

ĐẠI HỌC ĐÀ NẪNG TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA CÔNG NGHỆ THÔNG TIN





BÁO CÁO PBL6: ĐỒ ÁN CHUYÊN NGÀNH

ĐỂ TÀI: " XÂY DỰNG WEBSITE NHẬN DIỆN THỂ SINH VIÊN KẾT HỢP SO KHỚP KHUÔN MẶT ĐỂ HỖ TRỢ ĐIỂM DANH TRONG THI ONLINE "

GIẢNG VIÊN HƯỚNG DẪN: TS. PHẠM MINH TUẦN

SINH VIÊN THỰC HIỆN:

DÕ VĂN TRÌNH
 PHẠM TRẦN NHÂN NGHĨA
 CHÂU BẢO NHẬT VY
 102180276
 18N16
 18N16
 102180280
 18N16

Đà Nẵng, THÁNG 12/2021



MỤC LỤC

DANH MỤC HINH ANH	5
DANH MỤC BẢNG BIỂU	6
LỜI CẨM ƠN	7
CHƯƠNG 1 : MỞ ĐẦU	8
1.1 Đặt vấn đề	8
1.2 Mục đích	8
1.3 Quá trình triển khai	9
1.4 Phạm vi	9
1.5 Các vấn đề cần giải quyết	10
1.6 Giải pháp tổng quan	10
1.7 Công cụ phát triển	11
CHƯƠNG 2: TỔNG QUAN	12
2.1 Bài toán so khớp khuôn mặt	12
2.2 Bài toán nhận diện thông tin	13
CHƯƠNG 3: CƠ SỞ LÝ THUYẾT	14
3.1 Ngôn ngữ lập trình Python	14
3.2 Mô hình Message Queue	14
3.3 Cơ sở dữ liệu	15
3.3.1 MySQL	15
3.3.2 Redis	15
3.4 Website	16
3.4.1 Web Server Nginx	16
3.4.2 GoAccess	17
3.4.3 NodeJS	17
3.4.4 VueJS	17
3.4.5 Giao thức HTTPS	17

3.5 Kiểm thử phần mềm	18
3.5.1 Tổng quan	18
3.5.2 Các loại kiểm thử	18
3.5.3 Giới thiệu Selenium	19
3.5.4 Công cụ kiểm thử k6	19
CHƯƠNG 4 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG	20
4.1 Phân tích chức năng	20
4.1.1 Tác nhân	20
4.1.2 Biểu đồ ca sử dụng	20
4.1.3 Biểu đồ hoạt động	21
4.2 Thiết kế cơ sở dữ liệu	26
4.3 Thiết kế hệ thống	27
4.3.1 Các thành phần trong hệ thống	27
4.3.2 Quy trình xử lý	28
CHƯƠNG 5: TRIỂN KHAI HỆ THỐNG	29
5.1 Bài toán nhận diện thông tin thẻ sinh viên	29
5.1.1 Tách thể sinh viên từ ảnh đầu vào (Card Cropper)	29
5.1.2 Phát hiện thông tin sinh viên (Card Detector)	30
5.1.3 Đọc thông tin sinh viên (Card Reader)	33
5.2 Bài toán so khớp khuôn mặt	34
5.2.1 Phát hiện khuôn mặt (Face Detector)	34
5.2.2 Căn chỉnh khuôn mặt (Face Alignment)	34
5.2.3 Tạo vector đặc trưng (Face Embedding)	35
CHƯƠNG 6: CẤU HÌNH MÁY CHỦ	36
6.1 Cấu hình tường lửa cho máy chủ Ubuntu Server	36
6.2 Cài đặt Nginx	36
6.3 Viết các file cấu hình dùng chung	
6.4 Cấu hình dịch vụ HTTP và Reverse Proxy	38
6.5 Cấu hình SSL	

6.6 Cấu hình quản lý lưu lượng truy cập	41
CHƯƠNG 7: KIỂM THỬ HỆ THỐNG	45
7.1 Kiểm thử tự động các chức năng	45
7.1.1 Cài đặt Selenium IDE	45
7.1.2 Sử dụng Selenium IDE để kiểm thử trang web	45
7.2 Kiểm thử hiệu năng	47
7.2.1 Kịch bản kiểm thử	47
7.2.2 Tiến hành kiểm thử	47
CHƯƠNG 8: ĐÁNH GIÁ KẾT QUẢ	
8.1 Các chức năng đã triển khai:	
8.1.1 Giảng viên	51
8.1.2 Sinh viên	
8.2 Đánh giá	
CHƯƠNG 9: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	
9.1 Kết luận	
9.2 Hướng phát triển	
TÀI LIỆU THAM KHẢO	

DANH MỤC HÌNH ẢNH

Hình 1: Mô hình tông quan	10
Hình 2: Mô hình cho bài toán so khớp khuôn mặt	12
Hình 3: Mô hình cho bài toán nhận diện thông tin	13
Hình 4: Mô hình Message Queue	14
Hình 5: Cơ sở dữ liệu Redis	16
Hình 6: Biểu đồ ca sử dụng của Giảng viên	20
Hình 7: Biểu đồ ca sử dụng của Sinh viên	21
Hình 8: Sơ đồ hoạt động Đăng nhập	21
Hình 9: Sơ đồ hoạt động Xem ca thi	22
Hình 10: Sơ đồ hoạt động Thêm ca thi	22
Hình 11: Sơ đồ hoạt động Sửa ca thi	23
Hình 12: Sơ đồ hoạt động Sửa kết quả điểm danh	24
Hình 13: Sơ đồ hoạt động Xem kết quả điểm danh	24
Hình 14: Sơ đồ hoạt động kiểm tra khung giờ điểm danh	25
Hình 15: Sơ đồ hoạt động Điểm danh	25
Hình 16: Mô tả cơ sở dữ liệu	26
Hình 17: Kiến trúc hệ thống	27
Hình 18: Mô hình giải quyết bài toán nhận diện thông tin thẻ sinh viên	29
Hình 19: Mục tiêu của khối Tách thẻ sinh viên	29
Hình 20: Mục tiêu khối phát hiện thông tin sinh viên	30
Hình 21: Dữ liệu huấn luyện	31
Hình 22: Mô tả file nhãn dữ liệu	31
Hình 23: Kết quả quá trình huấn luyện	32
Hình 24: Quá trình huấn luyện mô hình	32
Hình 25: Khối đọc thông tin sinh viên	33
Hình 26: Mô hình giải quyết bài toán so khớp khuôn mặt	
Hình 27: Kết quả phát hiện khuôn mặt	
Hình 28: Khoảng cách giữa 2 khuôn mặt tương đồng và không tương đồng	35
Hình 29: Kết quả cấu hình tường lửa	36
Hình 30: Kết quả cài đặt Nginx	36
Hình 31: Nội dung file general.conf	37
Hình 32: Nội dung file security.conf	37
Hình 33: Nội dung file proxy.conf	38
Hình 34: Giao diện đăng nhập trang quản lý lưu lượng truy cập	44
Hình 35: Trang quản lý lưu lượng truy cập	44
Hình 36: Cài đặt tiện ích Selenium	45

Hình 37: Mô tả tiện ích Selenium	45
Hình 38: Tạo project trong Selenium	46
Hình 39: Chuẩn bị kiểm thử bằng Selenium	46
Hình 40: Kiểm thử chức năng Đăng nhập	46
Hình 41: Các chức năng đã kiểm thử thành công	47
Hình 42: Giao diện tạo kịch bản kiểm thử	48
Hình 43: Kịch bản kiểm thử hiệu năng đã tạo	49
Hình 44: Kết quả kiểm thử hiệu năng (1)	49
Hình 45: Kết quả kiểm thử hiệu năng (2)	50
Hình 46: Mô tả chức năng Đăng nhập	51
Hình 47: Mô tả chức năng Đăng xuất	52
Hình 48: Mô tả chức năng Xem danh sách ca thi	53
Hình 49: Mô tả chức năng Thêm ca thi	54
Hình 50: Mô tả chức năng Sửa ca thi	56
Hình 51: Mô tả chức năng Xem danh sách diễm danh	57
Hình 52: Mô tả chức năng Sửa kết quả điểm danh	59
Hình 53: Mô tả chức năng Kiểm tra khung giờ điểm danh	61
Hình 54: Hình mô tả các trường hợp xảy ra khi Điểm danh	62
Hình 55: Hình mô tả chức năng Điểm danh	62
DANH MỤC BẨNG BIỂU	
Bảng 1: Quá trình triển khai	9
Bảng 2: Mô tả các tác nhân tham gia vào trong hệ thống	20
Bảng 3: Bảng mô tả cơ sở dữ liệu	26
Bảng 4: Bảng mô tả các thành phần trong hệ thống	27
Bảng 5: Bảng mô tả chức năng Đăng nhập	52
Bảng 6: Bảng mô tả chức năng Đăng xuất	53
Bảng 7: Bảng mô tả chức năng Xem danh sách ca thi	54
Bảng 8: Bảng mô tả chức năng Thêm ca thi	55
Bảng 9: Bảng mô tả chức năng Sửa thông tin ca thi	57
Bảng 10: Bảng mô tả chức năng Xem kết quả điểm danh	
Bảng 11: Bảng mô tả chức năng Sửa kết quả điểm danh	60
Bảng 12: Bảng mô tả chức năng Kiểm tra khung giờ điểm danh	62
Bảng 13: Bảng mô tả chức năng Điểm danh	64

DANH SÁCH CÁC TỪ VIẾT TẮT

LÒI CẨM ƠN

Lời đầu tiên, chúng em xin trận trọng cảm ơn và bày tỏ lòng biết ơn sâu sắc nhất tới thầy Phạm Minh Tuấn – Giảng viên khoa công nghệ thông tin Trường Đại học Bách Khoa Đà Nẵng đã hướng dẫn đồ án PBL6 một cách nhiệt tình và chu đáo. Cảm ơn thầy đã tổ chức các buổi thông đồ án để báo cáo tiến độ hàng tuần, theo sát tiến độ báo cáo và không bị trễ han.

Cuối cùng, chúng tôi xin dành lời cảm ơn chân thành tới bạn bè đã động viên, khuyến khích và tạo điều kiện cho chúng tôi hoàn thành tốt đề tài của mình.

Xin chân thành cảm ơn!

Đà Nẵng, tháng 12 năm 2021

Nhóm sinh viên

Đỗ Văn Trình

Phạm Trần Nhân Nghĩa

Châu Bảo Nhật Vy

CHƯƠNG 1: MỞ ĐẦU

1.1 Đặt vấn đề

Thế giới đang đương đầu với đại dịch COVID-19. Đến nay, nó đã gây ra rất nhiều khó khăn và hệ luy cho đời sống của con người. Từ kinh tế, xã hội, y tế,... tất cả các lĩnh vực ngành nghề đều gặp nhiều trắc trở. Và "Giáo Dục" là một trong những lĩnh vực quan trọng nhất cần được giải quyết nhanh chóng và mang tính hiệu quả cao.

Từ vấn đề đó, việc tổ chức học và thi online đã trở nên phổ biến hơn. Nhưng bên cạnh những điểm tốt đó, chúng ta cũng gặp phải những mặt hạn chế trong việc học và thi cử. Đặc biệt là việc điểm danh thủ công của các ca thi luôn tốn rất nhiều thời gian. Từ đó, ý tưởng của đồ án nhóm chúng em mong muốn làm nên một trang web điểm danh online trong thi cử để hỗ trợ các giảng viên trong việc điểm danh sinh viên. Cụ thể là nhận diện so khớp khuôn mặt và nhận điện ảnh thẻ sinh viên để tự động điểm danh.

Sử dụng thuật toán nhận dạng ảnh thẻ, nhận dạng gương mặt con người và triển khai hệ thống trên website để điểm danh trực tuyến. Ý tưởng đồ án đem lại những tiện ích trong việc điểm danh, rút ngắn thời gian cho việc điểm danh, từ đó giúp cho việc tổ chức thi trở nên hiệu quả hơn.

1.2 Mục đích

- Xây dựng một hệ thống Website cho phép giảng viên quản lý các ca thi và cho phép sinh viên vào điểm danh cho ca thi tương ứng
- Các kết quả điểm danh sẽ được lưu trữ lại để hiển thị cho giảng viên

1.3 Quá trình triển khai

Tuần	Nhiệm vụ	Phụ Trách	Mức độ hoàn thành
6-7	Lên ý tưởng + Tìm hiểu thông tin và đọc tài liệu	Cả nhóm	Hoàn thành
8	Thiết kế CSDL, tìm hiểu thuật toán học máy và thu thập dữ liệu	Cả nhóm	Hoàn thành
9-11	Thiết kế giao diện và triển khai giao diện Website	Vy	Hoàn thành
9-12	Triển khai AI và Backend Website	Trình	Hoành thành
10-12	Cấu hình máy chủ	Nghĩa	Hoàn thành
12-14	Tích hợp Backend với Website	Trình, Vy	Hoàn thành
15	Thực hiện kiểm thử	Nghĩa	Hoàn thành
16	Sửa lỗi và hoàn thiện	Cả nhóm	Hoàn thành
17	Viết slide, báo cáo	Cả nhóm	Hoàn thành

Bảng 1: Quá trình triển khai

1.4 Phạm vi

- Loại ứng dụng: Ứng dụng Web
- Phạm vi sử dụng : Việt Nam (Trường Đại học Bách Khoa Đại học Đà Nẵng)
- Giao diện : Tiếng Anh, tiếng Việt
- Phục vụ đa truy cập (Khoảng 50 người dùng cùng lúc)
- Đối tượng người dùng:
 - Giảng viên
 - Sinh viên

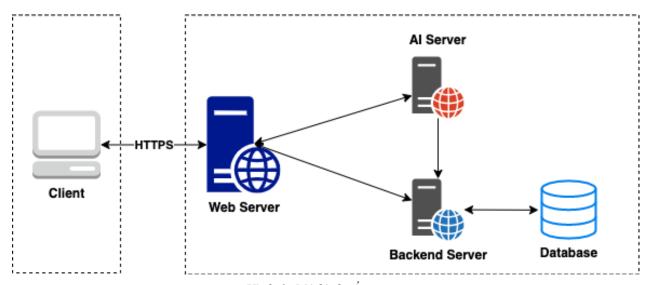
1.5 Các vấn đề cần giải quyết

Thông qua quá trình tìm hiểu và phân tích đề tài, nhóm đã tổng hợp và tự đưa ra các vấn đề cần phải giải quyết như sau:

- Phải xây dựng Website cho giảng viên và sinh viên, trong đó:
- Website dành cho giảng viên phải có các chức năng về quản lý ca thi và quản lý kết quả điểm danh
- Website dành cho sinh viên phải có chức năng điểm danh sử dụng thẻ sinh viên được tải lên và ảnh khuôn mặt sinh viên được chụp từ webcam
- Phải giải quyết được bài toán so khớp khuôn mặt và trích xuất thông tin với độ chính xác cao và tốc đô nhanh nhất có thể
- Lựa chọn các công nghệ, xây dựng các giải thuật phù hợp với cấu hình phần cứng để tối ưu hoá hiệu năng hệ thống
- Thu thập dữ liệu đủ lớn để huấn luyện mô hình học sâu
- Hệ thống phải có khả năng phục vụ đa truy cập (ít nhất 50 sinh viên điểm danh cùng lúc) và phải có khả năng mở rộng

1.6 Giải pháp tổng quan

Sơ đồ tổng quan hệ thống



Hình 1: Mô hình tổng quan

Hệ thống được xây dựng sẽ bao gồm các thành phần chính như sau:

- Client: Là giao diện của hệ thống để người dùng thao tác và thực thi các chức năng tương ứng
- Web Server: Là máy chủ Web có nhiệm vụ tiếp nhận các yêu cầu từ Client và sau đó chuyển tiếp yêu cầu đến máy chủ AI (AI Server) hoặc đến máy chủ truy xuất dữ liệu (Backend Server) tương ứng
- AI Server: Là máy chủ có nhiệm vụ xử lý các yêu cầu có sử dụng học máy (nhận diện thẻ sinh viên và so khớp khuôn mặt)
- Backend Server: Là máy chủ phục vụ các yêu cầu thao tác dữ liệu
- Database: Là nơi lưu trữ dữ liệu cho toàn bộ hệ thống

1.7 Công cụ phát triển

- Hệ điều hành máy chủ: Ubuntu 20.04

- **Ngôn ngữ:** JavaScript, Python

- **Framework :** VueJS, FastAPI, ExpressJS

- **Database**: MySQL, Redis

- Web Server: Nginx

- Quản lý Server: GoAccess

- **Kiểm thử:** Kiểm thử tự động, kiểm thử hiệu năng

- **Công cụ kiểm thử:** Selenium, k6.io

- Quản lý mã nguồn:

Việc quản lý mã nguồn được thực hiện thông qua Github

Link mã nguồn: https://github.com/trinhdvt/pbl6-attendance-support

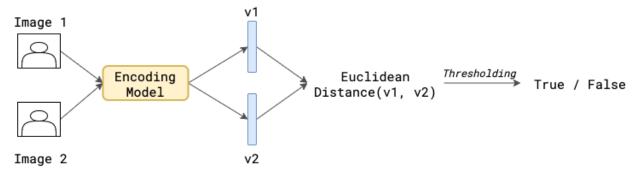
CHƯƠNG 2: TỔNG QUAN

2.1 Bài toán so khớp khuôn mặt

Nhận diện khuôn mặt là một bài toán phổ biến trong lĩnh vực học sâu, được áp dụng nhiều trong các hệ thống xác thực. So khớp khuôn mặt (hay còn gọi là xác thực khuôn mặt) là một trường hợp của bài toán.

Bài toán so khớp khuôn mặt trả lời cho câu hỏi "có phải 2 ảnh đầu vào là của cùng một người hay không ?". Kết quả của bài toán này là một trong hai giá trị Yes hoặc No. Bài toán này thường được ứng dụng trong các hệ thống bảo mật, ví dụ như mở khoá bằng khuôn mặt trên điện thoại.

Để giải quyết bài toán này, phương pháp thường được áp dụng đó là đo khoảng cách giữa 2 ảnh đầu vào và so sánh với một ngưỡng cố định. Khoảng cách thường được sử dụng là khoảng cách Euclid.



Hình 2: Mô hình cho bài toán so khớp khuôn mặt

Phương pháp này có ưu điểm là không bị phụ thuộc vào số lượng khuôn mặt đã biết do không cần thiết phải huấn luyện mô hình cho từng khuôn mặt. Tuy nhiên, phương pháp này đòi hỏi phải có một mô hình đủ tốt để mã hoá khuôn mặt thành một vector đặc trưng n chiều. Sau đó sử dụng khoảng cách để quyết định liệu 2 khuôn mặt có tương đồng hay không.

2.2 Bài toán nhận diện thông tin

Nhận diện thông tin là một bài toán phổ biến trong lĩnh vực học sâu và đặc biệt trong lĩnh vực thị giác máy tính. Nhận diện thông tin là bài toán chuyển đổi các thông tin có trong ảnh đầu vào thành các văn bản dạng text có thể thao tác và lưu trữ được. Ví dụ đối với bài toán nhận diện thông tin thẻ sinh viên, sẽ có đầu vào là thẻ sinh viên và đầu ra là các thông tin tướng ứng của sinh viên đó.



Hình 3: Mô hình cho bài toán nhận diện thông tin

Việc giải quyết bài toán này thông thường phải trải qua 3 bước như sau:

- Tiền xử lý: Hình ảnh đầu vào cho bài toán này thường có nhiều hình dạng khác nhau. Những hình ảnh có thể bị nghiêng, chứa nhiễu, ... Để làm cho kết quả nhận diện trở nên chính xác hơn, ta cần phải thực hiện một vài xử lý trước trên ảnh đầu vào.
- Phát hiện thông tin: Trước khi thực hiện việc chuyển từ hình ảnh sang dạng text, chúng ta cần phải phát hiện và cô lập các vùng thông tin trên ảnh đầu vào. Một số thuật toán có thể được dùng ở bước này như là Faster RCNN, YOLO, ...
- Nhận diện ký tự: Đây là bước cuối cùng trong mô hình giải quyết bài toán nhận diện thông tin. Có nhiệm vụ nhận dạng nội dung dạng text trong ảnh đầu vào.

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

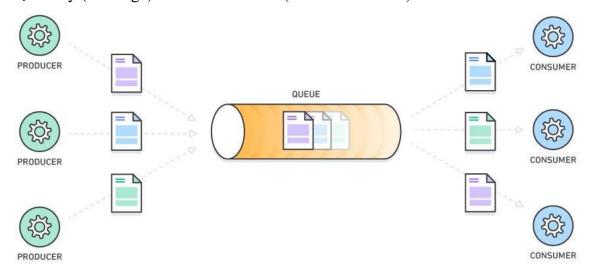
3.1 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao hướng đối tượng, hiện đang được sử dụng nhiều trong ứng dụng thực tế, đặc biệt là trong lĩnh vực trí tuệ nhân tạo và có một cộng đồng lớn mạnh.

Một số thư viện Python phổ biến trong lĩnh trí tuệ nhân tạo có thể kể đến là OpenCV, NumPy, Tensorflow, PyTorch, ... Ngoài ra Python còn có khả năng hoạt động trên nhiều hệ điều hành khác nhau như Window, Linux và MacOS, điều này giúp lập trình viên có thể viết và triển khai các ứng dụng Python một cách thuận tiện.

3.2 Mô hình Message Queue

Message Queue là kiến trúc sử dụng một hàng đợi (Queue) chứa các thông tin cần được xử lý (Message) theo cơ chế FIFO (First In First Out).



Hình 4: Mô hình Message Queue

Mô hình Message Queue bao gồm các thành phần chính sau:

- Message: Là các thông tin, dữ liệu được gửi tới và cần được xử lý (có thể là dạng text, JSON, ...).
- Producer: Là thành phần tiếp nhận các Message và đẩy vào hàng đợi.
- Queue: Là một hàng đợi hoạt động theo cơ chế FIFO (First In First Out) chứa các
 Message cần được xử lý và cho phép Producer và Consumer giao tiếp với nhau.
- Consumer: Là thành phần tiếp nhận các Message từ hàng đợi và xử lý chúng.

Uu điểm của mô hình:

- Đảm bảo tính ổn định, khả năng phục hồi: Việc đẩy các yêu cầu vào hàng đợi và xử lý tuần tự thay vì xử lý và trả kết quả trực tiếp giúp cho máy chủ vẫn có thể hoạt động ổn định trong trường hợp có nhiều yêu cầu được gửi đến. Ngoài ra, do các Message được lưu trữ trong hàng đợi, nên dữ liệu sẽ không bị mất đi và vẫn được tiếp tục xử lý sau khi khởi động lại hệ thống.
- Phân tách hệ thống: Khi sử dụng mô hình này, hệ thống sẽ được phân tách thành các thành phần nhỏ hơn và mỗi thành phần chỉ xử lý 1 chức năng nhất định, giúp cho dễ dàng hơn trong việc phát triển và kiểm thử.
- Dễ dàng mở rộng hệ thống: Trong các trường hợp cần xử lý nhiều yêu cầu, ta có thể tăng số lượng Consumer để xử lý được nhiều Message hơn. Ngược lại trong trường hợp không cần thiết, ta có thể giảm số lượng Consumer.

Nhược điểm của mô hình:

- Thời gian phản hồi: Việc xử lý các yêu cầu một cách tuần tự khiến cho thời gian phản hồi ngoài phụ thuộc vào tốc độ xử lý còn phụ thuộc vào số lượng Consumer. Trong trường hợp có nhiều yêu cầu sẽ khiến cho các yêu cầu tới sau có thời gian phản hồi chậm hơn.
- Tăng kích thước hàng đợi: Việc xử lý chậm có thể khiến hàng đợi phải chứa nhiều Message và có thể vượt quá giới hạn tài nguyên.
- Việc triển khai phức tạp hơn và cần có thêm các giải pháp để quản lý hàng đợi.

3.3 Cơ sở dữ liệu

3.3.1 MySQL

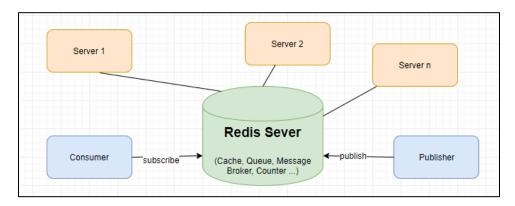
MySQL là hệ cơ sở dữ liệu quan hệ mã nguồn mở, hoạt động theo mô hình clientserver cho phép người dùng có thể thực hiện các thao tác liên quan đến cơ sở dữ liệu.

MySQL có ưu điểm là dễ sử dụng, dễ cài đặt, hỗ trợ nhiều ngôn ngữ lập trình và hoạt động trên nhiều hệ điều hành. Thích hợp trong việc triển khai các dự án nhỏ.

3.3.2 Redis

Redis là cơ sở dữ liệu lưu trữ dữ liệu với dạng key-value với nhiều tính năng được sử dụng rộng rãi. Nó có thể hỗ trợ nhiều kiểu dữ liệu như: string, hashe, list, set.

Khác với các hệ quản trị cơ sở dữ liệu thông thường, Redis lưu trữ dữ liệu trên RAM thay vì trên ổ cứng. Chính vì thế Redis có tốc độ truy xuất dữ liệu rất nhanh (chưa đến một mili giây). Tuy nhiên vì dữ liệu được lưu trên RAM nên sẽ mất đi khi tắt máy tính.



Hình 5: Cơ sở dữ liêu Redis

Các ứng dụng chính của Redis:

- Caching: Sử dụng làm bộ nhớ đệm. Chính tốc độ đọc ghi nhanh mà Redis có thể làm bộ nhớ đệm, nơi chia sẻ dữ liệu giữa các ứng dụng hoặc làm database tạm thời để giảm tải cho hệ thống.
- Message Queue: Redis cho phép lưu trữ theo list và cung cấp rất nhiều thao tác với các phần tử trong list, vì vậy nó còn được sử dụng như một hàng đợi trong mô hình Message Queue

3.4 Website

3.4.1 Web Server Nginx

Nginx là một máy chủ Web mã nguồn mở nổi tiếng ra đời vào năm 2004. Nginx được biết đến với các ưu điểm như hiệu suất cao, ổn định, cấu hình đơn giản và sử dụng ít tài nguyên.

Khác với Apache Web Server – máy chủ Web sử dụng kiến trúc đa luồng và đa tiến trình, Nginx sử dụng kiến trúc bất đồng bộ và hướng sự kiện thay cho việc tạo các luồng tướng ứng với mỗi kết nối để xử lý. Chính nhờ vào kiến trúc này mà Nginx có khả năng chịu tải cao hơn, có thể xử lý hàng ngàn kết nối cùng lúc với mức tiêu hao bộ nhớ thấp hơn so với Apache Web Server.

Ngoài chức năng làm máy chủ Web, Nginx còn có thể được sử dụng để làm Reverse Proxy cho các máy chủ Backend, hoặc làm máy chủ Cache, cân bằng tải, ...

3.4.2 GoAccess

GoAccess là một ứng dụng phân tích web nguồn mở cho các hệ điều hành Unix. Nó có cả giao diện dòng lệnh và giao diện web. GoAccess cung cấp các phân tích thời gian thực bằng cách liên tục theo dõi nhật ký máy chủ web.

3.4.3 NodeJS

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome, cho phép thực thi mã JavaScript ở phía máy chủ. Các thành phần chính của NodeJS được xây dựng dựa trên C++ nên cho tốc độ xử lý và hiệu năng khá cao. NodeJS có thể hoạt động trên nhiều hệ điều hành khác nhau như Window, Linux và MacOS.

Tương đồng với Nginx, NodeJS cũng được xây dựng theo kiến trúc bất đồng bộ hướng sự kiện. Trong các ngôn ngữ như Java và PHP, mỗi kết nối tạo ra một luồng mới và mỗi luồng mới có thể yêu cầu 2 MB bộ nhớ. Ví dụ, nếu trên hệ thống có 8 GB RAM, số lượng kết nối đồng thời tối đa theo lý thuyết là 4.000 kết nối. Trong khi đó, nhờ vào kiến trúc bất đồng bộ, NodeJS chỉ cần sử dụng một luồng duy nhất (single-thread) mà vẫn có thể xử lý hàng ngàn kết nối đồng thời. Chính vì ưu điểm này mà NodeJS rất thích hợp cho việc phát triển các API phục vụ truy xuất dữ liệu.

3.4.4 VueJS

VueJS là một JavaScript framework mã nguồn mở phổ biến, được sử dụng để xây dựng giao diện cho Website. VueJS cũng sở hữu một hệ sinh thái lớn, nên nó được cung cấp nhiều các thư viện khác hỗ trợ phát triển nhanh Website.

3.4.5 Giao thức HTTPS

HTTPS (Hypertext Transfer Protocol Secure) là giao thức truyền tải siêu văn bản an toàn. HTTPS có cách thức hoạt động tương tự như HTTP, tuy nhiên được bổ sung thêm chứng chỉ SSL (Secure Socket Layer) hoặc TLS (Transport Layer Secure) nhằm mã hoá các thông tin giao tiếp giữa Client – Server để tăng tính bảo mật.

Cả SSL và TLS đều sử dụng hệ thống PKI (Public Key Infrastructure) không đối xứng. Hệ thống này sử dụng hai "khóa" để mã hóa thông tin liên lạc, "khóa công khai"

(public key) và "khóa riêng" (private key). Các tiêu chuẩn này đảm bảo các nội dung sẽ được mã hóa trước khi truyền đi, và giải mã khi nhận. Điều này khiến kẻ tấn công dù có lấy được thông tin cũng không thể "hiểu" được thông tin đó.

3.5 Kiểm thử phần mềm

3.5.1 Tổng quan

Kiểm thử phần mềm là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, hoạt động đúng và đầy đủ các yêu cầu đã đặt ra.

Các giai đoạn kiểm thử phần mềm:

- Kiểm thử đơn vị (Unit Test): là hoạt động kiểm thử thực hiện trên các thành phần nhỏ nhất của hệ thống như hàm, thủ tục, ... Hoạt động này thường được tiến hàng bởi lập trình viên.
- Kiểm thử tích hợp (Integration Test): là hoạt động kiểm thử được thực hiện sau khi đã tích hợp các module lại với nhau nhằm kiểm tra sự giao tiếp giữa các module.
- Kiểm thử hệ thống (System Test): là hoạt động kiểm thử được tiến hành trên một hệ thống đã được hoàn chỉnh. Nhằm mục đích kiểm tra chức năng, giao diện ... của hệ thống so với yêu cầu ban đầu.

3.5.2 Các loại kiểm thử

- Kiểm thử thủ công: là việc tiến hành kiểm thử hoàn toàn bằng tay, từ việc viết ca kiểm thử đến việc thực thi, so sánh kết quả và tổng hợp báo cáo.
- Kiểm thử tự động: khác với kiểm thử thủ công, kiểm thử tự động sử dụng các công cụ đặc biệt để thực hiện các bài kiểm tra, so sánh với kết quả thực tế với kết quả mong muốn và tạo báo cáo kiểm thử chi tiết. Kiểm thử tự động được sử dụng để giảm chi phí trong các trường hợp ca kiểm thử được lặp đi lặp lại.
- Kiểm thử hiệu năng: là một loại kiểm thử phần mềm không nhằm mục đích tìm lỗi của các chức năng, mà nhằm xác định băng thông, khả năng xử lý, khả năng mở rộng hay nói chung là hiệu năng của hệ thống dưới khối lượng truy cập, khối lượng công việc xác định. Các dạng kiểm thử hiệu năng thường gặp là Load Testing, Stress Testing, Smoke Testing, ...

3.5.3 Giới thiệu Selenium

Là một bộ công cụ kiểm thử tự động mã nguồn mở phổ biến, được sử dụng trong việc kiểm thử tự động các ứng dụng web. Selenium có thể hoạt động trên nhiều trình duyệt thuộc các nền tảng khác nhau như Window, Linux, MacOS, ... Selenium có ưu điểm là quá trình cài đặt và sử dụng dễ dàng, không yêu cầu người sử dụng phải có kỹ năng lập trình.

Các thành phần chính của Selenium:

- Selenium IDE: là công cụ tích hợp trên trình duyệt, hỗ trợ ghi hình các thao tác để tạo kịch bản kiểm thử và thực thi trên các trình duyệt khác.
- Selenium RC: Đối với các kịch bản thực tiễn, cần phải thực hiện các công việc kiểm tra phức tạp với các câu lệnh, Selenium RC hỗ trợ tối đa các công việc này
- Selenium Grid: Hỗ trợ thực hiện kiểm thử trên các trình duyệt song song mà không cần chỉnh sửa kịch bản.
- Selenium WebDriver: Là bộ thư viện API nhằm giúp xây dựng ca kiểm thử trên bằng ngôn ngữ lập trình

3.5.4 Công cụ kiểm thử k6

- k6 là một công cụ mã nguồn mở được sử dụng để kiểm tra hiệu năng hệ thống. Cũng như các công cụ khác (Jmeter, Artillery, ...), k6 thực hiện việc kiểm thử một cách tự động theo các ngữ cảnh được định nghĩa trong tệp cấu hình.
- Ưu điểm lớn nhất của k6 đó là ngoài giao diện dòng lệnh, k6 cho phép thực hiện kiểm thử hiệu năng thông qua website k6.io với giao diện đồ hoạ, xuất và vẽ các biểu đồ thể hiện kết quả một cách trực quan.

CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1 Phân tích chức năng

4.1.1 Tác nhân

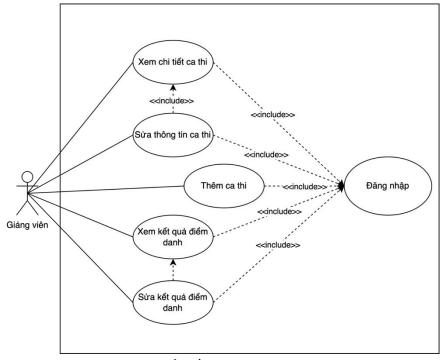
Bảng dưới đây sẽ mô tả tất cả các tác nhân tham gia vào hệ thống. Mỗi tác nhân sẽ có một vai trò khác nhau trong hệ thống.

Tác nhân	Mô tả
Giảng viên (Quản trị viên)	 Có quyền đăng nhập, đăng xuất Có quyền quản lý các ca thi trong hệ thống (thêm, xem, sửa) Có quyền quản lý các kết quả điểm danh của sinh viên (xem, sửa)
Sinh viên	 Có quyền điểm danh tương ứng với ca thi đã được giảng viên tạo (mã ca thi sẽ được giảng viên cung cấp)

Bảng 2: Mô tả các tác nhân tham gia vào trong hệ thống

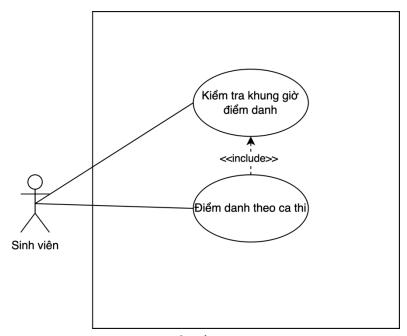
4.1.2 Biểu đồ ca sử dụng

4.1.2.1 Biểu đồ ca sử dụng cho Giảng viên



Hình 6: Biểu đồ ca sử dụng của Giảng viên

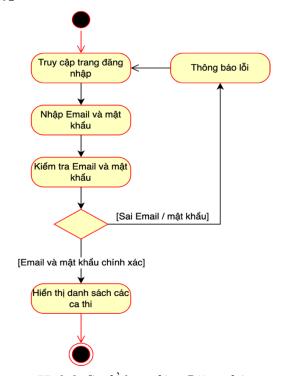
4.1.2.2 Biểu đồ ca sử dụng cho Sinh viên



<u>Hì</u>nh 7: Biểu đồ ca sử dụng của Sinh viên

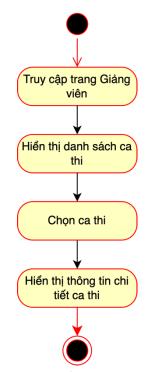
4.1.3 Biểu đồ hoạt động

4.1.3.1 Chức năng đăng nhập

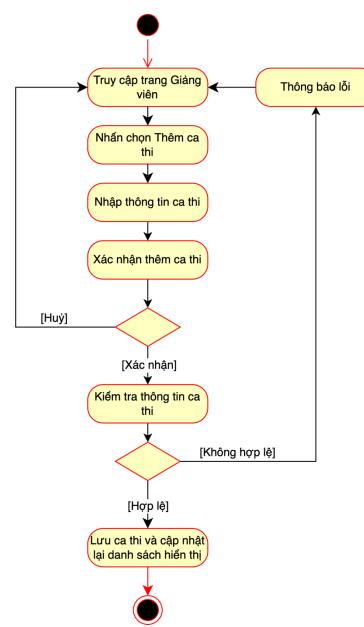


Hình 8: Sơ đồ hoạt động Đăng nhập

4.1.3.2 Chức năng Xem ca thi và Thêm ca thi

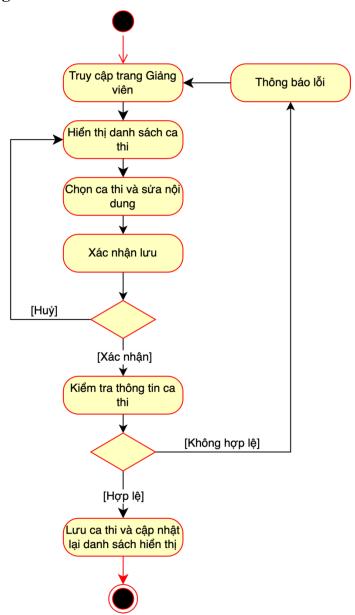


Hình 9: Sơ đồ hoạt động Xem ca thi



Hình 10: Sơ đồ hoạt động Thêm ca thi

4.1.3.3 Chức năng Sửa ca thi

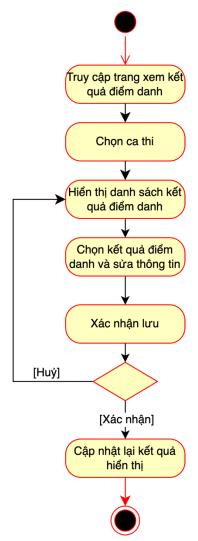


Hình 11: Sơ đồ hoạt động Sửa ca thi

4.1.3.4 Chức năng Xem và Sửa kết quả điểm danh

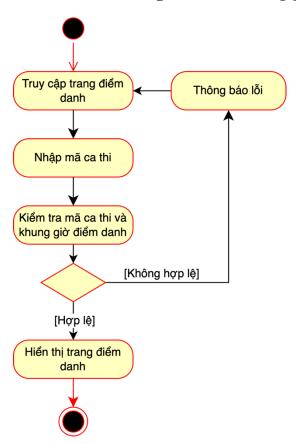


Hình 13: Sơ đồ hoạt động Xem kết quả điểm danh

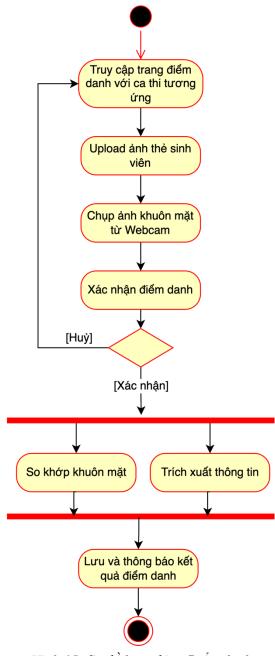


Hình 12: Sơ đồ hoạt động Sửa kết quả điểm danh

4.1.3.6 Chức năng Kiểm tra khung giờ điểm danh và Điểm danh

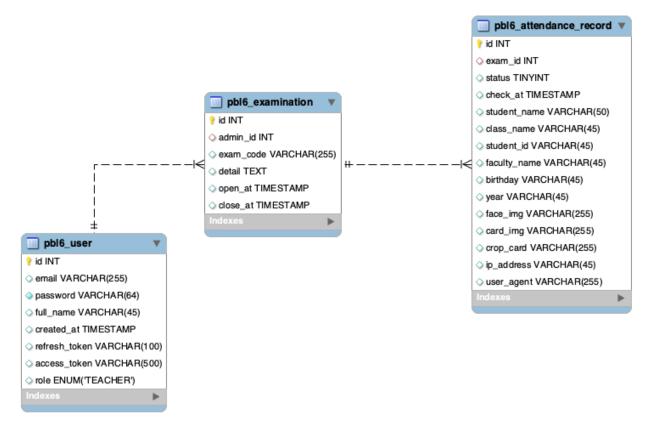


Hình 14: Sơ đồ hoạt động kiểm tra khung giờ điểm danh



Hình 15: Sơ đồ hoạt động Điểm danh

4.2 Thiết kế cơ sở dữ liệu



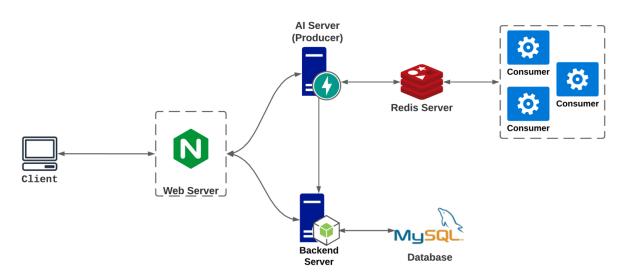
Hình 16: Mô tả cơ sở dữ liệu

Bång	Mô tả
PBL6_User	Lưu các thông tin của giảng viên, phục vụ cho việc xác
	thực như: Email đăng nhập, Mật khẩu đã mã hoá,
PBL6_Examination	Lưu thông tin các ca thi mà giảng viên đã tạo như: Mã ca
	thi, thời gian bắt đầu, thời gian kết thúc,
PBL6_Attendance_Record	Lưu các kết quả điểm danh của Sinh viên tướng ứng với
	mỗi ca thi, bao gồm: thông tin trích xuất từ thẻ sinh viên,
	kết quả so khớp khuôn mặt và các thông tin khác

Bảng 3: Bảng mô tả cơ sở dữ liệu

4.3 Thiết kế hệ thống

4.3.1 Các thành phần trong hệ thống



Hình 17: Kiến trúc hệ thống

Thành phần	Nhiệm vụ	Công nghệ
1. Client	 Giao diện của hệ thống để người dùng thao tác và thực thi các chức năng tương ứng 	VueJS
2. Web Server	 Đóng vai trò là một máy chủ Web phục vụ các yêu cầu tĩnh (file html, file ảnh,) 	Nginx
	 Và là một Reverse Proxy phân phối yêu cầu đến cho các máy chủ phía sau. 	
3. AI Server	 Đóng vai trò là một máy chủ Web tiếp nhận các 	Python
(Producer)	yêu cầu cần xử lý học máy	
	 Producer trong mô hình Message Queue 	
4. Consumer	 Là tiến trình sử dụng các thuật toán học máy để xử lý yêu cầu từ Producer 	Python
5. Redis Server	 Làm Message Queue kết nối Producer và 	Redis
	Consumer.	
	 Lưu trữ tạm thời kết quả đã xử lý của Consumer 	
6. Backend	 Kết nối với cơ sở dữ liệu và phục vụ các yêu cầu 	NodeJS
Server	truy xuất dữ liệu.	
7. Database	- Lưu trữ dữ liệu cho toàn bộ hệ thống	MySQL

Bảng 4: Bảng mô tả các thành phần trong hệ thống

4.3.2 Quy trình xử lý

- Từ Client gửi yêu cầu đến Web Server
- Sau đó Web Server sẽ dựa vào loại yêu cầu để điều hướng đến một trong 2 máy chủ phía sau:
- Nếu là yêu cầu truy xuất dữ liệu, thì sẽ được chuyển đến Backend Server. Lúc này
 Backend Server sẽ lấy dữ liệu từ Database và trả về Client
- Nếu là yêu cầu cần xử lý học máy (nhận dạng thẻ sinh viên và so khớp khuôn mặt) thì sẽ được chuyển đến AI Server. AI Server sẽ đẩy dữ liệu vào hàng đợi cho Consumer xử lý và trả về một mã định danh ID cho Client để lấy kết quả. Sau khi Consumer xử lý xong, kết quả sẽ được trả về Client, đồng thời sẽ được gửi qua Backend Server để lưu trữ vào cơ sở dữ liệu.

CHƯƠNG 5: TRIỂN KHAI HỆ THỐNG

5.1 Bài toán nhận diện thông tin thẻ sinh viên

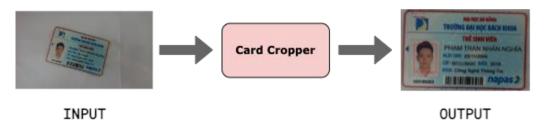
Để giải quyết bài toán này, nhóm sẽ chia việc xử lý thành 3 khối chính: **Card Cropper**, **Card Detector**, **Card Reader**. Mỗi khối sẽ sử dụng một thuật toán riêng, quy trình huấn luyện và kiểm thử riêng. Vì vậy có thể dễ dàng thay đổi từng khối nhỏ để cải thiện hệ thống.



Hình 18: Mô hình giải quyết bài toán nhân diên thông tin thẻ sinh viên

5.1.1 Tách thể sinh viên từ ảnh đầu vào (Card Cropper)

Để xây dựng khối này, nhóm sẽ sử dụng các phép xử lý ảnh cơ bản từ thư viện OpenCV để có được tốc độ xử lý nhanh nhất mà vẫn có được độ chính xác ổn định thay vì phải sử dụng các mô hình Object Detection.



Hình 19: Mục tiêu của khối Tách thẻ sinh viên

Việc xử lý bên trong khối này sẽ bao gồm 3 bước:

Bước 1: Tiền xử lý ảnh

Đầu tiên, ảnh đầu vào ở dạng RGB sẽ được lọc nhiễu bằng bộ lọc hai chiều (Bilateral Filter) nhằm loại bỏ nhiễu nhưng vẫn giữ được các đặc trưng cạnh. Sau đó, ảnh được đưa về dạng ảnh xám và áp dụng thuật toán Canny để xác định các canh thẻ.

Bước 2: Xác định vị trí thẻ

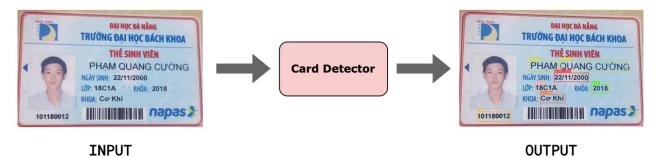
- Vì ảnh đầu vào yêu cầu chỉ chứa thẻ sinh viên, nên ta có thể xác định vị trí thẻ bằng cách lấy ra contour lớn nhất trong ảnh. Sau đó xấp xỉ contour thu được bằng một đa giác và chọn ra 4 điểm đại diện cho 4 góc thẻ

Bước 3: Cắt và Xoay về phương ngang

- Sau khi có được 4 góc thẻ, ta tính toán kích thước phần ảnh sẽ được cắt ra và sử dụng phép biến đổi phối cảnh để cắt và xoay về phương ngang.

5.1.2 Phát hiện thông tin sinh viên (Card Detector)

Khối này có nhiệm vụ phát hiện và tách các vùng chứa thông tin sinh viên từ ảnh đầu vào. Kết quả của khối này sẽ làm đầu vào cho khối tiếp theo (Card Reader)



Hình 20: Mục tiêu khối phát hiện thông tin sinh viên

Để xây dựng giải quyết bài toán phát hiện thông tin sinh viên, nhóm đã sử dụng mô hình YOLOv5^[5]. Là một trong số những mô hình mạng nơ-ron tích chập được dùng để giải quyết bài toán Object Detection mà ưu điểm nổi trội đó là tốc độ xử lý nhanh hơn nhiều so với các mô hình khác trên các máy tính cấu hình thấp. Mặc dù YOLOv5 chưa được công bố chính thức dưới dạng một bài báo khoa học nhưng qua tham khảo đánh giá từ cộng đồng và quá trình thử nghiệm, nhóm nhận thấy mô hình YOLOv5 có tốc độ thực thi và thời gian huấn luyện nhanh hơn rất nhiều so với một mô hình nổi bật khác là YOLOv4-tiny^[1] trong khi độ chính xác không chênh lệch quá nhiều.

- Huấn luyện mô hình YOLOv5
- Thu thập dữ liệu: Tập dữ liệu được dùng để huấn luyện có kích thước gồm 750 ảnh thẻ sinh viên, được thu thập trên thực tế và thông qua Internet, sau đó áp dụng các kĩ thuật tăng cường dữ liệu.



Hình 21: Dữ liệu huấn luyện

Gắn nhãn dữ liệu:Để gắn nhãn dữ liệu, nhóm sử dụng công cụ mã nguồn mở LabelImg. Mỗi ảnh đầu vào sẽ được vẽ các bounding box đánh nhãn riêng biệt cho các vùng thông tin cần được phát hiện. Sau khi kết thúc việc gắn nhãn sẽ thu được một file text có chứa thông tin lớp tương ứng với toạ độ của bounding box có dạng như sau:

3 0.1250918992209712 0.7538396328808299 0.17734294824050595 0.06102977333377114 2 0.421782566332172 0.5905922526542191 0.11654869552046954 0.06899207720884881 1 0.4451558901212746 0.5025456324339834 0.2066369458226958 0.07198868915726372 5 0.7523180403407133 0.5023975745970556 0.08128789638575448 0.0715473842014466

Hình 22: Mô tả file nhãn dữ liệu

- Phân chia dữ liêu:

Tập dữ liệu được chia thành 3 tập: huấn luyện / hiệu chỉnh / kiểm thử với tỉ lệ kích thước lần lượt là 80%:10%:10% tổng số ảnh

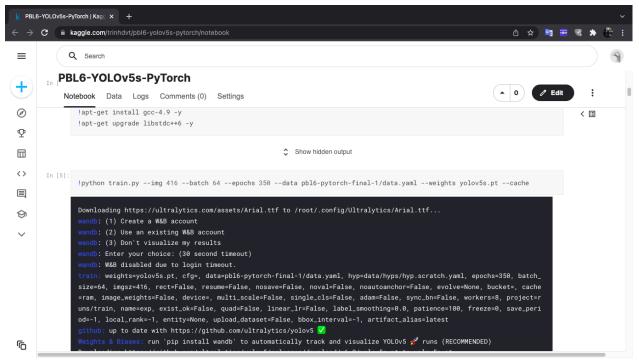
Link tập dữ liệu: https://bom.so/pbl6-train-ds

- Quá trình huấn luyện:

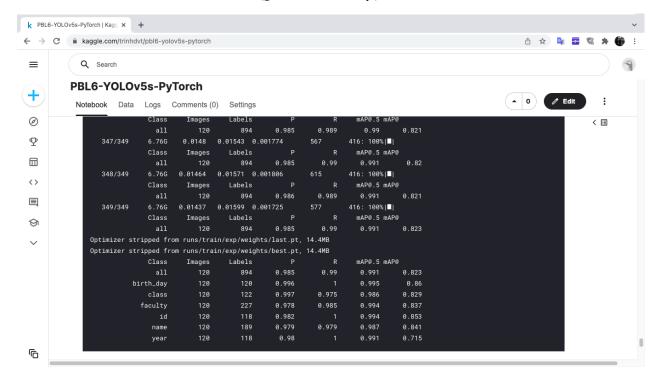
Để huấn luyện mô hình, nhóm sử dụng thư viện YOLOv $5^{[5]}$ – một thư viện mã nguồn mở để huấn luyện mô hình

Mô hình được huấn luyện trên Kaggle trong thời gian hơn 4 tiếng

Link notebook: <u>kaggle.com/trinhdvt/pbl6-yolov5s-pytorch</u>



Hình 24: Quá trình huấn luyện mô hình



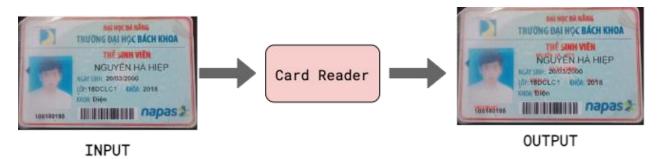
Hình 23: Kết quả quá trình huấn luyện

5.1.3 Đọc thông tin sinh viên (Card Reader)

Khối Reader có nhiệm vụ đọc thông tin sinh viên được phát hiện từ khối Detector sang dạng văn bản.

Ở bước này, một số phương pháp có thể được áp dụng như là: tách từng ký tự và thực hiện phân loại, sử dụng CRNN (Convolutional recurrent neural network), sử dụng Transformer, ...

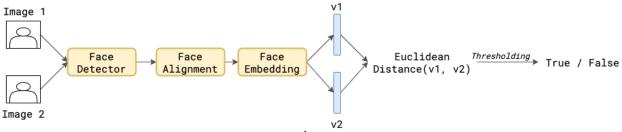
Trong các phương pháp kể trên, phương pháp sử dụng Transformer^{[3][6]} cho kết quả có độ chính xác cao hơn 2 phương pháp còn lại. Chính vì thế, ở khối Reader này, nhóm đã sử dụng mô hình Transformer từ thư viện VietOCR^[3], mô hình được huấn luyên sẵn từ thư viên này cho tốc đô xử lý nhanh và đô chính xác cao với tiếng Viêt.



Hình 25: Khối đọc thông tin sinh viên

5.2 Bài toán so khớp khuôn mặt

Để giải quyết bài toán so khớp khuôn mặt, nhóm sẽ chia thành nhiều bài toán nhỏ hơn và giải quyết lần lượt

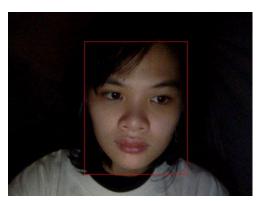


Hình 26: Mô hình giải quyết bài toán so khớp khuôn mặt

5.2.1 Phát hiện khuôn mặt (Face Detector)

Có nhiều phương pháp để phát hiện vị trí khuôn mặt trong bức ảnh, trong đó có 2 phương pháp phổ biến nhất có thể kể đến như Haar Casacade và CNN. Trong 2 phương pháp kể trên, phương pháp Haar Casacade cho tốc độ xử lý nhanh hơn, tuy nhiên độ chính xác thì không được tốt như phương pháp sử dụng CNN. Việc sử dụng CNN giúp đem lại độ chính xác cao hơn, có khả năng phát hiện được các khuôn mặt ở vị trí cường độ sáng và góc nghiêng mà phương pháp Haar Casacade không thể phát hiện được.

Trong hệ thống này, nhóm đã sử dụng mô hình CNN từ thư viện face_recognition^[2] để làm nhiệm vụ phát hiện khuôn mặt. Mô hình cho độ chính xác cao và tốc độ thực thi tương đối nhanh



Hình 27: Kết quả phát hiện khuôn mặt

5.2.2 Căn chỉnh khuôn mặt (Face Alignment)

Để đạt được hiệu quả tốt nhất trong bài toán so khớp khuôn mặt, ảnh khuôn mặt sau khi được cắt ra cần phải được căn chỉnh về hướng chính diện.

Bước 1: Phát hiện vi trí đặc trưng trên khuôn mặt

Bước 2: Tính toán khoảng cách giữa 2 mắt và xác định điểm thứ 3 để tạo thành tam giác vuông. Sau đó tính góc xoay cần thiết

Bước 3: Xoay khuôn mặt về hướng chính diên

5.2.3 Tạo vector đặc trưng (Face Embedding)

Ảnh khuôn mặt sau khi được căn chỉnh sẽ được đưa qua một mô hình trích xuất đặc trưng để thu được vector đặc trưng. Các phương pháp phổ biến có thể kể đến như là: PCA (Principal component analysis), phân tích đặc trưng Landmark, **CNN** (Convolutional neural network), ...

Trong các phương pháp kể trên, CNN hiện đang là một trong những phương pháp có độ hiểu quả cao hơn so với các phương pháp còn lại. Ở bước này nhóm đã sử dụng mang CNN (từ thư viên face recognition^[2]) làm nhiêm vụ trích xuất đặc trưng khuôn mặt.

Đầu ra của mạng CNN sẽ là một vector 128 chiều đại diện cho khuôn mặt đó. Sau đó, để xác đinh 2 khuôn mặt có thuộc cùng một người hay không, ta sẽ tính khoảng cách Euclid (l_2) giữa 2 vector đặc trưng và so sánh với một ngưỡng threshold (τ) = 0.55. Nếu khoảng cách nhỏ hơn ngưỡng thì 2 ảnh đầu vào được xem là cùng một người và ngược lai.

$$\begin{cases} l_2(v1,v2) \leq \tau \rightarrow gi \tilde{o} ng \ nhau \\ l_2(v1,v2) > \tau \rightarrow kh \acute{a} c \ nhau \end{cases}$$

Hình 28: Khoảng cách giữa 2 khuôn mặt tương đồng và không tương đồng

Euclid Distance =



Tương đồng





Euclid Distance = 0.69

Không tương đồng

Đỗ Văn Trình – Châu Bảo Nhât Vy – Pham Trần Nhân Nghĩa

CHƯƠNG 6: CẦU HÌNH MÁY CHỦ

6.1 Cấu hình tường lửa cho máy chủ Ubuntu Server

Để quản lý và cho phép các kết nối từ bên ngoài vào thông qua các cổng, việc đầu tiên là cấu hình tường lửa cho máy chủ.

Bước 1: Kích hoạt tường lửa từ Terminal bằng câu lệnh sau: sudo ufw enable

Bước 2: Mở các cổng dịch vụ web HTTP và HTTPS. Ngoài ra cổng 22 (SSH) cũng cần được mở để có thể truy cập từ xa.

- sudo ufw allow 22
- sudo ufw allow 80
- sudo ufw allow 443

Bước 3: Áp dụng các thay đổi bằng lệnh: sudo ufw reload

Bước 4: Kiểm tra lại bằng lệnh sudo ufw status verbose

Hình 29: Kết quả cấu hình tường lửa

6.2 Cài đặt Nginx

Bước 1: Từ Terminal chạy lần lượt các lệnh sau:

- sudo apt update && apt-get update
- sudo apt install nginx

Bước 2: Kiểm tra kết quả cài đặt bằng lệnh: nginx -v

```
root@pbl6-ubuntu-1-2:~# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

Hình 30: Kết quả cài đặt Nginx

6.3 Viết các file cấu hình dùng chung

Nginx cho phép chúng ta nhúng các file cấu hình khác nhau lại với nhau để giảm việc lặp lại các đoạn mã cấu hình.

Bước 1: Tạo thư mục chứa các file cấu hình chung: sudo mkdir -p /etc/nginx/utils

Bước 2: Tạo file cấu hình chung cho việc caching *general.conf* với nội dung như sau:

Hình 31: Nội dung file general.conf

Bước 3: Tạo file cấu hình chung cho việc bảo mật security.conf với nội dung như sau:

```
# security headers
add_header X-XSS-Protection
add_header X-Content-Type-Options
add_header Referrer-Policy
add_header Content-Security-Policy
add_header Permissions-Policy
add_header Strict-Transport-Security
# . files
location ~ /\.(?!well-known) {
    deny all;
}
"1; mode=block" always;
"nosniff" always;
"strict-origin-when-cross-origin" always;
"default-src 'self' http: https: data: blob: 'unsafe-inline'; frame-ancestors 'self';" always;
always;
always;
"interest-cohort=()" always;
always;
always;
"interest-cohort=()" always;
always;
always;
"interest-cohort=()" always;
always;
always;
always;
"offiles
location ~ /\.(?!well-known) {
    deny all;
}
```

Hình 32: Nội dung file security.conf

Bước 4: Tạo file cấu hình chung cho việc cấu hình Reverse Proxy *proxy.conf* với nội dung sau:

```
proxy_http_version
                                             $http_upgrade;
proxy_cache_bypass
# Proxy headers
proxy_set_header Upgrade
                                            $http_upgrade;
$connection_upgrade;
proxy_set_header Connection
proxy_set_header Host
proxy_set_header X-Real-IP
proxy_set_header X-Forwarded-For
                                             $remote_addr;
                                             $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Port $server_port;
# Proxy timeouts
proxy_connect_timeout
proxy_send_timeout
proxy_read_timeout
                                             60s;
```

Hình 33: Nội dung file proxy.conf

6.4 Cấu hình dịch vụ HTTP và Reverse Proxy

Ở đây ta sẽ cấu hình dịch vụ HTTP và Reverse Proxy trên Nginx với các điều kiện sau:

- Máy chủ AI Server có địa chỉ **127.0.0.1:8888**
- Máy chủ Backend Server có địa chỉ 127.0.0.1:9999
- Tên miền cho trang Giảng viên là admin.illusion.codes
- Tên miền cho trang Sinh viên là **sv.illusion.codes**
- Thư mục chứa các file tĩnh cho trang Giảng viên có đường dẫn là /var/www/pbl6/admin
- Thư mục chứa các file tĩnh cho trang Sinh viên có đường dẫn là /var/www/pbl6/user

Bước 1: Trong thư mục /etc/nginx/sites-available, tạo file pbl6.conf có nội dung sau:

```
# back end server definition

upstream ai_server {
    server 127.0.0.1:8888;
}

upstream backend_server {
    server 127.0.0.1:9999;
}

# student page config

server {
    server_name sv.illusion.codes;
    # ssl config
    listen [::]:443 ssl;
    listen 443 ssl;
```

```
include utils/security.conf;
    # service config
   root /var/www/pbl6/student;
   index index.html index.htm;
    location / {
      try_files $uri /index.html =404;
    # reverse proxy setting
    location = /api/exams/check {
     proxy_pass http://backend_server;
     include utils/proxy.conf;
   location /api {
     proxy_pass http://ai_server;
       include utils/proxy.conf;
    }
   include utils/general.conf;
}
server {
   if ($host = sv.illusion.codes) {
        return 301 https://$host$request uri;
    }
   server name sv.illusion.codes;
   listen 80;
   listen [::]:80;
}
# admin page config
server {
    server name admin.illusion.codes;
   listen [::]:443 ssl;
   listen 443 ssl;
```

```
include /etc/letsencrypt/options-ssl-nginx.conf;
    root /var/www/pbl6/admin;
    access log /var/log/nginx/pb16-admin-access.log;
    error log /var/log/nginx/pb16-admin-error.log;
    include utils/security.conf;
    location / {
        index index.html index.htm;
        try files $uri /index.html =404;
    }
    location /api {
        # reverse proxy setting
        proxy pass http://backend server;
        include utils/proxy.conf;
    location /images/ {
        root /home/pb16;
        access log off;
        autoindex on;
        expires 365d;
        add header Cache-Control public;
        add header Pragma public;
        add header Vary Accept-Encoding;
    }
}
server {
    if ($host = admin.illusion.codes) {
        return 301 https://$host$request_uri;
    }
    server name admin.illusion.codes;
   listen 80;
    listen [::]:80;
```

```
return 404;
```

Bước 2: Kích hoạt file cấu hình bằng các lệnh sau:

- sudo ln -s /etc/nginx/sites-available/pbl6.conf /etc/nginx/sites-enabled/
- sudo nginx -s reload

6.5 Cấu hình SSL

Trong phần này, ta sẽ cấu hình SSL cho trang web bằng cách dùng chứng chỉ SSL của Let' Encrypt thông qua *certbot*. Ưu điểm của cách làm này là đơn giản và không mất phí, tuy nhiên chứng chỉ từ Let's Encrypt chỉ có hạn sử dụng 90 ngày.

Buốc 1: Cài đặt Cerbot Let's Encrypt Client:

- sudo apt update && apt-get update
- sudo apt install certbot
- sudo apt-get install python3-certbot-nginx -y

Bước 2: Cài đặt SSL cho trang Giảng viên và Sinh viên

• certbot --nginx -d sv.illusion.codes -d admin.illusion.codes

Bước 3: Áp dụng thay đổi:

• sudo nginx -s reload

6.6 Cấu hình quản lý lưu lượng truy cập

Ở đây ta sẽ cấu hình việc quản lý truy cập với các điều kiện sau:

- Tên miền cho trang quản lý là status.illusion.codes
- Tài khoản và mật khẩu để đăng nhập vào trang quản lý là "admin"

Bước 1: Cài đặt GoAccess theo hướng dẫn tại địa chỉ: https://goaccess.io/download

Bước 2: Tạo thư mục chứa file tổng hợp để hiển thị lên website

• sudo mkdir -p /var/www/goaccess

Bước 3: Khởi chạy GoAccess để tạo file tổng hợp:

Bước 4: Cài đặt xác thực cho trang quản lý.

Khi truy cập vào trang quản lý, người dùng sẽ được yêu cầu phải đăng nhập. Ở đây ta cấu hình đăng nhập với tài khoản là "admin" và mật khẩu là "admin" với các lệnh sau:

- sudo sh -c "echo -n 'admin:' >> /etc/nginx/.htpasswd"
- sudo sh c "openssl passwd admin >> /etc/nginx/.htpasswd"

Bước 5: Tạo file cấu hình Nginx cho trang quản lý.

Trong thư mục /etc/nginx/sites-available tạo file status.conf với nội dung sau:

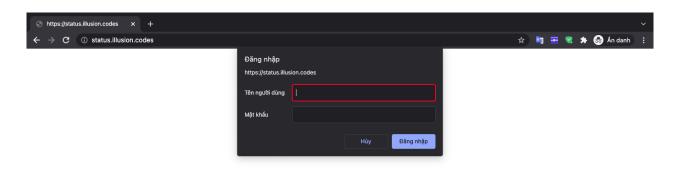
```
upstream gwsocket {
    server 127.0.0.1:7890;
}
server {
    server_name status.illusion.codes;
    listen 80;
    listen [::]:80;
    # redirect all http to https
    return 301 https://$host$request_uri;
}
server {
    server_name status.illusion.codes;
    # ssl config
```

```
listen [::]:443 ssl;
listen 443 ssl;
root /var/www/goaccess;
access log /var/log/nginx/goaccess-access.log;
error log /var/log/nginx/goaccess-error.log warn;
location / {
 auth basic "Restricted Access!";
    auth basic user file /etc/nginx/.htpasswd;
  try files $uri/report.html =404;
location /ws {
   proxy_http_version 1.1;
   proxy set header Upgrade $http upgrade;
   proxy set header Connection $connection upgrade;
   proxy pass http://gwsocket;
   proxy buffering off;
   proxy read timeout 7d;
```

Bước 6: Kích hoạt file cấu hình

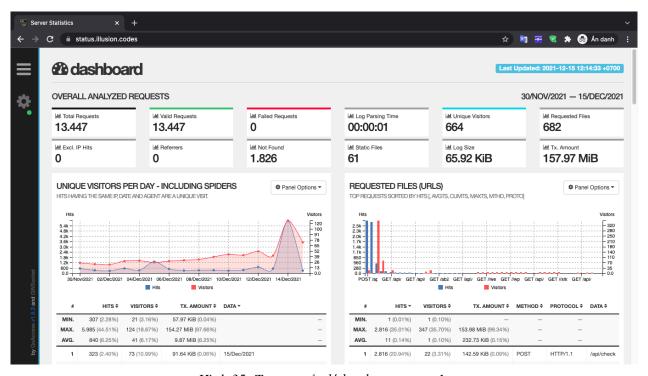
- sudo ln -s /etc/nginx/sites-available/status.conf /etc/nginx/sites-enabled/
- sudo nginx -s reload

Bước 7: Truy cập vào địa chỉ của trang cấu hình và tiến hành đăng nhập:



Hình 34: Giao diện đăng nhập trang quản lý lưu lượng truy cập

Sau khi đăng nhập với tài khoản và mật khẩu đã tạo, giao diện quản lý như sau:



Hình 35: Trang quản lý lưu lượng truy cập

CHƯƠNG 7: KIỂM THỬ HỆ THỐNG

7.1 Kiểm thử tự động các chức năng

7.1.1 Cài đặt Selenium IDE

- Bước 1: Truy cập cửa hàng Chrome bằng đường Link:
 https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=vi
- Bước 2: Tiến thành thêm tiện ích Selenium IDE vào Chrome. Selenium IDE sẽ được tự động cài đặt



Hình 36: Cài đặt tiên ích Selenium

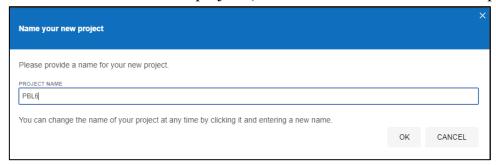
7.1.2 Sử dụng Selenium IDE để kiểm thử trang web

- **Bước 1**: Mở tiện ích Selenium bên góc phải của trang web



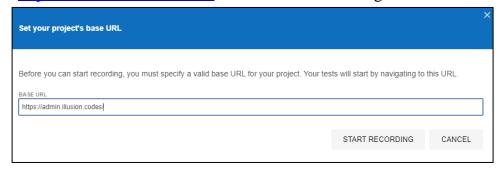
Hình 37: Mô tả tiện ích Selenium

- **Buốc 2**: Tạo record mới cho project (Chọn Record a new test in a new project)



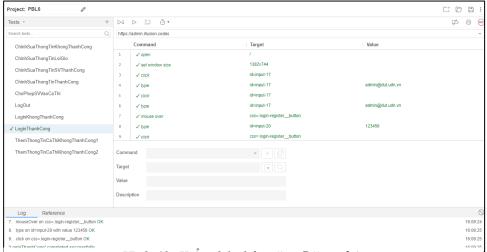
Hình 38: Tạo project trong Selenium

- **Bước 3**: Nhập địa chỉ trang web cần thực hiện các test Vd: https://admin.illusion.codes/ và bấm Start Recording



Hình 39: Chuẩn bị kiểm thử bằng Selenium

Bước 4: Tiến hành kiểm thử để Selenium tự động lưu record lại
Bấm vào dấu cộng để thêm các test, sau đó bấm nút record màu đỏ bên góc phải
Sau khi hoàn thành xong test -> chuột phải chọn Selenium IDE -> chọn Assert -> chọn test. Click vào nút đỏ để Tắt record



Hình 40: Kiểm thử chức năng Đăng nhập

- **Bước 5**: Xem kết quả tổng hợp



Hình 41: Các chức năng đã kiểm thử thành công

7.2 Kiểm thử hiệu năng

Việc kiểm thử được tiến hành với các điều kiên sau:

- Cấu hình máy chủ: CPU 2 Core, RAM 4G, SSD 50GB

7.2.1 Kịch bản kiểm thử

Bài test được diễn ra trong thời gian 5 phút với số lượng người dùng ảo truy cập cùng lúc vào website được mô tả như sau:

- Trong khoảng thời gian 1 phút đầu tiên: số lượng người dùng tăng từ 0 đến 50 cùng truy cập
- Trong 3 phút tiếp theo: 50 người dùng liên tục truy cập cùng lúc vào website
- Trong 1 phút còn lại: số lượng người dùng truy cập cùng lúc giảm dần từ 50 về 0

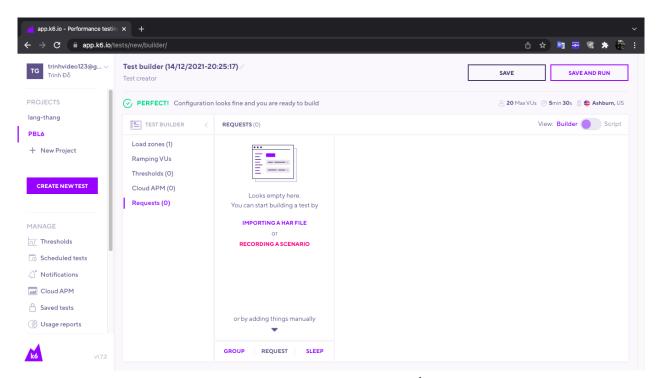
Trong bài test này, các người dùng ảo sẽ lần lượt gửi yêu cầu đến các địa chỉ sau đây:

- Đăng nhập: POST https://admin.illusion.codes/api/auth/login
- Xem danh sách ca thi: GET https://admin.illusion.codes/api/exams
- Điểm danh: POST https://sv.illusion.codes/api/check
- Lấy kết quả điểm danh: GET https://sv.illusion.codes/api/results

7.2.2 Tiến hành kiểm thử

Bước 1: Đăng nhập vào website k6.io

Bước 2: Chọn Create New Test > Start Building để tạo kịch bản kiểm thử

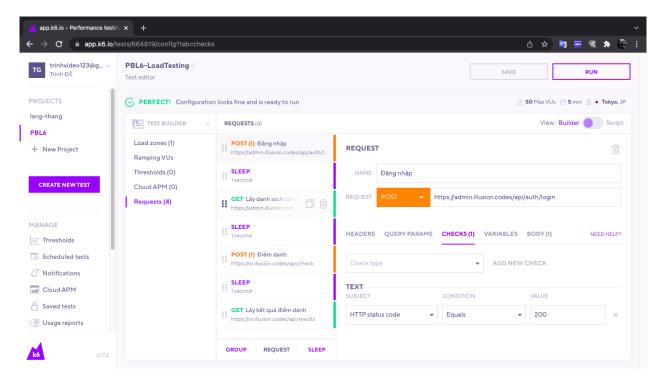


Hình 42: Giao diện tạo kịch bản kiểm thử

Các thành phần chính trong giao diện tạo kịch bản kiểm thử:

- Load Zones: là khu vực địa lý mà các yêu cầu sẽ được gửi đến máy chủ. Tuỳ thuộc
 vào vị trí đặt máy chủ mà ta có thể chọn phù hợp
- Ramping VUs: Thiết lặp số lượng người dùng ảo và thời gian thực hiện kiểm thử.
- Thresholds: Thiết lặp các điều kiện để đánh giá kết quả kiểm thử
- Requests: Danh sách các yêu cầu sẽ được gửi đi

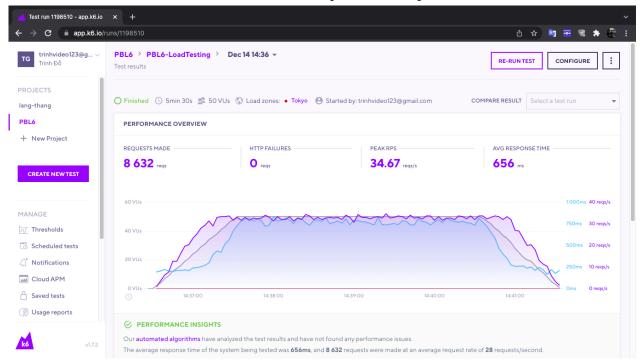
Bước 3: Lần lượt thiết lập các cấu hình cần thiết theo như kịch bản đã đề ra



Hình 43: Kịch bản kiểm thử hiệu năng đã tạo

Trong quá trình tạo các yêu cầu, ta có thể thêm Headers, hoặc các điều kiện đánh giá Checks nếu cần thiết.

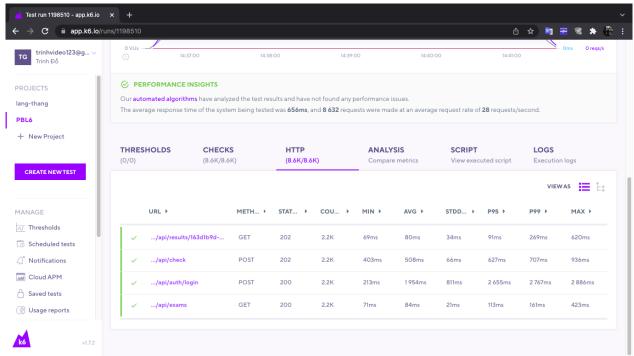
Bước 4: Bấm Run để thực thi kiểm thử và quan sát kết quả



Hình 44: Kết quả kiểm thử hiệu năng (1)

Dựa vào hình đầu tiên ta có thể thấy được kết quả tổng quan như sau:

- Số lượng yêu cầu đã gửi đi: 8632 yêu cầu
- Số lượng yêu cầu thất bại: 0 yêu cầu
- Số lượng yêu cầu tối đa tại một thời điểm: 35 yêu cầu
- Thời gian phản hồi trung bình: 656ms



Hình 45: Kết quả kiểm thử hiệu năng (2)

Hình trên sẽ cho ta thấy được kết quả chi tiết với mỗi loại yêu cầu, trong đó:

- Yêu cầu Đăng nhập có thời gian phản hồi lâu nhất là 2886ms, trung bình 1954ms
- Yêu cầu Điểm danh có thời gian phản hồi trung bình là 508ms, 99% tổng số yêu cầu có thời gian phản hồi nhỏ hơn 707ms
- Yêu cầu Xem danh sách ca thi và Lấy kết quả điểm danh có thời gian phản hồi trung bình là 80ms, thời gian phản hồi tối đa là khoảng 600ms
- Thông qua kết quả kiểm thử hiệu năng có thể thấy rằng hệ thống vẫn có thể hoạt động ổn định trong điều kiện có nhiều yêu cầu được gửi đến, thời gian phản hồi trong điều kiện đó không quá chậm. Tuy nhiên vì hệ thống chỉ sử dụng 3 Consumer để xử lý nên nhiều yêu cầu chưa được xử lý tồn tại trong hàng đợi làm tăng kích thước hàng đợi, khiến máy chủ gần đạt tới giới hạn phần cứng.

CHƯƠNG 8: ĐÁNH GIÁ KẾT QUẢ

8.1 Các chức năng đã triển khai:

8.1.1 Giảng viên

8.1.1.1 Đăng nhập



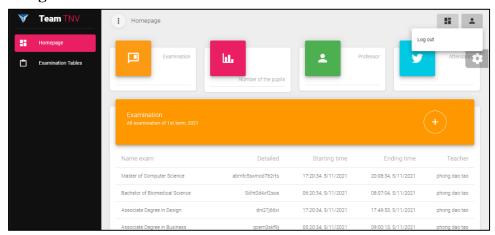
Hình 46: Mô tả chức năng Đăng nhập

Giao diện	Log in.			
Mô tả	Cho phép giảng viên truy cập vào trong hệ thống website.			
Truy cập	Giảng viên truy cập vào	trang https://admin.ill	usion.codes/.	
	Nội du	ng giao diện		
Thành phần	Kiểu	Dữ liệu	Mô tả	
Email	TextBox – String(100)	Email đăng nhập của giảng viên	Trường dành cho giảng viên điền email	
Password	TextBox – String(100)	Mật khẩu của tài khoản email của giảng viên	Trường dành cho giảng viên điền mật khẩu	
	Các hành động trong giao diện			
Hành động	Mô tả	Thành công	Lỗi	

Submit	Khi giảng viên nhập	Nếu đăng nhập	Đăng nhập không
	đầy đủ thông tin email	thành công, giảng	thành công sẽ hiện ra
	và password sau đó	viên sẽ được	thông báo "Invalid
	nhấn Submit	chuyển đến trang	email or password
		cho giảng viên	

Bảng 5: Bảng mô tả chức năng Đăng nhập

8.1.1.2 Đăng xuất



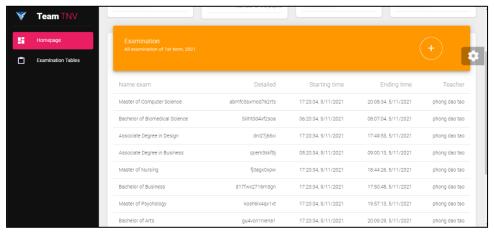
Hình 47: Mô tả chức năng Đăng xuất

Giao diện	Log out.			
Mô tả	Cho phép giảng viên đăng xuất khỏi hệ thống.			
Truy cập	Giảng viên đăng nhập tha	Giảng viên đăng nhập thành công vào trang web		
	https://admin.illusion.cog	des/		
	Nội dung	giao diện		
Thành phần	Kiểu	Dữ liệu	Mô tả	
TeamTNV	Label			
HomePage	Label			
HomePage	Button		Truy cập vào	
			HomePage	
Log out	Button		Dùng để đăng xuất ra khỏi hệ thống	
			_	

HomePage	Link		Truy cập vào HomePage
Examination Table	link		Truy cập vào Examination Table
	Các hành động	trong giao diện	
Hành động	Mô tả	Thành công	Lỗi
Log out	Khi giảng viên muốn đăng xuất ra ngoài khỏi hệ thống, giảng viên bấm vào biểu tượng Log out	Giảng viên đăng xuất ra ngoài khỏi hệ thống	
HomePage	Giảng viên click vào nút HomePage	Chuyển đến trang HomePage.	
Examination Table	Giảng viên click vào nút Examination Table	Chuyển đến trang xem các ca thi	

Bảng 6: Bảng mô tả chức năng Đăng xuất

8.1.1.3 Xem danh sách ca thi

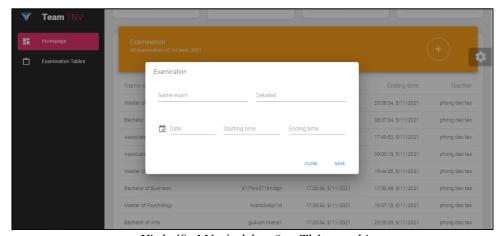


Hình 48: Mô tả chức năng Xem danh sách ca thi

Giao diện	Xem danh sách các ca thi		
Mô tả	Admin xem danh sách các ca thi có trong hệ thống		
Truy cập	Admin truy cập vào trang https://admin.illusion.codes		
	Nội dung g	giao diện	
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả
Examination Table	Data table	List các ca thi (bao gồm các trường: Name exam, Detailed, Starting Time, Ending Time, Teacher)	Data table hiển thị danh sách của tất cả các ca thi có trong hệ thống.
	Các hành động t	rong giao diện	
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
Click vào một ca thi	Xem cụ thể thông tin ca thi	Xuất hiện một dialog hiển thị tất cả các thông tin của ca thi đã click chọn xem	

Bảng 7: Bảng mô tả chức năng Xem danh sách ca thi

8.1.1.4 Thêm ca thi

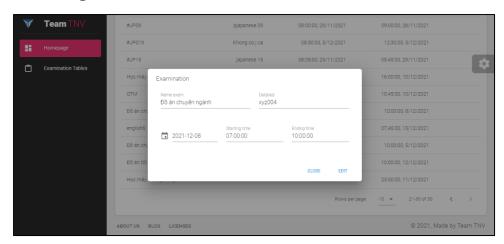


Hình 49: Mô tả chức năng Thêm ca thi

Giao diện	Thêm ca thi		
Mô tả	Admin thêm một ca thi, trong trang chủ [Homepage] bấm chọn button (+) ở trên header của Examination Table để tiến hành thêm một ca thi		
Truy cập	Admin truy cập vào trang https://admin.illusion.codes		
		Giao diện	
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả
Name exam	String	Tên của ca thi	Nhập thêm tên của ca thi đó
Detailed	String	Mô tả của ca thi	Nhập vào mô tả về ca thi
Date	Date	Ngày điểm danh của ca thi	Chọn ngày điểm danh bằng date picker
Starting time	String	Thời gian bắt đầu điểm danh	Nhập string có định dạng hh:mm:ss
Ending time	String	Thời gian kết thúc điểm danh	Nhập string có định dạng hh:mm:ss
	Các hành	động trong giao diện	
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
Save	Tạo mới ca thi	Dialog đóng lại và hệ thống thêm ca thi vào cơ sở dữ liệu, Examination table sẽ xuất hiện ca thi đó	Hiển thị lỗi bằng thông báo pop up
Cancel	Đóng Dialog	Dialog được đóng lại	

Bảng 8: Bảng mô tả chức năng Thêm ca thi

8.1.1.5 Sửa thông tin ca thi



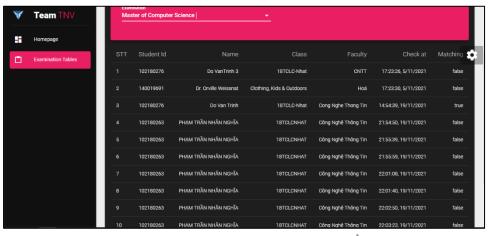
Hình 50: Mô tả chức năng Sửa ca thi

Giao diện	Chỉnh sửa ca thi			
Mô tả	Admin bấm chọn cụ thể một ca thi và chỉnh sửa ca thi đó			
Truy cập	Admin truy cập vào t	Admin truy cập vào trang https://admin.illusion.codes		
	Giao	diện		
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả	
Name exam	String	Tên của ca thi	Nhập thêm tên của ca thi đó	
Detailed	String	Mô tả của ca thi	Nhập vào mô tả về ca thi	
Date	Date	Ngày điểm danh của ca thi	Chọn ngày điểm danh bằng date picker	
Starting time	String	Thời gian bắt đầu điểm danh	Nhập string có định dạng hh:mm:ss	
Ending time	String	Thời gian kết thúc điểm danh	Nhập string có định dạng hh:mm:ss	

Các hành động trong giao diện			
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
Chỉnh sửa thông tin	Chỉnh sửa các thông tin muốn thay đổi của ca thi		
Save	Xem cụ thể thông tin ca thi	Dialog được đóng lại và thông tin của ca thi được cập nhật lên hệ thống	Hiển thị lỗi bằng thông báo pop up
Cancel	Đóng Dialog	Dialog được đóng lại	

Bảng 9: Bảng mô tả chức năng Sửa thông tin ca thi

8.1.1.6 Xem danh sách sinh viên đã điểm danh



Hình 51: Mô tả chức năng Xem danh sách diểm danh

Giao diện	Xem danh sách sinh viên đã điểm danh		
Mô tả	Admin xem danh sách các sinh viên trong một ca thi cụ thể		
Truy cập	Admin truy cập vào trang [Exam attendance] (https://admin.illusion.codes/tables/regular/)		
	Nội dung giao diện		

Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả
Examination	Select box	List các ca thi có trong hệ thống	Chọn một ca thi trong list danh sách của select box [Examination]
List Table	Data table	List danh sách các sinh viên của một ca thi (bao gồm các trường: STT, Student ID, Name, Class, faculty, Check at, Matching)	Data table hiển thị danh sách của tất cả các sinh viên đã điểm danh trong ca thi.
	Các hành động tr	rong giao diện	
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
[Examination] Select box	Click vào select box, danh sách tất cả các ca thi hiện ra và bấm chọn một ca thi cụ thể	Data table sẽ xuất hiện danh sách các sinh viên đã điểm danh của ca thi đó	
Click vào một sinh viên	Xem cụ thể thông tin sinh viên	Xuất hiện một dialog hiển thị tất cả các thông tin của sinh viên	

Bảng 10: Bảng mô tả chức năng Xem kết quả điểm danh

8.1.1.7 Chỉnh sửa thông tin điểm danh của sinh viên



Hình 52: Mô tả chức năng Sửa kết quả điểm danh

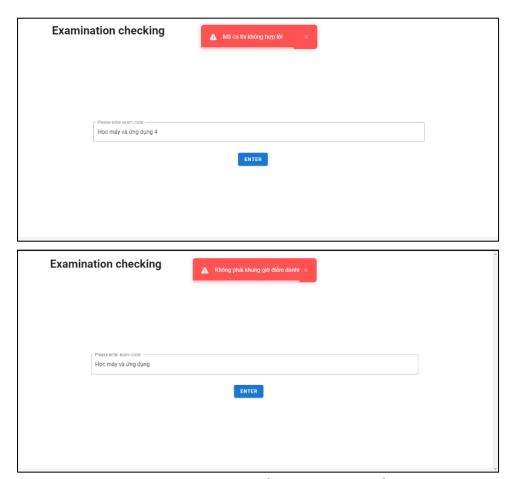
Giao diện	Chỉnh sửa thông tin đ	liểm danh của sinh viên	1
Mô tả	Admin bấm chọn cụ thể một sinh viên và chỉnh sửa thông tin của sinh viên đó, một Dialog sẽ hiện ra bao gồm tất cả các thông tin liên quan của sinh viên đó		
Truy cập		rang [Exam attendance n.codes/tables/regular/)	
	Nội dung	giao diện	
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả
Student Name	String	Tên của sinh viên	Nhập và chỉnh sửa những thông tin muốn thay đổi
Student ID	String	Mã số sinh viên	
Birthday	String	Ngày tháng năm sinh	
Year	String	Năm nhập học	
Class name	Date	Lớp sinh hoạt	

Matching	Select box (True/False)	Điểm danh hợp lệ	
Face Image	Image	Ånh gương mặt	
Card Image	Image	Ånh thẻ sinh viên	
Crop Card Image	Image	Ånh thẻ sinh viên được cắt ra để nhận dạng	
	Các hành động	trong giao diện	
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
	NIO ta	Thain cong	That bật (E01)
Chỉnh sửa thông tin	Nhập vào các thông tin muốn thay đổi của ca thi	Thann cong	That bật (Lioi)
	Nhập vào các thông tin muốn thay đổi	Dialog được đóng lại và thông tin của sinh viên được cập nhật lên hệ thống	That bật (Lioi)

Bảng 11: Bảng mô tả chức năng Sửa kết quả điểm danh

8.1.2 Sinh viên

8.1.2.1 Kiểm tra mã ca thi



Hình 53: Mô tả chức năng Kiểm tra khung giờ điểm danh

Giao diện	Nhập mã ca thi để	Nhập mã ca thi để điểm danh		
Mô tả	Cho phép sinh vié	Cho phép sinh viên nhập mã ca thi để tiến hành việc điểm danh		
Truy cập	Sinh viên truy cập	Sinh viên truy cập vào trang https://sv.illusion.codes		
Nội dung giao diện				
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả	
Exam code	Text Field	Mã ca thi	Trường nhập mã code để thực hiện điểm danh cho ca thi đó	
Các hành động trong giao diện				

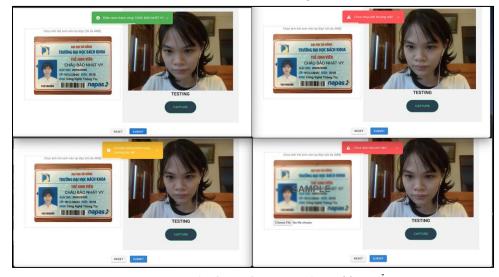
Hành động	Mô tả	Thành công	Thất bại (Lỗi)
Submit	Nhập mã ca thi, sau đó bấm nút enter	Chuyển đến trang điểm danh ca thi	Chưa đến thời gian điểm danh: Hiển thị thông báo "Không phải khung giờ điểm danh" Sai mã ca thi: Hiển thị thông báo "Mã ca thi không hợp lệ"

Bảng 12: Bảng mô tả chức năng Kiểm tra khung giờ điểm danh

8.1.2.2 Điểm danh



Hình 55: Hình mô tả chức năng Điểm danh



Hình 54: Hình mô tả các trường hợp xảy ra khi Điểm danh

Giao diện	Điểm danh so khớp gương mặt và nhận diện thẻ sinh viên					
Mô tả	Sinh viên upload ảnh thẻ sinh viên và chụp ảnh gương mặt để điểm danh					
Truy cập	Sinh viên nhập mã ca thi thành công và chuyển đến trang điểm danh https://sv.illusion.codes/checking/exam-code					
	Nội	dung giao diện				
Thành phần	Kiểu dữ liệu	Dữ liệu	Mô tả			
Examination	Text Field	Mã ca thi (exam code)	Text field hiển thị mã ca thi (không thể thay đổi)			
Card Image	Image - Base 64	Ånh thẻ sinh viên	Ånh thẻ được upload lên website			
Face Image	Image - Base 64	Ånh chụp gương mặt	Ånh gương mặt được chụp trên website			
	Các hành động trong giao diện					
Hành động	Mô tả	Thành công	Thất bại (Lỗi)			
Submit			Hiển thị thông báo "So khớp gương mặt không thành công. Vui lòng thử lại!"			
	Submit ảnh thẻ sinh viên và ảnh gương mặt lên server và tiến hành điểm danh	Hiển thị thông báo:"Điểm danh thành công! [TenSinhVien]"	Chưa có thẻ sinh viên: Hiển thi thông báo "Chưa có ảnh thẻ sinh viên"			
			Chưa có ảnh chụp gương mặt: Hiển thị thông báo: "Chưa chụp ảnh gương			

Reset lại ảnh sinh viên đã upload và ảnh gương mặt đã chụp	Xoá ảnh thẻ sinh viên và ảnh gương mặt đã chụp
--	--

Bảng 13: Bảng mô tả chức năng Điểm danh

8.2 Đánh giá

Ưu điểm

- Hệ thống được xây dựng đã hoàn thành các yêu cầu chức năng và phi chức năng đã đề ra.
- Nhờ vào việc lựa chọn các công nghệ và kiến trúc hợp lí, hệ thống có khả năng chịu tải cao, tốc độ xử lý đã được tối ưu với phần cứng đã triển khai, có khả năng mở rộng và triển khai dễ dàng.
- Ngoài ra, hệ thống còn được trang bị các biện pháp bảo mật để bảo vệ dữ liệu và chính hệ thống.

• Nhược điểm

- Mặc dù đã giải quyết được bài toán so khớp khuôn mặt tuy nhiên độ chính xác chưa được ổn định, phụ thuộc nhiều vào chất lượng ảnh đầu vào và vẫn còn thiếu các giải pháp chống giả mạo khuôn mặt.
- Hệ thống đã triển khai bao gồm nhiều thành phần khiến cho việc quản lý trở nên phức tạp và khó khăn.

CHƯƠNG 9: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

9.1 Kết luận

- Hệ thống đã đáp ứng được các yêu cầu mà đề tài đã đặt ra, giải quyết được các vấn đề đã đề câp và đáp ứng các yêu tiêu chí mà nhóm đã xác đinh khi bắt đầu đồ án.
- Hệ thống tuy hoạt động nhưng còn một số hạn chế như: thiếu giải pháp chống giả mạo khuôn mặt, số lượng chức năng vẫn còn tương đối ít, ...

9.2 Hướng phát triển

- Triển khai thêm phần thi Online để tạo thành một hệ thống thi Online hoàn chỉnh. Tích hợp các giải pháp chống giả mạo khuôn mặt vào trong quá trình thi Online
- Bổ sung thêm các chức năng quản lý khác cho Giảng viên như xuất báo cáo, cảnh báo trên điện thoại, ...
- Xây dựng phiên bản ứng dụng điện thoại để tận dụng khả năng của Camera điện thoại trong việc cải thiện độ chính xác mô hình so khớp khuôn mặt.
- Triển khai hệ thống với phần cứng mạnh hơn để có thể đáp ứng được nhiều lượt truy cập cùng lúc.

TÀI LIỆU THAM KHẢO

- [1] Roboflow, "YOLOv4 versus YOLOv5," https://blog.roboflow.com/yolov4-versus-yolov5/.
- [2] face_recognition. https://github.com/ageitgey/face_recognition.
- [3] P. B. C. Quốc, "Nhận Dạng Tiếng Việt Sử Dụng Mô Hình Transformer và AttentionOCR," https://pbcquoc.github.io/vietocr/..
- [4] P. D. Khánh, "Khoa học dữ liệu Khanh's blog," https://phamdinhkhanh.github.io/...
- [5] Ultralytics, "YOLOv5," https://github.com/ultralytics/yolov5.
- [6] N. V. Anh, "Viblo," https://viblo.asia/p/transformers-nguoi-may-bien-hinh-bien-doi-the-gioi-nlp-924lJPOXKPM.
- [7] Nginx, "Nginx Document," https://docs.nginx.com/.
- [8] F. A. Documentation. https://fastapi.tiangolo.com/.
- [9] L. |. P. Yang, "Message Queue Architecture," https://medium.com/must-know-computer-science/system-design-message-queues-245612428a22.
- [10] A. Thevenot. https://towardsdatascience.com/image-augmentation-mastering-15-techniques-and-useful-functions-with-python-codes-44c3f8c1ea1f.
- [11] MATT HOLDSWORTH. https://bytes.fyi/real-time-goaccess-reports-with-nginx/.