

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Hệ thống quản lý dạy học và nhân sự cho  
Lớp học Cầu Vồng**

**NGUYỄN THỊ HẢI THANH**  
thanh.nth176871@sis.hust.edu.vn

**Ngành Công nghệ thông tin**

**Giảng viên hướng dẫn:** PGS. TS. Cao Tuấn Dũng

Chữ ký GVHD

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# LỜI CẢM ƠN

Vậy là thoáng cái, năm năm học đã trôi qua. Mới ngày nào còn bỡ ngỡ, bước chân vào cổng Trường Đại học Bách Khoa Hà Nội, choáng ngợp trước sự đẹp đẽ và rộng lớn của ngôi trường. Vậy mà giờ đây, em đã và đang trải qua những chặng đường cuối cùng để chuẩn bị cho một hành trình mới, một hành trình mà ở đó không có thầy cô tận tình chỉ bảo, không có những tháng ngày thi cử mệt nhọc, nhưng đó sẽ là hành trình mà em có thể chứng tỏ bản thân thông qua việc áp dụng những kiến thức đã được học vào trong công việc và cuộc sống hàng ngày.

Trước hết, em xin được gửi lời cảm ơn sâu sắc nhất tới thầy giáo, PGS.TS Cao Tuấn Dũng đã giúp đỡ và hướng dẫn em trong suốt quá trình thực hiện đồ án tốt nghiệp. Thầy luôn tận tình, chỉ bảo, đôn đốc em xây dựng một lộ trình làm đồ án tốt nghiệp. Bên cạnh đó, thầy cũng thường xuyên đưa ra những nhận xét, định hướng giúp em hoàn thành đồ án một cách tốt nhất. Nếu không có những lời góp ý của thầy thì chắc chắn sản phẩm của em sẽ không có kết quả như ngày hôm nay. Em cảm thấy rất biết ơn vì đã được thầy hướng dẫn trong quá trình hoàn thành đồ án tốt nghiệp này. Em chúc thầy luôn mạnh khỏe, công tác tốt để có thể xây dựng một ngôi Trường Công nghệ thông tin và Truyền thông vững mạnh, đào tạo được thêm nhiều thế hệ sinh viên Bách Khoa bay cao, bay xa, góp phần xây dựng sự tiến bộ Công nghệ kĩ thuật của nước nhà!

Tiếp theo, em xin gửi lời cảm ơn đến tổ chức Lớp học Cầu Vồng – The Rainbow Class đã đồng ý hợp tác cùng em để có thể xây dựng, hoàn thành hệ thống quản lý dành riêng cho Lớp học Cầu Vồng. Thông qua những buổi nói chuyện, phỏng vấn cùng người phụ trách của tổ chức, em đã có một cái nhìn tổng quan hơn về công việc phân tích yêu cầu của khách hàng, góp phần tích lũy kinh nghiệm thực tế, áp dụng vào công việc sau này. Chúc Lớp học Cầu Vồng ngày càng phát triển, có thể giúp đỡ được nhiều mảnh đời kém may mắn, luôn lan tỏa những điều tốt đẹp và tích cực đến với cộng đồng!

Cuối cùng, xin được cảm ơn tới những người bạn yêu quý đã luôn kề vai sát cánh trong những năm tháng sinh viên tại trường Đại học Bách Khoa Hà Nội. Không chỉ trong học tập, cảm ơn các bạn đã chia sẻ cùng tôi những niềm vui, nỗi buồn, những trải nghiệm của năm tháng sinh viên không thể nào quên. Ai cũng có những quyết định, có lối đi riêng mình, hy vọng tất cả chúng ta đều thành công và vững bước trên con đường mình đã chọn!

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Lớp học Cầu vồng là một tổ chức từ thiện phi lợi nhuận được thành lập nhằm tạo điều kiện giúp đỡ, động viên những mảnh đời kém may mắn về cả vật chất lẫn tình thương. Bên cạnh đó Lớp học tìm kiếm những bạn Tình nguyện viên hỗ trợ dạy học, giúp đỡ các em nhỏ có thêm cơ hội mở rộng cánh cửa tương lai.

Hiện nay, Lớp học Cầu Vồng đang tổ chức các lớp học tình thương, giảng dạy trực tiếp cho các em tại các tỉnh thành như Hà Nội, Đà Nẵng và Lào Cai cùng nhiều lớp học trực tuyến khác. Với số lượng gần 200 tình nguyện viên đang tham gia giảng dạy cho nhiều trẻ em có hoàn cảnh khó khăn, việc quản lý lớp học và các buổi học cho các lớp cũng trở nên phức tạp bởi vì cách thức quản lý hiện nay của Lớp học Cầu Vồng chủ yếu là sử dụng các dịch vụ của Google hoặc liên lạc thông qua các trang mạng xã hội như Facebook, Zalo,... hoặc các phương tiện cá nhân, chưa có một hệ thống riêng đặc trưng phục vụ việc quản lý và liên lạc.

Chính vì vậy, với mong muốn đóng góp một phần công sức giúp đỡ Lớp học Cầu Vồng có thể dễ dàng hơn trong việc quản lý dạy học và nhân sự, cũng là đóng góp một phần giúp cho tổ chức ngày càng thêm lớn mạnh, em đã có mong muốn phát triển một hệ thống quản lý dạy học và nhân sự dành riêng cho Lớp học Cầu Vồng, trước hết hướng tới đối tượng là người quản trị viên, các tình nguyện viên tham gia tổ chức, những đối tượng có hứng thú với Lớp học Cầu Vồng và có mong muốn trở thành Tình nguyện viên của tổ chức trên nền tảng Web. Hệ thống sẽ có các chức năng quản lý nhân sự, quản lý lớp và các bài học dựa theo đặc thù và các quy trình nghiệp vụ mà tổ chức đã quy định. Bên cạnh đó cải thiện và tự động hóa một số quy trình nghiệp vụ, giúp người sử dụng giảm bớt khối lượng công việc, tiết kiệm thời gian hơn để thực hiện các nghiệp vụ quản lý khác.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án .....	4
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>6</b>
2.1 Khảo sát hiện trạng .....	6
2.1.1 Tổng quan về Lớp học Cầu Vồng .....	6
2.1.2 Cơ cấu tổ chức của Lớp học Cầu Vồng .....	7
2.1.3 Các hoạt động quản lý công việc của Lớp học Cầu Vồng hiện nay ..	8
2.2 Khảo sát các chức năng cần thiết cho hệ thống .....	10
2.3 Tổng quan chức năng .....	11
2.3.1 Biểu đồ use case tổng quát .....	11
2.3.2 Biểu đồ use case phân rã.....	15
2.3.3 Quy trình nghiệp vụ .....	21
2.4 Đặc tả chức năng .....	23
2.4.1 Đặc tả use case “Đăng ký làm tình nguyện viên” .....	23
2.4.2 Đặc tả use case “Đăng ký dạy học với lớp theo hình thức dạy tập trung” .....	24
2.4.3 Đặc tả use case “Nhập ngày nghỉ hoạt động cho TNV” .....	25
2.5 Yêu cầu phi chức năng .....	27
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>28</b>
3.1 Mô hình Client-Server.....	28

3.2 MERN Stack .....	28
3.2.1 Tổng quan về MERN Stack .....	28
3.2.2 MongoDB .....	29
3.2.3 NodeJS .....	30
3.2.4 ExpressJS .....	31
3.2.5 ReactJS .....	32
3.3 Dịch vụ lưu trữ dữ liệu Azure Blob Storage.....	33
3.4 Các thư viện được sử dụng.....	34
3.4.1 Thư viện frontend .....	34
3.4.2 Thư viện backend.....	34
3.5 OAuth 2 .....	35
<b>CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG.....</b>	<b>36</b>
4.1 Thiết kế kiến trúc.....	36
4.1.1 Lựa chọn kiến trúc phần mềm .....	36
4.1.2 Thiết kế tổng quan.....	37
4.1.3 Thiết kế chi tiết gói .....	38
4.2 Thiết kế chi tiết.....	40
4.2.1 Thiết kế giao diện .....	40
4.2.2 Thiết kế lớp .....	42
4.2.3 Thiết kế cơ sở dữ liệu .....	49
4.3 Xây dựng ứng dụng.....	56
4.3.1 Thư viện và công cụ sử dụng.....	56
4.3.2 Kết quả đạt được .....	56
4.3.3 Minh họa các chức năng chính .....	58
4.4 Kiểm thử.....	65
4.5 Triển khai .....	69

**CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT ..... 71**

5.1 Hỗ trợ cải tiến quy trình thu thập hồ sơ đăng ký tình nguyện viên.....	71
5.1.1 Cách thức tuyển tình nguyện viên khi chưa sử dụng hệ thống .....	71
5.1.2 Hỗ trợ cải tiến quy trình thu thập hồ sơ đăng ký tình nguyện viên ...	72
5.2 Hỗ trợ cải tiến nghiệp vụ xét duyệt hồ sơ đăng ký tình nguyện viên .....	73
5.3 Hỗ trợ thông báo đến người dùng .....	75
5.4 Hỗ trợ cải tiến nghiệp vụ đăng ký dạy học.....	76

**CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ..... 78**

6.1 Kết luận .....	78
6.2 Hướng phát triển.....	79

**TÀI LIỆU THAM KHẢO..... 80**



## DANH MỤC HÌNH VẼ

Hình 1.1	Chứng nhận LHCV là thành viên mạng lưới Tình nguyện Quốc gia khu vực miền Bắc trực thuộc VCC . . . . .	1
Hình 2.1	Tình nguyện viên và các em học sinh tại Lớp học Sapa . . . . .	6
Hình 2.2	LHCV trở lại học trực tiếp sau những ngày học trực tuyến vì dịch Covid-19 . . . . .	9
Hình 2.3	Biểu đồ use case tổng quát . . . . .	12
Hình 2.4	Biểu đồ use case phân rã quản lý Tình nguyện viên . . . . .	15
Hình 2.5	Biểu đồ use case phân rã Quản lý học sinh . . . . .	16
Hình 2.6	Biểu đồ use case phân rã Quản lý CV . . . . .	18
Hình 2.7	Biểu đồ use case phân rã Quản lý lớp học . . . . .	19
Hình 2.8	Biểu đồ use case phân rã Quản lý bài học theo hai hình thức dạy . . . . .	20
Hình 2.9	Biểu đồ use case phân rã Quản lý lịch cá nhân . . . . .	21
Hình 2.10	Quy trình nghiệp vụ duyệt CV . . . . .	22
Hình 2.11	Sơ đồ luồng đặc tả use case đăng ký làm TNV . . . . .	23
Hình 2.12	Sơ đồ luồng đặc tả use case đăng ký dạy đối với lớp theo hình thức dạy tập trung . . . . .	24
Hình 2.13	Sơ đồ luồng đặc tả use case "Nhập ngày nghỉ hoạt động cho TNV" . . . . .	26
Hình 3.1	Mô hình client - server . . . . .	28
Hình 3.2	Xử lý dữ liệu trong MERN Stack . . . . .	29
Hình 3.3	Ví dụ dữ liệu kiểu BSON . . . . .	29
Hình 3.4	Kết quả khảo sát web framework được ưa thích nhất của Stack Overflow . . . . .	33
Hình 3.5	Mối quan hệ giữa các thành phần của Azure Blob Storage . . . . .	34
Hình 3.6	Quá trình xử lý khi đăng nhập bằng Oauth2 . . . . .	35
Hình 4.1	Kiến trúc nguyên khối (Monolithic Architecture) . . . . .	36
Hình 4.2	Thiết kế tổng quan của hệ thống . . . . .	37
Hình 4.3	Thiết kế gói module gửi email . . . . .	38
Hình 4.4	Thiết kế gói module tải lên tệp tin . . . . .	39
Hình 4.5	Thiết kế gói module quản lý . . . . .	40
Hình 4.6	Thiết kế mockup giao diện cho màn hình đăng ký làm TNV . . . . .	41
Hình 4.7	Thiết kế mockup giao diện cho Login Layout . . . . .	41

Hình 4.8	Thiết kế giao diện mockup cho các màn hình quản lý . . . . .	42
Hình 4.9	Biểu đồ tuần tự cho chức năng "đăng ký làm TNV" . . . . .	48
Hình 4.10	Biểu đồ thực thể liên kết . . . . .	50
Hình 4.11	Màn hình giới thiệu tổ chức trên mobile và trên desktop . . . . .	58
Hình 4.12	Màn hình hỗ trợ đăng ký làm TNV trên mobile và trên desktop	58
Hình 4.13	Màn hình đăng nhập trên desktop . . . . .	59
Hình 4.14	Màn hình Dashboard của người dùng có chức vụ là quản trị viên . . . . .	60
Hình 4.15	Màn hình Dashboard của người dùng có chức vụ là học sinh .	60
Hình 4.16	Màn hình Quản lý lịch lớp của người dùng có chức vụ là cán sự lớp . . . . .	61
Hình 4.17	Màn hình danh sách Quản trị viên . . . . .	61
Hình 4.18	Màn hình danh sách tình nguyện viên . . . . .	62
Hình 4.19	Màn hình danh sách học sinh . . . . .	62
Hình 4.20	Màn hình chi tiết thông tin TNV . . . . .	63
Hình 4.21	Màn hình chi tiết lớp học theo kiểu dạy tập trung - tab thành tích . . . . .	63
Hình 4.22	Màn hình chi tiết lớp học theo kiểu dạy kèm - tab danh sách dạy kèm . . . . .	64
Hình 4.23	Màn hình danh sách những hồ sơ bị loại . . . . .	64
Hình 4.24	Màn hình Cài đặt chung - tab quản lý các môn học . . . . .	65
Hình 4.25	Email mà người dùng nhận được để kích hoạt tài khoản . . . . .	65
Hình 5.1	Cách thức đăng ký trở thành TNV của Lớp học Cầu Vồng . . . . .	71
Hình 5.2	Các câu hỏi được QTV cài đặt tại website quản lý . . . . .	72
Hình 5.3	Các câu hỏi được QTV cài đặt hiển thị trên website hỗ trợ đăng ký làm TNV . . . . .	73
Hình 5.4	Sơ đồ quy trình nghiệp vụ tổng hợp danh sách TNV đăng ký dạy học hàng tuần khi không sử dụng hệ thống . . . . .	77

## DANH MỤC BẢNG BIỂU

Bảng 2.1	Phân tích use case với tác nhân là quản trị viên . . . . .	13
Bảng 2.2	Phân tích use case với tác nhân là cán sự lớp . . . . .	14
Bảng 2.3	Phân tích use case với tác nhân là tình nguyện viên . . . . .	14
Bảng 2.4	Phân tích use case với tác nhân là Học sinh . . . . .	15
Bảng 2.5	Phân tích use case với tác nhân là Khách . . . . .	15
Bảng 2.6	Mô tả use cây phân rã Quản lý tình nguyện viên . . . . .	16
Bảng 2.7	Mô tả use cây phân rã Quản lý học sinh . . . . .	17
Bảng 2.8	Mô tả use cây phân rã Quản lý CV . . . . .	19
Bảng 2.9	Mô tả use cây phân rã Quản lý lớp học . . . . .	20
Bảng 2.10	Mô tả use cây phân rã Quản lý bài học . . . . .	21
Bảng 2.11	Mô tả use cây phân rã Quản lý Lịch cá nhân . . . . .	21
Bảng 2.12	Phân tích use case đăng ký làm TNV với tác nhân là Khách . .	23
Bảng 2.13	Phân tích use case Đăng ký dạy đối với lớp theo hình thức dạy tập trung . . . . .	25
Bảng 2.14	Phân tích use case "Nhập ngày nghỉ hoạt động cho TNV" . .	25
Bảng 4.1	Lớp Upload Controller . . . . .	42
Bảng 4.2	Lớp Upload File Service . . . . .	43
Bảng 4.3	Lớp SendMailService . . . . .	43
Bảng 4.4	Lớp User Controller . . . . .	44
Bảng 4.5	Lớp StudentController . . . . .	44
Bảng 4.6	Lớp VolunteerController . . . . .	45
Bảng 4.7	Lớp CVRepository . . . . .	45
Bảng 4.8	Lớp CVController . . . . .	46
Bảng 4.9	Lớp Class Repository . . . . .	46
Bảng 4.10	Lớp ClassController . . . . .	47
Bảng 4.11	Lớp VolunteerController . . . . .	48
Bảng 4.12	Bảng dữ liệu User lưu các thông tin người dùng . . . . .	51
Bảng 4.13	Bảng dữ liệu Student lưu các thông tin về học sinh . . . . .	51
Bảng 4.14	Bảng dữ liệu Volunteer lưu các thông tin về tình nguyện viên .	52
Bảng 4.15	Bảng dữ liệu Class lưu các thông tin về các lớp học của Lớp học Cầu Vồng . . . . .	52
Bảng 4.16	Bảng dữ liệu PairTeaching lưu thông tin về các cặp dạy kèm .	53
Bảng 4.17	Bảng dữ liệu Lesson lưu các thông tin về bài học . . . . .	53
Bảng 4.18	Bảng dữ liệu Schedule lưu các thông tin về lịch hoạt động . .	54

Bảng 4.19	Bảng dữ liệu Achievement lưu các thông tin về thành tích của học sinh . . . . .	54
Bảng 4.20	Bảng dữ liệu Report lưu các thông tin báo cáo sau buổi học . . . . .	55
Bảng 4.21	Bảng dữ liệu CV lưu các hồ sơ đăng ký làm tình nguyện viên . . . . .	55
Bảng 4.22	Bảng dữ liệu Notification lưu các thông báo . . . . .	56
Bảng 4.23	Thư viện và công cụ sử dụng . . . . .	56
Bảng 4.24	Kết quả đóng gói sản phẩm . . . . .	57
Bảng 4.25	Kịch bản kiểm thử cho chức năng đăng ký làm tình nguyện viên . . . . .	66
Bảng 4.26	Kịch bản kiểm thử cho chức năng duyệt CV . . . . .	68
Bảng 4.27	Kịch bản kiểm thử cho chức năng quản lý bài học (lớp dạy theo hình thức tập trung) . . . . .	69
Bảng 5.1	So sánh ưu điểm khi người phụ trách duyệt hồ sơ sử dụng hệ thống . . . . .	75
Bảng 5.2	Các trường hợp gửi thông báo đến người dùng . . . . .	76

## DANH MỤC VIẾT TẮT

<b>Viết tắt</b>	<b>Ý nghĩa</b>
<b>LHCV</b>	Lớp học Cầu Vồng
<b>VCC</b>	Vietnam Volunteer Center - Trung tâm tình nguyện quốc gia
<b>QTV</b>	Quản trị viên
<b>CSL</b>	Cán sự lớp
<b>TNV</b>	Tình nguyện viên
<b>HS</b>	Học sinh
<b>CV</b>	Curriculum Vitae - Hồ sơ ứng tuyển
<b>CSDL</b>	Cơ sở dữ liệu
<b>ĐATN</b>	Đồ án tốt nghiệp

## DANH MỤC THUẬT NGỮ

Thuật ngữ	Ý nghĩa
<b>Browser</b>	Trình duyệt
<b>Framework</b>	Các đoạn code đã được viết sẵn, cấu thành nên bộ khung và các thư viện được đóng gói
<b>Backend</b>	Tầng truy cập xử lý dữ liệu, xử lý nghiệp vụ của phần mềm
<b>API</b>	Application Programming Interface - Giao diện lập trình ứng dụng
<b>Frontend</b>	Tầng trình diễn (giao diện người dùng)
<b>Component</b>	Các thành phần giao diện độc lập có thể tái sử dụng trong thư viện ReactJS
<b>File</b>	Tập tin
<b>Deploy</b>	Triển khai

## CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

### 1.1 Đặt vấn đề

Lớp học Cầu Vồng là một tổ chức từ thiện phi lợi nhuận, là thành viên mạng lưới Tình nguyện Quốc gia khu vực miền Bắc trực thuộc VCC, được thành lập nhằm tạo điều kiện giúp đỡ, động viên những mảnh đời kém may mắn về cả vật chất lẫn tinh thần. Đồng thời, lớp học cũng tìm kiếm những bạn tình nguyện viên tình nguyện tham gia hỗ trợ dạy học, giúp đỡ các em nhỏ có thêm cơ hội mở rộng cánh cửa tương lai.

Hiện nay, Lớp học Cầu Vồng đang tổ chức các lớp học tình thương, giảng dạy cho các em tại nhiều tỉnh thành trên cả nước. Với số lượng 18 lớp học trực tiếp cùng với hơn 200 bạn tình nguyện viên đang hoạt động, quy mô của Lớp học Cầu Vồng ngày càng được mở rộng, số lượng thành viên đông đảo khiến cho việc quản lý các buổi học, nhân sự của các lớp cũng dần trở nên phức tạp.



**Hình 1.1:** Chứng nhận LHCV là thành viên mạng lưới Tình nguyện Quốc gia khu vực miền Bắc trực thuộc VCC<sup>1</sup>.

Cách tổ chức hiện tại của Lớp học Cầu Vồng trước hết là chọn ra những tình nguyện viên ưu tú của từng lớp để trực tiếp quản lý các hoạt động như tuyển tình nguyện viên mới hay xây dựng các lịch trình giảng dạy cho lớp học đó. Các lớp

<sup>1</sup>Nguồn: <https://www.facebook.com/therainbowclass/photos/a.184942138578408/1252316485174296/>

học như vậy sẽ tổ chức các hoạt động một cách độc lập, riêng lẻ. Bên cạnh đó các tình nguyện viên thuộc ban quản lý sẽ tham gia một nhóm chat chung để có thể chia sẻ, cập nhật thông tin và các thành viên thuộc ban quản lý cũng phải tham gia nhiều nhóm chat riêng của các lớp. Như vậy, cách thức hoạt động này không phải là phương án tiện nhất và ban quản lý của Lớp học Cầu Vồng sẽ gặp khó khăn trong việc chủ động theo dõi được các kế hoạch, lịch trình của từng lớp học.Thêm vào đó, việc tương tác để sắp xếp lịch dạy giữa các thành viên trong một lớp thông qua các trang mạng xã hội phải thực hiện các bước lặp đi lặp lại hằng tuần (ví dụ như tạo các khảo sát hàng tuần trên Facebook để xác nhận các tình nguyện viên đăng ký dạy trong tuần rồi sau đó phải tổng hợp lại vào Google sheet) khiến khối lượng công việc quản lý trở nên nhiều hơn.

Bên cạnh đó, việc theo dõi, quản lý các tình nguyện viên mới cũng khó khăn do các hồ sơ đăng ký ứng tuyển vào tổ chức đều được gửi đến một địa chỉ email nhất định trong khi các lớp học hoạt động riêng lẻ. Những người phụ trách sẽ phải thực hiện các hành động thủ công để tìm kiếm, chọn lọc những hồ sơ rồi phân loại hồ sơ, trả về cho từng lớp học để thực hiện liên hệ và phỏng vấn.

Chính vì vậy, xét đến sự phát triển của Lớp học Cầu Vòng trong tương lai vẫn được duy trì và mở rộng quy mô hoạt động, nếu chỉ sử dụng những dịch vụ của Google, các hệ thống mạng xã hội như Zalo, Facebook,... để quản lý những công việc trên thì sẽ mang lại nhiều khó khăn như tốn thời gian sắp xếp nhân sự cũng như các tiết giảng dạy đối với ban quản lý nói riêng và các bạn tình nguyện viên nói chung. Nếu có một hệ thống đặc thù hỗ trợ việc quản lý cho Lớp học Cầu Vòng được phát triển thì sẽ có thể cải tiến các quy trình nghiệp vụ và tiết kiệm được thêm nhiều thời gian hơn cho các thành viên của tổ chức trong các hoạt động thiện nguyện. Đó chính là lý do và động lực khiến em thực hiện đồ án "Hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vòng" này.

## 1.2 Mục tiêu và phạm vi đề tài

Chính vì những khó khăn được nêu như ở trên, em nhận thấy Lớp học Cầu Vòng cần có một hệ thống đặc thù hỗ trợ công việc quản lý dạy học và nhân sự riêng, trước hết hướng tới đối tượng là người quản trị và các tình nguyện viên tham gia lớp học trên nền tảng Web và mở rộng đối tượng là các em học sinh hoặc phụ huynh. Hệ thống này sẽ tập trung giải quyết hai vấn đề chính là vấn đề quản lý lớp học và vấn đề quản lý nhân sự.

**Vấn đề quản lý dạy học:** Hệ thống cần hỗ trợ ban quản lý và các bạn tình nguyện viên trong công việc quản lý lớp học và những vấn đề liên quan đến lớp học như cho phép quản lý lịch học của từng lớp; quản lý nội dung các bài học và

thống kê những tình nguyện viên đăng ký tham gia dạy học; quản lý báo cáo sau mỗi buổi học; quản lý thành tích của từng học sinh tham gia lớp học theo từng thời kì; tự động đặt lịch, thông báo, nhắc nhở các tình nguyện viên hoàn thành các công việc cá nhân cũng như các công việc chung của lớp học.

**Về vấn đề quản lý nhân sự:** Hệ thống cần hỗ trợ ban quản lý trong công việc quản lý nhân sự của các lớp học và các vấn đề liên quan cơ bản nhất như thông tin cơ bản và số lượng các tình nguyện viên; thông tin cá nhân và thông tin về thành tích các học sinh tham gia lớp học. Bên cạnh đó, một trong những điểm nổi bật của hệ thống là việc hệ thống hỗ trợ nhận, lưu trữ các hồ sơ đăng ký tham gia làm tình nguyện viên của Lớp học Cầu Vồng. Với chức năng này, ban quản lý có thể theo dõi được thông tin và lịch phỏng vấn các tình nguyện viên mới một cách nhanh chóng, chủ động nhất. Bên cạnh đó, có thể giúp cải tiến, hệ thống hóa quy trình nghiệp vụ xét duyệt hồ sơ đăng ký ứng tuyển làm tình nguyện viên.

Riêng đối với học sinh tham gia học tập tại Lớp học Cầu Vồng, do phần lớn các em có hoàn cảnh khó khăn và là đối tượng yếu thế trong xã hội nên không phải học sinh nào cũng được tiếp xúc với các thiết bị thông minh để dùng hệ thống. Chính vì vậy, trong phạm vi đồ án, em chỉ xây dựng một vài tính năng cơ bản hướng tới đối tượng là học sinh.

### 1.3 Định hướng giải pháp

Từ những vấn đề đã được nêu ở mục 1.1, với mong muốn đóng góp một phần công sức giúp đỡ Lớp học Cầu Vồng có thể dễ dàng hơn trong việc quản lý dạy học và nhân sự, cũng là đóng góp một phần giúp cho Lớp học Cầu Vồng thêm lớn mạnh, giải pháp em đặt ra là xây dựng một hệ thống Web bao gồm một website quản lý dạy học và nhân sự dành cho những thành viên thuộc Lớp học Cầu Vồng và một website hỗ trợ những người dùng có mong muốn tham gia làm tình nguyện viên của tổ chức đăng ký tham gia làm tình nguyện viên. Hai trang web này sẽ sử dụng chung một Backend để lấy các API thực hiện các chức năng chính. Nội dung cụ thể của hai trang web này như sau:

Tại **trang web quản lý dạy học và nhân sự** dành cho các tình nguyện viên thuộc Lớp học Cầu Vồng, mỗi tình nguyện viên sẽ được cung cấp một tài khoản với các vai trò khác nhau, tùy vào chức vụ của tình nguyện viên trong tổ chức. Khi đăng nhập vào hệ thống, các tình nguyện viên sẽ có thể thực hiện được các thao tác liên quan đến quản lý dạy học hoặc quản lý nhân sự tùy thuộc vào quyền hạn mà tình nguyện viên đó được cấp. Các tài khoản được cấp sẽ được chia thành ba vai trò chính:(i) Quản trị viên – những tình nguyện viên quản lý tổ chức: có mọi quyền hạn sử dụng các chức năng trong hệ thống ; (ii) Quản lý lớp (thường được gọi là

cán sự lớp, lớp trưởng, lớp phó): có mọi quyền hạn sử dụng các chức năng của lớp mà họ quản lý; (iii) Tình nguyện viên – những tình nguyện viên chỉ tham gia dạy học: đăng ký tham gia dạy học tại lớp mà họ tham gia.

**Trang web hỗ trợ đăng ký làm tình nguyện viên** thu thập hồ sơ đăng ký từ những người dùng quan tâm và có mong muốn tham gia làm tình nguyện viên của Lớp học Cầu Vồng, người dùng không cần đăng nhập vào hệ thống mà vẫn có thể xem được các thông tin giới thiệu về lớp học, các hoạt động thường niên mà lớp học tổ chức. Bên cạnh đó, những người dùng có nhu cầu đăng ký trở thành tình nguyện viên của Lớp học Cầu Vồng có thể điền form thông tin, gửi CV của bản thân về hệ thống và chờ thông tin phản hồi.

Trong phạm vi đồ án, hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng được xây dựng bằng cách sử dụng NodeJS ứng dụng ExpressJS framework ở phía backend và ReactJS framework ở phía frontend.

#### 1.4 Bố cục đồ án

Sau khi xác định được vấn đề, mục tiêu, phạm vi đề tài và định hướng giải pháp, phần còn lại của đồ án được trình bày theo cấu trúc như sau:

Trong Chương 2, em sẽ trình bày chi tiết quá trình phân tích ứng dụng. Trước hết là việc khảo sát hiện trạng thực tế cách quản lý lớp học của Lớp học Cầu Vồng, những vấn đề mà Lớp học Cầu Vồng gặp phải và cần giải quyết để từ đó phân tích các tác nhân và các chức năng chính của hệ thống thông qua các biểu đồ use case và biểu đồ phân tích nghiệp vụ.

Chương 3 giới thiệu cụ thể hơn về các công nghệ sử dụng để xây dựng được hệ thống đảm bảo được mục tiêu đề ra ở phần 2.2. Trong chương này em sẽ trình bày các công nghệ được sử dụng để phát triển ứng dụng và lý do tại sao lựa chọn các công nghệ đó.

Chương 4 trình bày về cách kết hợp những công nghệ đã được trình bày ở Chương 3 để tạo nên kiến trúc cho toàn bộ hệ thống. Phần Backend và Frontend được phát triển độc lập, giao tiếp với nhau thông qua API. Từ kiến trúc đó em sẽ trình bày thêm về việc phát triển từng chức năng của ứng dụng, cùng với đó là quá trình kiểm thử để đi đến kết quả hoàn thiện.

Chương 5, em sẽ trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Cụ thể là phân tích các chức năng được cải tiến và hệ thống hóa một số quy trình nghiệp vụ, giúp người dùng bỏ qua những bước lặp lại thường xuyên, tiết kiệm thời gian và tối ưu hóa công việc quản lý.

Cuối cùng, chương 6 là chương kết của báo cáo đồ án. Thông qua chương này,

em nhìn lại một cách tổng quan kết quả mình đã đạt được, tổng kết kiến thức đã học và các kinh nghiệm rút ra trong quá trình làm ĐATN. Cuối cùng là một vài hướng phát triển để cải thiện, nâng cao chất lượng hệ thống cũng như quy trình nghiệp vụ mà em đã chưa kịp thực hiện được.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

#### 2.1.1 Tổng quan về Lớp học Cầu Vồng

**Lớp học Cầu vồng - The Rainbow Class** là một dự án từ thiện phi lợi nhuận được thành lập nhằm tạo điều kiện giúp đỡ, động viên những mảnh đời kén may mắn về cả vật chất lẫn tình thương, đặc biệt là những trẻ em chậm phát triển, không có cơ hội được đến trường; đồng thời tìm kiếm những bạn tình nguyện viên để hỗ trợ dạy học, giúp đỡ các em nhỏ có thêm cơ hội mở rộng cánh cửa tương lai. Chính vì vậy, hoạt động chính của Lớp học Cầu Vồng chính là dạy học tình nguyện.

Hiện tại Lớp học Cầu Vồng đang tổ chức 19 lớp học tình thương giảng dạy trực tiếp trên 3 tỉnh thành: Hà Nội, Lào Cai, Đà Nẵng. Bao gồm: Lớp Phúc Xá, Lớp Phúc Tuệ, Blossom House, Ánh sáng Kim Ngưu (3 lớp), Bright Future, Lớp Cánh Diều, Lớp Ong Mật, Lớp Bắc Hà, Lớp Sa Pa, Lớp Simacai, Lớp Y Tý, Lớp Rồng Xanh, Lớp Bốn Mùa, Lớp Quán Thánh, Lớp Hàn Gắn, Lớp Tiếng Anh online cho trẻ em nông thôn và chấn thương cột sống, Lớp Tiếng Anh online cho người khiếm thị.



**Hình 2.1:** Tình nguyện viên và các em học sinh tại Lớp học Sapa<sup>1</sup>.

<sup>1</sup>Nguồn: <https://www.facebook.com/therainbowclass/posts/pfbid0gp2C3w5ohGQoRqtHYkLJLkkaib735U8iRMyWEnq9Ubdc7DAi5pWqzDyVNfXFDZ6il>

Ngoài ra Lớp học Cầu Vồng có tổ chức các lớp học trực tuyến dành cho các học sinh không thuộc ba tỉnh thành trên và sắp tới cũng có dự định mở rộng hoạt động tại Thành phố Hồ Chí Minh.

Các lớp học của Lớp học Cầu Vồng có thể tổ chức dạy học theo hình thức tập trung (nhiều tình nguyện viên tham gia dạy học nhiều học sinh tại một địa điểm) hoặc hình thức dạy kèm 1-1 (một tình nguyện viên tham gia dạy học cho một học sinh tại nhà riêng của học sinh hoặc trực tuyến)

### **2.1.2 Cơ cấu tổ chức của Lớp học Cầu Vồng**

Cơ cấu tổ chức của Lớp học Cầu Vồng bao gồm: ban quản lý của tổ chức, ban quản lý của từng lớp học, các bạn tình nguyện viên tham gia dạy học và các em học sinh tham gia lớp học.

Với những người thuộc ban quản lý của Lớp học Cầu Vồng, họ sẽ thường xuyên kiểm tra báo cáo, đánh giá điểm của các tình nguyện viên cho mỗi học sinh sau các buổi học định kỳ; tuyên dương, trao học bổng, quà tặng đến những em học sinh có học lực tốt, có tiến bộ trong học tập; tổ chức các buổi từ thiện tặng quà đến cho các em có hoàn cảnh khó khăn.

Lớp học Cầu Vồng bao gồm một hệ thống các lớp học (trực tuyến hoặc trực tiếp) hoạt động một cách độc lập nên mỗi lớp học nhỏ thường có một ban quản lý riêng, chính là các bạn tình nguyện viên được gọi là “Lớp trưởng” và “Lớp phó” (gọi chung là "Cán sự lớp"). Ban cán sự lớp này sẽ có nhiệm vụ quản lý các hoạt động chung của lớp (tuyển tình nguyện viên mới, tổ chức các buổi họp nội bộ trong lớp, tổ chức dạy học cho các em, ...), là cầu nối giữa những em học sinh, các bạn tình nguyện viên tham gia lớp học với ban quản lý tổ chức Lớp học Cầu Vồng.

Các bạn tình nguyện viên tham gia Lớp học Cầu Vồng sẽ tham gia các hoạt động dạy học, các sự kiện chung của lớp. Ngoài việc dạy học, các bạn tình nguyện viên có thể tham gia các hoạt động thiện nguyện khác như trao quà đến các em học sinh có hoàn cảnh khó khăn, thành tích tốt trong học tập,...

Lớp học Cầu Vồng được thành lập với mục đích là chia sẻ tri thức đến với những trẻ em có hoàn cảnh kém may mắn nên đối tượng chính mà Lớp học Cầu Vồng hướng tới là những trẻ em nghèo, trẻ em khuyết tật trí tuệ, trẻ em nông thôn và chấn thương cột sống, người khiếm thị, trẻ em ung thư,... Lớp học Cầu Vồng sẽ đem đến tri thức cho các trẻ em yếu thế trong xã hội với mong muốn lan tỏa những điều tốt đẹp.

### **2.1.3 Các hoạt động quản lý công việc của Lớp học Cầu Vồng hiện nay**

#### **a, Hoạt động quản lý dạy học**

Hầu hết các công việc quản lý dạy học của Lớp học Cầu Vồng được quản lý bằng cách làm tài liệu dựa vào các công cụ phổ biến của Google như Google Docs, Google Sheet, Google Calendar,... hoặc liên lạc thông qua các mạng xã hội như Facebook, Zalo,... Đối với cá nhân từng tình nguyện viên ngoài việc sử dụng các công cụ như trên thì thường sử dụng thủ công các ứng dụng ghi chú trên điện thoại, máy tính để ghi chú công việc.

Dưới đây là một ví dụ về công việc của các Tình nguyện viên khi tham gia giảng dạy ở Lớp học Ong Mật.

Trước khi một tuần học mới bắt đầu, lớp trưởng cần phải thực hiện tạo một cuộc khảo sát trên nhóm chat Facebook, với mục đích lấy danh sách các bạn tình nguyện viên tham gia dạy học trong tuần tới và hẹn hạn chót đăng ký (thường sẽ tạo khảo sát vào thứ bảy và hạn chót là tối chủ nhật). Sau khi hoàn thành cuộc khảo sát, cán sự lớp phải tổng hợp danh sách các tình nguyện viên sẽ tham gia dạy học, chọn ra người phụ trách các buổi học rồi điền lại vào Google Sheet. Cuối cùng là đăng lịch dạy sau khi đã sắp xếp lên nhóm chat để mọi người có thể theo dõi, liên lạc và thông báo cho phụ huynh.

Đối với các lớp học dạy theo hình thức trực tiếp, mỗi tình nguyện viên sẽ được giao phụ trách hướng dẫn một hoặc một vài em học sinh nhất định. Các bạn tình nguyện viên có thể sẽ tạo giáo án riêng tùy vào năng lực của các học sinh mà mình hướng dẫn hoặc sử dụng giáo án cố định của Lớp học Cầu Vồng đã chuẩn bị trước dành cho từng lớp học đặc biệt. Việc giao bài tập về nhà và kiểm tra sẽ diễn ra trực tiếp khi tham gia giảng dạy.

Đối với các lớp học dạy theo hình thức trực tuyến, sau buổi học, các tình nguyện viên phải gửi bài tập lên cho lớp trưởng/lớp phó rồi sau đó lớp trưởng/lớp phó gửi tài liệu học và bài tập về nhà đến các em và phụ huynh thông qua Zalo, phụ huynh chụp bài tập về nhà vào nhóm Zalo để các bạn tình nguyện viên kiểm tra.

Bên cạnh đó, sau mỗi buổi học, các tình nguyện viên cũng cần viết lại báo cáo và nộp lại cho cán sự lớp về những công việc đã làm trong buổi học và để lại các nhận xét (ít nhất khoảng 3 dòng) cho học sinh. Cứ mỗi tháng thì quản trị viên của tổ chức sẽ xem tổng hợp những nhận xét đó để đánh giá tiến độ giảng dạy của các lớp học.

Trong thời điểm tình hình dịch Covid-19 diễn ra phức tạp, để đảm bảo an toàn phòng chống dịch, hầu hết các lớp học tại Lớp học Cầu Vồng đều được chuyển từ

hình thức dạy-học trực tiếp sang dạy học trực tuyến. Cũng chính vì thế nên việc quản lý các lớp học cũng trực tuyến hoàn toàn.



**Hình 2.2:** LHCV trở lại học trực tiếp sau những ngày học trực tuyến vì dịch Covid-19<sup>2</sup>

### b, Hoạt động tuyển tình nguyện viên

Lớp học Cầu Vồng thường đăng các bài tuyển tình nguyện viên lên trang fanpage chính thức của tổ chức trên Facebook<sup>3</sup>. Các bài tuyển tình nguyện viên thường có chung một mẫu. Những người có mong muốn tham gia trở thành tình nguyện viên của Lớp học Cầu Vồng sẽ phải gửi CV của bản thân và câu trả lời cho những câu hỏi được đưa trên bài đăng về một địa chỉ email của tổ chức. Ngoài ra đối với những bạn ứng tuyển làm tình nguyện viên dạy tiếng Anh sẽ cần phải gửi kèm thêm một bản ghi âm giới thiệu bản thân bằng tiếng Anh.

Cách thức phỏng vấn được diễn ra bằng hình thức trực tuyến. Do các lớp học hoạt động độc lập và riêng lẻ nên công việc chọn và duyệt hồ sơ đăng ký được ban nhân sự phụ trách. Sau khi duyệt xong, ban nhân sự sẽ chuyển tiếp các hồ sơ đó cho ban quản lý của lớp học liên quan để liên hệ và lên lịch phỏng vấn. Ban quản lý của Lớp học Cầu Vồng không trực tiếp tham gia quá trình phỏng vấn mà chỉ theo

<sup>2</sup>Nguồn: <https://www.facebook.com/therainbowclass/photos/a.225545847851370/1287147145024563/>

<sup>3</sup>Link fanpage của Lớp học Cầu Vồng: <https://www.facebook.com/therainbowclass>

dõi, xem kết quả. Chính vì vậy, việc gửi email hồ sơ đăng ký về chung một email của tổ chức sẽ dẫn đến mất thời gian và bất tiện cho các tình nguyện viên quản lý lớp học.

Trên đây là một số khảo sát<sup>4</sup> về đối tượng mà Lớp học Cầu Vồng hướng tới, cách quản lý công việc hiện nay của lớp, và một số vấn đề xung quanh mà em đã thu thập được. Số lượng người tham gia khảo sát trên là các bạn tình nguyện viên đang tham gia giảng dạy tại Lớp học Cầu Vồng (chủ yếu là các bạn tình nguyện viên tham gia quản lý lớp). Tuy số lượng người tham gia khảo sát không quá nhiều, nhưng cũng giúp hiểu được phần nào các vấn đề thực tế còn tồn tại khi tổ chức các hoạt động tại Lớp học Cầu Vồng.

## 2.2 Khảo sát các chức năng cần thiết cho hệ thống

Dựa trên các khảo sát sơ bộ về các hoạt động hiện tại của Lớp học Cầu Vồng, và khảo sát các chức năng cần thiết, em đã phân tích yêu cầu về quản lý nhân sự và quản lý giảng dạy của tổ chức, từ đó đưa ra các giả thiết về một số chức năng chính và thực hiện khảo sát mức độ ưu tiên của từng chức năng đối với quản lý của Lớp học Cầu Vồng dựa theo chức vụ hiện có của người dùng.

- **Đối với quản trị viên:**

- Theo dõi được lịch dạy học, lịch phỏng vấn của từng lớp thuộc tổ chức.
- Theo dõi được nội dung bài học, báo cáo sau mỗi buổi học của từng lớp.
- Theo dõi được đánh giá, nhận xét của từng tình nguyện viên tham gia dạy học cho các học sinh theo từng buổi.
- Thêm, cập nhật các thông tin, danh sách các tình nguyện viên/học sinh tham gia tổ chức.
- Chỉnh sửa quyền cho các tình nguyện viên tham gia dạy học trở thành quản lý lớp.
- Xem xét, duyệt hồ sơ, theo dõi lịch phỏng vấn đối với những người có mong muốn tham gia làm tình nguyện viên.
- Tùy chỉnh những câu hỏi trong form đăng ký làm Tình nguyện viên.

- **Đối với quản lý lớp:**

- Theo dõi, thay đổi được lịch dạy, nội dung bài học của lớp mà họ phụ trách.
- Theo dõi được các báo cáo, đánh giá, nhận xét của những tình nguyện

---

<sup>4</sup>Các kết quả khảo sát: <https://sal.vn/oyNef7>, <https://sal.vn/rMSW0d>

viên tham gia dạy học cho học sinh tham gia lớp mà họ phụ trách.

- Thêm, cập nhật các thông tin, danh sách tình nguyện viên/học sinh tham gia lớp mà họ phụ trách.
- Xem xét, duyệt hồ sơ, theo dõi lịch phỏng vấn đối với những người có mong muốn tham gia làm tình nguyện viên tại lớp mà họ phụ trách.
- Xem xét, duyệt hồ sơ, theo dõi lịch phỏng vấn đối với những người có mong muốn tham gia làm tình nguyện viên tại lớp mà họ phụ trách.

• **Đối với tình nguyện viên:**

- Theo dõi lịch học, nội dung bài học và đăng ký tham gia dạy học của lớp mà họ đang tham gia.
- Thực hiện nộp báo cáo, đánh giá, nhận xét học sinh định kỳ sau mỗi buổi học diễn ra.
- Xem thông tin các học sinh, tình nguyện viên thuộc lớp mà họ đang tham gia

• **Đối với học sinh:**

- Theo dõi lịch học của bản thân.
- Theo dõi thông tin các bài giảng.
- Theo dõi các đánh giá của tình nguyện viên sau mỗi buổi học.

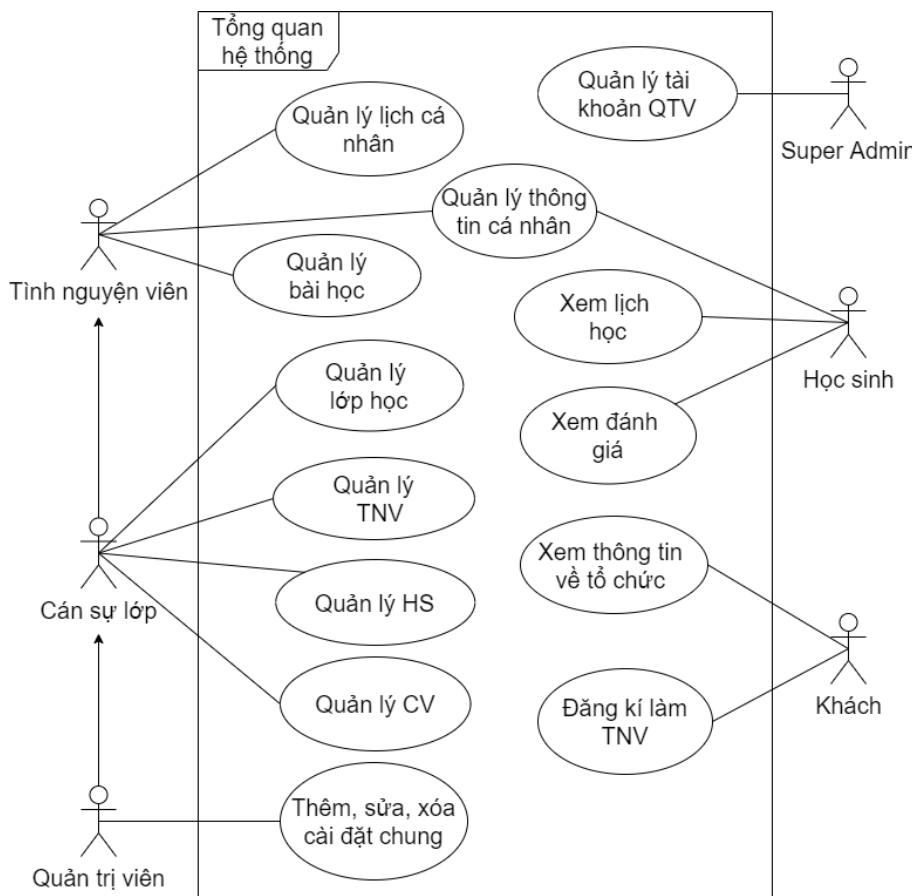
• **Đối với khách:**

- Xem được các thông tin cơ bản, công khai về tổ chức.
- Đăng ký làm tình nguyện viên.

### 2.3 Tổng quan chức năng

#### 2.3.1 Biểu đồ use case tổng quát

Biểu đồ use case tổng quát của hệ thống được biểu diễn như hình 2.3

**Hình 2.3:** Biểu đồ use case tổng quát

### Mô tả biểu đồ use case tổng quát

**Tác nhân Super Admin:** Quản lý các tài khoản quản trị viên bằng cách đưa một tài khoản tình nguyện viên trở thành một tài khoản quản trị viên hoặc đưa một tài khoản quản trị viên trở thành một tài khoản tình nguyện viên. Hệ thống sẽ có một tài khoản Super Admin duy nhất được cung cấp bởi nhà phát triển.

**Tác nhân Quản trị viên:** Tất cả các chức năng của quản trị viên phải đăng nhập vào hệ thống mới có thể thực hiện được.

Tên use case	Mô tả tóm tắt
Quản lý tình nguyện viên	Quản trị viên có thể thêm, sửa, xóa thông tin tất cả tình nguyện viên
Quản lý học sinh	Quản trị viên có thể thêm, sửa, xóa thông tin tất cả học sinh
Quản lý CV	Quản trị viên có thể xem, duyệt, lên lịch phỏng vấn cho các CV; thêm, sửa, xóa các câu hỏi cho form đăng ký làm tình nguyện viên

Quản lý lớp học	Quản trị viên có thể thêm, sửa, xóa thông tin tất cả các lớp học; theo dõi lịch giảng dạy/lịch phỏng vấn của tất cả các lớp; chỉnh sửa cán sự lớp cho tất cả các lớp
Quản lý thông tin cá nhân	Quản trị viên có thể sửa, xóa các thông tin cá nhân, avatar; thay đổi password cho tài khoản của mình
Thêm, sửa, xóa các cài đặt chung	Quản trị viên có thể thêm, sửa, xóa các thông tin chung như: phân loại học sinh, môn học, khối lớp

**Bảng 2.1:** Phân tích use case với tác nhân là quản trị viên

**Tác nhân Cán sự lớp:** Tất cả các chức năng của Cán sự lớp thì cần phải đăng nhập vào hệ thống mới có thể thực hiện được

Tên use case	Mô tả tóm tắt
Quản lý tình nguyện viên	Cán sự lớp có thể thêm, sửa, xóa thông tin tất cả các tình nguyện viên thuộc lớp mà họ quản lý
Quản lý học sinh	Cán sự lớp có thể thêm, sửa, xóa thông tin tất cả các học sinh thuộc lớp mà họ quản lý
Quản lý CV	Cán sự lớp có thể xem, duyệt, lên lịch phỏng vấn cho tất cả các CV thuộc lớp mà họ quản lý
Quản lý lớp học	Cán sự lớp có thể thêm, sửa, xóa thông tin lớp học mà mình đang quản lý; theo dõi lịch giảng dạy/lịch phỏng vấn của tất cả các lớp mà mình đang quản lý; chỉnh sửa cán sự lớp của lớp mà họ phụ trách Với lớp học theo hình thức dạy kèm 1-1: Cán sự lớp có thể sắp xếp các cặp tình nguyện viên với học sinh tạo thành cặp dạy kèm
Quản lý bài học	Với lớp học theo hình thức dạy tập trung: Cán sự lớp có thể thêm, sửa, xóa các bài học cho lớp học mà mình đang quản lý; tạo báo cáo sau mỗi bài học Với lớp học theo hình thức dạy kèm 1-1: Cán sự lớp có thể thêm, sửa, xóa các bài học cho học sinh họ đang được ghép cặp; tạo báo cáo sau mỗi bài học
Quản lý thông tin cá nhân	Cán sự lớp có thể sửa, xóa các thông tin cá nhân, avatar; thay đổi password cho tài khoản của mình

Quản lý lịch cá nhân	Cán sự lớp có thể lên lịch phỏng vấn; đăng kí dạy học cho các buổi học
----------------------	--

**Bảng 2.2:** Phân tích use case với tác nhân là cán sự lớp

**Tác nhân tình nguyện viên:** Tất cả các chức năng của tình nguyện viên thì cần phải đăng nhập vào hệ thống mới có thể thực hiện được

Tên use case	Mô tả tóm tắt
Quản lý bài học	Với lớp học theo hình thức dạy tập trung: tình nguyện viên không thể tự tạo bài học; tạo báo cáo sau mỗi bài học mình đăng kí giảng dạy Với lớp học theo hình thức dạy kèm 1-1: tình nguyện viên có thể thêm, sửa, xóa bài học; tạo báo cáo sau mỗi bài học
Quản lý lịch cá nhân	Với lớp học theo hình thức dạy tập trung: tình nguyện viên có thể đăng kí tham gia bài giảng theo nguyện vọng của mình. Với lớp học theo hình thức dạy kèm 1-1: tình nguyện viên có thể sắp xếp lịch bài học cho phù hợp với lịch của bản thân và học sinh day kèm
Quản lý thông tin cá nhân	Tình nguyện viên có thể sửa, xóa các thông tin cá nhân, avatar; thay đổi password cho tài khoản của mình

**Bảng 2.3:** Phân tích use case với tác nhân là tình nguyện viên

**Tác nhân học sinh:** Tất cả các chức năng của Học sinh thì cần phải đăng nhập vào hệ thống mới có thể thực hiện được

Tên use case	Mô tả tóm tắt
Xem đánh giá	Học sinh có thể xem được đánh giá của tình nguyện viên cho mình sau mỗi buổi học

Xem lịch học	Với lớp học theo hình thức dạy tập trung: Học sinh có thể xem lịch học của lớp Với lớp học theo hình thức dạy kèm 1: Học sinh có thể xem lịch học của bản thân mình
Quản lý thông tin cá nhân	Học sinh có thể sửa, xóa các thông tin cá nhân, avatar; thay đổi password cho tài khoản của mình

**Bảng 2.4:** Phân tích use case với tác nhân là Học sinh

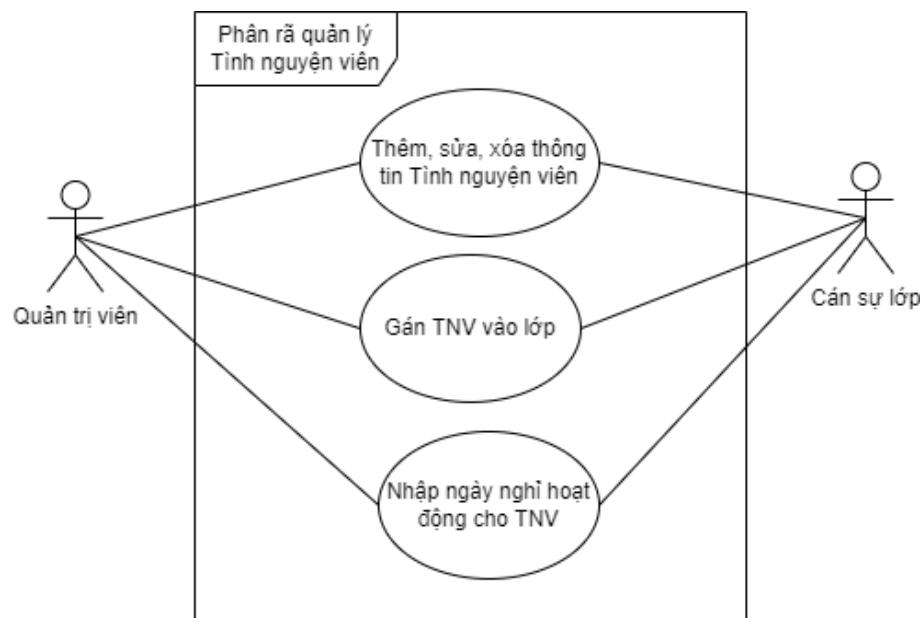
**Tác nhân khách:** Tất cả các chức năng của Khách thì không cần phải đăng nhập vào hệ thống vẫn có thể thực hiện được.

Tên use case	Mô tả tóm tắt
Xem thông tin về tổ chức	Khách có thể xem thông tin công khai của LHCV
Đăng ký làm tình nguyện viên	Khách có thể đăng ký làm tình nguyện viên

**Bảng 2.5:** Phân tích use case với tác nhân là Khách

### 2.3.2 Biểu đồ use case phân rã

#### Biểu đồ use case phân rã Quản lý tình nguyện viên

**Hình 2.4:** Biểu đồ use case phân rã quản lý Tình nguyện viên

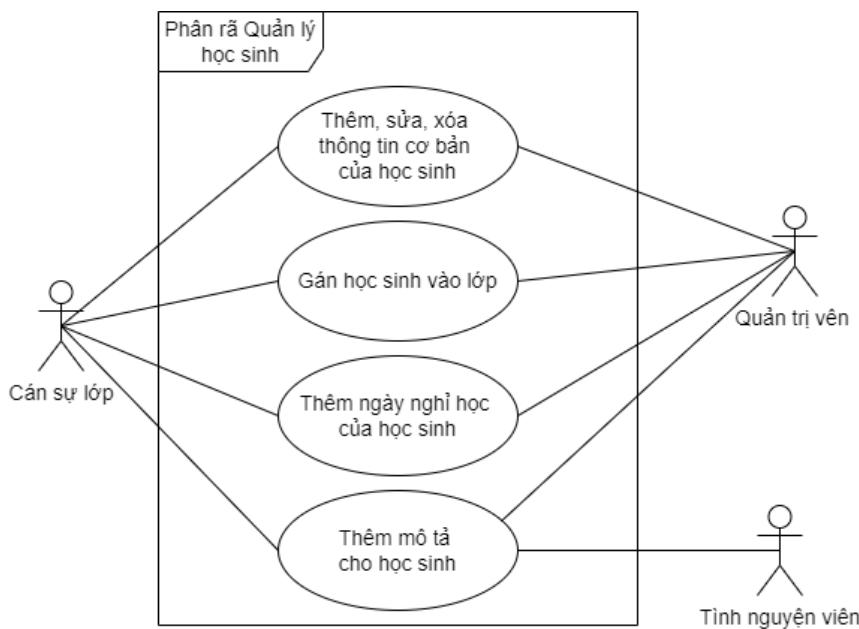
Tác nhân: Quản trị viên, cán sự lớp

Mô tả use case:

Tên use case	Mô tả tóm tắt
Thêm, sửa, xóa tình nguyện viên	<ul style="list-style-type: none"> <li>+ Sau khi thêm mới một tình nguyện viên, hệ thống tự động gửi mail đến cho email của tình nguyện viên đó để kích hoạt tài khoản và đăng nhập vào hệ thống.</li> <li>+ Xem, sửa thông tin của tình nguyện viên.</li> <li>+ Xóa thông tin tình nguyện viên (Không thể xóa trong trường hợp tình nguyện viên là quản trị viên)</li> </ul>
Gán tình nguyện viên vào lớp	<ul style="list-style-type: none"> <li>+ Hệ thống đưa ra danh sách lớp học, tình nguyện viên được gán vào một lớp duy nhất trong một thời điểm.</li> </ul>
Nhập ngày nghỉ hoạt động cho tình nguyện viên	<ul style="list-style-type: none"> <li>+ Quản trị viên hoặc cán sự lớp nhập ngày nghỉ hoạt động cho tình nguyện viên.</li> <li>+ Sau khi qua ngày nghỉ hoạt động thì trạng thái tài khoản của tình nguyện viên trở thành inactive, tình nguyện viên không thể đăng nhập vào hệ thống.</li> <li>+ Các dữ liệu liên quan đến tình nguyện viên vẫn còn</li> </ul>

**Bảng 2.6:** Mô tả use case phân rã Quản lý tình nguyện viên

### Biểu đồ use case phân rã Quản lý học sinh



**Hình 2.5:** Biểu đồ use case phân rã Quản lý học sinh

Tác nhân: Quản trị viên, cán sự lớp, tình nguyện viên

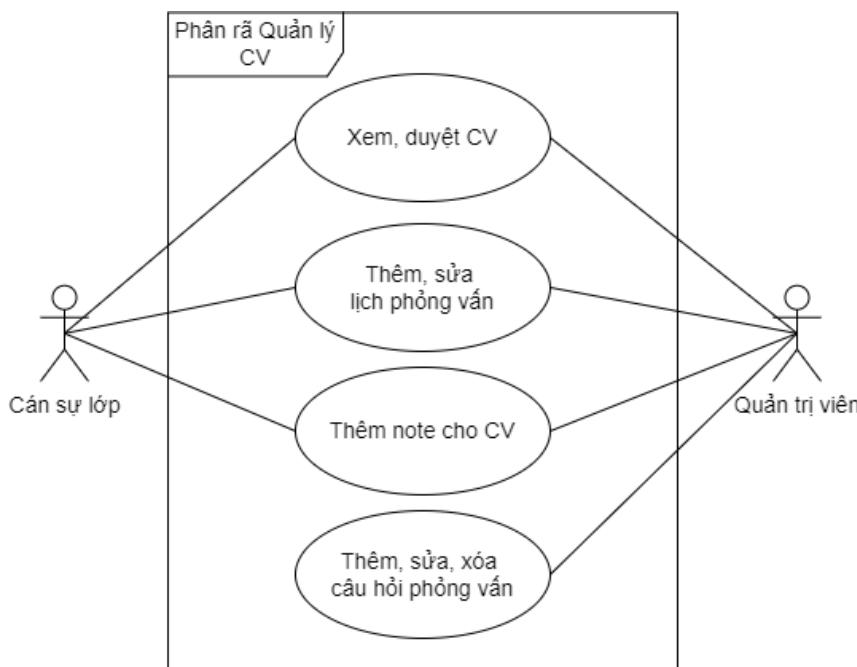
Mô tả use case:

Tên use case	Mô tả tóm tắt
Thêm, sửa, xóa thông tin cơ bản của học sinh	<ul style="list-style-type: none"> <li>+ Sau khi thêm mới một Học sinh, hệ thống tự động gửi mail đến cho email của Học sinh đó để kích hoạt tài khoản và đăng nhập vào hệ thống.</li> <li>+ Xem, sửa thông tin của học sinh.</li> <li>+ Xóa thông tin học sinh (Không thể xóa trong trường hợp học sinh đã tham gia học và có thành tích)</li> </ul>
Gán học sinh vào lớp	<ul style="list-style-type: none"> <li>+ Hệ thống đưa ra danh sách các lớp học, học sinh được gán vào một lớp duy nhất trong một thời điểm.</li> </ul>
Nhập ngày nghỉ học cho học sinh	<ul style="list-style-type: none"> <li>+ Quản trị viên hoặc cán sự lớp nhập ngày nghỉ học cho học sinh.</li> <li>+ Sau khi qua ngày nghỉ hoạt động thì trạng thái tài khoản của học sinh trở thành inactive, không thể đăng nhập vào hệ thống.</li> <li>+ Các dữ liệu liên quan đến học sinh vẫn còn.</li> </ul>
Thêm mô tả cho học sinh	Vì mỗi em học sinh có tính cách và hoàn cảnh khác nhau nên các tác nhân có thể thêm mô tả cho học sinh để sau này những tình nguyện viên mới vào có thể dễ tiếp cận và hiểu hơn về tính cách, hoàn cảnh của các em học sinh.

**Bảng 2.7:** Mô tả use case phân rã Quản lý học sinh

### Biểu đồ use case phân rã Quản lý CV

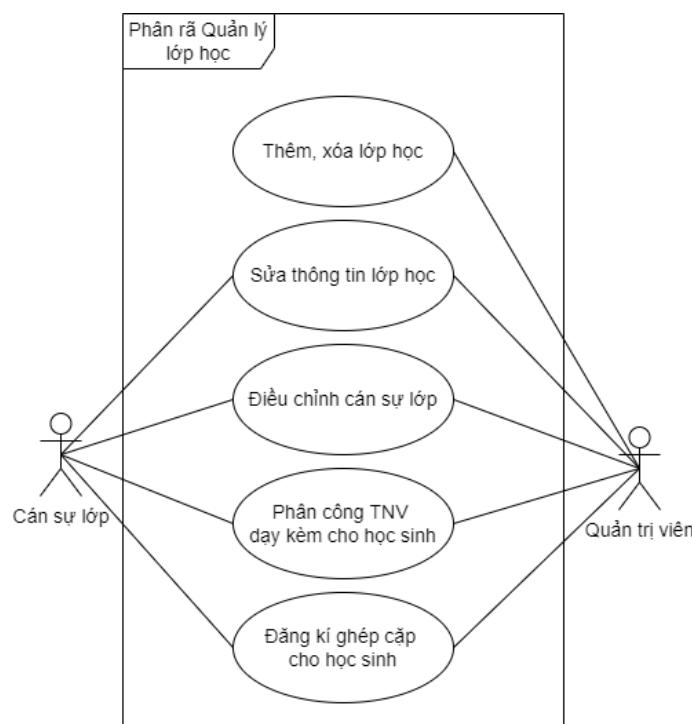
Tác nhân: Quản trị viên, cán sự lớp

**Hình 2.6:** Biểu đồ use case phân rã Quản lý CV

Mô tả use case:

Tên use case	Mô tả tóm tắt
Xem, duyệt CV	<ul style="list-style-type: none"> <li>+ Xem CV. Trạng thái của những CV chưa được duyệt qua là “Chưa duyệt”</li> <li>+ Sau khi xem CV, quản trị viên hoặc cán sự lớp có thể duyệt CV theo 3 trạng thái: “Chờ phỏng vấn”, “Đỗ”, “Trượt”</li> <li>+ Chỉ có thể chuyển CV từ trạng thái “Chưa duyệt” sang trạng thái “Chờ phỏng vấn” hoặc “Trượt”</li> <li>+ Chỉ có thể chuyển CV từ trạng thái “Chờ phỏng vấn” sang trạng thái “Trượt” hoặc “Đỗ”</li> <li>+ CV đang ở trạng thái “Trượt” hoặc “Đỗ” không thể chuyển trạng thái</li> <li>+ Hệ thống sẽ tự động gửi mail đến người đăng ký làm tình nguyện viên (người gửi CV) với nội dung tương ứng với 3 trạng thái trên.</li> <li>+ Nếu CV được duyệt, chuyển qua trạng thái đỗ thì hệ thống tự động thêm tình nguyện viên mới với các thông tin đã được đăng ký.</li> </ul>

Thêm, sửa lịch phỏng vấn	<ul style="list-style-type: none"> <li>+ Nếu CV được chuyển về trạng thái “Chờ phỏng vấn” thì quản trị viên hoặc cán sự lớp sẽ có thể thêm/ sửa lịch phỏng vấn online với các thông tin về thời gian, link phỏng vấn, người phụ trách.</li> <li>+ Người đăng ký làm tình nguyện viên sẽ được nhận email nếu có sự thay đổi về lịch phỏng vấn</li> </ul>
Thêm note cho CV	<ul style="list-style-type: none"> <li>+ Thêm, sửa, xóa note về các vấn đề liên quan đến buổi phỏng vấn.</li> </ul>
Thêm, sửa, xóa câu hỏi phỏng vấn	<ul style="list-style-type: none"> <li>+ Quản trị viên thêm, sửa, xóa các câu hỏi xuất hiện trong form đăng ký làm tình nguyện viên</li> </ul>

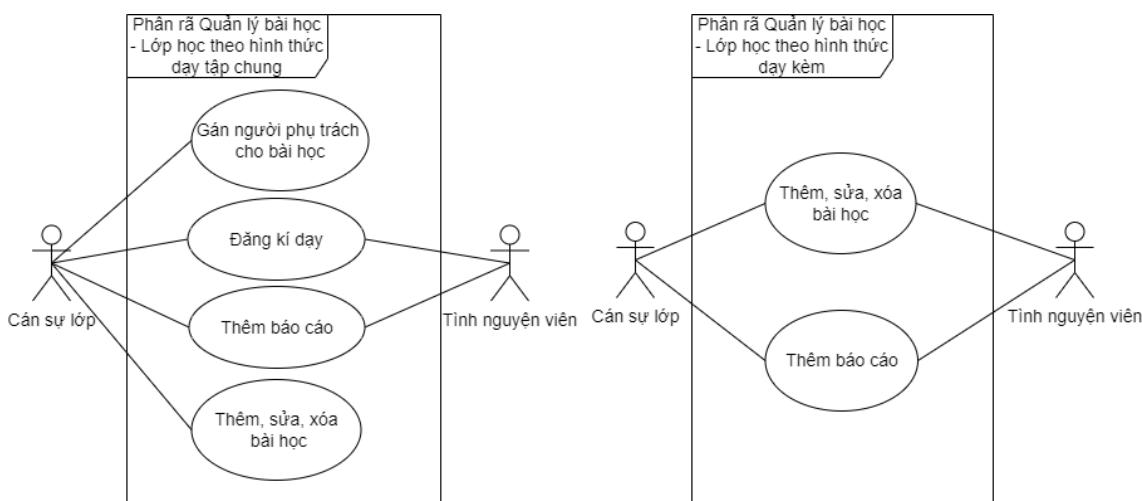
**Bảng 2.8:** Mô tả use case phân rã Quản lý CV**Biểu đồ use case phân rã Quản lý lớp học****Hình 2.7:** Biểu đồ use case phân rã Quản lý lớp học

Tác nhân: Quản trị viên, cán sự lớp

Mô tả use case:

Tên use case	Mô tả tóm tắt
Thêm, xóa lớp học	+ Chỉ Quản trị viên mới có thể thêm hoặc xóa lớp học

Sửa thông tin lớp học	+ Quản trị viên và cán sự lớp có thể chỉnh sửa thông tin lớp học.
Đăng ký ghép cặp cho học sinh	+ Với lớp học dạy theo hình thức 1-1, cần đăng ký ghép cặp cho học sinh bao gồm các thông tin như: môn học, khối lớp, số buổi một tuần, hình thức dạy,...
Phân công tình nguyện viên dạy kèm cho học sinh	+ Gán một tình nguyện viên dạy kèm cho học sinh. Mỗi tình nguyện viên chỉ dạy kèm cho một học sinh.

**Bảng 2.9:** Mô tả use case phân rã Quản lý lớp học**Biểu đồ use case phân rã Quản lý bài học****Hình 2.8:** Biểu đồ use case phân rã Quản lý bài học theo hai hình thức dạy

Tác nhân: Cán sự lớp, tình nguyện viên

Mô tả use case:

Tên use case	Mô tả tóm tắt
Thêm, sửa, xóa bài học	+ Với lớp có hình thức dạy tập trung: chỉ có Cán sự lớp mới có thể thêm, sửa, xóa bài học + Với lớp có hình thức dạy kèm 1-1: Chỉ có tình nguyện viên đã được xếp cặp dạy kèm với học sinh mới có thể thêm, sửa, xóa bài học
Đăng ký dạy (Chỉ lớp dạy theo hình thức tập trung)	+ Với lớp có hình thức dạy tập trung: Các tình nguyện viên và cán sự có thể đăng ký dạy học theo nguyện vọng

Gán người phụ trách (Chỉ lớp dạy theo hình thức tập trung)	+ Với lớp có hình thức dạy tập trung: Cán sự lớp chọn ra một tình nguyện viên đã đăng ký dạy làm người phụ trách buổi học
Thêm báo cáo	+ Các tình nguyện viên sau khi tham gia dạy học thì sau buổi học phải thêm báo cáo nhận xét học sinh trong buổi học và chấm điểm cho học sinh trên thang điểm 10.

**Bảng 2.10:** Mô tả use case phân rã Quản lý bài học**Biểu đồ use case phân rã Quản lý lịch cá nhân****Hình 2.9:** Biểu đồ use case phân rã Quản lý lịch cá nhân

Tác nhân: Cán sự lớp, tình nguyện viên

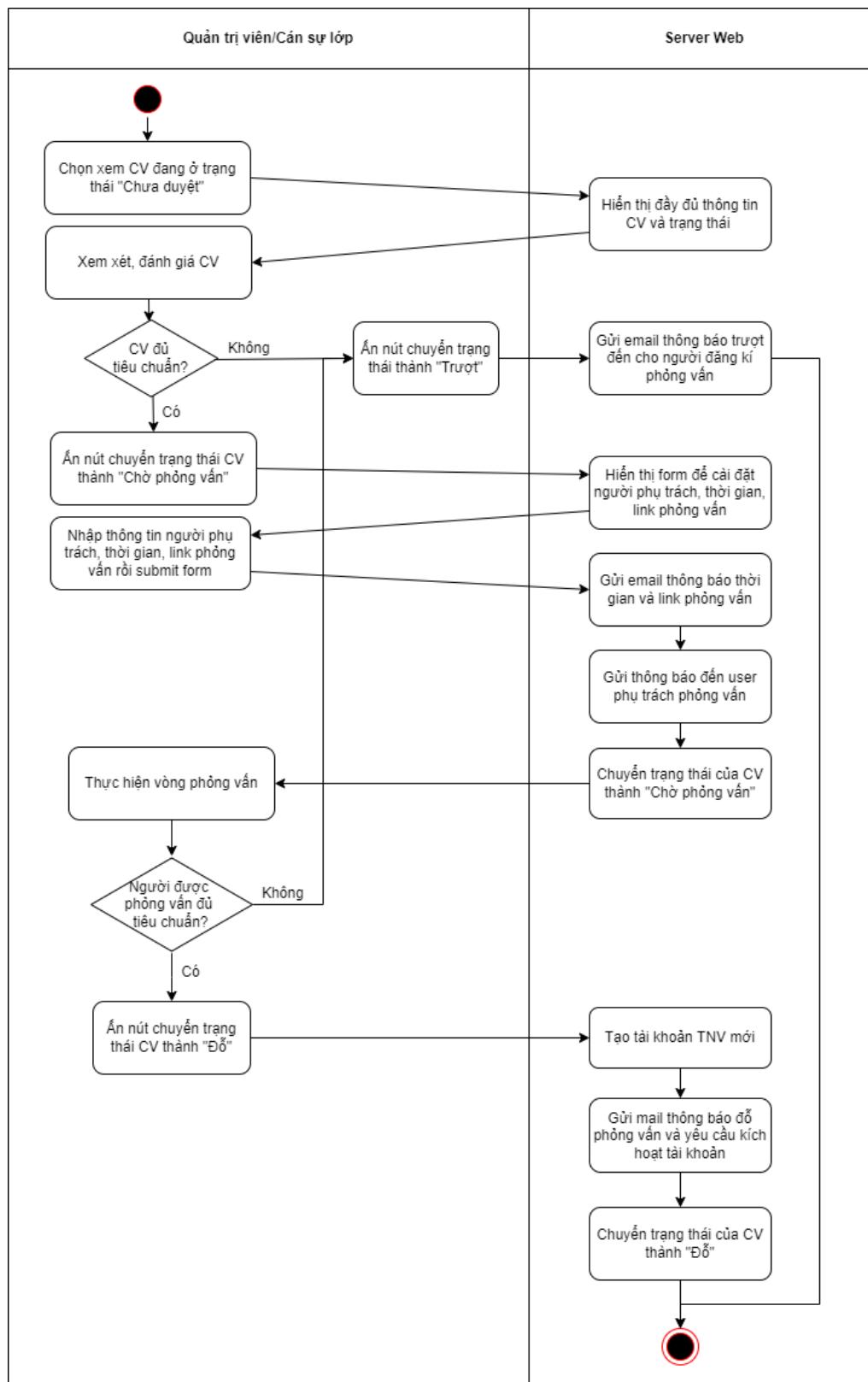
Mô tả use case:

Tên use case	Mô tả tóm tắt
Thêm, sửa, xóa lịch dạy học	Tình nguyện viên và cán sự lớp có thể thêm, sửa, xóa hoặc đăng ký lịch dạy phù hợp với nguyện vọng
Thêm, sửa lịch phỏng vấn	Cán sự lớp có thể thêm, sửa lịch phỏng vấn để phù hợp với lịch cá nhân của bản thân và người tham gia phỏng vấn

**Bảng 2.11:** Mô tả use case phân rã Quản lý Lịch cá nhân**2.3.3 Quy trình nghiệp vụ****Quy trình nghiệp vụ “Duyệt CV và phỏng vấn”**

Tác nhân: Quản trị viên/Cán sự lớp

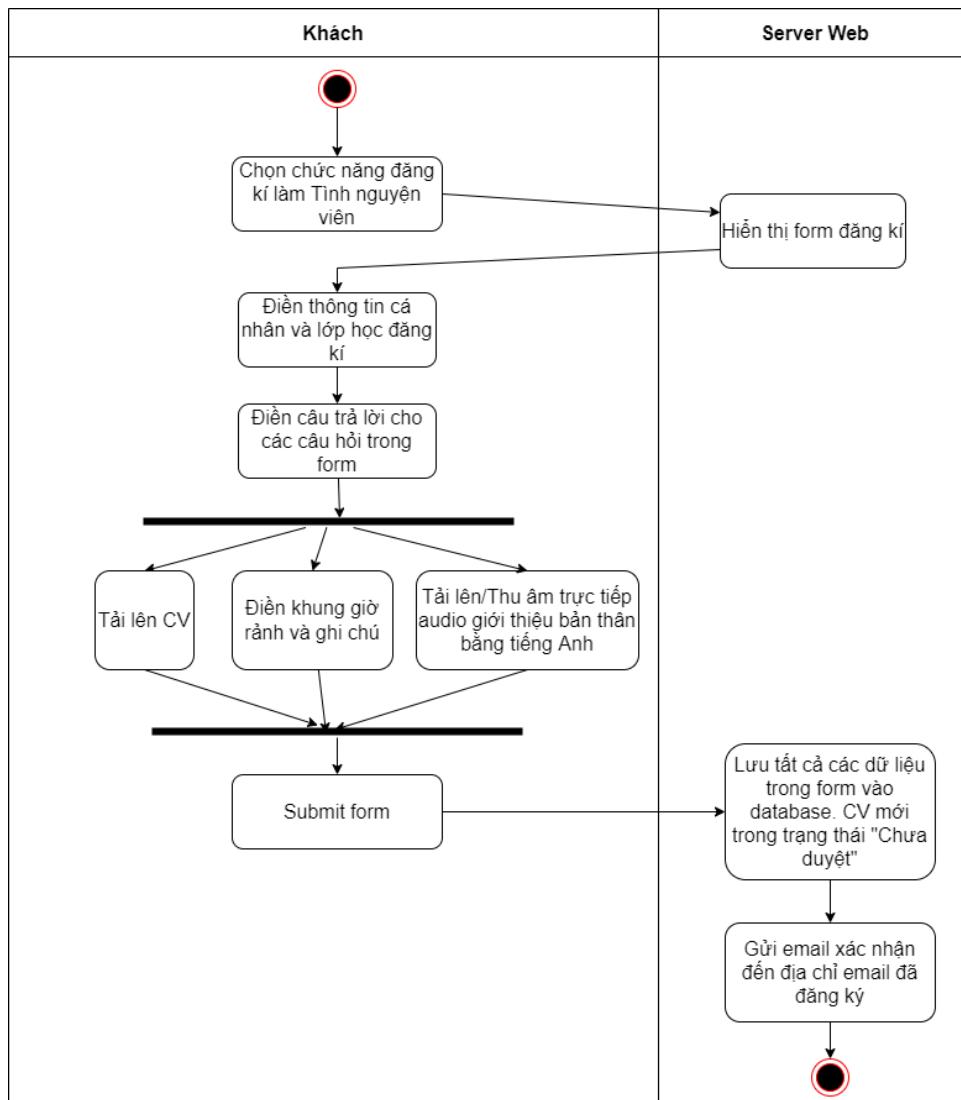
Mô tả: Khi quản trị viên/Cán sự lớp duyệt CV, thay đổi trạng thái của CV thì ứng với mỗi trạng thái khác nhau, hệ thống phải xử lý theo các hướng khác nhau.



**Hình 2.10:** Quy trình nghiệp vụ duyệt CV

## 2.4 Đặc tả chức năng

### 2.4.1 Đặc tả use case “Đăng ký làm tình nguyện viên”

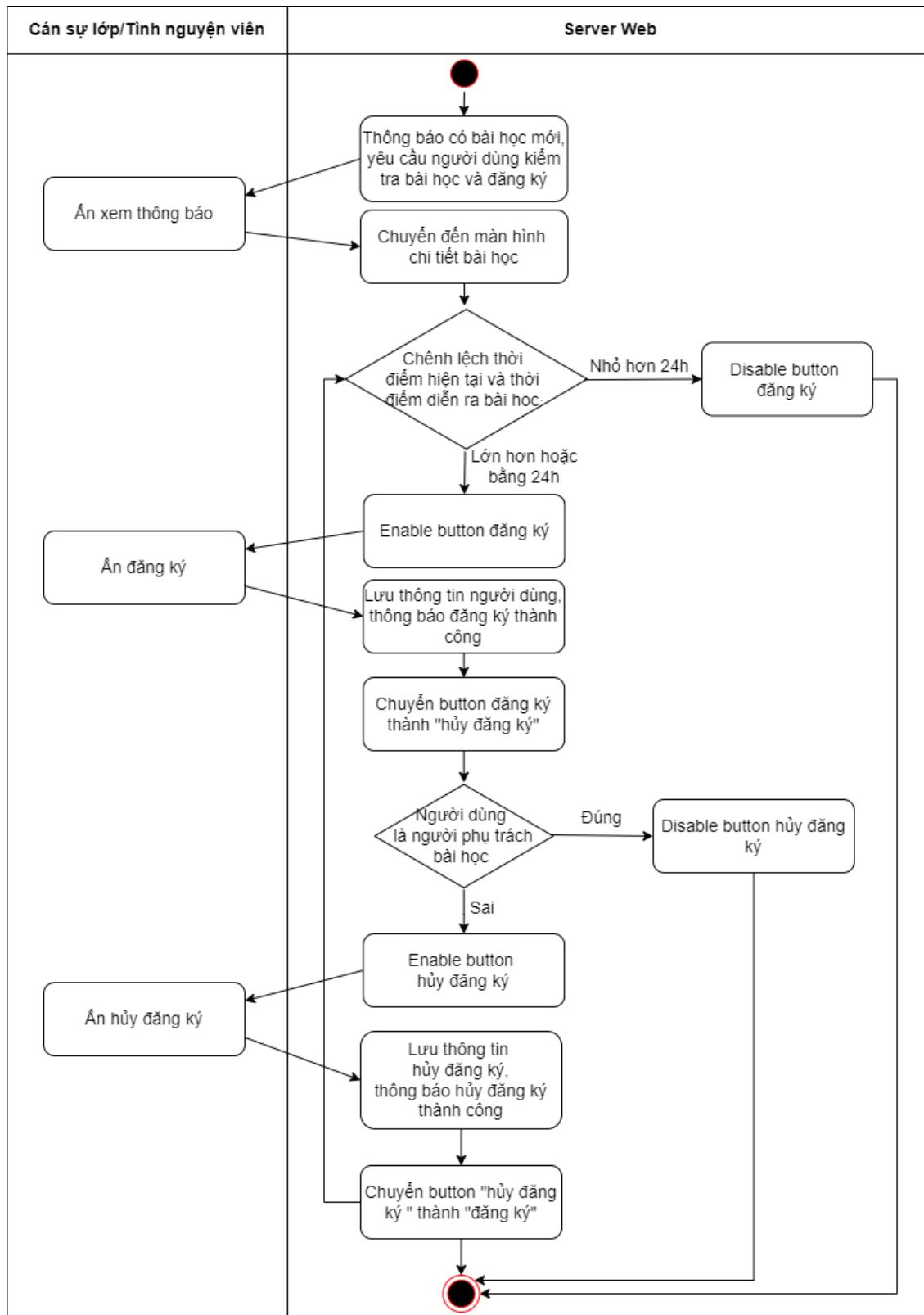


**Hình 2.11:** Sơ đồ luồng đặc tả use case đăng ký làm TNV

Tác nhân: Khách	Tiền điều kiện: Không phải đăng nhập
<p>Mô tả tóm tắt: Người khách đăng ký thông tin cá nhân của mình (họ tên, email, số điện thoại, lớp học đăng ký), trả lời các câu hỏi được quản trị viên cài đặt sẵn trong hệ thống, tải CV của bản thân, tùy chọn ghi âm/tải lên file audio giới thiệu bản thân bằng tiếng Anh và điền thời gian rảnh vào form đăng ký làm tình nguyện viên.</p>	

**Bảng 2.12:** Phân tích use case đăng ký làm TNV với tác nhân là Khách

### 2.4.2 Đặc tả use case ‘Đăng ký dạy học với lớp theo hình thức dạy tập trung’



**Hình 2.12:** Sơ đồ luồng đặc tả use case đăng ký dạy đối với lớp theo hình thức dạy tập trung

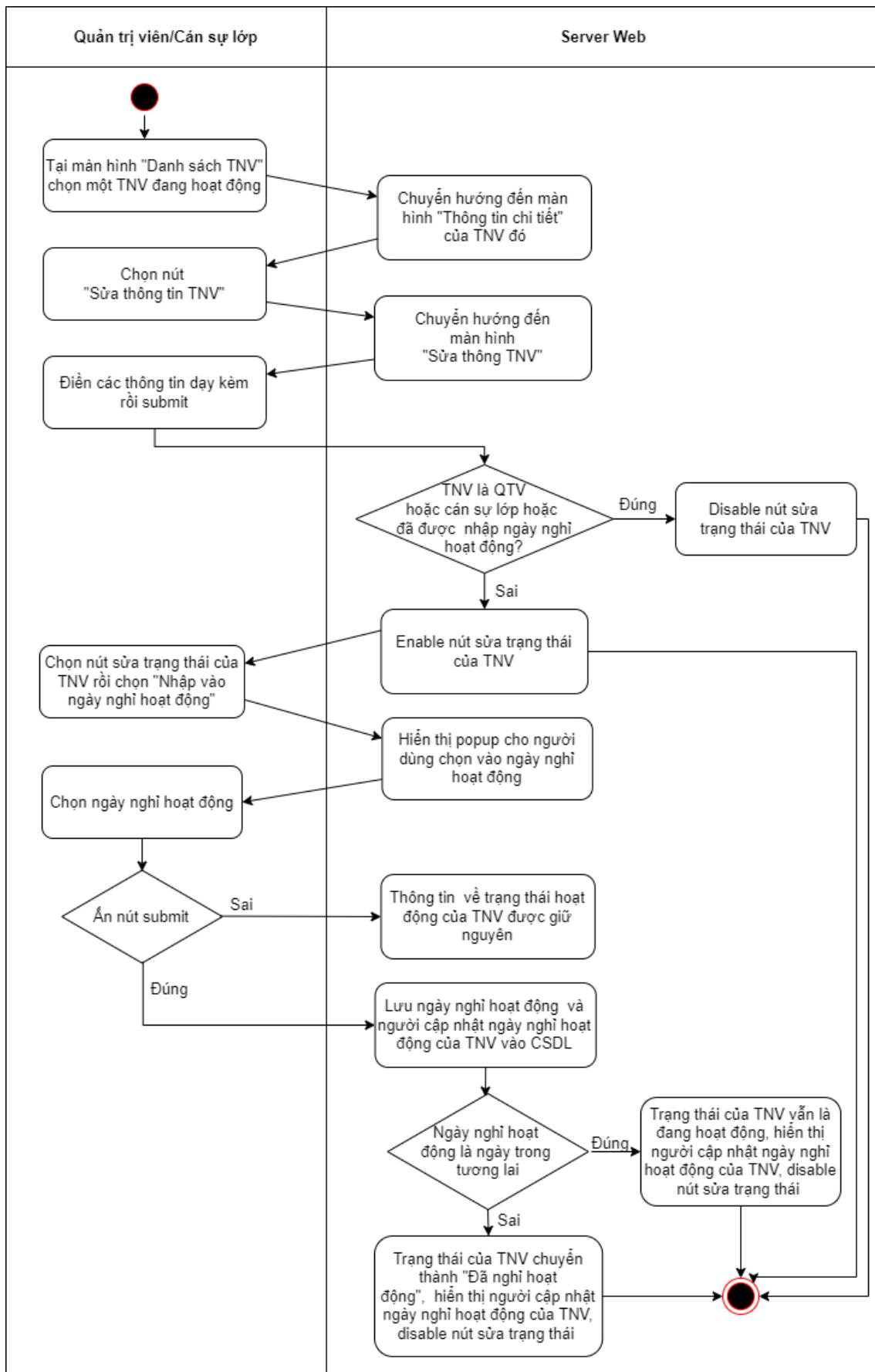
Tác nhân: Cán sự lớp/Tình nguyện viên	Tiền điều kiện: Phải đăng nhập
Mô tả tóm tắt: Sau khi một bài học mới được tạo, thông báo sẽ gửi về cho tình nguyện viên để tình nguyện viên đăng ký dạy tùy theo nguyện vọng của mình. Tình nguyện viên chỉ có thể đăng ký dạy hoặc hủy đăng ký cho đến trước khi bài học diễn ra trước 1 ngày. Nếu tình nguyện viên được chọn làm người phụ trách cho buổi học thì không thể hủy đăng ký dạy học	

**Bảng 2.13:** Phân tích use case Đăng ký dạy đối với lớp theo hình thức dạy tập trung

#### 2.4.3 Đặc tả use case “Nhập ngày nghỉ hoạt động cho TNV”

Tác nhân: Quản trị viên, cán sự lớp	Tiền điều kiện: Phải đăng nhập
<p>Mô tả tóm tắt:</p> <ul style="list-style-type: none"> <li>- Quản trị viên hoặc cán sự lớp có thể nhập vào ngày nghỉ hoạt động cho tình nguyện viên thông qua màn hình "chi tiết tình nguyện viên".</li> <li>- Nếu tình nguyện viên có chức vụ hiện tại là quản trị viên hoặc cán sự lớp thì nút thay đổi trạng thái sẽ bị disable.</li> <li>- Một tình nguyện viên sau khi đã được nhập vào ngày nghỉ hoạt động thì tại màn hình chi tiết sẽ hiển thị thông tin ngày nghỉ hoạt động, thông tin người cập nhật ngày nghỉ hoạt động và disable nút thay đổi trạng thái.</li> <li>- Bên cạnh đó, nếu ngày nghỉ hoạt động là một thời điểm trong tương lai thì trạng thái của tình nguyện viên đó là "Đã nghỉ hoạt động". Ngược lại, trạng thái của tình nguyện viên là "Đang hoạt động".</li> </ul>	

**Bảng 2.14:** Phân tích use case "Nhập ngày nghỉ hoạt động cho TNV"



**Hình 2.13:** Sơ đồ luồng đặc tả use case "Nhập ngày nghỉ hoạt động cho TNV"

## 2.5 Yêu cầu phi chức năng

Bên cạnh các yêu cầu về chức năng, hệ thống còn đáp ứng các yêu cầu phi chức năng như sau:

- Tính dễ dùng: Hệ thống có giao diện sử dụng, thân thiện với người dùng, có sự thống nhất về màu sắc, vị trí, kích thước của các phần tử trong một trang và giữa các trang với nhau; có hỗ trợ responsive màn hình mobile đối với trang web hỗ trợ đăng ký làm tình nguyện viên
- Tính riêng tư, an toàn: hệ thống có xác thực người dùng với từng chức năng của họ cả phía Backend và Fronend
- Tính dễ bảo trì: hệ thống được chia thành các module và component cụ thể nên việc bảo trì trở nên dễ dàng hơn.

Trong chương này, em đã tìm hiểu và rút ra một số yêu cầu cần thiết cho hệ thống. Các chức năng của hệ thống cũng được biểu diễn qua biểu đồ use case và quy trình nghiệp vụ. Tiếp theo, ở chương 3 em sẽ trình bày về công nghệ sử dụng và các thuật toán áp dụng trong hệ thống.

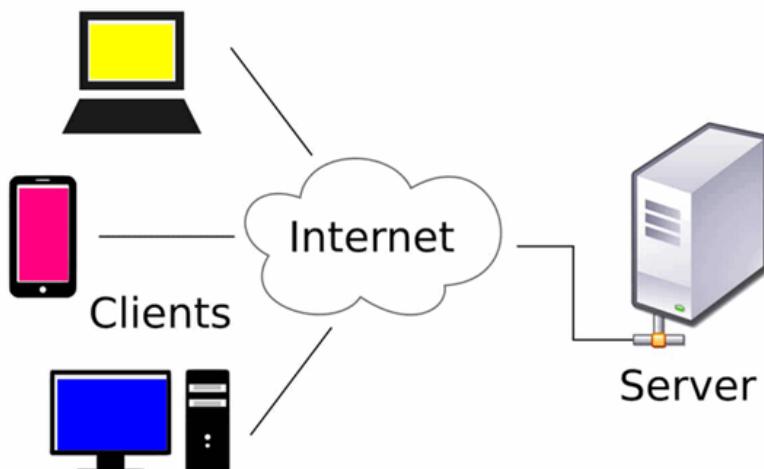
## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Mô hình Client-Server

Mô hình client-server [1] là một mô hình mạng máy tính được sử dụng rộng rãi. Bản chất của mô hình client-server là các máy tính, bao gồm một máy chủ (server) và nhiều máy khách (client) giao tiếp, trao đổi dữ liệu với nhau thông qua môi trường mạng. Các máy client khi cần thông tin sẽ gửi các yêu cầu đến máy chủ. Máy chủ luôn luôn trong trạng thái sẵn sàng nhận request, sẽ xử lý yêu cầu được gửi đến từ client rồi sau đó trả về kết quả lại trong thời gian ngắn nhất.

Để mô hình client-server hoạt động thì giữa các máy server và client cần có một chuẩn nhất định gọi là giao thức (ví dụ như HTTPS, TCP/IP, FPT,...) Bên cạnh đó, cấu hình của server phải cao hơn cấu hình của các client.

Mô hình client-server giúp chúng ta có thể làm việc trên bất kỳ một máy tính nào có hỗ trợ truyền thông. Chúng chỉ mang đặc điểm của phần mềm mà không hề liên quan đến phần cứng, có thể hỗ trợ người dùng sử dụng các dịch vụ một cách đa dạng và tiện dụng nhờ khả năng truy cập từ xa.



Hình 3.1: Mô hình client - server <sup>1</sup>

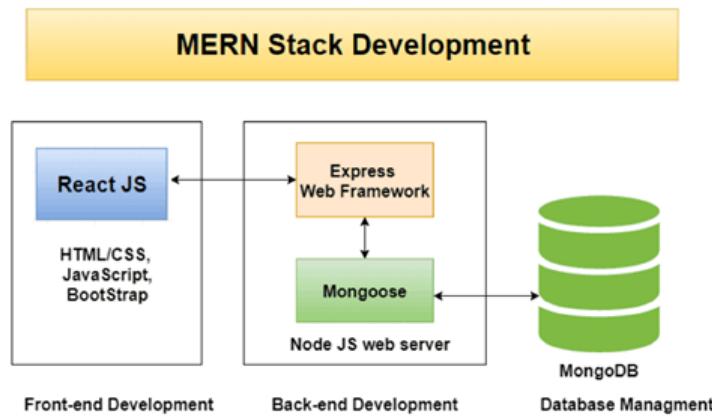
### 3.2 MERN Stack

#### 3.2.1 Tổng quan về MERN Stack

MERN Stack [2] là một thuật ngữ chỉ một bộ các công nghệ mã nguồn mở liên quan đến Javascript, bao gồm MongoDB, ExpressJS, React JS/React Native, Nodejs. Những ứng dụng được xây dựng từ MERN Stack được gọi là React Universal App.

<sup>1</sup>Nguồn: <https://tanhongit.com/mo-hinh-client-server-la-gi-va-cac-hoat-dong-client-server/>

MERN Stack được tạo ra để cung cấp quá trình phát triển mượt mà hơn. Stack Development này cung cấp một khuôn khổ end-to-end cho các nhà phát triển và đang được phổ biến ngày càng rộng rãi.



**Hình 3.2:** Xử lý dữ liệu trong MERN Stack<sup>2</sup>

### 3.2.2 MongoDB

MongoDB [3] là hệ quản trị cơ sở dữ liệu mã nguồn mở phi quan hệ (NoSQL - None-Relational SQL) và hướng tài liệu (document). Các dữ liệu được MongoDB lưu trữ được lưu dưới dạng JSON (cụ thể là BSON) trong document.

BSON (Binary JSON) là dữ liệu có định dạng kiểu JSON theo dạng (key: value). Dữ liệu được lưu trong MongoDB được gọi là các BSON document. Các dữ liệu kiểu BSON ở đây có thể dạng chuỗi, mảng, document khác, hoặc mảng document... Dưới đây là ví dụ một dữ liệu kiểu BSON:

```
{
    name: "Thanh", // string
    role: 1, // integer
    favourite: ["Bóng đá", "Đá cầu"], //array
    address: {
        province: "Hải Dương",
        district: "Ninh Giang",
        ward: "Tân Phong"
    } //nested document
}
```

**Hình 3.3:** Ví dụ dữ liệu kiểu BSON

Một số thuật ngữ được dùng trong MongoDB:

<sup>2</sup>Nguồn: <https://www.bocasay.com/how-does-the-mern-stack-work/>

**\_id:** trường bắt buộc có trong mỗi document. Đây là trường được MongoDB tự động sinh ra mỗi khi thêm mới một document vào collection. Giá trị \_id này là duy nhất trong cơ sở dữ liệu MongoDB nên nó cũng có thể được hiểu là khóa chính trong document, đại diện cho document.

**document:** là một bản ghi thuộc một collection, bao gồm các cặp giá trị (key: value) lưu dưới dạng BSON. **collection:** Nếu cơ sở dữ liệu quan hệ sử dụng các bảng (table) để lưu trữ dữ liệu thì MongoDB sử dụng collection. Collection là nhóm của nhiều document, và ta cũng không cần phải định nghĩa các cột, hàng hay kiểu dữ liệu trước.

**database:** chứa các collection.

Do MongoDB lưu trữ dữ liệu dưới dạng Document JSON, không có tính ràng buộc nên việc lưu trữ và truy vấn dữ liệu trở lên nhanh và linh hoạt. Việc mở rộng dữ liệu cũng trở nên dễ dàng hơn. Từ những thử nghiệm cho thấy tốc độ của MongoDB có thể nhanh gấp 100 lần so với My SQL. Chính vì vậy mà MongoDB đã trở thành một trong những NoSQL database nổi bật nhất, được dùng cho nhiều website như eBay, The New York Times,...

Tuy nhiên, bên cạnh những ưu điểm vượt trội nêu trên, MongoDB vẫn có một số nhược điểm nhất định. Do không có ràng buộc nên MongoDB không được sử dụng cho các mô hình giao dịch yêu cầu độ chính xác cao. Việc lưu trữ các bản ghi dưới dạng (key: value) dẫn đến việc tốn bộ nhớ và dư thừa dữ liệu. Bên cạnh đó chính là nguy cơ mất dữ liệu khi chưa hoàn thành bản lưu do quá trình truy vấn dữ liệu của MongoDB không cập nhật ngay xuống ổ cứng nên nếu xảy ra mất điện trong khi truy vấn thì sẽ bị mất dữ liệu.

### 3.2.3 NodeJS

NodeJS [4] là một nền tảng phát triển độc lập được xây dựng trên Javascript Runtime của Chrome (V8) với mục đích xây dựng các ứng dụng mạng nhanh chóng và có thể mở rộng một cách dễ dàng. NodeJS sử dụng mô hình event driven (mô hình lập trình hướng sự kiện), non-blocking nên tiết kiệm được dữ liệu và hiệu quả, là công cụ hoàn hảo cho các ứng dụng thời gian thực. NodeJS cũng cung cấp một thư viện bao gồm các module Javascript khác nhau nhằm đơn giản hóa việc phát triển ứng dụng web. Ngoài ra, hệ thống sử dụng ReactJS/React Native ở tầng frontend, việc sử dụng đồng nhất Javascript ở cả hai phía Backend và Frontend sẽ thuận tiện cho việc viết phần mềm, tối ưu hóa thời gian phát triển hệ thống.

Một số thuật ngữ dùng trong NodeJS [5]:

**NPM (Node Package Manager):** là chương trình quản lý thư viện ngầm định

trong môi trường NodeJS, được tích hợp sẵn trong gói cài đặt NodeJS. NPM gồm hai thanh phần quan trọng: Kho chứa các gói (module, thư viện, framework) mã nguồn mở cho nền tảng NodeJS và chương trình chạy ở chế độ dòng lệnh (command line -npm) để tương tác với các kho chứa nhằm mục đích cài đặt, quản lý các phiên bản và dependency cho các gói.

**Module:** là các phần mã nguồn riêng biệt, được kết hợp với nhau để tạo một chương trình hoàn chỉnh. Lập trình theo hướng module hóa có nghĩa là chia chương trình thành các module nhỏ nhằm dễ kiểm soát và tái sử dụng.

**Require:** là một hàm gọi module trong NodeJS. NodeJS có các tính năng nổi bật như sau:

Lập trình hướng sự kiện và bất đồng bộ: Toàn bộ API trong thư viện NodeJS đều bất đồng bộ. Một server sử dụng NodeJS sẽ không bao giờ chờ một API trả về dữ liệu mà sẽ chuyển sang API kế tiếp sau đó và cơ chế thông báo Events trong NodeJS sẽ giúp server nhận được phản hồi từ lần gọi API trước.

Nhanh chóng: Được xây dựng trên công cụ Javascript V8 của Google Chrome, thư viện Node.js có khả năng xử lý mã vô cùng nhanh.

Đơn luồng nhưng có khả năng mở rộng cao: NodeJS sử dụng mô hình đơn luồng với vòng lặp sự kiện/event. Cơ chế event cho phép máy chủ phản hồi non-blocking và cũng cho phép khả năng mở rộng cao hơn so với các server truyền thống hỗ trợ giới hạn các thread để xử lý yêu cầu. Không có buffer: Các ứng dụng NodeJS không có vùng nhớ tạm thời cho bấy kỳ dữ liệu nào mà chỉ xuất dữ liệu theo khối.

Nhờ những tính năng nổi bật trên mà NodeJS có rất nhiều ưu điểm như nhận và xử lý nhiều kết nối chỉ với một single-thread giúp ngắn ít RAM nhất và chạy nhanh nhất, phù hợp xây dựng các ứng dụng Single Page Application; nhờ non-blocking I/O của Javascript nên NodeJS tận dụng tài nguyên của server một cách tối đa mà không tạo ra độ trễ; chạy đa nền tảng trên Windows, MAC hoặc Linux.

Bên cạnh đó, NodeJS tồn tại một vài nhược điểm như việc triển khai trên Host không dễ dàng; NodeJS không sử dụng để xây dựng các ứng dụng nặng và tốn tài nguyên; thiếu sự kiểm duyệt chất lượng các module NodeJS. Một số công ty/tổ chức hiện đang sử dụng NodeJS có thể kể đến eBay, Microsoft, PayPal, Uber, Yammer,...

### 3.2.4 ExpressJS

ExpressJS [6] là một framework mã nguồn mở, được xây dựng trên nền tảng NodeJS, cung cấp các tính năng mạnh mẽ để phát triển các ứng dụng Web hoặc Mobile.

ExpressJS giúp tổ chức các ứng dụng web ở phía server thành kiến trúc MVC (Model - View - Controller). Bên cạnh đó ExpressJS hỗ trợ các phương thức HTTP và middleware tạo ra API mạnh mẽ và dễ sử dụng. Nếu không sử dụng ExpressJS, thì lập trình viên sẽ phải thực hiện các bước phức tạp để xây dựng một API hiệu quả. ExpressJS có thể làm giảm một nửa thời gian viết mã mà vẫn có thể xây dựng ứng dụng web hiệu quả.

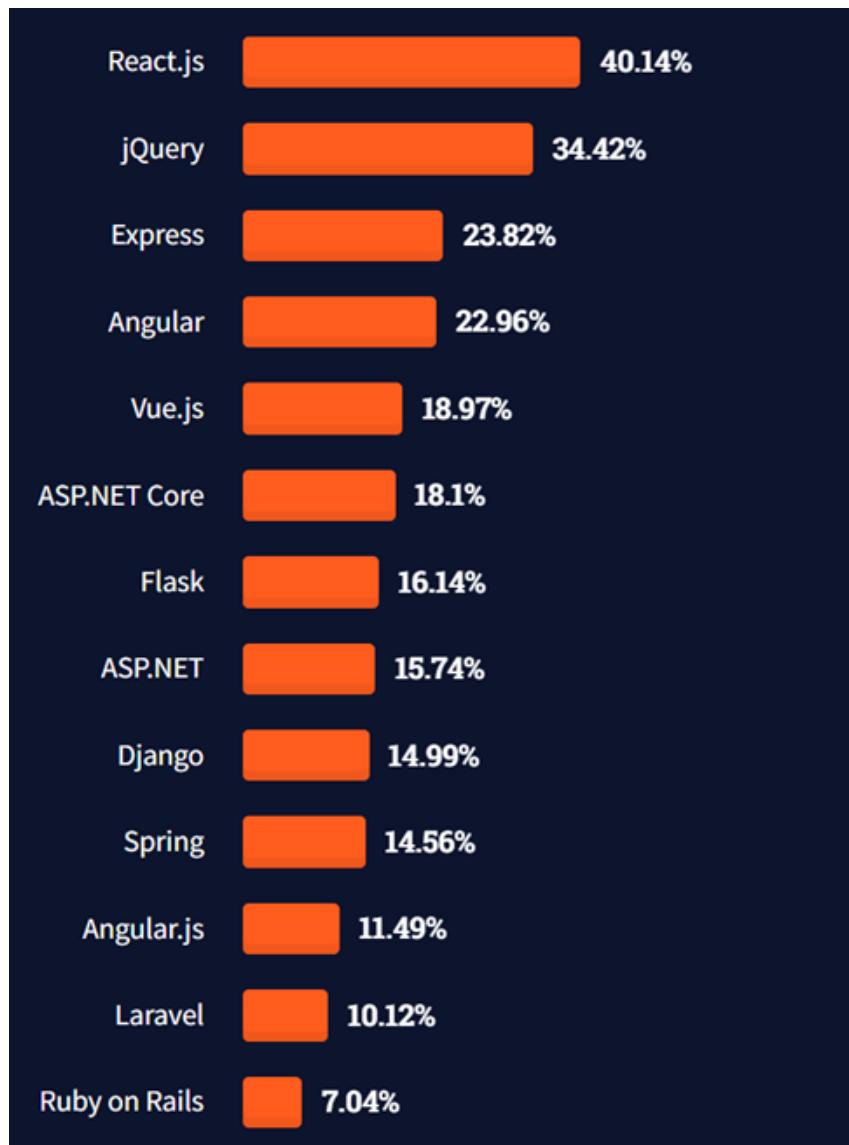
### 3.2.5 ReactJS

ReactJS là một thư viện mã nguồn mở dựa trên JavaScript, được tạo ra từ Facebook. ReactJS được sử dụng để xây dựng các thành phần giao diện người dùng có thể tái sử dụng và hiển thị thay đổi theo thời gian. ReactJS thường được sử dụng thay thế cho View trong mô hình kiến trúc MVC do hiệu suất và tốc độ nhanh, hiệu quả cao, bên cạnh đó là khả năng xử lý bất đồng bộ.

React cho phép chia giao diện trang web thành các component. Các component có đặc điểm nhỏ gọn và có thể tái sử dụng. React có hai loại component là Class Component (Stateful) và Functional Component (Stateless). Với Class Component, chúng ta có thể sử dụng được state một cách dễ dàng khi có thể khởi tạo ngay trong phương thức constructor của Class Component. Ngược lại, Functional Component không thể làm nhiều công việc quản lý state phức tạp giống như Class Component. Tuy nhiên, từ version 16.8, với sự ra đời của React Hooks thì việc quản lý state đã trở nên dễ dàng không kém gì Class Component. Bên cạnh đó, với quan điểm cá nhân, Functional Component tiện lợi cho việc bảo trì cũng như dễ hiểu hơn Class Component nên em đã chọn sử dụng Functional để phát triển ứng dụng này.

Bên cạnh đó, để cải tiến hiệu năng cho các trang web, React đã sử dụng Virtual DOM giúp lưu lại cấu trúc giao diện và đồng bộ với cấu trúc DOM. Mỗi khi có sự thay đổi, React chỉ render lại các phần tử cần thay đổi mà không render lại toàn bộ trang web. Điều này khiến giảm thời gian tính toán và tăng hiệu năng trang web một cách đáng kể.

Báo cáo Khảo sát dành cho nhà phát triển StackOverflow mới nhất năm 2021 đã khẳng định React là web framework được các nhà phát triển lựa chọn nhiều nhất. Theo báo cáo này, ở vị trí dẫn đầu, với khoảng 40.14% lập trình viên chọn dùng React để phát triển cho ứng dụng của mình.



**Hình 3.4:** Kết quả khảo sát web framework được ưa thích nhất của Stack Overflow<sup>3</sup>

Chính vì mang lại được hiệu suất và khả năng xử lý bất đồng bộ nên em đã chọn dùng framework ReactJS để phát triển cho hệ thống của mình.

### 3.3 Dịch vụ lưu trữ dữ liệu Azure Blob Storage

Azure Blob Storage [7] là một công cụ cho phép lưu trữ các dữ liệu phi cấu trúc trên cloud. Các dữ liệu đó có thể là text, dữ liệu nhị phân, tệp tài liệu hay tệp phương tiện,... được lưu dưới dạng object, được gọi là blob object.

Azure Blob Storage lưu trữ dữ liệu theo kiểu thư mục – file. Mỗi file dữ liệu lưu trữ được gọi là blob. Các blob nằm trong một container tương tự như một tệp nằm trong thư mục. Các container nằm trong storage account.

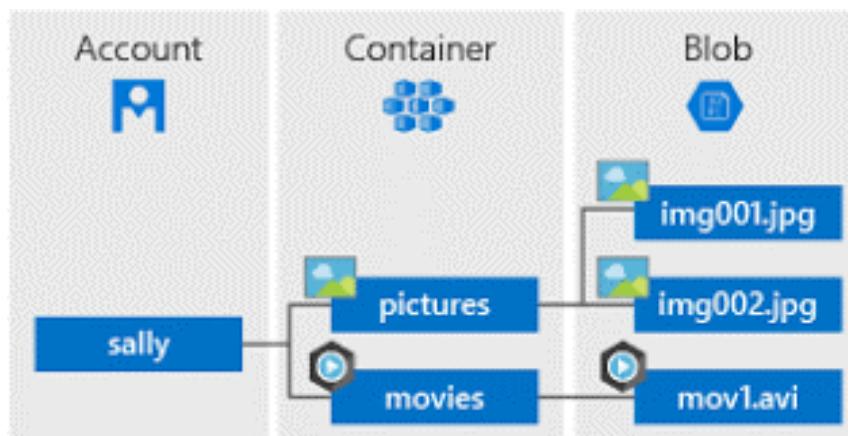
Azure Blob Storage chia blob ra thành 3 loại:

<sup>3</sup>Nguồn: <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>

**Block blob** dùng để lưu trữ text hoặc các file đã xác định độ lớn.

**Append blob** dùng để lưu các file logging, có thể thay đổi độ lớn liên tục.

**Page blob** sử dụng cho cơ chế đọc viết, với dung lượng lớn lên đến 1TB.



Hình 3.5: Mối quan hệ giữa các thành phần của Azure Blob Storage<sup>4</sup>

## 3.4 Các thư viện được sử dụng

### 3.4.1 Thư viện frontend

**Thư viện Ant Design** là tập hợp các components của React, được xây dựng theo chuẩn thiết kế của Ant UED Team. Các component của Ant Design đều là các component thông dụng trong web hiện đại ví dụ như Layout, Button, Icon,... Bên cạnh đó, Ant Design cho phép người dùng tùy chỉnh các thuộc tính một cách dễ dàng.

**Thư viện axios** là một thư viện HTTP Client, được dùng để hỗ trợ cho việc xây dựng các ứng dụng API.

**Thư viện react-i18next** cung cấp một số hàm để cấu hình đa ngôn ngữ.

**Thư viện Formik và Yup** hỗ trợ quản lý form như nhận giá trị trong form, xác thực dữ liệu và đưa ra lỗi, xử lý việc gửi form

### 3.4.2 Thư viện backend

**Thư viện @azure/storage-blob** giúp kết nối lưu trữ dữ liệu lên Azure Blob Storage.

**Thư viện bcrypt** hỗ trợ mã hóa mật khẩu dựa trên thuật toán Bcrypt. Thuật toán Bcrypt được đánh giá là bảo mật và an toàn bởi mỗi lần thực hiện hàm băm nó lại trả về một giá trị khác nhau, khiến cho việc dò tìm mật khẩu trở nên khó hơn.

<sup>4</sup>Nguồn: <https://know.panzura.com/creating-a-microsoft-azure-blob-storage>

**Thư viện multer** như một middle ware hỗ trợ Express JS và Node JS dễ dàng xử lý dữ liệu dưới dạng multipart/form-data khi người dùng tải lên các file từ phía client để lưu trữ.

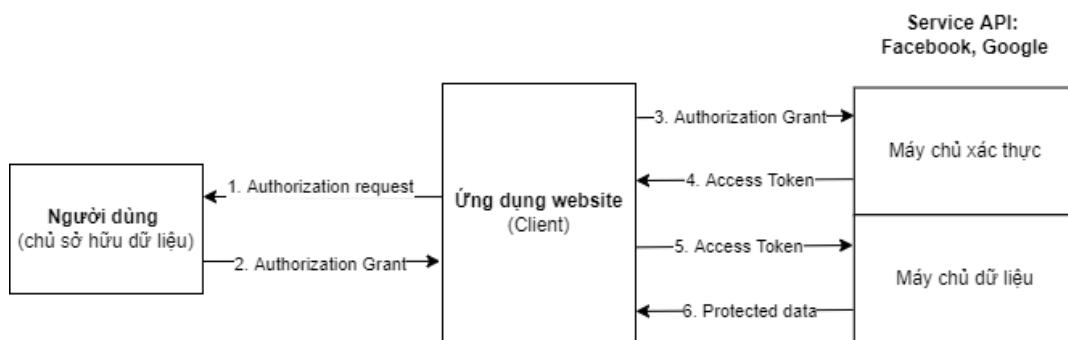
**Thư viện nodemailer** sử dụng giao thức SMTP để hỗ trợ Node JS khi gửi email.

**Thư viện socket.io** hỗ trợ xây dựng ứng dụng thời gian thực. Socket.io giúp các máy kết nối và truyền dữ liệu với nhau ngay lập tức thông qua server trung gian.

### 3.5 OAuth 2

OAuth 2 (Open Authentication/Authorization) [8] là một phương thức chứng thực giúp các ứng dụng chia sẻ tài nguyên với nhau mà không cần chia sẻ thông tin như username, password.

Cách đăng nhập bằng phương thức OAuth được đánh giá là an toàn cho người dùng cuối. Đầu tiên, các ứng dụng (các website, mobile app) yêu cầu ủy quyền để truy cập vào resource Server như (Gmail, Facebook, Github,...) thông qua user. Sau khi user ủy quyền cho yêu cầu trên thì một token string sẽ được sinh ra. Khi ứng dụng gửi thông tin định danh (ID) của mình kèm theo token string ủy quyền tới Authorization Server, nếu thông tin định danh được xác thực và ủy quyền hợp lệ thì Authorization Server sẽ trả về cho ứng dụng một access token. Như vậy, quá trình ủy quyền được hoàn tất.



**Hình 3.6:** Quá trình xử lý khi đăng nhập bằng Oauth2

Để truy cập vào tài nguyên từ Resource Server và lấy thông tin thì ứng dụng sẽ sử dụng access token để xác thực. Sau khi kiểm tra tính hợp lệ của access token thì Resource server sẽ trả về dữ liệu của tài nguyên đã được yêu cầu cho ứng dụng.

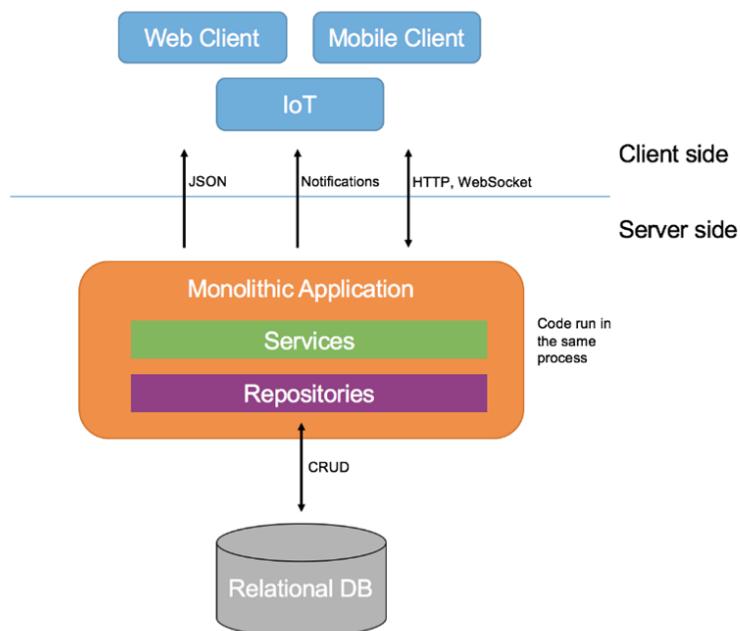
Chính vì vậy, trong trường hợp website bị tấn công thì chỉ có các thông tin hay hoạt động của user trên website đó bị lộ, ngoài ra thì không ảnh hưởng đến các website khác mà user đang sử dụng.

## CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

Trong hệ thống này, em đã lựa chọn kiến trúc phần mềm là kiến trúc nguyên khối (Monolithic Architecture). Trước microservices[9], kiến trúc chính cho các dịch vụ web là kiến trúc nguyên khối. Thông thường, ứng dụng nguyên khối bao gồm giao diện người dùng, ứng dụng phía máy chủ và cơ sở dữ liệu. Tất cả các bộ phận của phần mềm được hợp nhất và tất cả các chức năng của phần mềm được quản lý ở một nơi. Điều đó có nghĩa là tất cả các thành phần nguyên khối được kết nối và phụ thuộc lẫn nhau, các tính toán hay biến đổi,... đều được xảy ra trong một quy trình ứng dụng.



**Hình 4.1:** Kiến trúc nguyên khối (Monolithic Architecture)<sup>1</sup>

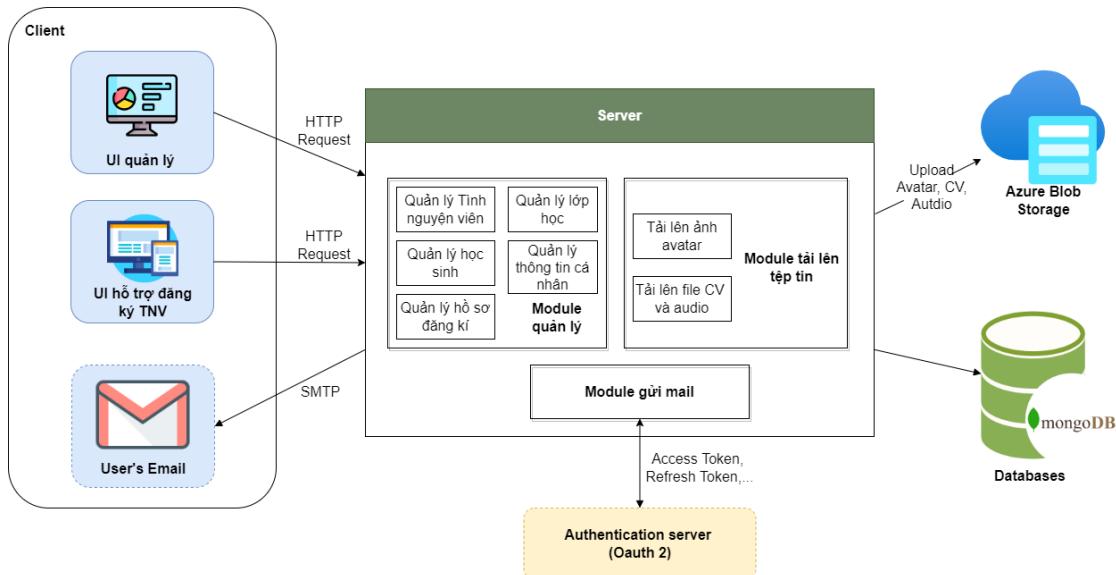
Client sẽ gửi các HTTP Request tới cổng API của server để lấy dữ liệu. Phía server sẽ nhận các request đó, tính toán và xử lý rồi phản hồi Client dưới dạng JSON. Dựa trên những dữ liệu trả về, client sẽ hiển thị UI tương ứng.

Hệ thống em xây dựng tuân thủ theo kiến trúc Monolithic. Chi tiết các thiết kế, em sẽ trình bày tại các phần 4.1.2, 4.1.3 và 4.2

<sup>1</sup><https://morioh.com/p/d925f09098cb>

### 4.1.2 Thiết kế tổng quan

Hệ thống bao gồm giao diện phía người dùng, web server, databases và storage. Có hai giao diện phía người dùng đó là: giao diện website quản lý và giao diện website hỗ trợ đăng ký làm tình nguyện viên.



**Hình 4.2:** Thiết kế tổng quan của hệ thống

**Website quản lý** cung cấp tất cả các chức năng liên quan đến quản lý dạy học và nhân sự. Website này hướng tới người dùng là quản trị viên, tình nguyện viên tham gia dạy học và học sinh tham gia học.

**Website hỗ trợ đăng ký làm tình nguyện viên** hướng tới người dùng là những người có hứng thú tìm hiểu về Lớp học Cầu Vồng, mong muốn trở thành tình nguyện viên của tổ chức. Website này không yêu cầu người dùng phải đăng nhập mà vẫn có thể xem được các thông tin giới thiệu về tổ chức và gửi CV/hồ sơ đăng ký về hệ thống và chờ duyệt.

Hai website này sử dụng chung một **backend server**. Backend server này cung cấp các api để thực hiện xử lý các request được gửi lên từ client, truy vấn dữ liệu từ phía database hoặc tương tác với Storage để lưu các dữ liệu dạng Blob.

Tại backend server được chia thành ba module chính là module quản lý module đăng ký hồ sơ làm Tình nguyện viên và module gửi email.

**Module quản lý** cung cấp các chức năng dành cho quản trị viên và những người dùng bắt buộc phải đăng nhập hệ thống (tại website quản lý) ví dụ như quản lý tình nguyện viên, quản lý học sinh, quản lý CV, quản lý lớp học hay quản lý thông tin cá nhân. Đối với những user thuộc role khác nhau thì các quyền hạn và chức năng sẽ khác nhau.

**Module đăng ký** cung cấp chức năng dành cho những người dùng là khách, không cần đăng nhập, có nguyện vọng đăng ký trở thành tình nguyện viên của tổ chức (tại website tổng quan và hỗ trợ đăng ký làm Tình nguyện viên) ví dụ như nhập liệu đăng ký các thông tin cá nhân, tải lên file CV và audio.

**Module gửi email** sử dụng Oauth 2 của Google để xác minh tài khoản rồi sau đó hỗ trợ hệ thống gửi email một cách tự động đến user trong các tình huống khác nhau như: kích hoạt tài khoản, đặt lại mật khẩu khi quên mật khẩu, thông báo quá trình xử lý tiếp nhận hồ sơ đăng ký làm Tình nguyện viên và thông báo lịch phỏng vấn,...

#### 4.1.3 Thiết kế chi tiết gói

Web server được thiết kế và chia thành các gói router, controller, service, repository, model.

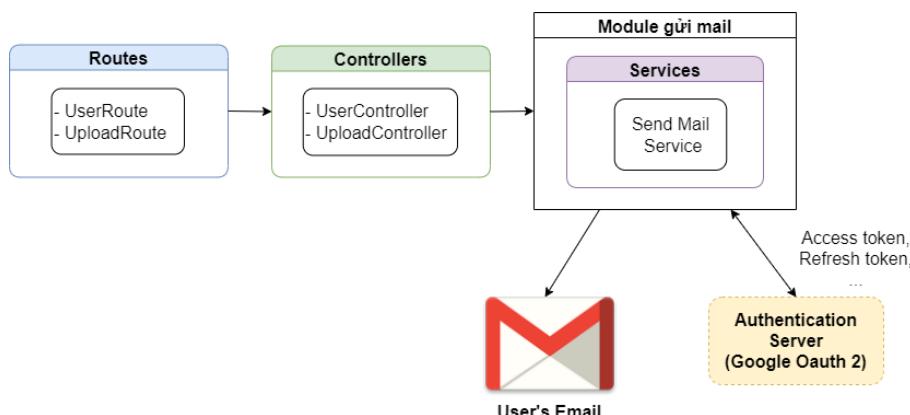
**Router** là một bộ định tuyến giúp cho chúng ta định danh ra các url api được gửi từ client và hành động kèm theo nó.

**Controller** tiếp nhận các request chứa yêu cầu, điều hướng xử lý tới Service hoặc xử lý logic nghiệp vụ bằng Repository và trả về response.

**Service** tiếp nhận, xử lý các yêu cầu được gửi từ Controller.

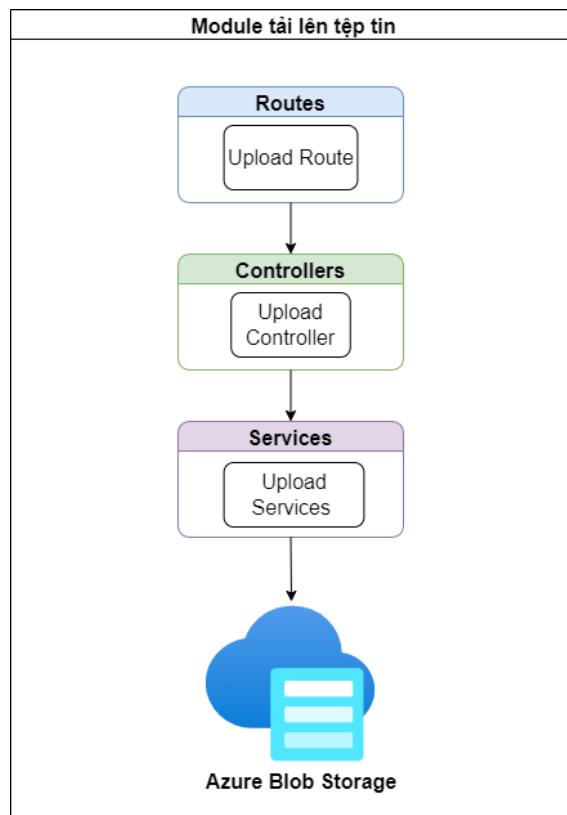
**Repository** nhận yêu cầu từ Controller và tương tác trực tiếp với Database trả về dữ liệu được yêu cầu.

Tại **Module gửi mail**, hệ thống quản lý sẽ trở thành ứng dụng bên thứ ba, kết nối với ứng dụng gốc là gmail của Google. Nhờ Oauth2, ta có thể lấy được ủy quyền gửi email từ một địa chỉ email nhất định thông qua client\_id, client\_secret, access\_token, refresh\_token. Sau khi lấy được ủy quyền gửi mail rồi, thông qua nodemailer để gửi các email đến người dùng



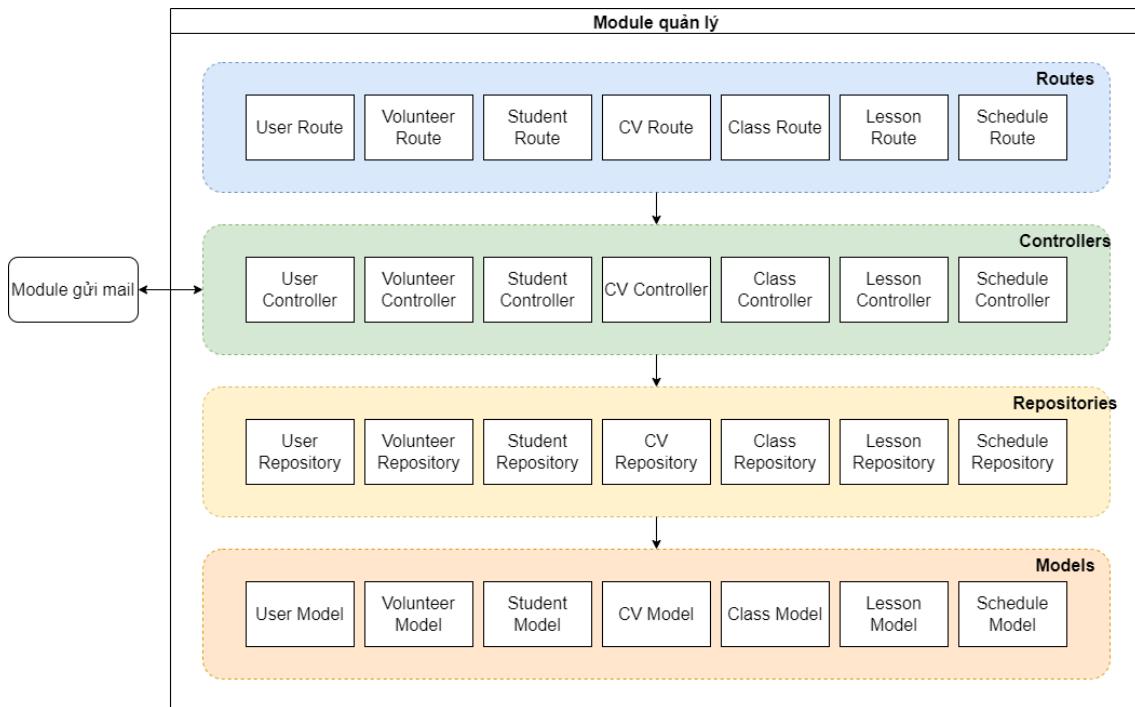
Hình 4.3: Thiết kế gói module gửi email

**Module tải lên tệp tin** bao gồm 4 thành phần chính là Route, Controller, Services và Storage. Khi client gửi một tệp tin lên server, uploadRoute sẽ định tuyến và gọi đến UploadController để xử lý. Controller sẽ lấy các dữ liệu cần thiết để gọi đến UploadService. Dịch vụ này không kết nối đến cơ sở dữ liệu mà sử dụng kho lưu trữ Blob Storage của Azure để lưu trữ tài nguyên.



**Hình 4.4:** Thiết kế gói module tải lên tệp tin

**Module quản lý** được chia thành 4 thành phần chính là Routes, Controllers, Repositories và Models. Routes được chia thành các module nhỏ hơn như User Route, Volunteer Route, Student Route, Class Route, CV Route, Lesson Route, Schedule Route ... Và tương ứng với mỗi modules nhỏ của Route thì các thành phần Controllers, Repository, Model cũng có những modules tương ứng để tương tác. Do số lượng các phần tử thuộc module quản lý nhiều và cách thức hoạt động giống nhau nên em sẽ chỉ tập trung mô tả một vài thành phần quan trọng.



**Hình 4.5:** Thiết kế gói module quản lý

Khi client gửi request đến server thông qua REST API, các Routers sẽ định tuyến và gọi đến các Controllers tương ứng xử lý. Controllers sẽ có nhiệm vụ lấy các dữ liệu đầu vào cần thiết và chuyển đến Repositories. Repositories sẽ là nơi xử lý tất cả những logic liên quan đến nghiệp vụ của ứng dụng và cũng thực hiện truy vấn tới cơ sở dữ liệu MongoDB thông qua Models. Ngoài ra, User Controller và CV Controller cũng sẽ tương tác với module gửi mail.

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

Đối với **website hỗ trợ đăng ký làm Tình nguyện viên**, do không yêu cầu đăng nhập và ít chức năng nên hệ thống hỗ trợ cả giao diện mobile lẫn giao diện desktop, hiển thị trên mọi màn hình với kích thước khác nhau để bất kỳ ai cũng có thể xem được thông tin về lớp học và đăng ký trở thành tình nguyện viên. Hơn nữa, website này hướng tới đối tượng người dùng là bất kỳ ai nên cũng được thiết kế một cách đẹp và bắt mắt hơn. Các trình duyệt hỗ trợ như Chrome, Firefox, Safari, Microsoft Edge.

Ngoài ra, tại màn hình mobile và màn hình desktop có một sự khác biệt giữa thanh navigation bar đó là: màn hình desktop có 3 button là “Trang chủ”, “Nộp CV”, “Đăng nhập” trong khi đó ở màn hình mobile chỉ xuất hiện hai button “Trang chủ” và “Nộp CV”.

Dưới đây là thiết kế mockup màn hình mobile và desktop của màn đăng ký làm

Tính nguyện viên.

**Đăng ký làm Tình nguyện viên**

\*Họ và tên

\*Địa chỉ email

\*Số điện thoại

\*Lớp học đăng ký

[Tiếp theo](#)

Footer

[Trang chủ](#) [Nộp CV](#) [Đăng nhập](#)

**Đăng ký làm Tình nguyện viên**

\*Họ và tên

\*Địa chỉ email

\*Số điện thoại

\*Lớp học đăng ký

[Tiếp theo](#)

Footer

**Hình 4.6:** Thiết kế mockup giao diện cho màn hình đăng ký làm TNV

Đối với **website quản lý**, hệ thống chỉ hỗ trợ giao diện desktop. Website quản lý bao gồm 2 layout chính: layout cho trang login, reset password và layout cho các trang quản lý.

Layout cho trang login hoặc reset password gồm hai thành phần chính là logo của Lớp học Cầu Vồng (nằm phía bên phải) và form login/reset password (nằm phía bên trái)

Chào mừng đến với  
Lớp học Cầu Vồng!

Logo

**Đăng nhập**

Địa chỉ email \*

Mật khẩu \*

[Quên mật khẩu?](#)

[Đăng nhập](#)

**Hình 4.7:** Thiết kế mockup giao diện cho Login Layout

Với giao diện cho các màn hình quản lý thì ở mỗi trang đều có header menu (bao gồm tên tổ chức, icon thông báo, avatar, họ tên và role của login user), một menu sidebar ở phía bên trái để người dùng có thể truy cập các màn quản lý và

phần content chứa các nội dung hiển thị. Ngoài ra ở phía bên dưới phần content còn có phần liên kết đến fanpage chính thức của Lớp học Cầu Vồng và đường link liên kết đến form phản hồi về các lỗi hoặc mong muốn cải thiện chức năng cho hệ thống.

Dưới đây là thiết kế giao diện mockup cho layout quản lý:



**Hình 4.8:** Thiết kế giao diện mockup cho các màn hình quản lý

Tại website quản lý, các phần tử như button, modal, header, table, popover, icon, form đều được thiết kế đồng nhất cho từng trường hợp cụ thể từ màu sắc, kiểu dáng. Các màu sắc về trạng thái hoặc kích cỡ phông chữ cũng đều được quy định sẵn, tạo nên sự đồng bộ giữa các màn hình chức năng.

#### 4.2.2 Thiết kế lớp

##### a, Module tải lên tệp tin

##### Lớp UploadController

Upload Controller
+ uploadAvatar(): void
+ uploadCV (): void

**Bảng 4.1:** Lớp Upload Controller

Ý nghĩa các phương thức:

+ uploadAvatar(): nhận request file phương tiện từ client rồi gọi đến upload service để lưu file. Sau đó trả về đường dẫn đến file phương tiện cho client.

+ uploadCV(): nhận request chứa file CV và file audio từ client rồi gọi đến upload service để lưu file. Sau đó lấy đường dẫn trả về từ upload service để lưu thông tin file vào cơ sở dữ liệu.

### Lớp UploadFileService

<b>UploadFileService</b>
- storageMulter: MulterAzureStorage
+ uploadImage(): void
+ uploadCVFile(): void

**Bảng 4.2:** Lớp Upload File Service

Ý nghĩa các phương thức:

- + UploadImage(): Tải image lên AzureBlobStorage
- + UploadCVFile(): Tải file CV (dưới dạng PDF) và audio giới thiệu bằng tiếng Anh (dạng MP3) lên AzureBlobStorage

### b, Module gửi mail

#### Lớp SendMailService

<b>SendMailService</b>
- myOAuth2Client: OAuth2Client
- myAccessTokenObject: Object
- myAccessToken: String
- transporter: Object
+ sendMailInterview(): Boolean
+ sendMailAccount (): Boolean

**Bảng 4.3:** Lớp SendMailService

Ý nghĩa các phương thức:

- + sendMailInterview(): Gửi email cho người dùng sau mỗi lần trạng thái hồ sơ của họ được cập nhật.
- + sendMailAccount(): Gửi email kích hoạt tài khoản cho người dùng mới hoặc gửi email reset password đến những người quên mật khẩu kèm theo token định danh.

### c, Module Quản lý

Do số lượng các lớp và chức năng là nhiều nên em sẽ chỉ đưa ra thiết kế lớp của một vài lớp quan trọng.

#### Lớp UserController

User Controller
+ getMySchedule(): Array + getClassSchedule(): Array + getCurrentUser(): Object

**Bảng 4.4:** Lớp User Controller

Ý nghĩa các phương thức:

- + getMySchedule(): Lấy dữ liệu lịch trình của người dùng hiện tại
- + getClassSchedule(): Lấy dữ liệu lịch trình của lớp học mà người dùng hiện tại đang trực thuộc
- + getCurrentUser(): Lấy dữ liệu thông tin về người dùng hiện tại

#### Lớp StudentController

StudentController
+ getStudents(): Array + getStudentInfo(): Object + changeStudentStatus(): void + updateStudentOverview(): Void + getStudentsDataByFilter(): Array

**Bảng 4.5:** Lớp StudentController

Ý nghĩa các phương thức: + getStudents(): Lấy dữ liệu của tất cả học sinh

- + getStudentInfo(): Lấy tất cả dữ liệu của một học sinh
- + changeStudentStatus(): Thay đổi trạng thái, ngày nghỉ học của học sinh
- + updateStudentOverview(): Cập nhật mô tả cho học sinh
- + getStudentsDataByFilter(): Lấy dữ liệu của học sinh dựa trên bộ lọc

#### Lớp VolunteerController

<b>VolunteerController</b>
<ul style="list-style-type: none"> <li>+ getAllVolunteers(): Array</li> <li>+ getVolunteerData(): Object</li> <li>+ getCurrentVolunteer(): Object</li> <li>+ changeVolunteerStatus(): Void</li> </ul>

**Bảng 4.6:** Lớp VolunteerController

Ý nghĩa các phương thức:

- + getAllVolunteers(): Lấy dữ liệu của tất cả tình nguyện viên
- + getVolunteerData(): Lấy tất cả dữ liệu của một tình nguyện viên
- + getCurrentVolunteer(): Lấy dữ liệu của tình nguyện viên hiện tại
- + changeVolunteerStatus(): Thay đổi trạng thái, ngày nghỉ hoạt động của tình nguyện viên

### Lớp CV Repository

<b>CVRepository</b>
<ul style="list-style-type: none"> <li>+ storeCV(): void</li> <li>+ getAllCVs(): Array</li> <li>+ getCVById(): CV</li> <li>+ updateCV(): void</li> <li>+ getInterviewScheduleByClassAndMonth(): Array</li> </ul>

**Bảng 4.7:** Lớp CVRepository

Ý nghĩa các phương thức:

- + storeCV(): Lưu một CV mới
- + getCVById(): Trả về dữ liệu của CV theo \_id
- + getAllCVs(): Lấy tất cả CV
- + updateCV(): Cập nhật trạng thái và thời gian phỏng vấn của CV
- + getInterviewScheduleByClassAndMonth(): Lấy tất cả CV và lịch phỏng vấn của CV theo lớp và tháng.

### Lớp CV Controller:

<b>CVController</b>
+ getListCV(): Array + getCVDataById(): CV + updateCVStatus(): void + getNumberOfCV(): Object

**Bảng 4.8:** Lớp CVController

Ý nghĩa các phương thức:

- + getListCV(): Nhận request từ Client, trả về danh sách tất cả các CV
- + getCVDataById(): Nhận request từ Client, trả về thông tin CV dựa theo \_id
- + updateCVStatus(): Nhận request từ Client, gọi đến updateCV ở CVRepository để thay đổi trạng thái, lịch phỏng vấn
- + getNumberOfCV(): Trả về tổng số lượng CV và số lượng CV theo từng trạng thái

### Lớp ClassRepository

<b>ClassRepository</b>
- transformClassData(): Object
+ storeClass(): void
+ editClass(): void
+ deleteClass(): void
+ getClassScheduleByUser(): Array
+ getAllClassesData(): Array
+ setMonitor(): void,
+ listClassWithName(): Array,
+ getNumberOfClassesData(): Object

**Bảng 4.9:** Lớp Class Repository

Ý nghĩa các phương thức:

- + transformClassData(): trả về dữ liệu của một lớp học với các thông tin khác như danh sách học sinh, danh sách tình nguyện viên, ...
- + storeClass(): Lưu thông tin lớp mới
- + editClass(): Sửa đổi thông tin lớp

- + deleteClass(): Xóa lớp
- + getClassScheduleByUser(): Lấy tất cả các lịch hoạt động của lớp mà user trực thuộc
- + getAllClassesData(): Lấy dữ liệu của tất cả các lớp học
- + setMonitor(): Chính sửa cán sự cho lớp học
- + listClassWithName(): Trả về tên của tất cả các lớp học
- + getNumberOfClassesData(): Trả về tổng số lượng lớp học và số lượng lớp học theo từng hình thức dạy

### Lớp ClassController

ClassController
<ul style="list-style-type: none"><li>+ getAllClasses(): Array</li><li>+ getClassData(): Object</li><li>+ getClassSchedule(): Object</li><li>+ setClassMonitor(): Void</li><li>+ getStudentWithAchievementByClass(): Object</li><li>+ getAdminAndCurrentMonitor(): Array,</li></ul>

**Bảng 4.10:** Lớp ClassController

Ý nghĩa các phương thức:

- + getAllClasses(): Lấy dữ liệu của tất cả các lớp học
- + getClassData(): Lấy thông tin dữ liệu của một lớp học
- + getClassSchedule(): Lấy tất cả lịch trình hoạt động của một lớp học
- + setClassMonitor(): Cập nhật vai trò cán sự lớp cho một lớp
- + getStudentWithAchievementByClass(): Lấy dữ liệu thành tích của học sinh thuộc lớp đó
- + getAdminAndCurrentMonitor(): Lấy ra danh sách các quản trị viên và cán sự lớp của lớp học hiện tại

### Lớp ReportController

ReportController

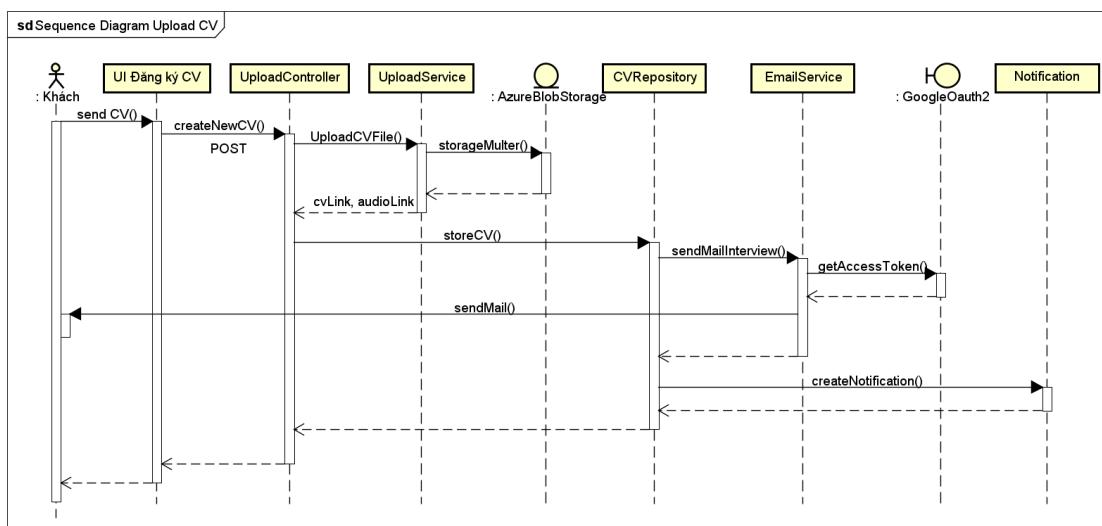
```
+ getReportByPairAndMonth(): Array
+ getReportByStudentAndMonth():
    Array
+ getReportByClassAndMonth(): Array
+ getReportByPair(): Array
```

**Bảng 4.11:** Lớp VolunteerController

Ý nghĩa các phương thức:

- + getReportByPairAndMonth(): Lấy dữ liệu báo cáo theo cặp và theo tháng
- + getReportByStudentAndMonth(): Lấy dữ liệu báo cáo theo học sinh và theo tháng
- + getReportByClassAndMonth(): Lấy dữ liệu báo cáo theo lớp và theo tháng
- + getReportByPair(): Lấy dữ liệu báo cáo theo cặp

#### d, Biểu đồ tuần tự cho chức năng “Đăng ký làm TNV”

**Hình 4.9:** Biểu đồ tuần tự cho chức năng "đăng ký làm TNV"

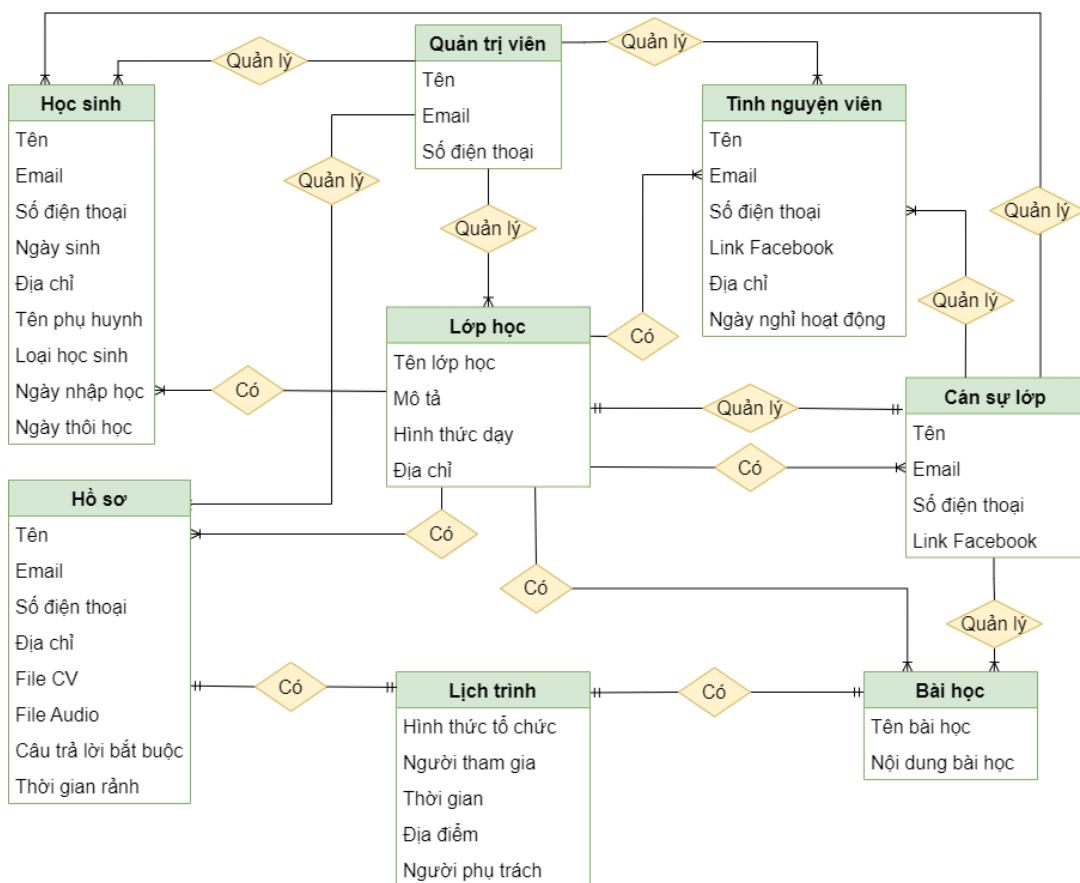
Biểu đồ tuần tự trên gồm 9 đối tượng: Khách (người dùng website đăng ký làm Tình nguyện viên), UI đăng ký CV (giao diện đăng ký), Upload Controller (đối tượng điều khiển API đăng ký làm tình nguyện viên), UploadService (Đối tượng thực hiện các yêu cầu của Controller và lưu file lên Storage), AzureAppService (đối tượng lưu trữ các file trên nền tảng đám mây), CVRespository (đối tượng thực thi các lệnh từ Controller yêu cầu, tương tác với service và lưu dữ liệu vào cơ sở dữ liệu), Email Service (đối tượng thực hiện gửi email tới người dùng), Google

Oauth2 (Đối tượng ủy quyền cho phép email được chỉ định tự động gửi email), NotificationRepository (Đối tượng thực hiện tạo mới và lưu thông báo đến người dùng).

Luồng xử lý có thể diễn tả như sau: Người dùng thao tác với giao diện đăng ký CV yêu cầu sử dụng chức năng “đăng ký làm Tình nguyện viên”. UI đăng ký CV sẽ gửi yêu cầu đăng ký đến UploadController. Sau đó, UploadController gọi đến upload service để lưu file mà người dùng gửi lên trên Azure Blob Storage. Sau khi hoàn thành lưu file, Upload Service sẽ trả về các đường link tương ứng với các file đã được lưu trên AzureBlob Storage cho Upload Controller. Lúc này, Upload Controller ra lệnh cho CVRepository lưu thông tin CV vào cơ sở dữ liệu. Sau khi CVRepository lưu CV thì nó sẽ gọi đến Email Service để gửi email thông báo với người dùng là hệ thống là đã nhận được email. Sau khi gửi email xong thì CVRepository lại gửi yêu cầu đến NotificationRepository tạo các thông báo đến những người liên quan (cụ thể là quản trị viên và cán sự lớp phụ trách lớp mà khách đăng ký). Sau khi hoàn thành tạo thông báo thì UploadController gửi đến UI thông điệp thành công. UI lúc này có nhiệm vụ thông báo cho người dùng (khách) rằng yêu cầu đã được xử lý thành công.

#### **4.2.3 Thiết kế cơ sở dữ liệu**

Thông qua các khảo sát, để có thể thấy được các thực thể, thuộc tính và mối liên hệ giữa các thực thể, trước hết em xây dựng biểu đồ thực thể - liên kết như hình dưới đây:



Hình 4.10: Biểu đồ thực thể liên kết

Từ sơ đồ thực thể liên kết, em tạo ra một cơ sở dữ liệu NoSQL bao gồm 22 bảng dữ liệu. Tuy nhiên, do giới hạn độ dài của đồ án nên em chỉ mô tả một số bảng chính như: bảng User, Student, Volunteer, Class, PairTeaching, Lesson, Schedule, Achievement, Report, CV và Notification.

#### Bảng User: Lưu thông tin về người dùng

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của người dùng
name	String	Họ và tên của người dùng
email	String	Địa chỉ email của người dùng
phoneNumber	String	Số điện thoại của người dùng
linkFacebook	String	Đường dẫn đến trang cá nhân trên mạng xã hội Facebook của người dùng
gender	Number	Giới tính của người dùng
address	ObjectId	Địa chỉ của người dùng, tham chiếu đến bảng Address

password	String	Mật khẩu đăng nhập của người dùng
image	String	Đường dẫn đến ảnh đại diện của người dùng
token	String	Chuỗi định danh người dùng
isActive	Boolean	Trạng thái tài khoản người dùng
role	Number	Chức vụ của người dùng
class	ObjectId	Lớp học mà người dùng trực thuộc, tham chiếu đến bảng Class

**Bảng 4.12:** Bảng dữ liệu User lưu các thông tin người dùng**Bảng Student:** Lưu thông tin về học sinh

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Khóa chính, id của học sinh
user	ObjectId	Thông tin người dùng của học sinh, tham chiếu đến bảng User
parentName	String	Tên bố hoặc mẹ của học sinh
interest	String	Sở thích của học sinh
character	String	Tính cách của học sinh
overview	String	Tổng quan về học sinh
birdthday	timestamp	Ngày sinh của học sinh
admissionDate	timestamp	Ngày học sinh tham gia lớp học
retirementDate	timestamp	Ngày nghỉ học của học sinh
updateBy	ObjectId	Người update trạng thái nghỉ học của học sinh, tham chiếu đến bảng User
status	Boolean	Trạng thái học tập của học sinh
studentTypes	[ObjectId]	Xếp loại học sinh, tham chiếu đến bảng StudentTypes

**Bảng 4.13:** Bảng dữ liệu Student lưu các thông tin về học sinh**Bảng Volunteer:** Lưu thông tin về tình nguyện viên

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Khóa chính, id của tình nguyện viên

user	ObjectId	Thông tin người dùng của tình nguyện viên, tham chiếu đến bảng User
role	Number	Chức vụ của Tình nguyện viên
isAdmin	Boolean	Tình nguyện viên có phải đang làm quản trị viên hay không
retirementDate	timestamp	Ngày nghỉ hoạt động của tình nguyện viên
updatedBy	ObjectId	Người cập nhật trạng thái nghỉ hoạt động của Tình nguyện viên, tham chiếu đến bảng User

**Bảng 4.14:** Bảng dữ liệu Volunteer lưu các thông tin về tình nguyện viên**Bảng Class:** Lưu thông tin dữ liệu về các lớp học

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của lớp học
name	String	Tên lớp học
description	String	Mô tả về lớp học
address	ObjectId	Địa chỉ của lớp học, tham chiếu đến bảng Address
classMonitor	ObjectId	Lớp trưởng của lớp học, tham chiếu đến bảng User
subClassMonitor	ObjectId	Lớp phó của lớp học, tham chiếu đến bảng User
studentTypes	[ObjectId]	Những đối tượng mà lớp học hướng tới, tham chiếu đến bảng StudentTypes
teachingOption	Boolean	Hình thức dạy học của lớp
defaultSchedule	[Object]	Lịch học cố định của lớp học

**Bảng 4.15:** Bảng dữ liệu Class lưu các thông tin về các lớp học của Lớp học Cầu Vồng**Bảng PairTeaching:** Lưu thông tin về các cặp dạy kèm

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của cặp dạy kèm

student	ObjectId	Học sinh đăng ký dạy kèm, tham chiếu đến bảng Student
volunteer	ObjectId	Tình nguyện viên dạy kèm, tham chiếu đến bảng Volunteer
class	ObjectId	Lớp học trực thuộc, tham chiếu đến bảng Class
address	ObjectId	Địa điểm dạy học (nếu có), tham chiếu đến bảng Address
teachOption	Boolean	Hình thức dạy
subjects	[ObjectID]	Các môn học dạy kèm, tham chiếu đến bảng Subject
grade	ObjectId	Khối lớp mà học sinh đang học, tham chiếu đến bảng Grade
noLessonPerWeek	Number	Số lượng buổi dạy mỗi tuần
status	Number	Trạng thái ghép cặp

**Bảng 4.16:** Bảng dữ liệu PairTeaching lưu thông tin về các cặp dạy kèm**Bảng Lesson:** Lưu thông tin về các bài học

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của bài học
title	String	Tên bài học
description	String	Mô tả bài học
class	ObjetId	Lớp học trực thuộc, tham chiếu đến bảng Class
schedule	ObjectId	Thời gian diễn ra buổi học, tham chiếu đến bảng Schedule
pairTeaching	ObjectId	Cặp dạy kèm, tham chiếu đến bảng pairTeaching

**Bảng 4.17:** Bảng dữ liệu Lesson lưu các thông tin về bài học

**Bảng Schedule:** Lưu thông tin về các lịch hoạt động

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của hoạt động
scheduleType	Number	Kiểu hoạt động
personIncharge	ObjectId	Người phụ trách hoạt động, tham chiếu đến bảng User
teachOption	Number	Hình thức hoạt động
linkOnline	String	Đường link online của hoạt động (nếu có)
address	ObjectId	Địa chỉ tổ chức hoạt động (nếu có), tham chiếu đến bảng Address
participants	[ObjectId]	Tình nguyện viên tham gia hoạt động, tham chiếu đến bảng User
time	NestedObject	Thời gian diễn ra hoạt động

**Bảng 4.18:** Bảng dữ liệu Schedule lưu các thông tin về lịch hoạt động

**Bảng Achievement:** Lưu thông tin về thành tích của học sinh

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của thành tích
student	ObjectId	Học sinh, tham chiếu đến bảng Student
lesson	ObjectId	Bài học tương ứng, tham chiếu đến bảng Lesson
point	Number	Điểm số
comment	String	Nhận xét của tình nguyện viên

**Bảng 4.19:** Bảng dữ liệu Achievement lưu các thông tin về thành tích của học sinh

**Bảng Report:** Lưu thông tin báo cáo của tình nguyện viên sau mỗi buổi học

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của báo cáo
lessonDescription	String	Mô tả tổng quan buổi học

createdBy	ObjectId	Người tạo báo cáo, tham chiếu đến bảng User
achievement	ObjetId	Thành tích tương ứng, tham chiếu đến bảng Achievement
subject	ObjectId	Môn học, tham chiếu đến bảng Subject
pairTeaching	ObjectId	Cặp dạy kèm, tham chiếu đến bảng pairTeaching

**Bảng 4.20:** Bảng dữ liệu Report lưu các thông tin báo cáo sau buổi học**Bảng CV:** Lưu thông tin hồ sơ đăng ký làm Tình nguyện viên

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id hồ sơ đăng ký
email	String	Email của người đăng ký
phoneNumber	String	Số điện thoại của người đăng ký
cvFileLink	String	Đường dẫn đến file CV của người đăng ký
audioFileLink	String	Đường dẫn đến file audio giới thiệu bản thân bằng tiếng Anh của người đăng ký
schedule	ObjectId	Thời gian phỏng vấn, tham chiếu đến bảng Schedule
status	Number	Trạng thái của hồ sơ
note	String	Ghi chú của người nộp hồ sơ
class	ObjectId	Lớp học mà người đăng ký muốn tham gia làm Tình nguyện viên

**Bảng 4.21:** Bảng dữ liệu CV lưu các hồ sơ đăng ký làm tình nguyện viên**Bảng Notification:** Lưu các thông báo của mọi người dùng

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectID	Khóa chính, id của thông báo
user	ObjectId	Người dùng nhận thông báo, tham chiếu đến bảng User
type	Number	Kiểu thông báo

read	Boolean	trạng thái đã đọc thông báo
content	NestedObject	Nội dung thông báo

**Bảng 4.22:** Bảng dữ liệu Notification lưu các thông báo

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Để hoàn thành hệ thống, em đã sử dụng các thư viện và công cụ hỗ trợ sau đây:

Mục đích	Công cụ	Địa chỉ URL
Cơ sở dữ liệu	MongoDB	<a href="https://www.mongodb.com">https://www.mongodb.com</a>
Frontend Framework	ReactJS	<a href="https://reactjs.org">https://reactjs.org</a>
BackendFramework	ExpressJS	<a href="https://expressjs.com">https://expressjs.com</a>
Thư viện	Ant Design	<a href="https://ant.design">https://ant.design</a>
	Axios	<a href="https://axios-http.com">https://axios-http.com</a>
	Formik	<a href="https://formik.org">https://formik.org</a>
	I18next	<a href="https://www.i18next.com">https://www.i18next.com</a>
	Nodemon	<a href="https://nodemon.io">https://nodemon.io</a>
	Nodemailer	<a href="https://nodemailer.com">https://nodemailer.com</a>
	Bcrypt	<a href="https://www.npmjs.com/package/bcrypt">https://www.npmjs.com/package/bcrypt</a>
	Cookie-parser	<a href="https://www.npmjs.com/package/cookie-parser">https://www.npmjs.com/package/cookie-parser</a>
	Socket.io	<a href="https://socket.io/">https://socket.io/</a>
Storage	Azure Blob Storage	<a href="https://azure.microsoft.com/en-us/services/storage/blobs/">https://azure.microsoft.com/en-us/services/storage/blobs/</a>
Oauth2	Google Oauth2	<a href="https://developers.google.com/identity/protocols/oauth2">https://developers.google.com/identity/protocols/oauth2</a>
IDE Lập trình	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>
Công cụ quản lý source code	Git/Github	<a href="https://git-scm.com/">https://git-scm.com/</a> <a href="https://github.com/">https://github.com/</a>

**Bảng 4.23:** Thư viện và công cụ sử dụng

#### 4.3.2 Kết quả đạt được

Kết quả đạt được là hệ thống gồm hai website và một Web server ExpressJS là máy chủ quản lý dữ liệu của hệ thống:

**Website quản lý** dành cho quản trị viên, cán sự quản lý lớp, tình nguyện viên và học sinh được deploy thử nghiệm trên đường link: <http://the-rainbow-class-1.japaneast.cloudapp.azure.com:8088/>

**Website hỗ trợ đăng ký làm tình nguyện viên** hướng tới đối tượng là những người có hứng thú với Lớp học Cầu Vồng, và mong muốn trở thành tình nguyện viên của tổ chức được deploy thử nghiệm với đường link: <http://the-rainbow-class-1.japaneast.cloudapp.azure.com:8080/>

Module	Dung lượng mã nguồn	Sản phẩm đóng gói	Dung lượng sản phẩm đóng gói
Website quản lý	1.23 MB	Không có	Không có
Website hỗ trợ đăng ký làm Tình nguyện viên	3.74MB	Không có	Không có
Web server ExpressJS	205KB	Không có	Không có

**Bảng 4.24:** Kết quả đóng gói sản phẩm

Sau khi hoàn thành các chức năng cơ bản của hệ thống, em đã thực hành khảo sát chất lượng sản phẩm với thành viên quản lý của Lớp học Cầu Vồng<sup>2</sup>. Kết quả khảo sát dựa trên đánh giá về mặt giao diện và độ hoàn thiện các chức năng của hệ thống. Dưới đây là kết quả chi tiết:

#### Về website hỗ trợ đăng ký làm tình nguyện viên:

- Mức độ bắt mắt, thân thiện, dễ sử dụng: 5/5
- Tốc độ phản hồi của hệ thống: 4/5
- Mức độ hoàn thiện chức năng: 4/5
- Mức độ đầy đủ các trường thông tin cần thiết tại form đăng ký ứng tuyển làm tình nguyện viên: 5/5

#### Về website quản lý dạy học và nhân sự:

- Mức độ thân thiện, dễ sử dụng: 4/5
- Mức độ đầy đủ các chức năng cần thiết: 4/5
- Mức độ hoàn thiện chức năng: 4/5
- Mức độ phản hồi của hệ thống: 4/5

<sup>2</sup>Kết quả khảo sát: <https://sal.vn/zsCEAX>

### 4.3.3 Minh họa các chức năng chính

#### a, Website hỗ trợ đăng ký làm Tình nguyện viên

Màn hình giới thiệu về tổ chức trên mobile và desktop:



Hình 4.11: Màn hình giới thiệu tổ chức trên mobile và trên desktop

Tại đây, người dùng có thể xem được toàn bộ các thông tin công khai, giới thiệu về tổ chức Lớp học Cầu Vồng.

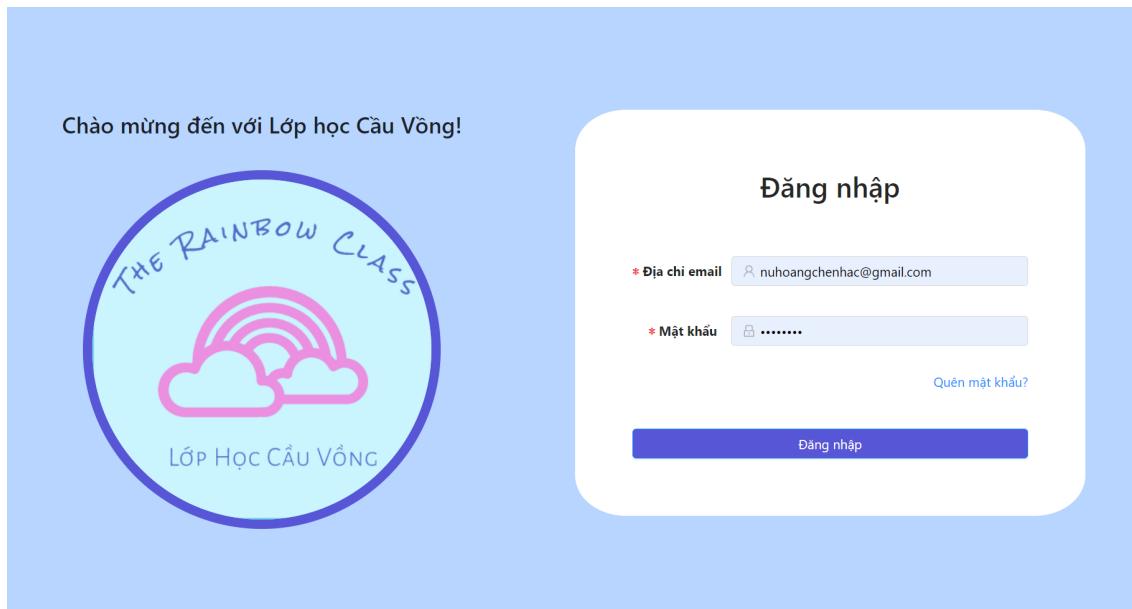
#### Màn hình hỗ trợ đăng ký làm TNV trên mobile và desktop:

Hình 4.12: Màn hình hỗ trợ đăng ký làm TNV trên mobile và trên desktop

Tại đây, người dùng có thể điền form để đăng ký làm Tình nguyện viên. Màn hình đăng ký gồm có 3 trang chính: (i) trang điền thông tin cá nhân và lớp học đăng ký, (ii) trang điền các câu trả lời cho câu hỏi bắt buộc được quản trị viên thiết lập sẵn, (iii) trang điền thông tin về thời gian rảnh của ứng viên, tải lên file audio giới thiệu bản thân bằng tiếng Anh và file CV.

### b, Website quản lý

#### Màn hình đăng nhập vào hệ thống

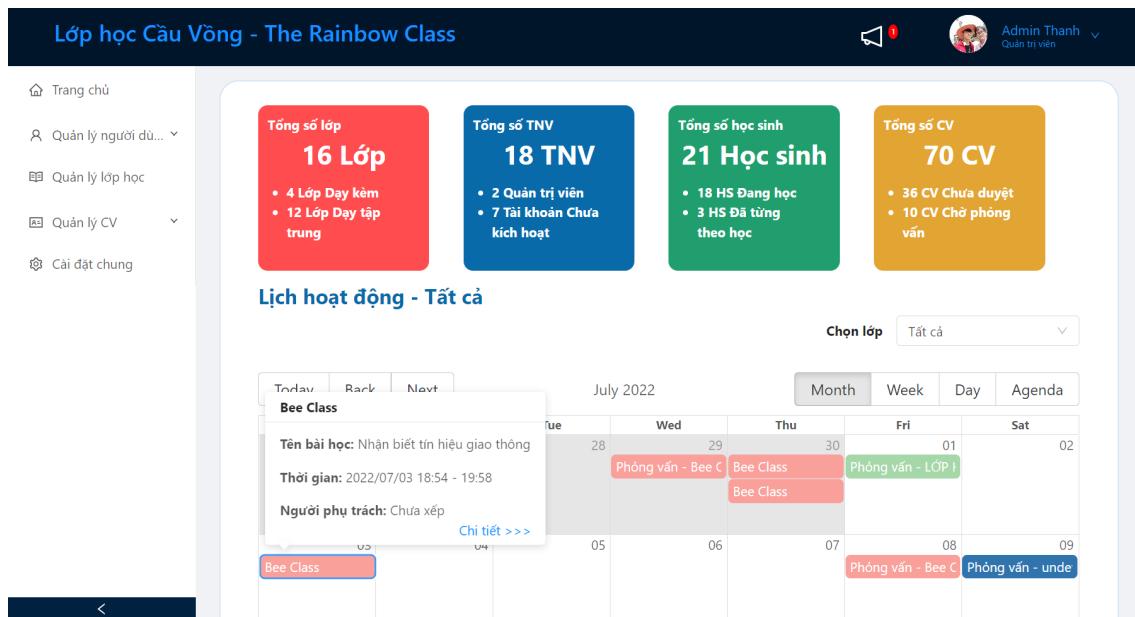


**Hình 4.13:** Màn hình đăng nhập trên desktop

Tại đây, người dùng nhập các thông tin như email, mật khẩu để đăng nhập vào hệ thống. Trong trường hợp người dùng quên mật khẩu, khi ấn vào nút "Quên mật khẩu?" sẽ có một popup hiển thị yêu cầu người dùng nhập vào địa chỉ email của mình để đặt lại mật khẩu. Sau khi người dùng nhấn nút submit thì hệ thống sẽ gửi mail đặt lại mật khẩu đến tài khoản email đó.

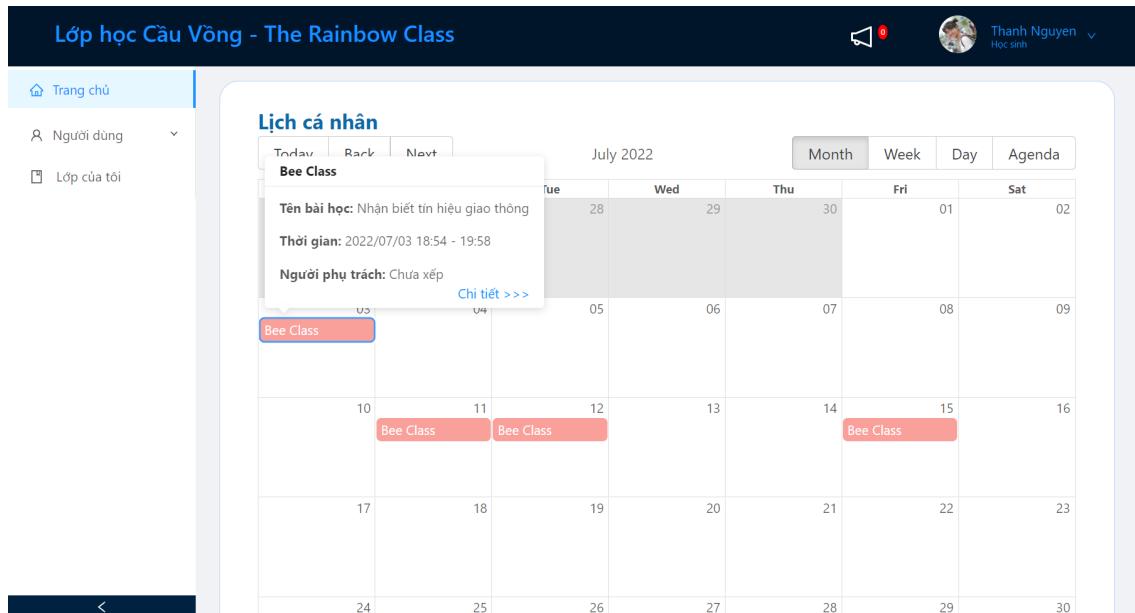
**Màn hình dashboard (chức vụ quản trị viên)** giúp người quản trị viên nắm được những thông tin tổng thể của lớp học, bao gồm hai thành phần chính: Phần thống kê (nằm phía trên) có thống kê tổng số và phân loại lớp học, số Tình nguyện viên đã và đang hoạt động dạy học tại tổ chức, số Học sinh đã và đang học, số lượng CV ở trạng thái chưa duyệt và chờ phỏng vấn. Người dùng là quản trị viên có thể truy cập nhanh đến các màn hình tương ứng.

Phần tổng quan lịch học các lớp (nằm phía bên dưới) là một lịch tổng quan các hoạt động, sự kiện diễn ra tại các lớp học trực thuộc tổ chức. Các hoạt động sự kiện đó có thể là lịch dạy học hay lịch phỏng vấn. Quản trị viên có thể chọn xem lịch dạy của tất cả các lớp hoặc của từng lớp học thuộc tổ chức thông qua select box. Lịch hoạt động của mỗi lớp học được quy định bằng một màu sắc khác nhau giúp cho người quản trị có thể nắm được một cách tổng quát.



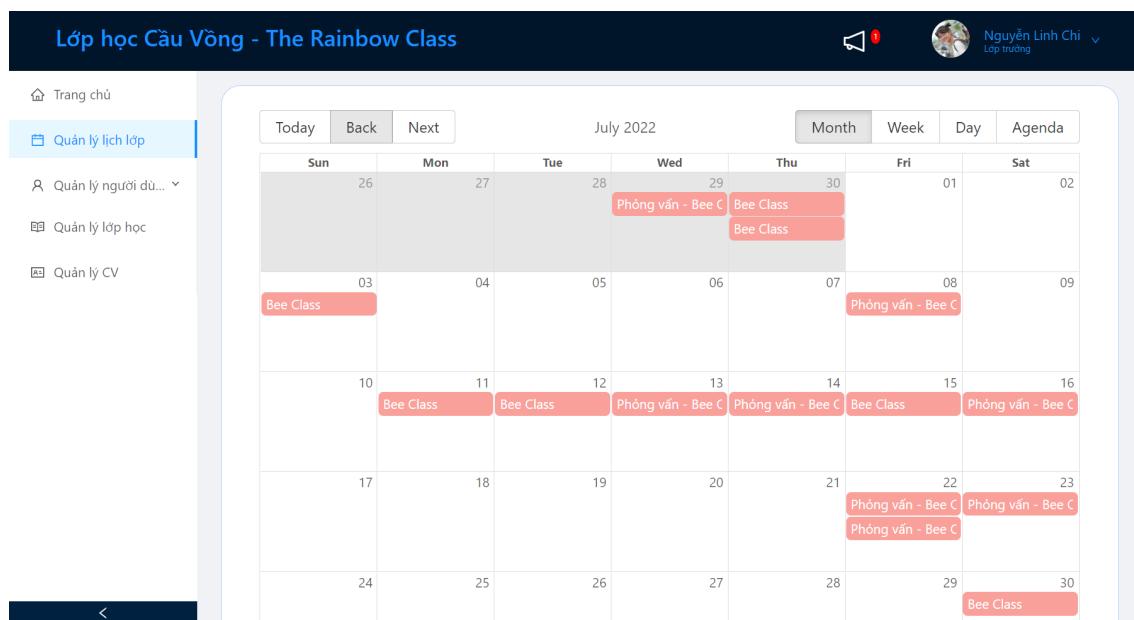
**Hình 4.14:** Màn hình Dashboard của người dùng có chức vụ là quản trị viên

**Màn hình dashboard (cho chức vụ tình nguyện viên, cán sự lớp, học sinh)** hiển thị lịch hoạt động cá nhân của người dùng. Đó có thể là lịch dạy học, lịch phỏng vấn (đối với tình nguyện viên hoặc cán sự lớp) hay lịch học cá nhân (đối với học sinh).



**Hình 4.15:** Màn hình Dashboard của người dùng có chức vụ là học sinh

**Màn hình quản lý lịch lớp (cho chức vụ cán sự lớp)** hiển thị toàn bộ lịch hoạt động của lớp học mà người dùng là cán sự lớp (Lớp trưởng hoặc lớp phó) đang quản lý. Các hoạt động ở đây bao gồm lịch phỏng vấn và lịch dạy học.



**Hình 4.16:** Màn hình Quản lý lịch lớp của người dùng có chức vụ là cán sự lớp

**Chức năng quản lý người dùng (dành cho người dùng có chức vụ là quản trị viên hoặc cán sự lớp)** chia người dùng thành ba đối tượng chính là quản trị viên tinh nguyện viên và học sinh. Tương ứng với mỗi đối tượng tinh nguyện viên và học sinh sẽ có các màn danh sách, thêm, sửa, chi tiết thông tin tương ứng. Riêng đối với đối tượng là quản trị viên thì người dùng chỉ có thể xem được danh sách các quản trị viên.

Dưới đây là các hình minh họa cho màn hình danh sách Quản trị viên (Hình 4.17), Màn hình danh sách Tình nguyện viên (Hình 4.18), Màn hình danh sách Học sinh (Hình 4.19)

Họ và tên	Số điện thoại	Địa chỉ email
Admin Thanh	0382195410	thanh.nth17687@sis.hust.edu.vn
Hoàng Thị Thanh Hằng	0986202375	beststu@gmail.com

**Hình 4.17:** Màn hình danh sách Quản trị viên

Họ và tên	Tên lớp	Số điện thoại	Địa chỉ email
Nguyễn Thị Mến (LP)	Lớp học tiếng Anh bổ sung đặc biệt	0322343723	mennguyen2011@gmail.com
Nguyễn Công Hậu	LỚP HỌC ANH SÁNG KIM NGƯỜU	04382374342	haucong2701@gmail.com
Triệu Thị Thảo	Lớp học tiếng Anh bổ sung đặc biệt	0382195410	abcvolunteer@gmail.com
Nguyễn Thị Thùy (LT)	Lớp học tiếng Anh bổ sung đặc biệt	0382195410	thuy.nth289382938@gmail.com
Đào Đăng Huy	Lớp học tiếng Anh bổ sung đặc biệt	012382364	danghuy2304@gmail.com

**Hình 4.18:** Màn hình danh sách tình nguyện viên

Họ và tên	Tên lớp	Số điện thoại	Đối tượng	Trạng thái
Nguyễn Thị Trúc	Lớp học tiếng Anh bổ sung đặc biệt	093293829	Học sinh khuyết tật, học sinh nghèo	Đang học
Nguyễn Văn Nguyên	Lớp học tiếng Anh bổ sung đặc biệt	0987654312	Học sinh khuyết tật, Học sinh khiêm thính, Học sinh bại não	Đang học
Lương Đào Quang Anh	Lớp học tiếng Anh bổ sung đặc biệt	0382195410	học sinh nghèo	Đang học
Nguyễn Thị Thúy Hiền	Lớp học tiếng Anh bổ sung đặc biệt	0382195410	học sinh nghèo, Học sinh khiêm thính, Học sinh bại não	Đang học
Nguyễn Thu Phương	Lớp học tiếng Anh bổ sung đặc biệt	0382195410	Học sinh chậm phát triển	Đang học

**Hình 4.19:** Màn hình danh sách học sinh

**Màn hình chi tiết thông tin tình nguyện viên** hiển thị các thông tin cơ bản về tình nguyện viên và trạng thái hoạt động của tình nguyện viên đó. Nếu đã được nhập vào ngày nghỉ hoạt động thì sẽ hiển thị thêm ngày nghỉ hoạt động và thông tin về người đã cập nhật thông tin về ngày nghỉ hoạt động của tình nguyện viên đó.

**Thông tin chi tiết Tình nguyện viên**

Họ và tên: Triệu Thị Thảo  
Địa chỉ email: abcvolunteer@gmail.com  
Địa chỉ: Trại Chuồng, Tân Phong, Ninh Giang, Hải Dương  
Số điện thoại: 0382195410  
Link Facebook: <https://www.facebook.com/thao.trieu.1069020>  
Chức vụ: Tình nguyện viên - Lớp học tiếng Anh bổ sung đặc biệt

**Hình 4.20:** Màn hình chi tiết thông tin TNV

**Chức năng quản lý lớp học (cho chức vụ quản trị viên và cán sự lớp)** bao gồm các màn hình như: danh sách các lớp học, thêm, sửa, chi tiết lớp học. Tại màn chi tiết lớp học thì được chia ra thành các tab khác nhau, tùy theo chức vụ người dùng hay hình thức dạy học của lớp (dạy tập trung hay dạy kèm 1-1) mà cách hiển thị cũng khác nhau. Ví dụ, ở vai trò quản trị viên màn chi tiết thông tin lớp học theo hình thức dạy tập trung sẽ có 3 tab: thông tin cơ bản, danh sách các bài học, thành tích của lớp học (Hình 4.21); tuy nhiên cũng ở vai trò quản trị viên màn chi tiết thông tin lớp học theo hình thức dạy kèm 1-1 sẽ hiển thị 2 tab là thông tin cơ bản và danh sách các cặp dạy kèm (Hình 4.22).

**Chi tiết lớp học - Bee Class**

Tên học sinh	Trung bình	Thành tích tháng	Chọn tháng
Thanh Nguyen	9	9 🎯	2022/06
Hà Văn Kỳ	10	10 🎯	-
Nguyễn Văn Minh	6.5	5 🎯	8 🎯
Bùi Tuấn Thành	7.5	5 🎯	10 🎯

**Hình 4.21:** Màn hình chi tiết lớp học theo kiểu dạy tập trung - tab thành tích

Tên học sinh	TNV phụ trách	Hình thức	Khối	Môn học	Số buổi	Hành động
Nguyễn Thị Thúy	Chưa được xếp ↴	offline	Lớp 2	Tiếng Anh, Toán	0	
Lương Đào Quang Anh	Chưa được xếp ↴	offline	Lớp 2	Tin học	0	
Nguyen Duy Long	Triệu Thị Thảo ↴	online	Lớp 2	Vật lý	0	
Nguyen Quang Phuc	Đào Đăng Huy ↴	online	Lớp 2	Tiếng Anh	0	

**Hình 4.22:** Màn hình chi tiết lớp học theo kiểu dạy kèm - tab danh sách dạy kèm

**Màn hình danh sách CV** hiển thị các thông tin cơ bản về hồ sơ đăng ký làm tình nguyện viên như thông tin cá nhân người gửi đăng ký, lớp học họ đăng ký, thời gian tạo và trạng thái duyệt của hồ sơ đó. Người dùng có thể tìm kiếm CV theo tên, địa chỉ email hoặc số điện thoại của người đăng ký hoặc có thể lọc theo trạng thái CV, lớp học đăng ký. Dưới đây là hình ảnh minh họa cho những hồ sơ có trạng thái bị loại:

Họ và tên	Số điện thoại	Địa chỉ email	Lớp học đăng kí	Trạng thái	Thời gian tạo
Nguyễn Thị Hải Thanh	0382195410	nuhoangchenhac@...	Bee Class		2022/03/06
Nguyễn Quang Long	0382195410	canhcuadong@gma...	Bee Class		2022/03/06
Nguyễn Thị Ngọc Ánh	123232	msanh.bk@gmail.com	Bee Class		2022/03/21
Đào Đăng Huy	123232	huydaodang.bk@g...	Bee Class		2022/03/21
Nguyễn Thị Thúy	0293823	thuynt20329@sun...	Bee Class		2022/06/01
Vũ Thị Dung	0382195410	leader.tn.tship@gm...	Bee Class		2022/06/05

**Hình 4.23:** Màn hình danh sách những hồ sơ bị loại

**Chức năng Cài đặt chung** hỗ trợ quản trị viên chỉnh sửa, cài đặt các thông tin chung như: phân loại học sinh, môn học, kỳ học, khối lớp mà học sinh đang theo

học. Các thông tin chung được chia thành các tab. Quản trị viên có thể thực hiện các thao tác thêm, sửa, xóa những thông tin chung ấy.

**Hình 4.24:** Màn hình Cài đặt chung - tab quản lý các môn học

**Chức năng gửi email:** Mỗi khi người dùng kích hoạt tài khoản hoặc đặt lại mật khẩu thì hệ thống sẽ gửi một email đến địa chỉ email của người dùng kèm theo một đường link dẫn đến trang đặt lại mật khẩu.



## Active account

Hi Thanh Nguyen!  
Please [click here](#) to active your account.  
Thanks,  
The Rainbow Class

**Hình 4.25:** Email mà người dùng nhận được để kích hoạt tài khoản

### 4.4 Kiểm thử

Do hệ thống có nhiều chức năng nên tại đây, em sẽ trình bày nội dung kiểm thử của 3 chức năng chính là chức năng đăng ký tình nguyện viên, chức năng duyệt hồ sơ đăng ký làm Tình nguyện viên và chức năng quản lý bài học(với lớp dạy theo hình thức tập trung).

#### Kiểm thử chức năng đăng ký làm tình nguyện viên

STT	Kịch bản kiểm thử	Kết quả mong muốn
1	Tại màn hình tổng quan lớp học, người dùng ấn nút "Nộp CV"	Chuyển hướng đến màn hình form đăng ký làm tình nguyện viên. Màn hình form đăng ký làm tình nguyện viên bao gồm ba trang. Trang hiển thị mặc định yêu cầu người dùng điền tên, email, số điện thoại và lớp học đăng ký.
2	Người dùng điền đầy đủ thông tin tên, email, số điện thoại thì người dùng bấm vào nút "Tiếp theo" để chuyển sang trang thứ hai.	Hiển thị trang thứ hai là một form bao gồm tất cả các câu hỏi mà đã được quản trị viên cài đặt sẵn.
3	Người dùng điền đầy đủ các câu trả lời cho các câu hỏi tại trang thứ hai và bấm nút "Tiếp theo" để chuyển đến trang cuối	Hiển thị trang cuối là một form yêu cầu người dùng tải lên file audio dưới dạng mp3 hoặc ghi âm trực tiếp, CV dưới dạng PDF, chọn khung lịch rảnh và ghi chú thêm.
4	Tại trang cuối cùng, chọn ghi âm audio bằng cách record trực tiếp thông qua trình duyệt, sau đó ấn nút "save" để lưu bản ghi âm.	Sau khi ấn nút "save" thì hiển thị thông báo ghi âm thành công.
5	Tại trang cuối cùng, chọn tải lên tệp âm thanh từ máy tính cá nhân, người dùng tải lên tệp có định dạng MP3 từ máy tính cá nhân	Tải file lên thành công
6	Người dùng gửi form sau khi đã điền đầy đủ các trường thông tin bắt buộc	Chuyển hướng đến trang cảm ơn và thông báo đến người dùng đã ghi lại hồ sơ người dùng trong cơ sở dữ liệu, nhắc nhở người dùng thường xuyên kiểm tra email để cập nhật lịch phỏng vấn

**Bảng 4.25:** Kịch bản kiểm thử cho chức năng đăng ký làm tình nguyện viên

### Kiểm thử chức năng duyệt CV

STT	Kịch bản kiểm thử	Kết quả mong muốn
1	<p>Chuyển đổi trạng thái CV thành “trượt”:</p> <ol style="list-style-type: none"> <li>1. Chọn một CV đang ở trạng thái "Chờ phỏng vấn" hoặc "Chờ duyệt".</li> <li>2. Chọn button thay đổi trạng thái CV.</li> <li>3. Chuyển trạng thái CV thành "Trượt"</li> </ol>	<p>1. Khi chọn option “Trượt” thì hiển thị popup xác nhận đổi trạng thái. Nếu người dùng ấn “Cancel” thì trạng thái của CV không bị thay đổi. Nếu người dùng ấn “Submit” thì trạng thái CV thay đổi thành “Trượt”.</p> <p>2. Sau khi submit thì không thể thay đổi trạng thái, người đăng ký nhận được email thông báo trượt</p>
2	<p>Chuyển đổi trạng thái CV từ “chờ duyệt” thành “Chờ phỏng vấn”:</p> <ol style="list-style-type: none"> <li>1. Chọn một CV đang ở trạng thái "Chờ duyệt".</li> <li>2. Chọn button thay đổi trạng thái CV</li> <li>3. Chuyển trạng thái CV thành "Chờ phỏng vấn"</li> </ol>	<p>1. Khi chọn option “Chờ phỏng vấn” thì hiển thị popup điền thông tin thời gian, link phỏng vấn online và người phụ trách. Nếu người dùng ấn “Cancel” thì trạng thái của CV không bị thay đổi. Nếu người dùng ấn “Submit” thì trạng thái CV thay đổi thành “Chờ phỏng vấn”.</p> <p>2. Sau khi submit, người phụ trách phỏng vấn nhận được thông báo, người đăng ký thì nhận được email chứa các thông tin phỏng vấn (thời gian và link phỏng vấn)</p>

3	<p>Chuyển trạng thái CV từ “chờ phỏng vấn” sang “Đỗ”:</p> <ol style="list-style-type: none"> <li>1. Chọn một CV đang ở trạng thái "Chờ phỏng vấn".</li> <li>2. Chọn button thay đổi trạng thái CV</li> <li>3. Chuyển trạng thái CV thành "Đỗ"</li> </ol>	<p>1. Khi chọn option “Đỗ” thì hiển thị popup xác nhận. Nếu người dùng ấn “Cancel” thì trạng thái của CV không bị thay đổi. Nếu người dùng ấn “Submit” thì trạng thái CV thay đổi thành “Đỗ”.</p> <p>2. Hệ thống kiểm tra email của người đã phỏng vấn xem có bị trùng với email đã tồn tại trong hệ thống chưa.</p> <ol style="list-style-type: none"> <li>2.1. Nếu không trùng thì tự động tạo tài khoản tình nguyện viên mới bằng email đó, gửi email thông báo đỗ phỏng vấn kèm theo link kích hoạt tài khoản đến cho người đỗ phỏng vấn.</li> <li>2.2. Nếu email đã tồn tại trong hệ thống thì hiển thị popup báo đến quản trị viên/cán sự lớp là không thể tự động tạo tài khoản mới, gửi email thông báo đỗ phỏng vấn đến cho người đỗ phỏng vấn</li> </ol>
---	--	--

**Bảng 4.26:** Kịch bản kiểm thử cho chức năng duyệt CV**Kiểm thử chức năng quản lý bài học theo hình thức dạy tập trung**

<b>STT</b>	<b>Kịch bản kiểm thử</b>	<b>Kết quả mong muốn</b>
1	<p>Tạo mới một bài học:</p> <ol style="list-style-type: none"> <li>1.Tại màn hình danh sách bài học, ấn vào nút tạo mới bài học</li> <li>2.Điền đầy đủ các thông tin cho bài học rồi submit</li> </ol>	<ol style="list-style-type: none"> <li>1. Chỉ hiển thị button tạo mới bài học với role Admin/Cán sự lớp</li> <li>2. Sau khi điền thông tin và submit, hệ thống thông báo tạo mới bài học thành công</li> <li>3. Bài học mới được hiển thị trong danh sách bài học</li> <li>4. Các Tình nguyện viên thuộc lớp học đó sẽ nhận được thông báo</li> </ol>

2	<p>Đăng ký/hủy đăng ký dạy học :</p> <ol style="list-style-type: none"> <li>1.Tại màn hình danh sách bài học, ấn vào xem chi tiết bài học</li> <li>2.Tại màn hình chi tiết bài học, người dùng ấn nút đăng ký/hủy đăng ký tham gia dạy học</li> </ol>	<ol style="list-style-type: none"> <li>1. Nút "đăng ký"/"hủy đăng kí" chỉ hiển thị với chức vụ Cán sự lớp và Tình nguyện viên</li> <li>2. Nếu độ chênh lệch thời gian từ hiện tại tới lúc diễn ra bài học nhỏ hơn 24h hoặc người dùng hiện tại đã được phân công phụ trách bài học thì disable button đăng ký/hủy đăng ký</li> <li>3. Nếu độ chênh lệch thời gian từ hiện tại tới lúc diễn ra bài học lớn hơn 24h thì enable button đăng ký/hủy đăng ký.</li> <li>3.1. Người dùng ấn nút đăng ký thì thông báo đăng ký thành công, tên của người dùng đó hiển thị trong danh sách những người tham gia dạy học</li> <li>3.2. Người dùng ấn nút hủy đăng ký thì thông báo hủy đăng ký thành công, tên của người dùng đó không còn được hiển thị trong danh sách những người tham gia dạy học</li> </ol>
3	<p>Chọn người phụ trách bài học:</p> <ol style="list-style-type: none"> <li>1.Tại màn hình chi tiết bài học, nhấn nút phân công phụ trách</li> <li>2. Chọn người phụ trách bài học rồi submit</li> </ol>	<ol style="list-style-type: none"> <li>1. Nút phân công phụ trách chỉ hiển thị với người dùng là Cán sự lớp</li> <li>2. Danh sách dropdown chứa tên những tình nguyện viên đã tham gia đăng ký dạy</li> <li>3. Sau khi submit thì thông báo chọn người phụ trách thành công, trên màn hình hiển thị tên người phụ trách</li> <li>4. Người được phân công phụ trách nhận được thông báo</li> </ol>

**Bảng 4.27:** Kịch bản kiểm thử cho chức năng quản lý bài học (lớp dạy theo hình thức tập trung)

## 4.5 Triển khai

Hệ thống được đóng gói thành 3 phần lần lượt là Trang web quản lý, Trang web giới thiệu và hỗ trợ đăng ký làm tình nguyện viên, API Service.

**Thuê và cài đặt server:** Em chọn thuê server trên Portal Azure, hệ thống được

vận hành bởi máy ảo Ubuntu với cấu hình 2 CPU, 4G RAM và chạy hệ điều hành Ubuntu.

Các bước triển khai:

- Tải code từ github
- Install Nginx và pm2
  - Vào thư mục API service, install các thư viện cần thiết, cấu hình file .env và khởi động chạy pm2
    - Vào thư mục chứa source code trang quản lý, install các thư viện cần thiết, cấu hình file .env và build hệ thống. Sau đó config file conf.d tại thư mục ./etc/nginx, chọn đường dẫn đến file build
    - Vào thư mục chứa source code trang đăng ký làm tình nguyện viên, install các thư viện cần thiết, cấu hình file .env và build hệ thống. Sau đó config file conf.d tại thư mục ./etc/nginx, chọn đường dẫn đến file build
  - Khởi động lại nginx service

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

### 5.1 Hỗ trợ cải tiến quy trình thu thập hồ sơ đăng ký tình nguyện viên

#### 5.1.1 Cách thức tuyển tình nguyện viên khi chưa sử dụng hệ thống

Cách thức tuyển tình nguyện viên của tổ chức khi không sử dụng hệ thống là đăng bài tuyển tình nguyện viên trên fanpage chính thức của Lớp học Cầu Vồng trên trang mạng xã hội Facebook. Mọi bài đăng tuyển tình nguyện viên cho các lớp học có cấu trúc như nhau, bao gồm 6 phần chính:

- (i) Giới thiệu về Lớp học Cầu Vồng và lý do cần tuyển tình nguyện viên
- (ii) Liệt kê các lớp học cần tuyển tình nguyện viên, số lượng tình nguyện viên cần tuyển, hình thức dạy học, môn học, địa chỉ, thời gian dạy.
- (iii) Yêu cầu chung khi trở thành tình nguyện viên của Lớp học Cầu Vồng
- (iv) Quyền lợi của tình nguyện viên khi tham gia Lớp học Cầu Vồng
- (v) Cách thức đăng ký trở thành tình nguyện viên của Lớp học Cầu Vồng.
- (vi) Một số lưu ý khác khi tham gia làm thành viên của Lớp học Cầu Vồng.

Trong đó, mục (v) **cách thức đăng ký trở thành tình nguyện viên** là mục em chú trọng cải tiến quy trình nghiệp vụ. Cụ thể quy trình như sau: các bạn có mong muốn trở thành tình nguyện viên của Lớp học Cầu Vồng phải gửi CV cá nhân và câu trả lời cho các câu hỏi mà ban quản lý Lớp học Cầu Vồng đưa ra tới địa chỉ email: lophoccauvong15@gmail.com; với tiêu đề: "[The Rainbow Class] Vị Trí Ứng Tuyển-Họ Tên" (Ví dụ: [The Rainbow Class] Lớp Học Phúc Xá-Nguyễn Thị A). Nếu đơn đăng ký không có CV hoặc thiếu câu trả lời sẽ bị loại.

##### CÁCH THỨC ĐĂNG KÝ:

Gửi CV cá nhân và câu trả lời cho các câu hỏi dưới đây tới Email: lophoccauvong15@gmail.com với tiêu đề: "[The Rainbow Class] Vị Trí Ứng Tuyển-Họ Tên" (Ví dụ: [The Rainbow Class] Lớp Học Phúc Xá-Nguyễn Thị A):

1. Bạn đã từng tham gia những hoạt động thiện nguyện nào?
  2. Tại sao bạn lựa chọn công việc ở Lớp học Cầu Vồng? Ý nghĩa của làm việc tốt với bạn là gì?
  3. Bạn có đảm bảo rằng mình sẽ nhiệt tình, sẵn sàng tham gia cả những hoạt động khác của lớp học Cầu Vồng như quyền quỹ hay là tổ chức sự kiện, ngoài việc đồng hành cùng lớp học mà bạn đã đăng kí? Hãy nêu ra một hoặc nhiều ví dụ thể hiện bạn là người nhiệt tình trong các công việc thiện nguyện.
  4. Bạn có nhu cầu học tiếng anh giao tiếp với mức giá rẻ không?
  5. [Dành riêng cho TNV ứng tuyển dạy tiếng anh] Hãy gửi bản ghi âm ngắn bằng tiếng anh giới thiệu về bản thân.
- X** Tất cả các đơn đăng ký không có CV hoặc thiếu câu trả lời sẽ bị LOẠI. CV có thông tin link Facebook cá nhân. Các bạn có thể tham khảo mẫu CV ở link sau:

**Hình 5.1:** Cách thức đăng ký trở thành TNV của Lớp học Cầu Vồng<sup>1</sup>

<sup>1</sup>Tham khảo bài đăng tuyển tình nguyện viên đầy đủ tại đường dẫn <https://www.facebook.com/therainbowclass/photos/a.184942138578408/1320108475061763/>

Chính vì tất cả các CV của người đăng ký đều được gửi về chung một địa chỉ email nên công việc quản lý, theo dõi và lọc các CV sẽ trở nên phức tạp và tốn thời gian đối với các quản trị viên (ban nhân sự) nói chung và quản lý từng lớp học nói riêng.

### 5.1.2 Hỗ trợ cải tiến quy trình thu thập hồ sơ đăng ký tình nguyện viên

Hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng đã hỗ trợ cải tiến và hệ thống hóa quy trình thu thập hồ sơ đăng ký làm Tình nguyện viên. Quy trình được mô hình hóa tại hình 2.10, cụ thể như sau:

**Website hỗ trợ đăng ký làm tình nguyện viên** có chức năng chính là thu thập các hồ sơ đăng ký của người có nguyện vọng trở thành Tình nguyện viên của Lớp học Cầu Vồng.

Với **tính năng xác thực các trường bắt buộc trước khi gửi form**, hệ thống yêu cầu người dùng phải điền đầy đủ các thông tin cá nhân (như tên, email, số điện thoại), thông tin lớp học đăng ký, trả lời tất cả các câu hỏi mà Quản trị viên đã cài đặt sẵn trên website quản lý và bắt buộc upload file CV cá nhân dưới dạng PDF. Điều này sẽ giúp quản trị viên và người duyệt CV tránh được những trường hợp phải mất thời gian duyệt và loại những CV không có đầy đủ thông tin, giúp nâng cao chất lượng cho các hồ sơ gửi về hệ thống. Bên cạnh đó, website cũng **bắt buộc người đăng ký phải điền vào khung giờ rảnh** của mình để hỗ trợ người duyệt hồ sơ có thể chủ động sắp xếp lịch phỏng vấn một cách dễ dàng và chủ động hơn.

Một tính năng giúp quản trị viên có thể linh động hơn trong việc đặt ra các câu hỏi bắt buộc chính là **chức năng cài đặt các câu hỏi phỏng vấn** trong phần quản lý CV.

Lớp học Cầu Vồng - The Rainbow Class

Trang chủ

Quản lý người dùng

Quản lý lớp học

Quản lý CV

Danh sách CV

Câu hỏi phỏng vấn

Cài đặt chung

Admin Thành Quản trị viên

Câu hỏi phỏng vấn

Câu hỏi 1 (Không bắt buộc)

Bạn đã từng tham gia những hoạt động thiện nguyện nào?

Câu hỏi 2 (Không bắt buộc)

Tại sao bạn lựa chọn công việc ở Lớp học Cầu Vồng? Ý nghĩa của làm việc tốt với bạn là gì?

Câu hỏi 3 (Bắt buộc)

Bạn có đảm bảo rằng mình sẽ nhiệt tình, sẵn sàng tham gia cá những hoạt động khác của lớp học Cầu Vồng như quyền quý hay là tổ chức sự kiện, ngoài việc đồng hành cùng lớp học mà bạn đã đăng kí? Hãy nêu ra một hoặc nhiều ví dụ thể hiện bạn là người nhiệt tình trong các công việc thiện nguyện.

Câu hỏi 4 (Bắt buộc)

Bạn có nhu cầu học tiếng anh giao tiếp với mức giá rẻ không?

**Hình 5.2:** Các câu hỏi được QTV cài đặt tại website quản lý

The screenshot shows a web page titled "Đăng kí làm Tình nguyện viên". At the top right are buttons for "Trang Chủ" (Home), "Nộp CV" (Submit CV), and "Đăng nhập" (Log in). Below the title, there are three input fields with placeholder text: "Hãy trả lời các câu hỏi sau:", "Bạn đã từng tham gia những hoạt động thiện nguyện nào?", and "Tại sao bạn lựa chọn công việc ở Lớp học Cầu Vồng? Ý nghĩa của làm việc tốt với bạn là gì?". Each field has a corresponding "Nhập vào câu trả lời của bạn" (Enter your answer) button below it. A note at the bottom left of the form area states: "Bạn có đảm bảo rằng mình sẽ nhiệt tình, sẵn sàng tham gia cá những hoạt động khác của lớp học Cầu Vồng như quyên quỹ hay là tổ chức sự kiện, ngoài việc đóng hành cùng lớp học mà bạn đã đăng ký? Hãy nêu ra một hoặc nhiều ví dụ thể hiện bạn là người nhiệt tình trong các công việc thiện nguyện." (You ensure that you will be enthusiastic, willing to participate in other activities of the Cầu Vồng class such as charity, or organizing events, in addition to the work you have registered for? Please list one or more examples that show you are a enthusiastic person in voluntary work.)

**Hình 5.3:** Các câu hỏi được QTV cài đặt hiển thị trên website hỗ trợ đăng ký làm TNV

Với tính năng này, quản trị viên có thể dễ dàng thêm, sửa, xóa các câu hỏi mà người đăng ký cần phải trả lời khi đăng ký làm Tình nguyện viên.

Với **tính năng phân quyền** ở màn hình quản lý CV người quản trị viên có thể xem được tất cả các hồ sơ gửi về hệ thống và theo dõi trực tiếp các trạng thái hồ sơ hay lịch phỏng vấn của tất cả các lớp. Ngoài ra, quản trị viên cũng có thể tìm kiếm, lọc các hồ sơ một cách dễ dàng. Bên cạnh đó cán sự lớp thì chỉ có thể xem được các hồ sơ đăng ký thuộc về lớp mà họ đang quản lý, không cần qua bất kỳ bước trung gian nào mà vẫn có thể chủ động xét duyệt hồ sơ và lên lịch phỏng vấn với người đăng ký.

## 5.2 Hỗ trợ cải tiến nghiệp vụ xét duyệt hồ sơ đăng ký tình nguyện viên

Với quy trình nghiệp vụ thủ công, sau mỗi lần duyệt hồ sơ hay phỏng vấn, người phụ trách phải có nhiệm vụ gửi thông báo về cho người đăng ký ứng tuyển kết quả cuối cùng hoặc phải liên lạc và sắp xếp thời gian phỏng vấn với ứng viên. Tuy nhiên nếu sử dụng hệ thống thì người phụ trách sẽ được giảm thiểu đi các bước như gửi thông báo hoặc hạn chế trao đổi về lịch phỏng vấn với ứng viên. Các bước hệ thống hóa quy trình xét duyệt hồ sơ được mô tả ở hình ??, cụ thể như sau:

Trước hết, tại màn hình đăng ký hồ sơ ứng tuyển, hệ thống yêu cầu người dùng **điền vào khung thời gian rảnh** của mình theo các ngày trong tuần. Kèm theo đó là một trường nhập vào note dành cho những buổi ngoại lệ. Ví dụ: người dùng có khung thời gian rảnh vào lúc 8h – 11h sáng thứ ba, thứ năm, thứ bảy; tuy nhiên riêng thứ bảy ngày (11/8) thì ngoại lệ là phải tham gia đám cưới anh trai nên sẽ không rảnh. Việc thu thập khung thời gian rảnh không chỉ hỗ trợ người phụ trách chủ động sắp xếp lịch phỏng vấn hạn chế việc liên lạc trao đổi không cần thiết, mà

còn giúp cho người duyệt hồ sơ phán đoán với số lượng thời gian rảnh trong một tuần như vậy thì ứng viên có đủ thời gian để duy trì hoạt động dạy học tại Lớp học Cầu Vồng hay không.

Tiếp đến, hệ thống **hỗ trợ tự động gửi email** đến các ứng viên sau mỗi bước thay đổi trạng thái của hồ sơ. Dưới đây là bảng so sánh ưu điểm khi sử dụng hệ thống và khi không sử dụng hệ thống quản lý trong việc gửi mail xét duyệt hồ sơ theo các trường hợp cụ thể.

<b>STT</b>	<b>Hoàn cảnh</b>	<b>Khi không sử dụng hệ thống</b>	<b>Khi sử dụng hệ thống</b>
1	Sau khi ứng viên gửi CV về địa chỉ email của Lớp học Cầu Vồng	Người phụ trách gửi email xác nhận đã nhận được hồ sơ đến ứng viên	Hệ thống tự động gửi email xác nhận đã nhận được hồ sơ đến ứng viên
2	Hồ sơ của ứng viên không đạt	Người phụ trách gửi email thông báo hồ sơ không đạt đến ứng viên	Người phụ trách chuyển trạng thái hồ sơ thành “Trượt”, hệ thống tự động gửi mail thông báo hồ sơ không đạt đến ứng viên
3	Hồ sơ của ứng viên đạt, tiến hành lên lịch phỏng vấn	Người phụ trách liên lạc với ứng viên và trao đổi, sắp xếp thời gian phỏng vấn.	Người phụ trách chuyển trạng thái hồ sơ thành “Chờ phỏng vấn”, dựa vào bảng lịch rảnh của ứng viên lên lịch phỏng vấn. Hệ thống tự động gửi email thông báo thời gian, cách thức phỏng vấn đến ứng viên. Chỉ trong trường hợp ứng viên có nhu cầu thay đổi thời gian phỏng vấn thì sẽ liên lạc và thảo luận lại thời gian phỏng vấn.
4	Ứng viên không đạt sau vòng phỏng vấn	Người phụ trách gửi email thông báo không đỗ phỏng vấn đến ứng viên	Người phụ trách chuyển trạng thái hồ sơ thành “Trượt”, hệ thống tự động gửi email thông báo không đỗ phỏng vấn đến ứng viên

5	Ứng viên đỗ vòng phỏng vấn	Người phụ trách gửi email thông báo đỗ phỏng vấn và các thủ tục sau phỏng vấn đến ứng viên	Người phụ trách chuyển trạng thái hồ sơ thành “Đỗ”, hệ thống tự động gửi email thông báo đỗ và thủ tục sau phỏng vấn đến cho ứng viên.
---	----------------------------	--	--

**Bảng 5.1:** So sánh ưu điểm khi người phụ trách duyệt hồ sơ sử dụng hệ thống

### 5.3 Hỗ trợ thông báo đến người dùng

Thông thường, các cán sự lớp hoặc quản trị viên sẽ phải thông qua các kênh liên lạc, các mạng xã hội như Facebook, Zalo,... để nhắc nhở hoàn thành công việc một cách thủ công. Tuy nhiên hệ thống quản lý hỗ trợ tự động thông báo đến người dùng, giúp cho các bạn tình nguyện viên chủ động hơn trong quá trình hoàn thành công việc. Dưới đây là bảng liệt kê các trường hợp gửi thông báo đến người dùng.

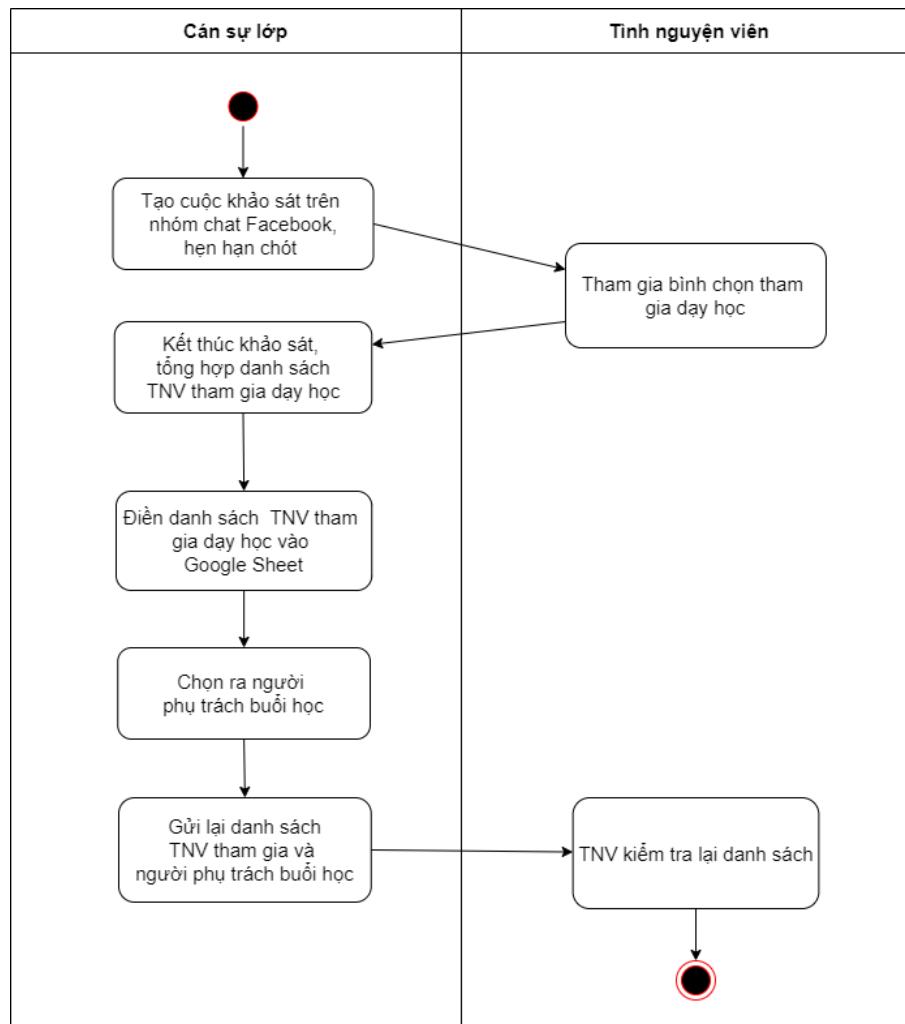
QTV	CSL	TNV	HS	Thời điểm hiển thị	Nội dung thông báo
x	x			Khi có CV mới được tạo + QTV: Nhận thông báo khi có bất kì CV nào được gửi đến + CSL: Nhận thông báo khi có CV gửi đến lớp mình phụ trách	Thông báo có CV mới vừa được gửi đến
	x	x		Ngay sau khi có bài học mới được tạo	Nhắc nhở những người dùng trực thuộc lớp đó đăng ký dạy cho bài học được tạo
	x	x		Ngay sau khi người dùng được phân công phụ trách bài học	Nhắc nhở người dùng được phân công phụ trách cho bài học
	x	x		Ngay sau khi người dùng bị hủy phân công phụ trách cho bài học	Nhắc nhở người dùng không còn được phân công phụ trách cho bài học
	x	x		Ngay sau khi có bài học kết thúc	Nhắc nhở người dùng viết báo cáo sau mỗi buổi học
	x			Ngay sau khi người dùng được phân công phụ trách phỏng vấn	Nhắc nhở người dùng được phân công phụ trách phỏng vấn

	x			Ngay sau khi người dùng bị hủy phân công phụ trách phỏng vấn	Nhắc nhở người dùng không còn được phân công phụ trách cho buổi phỏng vấn
	x	x		Ngay sau khi có bài học mới được tạo	Nhắc nhở người dùng trực thuộc lớp đó đăng ký dạy cho bài học được tạo
	x			Ngay sau khi đăng ký dạy kèm	Nhắc nhở phân công TNV phụ trách dạy kèm
	x	x		Ngay sau khi TNV được phân công phụ trách dạy kèm cho HS	Thông báo đến TNV đã được phân công phụ trách dạy kèm HS
			x	Ngay sau khi kết thúc buổi học	Thông báo bài tập về nhà cho học sinh

**Bảng 5.2:** Các trường hợp gửi thông báo đến người dùng

#### 5.4 Hỗ trợ cải tiến nghiệp vụ đăng ký dạy học

Như đã được trình bày tại mục 2.3.1, các cán sự lớp của lớp theo hình thức dạy tập trung phải thực hiện tạo các bài khảo sát với mục đích lập ra danh sách các tình nguyện viên đăng ký dạy học hàng tuần. Việc này khiến cho quy trình nghiệp vụ quản lý lớp học bị lặp lại không cần thiết và tốn thời gian của cả những thành viên quản trị lớp (Hình 5.4). Chính vì vậy, em đã cải tiến, hệ thống hóa quy trình nghiệp vụ đăng ký dạy học như hình 2.12.



**Hình 5.4:** Sơ đồ quy trình nghiệp vụ tổng hợp danh sách TNV đăng ký dạy học hàng tuần khi không sử dụng hệ thống

Với việc sử dụng hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng, các cán sự lớp chỉ cần tạo bài học mới. Sau đó hệ thống sẽ gửi thông báo đăng ký dạy học đến cho các tình nguyện viên liên quan. Sau khi các tình nguyện viên đăng ký dạy học thì hệ thống tự cập nhật danh sách những tình nguyện viên tham gia dạy học vào buổi đó. Như vậy, các cán sự lớp cũng không cần trải qua bước tổng hợp và điền danh sách vào Google Sheet nữa.

Bên cạnh đó, với danh sách các tình nguyện viên tham gia dạy học, các cán sự lớp cũng có thể thêm, sửa, xóa tình nguyện viên phụ trách bài học một cách chủ động, quản trị viên cũng có thể trực tiếp theo dõi danh sách các tình nguyện viên tham gia giảng dạy thông qua các bài học. Sau khi tình nguyện viên được phân công phụ trách thì cũng có thông báo gửi đến cho tình nguyện viên đó.

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Trong suốt quá trình thực hiện xây dựng hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng, em đã có cơ hội tự xây dựng một ứng dụng quản lý không chỉ dựa trên những kiến thức đã được học tại trường lớp, mà còn thực hành trao đổi, khảo sát trực tiếp với khách hàng, cụ thể là quản lý và các bạn Tình nguyện viên đang tham gia hoạt động giảng dạy tại Lớp học Cầu Vồng.

Trong quá trình phát triển hệ thống, phân tích và định nghĩa yêu cầu người dùng là bước đầu tiên nhưng cũng là một bước quan trọng và không hề đơn giản. Hoạt động này đòi hỏi sự kết hợp chặt chẽ giữa khách hàng và người phân tích để có thể rút ra được những yêu cầu của khách hàng, sắp xếp độ ưu tiên cho các chức năng cần thực hiện. Thoạt nhìn, quá trình này có vẻ đơn giản và dễ làm nhưng thực tế lại khó khăn hơn những gì mà ta tưởng tượng.

Trên thực tế, quá trình phân tích yêu cầu người dùng được diễn ra từ khâu khảo sát hiện trạng dạy-học và cơ cấu nhân sự của tổ chức với mục đích là tìm hiểu yêu cầu của hệ thống. Tiếp đến là phân tích các yêu cầu, xây dựng giả thiết cho các chức năng cần thiết. Sau đó mô hình hóa và thực hiện khảo sát để xác định yêu cầu và độ ưu tiên cho các chức năng. Sau đó là phát triển hệ thống và tiếp nhận phản hồi. Đối tượng của các cuộc khảo sát chính là các Tình nguyện viên đang hoạt động và quản trị viên, những người trực tiếp tham gia các nghiệp vụ của tổ chức. Thông qua các cuộc khảo sát, em đã tiếp thu được những kiến thức như phân tích, xây dựng các chức năng của hệ thống dưới góc nhìn của người dùng và học hỏi được thêm về việc tư vấn cải thiện quy trình nghiệp vụ tới người dùng.

Ngoài những kiến thức về phân tích, yêu cầu người dùng, em đã có thêm cơ hội học hỏi thêm các công nghệ về lập trình, triển khai ứng dụng web, hay những công nghệ mà trước đó chưa sử dụng, cách kết hợp các công nghệ và ứng dụng chúng để xây dựng hệ thống.

Sau quá trình làm đồ án, em đã tạo ra một hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng có đầy đủ các tính năng cơ bản, cần thiết như quản lý nguyện viên, quản lý học sinh, quản lý lớp học, bài học,... Bên cạnh đó là một số tính năng giúp hỗ trợ cải thiện quy trình nghiệp vụ như hỗ trợ đăng ký tình nguyện viên, quản lý hồ sơ đăng ký làm tình nguyện viên, hỗ trợ gửi email/thông báo đến người dùng, thông kê báo cáo sau buổi học. Tuy nhiên, hệ thống vẫn còn nhiều điểm hạn chế như: Giao diện website quản lý còn chưa được đẹp, vẫn còn xảy ra

một vài lỗi nhất định và vẫn còn đang trong quá trình tiếp nhận phản hồi nên chưa được triển khai thực nghiệm.

## 6.2 Hướng phát triển

Trong phạm vi đồ án, em đã hoàn thành cơ bản những chức năng cần có của một hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng nhưng vì thời gian còn hạn chế, vẫn còn những chức năng chưa kịp phát triển. Em xin đề xuất hướng phát triển tiếp theo cho bài toán như sau:

- Phát triển hệ thống quản lý trên web mobile
- Thêm chức năng liên kết với Google Calendar tại trang quản lý với các lịch trình sự kiện của tổ chức.
- Thêm chức năng record video trên trình duyệt giúp người dùng đăng ký làm Tình nguyện viên có thể thu video giới thiệu bản thân bằng tiếng Anh trực tiếp trên trình duyệt
- Thêm chức năng cho người dùng là các nhà hảo tâm, khách có thể xem các hoạt động giảng dạy ở Lớp học Cầu Vồng
- Thêm các chức năng nộp bài tập về nhà đối với người dùng là học sinh
- Thêm chức năng quản lý các bài đăng trên trang website hỗ trợ đăng ký làm Tình nguyện viên
- Cải thiện giao diện người dùng trang quản lý
- Triển khai thực nghiệm hệ thống

Qua quá trình làm đồ án, dưới sự hướng dẫn tận tình của PGS.TS. Cao Tuấn Dũng, em đã hoàn thành hệ thống quản lý dạy học và nhân sự cho Lớp học Cầu Vồng. Đồng thời em học thêm được rất nhiều kiến thức mới, khả năng phân tích và giải quyết vấn đề, kỹ năng lập trình cũng được nâng cao. Em mong rằng hệ thống của em sẽ sớm được quản trị viên Lớp học Cầu Vồng phê duyệt và sớm đi vào hoạt động, hỗ trợ cải tiến và hệ thống hóa quy trình nghiệp vụ, góp phần xây dựng một Lớp học Cầu Vồng lớn mạnh.

## TÀI LIỆU THAM KHẢO

- [1] Hybrid Technology, *Tìm hiểu về mô hình client-server*. [Online]. Available: <https://jobs.hybrid-technologies.vn/blog/tim-hieu-ve-mo-hinh-client-server/> (visited on 05/30/2022).
- [2] Mern stack: Full-stack web application development. [Online]. Available: <https://www.mongodb.com/mern-stack> (visited on 05/30/2022).
- [3] Đoàn Thị Phương Thảo, *Tìm hiểu về mongodb*. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajG1Y3> (visited on 05/30/2022).
- [4] T. Đức Thành, *Tìm hiểu về node js cơ bản*. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-node-js-co-ban-ojaqG0dGEKwZ> (visited on 05/30/2022).
- [5] Node.js v14.20.0 documentation. [Online]. Available: <https://nodejs.org/docs/latest-v14.x/api/> (visited on 05/30/2022).
- [6] Express documents. [Online]. Available: <https://expressjs.com> (visited on 05/30/2022).
- [7] Azure blob storage. [Online]. Available: <https://azure.microsoft.com/en-us/services/storage/blobs/#overview> (visited on 07/30/2022).
- [8] Using oauth 2.0 to access google apis. [Online]. Available: <https://developers.google.com/identity/protocols/oauth2> (visited on 07/30/2022).
- [9] Pattern: Microservice architecture. [Online]. Available: <https://microservices.io/patterns/microservices.html> (visited on 08/06/2022).