUẬN .....	39
TÀI LIỆU THAM KHẢO .....	41

## DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
LBPH	Local Binary Patterns Histograms
LBP	Local Binary Patterns
SIFT	Scale Invariant Feature Transform
SURF	Speed Up Robust Features
FRGC	Face Recognition Grand Challenge
CV	Computer Vision
PET	Positron Emission Tomography
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
CSDL	Cơ sở dữ liệu
API	Application Programming Interface
POI	Poor Obfuscation Implementation
DB	Database
UC	Use Case
SPA	Single Page Application
MIT	Massachusetts Institute of Technology
AOP	Aspect Oriented Programming
XML	eXtensible Markup Language
IDE	Integrated Development Environment
MD5	Message-Digest Algorithms
SHA	Secure Hash Algorithms
SQL	Structured Query Language
GPL	General Public License

## DANH MỤC BẢNG

Bảng 2.1: UC-Đăng nhập .....	16
Bảng 2.2: UC-Cập nhật thông tin nhân viên .....	16
Bảng 2.3: UC-Xóa nhân viên .....	17
Bảng 2.4: UC-Tìm kiếm thông tin nhân viên .....	17
Bảng 2.5: UC-Xem thông tin chi tiết của nhân viên .....	18
Bảng 2.6: UC-Kích hoạt tài khoản .....	19
Bảng 2.7: UC-Quản lý nhật ký .....	19
Bảng 2.8: UC-Xuất tập tin báo cáo .....	20
Bảng 2.9: UC-Huấn luyện mô hình .....	20
Bảng 2.10: UC-Xóa nhân viên khỏi hệ thống nhận dạng .....	21
Bảng 2.11: UC-Khởi chạy nhận dạng .....	21
Bảng 2.12: UC-Tạm dừng nhận dạng .....	22
Bảng 2.13: UC-Chấm công thủ công .....	22
Bảng 2.14: UC-Đăng ký nghỉ phép .....	23
Bảng 2.15: tbl_user .....	27
Bảng 2.16: tbl_department .....	27
Bảng 2.17: tbl_base_salary .....	27
Bảng 2.18: tbl_cal_salary .....	28
Bảng 2.19: tbl_holiday .....	28
Bảng 2.20: tbl_log .....	28
Bảng 2.21: tbl_time_in .....	28
Bảng 2.22: tbl_time_out .....	29
Bảng 2.23: tbl_status .....	29

## DANH MỤC HÌNH

Hình 1.1: Ví dụ giảm nhiễu .....	5
Hình 1.2: Một số ví dụ giảm nhiễu.....	6
Hình 1.3: Ví dụ điều chỉnh độ tương phản .....	6
Hình 1.4: Ví dụ tìm cạnh (Candy) .....	6
Hình 1.5: Ví dụ nén ảnh .....	7
Hình 1.6: Ví dụ phân vùng .....	7
Hình 1.7: Ví dụ khôi phục ảnh .....	7
Hình 1.8: Kiến trúc chung của một hệ thống nhận dạng khuôn mặt.....	8
Hình 1.9: Tìm tất cả khuôn mặt có trong ảnh.....	11
Hình 1.10: Tìm vị trí viền xung quanh khuôn mặt (face_landmarks).....	11
Hình 1.11: Nhận dạng những khuôn mặt có trong ảnh .....	12
Hình 2.1: Mô hình chung của hệ thống .....	14
Hình 2.2: Biểu đồ usecase .....	15
Hình 2.3: Biểu đồ tuần tự quá trình đăng nhập .....	24
Hình 2.4: Biểu đồ tuần tự quá trình nhận dạng chấm công .....	24
Hình 2.5: Biểu đồ triển khai hệ thống .....	26
Hình 2.6: Cơ sở dữ liệu .....	26
Hình 2.7: Màn hình đăng nhập (website) .....	29
Hình 2.8: Màn hình trang chủ (website).....	30
Hình 2.9: Màn hình quản lý người dùng (website) .....	30
Hình 2.10: Màn hình quản lý nhật kí (website).....	31
Hình 2.11: Màn hình đăng ký nghỉ phép (website).....	32
Hình 2.12: Màn hình chấm công thủ công (website) .....	32
Hình 2.13: Màn hình giới thiệu (Desktop) .....	33
Hình 2.14: Màn hình đăng nhập (Desktop).....	33
Hình 2.15: Màn hình trang chủ (Desktop).....	34
Hình 3.1: Cấu tạo Spring Boot Framework.....	37
Hình 3.2: Mẫu tập tin báo cáo .....	38



## MỞ ĐẦU

Nhận dạng không phải là công nghệ mới trong lịch vực xử lý ảnh. Nhưng việc ứng dụng công nghệ nhận dạng lại được sử dụng rộng rãi trong nhiều lĩnh vực của đời sống. Trong những năm gần đây, ứng dụng công nghệ nhận dạng vào việc chấm công đang được phát triển mạnh mẽ. Hiện nay, giải pháp chấm công nhận diện khuôn mặt được đánh giá là hiện đại và có tính bảo mật cao nhất. Công nghệ này có khả năng nhận diện các đặc điểm khuôn mặt một cách chính xác với độ bảo mật là tuyệt đối.

Chấm công nhận diện khuôn mặt là việc sử dụng một thiết bị có khả năng ghi nhận thời gian ra vào của nhân viên trong một công ty, doanh nghiệp, những nơi làm việc theo dõi thời gian đi làm của nhân viên bằng công nghệ nhận dạng khuôn mặt. Hiểu một cách đơn giản thì thiết bị này sẽ dựa vào các đặc điểm trên khuôn mặt của mỗi người để so sánh, nhận dạng và định danh người đó một cách tự động thông qua cơ sở dữ liệu khuôn mặt đã được lưu trữ trước đó.

Hiện nay có nhiều phương pháp và công cụ khác nhau hỗ trợ việc nhận dạng một người cụ thể trong thế giới thực với độ chính xác tương đối cao. Vì vậy việc lựa chọn công nghệ để nghiên cứu và xây dựng một chương trình chấm công tự động, sử dụng phương pháp nhận dạng có độ chính xác cao cùng thời gian tính toán ít là điều hết sức cần thiết. Đó chính là lý do tôi lựa chọn đề tài ***“Tìm hiểu công nghệ nhận dạng khuôn mặt ứng dụng vào chấm công tự động”*** cho khóa luận tốt nghiệp của mình.

Khóa luận được tổ chức thành 3 chương với nội dung như sau:

- Chương 1: Cơ sở lý thuyết. Trình bày tổng quan về nhận dạng khuôn mặt, các thư viện hỗ trợ cho việc nhận dạng.
- Chương 2: Phân tích và thiết kế hệ thống nhận dạng khuôn mặt trong chấm công tự động. Chương 2 gồm: phát biểu bài toán bài toán, phân tích thiết kế hệ thống, thiết kế cơ sở dữ liệu và các chức năng chính.
- Chương 3: Xây dựng và triển khai ứng dụng. Chương này trình bày về việc lựa chọn công nghệ, tổ chức cài đặt và triển khai thử nghiệm hệ thống.

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Nội dung chương này sẽ trình bày tổng quan về các phương pháp nhận dạng khuôn mặt, các thư viện hỗ trợ cho việc nhận dạng khuôn mặt làm cơ sở lý thuyết cho các phân tích ở chương 2.

### 1.1. Tổng quan về nhận dạng khuôn mặt

#### 1.1.1. Khái niệm hệ thống nhận dạng khuôn mặt

Hệ thống nhận dạng khuôn mặt là một ứng dụng máy tính tự động xác định hoặc nhận dạng một người nào đó từ một bức ảnh kỹ thuật số hoặc một khung hình video từ một nguồn video. Một trong những cách để thực hiện điều này là so sánh các đặc điểm khuôn mặt chọn trước từ hình ảnh và một cơ sở dữ liệu về khuôn mặt (nguồn wikipedia).

#### 1.1.2. Một số thuật toán nhận dạng khuôn mặt

##### a) *Eigenface* (1991)

*Eigenface* là tên được đặt cho một tập hợp các *eigenvector* khi được sử dụng trong bài toán thị giác máy tính (*computer vision*) để nhận dạng khuôn mặt người. Phương pháp này sử dụng *eigenfaces* để nhận dạng được phát triển bởi Sirovich cùng với Kirby (1987), được Matthew Turk và Alex Pentland sử dụng trong phân loại khuôn mặt. Các *eigenvector* được suy ra từ ma trận hiệp phương sai của phân bố xác suất trên không gian vector chiều cao của ảnh khuôn mặt. Bản thân các *eigenface* tạo thành một tập hợp cơ sở của tất cả các hình ảnh được sử dụng để xây dựng ma trận hiệp phương sai. Điều này tạo ra sự giảm kích thước bằng cách cho phép tập hợp các hình ảnh cơ sở nhỏ hơn đại diện cho các hình ảnh đào tạo ban đầu. Việc phân loại có thể đạt được bằng cách so sánh các mặt được biểu diễn bởi bộ cơ sở. [21]

##### b) *Local Binary Patterns Histograms (LBPH)* (1996)

*Local Binary Patterns* (hay còn viết tắt là LBP) được Ojala trình bày vào năm 1996 như là một cách đo độ tương phản cục bộ của ảnh. Phiên bản đầu tiên của LBP được dùng với 8 điểm ảnh xung quanh và sử dụng giá trị của điểm ảnh ở trung tâm làm ngưỡng. Giá trị LBP được xác định bằng cách nhân các giá trị ngưỡng với trọng số ứng với mỗi điểm ảnh sau đó cộng tổng lại.

LBP là một phương pháp rút trích đặc trưng trong xử lý ảnh. Đặc trưng được rút trích sẽ tiếp tục được tiến hành chọn lọc (feature selection) thu gọn thành vector đặc trưng. Vector đặc trưng này sau đó có thể dùng để đưa vào mô hình học máy để học hoặc phân loại. [20]

##### c) *Fisherfaces* (1997)

*FisherFaces* là một thuật toán cải tiến của *EigenFaces*. *EigenFaces* xem độ sáng của hình ảnh là một đặc điểm nhận dạng quan trọng để nhận dạng khuôn mặt nhưng thực tế thì độ sáng không phải là một đặc điểm quan trọng để nhận dạng khuôn mặt. Việc xem độ sáng là một đặc điểm nhận dạng quan trọng có thể bỏ qua các đặc điểm nhận khác vì chúng kém hữu ích hơn. Để khắc phục điều này, FisherFaces điều chỉnh các eigenfaces để trích xuất các đặc điểm nhận dạng chính của người này để so sánh với người khác thay vì xem xét tổng thể. Vì vậy, ngay cả khi dữ liệu khuôn mặt của một người có thay đổi lớn về độ chiếu sáng cũng không ảnh hưởng đến các đặc điểm nhận dạng. [19]

*d) Scale Invariant Feature Transform (SIFT) (1999)*

*Scale Invariant Feature Transform* (SIFT) là một thuật toán nhận diện (detection algorithm) trong thị giác máy tính để phát hiện và mô tả các đặc trưng cục bộ trong hình ảnh. Nó được xuất bản bởi David Lowe vào năm 1999. Các ứng dụng bao gồm nhận dạng đối tượng, lập bản đồ và điều hướng bằng robot, ghép hình ảnh, mô hình 3D, nhận dạng cử chỉ, theo dõi video, nhận dạng cá thể động vật hoang dã và di chuyển. [17]

*e) Speed Up Robust Features (SURF) (2006)*

Trong thị giác máy tính, *Speed Up Robust Features* (SURF) là bộ mô tả và phát hiện các đặc điểm nhận dạng cục bộ đã được cấp bằng sáng chế. Nó có thể được sử dụng cho các nhiệm vụ như nhận dạng đối tượng, xử lý hình ảnh, phân loại hoặc tái tạo mô hình 3D. Nó một phần được lấy cảm hứng từ bộ mô tả biến đổi đặc trưng bất biến theo tỷ lệ (*SIFT*). Phiên bản tiêu chuẩn của *SURF* nhanh hơn nhiều lần so với *SIFT* và được các nhà sáng lập ra nó tuyên bố là mạnh mẽ hơn trong việc xử lý các biến đổi hình ảnh khác nhau so với *SIFT*. [18]

### **1.1.3. Lịch sử và ứng dụng của hệ thống nhận dạng khuôn mặt**

Những người tiên phong trong tự động nhận dạng khuôn mặt bao gồm Woody Bledsoe, Helen Chan Wolf, và Charles Bisson. Trong năm 1964 và 1965, Bledsoe, cùng với Helen Chan và Charles Bisson, bắt đầu ý tưởng sử dụng máy tính để nhận ra khuôn mặt của con người. Ông rất tự hào về công việc này, nhưng do kinh phí được cung cấp bởi một cơ quan tình báo giấu tên nên không cho phép công khai, rất ít tác phẩm đã được xuất bản. Với một cơ sở dữ liệu lớn các hình ảnh (thực tế là một cuốn sách ảnh thể) và một bức ảnh, vấn đề là phải lựa chọn từ cơ sở dữ liệu là một tập hợp nhỏ các hồ sơ hình ảnh như vậy có chứa các hình ảnh khớp với bức ảnh đưa ra. Sự thành công của phương pháp này có thể được đo bằng tỷ lệ câu trả lời đúng trên số lượng các hồ sơ trong cơ sở dữ liệu.

Dự án này đã được dán nhãn là "*man-machine*" bởi vì con người trích xuất tọa độ của một tập hợp các đặc điểm từ các hình ảnh, sau đó được máy tính sử dụng để nhận dạng. Sử dụng một graphic tablet (*GRAFACON* hoặc *RAND TABLET*) để lưu các tọa độ của các đặc điểm như tâm của con người, các góc bên trong mắt, góc ngoài của mắt,... Từ những tọa độ này, một danh sách 20 khoảng cách, như chiều rộng của miệng và khoảng cách giữa 2 mắt, từ con người trái đến con người phải sẽ được tính toán. Các toán tử có thể xử lý khoảng 40 hình ảnh trên một giờ. Khi xây dựng các cơ sở dữ liệu, tên của người trong bức ảnh đã được gắn với danh sách của các khoảng cách tính toán và được lưu trữ trong máy tính. Trong giai đoạn nhận dạng, tập hợp các khoảng cách được so sánh với khoảng cách tương ứng cho mỗi bức ảnh, cho ra một khoảng cách giữa các bức ảnh và các bản ghi cơ sở dữ liệu. Các hồ sơ gần nhất được trả về.

Bởi vì không chắc rằng bất kỳ hai hình ảnh sẽ khớp nhau khi xoay đầu, nghiêng hoặc cúi đầu, và tỉ lệ (khoảng cách tới máy ảnh), mỗi bộ khoảng cách được chuẩn hóa để đại diện cho khuôn mặt theo hướng nhìn từ phía trước. Để thực việc chuẩn hóa này, chương trình đầu tiên sẽ xác định độ nghiêng, xoay đầu, và cúi đầu. Sau đó, sử dụng các góc độ này, máy tính sẽ xóa những biến dạng này trên các khoảng cách tính toán.

Sau khi Bledsoe rời PRI vào năm 1966, công việc này được tiếp tục tại Viện nghiên cứu Stanford, chủ yếu bởi Peter Hart. Trong các thí nghiệm thực hiện trên một cơ sở dữ liệu hơn 2.000 bức ảnh, máy tính luôn vượt trội so với con người khi thực hiện cùng một nhiệm vụ nhận dạng (Bledsoe 1968).

Đến khoảng năm 1997, hệ thống được phát triển bởi Christoph von der Malsburg và các sinh viên sau đại học của trường Đại học Bochum ở Đức và Đại học Nam California tại Mỹ đã cho kết quả vượt trội so với hầu hết các hệ thống của Viện Công nghệ Massachusetts và Đại học Maryland. Hệ thống Bochum được phát triển thông qua tài trợ bởi Phòng Thí Nghiệm Nghiên cứu Quân đội Hoa Kỳ. Phần mềm này được bán với cái tên ZN-Face và sử dụng bởi các khách hàng như Deutsche Bank và các nhà điều hành sân bay. Phần mềm này “đủ mạnh mẽ để nhận dạng được gương mặt từ các góc nhìn ít lý tưởng hơn. Nó cũng có thể nhận dạng được gương mặt với những trở ngại như ria mép, râu, thay đổi kiểu tóc và thậm chí đeo kính râm”.

Trong khoảng tháng 1 năm 2007, ý tưởng tìm kiếm hình ảnh có thể “dựa trên ký tự xung quanh của bức ảnh đó” hay việc “yêu cầu người dùng nhập tên của những người mà họ nhận ra trong ảnh online”, điều này giúp cho việc xây dựng một cơ sở dữ liệu ngày càng nhiều hơn. Identix, một công ty từ Minnesota ở Mỹ, đã phát triển một phần mềm là FaceIt. FaceIt có thể nhận ra khuôn mặt của một ai đó trong đám đông và so sánh nó với cơ sở dữ liệu trên toàn thế giới để nhận dạng và đặt tên cho một khuôn mặt.

Phần mềm được viết để phát hiện nhiều đặc điểm trên khuôn mặt người. Nó có thể phát hiện khoảng cách giữa hai mắt, chiều rộng của mũi, hình dạng của xương gò má, độ dài của đường viền của hàm dưới và nhiều đặc điểm khác trên khuôn mặt. Nó thực hiện điều này bằng cách đưa hình ảnh của khuôn mặt vào một faceprint, một mã số đại diện cho gương mặt của con người. Phần mềm nhận dạng khuôn mặt trước kia thường phải dựa trên hình ảnh 2D của người cần nhận dạng gần như phải trực diện với máy ảnh. Giờ đây, với FaceIt, một hình ảnh 3D có thể được so sánh với một hình ảnh 2D bằng cách chọn 3 điểm cụ thể trên tấm hình 3D và chuyển đổi nó thành hình ảnh 2D sử dụng một thuật toán đặc biệt có thể được quét qua hầu như tất cả các cơ sở dữ liệu.

Năm 2006, các thuật toán nhận dạng khuôn mặt mới nhất đã được đánh giá trong Face Recognition Grand Challenge (FRGC). Hình ảnh gương mặt, hình ảnh scan gương mặt 3D và ảnh iris độ phân giải cao, được sử dụng trong các bài kiểm tra. Kết quả cho thấy rằng các thuật toán mới chính xác hơn 10 lần so với các thuật toán nhận dạng khuôn mặt của năm 2002 và chính xác hơn 100 lần so với các thuật toán của năm 1995. Một số thuật toán đã có thể nhận dạng tốt hơn dựa trên việc nhận diện khuôn mặt và đặc biệt có thể xác định từng người trong các cặp song sinh giống hệt nhau. [16]

## **1.2. Tổng quan hệ thống nhận dạng khuôn mặt**

### **1.2.1. Nhận dạng hình ảnh**

Trong Công nghệ thông tin, hai thuật ngữ “tầm nhìn máy tính” và “nhận dạng hình ảnh” có thể được sử dụng thay thế cho nhau. Thị giác máy tính (Computer Vision - CV) là để máy tính bắt chước tầm nhìn của con người và *thực hiện hành động*. Ví dụ, CV có thể được thiết kế để cảm nhận một đứa trẻ đang chạy trên đường và tạo ra tín hiệu cảnh báo cho người lái xe. Ngược lại, nhận dạng hình ảnh là phân tích pixel và mẫu của hình ảnh để nhận dạng hình ảnh như một đối tượng cụ thể. Thị giác máy tính có nghĩa là nó có thể “làm điều gì đó” với các hình ảnh được nhận dạng.

Ngày nay, công nghệ nhận dạng và xử lý ảnh đang là một lĩnh vực mà rất nhiều người quan tâm, nghiên cứu. Nhờ vào sự phát triển mạnh mẽ của *Machine Learning* - một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể. Xử lý ảnh đã và đang được ứng dụng vào nhiều lĩnh vực trong cuộc sống: y tế (X Ray Imaging, PET scan,...), thị giác máy tính (giúp máy tính có thể hiểu, nhận biết đồ vật như con người), các công nghệ nhận dạng (vân tay, khuôn mặt,...).

### **1.2.2. Các bước của quá trình nhận dạng**

#### **Bước 1: Chuẩn bị tập dữ liệu và rút trích đặc trưng**

Đây là công đoạn được xem là quan trọng nhất trong các bài toán về Machine Learning. Vì dữ liệu này được dùng cho quá trình học để tìm ra mô hình của bài toán. Nếu chúng ta xây dựng được một mô hình tốt thì kết quả sẽ cho độ chính xác cao hơn, vì vậy cần phải chọn ra được những đặc trưng tốt của dữ liệu, và loại bỏ những đặc trưng không tốt, gây nhiễu ở dữ liệu.

## **Bước 2: Xây dựng mô hình**

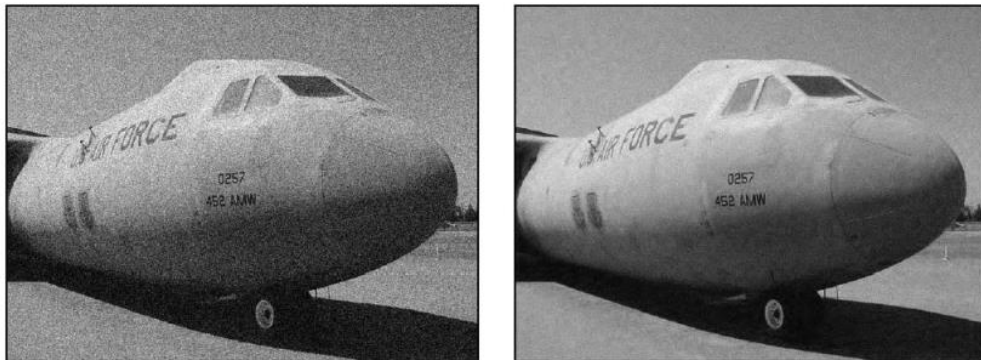
Mục đích của mô hình huấn luyện là tìm ra hàm  $f(x)$ , thông qua nó để dán nhãn cho dữ liệu. Bước này thường được gọi là học hay còn gọi là training. Thông thường để xây dựng mô hình phân lớp cho bài toán này, chúng ta sử dụng các thuật toán học có giám sát như: KNN, Neural Network, SVM, Decision Tree, Navie Bayers, Random Forest...

## **Bước 3: Đánh giá mô hình**

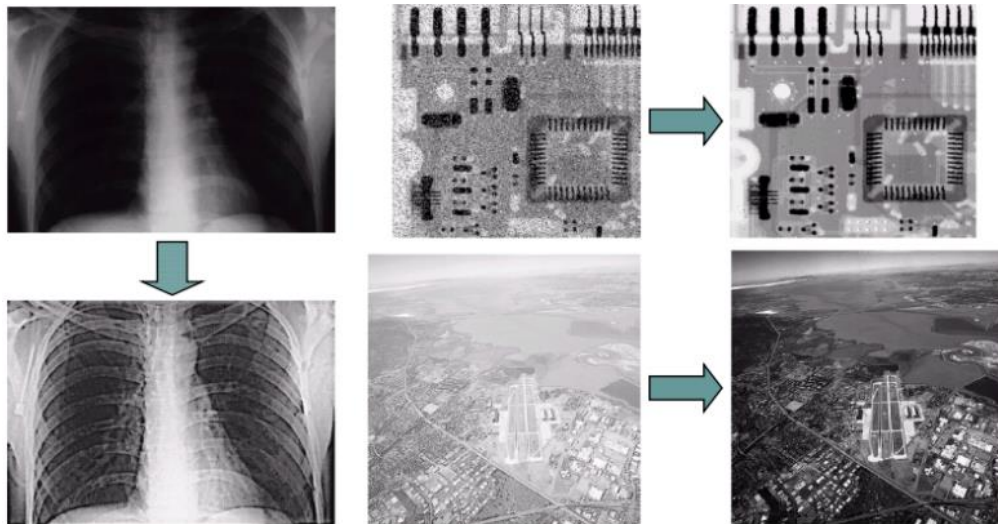
Ở bước này, chúng ta sẽ đánh giá mô hình bằng cách đánh giá độ chính xác của dữ liệu test thông qua mô hình vừa xây dựng. Nếu không đạt được kết quả như mong muốn, thì phải thay đổi các tham số của các thuật toán học để tìm ra mô hình tốt hơn và kiểm tra, đánh giá lại mô hình. Cuối cùng chọn ra mô hình tốt nhất cho bài toán đã đưa ra.

### **1.2.3. Một số kỹ thuật xử lý hình ảnh**

#### *a) Giảm nhiễu*

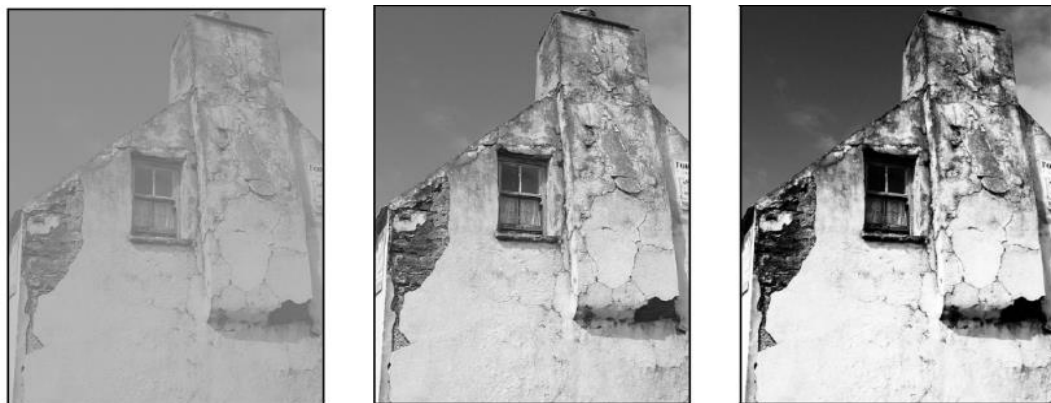


**Hình 1.1: Ví dụ giảm nhiễu**



Hình 1.2: Một số ví dụ giảm nhiễu

*b) Điều chỉnh độ tương phản*



Low Contrast

Original Contrast

High Contrast

Hình 1.3: Ví dụ điều chỉnh độ tương phản

*c) Tìm cạnh*



Hình 1.4: Ví dụ tìm cạnh (Candy)



*d) Nén ảnh*



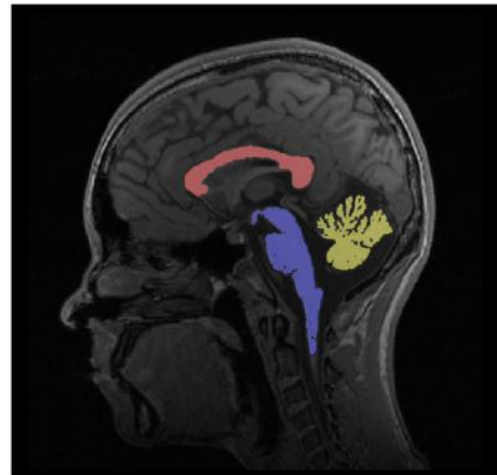
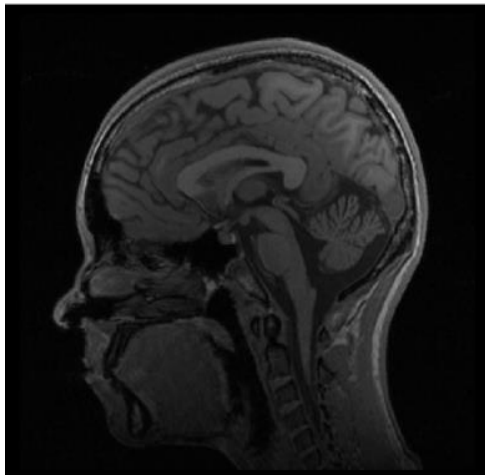
Original, 2.1MB



JPEG Compression, 308KB (15%)

**Hình 1.5: Ví dụ nén ảnh**

*e) Phân vùng*



**Hình 1.6: Ví dụ phân vùng**

*f) Khôi phục ảnh*



Credit: M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester: Image Inpainting, SIGGRAPH 2000

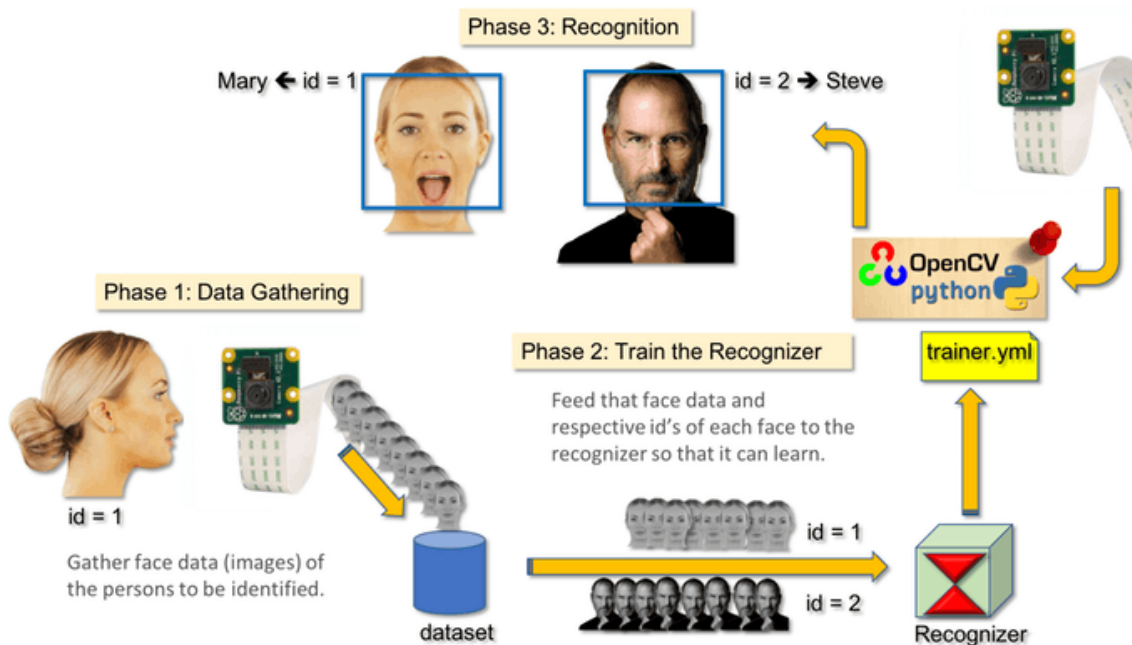
**Hình 1.7: Ví dụ khôi phục ảnh**



#### 1.2.4. Kiến trúc chung của một hệ thống nhận dạng khuôn mặt

Một hệ thống nhận dạng khuôn mặt thường phải trải qua 3 giai đoạn:

- Nhận diện khuôn mặt và thu thập dữ liệu (Face Detection and Data Gathering).
- Huấn luyện mô hình (Train the Recognizer).
- Nhận dạng khuôn mặt (Face Recognition).



Hình 1.8: Kiến trúc chung của một hệ thống nhận dạng khuôn mặt

##### Giai đoạn 1: Nhận diện khuôn mặt và thu thập dữ liệu

Công việc cơ bản mà mọi hệ thống nhận dạng khuôn mặt đều phải thực hiện đó là nhận diện khuôn mặt trong ảnh “Face Detecting”. Trước khi thực hiện bất cứ công việc tiếp theo nào thì ta phải phát hiện được những khuôn mặt có trong ảnh (Giai đoạn 1). Để có thể nhận dạng, ta so sánh những khuôn mặt cũ này với những khuôn mặt mới trong tương lai (Giai đoạn 3).

##### Giai đoạn 2: Huấn luyện mô hình

Trong giai đoạn thứ 2 này, chúng ta sẽ lấy tất cả dữ liệu của người dùng thu thập được sau kết thúc giai đoạn 1. Nguồn dữ liệu này (dataset) sẽ là dữ liệu đầu vào để huấn luyện mô hình học máy sao cho nó có thể nhận dạng được những khuôn mặt mới trong tương lai.

##### Giai đoạn 3: Nhận dạng khuôn mặt

Ở giai đoạn này, với những bức ảnh chụp được từ những khuôn mặt mới, nếu người này đã được thu thập và đào tạo cho mô hình trước đó, thì ở giai đoạn này chương

trình nhận dạng sẽ sử dụng mô hình đã huấn luyện để xử lý và đưa ra “dự đoán” về kết quả nhận dạng, có thể là trả về id hoặc chỉ số (index) của người đó trong tập dữ liệu, cùng với độ tin cậy.

### 1.3. Thư viện OpenCV

#### 1.3.1. Giới thiệu

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho những xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng ngôn ngữ C/C++, vì vậy tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. OpenCV có các giao diện (interface) cho C/C++, Python, Java vì vậy hỗ trợ được cho các hệ điều hành Windows, Linux, MacOS cả Android, iOS. Thư viện OpenCV có cộng đồng người dùng lớn và số lượng download hơn quá 6 triệu lượt.

#### 1.3.2. Ứng dụng thực tế

OpenCV được sử dụng cho nhiều loại ứng dụng khác nhau bao gồm :

- Hình ảnh street view
- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y học
- Tìm kiếm và phục hồi hình ảnh/video
- Phim – cấu trúc 3D tự chuyển động
- Nghệ thuật sắp đặt tương tác

#### 1.3.3. Các thao tác xử lý ảnh cơ bản

##### a) *Hiển thị hình ảnh*

```
img = cv2.imread('digital-neon.jpg')
cv2.imshow('Display Image', img)
cv2.waitKey(0)
```

##### b) *Lấy kích thước của ảnh*

```
img = cv2.imread('digital-neon.jpg')
(h, w, d) = img.shape
print("width={}, height={}, depth={}".format(w, h, d))
Output: width=580, height=326, depth=3
```

Ở đây biến `img` là một đối tượng của Numpy array chứa giá trị màu của từng điểm ảnh trên các không gian màu khác nhau. Lệnh `img.shape` để lấy ra kích thước của mảng này với `h`, `w`, `d` lần lượt là chiều cao, chiều rộng, độ sâu của bức ảnh. Với ảnh có màu thì độ sâu thường là 3, ảnh đen trắng là 1.

c) Lấy giá trị màu ở một điểm ảnh

```
(B, G, R) = img[50, 50]
print("R={}, G={}, B={}".format(R, G, B))
output: R=96, G=100, B=111
```

Giá trị các thành phần màu có giá trị từ 0 đến 255.

d) Cắt ảnh

```
roi = img[50:350, 60:360]
cv2.imshow('Region Of Interest', roi)
cv2.waitKey(0)
```

Đoạn lệnh phía trên cắt ảnh có tọa độ điểm trên cùng bên trái là (50, 60) và tọa độ điểm dưới cùng bên phải là (350, 360).

e) Thay đổi kích thước ảnh

```
img = cv2.imread('digital-neon.jpg')
(h, w, d) = img.shape
r = 300.0 / w
dim = (300, int(h * r))
resized = cv2.resize(img, dim)
```

Để thay đổi kích thước của ảnh trong OpenCV chúng ta dùng hàm resize. Hàm này cần xác định chiều cao, chiều rộng ảnh sau khi biến đổi. Tuy nhiên, ảnh sau khi thay đổi thường cùng tỷ lệ chiều cao, chiều rộng với ảnh gốc.

Đoạn lệnh ở trên biến đổi ảnh gốc có chiều rộng 580 pixel sang ảnh có chiều rộng là 300 pixel.

f) Xoay ảnh

```
img = cv2.imread('digital-neon.jpg')
(h, w, d) = img.shape
center = (w // 2, h // 2)
M = cv2.getRotationMatrix2D(center, 45, 1.0)
rotated = cv2.warpAffine(img, M, (w, h))
```

Để xoay được ảnh đầu tiên ta cần xác định góc xoay và hướng xoay. Sau đó, tính ma trận xoay bằng hàm getRotationMatrix2D trong OpenCV. Cuối cùng ta nhân ma trận này với ma trận ảnh gốc ta được ảnh sau khi xoay.

Đoạn lệnh ở phía trên xoay ảnh một góc 45 độ theo chiều ngược kim đồng hồ với góc xoay là điểm chính giữa hình ảnh.

## 1.4. Thư viện Face Recognition

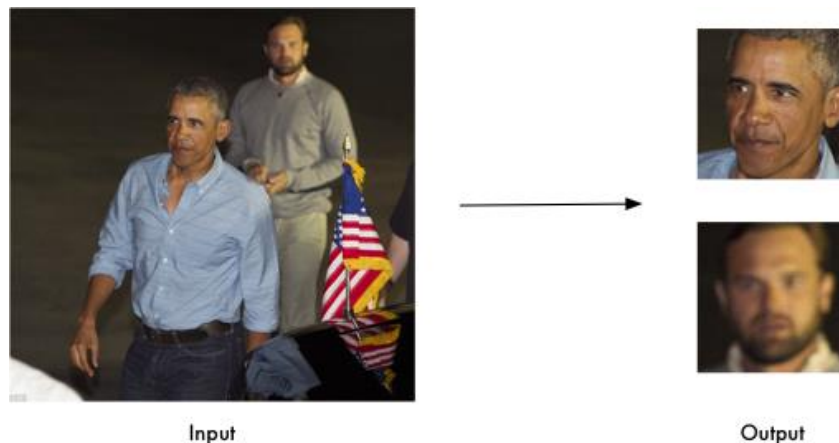
### 1.4.1. Giới thiệu

Face Recognition là một thư viện của Python, được công bố lần đầu vào ngày 13 tháng 3 năm 2017. Thư viện này là một công cụ cực kỳ hữu dụng cho các ứng dụng nhỏ

và rất phù hợp với mục đích học tập và nghiên cứu. Độ chính xác của Face Recognition có thể lên đến 99,38%. Tuy nhiên khả năng nhận dạng của Face Recognition lại bị hạn chế bởi các điều kiện bên ngoài và phụ thuộc vào độ phân giải của các bức ảnh, mô hình được đào tạo tối ưu cho ảnh người lớn nên sẽ gặp một số vấn đề về độ chính xác khi nhận dạng trên ảnh trẻ em, có thể khắc phục tạm thời bằng cách thiết lập tham số “distance 0.6”. Tuy nhiên, Face Recognition vẫn là một thư viện mạnh và hữu dụng cho các ứng dụng với Raspberry Pi và Face Recognition Process.

#### 1.4.2. Các thao tác xử lý ảnh cơ bản

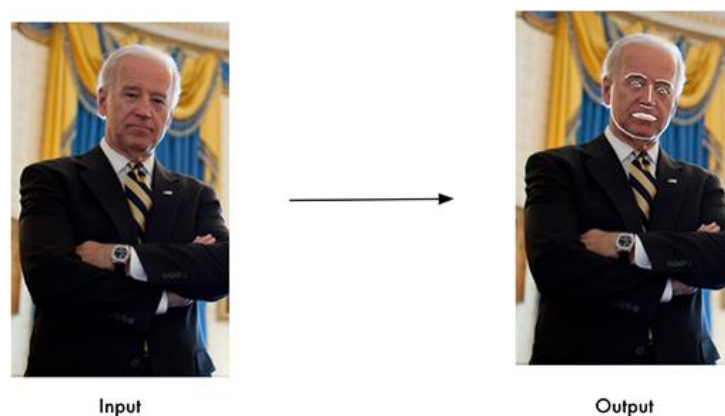
##### a) Tìm tất cả khuôn mặt trong ảnh



Hình 1.9: Tìm tất cả khuôn mặt có trong ảnh

```
import face_recognition
image = face_recognition.load_image_file("yourFile.jpg")
face_locations = face_recognition.face_locations(image)
```

##### b) Tìm các điểm nổi bật của khuôn mặt trong ảnh

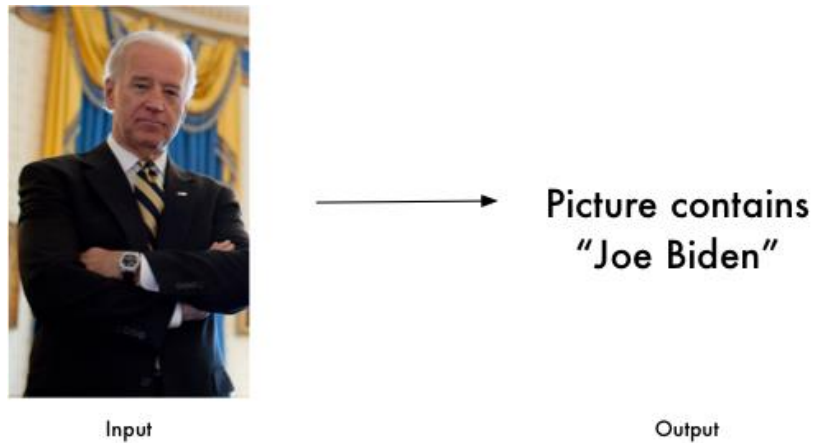


Hình 1.10: Tìm vị trí viền xung quanh khuôn mặt (face\_landmarks)

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```

##### c) Nhận dạng khuôn mặt

Nhận dạng những người trong ảnh.



Hình 1.11: Nhận dạng những khuôn mặt có trong ảnh

```
import face_recognition
known_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")
biden_encoding = face_recognition
                    .face_encodings(known_image)[0]
unknown_encoding = face_recognition
                    .face_encodings(unknown_image)[0]
results = face_recognition
           .compare_faces([biden_encoding], unknown_encoding)
```

## **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG NHẬN DẠNG KHUÔN MẶT ỨNG DỤNG VÀO CHẤM CÔNG TỰ ĐỘNG**

Chương 2 sẽ mô tả tất cả các đặc trưng, các bước phân tích thiết kế các tính năng của một hệ thống chấm công tự động. Để thấy được một “bức tranh toàn cảnh” về sự vận hành của một hệ thống chấm công tự động sử dụng công nghệ nhận dạng khuôn mặt trong môi trường thực tế.

### **2.1. Mô tả bài toán**

Ngày nay, khi công nghệ dần trở nên quen thuộc với cuộc sống hàng ngày của tất cả mọi người, không loại trừ các hoạt động vận hành thường nhật của doanh nghiệp như: chấm công, điểm danh... thì việc sử dụng phần mềm chấm công bằng nhận dạng khuôn mặt thay thế các thiết bị vật lý như máy vân tay, máy quét thẻ... đang dần trở thành xu hướng mới.

#### **Yêu cầu chung của bài toán chấm công tự động**

- Thông tin đầu vào:
  - + Thông tin nhân viên của một công ty
  - + Dữ liệu hình ảnh khuôn mặt của từng nhân viên
  - + Thông tin nghỉ phép, nghỉ lễ theo quy định
- Thông tin đầu ra:
  - + Bảng chấm công hàng tháng của nhân viên

#### **Một số yêu cầu riêng**

- Cho phép điều chỉnh thủ công trong các trường hợp: hệ thống mất điện, nhân viên nghỉ phép không có kế hoạch trước,...
- Xử lý điều chỉnh số công khi nhân viên đi trễ về sớm.

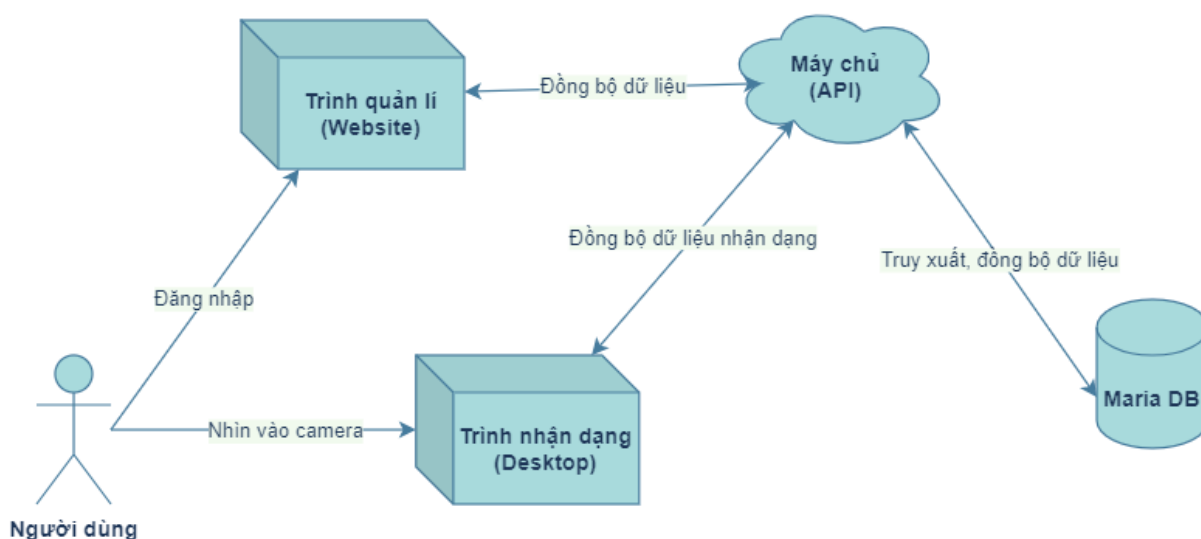
### **2.2. Mô hình hoạt động chung của hệ thống**

Để giải quyết những vấn đề nêu trên, tôi xây dựng hệ thống gồm ba thành phần chính: (1) Ứng dụng web quản lý chấm công, gồm thông tin nhân viên, nhật ký ra vào, nhật ký chấm công, cho phép chấm công thủ công dành cho những trường hợp đặc biệt, gặp sự cố,... (2) Ứng dụng chạy trên máy tính dùng để quản lý quá trình nhận dạng, huấn luyện mô hình và khởi chạy trình nhận dạng khuôn mặt. (3) Một API trung tâm,

dùng cho việc xử lý logic nhận dạng, chấm công, xuất báo cáo và thao tác trực tiếp với cơ sở dữ liệu.

Khi xây dựng và triển khai hệ thống, tôi sử dụng Spring Boot Framework để xây dựng một Web API giao tiếp với CSDL và một số nền tảng khác, dùng POI (Poor Obfuscation Implementation) để thao tác với tập tin Excel. Sử dụng Angular Framework kết hợp với Ant Design Of Angular để thiết kế và vẽ một giao diện web thân thiện với người dùng. Còn về phần ứng dụng trên máy tính, tôi sử dụng PyQt5, một thư viện hỗ trợ thiết kế giao diện không chuyên do Python cung cấp để xây dựng giao diện, dùng thư viện Face-Recognition để xử lý quá trình huấn luyện mô hình cũng như nhận dạng với độ chính xác lên tới 99,38%. [10]

Trong mô hình dưới, nhân viên công ty và khách mời là những người trực tiếp được camera ghi hình lại để xử lý và nhận dạng, sau đó những thông tin này được gửi lên máy chủ thông qua API để đối chiếu, xử lý và từ máy chủ sẽ lưu trữ trực tiếp vào CSDL gồm nhật ký vào ra và dữ liệu chấm công. Để theo dõi, quản lý những thông tin trên, người quản trị có thể đăng nhập trực tiếp vào hệ thống website quản trị với tài khoản hợp lệ, lúc này hệ thống sẽ gửi một yêu cầu xác thực lên máy chủ, sau khi xử lý xong máy chủ sẽ trả về kết quả, nếu kết quả là hợp lệ thì sẽ chuyển hướng đến trang quản trị. Tại đây, dữ liệu hiển thị trên giao diện web cũng không phải lấy trực tiếp từ CSDL, mà phải gửi yêu cầu truy xuất lên máy chủ, máy chủ sẽ xác thực, truy xuất dữ liệu và gửi trả kết quả về nếu yêu cầu hợp lệ.

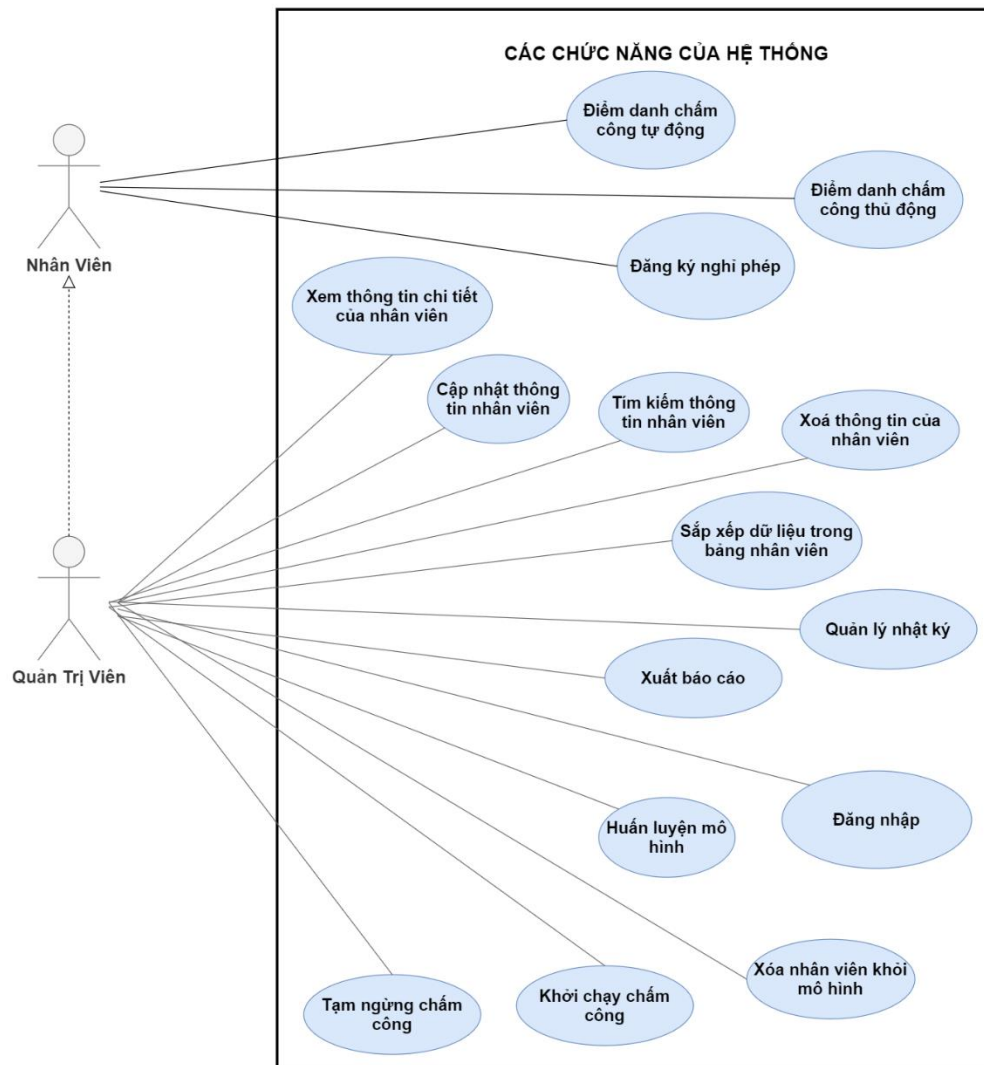


Hình 2.12: Mô hình chung của hệ thống

## 2.3. Phân tích và thiết kế hệ thống chấm công



### 2.3.1. Biểu đồ Usecase



Hình 2.2: Biểu đồ usecase

Qua biểu đồ ca sử dụng (usecase diagram) này ta có thể thấy, với vai trò là một người quản trị viên thì có thể thao tác, thực hiện một số chức năng như:

- Đăng nhập
- Cập nhật thông tin nhân viên
- Tìm kiếm thông tin nhân viên
- Quản trị viên nhật kí
- Xuất báo cáo
- Huấn luyện mô hình



- Khởi chạy và tạm ngừng chấm công

Còn với vai trò là một nhân viên thì có thể:

- Điểm danh chấm công tự động
- Đăng ký chấm công thủ công
- Đăng ký nghỉ phép

**Bảng 2.1: UC-Đăng nhập**

<b>Mã Use case</b>	UC-01
<b>Tên Use case</b>	Đăng nhập
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên đăng nhập vào hệ thống
<b>Tiền điều kiện (Preconditions)</b>	
<b>Kết quả (Postconditions)</b>	Hệ thống chuyển sang màn hình quản trị viên trên website hoặc desktop
<b>Điều kiện kích hoạt use case (Triggers – specific business event)</b>	Quản trị viên truy cập vào website quản trị viên hoặc ứng dụng desktop
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	1. Người dùng nhập tên đăng nhập và mật khẩu 2. Người dùng nhấn Đăng nhập 3. Hệ thống thông báo “Đăng nhập thành công” và hiển thị màn hình quản trị viên
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.2: UC-Cập nhật thông tin nhân viên**

<b>Mã Use case</b>	UC-02
<b>Tên Use case</b>	Cập nhật thông tin nhân viên
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên cập nhật thông tin nhân viên
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Thông tin nhân viên được cập nhật

<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn cập nhật thông tin của nhân viên
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	<ol style="list-style-type: none"> <li>1. Quản trị viên nhấn vào trang quản trị viên nhân viên trong thanh menu ở bên trái</li> <li>2. Nhấn vào icon “search” ở cuối mỗi bản ghi thông tin nhân viên</li> <li>3. Chỉnh sửa thông tin nhân viên</li> <li>4. Nhấn nút “Update”</li> <li>5. Hệ thống thông báo cập nhật thông tin thành công</li> </ol>
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.3: UC-Xóa nhân viên**

<b>Mã Use case</b>	UC-03
<b>Tên Use case</b>	Xóa nhân viên
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn</b> ( <i>Brief</i> )	Quản trị viên xóa nhân viên khỏi hệ thống
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Thông tin nhân viên được xóa khỏi hệ thống
<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn xóa nhân viên
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	<ol style="list-style-type: none"> <li>1. Quản trị viên chọn trang quản trị viên nhân viên trong thanh menu ở bên trái</li> <li>2. Nhấn vào icon “delete” ở cuối mỗi bản ghi thông tin nhân viên</li> <li>3. Nhấn chọn xác nhận (YES)</li> <li>4. Hệ thống thông báo xóa nhân viên thành công</li> </ol>
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.4: UC-Tìm kiếm thông tin nhân viên**

<b>Mã Use case</b>	UC-04
--------------------	-------

<b>Tên Use case</b>	Tìm kiếm thông tin nhân viên
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên tìm kiếm thông tin nhân viên
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đã đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Thông tin nhân viên sẽ hiển thị trên màn hình nếu có
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên muốn tìm kiếm thông tin nhân viên
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên chọn trang quản trị viên nhân viên trong thanh menu ở bên trái</li> <li>2. Nhấn vào icon “search” ở bên phải cột “username”</li> <li>3. Điền thông tin nhân viên cần tìm</li> <li>4. Nhấn nút “search”</li> </ol>
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.5: UC-Xem thông tin chi tiết của nhân viên**

<b>Mã Use case</b>	UC-05
<b>Tên Use case</b>	Xem thông tin chi tiết của nhân viên
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên xem thông tin nhân viên
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Thông tin nhân viên sẽ hiển thị trên màn hình
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên muốn xem thông tin chi tiết của nhân viên
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên nhấn vào trang quản trị viên nhân viên trong thanh menu ở bên trái</li> <li>2. Nhấn vào icon “search” ở cuối mỗi bản ghi thông tin nhân viên</li> </ol>
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.6: UC-Kích hoạt tài khoản**

<b>Mã Use case</b>	UC-06
<b>Tên Use case</b>	Kích hoạt tài khoản
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên muốn kích hoạt hoặc huỷ kích hoạt tài khoản
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Tài khoản sẽ chuyển sang trạng thái kích hoạt nếu trước đó tài khoản đang thao tác ở trạng thái chưa kích hoạt và ngược lại
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên muốn xem thông tin chi tiết của nhân viên
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên nhấn vào trang quản trị viên nhân viên trong thanh menu ở bên trái</li> <li>2. Nhấn vào icon “eye” ở cuối mỗi bản ghi thông tin nhân viên</li> <li>3. Hệ thống thông báo kích hoạt/huỷ kích hoạt thành công</li> </ol>
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.7: UC-Quản lý nhật ký**

<b>Mã Use case</b>	UC-07
<b>Tên Use case</b>	Quản lý nhật ký (logs)
<b>Tác nhân (Actor)</b>	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên xem nhật ký vào ra cửa
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Hệ thống hiển thị toàn bộ nhật kí có trong cơ sở dữ liệu
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên muốn xem nhật ký vào ra cửa

<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	1. Quản trị viên nhấn vào trang quản lý nhật ký trong thanh menu ở bên trái 2. Hệ thống chuyển hướng đến trang quản lý nhật ký
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.8: UC-Xuất tập tin báo cáo**

<b>Mã Use case</b>	UC-08
<b>Tên Use case</b>	Xuất tập tin báo cáo
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn</b> ( <i>Brief</i> )	Xuất file báo cáo dưới dạng excel (xlsx)
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Trình duyệt tự động tải file Excel
<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn xuất file báo cáo
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	1. Quản trị viên nhấn vào trang quản trị viên nhân viên trong thanh menu ở bên trái 2. Nhấn vào nút “Export (.xlsx)” 3. Hệ thống tự động tải file báo cáo
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.9: UC-Huấn luyện mô hình**

<b>Mã Use case</b>	UC-09
<b>Tên Use case</b>	Huấn luyện mô hình
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn</b> ( <i>Brief</i> )	Quản trị viên thêm nhân viên vào cơ sở dữ liệu nhận dạng
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Nhân viên có thể được nhận dạng bởi hệ thống camera

<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn thêm nhân viên vào hệ thống nhận dạng
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	1. Quản trị viên chọn file ảnh của nhân viên 2. Nhập id 3. Nhấn nút add 4. Hệ thống thông báo thêm nhân viên vào cơ sở dữ liệu nhận dạng thành công
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.10: UC-Xóa nhân viên khỏi hệ thống nhận dạng**

<b>Mã Use case</b>	UC-10
<b>Tên Use case</b>	Xóa nhân viên khỏi hệ thống nhận dạng
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn</b> ( <i>Brief</i> )	Quản trị viên xóa nhân viên khỏi cơ sở dữ liệu nhận dạng
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Nhân viên không còn được nhận dạng bởi hệ thống camera
<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn xóa nhân viên khỏi hệ thống nhận dạng
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	1. Quản trị viên nhập id 2. Nhấn nút “delete” 3. Hệ thống thông báo xóa nhân viên khỏi cơ sở dữ liệu nhận dạng thành công
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.11: UC-Khởi chạy nhận dạng**

<b>Mã Use case</b>	UC-11
<b>Tên Use case</b>	Khởi chạy nhận dạng
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal

<b>Mô tả ngắn</b> ( <i>Brief</i> )	Quản trị viên chạy chương trình chấm công tự động sử dụng công nghệ nhận dạng khuôn mặt
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Hệ thống bắt đầu quét khuôn mặt nhân viên tại cửa và chấm công
<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn chạy chương trình chấm công tự động sử dụng công nghệ nhận dạng khuôn mặt
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	Nhấn nút “START” ở bảng “FACE RECOGNITION”
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.12: UC-Tạm dừng nhận dạng**

<b>Mã Use case</b>	UC-12
<b>Tên Use case</b>	Tạm dừng nhận dạng
<b>Tác nhân</b> (Actor)	Quản trị viên
<b>Level</b>	User goal
<b>Mô tả ngắn</b> ( <i>Brief</i> )	Quản trị viên dừng chương trình chấm công tự động sử dụng công nghệ nhận dạng khuôn mặt
<b>Tiền điều kiện</b> (Preconditions)	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Hệ thống ngừng quét khuôn mặt nhân viên tại cửa và chấm công nhưng vẫn chạy chương trình phát hiện khuôn mặt
<b>Điều kiện kích hoạt use case</b> (Triggers specific business event)	Quản trị viên muốn chạy chương trình chấm công tự động sử dụng công nghệ nhận dạng khuôn mặt
<b>Luồng sự kiện chính</b> (Main scenario, basic flow)	Nhấn nút “STOP” ở bảng “FACE RECOGNITION”
<b>Luồng sự kiện phụ</b> (Extensions)	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.13: UC-Chấm công thủ công**

<b>Mã Use case</b>	UC-13
--------------------	-------

<b>Tên Use case</b>	Chấm công thủ công
<b>Tác nhân (Actor)</b>	Quản trị viên, nhân viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên chấm công thủ công cho những trường hợp đặc biệt
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Quản trị viên, nhân viên được chấm công thủ công thành công
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên, nhân viên có nhu cầu chấm công thủ công
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên nhấn vào trang chấm công thủ công trong thanh menu ở bên trái</li> <li>2. Nhập id</li> <li>3. Nhấn nút “Submit”</li> <li>3. Hệ thống thông báo chấm công thành công</li> </ol>
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

**Bảng 2.14: UC-Đăng ký nghỉ phép**

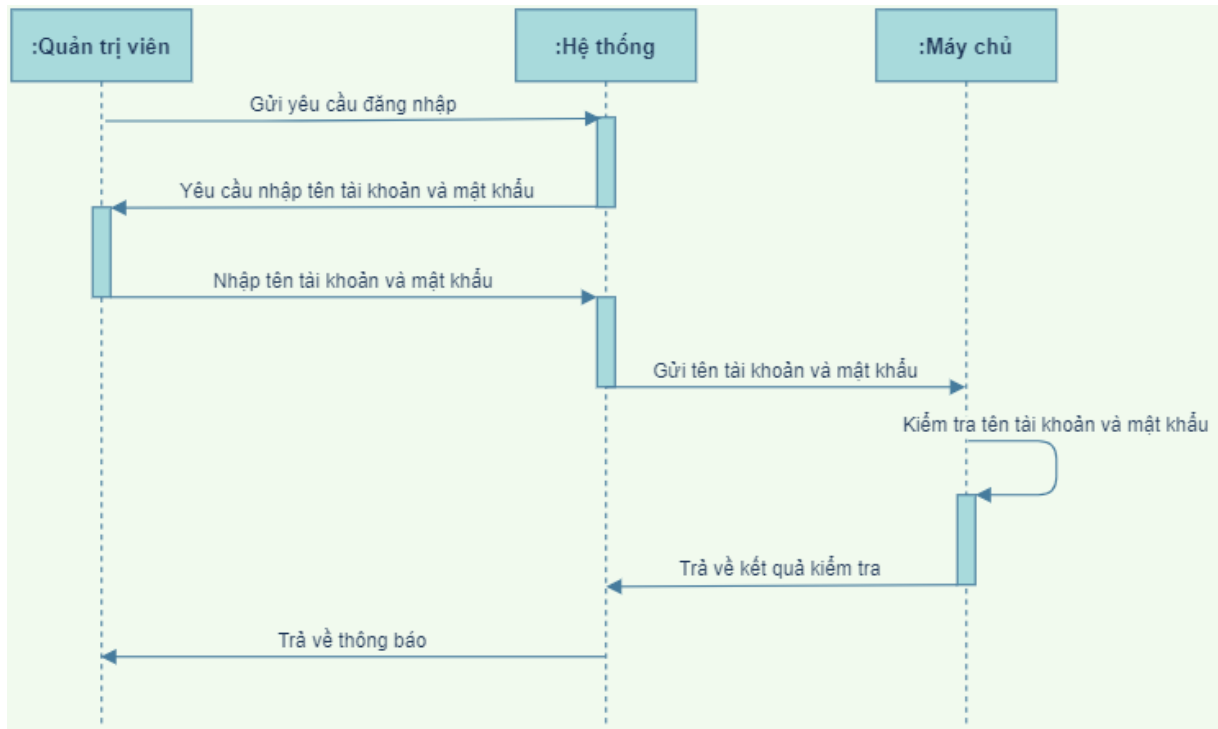
<b>Mã Use case</b>	UC-14
<b>Tên Use case</b>	Đăng ký nghỉ phép
<b>Tác nhân (Actor)</b>	Quản trị viên, nhân viên
<b>Level</b>	User goal
<b>Mô tả ngắn (Brief)</b>	Quản trị viên, nhân viên đăng ký nghỉ phép
<b>Tiền điều kiện (Preconditions)</b>	Quản trị viên đăng nhập thành công vào hệ thống
<b>Kết quả</b>	Quản trị viên, nhân viên đăng ký nghỉ phép thành công
<b>Điều kiện kích hoạt use case (Triggers specific business event)</b>	Quản trị viên, nhân viên có nhu cầu, mong muốn nghỉ phép
<b>Luồng sự kiện chính (Main scenario, basic flow)</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên nhấn vào trang đăng ký nghỉ phép trong thanh menu ở bên trái</li> <li>2. Nhập id</li> <li>3. Nhấn nút “register”</li> </ol>



	4. Hệ thống thông báo thành công
<b>Luồng sự kiện phụ (Extensions)</b>	Hệ thống báo lỗi dưới dạng văn bản

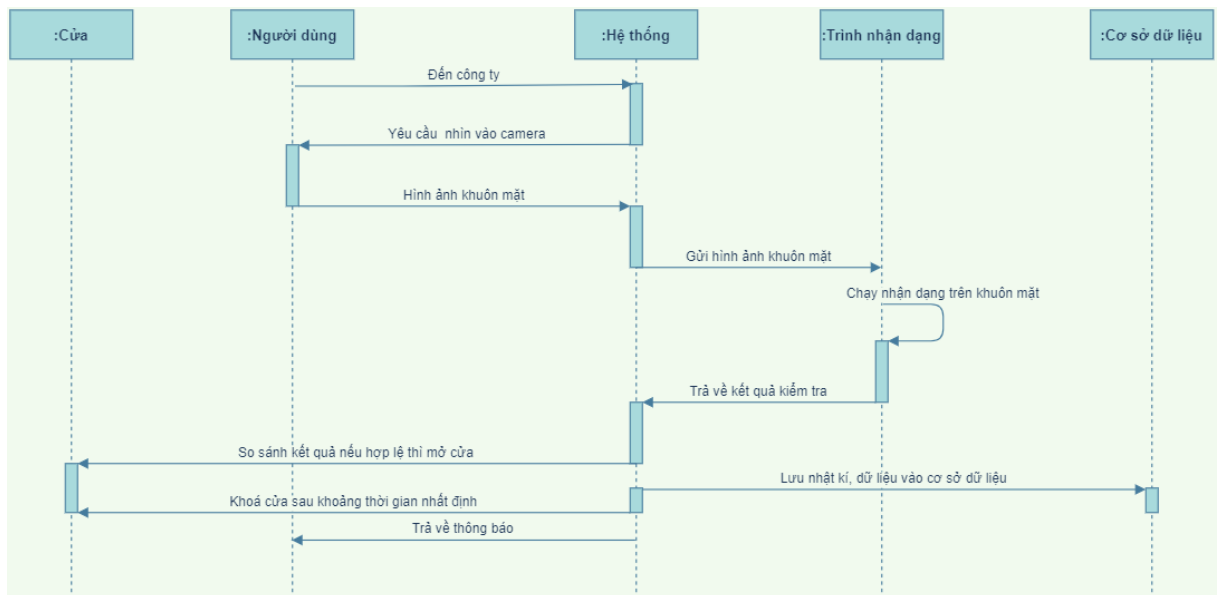
### 2.3.2. Biểu đồ tuần tự

#### a) Biểu đồ quá trình đăng nhập



**Hình 2.3: Biểu đồ tuần tự quá trình đăng nhập**

#### b) Biểu đồ quá trình nhận dạng chấm công



**Hình 2.4: Biểu đồ tuần tự quá trình nhận dạng chấm công**

### 2.3.3. Các thực thể

Tên lớp	Mô tả
Nhân viên/Quản trị viên	Thực thể là nhân viên và các quản trị viên trong công ty
Công	Thực thể là tổng số ngày làm việc mỗi tháng
Phạt	Thực thể là tổng số lần bị phạt việc mỗi tháng
Phòng	Thực thể là các phòng ban quản lý trong công ty
Báo cáo	Thực thể là các tập tin excel được hệ thống kết xuất
Lương	Thực thể là tổng số lương của mỗi nhân viên
Thông báo	Thực thể là nội dung thông báo được gửi đến người nhận
Thời gian vào	Thực thể là thời gian vào văn phòng của mỗi nhân viên
Thời gian ra	Thực thể là thời gian ra khỏi văn phòng của mỗi nhân viên

Mối quan hệ giữa các thực thể:

- Mỗi nhân viên đều thuộc một phòng.
- Mỗi nhân viên đều có một bản ghi Công, Phạt, Lương, Thời gian vào/ra.
- Mỗi quản trị viên có thể xuất báo cáo.

### 2.3.4. Kiến trúc phân tầng

Hệ thống được thiết kế theo hướng 3 tầng:

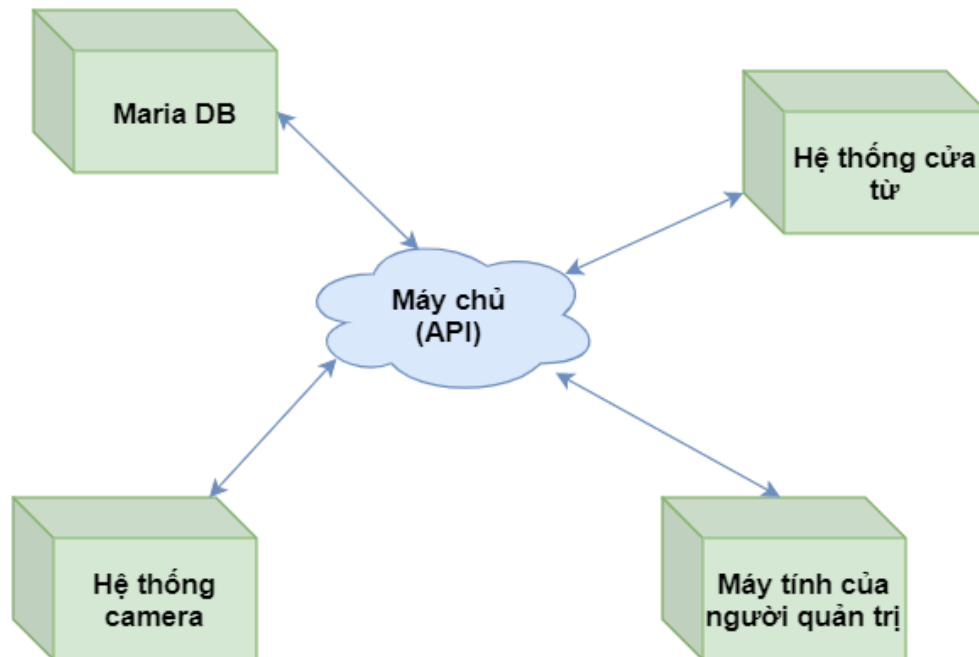
- Tầng trình bày: là tầng bao gồm các giao diện hiển thị và thao tác dành cho nhân viên, quản trị viên của chương trình.
- Tầng nghiệp vụ: là tầng xử lý trung gian giữa tầng trình bày và tầng truy cập dữ liệu. Tầng này chứa các lớp thực thể của chương trình.
- Tầng truy cập và lưu trữ dữ liệu: là tầng kết nối và thao tác trực tiếp với CSDL, xử lý dữ liệu và lưu trữ lại dữ liệu đã được xử lý.

### 2.3.5. Biểu đồ triển khai

Để triển khai hệ thống cần có các thiết bị và tổ chức như sau:

- Hệ thống camera: dùng để thu thập hình ảnh.
- Maria Database: chứa CSDL dùng chung của hệ thống.
- Hệ thống cửa từ: dùng để quản lý nhân viên ra vào.

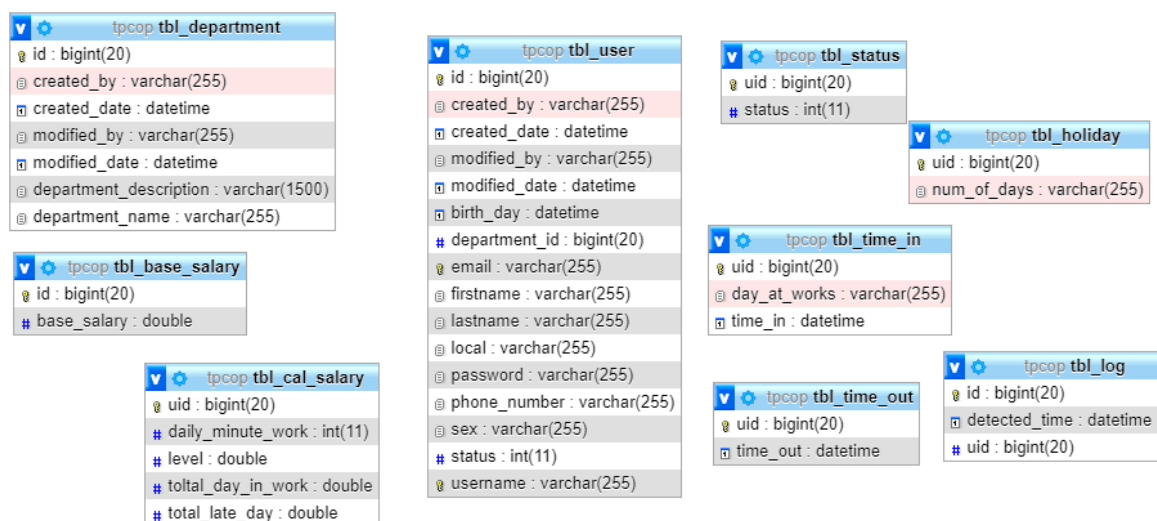
- Máy chủ: thao tác, xử lý tính toán và truy xuất trực tiếp tới Maria Database.
- Máy tính của người quản trị: thực hiện các chức năng của người quản trị hệ thống thông qua hệ thống máy chủ (API).
- Hệ thống mạng: sử dụng mạng Internet (nếu triển khai ứng dụng lên Server)



Hình 2.5: Biểu đồ triển khai hệ thống

## 2.4. Phân tích và thiết kế cơ sở dữ liệu

Dựa trên yêu cầu gửi và nhận thông tin của hệ thống, tôi xây dựng CSDL gồm một số bảng như sau:



Hình 2.6: Cơ sở dữ liệu

**Bảng 2.15: tbl\_user**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
Id	bigint(20)	
created_by	varchar(255)	
created_date	datetime	
modified_by	varchar(255)	
modified_date	datetime	
birth_day	datetime	
department_id	bigInt(20)	
Email	varchar(255)	
Firstname	varchar(255)	
Lastname	varchar(255)	
Local	varchar(255)	
Password	varchar(255)	Trường này được mã hoá md5
phone_number	varchar(255)	
Sex	varchar(255)	
Status	int(11)	
Username	varchar(255)	

Bảng tbl\_user chứa toàn bộ thông tin các nhân viên cũng như tài khoản trong công ty bao gồm cả mật khẩu đã được mã hoá.

**Bảng 2.16: tbl\_department**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
Id	bigint(20)	
created_by	varchar(255)	
created_date	Datetime	
modified_by	varchar(255)	
modified_date	Datetime	
department_description	varchar(1500)	
department_name	varchar(255)	

Bảng tbl\_department chứa thông tin các phòng ban trong công ty.

**Bảng 2.17: tbl\_base\_salary**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
Id	bigint(20)	

base_salary	double	Mức lương cơ bản
-------------	--------	------------------

Bảng tbl\_base\_salary lưu trữ mức lương cơ bản mục đích để tính tiền lương nhân viên sau này.

**Bảng 2.18: tbl\_cal\_salary**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
Uid	bigint(20)	id của người dùng
daily_minute_work	int(11)	Tổng thời gian làm việc trong ngày
Level	double	Hệ số lương
total_day_in_work	double	Tổng số ngày làm
total_late_day	double	Tổng số lần đi trễ/phạt

Bảng tbl\_cal\_salary chứa dữ liệu phục vụ mục đích tính tiền lương nhân viên. Mỗi nhân viên chỉ có 1 bản ghi.

**Bảng 2.19: tbl\_holiday**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
uid	bigint(20)	id của người dùng
num_of_days	varchar(255)	Ngày nghỉ phép

Bảng tbl\_holiday chứa các ngày nghỉ phép của toàn bộ nhân viên. Mỗi nhân viên chỉ có một bản ghi.

**Bảng 2.20: tbl\_log**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
Uid	bigint(20)	id của người dùng
detected_time	Datetime	
Uid	bigint(20)	

Bảng tbl\_log lưu trữ thời gian ra vào văn phòng của toàn bộ nhân viên. Mỗi nhân viên có thể có nhiều bản ghi.

**Bảng 2.21: tbl\_time\_in**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
uid	bigint(20)	id của người dùng
day_at_works	varchar(255)	
time_in	Datetime	

Bảng tbl\_time\_in ghi lại thời gian vào văn phòng của toàn bộ nhân viên. Mỗi nhân viên chỉ có 1 bản ghi.

**Bảng 2.22: tbl\_time\_out**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
uid	bigint(20)	id của người dùng
time_out	datetime	

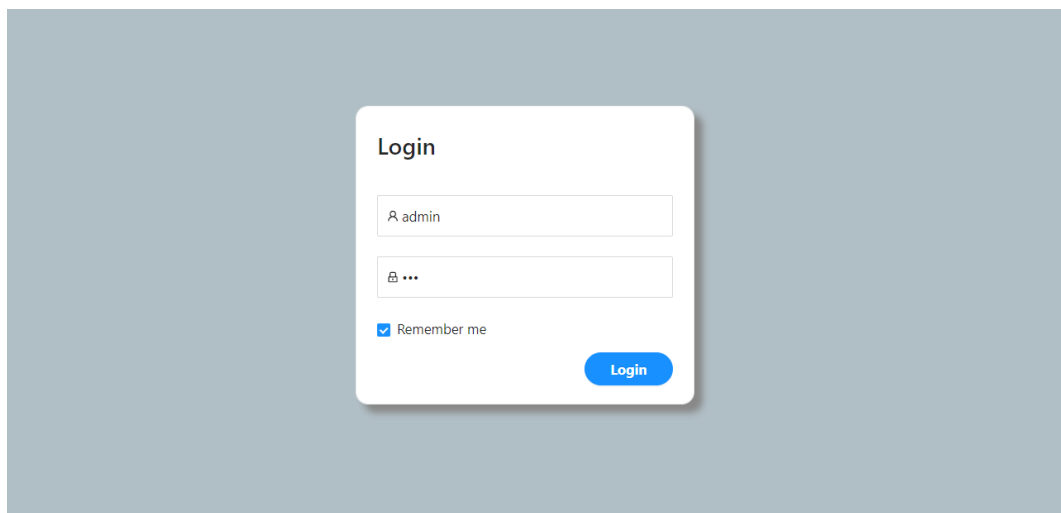
Bảng tbl\_time\_out ghi lại thời gian ra khỏi văn phòng của toàn bộ nhân viên. Mỗi nhân viên chỉ có 1 bản ghi.

**Bảng 2.23: tbl\_status**

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
uid	bigint(20)	id của người dùng
status	int(11)	1 vào, -1 ra

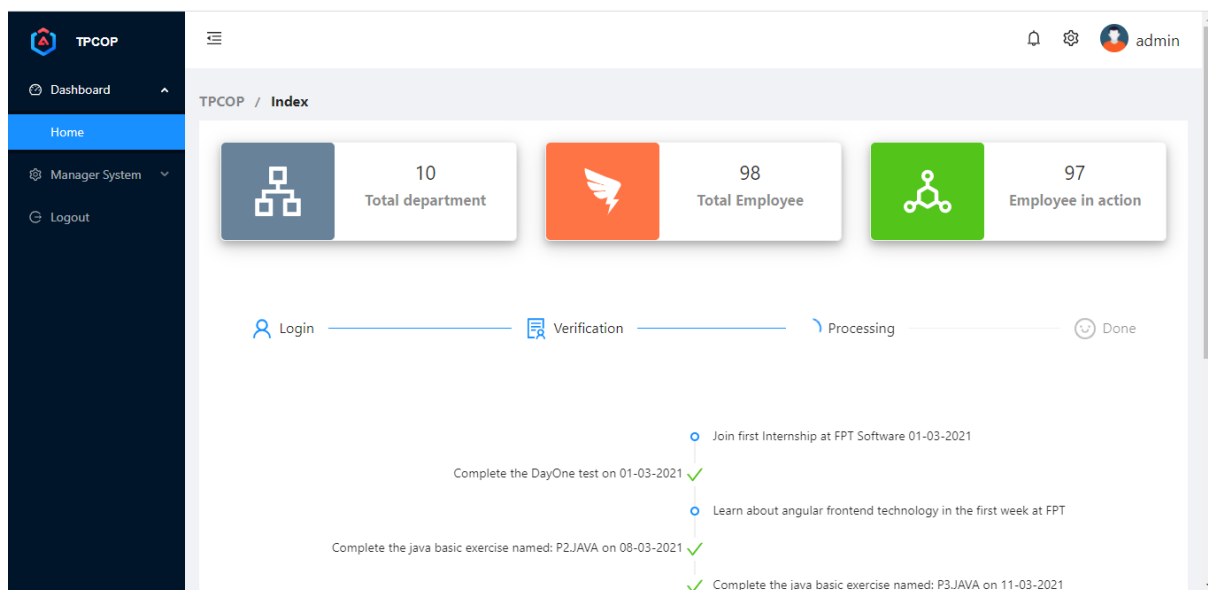
Bảng tbl\_status ghi lại trạng thái của nhân viên đang làm việc hay ra ngoài. Mỗi nhân viên chỉ có 1 bản ghi.

## 2.5. Thiết kế các màn hình chức năng

**Hình 2.7: Màn hình đăng nhập (website)**

### Các thành phần trong màn hình đăng nhập:

#	Đối tượng	Bắt buộc	Mô tả
1	Textbox Tên đăng nhập	X	Nhập tên đăng nhập
2	Textbox Mật khẩu	X	Nhập mật khẩu
3	Checkbox Lưu mật khẩu		Ghi nhớ mật khẩu
4	Button Login	X	Đăng nhập



**Hình 2.8: Màn hình trang chủ (website)**

**Các thành phần trong màn hình trang chủ:**

#	Đối tượng	Bắt buộc	Mô tả
1	Card Phòng ban		Hiển thị tổng số phòng ban
2	Card Nhân viên		Hiển thị tổng số nhân viên
3	Card Nhân viên còn đang hoạt động		Hiển thị tổng số nhân viên đang hoạt động

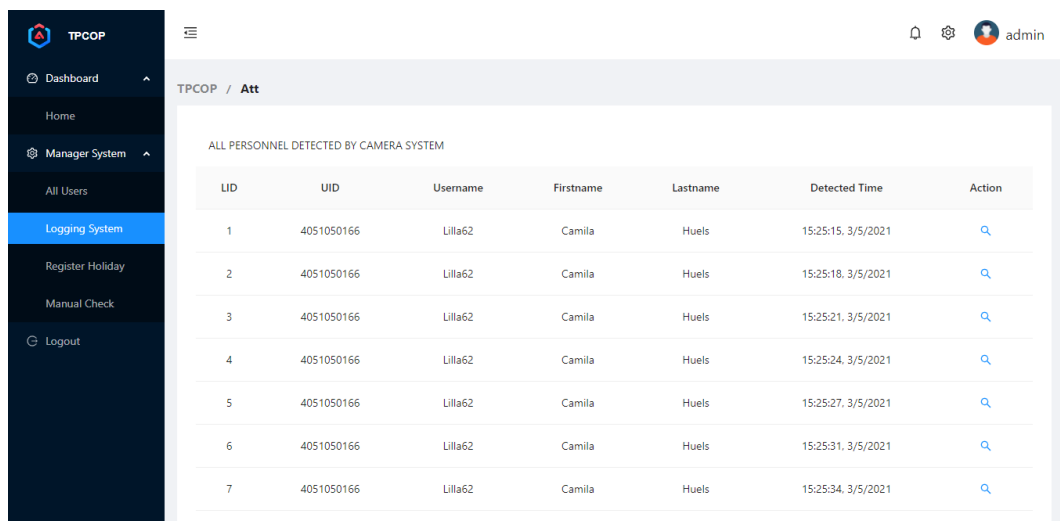
ID	Username	Created Date	Created By	Modified Date	Modified By	Status	Actions
-1	Naomi.Brakus95	14:49:04, 5/5/2021	SYSTEM	21:03:14, 7/5/2021	Marjory.Stroman	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
1	Marjory.Stroman	14:49:04, 5/5/2021	SYSTEM	15:25:36, 5/5/2021	Earline90@yahoo.com	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
2	Eveline84	14:49:04, 5/5/2021	SYSTEM	21:11:30, 7/5/2021	Marjory.Stroman	✗	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
3	Aisha_Metz	14:49:04, 5/5/2021	SYSTEM	21:03:09, 7/5/2021	Marjory.Stroman	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
7	Blake0	20:15:47, 11/5/2021	SYSTEM	20:15:47, 11/5/2021	SYSTEM	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
8	Lessie.Bernier	20:15:47, 11/5/2021	SYSTEM	20:15:47, 11/5/2021	SYSTEM	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
9	Iaren.Graham	20:15:47, 11/5/2021	SYSTEM	20:15:47, 11/5/2021	SYSTEM	✓	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>

**Hình 2.9: Màn hình quản lý người dùng (website)**

**Các thành phần trong màn hình quản lý người dùng:**

#	Đối tượng	Bắt buộc	Mô tả
---	-----------	----------	-------

1	Table Nhân sự		Hiển thị thông tin toàn bộ nhân viên
2	IconButton Tìm kiếm		Cho phép tìm kiếm thông tin nhân viên
3	IconButton Sắp xếp		Sắp xếp lại danh sách
4	Icon Trạng thái		Hiển thị trạng thái của nhân viên
5	IconButton Xóa		Xóa thông tin nhân viên đang thao tác khỏi hệ thống
6	IconButton Kích hoạt		Kích hoạt hoặc hủy kích hoạt tài khoản của nhân viên
7	Button Xuất báo cáo		Xuất báo cáo DSNS và bảng chấm công dưới dạng Excel

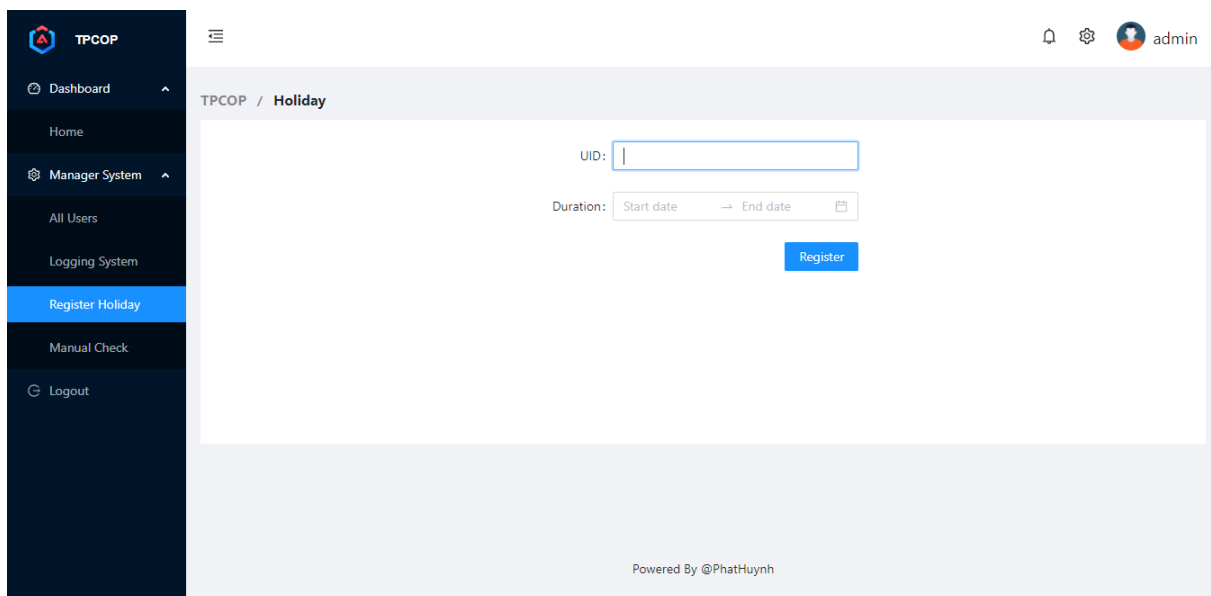


**Hình 2.10: Màn hình quản lý nhật kí (website)**

**Các thành phần trong màn hình quản lý nhật kí:**

#	Đối tượng	Bắt buộc	Mô tả
1	Table Nhật kí		Hiển thị toàn bộ nhật kí vào ra hệ thống ghi nhận được
2	IconButton Tìm kiếm		Cho phép tra cứu thông tin nhân viên

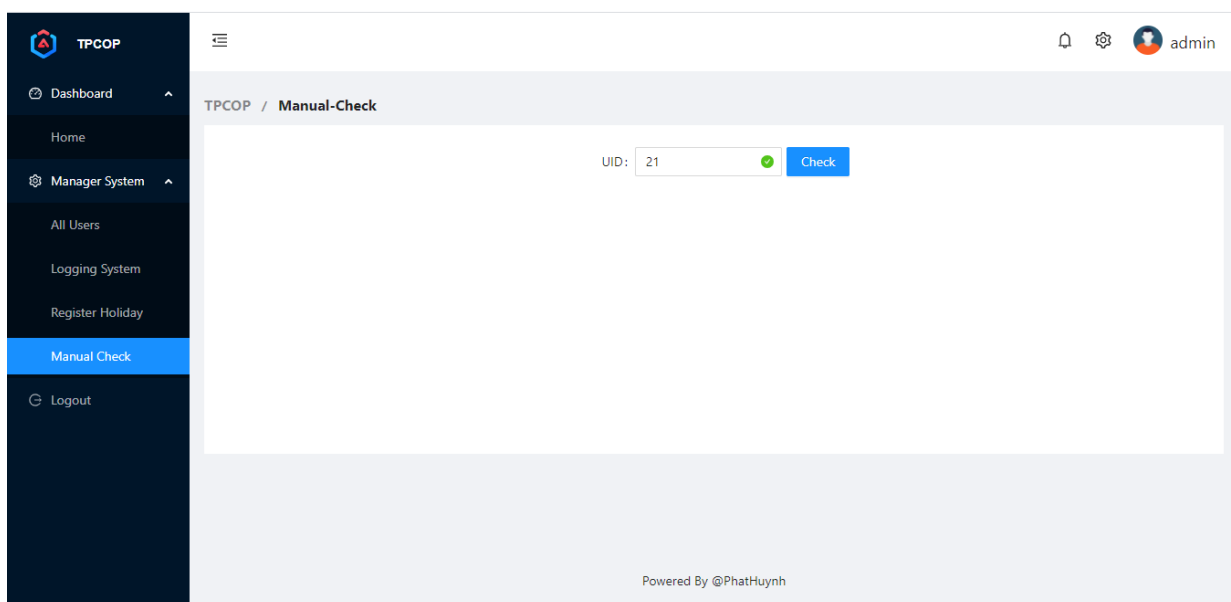




**Hình 2.11: Màn hình đăng ký nghỉ phép (website)**

**Các thành phần trong màn hình đăng ký nghỉ phép:**

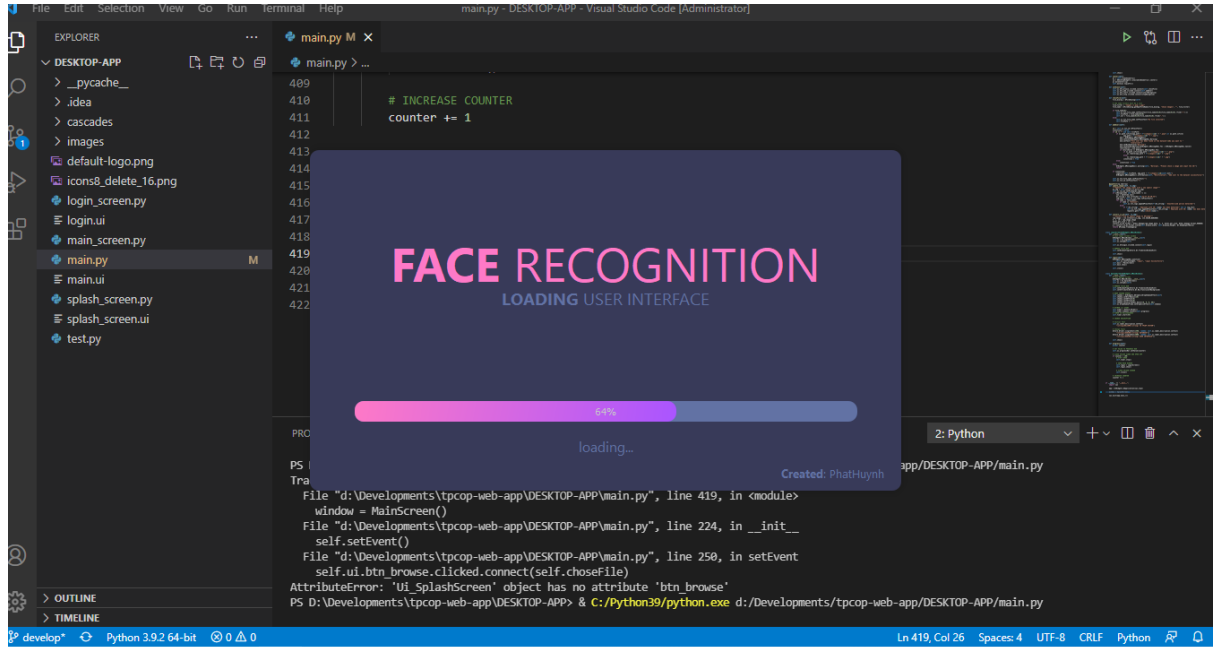
#	Đối tượng	Bắt buộc	Mô tả
1	Textbox Id	X	Nhập Id của nhân viên
2	TimePicker Thời gian bắt đầu	X	Chọn thời gian bắt đầu nghỉ phép
3	TimePicker Thời gian kết thúc	X	Chọn thời gian kết thúc nghỉ phép
4	Button Đăng ký	X	Đăng ký nghỉ phép cho nhân viên



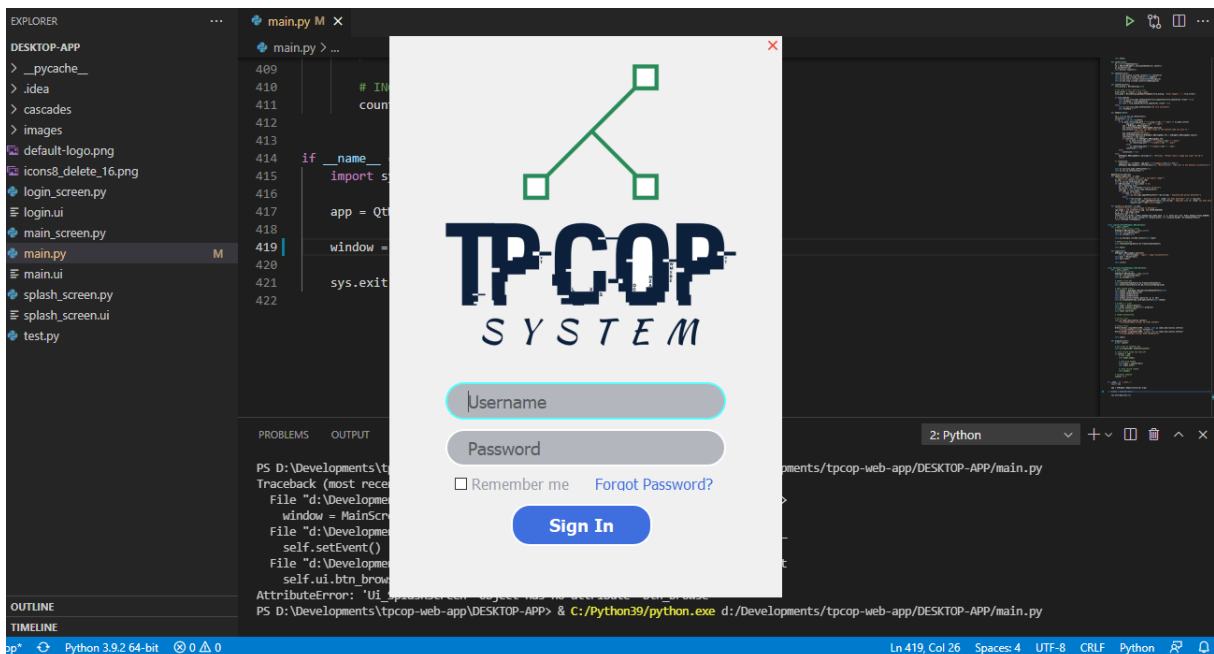
**Hình 2.12: Màn hình chấm công thủ công (website)**

**Các thành phần trong màn hình chấm công thủ công:**

#	Đối tượng	Bắt buộc	Mô tả
1	Textbox Id	X	Nhập Id của nhân viên
2	Button Điểm danh	X	Chấm công cho nhân viên



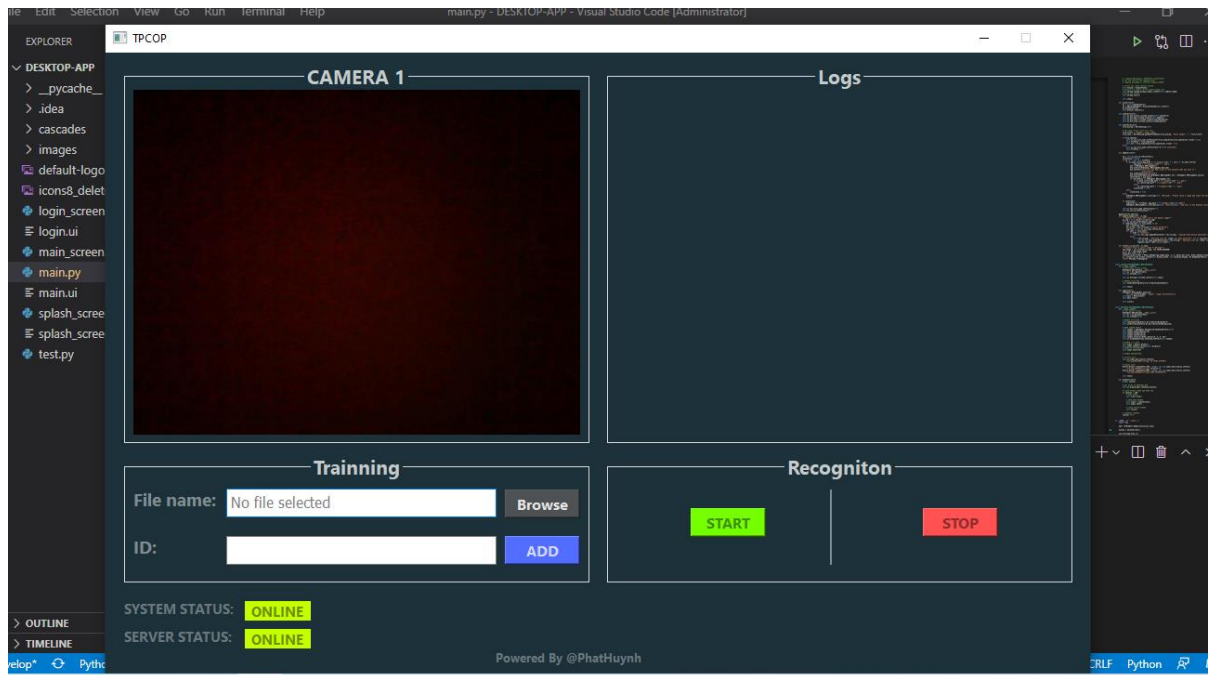
Hình 2.13: Màn hình giới thiệu (Desktop)



Hình 2.14: Màn hình đăng nhập (Desktop)

Các thành phần trong màn hình đăng nhập (Desktop):

#	Đối tượng	Bắt buộc	Mô tả
1	Textbox Tên đăng nhập	X	Nhập tên đăng nhập
2	Textbox Mật khẩu	X	Nhập mật khẩu
3	Checkbox Lưu mật khẩu		Ghi nhớ mật khẩu
4	Label Quên mật khẩu		Chuyển sang trang quên mật khẩu
5	Button Đăng nhập	X	Đăng nhập



Hình 2.15: Màn hình trang chủ (Desktop)

#### Các thành phần trong màn hình trang chủ (Desktop):

#	Đối tượng	Bắt buộc	Mô tả
1	Label Hình ảnh	X	Hiển thị dữ liệu hình ảnh lấy từ camera
2	Textbox Tên file	X	Hiển thị tên file được chọn
3	Textbox Id	X	Nhập id của nhân viên
4	Button Chọn file	X	Mở hộp thoại chọn file
5	Textbox Logs	X	Hiển thị những người được phát hiện bởi hệ thống
6	Button Nhận dạng	X	Khởi chạy trình nhận dạng
7	Button Dừng nhận dạng	X	Tạm dừng trình nhận dạng

## CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI ỨNG DỤNG

Chương này trình bày thêm một số thư viện được sử dụng để tổ chức cài đặt các chức năng của hệ thống và triển khai thử nghiệm hệ thống đã cài đặt theo mô hình client/server.

### 3.1. Các chức năng của hệ thống

Người quản trị hệ thống có khả năng quản lý toàn bộ hệ thống theo luồng dữ liệu khép kín được phân tích thiết kế hợp lý. Cấu trúc dữ liệu logic, dễ thao tác, dễ điều khiển.

Một số chức năng chính của hệ thống:

- Đăng nhập
- Quản lý nhân viên
- Quản lý nhật ký
- Chấm công tự động
- Chấm công thủ công
- Đào tạo mô hình
- Xuất tập tin báo cáo dưới dạng Excel

### 3.2. Lựa chọn công nghệ cài đặt

#### 3.2.1. Thư viện Ant Design of Angular

Angular là một Javascript Framework do Google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript, HTML và TypeScript. Angular cung cấp các tính năng tích hợp cho animation, http service và có các tính năng như auto-complete, navigation, toolbar, menus,... Code được viết bằng TypeScript, biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

NG-ZORRO (Ant Design of Angular) là một thư viện thành phần Angular UI cấp doanh nghiệp dựa trên Ant Design, tất cả các thành phần đều là mã nguồn mở và được sử dụng miễn phí theo giấy phép của MIT. Một số ưu điểm như:

- Một ngôn ngữ thiết kế giao diện người dùng cấp doanh nghiệp cho các ứng dụng Angular.
- Hơn 60 thành phần Angular chất lượng cao.
- Được viết bằng TypeScript với các kiểu hoàn chỉnh được xác định.
- Hỗ trợ chế độ OnPush, hiệu suất cao.

- Tùy biến chủ đề mạnh mẽ đến từng chi tiết.
- Hỗ trợ quốc tế hóa cho hàng chục ngôn ngữ.

### 3.2.2. Thư viện PyQt5

PyQt là một trong những ràng buộc Python đa nền tảng được ưa chuộng triển khai thư viện Qt cho Framework phát triển ứng dụng Qt (thuộc sở hữu của Nokia).

Hiện tại, PyQt có sẵn cho Unix/Linux, Windows, Mac OS X và Sharp Zaurus. Nó kết hợp những đặc tính tốt nhất của Python và Qt và tùy thuộc vào từng lập trình viên để quyết định tạo một chương trình bằng cách viết mã lệnh hay sử dụng Qt Designer để tạo các hộp thoại trực quan.

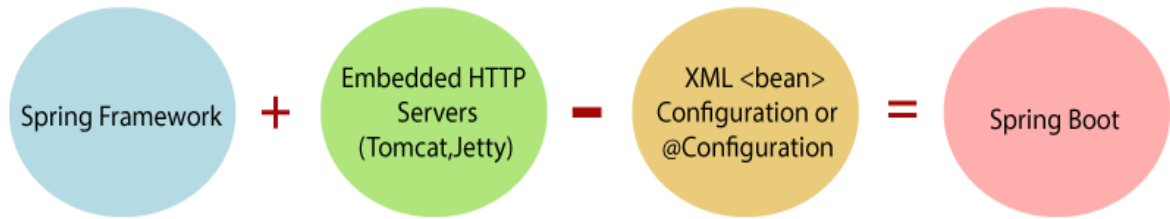
PyQt có sẵn trong cả giấy phép thương mại cũng như GPL (General Public License). Mặc dù một số tính năng có thể không có trong phiên bản miễn phí, nhưng nếu ứng dụng của chúng ta là mã nguồn mở thì có thể sử dụng nó miễn phí.

### 3.2.3. Spring Boot Framework

Spring Framework cung cấp các giải pháp cho nhiều thách thức kỹ thuật mà các nhà phát triển phần mềm Java và các tổ chức muốn xây dựng ứng dụng dựa trên nền tảng Java phải đối mặt với Java platform. Spring Framework cung cấp một mô hình nhất quán, có thể áp dụng vào hầu hết các loại ứng dụng được tạo trên nền tảng Java ngày nay. Framework Spring thực thi một phương thức làm việc dựa trên những thói quen thực hành tốt nhất và những tiêu chuẩn công nghiệp, đồng thời tạo cơ hội cho nhiều lĩnh vực trong Java sử dụng nó.

Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) như:

- Phát triển ứng dụng nhanh.
- Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
- Spring Boot không yêu cầu cấu hình XML
- Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho người lập trình.



**Hình 3.1: Cấu tạo Spring Boot Framework**

#### **3.2.4. Thư viện OpenCV**

Đã giới thiệu ở Chương 1

#### **3.2.5. Thư viện face-recognition**

Đã giới thiệu ở Chương 1

#### **3.2.6. Xampp**

Xampp là chương trình tạo máy chủ Web được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server, phpMyAdmin và các công cụ như phpMyAdmin. Không như Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

### **3.3. Yêu cầu hệ thống**

#### **3.3.1. Phần cứng (Hardware)**

Processor: Intel/ADM

Memory: 4GB

HDD Space: 500MB

Camera

#### **3.3.2. Phần mềm (Software)**

Hệ điều hành: Windows 7 64 Bit, Windows 8.1 64 Bit, Windows 8 64 Bit, Windows 10 64 Bit

Phần mềm cài đặt: JAVA, PYTHON, NODEJS, Maria DB

### **3.4. Xây dựng ứng dụng**

Website Application được viết thuần bằng ngôn ngữ TypeScript, trên nền tảng Angular Framework. Sử dụng bộ components được thiết kế sẵn của NG-ZORRO Framework. Mục đích thiết kế các giao diện để phục vụ quản lý, hiển thị thông tin.

Desktop Application sử dụng các thư viện: PyQt5, OpenCV, Face-Recognition được viết bằng ngôn ngữ Python. Python không phải là ngôn ngữ chuyên về front-end

Trung tâm xử lý dữ liệu (API) sử dụng Spring Boot Framework và được xây dựng bằng ngôn ngữ JAVA. Thao tác, truy xuất trực tiếp tới CSDL. Ở phần này yêu cầu lập trình xử lý tương đối phức tạp về logic và tính toán. Cả hai ứng dụng trên nền tảng website và desktop đều phải tương tác với API này để xử lý tính toán, không truy xuất trực tiếp đến CSDL.

Ứng dụng đã được triển khai trên localhost để chạy thử nghiệm, cho kết quả nhận dạng khuôn mặt khi sử dụng thư viện Face-Recognition với độ chính xác khoảng 99.38% khi sử dụng mô hình do thư viện này đào tạo. Quá trình chấm công được thực hiện theo thời gian thực. Kết quả của quá trình chấm công được cập nhật theo ngày, tháng và năm. Cứ mỗi ngày nếu nhân viên đi làm sau 8 giờ sáng sẽ bị tính 1 lần phạt, thời gian làm việc của nhân viên dưới 6 tiếng cũng sẽ bị tính 1 lần phạt. Cứ ngày cuối cùng của mỗi tháng người quản trị sẽ xuất tập tin báo cáo để theo dõi, xử lý.

Quy Nhơn, Ngày 5 Tháng 5 Năm 2021  
Trưởng Bộ Phận Nhân Sự

38

## KẾT LUẬN

Đề tài đã đạt được một số kết quả sau:

### ❖ Về kiến thức

Sau thời gian thực hiện khóa luận tốt nghiệp tôi đã đạt được nhiều tiến bộ đáng kể cả về mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình. Có thể nói, thông qua khóa luận này, tôi đã đạt được:

- Hiểu biết nhiều hơn về các kỹ thuật và kỹ xảo trong lập trình Java, TypeScript, JavaScript, Python, HTML, CSS.
- Phân tích thiết kế, xây dựng ứng dụng theo hướng chuyên nghiệp hơn.
- Hiểu hơn về công nghệ nhận dạng khuôn mặt.
- Nâng cao tinh thần tự học, tự nghiên cứu.

### ❖ Về chương trình

- Chương trình cơ bản thực hiện được các yêu cầu đưa ra của đề tài tốt nghiệp
- Ứng dụng chấm công được xây dựng và đáp ứng một số chức năng chính:
  - + Đăng nhập
  - + Quản lý nhân viên
  - + Quản lý nhật ký
  - + Chấm công tự động
  - + Cho phép linh hoạt chuyển sang chấm công thủ công trong một số trường hợp đặc biệt.
  - + Xử lý phạt với các trường hợp nhân viên đi làm trễ, về sớm
  - + Xuất tập tin báo cáo ngày cuối tháng theo mẫu, định dạng trước

### ❖ Hạn chế

- Do thời gian dành cho việc phân tích thiết và thiết kế hệ thống chưa nhiều so với một đề tài tương đối rộng và phong phú nên không tránh khỏi những thiếu sót nhất định. Bên cạnh đó, chương trình còn một số chức năng chưa hoàn thiện như kỳ vọng.
- Hơn nữa việc cố gắng học hỏi và sử dụng thêm nhiều ngôn ngữ lập trình, nhiều Framework để xây dựng thành công một ứng dụng với nhiều thành phần và đa ngôn ngữ cũng đem đến những trở ngại trong quá trình hoàn thành hệ thống.
- Đây là một ứng dụng dựa trên mô hình client/server, tách các phần giao diện, xử lý logic và CSDL ra thành các phần riêng biệt, với việc truy xuất dữ liệu trong CSDL liên tục. Để đáp ứng được điều này, hệ thống đòi hỏi một máy chủ mạnh



và ổn định. Nên quá trình triển khai thử nghiệm hệ thống cũng khó khăn và mất nhiều thời gian.

#### ❖ **Hướng phát triển**

Về cơ bản hệ thống có thể triển khai ứng dụng vào thực tế, nhưng tôi nhận thấy cần phải tiếp tục thực hiện một số công việc, cải tiến như sau:

- Xử lý các lỗi chằng chịt hơn.
- Mở rộng bài toán cho nhiều doanh nghiệp với nhiều ràng buộc riêng biệt.
- Bổ sung thêm nhiều mẫu báo cáo, thống kê.
- Nâng cấp hệ thống bảo mật (ví dụ thay mã hoá mật khẩu đăng nhập từ MD5 thành SHA)
- Phát triển các tính năng hỗ trợ khác.

## TÀI LIỆU THAM KHẢO

- [1]. Ng-Book: The Complete Book on Angular 4 - Ari Lerner, Felipe Coury, and Nate Murray – 2017.
- [2]. Spring in Action - Craig Walls – 2005.
- [3]. Artificial Intelligence: A Modern Approach - Peter Norvig and Stuart J. Russell – 1994.
- [4]. Effective TypeScript: 62 Specific Ways to Improve Your TypeScript - Dan Vanderkam – 2019.
- [5]. Learning Python - Mark Lutz – 1999.
- [6]. HTML & CSS: Design and Build Web Sites - Jon Duckett – 2011.
- [7]. <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- [8]. <https://viblo.asia/p/hoc-spring-boot-bat-dau-tu-dau-6J3ZgN7WKmB>
- [9]. <https://docs.opencv.org/master/index.html>
- [10]. [https://face-recognition.readthedocs.io/en/latest/face\\_recognition.html](https://face-recognition.readthedocs.io/en/latest/face_recognition.html)
- [11]. <https://www.apachefriends.org/index.html>
- [12]. <https://angular.io/docs>
- [13]. <https://spring.io/projects/spring-boot>
- [14]. [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- [15]. <https://ng.ant.design/docs/introduce/en>
- [16]. [https://vi.wikipedia.org/wiki/Hệ\\_thống\\_nhận\\_dạng\\_khuôn\\_mặt](https://vi.wikipedia.org/wiki/Hệ_thống_nhận_dạng_khuôn_mặt)
- [17]. [https://en.wikipedia.org/wiki/Scale-invariant\\_feature\\_transform](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform)
- [18]. [https://en.wikipedia.org/wiki/Speeded\\_up\\_robust\\_features](https://en.wikipedia.org/wiki/Speeded_up_robust_features)
- [19]. <https://cseweb.ucsd.edu/classes/wi14/cse152-a/fisherface-pami97.pdf>
- [20]. [https://en.wikipedia.org/wiki/Local\\_binary\\_patterns](https://en.wikipedia.org/wiki/Local_binary_patterns)
- [21]. <https://en.wikipedia.org/wiki/Eigenface>