BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐÀ LẠT

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC SINH VIÊN NĂM 2022

XÂY DỰNG ỨNG DỤNG HỖ TRỢ PHỤ HUYNH TRA CỨU TÌNH HÌNH HỌC TẬP CỦA SINH VIÊN

Thuộc nhóm ngành khoa học: Công nghệ Thông Tin

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐÀ LẠT

BÁO CÁO TỔNG KẾT

ĐỀ TÀI KHOA HỌC SINH VIÊN NĂM 2022

XÂY DỰNG ỨNG DỤNG HỖ TRỢ PHỤ HUYNH TRA CỨU TÌNH HÌNH HỌC TẬP CỦA SINH VIÊN

Thuộc nhóm ngành khoa học: Công nghệ Thông Tin

Sinh viên thực hiện: Nguyễn Khánh Linh Nam, Nữ: Nam

Dân tộc: Kinh

Lớp, khoa: CTK42, Khoa CNTT Năm thứ: 4 /Số năm Đào tạo: 4.5

Ngành học: Kỹ thuật phần mềm

Giảng viên hướng dẫn: ThS. Phan Thị Thanh Nga

MỤC LỤC

DANH MỤC BẢNG BIỂU 5

DANH MỤC HÌNH ẢNH 5

DANH SÁCH CHỮ VIẾT TẮT 7

MỞ ĐẦU 13

1. Tổng quan tình hình nghiên cứu của đề tài 13

2. Tính cần thiết của đề tài 13

3. Mục tiêu của đề tài 14

4. Phương pháp nghiên cứu 14

5. Phạm vi nghiên cứu 14

6. Kết quả nghiên cứu 15

I. GIỚI THIỆU TỔNG QUAN 16

1.1. Cung cấp dữ liệu bằng API 16

1.2. Một số framework hỗ trợ xây dựng ứng dụng web 17

1.2.1. CodeIgniter 17

1.2.2. Laravel Framework 17

1.2.3. CakePHP 18

1.2.4. Node.js 18

1.3. Framework hỗ trợ phát triển ứng dụng di động 18

1.3.1. Flutter 18

1.3.2. Xamarin 19

1.3.3. React Native 19

CHƯƠNG 2: TRIỂN KHAI ỨNG DỤNG 20

2.1. Phân tích hệ thống 20

2.1.1. Phân tích luồng xử lý của hệ thống 20

2.1.2. Danh sách actor 20

2.1.3. Danh sách Use Case 21

2.1.4. Sơ đồ Use Case 22

2.2. Mô hình triển khai 23

2.2.1. Phân hệ ứng dụng web 23

2.2.1.1. Model 24

2.2.1.2. Controller 24

2.2.1.3. View 25

2.2.2. Phân hệ ứng dụng di động 26

2.2.3. Phân hệ API 26

2.3. Kết quả 28

2.3.1. Phân hệ ứng dụng web 28

2.3.2. Phân hệ ứng dụng di động 32

CHƯƠNG 3: KẾT LUẬN 35
3.1. Kết quả đạt được 35
3.2. Hướng phát triển 35
TÀI LIỆU THAM KHẢO 36
PHỤ LỤC THUẬT NGỮ 37

DANH MỤC HÌNH ẢNH

Ảnh 1: Mô hình hoạt động của Web API 17
Ảnh 2: Các framework hỗ trợ xây dựng ứng dụng web phổ biến 18
Ảnh 3: Các framework hỗ trợ xây dựng ứng dụng di động phổ biến 19
Ảnh 4:Luồng xử lý của hệ thống 20
Ảnh 5: Sơ đồ Use case cho ứng dụng web 22
Ảnh 6: Sơ đồ Use Case cho ứng dụng di động 23
Ảnh 7: Mô hình MVC bao gồm Model, View và Controller 23
Ảnh 8: Hàm tạo lập model 24
Ảnh 9: Cách gọi và sử dụng model 24
Ảnh 10: Ví dụ về Controller AdminController 25
Ảnh 11: Ví dụ về View trong Laravel 25
Ảnh 12: Khả năng chạy đa nền tảng của flutter 26
Ảnh 13: Lấy thông tin gvcn với ứng dụng Postman 27
Ảnh 14: Thông tin sinh viên được API trả về 28
Ảnh 15: Đăng nhập 29
Ảnh 16: Tổng quan website quản lý 29
Ảnh 17: Màn hình thông tin đăng ký học phần 29
Ảnh 18: Màn hình chi tiết điểm danh sinh viên 30
Ảnh 19: Màn hình chi tiết điểm rèn luyện 30
Ảnh 20: Màn hình chi tiết giáo viên 30
Ảnh 21: Chọn lớp học phần 31
Ảnh 22: Màn hình chi tiết lớp học phần 31
Ảnh 23: Màn hình chi tiết phụ huynh 32
Ảnh 24: Màn hình đăng nhập chính 32
Ảnh 25: Màn hình quản lý của phụ huynh 33
Ảnh 26: Màn hình hiển thị lịch học điểm danh của sinh viên 33
Ảnh 27: Màn hình hiển thị điểm thi của sinh viên 34
Ảnh 28: Màn hình hiển thị học phí của sinh viên 34

DANH MỤC BẢNG BIỂU

Bảng 1: Danh sách Actor 23
Bảng 2: Danh sách Use case 24

DANH SÁCH CHỮ VIẾT TẮT

STT	Chữ viết tắt	Tên đầy đủ	Chú giải
1	API	Application Programming Interface	Giao diện lập trình ứng dụng
2	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
3	HTTPS	Hypertext Transfer Protocol Secure	Giao thức truyền tải siêu văn bản bảo mật
4	TCP	Transmission Control Protocol	Giao thức điều khiển truyền dẫn
5	IP	Internet Protocol	Giao thức Internet
6	FTP	File Transfer Protocol	Giao thức truyền tập tin
7	URL	Uniform Resource Locator	Hệ thống định vị tài nguyên thống nhất
8	XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng JSON
9	JSON	JavaScript Object Notation	Kiểu dữ liệu mở trong JavaScript
10	MVC	Model–view–controller	Mô hình thiết kế sử dụng trong kỹ thuật phần mềm
11	SQL	Structured Query Language	Ngôn ngữ truy vấn cơ sở dữ liệu

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐÀ LẠT

THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung:

- Tên đề tài: "Xây dựng ứng dụng hỗ trợ phụ huynh tra cứu tình hình học tập của sinh viên".

- Sinh viên thực hiện:

STT	Họ và tên	MSSV	Lớp	Email liên hệ
1	Trịnh Nguyễn Hồng Phi	1812818	CTK42	1812818@dlu.edu.vn
2	Nguyễn Khánh Linh	1812790	CTK42	1812790@dlu.edu.vn
3	Lê Sỹ Hùng	1812867	CTK42	1812867@dlu.edu.vn

Khóa: 42

Giáo viên hướng dẫn: ThS. Phan Thị Thanh Nga

1. Mục tiêu đề tài

1.1. Về lý thuyết

- Giúp sinh viên củng cố kiến thức về phát triển ứng dụng Web và Mobile.
- Áp dụng kiến thức đã học để xây dựng ứng dụng hỗ trợ phụ huynh tra cứu, theo dõi và quản lý thông tin học tập của sinh viên.

1.2. Về thực nghiệm

- Xây dựng và triển khai ứng dụng quản lý thông tin của sinh viên, giúp phụ huynh dễ dàng quản lý được tình hình học tập của con mình bằng điện thoại.
- Đánh giá kết quả, khó khăn, bất cập còn tồn tại và đề xuất phương hướng phát triển.

2. **Kết quả nghiên cứu**

Xây dựng thành công hệ thống, tiến hành so sánh hiệu quả đạt được của ứng dụng với cách gửi bảng điểm thủ công truyền thống hoặc hệ thống cung cấp điểm bằng tin nhắn.

3. **Khả năng áp dụng của đề tài**

Đề tài có thể áp dụng vào các trường đại học, triển khai cho từng khoa quản lý và áp dụng cho sinh viên.

Ngày tháng năm

Sinh viên chịu trách nhiệm chính thực hiện đề tài
(ký, họ và tên)

Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài *(phần này do người hướng dẫn ghi)*:

.....
...

.....
...

.....
...

.....
...

Xác nhận của trường đại học	Ngày tháng năm
(ký tên và đóng dấu)	**Người hướng dẫn**
	(ký, họ và tên)

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐÀ LẠT

THÔNG TIN SINH VỀ SINH VIÊN

CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI

I. SƠ LƯỢC VỀ SINH VIÊN

Họ và tên: Nguyễn Khánh Linh

Sinh ngày: 08 tháng 10 năm 2000

Nơi sinh: Bến Tre

Lớp: CTK42-PM Khóa: 42

Ảnh 4x6

Khoa: Công nghệ Thông tin

Địa chỉ liên hệ: Lớp CTK42-PM

Khoa Công nghệ Thông tin, Trường Đại học Đà Lạt

Điện thoại: 0776598805 Email: 1812790@dlu.edu.vn

II. QUÁ TRÌNH HỌC TẬP

- **Năm thứ 1:**

Ngành học: Khoa: Công nghệ Thông tin

Kết quả xếp loại học tập: ĐTBTL: 2.17/4.0 – Trung Bình

- **Năm thứ 2:**

Ngành học: Khoa: Công nghệ Thông tin

Kết quả xếp loại học tập: ĐTBTL: 2.18/4.0 – Trung Bình

- **Năm thứ 3:**

Ngành học: Khoa: Công nghệ Thông tin

Kết quả xếp loại học tập: ĐTBTL: 2.51/4.0 - Khá

Sơ lược thành tích: Học bổng KKHT học kỳ I

- **Năm thứ 4 – Học kỳ I:**

Ngành học: Kỹ thuật phần mềm Khoa: Công nghệ Thông tin

Kết quả xếp loại học tập: ĐTBTL: 2.77/4.0 – Khá

Ngày tháng năm

Xác nhận của trường đại học	**Sinh viên chịu trách nhiệm chính thực hiện đề tài**
(ký tên và đóng dấu)	*(ký, họ và tên)*

MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu của đề tài

Ngày nay, sự phát triển của công nghệ di động đã tạo ra một xu hướng mới cho hoạt động truyền tải thông tin. Với khả năng lưu trữ và truy cập các nguồn tài nguyên phong phú theo phương thức không những xa mà còn di động. Nói cách khác việc ứng dụng công nghệ di động vào các công việc quản lý thông tin giúp người dùng dễ dàng tiếp cận các nguồn thông tin thông qua một màn hình vào bất kì lúc nào và không cần phải ở một vị trí cố định.

Với khả năng ứng dụng vô cùng lớn của công nghệ di động, nhiều trường học đã áp dụng công nghệ này vào công tác quản lý học đường cho ra những ứng dụng vô cùng hiệu quả trong việc quản lý sinh viên. Đã có nhiều sinh viên của các trường đại học dựa trên ý tưởng xây dựng các ứng dụng di động dùng để liên kết phụ huynh với nhà trường trong việc quản lý sinh viên. Kể đến là "Ứng dụng công cụ quản trị chất lượng nhằm giảm tình trạng sinh viên bỏ học tại Khoa Đào tạo quốc tế, Đại học Duy Tân[2]", " Giải pháp giảm tỷ lệ sinh viên bỏ học tại trường Cao đẳng nghề công nghiệp Thanh Hóa[3]".

2. Tính cần thiết của đề tài

Tình trạng bỏ học của học sinh, sinh viên diễn ra cả ở những nước phát triển và những nước chậm phát triển, điều đó đã tác động xấu đến sự phát triển xã hội. Những nguyên nhân dẫn đến tình trạng sinh viên bỏ học như: khả năng không theo kịp chương trình ở bậc đại học, sinh viên không có tiền đóng học phí và chi phí sinh hoạt phục vụ học tập hay vì quá khó khăn về tiền bạc phải đi làm thêm, sao nhãng học hành dẫn đến bỏ học…

Để khắc phục tình trạng bỏ học của sinh viên cần có những biện pháp đồng bộ, trên cả diện vi mô và vĩ mô; đó là sự kết hợp trách nhiệm của cả gia đình, nhà trường và xã hội.

Đại học Đà Lạt nói riêng, mặc dù trong mấy năm qua, nhà trường có tìm hiểu và đưa ra cách khắc phục tình trạng bỏ học của sinh viên như: mở cổng thông tin cho phép phụ huynh gửi tin nhắn tra cứu bảng điểm của sinh viên, xây dựng trang web cho phép theo dõi bảng điểm, thời khóa biểu của sinh viên.

Nhưng hai cách trên vẫn còn nhiều mặt hạn chế như cách gửi tin nhắn để tra cứu

bảng điểm thì việc nhớ cú pháp gửi tin nhắn hoặc biết được thời gian nào bảng điểm được cập nhật thì rất khó để phụ huynh nắm bắt kịp thời. Trang web tra cứu thông tin thì ban đầu chỉ cung cung cấp tài khoản và mật khẩu cho sinh viên và việc sinh viên không cho phụ huynh thông tin truy cập vào bảng điểm của chính mình là điều không thể tránh khỏi. Nguyên nhân chính dẫn đến khó khăn này là do không có sự gắn kết giữa nhà trường và gia đình.

Để giải quyết cho vấn đề trên, nhóm đã đưa ra kế hoạch xây dựng ứng dụng quản lý thông tin sinh viên trên các thiết bị di động. Với các tiêu chí như:

- Ứng dụng dễ dàng cài đặt trên các thiết bị di động.
- Phụ huynh có thể dễ dàng theo dõi tình hình học tập hàng tuần của sinh viên bằng cách sử dụng ứng dụng này mà không cần đến các cách thức thủ công như: nhắn tin lên tổng đài

3. Mục tiêu của đề tài

Về lý thuyết:

- Giúp sinh viên củng cố kiến thức về phát triển ứng dụng Web và Mobile.
- Áp dụng kiến thức đã học để xây dựng ứng dụng hỗ trợ phụ huynh tra cứu, theo dõi và quản lý thông tin học tập của sinh viên.

Về thực nghiệm:

- Xây dựng và triển khai ứng dụng quản lý thông tin của sinh viên, giúp phụ huynh dễ dàng quản lý được tình hình học tập của con mình bằng điện thoại.
- Đánh giá kết quả, khó khăn, bất cập còn tồn tại và đề xuất phương hướng phát triển.

4. Phương pháp nghiên cứu

Thực hiện khảo sát thực tế, giải quyết các yêu cầu đặt ra trong việc quản lý tình hình học tập của sinh viên. Từ đó đưa ra các chức năng cần có của ứng dụng.

5. Phạm vi nghiên cứu

Đề tài tập trung xây dựng ứng dụng di động cho phép phụ huynh có thể xem thông tin về học phí, điểm học tập, điểm rèn luyện và tình hình tham gia học tập của sinh viên.

6. Kết quả nghiên cứu

Đề tài này chủ yếu tập trung vào xây dựng ba hệ thống bao gồm:

- **Ứng dụng web:** Cho phép giáo vụ khoa cập nhật thông tin đăng ký học phần, điểm danh sinh viên, điểm học tập và học phí của sinh viên. Giáo viên chủ nhiệm theo dõi tình hình tham gia học tập của sinh viên, của lớp dựa vào kết quả điểm danh.
- **Ứng dụng di động:** Phân hệ dành cho phụ huynh, hỗ trợ phụ huynh tra cứu giám sát quá trình học tập của con em mình.
- **API:** được hiểu là các phương thức, giao thức dùng để kết nối các thư viện và ứng dụng này tới khác thư viện và ứng dụng khác. Nhóm đã xây dựng một loạt các API cung cấp cho ứng dụng di động thông tin về thời khóa biểu, kết quả điểm danh, điểm học tập, điểm rèn luyện để ứng dụng di động có thể truy xuất để lấy thông tin hiển thị.

I. GIỚI THIỆU TỔNG QUAN

Với các yêu cầu của ứng dụng "Hỗ trợ phụ huynh tra cứu tình hình học tập của sinh viên". Thì ứng dụng được chia thành ba phân hệ chính: ứng dụng web dành cho giáo vụ vào giáo viên chủ nhiệm, ứng dụng di động dành cho phụ huynh, API cung cấp thông tin hiển thị cho ứng dụng di động. Sau đây chúng ta cùng tìm hiểu về những khái niệm công nghệ và bộ công cụ hỗ trợ lập trình cho từng phân hệ.

1.1. Cung cấp dữ liệu bằng API

Trong mỗi nền tảng khác nhau như server, web app, mobile app. Chúng có thể viết bằng nhiều ngôn ngữ lập trình khác nhau, cách hoạt động khác nhau. Nên có nhiều vấn đề như không đồng nhất dữ liệu đầu vào và đầu ra khi giao tiếp với nhau.

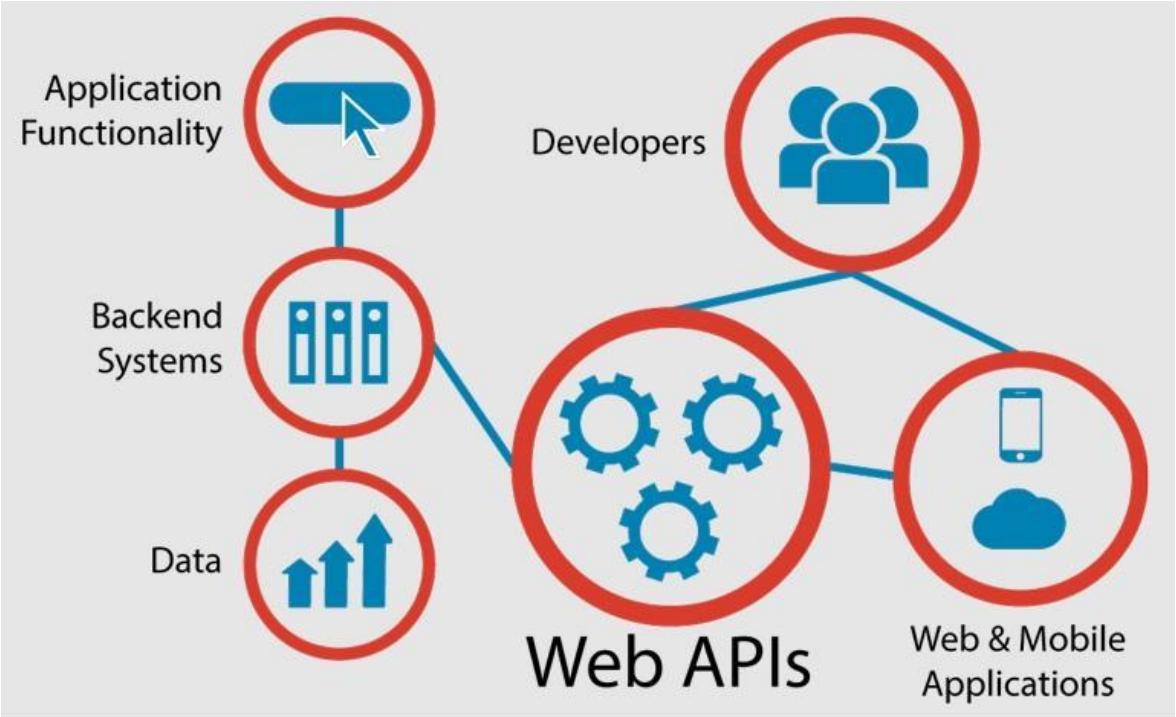
Để giải quyết vấn đề trên có một giải pháp rất hiệu quả dựa trên mô hình client – server đó chính là API.

API được hiểu là một giao diện lập trình phần mềm trung gian hỗ trợ các ứng dụng giao tiếp với nhau. API là từ viết tắt của cụm Application Programming Interface hay còn gọi là giao diện lập trình ứng dụng. Chúng cho phép các ứng dụng khác giao tiếp hiệu quả với nhau và dùng được cho web-based system, operating system, database system, computer hardware, or software library[4].

Cách thức hoạt động của API

Thông thường, một Web API sẽ hoạt động theo quy trình cơ bản sau:

- Tạo URL API để bên thứ 3 có thể gửi yêu cầu đến máy chủ nhờ cung cấp nội dung thông qua giao thức HTTP/HTTPS.
- Tại máy chủ, các ứng dụng nguồn tiến hành kiểm tra xác thực và trả về nội dung thích hợp.
- Máy chủ trả về dữ liệu theo định dạng XML hoặc JSON qua giao thức HTTP/HTTPS.
- Dữ liệu trả về sẽ được phân tích để xây dựng cơ sở dữ liệu tại nơi yêu cầu ban đầu (ứng dụng web, ứng dụng di động).



Ảnh 1: Mô hình hoạt động của Web API

1.2. Một số framework hỗ trợ xây dựng ứng dụng web

Framework là các đoạn code đã được viết sẵn, cấu thành nên một bộ khung và các thư viện lập trình được đóng gói. Chúng cung cấp các tính năng có sẵn như mô hình, API và các yếu tố khác để tối giản cho việc phát triển các ứng dụng web phong phú, năng động.

Sau đây là một số framework được sử dụng phổ biến hiện nay.

1.2.1. CodeIgniter

Đây là một Framework ứng dụng web mã nguồn mở có ngôn ngữ lập trình PHP. CodeIgniter được xây dựng bởi Rick Ellis với phiên bản đầu tiên phát hành ngày 28/02/2006. Hiện nay, Framework này đang được phát triển bởi ExpressionEngine Development Team thuộc EllisLab Inc. Mô hình mà CodeIgniter sử dụng là mô hình MVC.

1.2.2. Laravel Framework

Laravel là một Framework mã nguồn mở, viết bằng ngôn ngữ PHP và hiện đang được cung cấp miễn phí. Framework này được khởi tạo bởi Taylor Otwell vào năm 2011 như một giải pháp để thay thế cho Codeigniter. Tương tự như Codeigniter, Laravel cũng được xây dựng trên mô hình MVC. Các hệ thống mã code có trong Laravel rất linh động, mềm dẻo nhưng vẫn đảm bảo tính chặt chẽ cần thiết. Nhờ đó,

các lập trình viên có thể xây dựng được những khối code đúng chuẩn, sạch sẽ và dễ dàng để nâng cấp trong tương lai.

1.2.3. CakePHP

Đây là một Framework PHP cho web được xây dựng theo mô hình MVC. Mục đích chính của CakePHP là tạo ra một nền tảng phát triển miễn phí, giúp cộng đồng lập trình viên PHP xây dựng được những ứng dụng web nhanh, mạnh mẽ và có tính linh hoạt cao.

1.2.4. Node.js

Node.js là một Framework giúp tạo ra môi trường thích hợp để thực thi các câu lệnh viết bằng ngôn ngữ lập trình web javascript.



Ảnh 2: Các framework hỗ trợ xây dựng ứng dụng web phổ biến

1.3. Framework hỗ trợ phát triển ứng dụng di động

Về phát triển ứng dụng di động, có nhiều framework hỗ trợ xây dựng các ứng dụng di động một cách nhanh chóng và tiện lợi. Dưới đây là ba framework dùng để phát triển ứng dụng di động phổ biến hiện nay.

1.3.1. Flutter

Flutter là một SDK phát triển ứng dụng di động nguồn mở được tạo ra bởi Google. Nó được sử dụng để phát triển ứng ứng dụng cho Android và iOS, cũng là

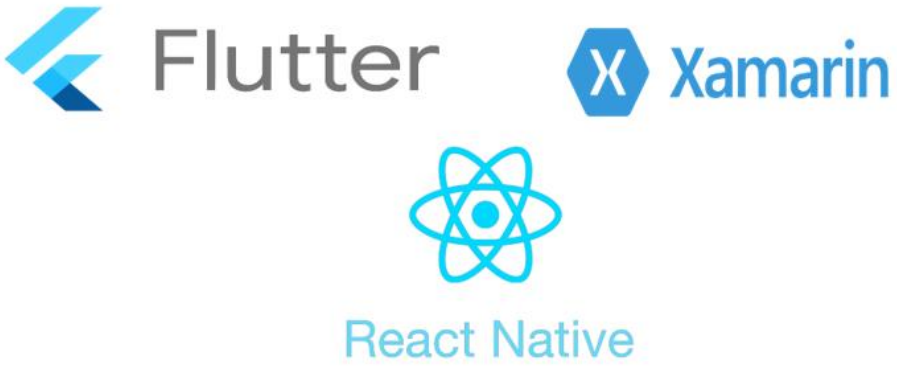
phương thức chính để tạo ứng dụng cho Google.

1.3.2. Xamarin

Xamarin là công nghệ web Framework được cho ra mắt vào tháng 5 năm 2011 bởi các kỹ sư đã tạo ra monoTouch cho Android với mục đích xây dựng và triển khai các ứng dụng trên các thiết bị đa nền tảng. Dựa trên ngôn ngữ lập trình C#, các nhà phát triển có thể sử dụng Xamarin để hỗ trợ coder viết chương trình trên mọi nền tảng di động khác nhau như IOS hoặc Android trên cùng một code project.

1.3.3. React Native

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook. Nó được sử dụng để phát triển ứng dụng di động Android, iOS, Web và UWP bằng cách cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc.

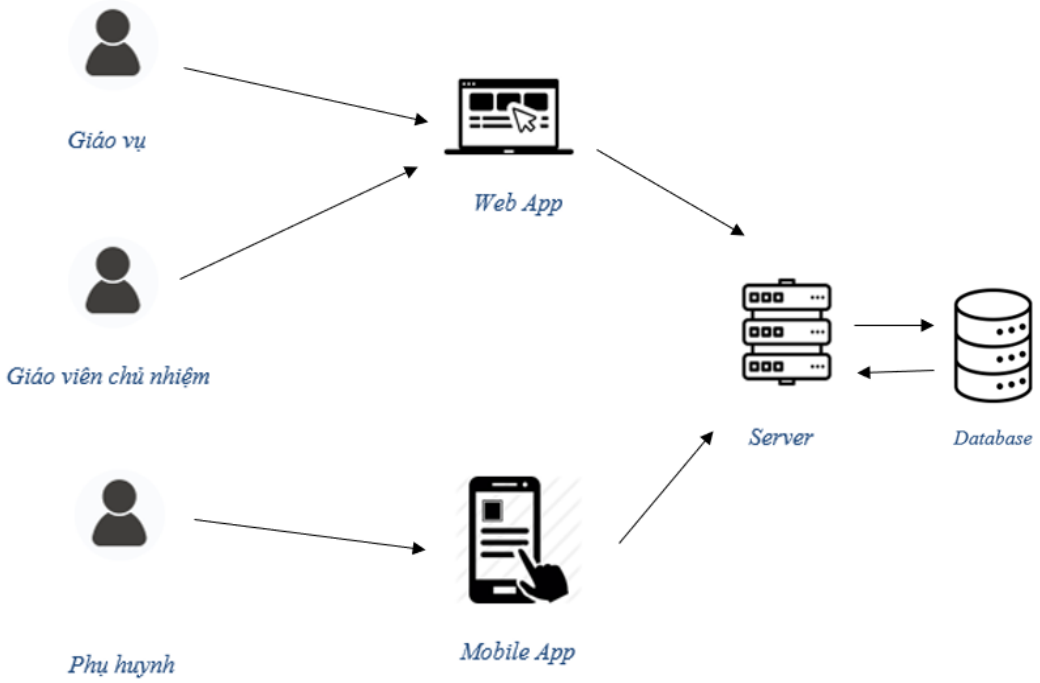


Ảnh 3: Các framework hỗ trợ xây dựng ứng dụng di động phổ biến

CHƯƠNG 2: TRIỂN KHAI ỨNG DỤNG

2.1. Phân tích hệ thống

2.1.1. Phân tích luồng xử lý của hệ thống



Ảnh 4:Luồng xử lý của hệ thống

Hệ thống có hai ứng dụng là web app và mobile app được phân hệ người dùng riêng biệt. Giáo viên truy cập đến web app để xem thông tin sinh viên của lớp mình để kịp thời nắm bắt tình hình, giáo vụ có quyền truy cập chỉnh sửa thông tin dữ liệu thông qua ứng dụng web. Phân hệ phụ huynh truy cập vào ứng dụng di động, xem thông tin về tình hình học tập của sinh viên.

2.1.2. Danh sách actor

ID	Tên Actor	Mô tả
A1	Giáo vụ	Người quản lý truy cập, chỉnh sửa dữ liệu, có các quyền truy cập toàn bộ thông tin trên web app. Có quyền thêm, thay đổi, xóa dữ liệu.

A1	Giáo viên chủ nhiệm	Là người có quyền truy cập, xem thông tin chi tiết về điểm danh, các môn học mà lớp họ chủ nhiệm, theo dõi tình hình học tập của các sinh viên thuộc lớp.
A3	Phụ huynh	Người sử dụng mobile app để xem thông tin về bảng điểm, thời khóa biểu, điểm rèn luyện,… của sinh viên thuộc phụ huynh này quản lý.

Bảng 1: Danh sách Actor

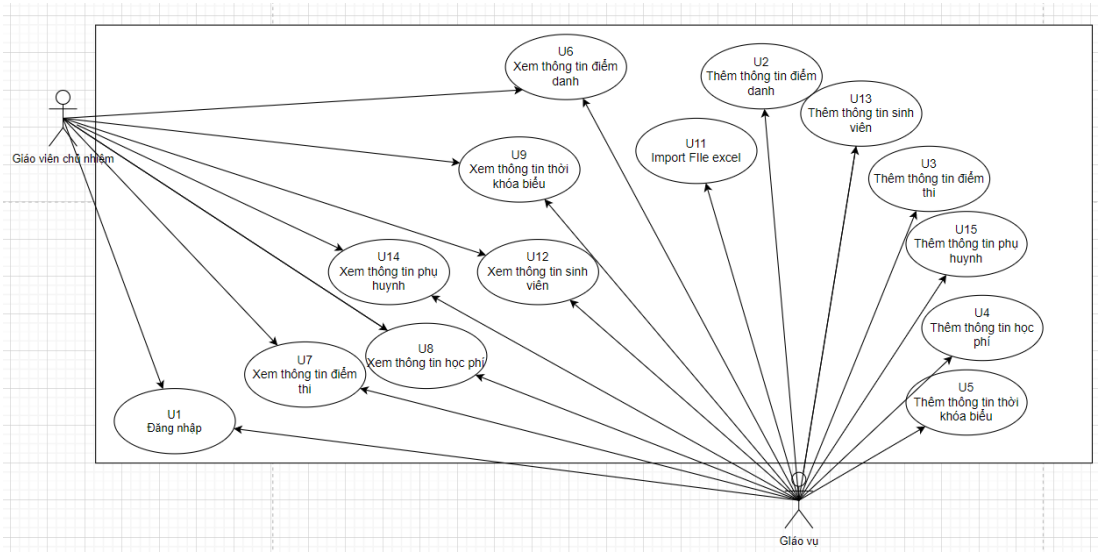
2.1.3. Danh sách Use Case

STT	ID	Tên Use Case	Mô tả
1	U1	Đăng nhập	Đăng nhập vào hệ thống để xác thực người dùng
2	U2	Thêm thông tin điểm danh	Thêm thông tin điểm danh lưu trữ trên hệ thống
3	U3	Thêm thông tin điểm thi	Thêm thông tin điểm thi lưu trữ trên hệ thống
4	U4	Thêm thông tin học phí	Thêm thông tin học phí lưu trữ trên hệ thống
5	U5	Thêm thông tin thời khóa biểu	Thêm thông tin thời khóa biểu lưu trữ trên hệ thống
6	U6	Xem thông tin điểm danh	Xem thông tin điểm danh lưu trữ trên hệ thống
7	U7	Xem thông tin điểm thi	Xem thông tin điểm thi lưu trữ trên hệ thống

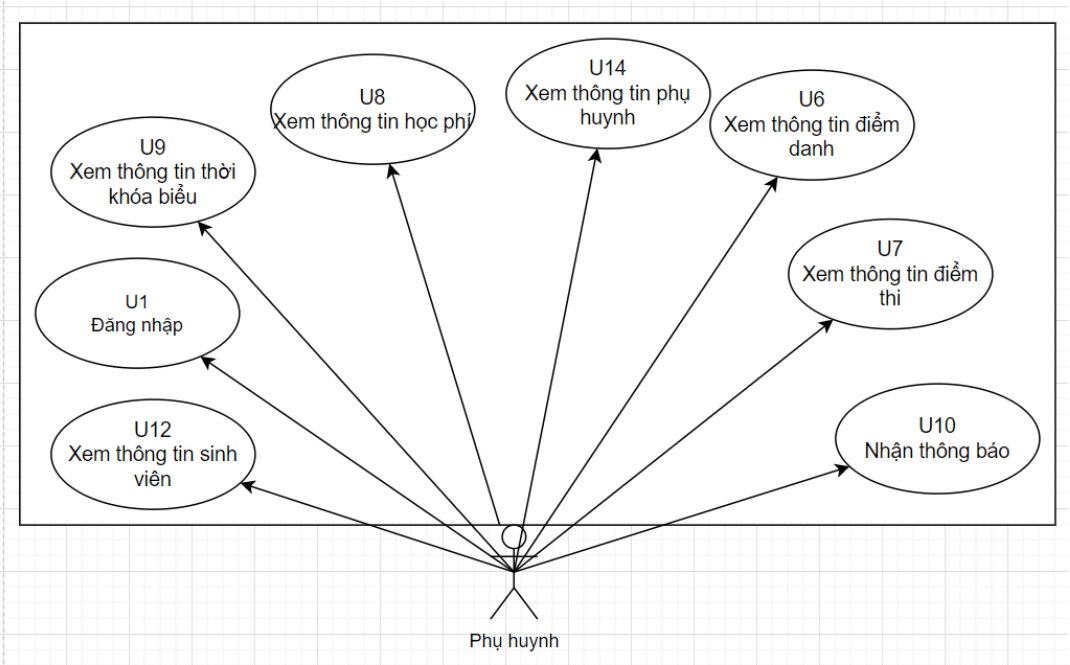
8	U8	Xem thông tin học phí	Xem thông tin học phí lưu trữ trên hệ thống
9	U9	Xem thông tin thời khóa biểu	Xem thông tin thời khóa biểu lưu trữ trên hệ thống
10	U10	Nhận thông báo	Hiển thị thông báo những thay đổi
11	U11	Import File Excel	Thêm thông tin về điểm danh, điểm thi, học phí, thời khóa biểu
12	U12	Xem thông tin sinh viên	Xem thông tin sinh viên được lưu trữ trên hệ thống
13	U13	Thêm thông tin sinh viên	Thêm thông tin sinh viên được lưu trữ trên hệ thống
14	U14	Xem thông tin phụ huynh	Xem thông tin phụ huynh được lưu trữ trên hệ thống
15	U15	Thêm thông tin phụ huynh	Thêm thông tin phụ huynh được lưu trữ trên hệ thống

Bảng 2: Danh sách Use case

2.1.4. Sơ đồ Use Case



Ảnh 5: Sơ đồ Use case cho ứng dụng web

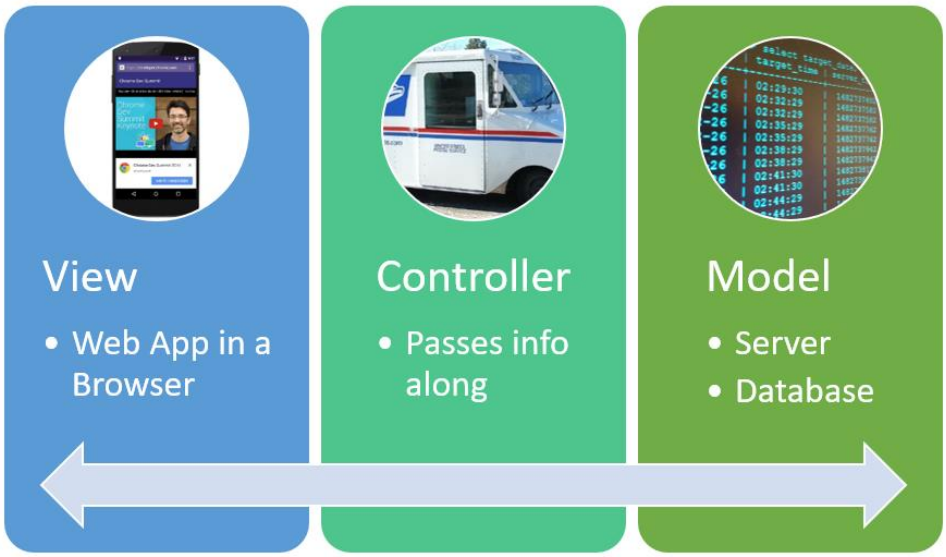


Ảnh 6: Sơ đồ Use Case cho ứng dụng di động

2.2. Mô hình triển khai

2.2.1. Phân hệ ứng dụng web

Trong đề tài này sẽ sử dụng mô hình MVC để xây dựng máy chủ web. Máy chủ web có vai trò quản lý thông tin điểm danh, điểm thi, học phí, thời khóa biểu c ủa sinh viên. Mô hình Model – View – Controller (MVC) là mô hình thiết kế được sử dụng để phân tách giao diện người dùng (view), dữ liệu (model) và logic ứng dụng (controller). Mô hình này giúp đạt được sự tách biệt các thành phần của website[5].



Ảnh 7: Mô hình MVC bao gồm Model, View và Controller

Sử dụng mô hình này cho việc phát triển các website, định tuyến các yêu cầu gửi

đến một Controller, Controller chịu trách nhiệm làm việc với Model để thực hiện các hành động và/hoặc nhận dữ liệu. Controller chọn View để hiển thị dữ liệu và cung cấp View cho Model. View xuất trang HTML dữ liệu dựa trên Model.

Laravel Framework mang đến một cách thức mạnh mẽ, dựa trên mô hình MVC này để xây dựng một website động cho phép sự tách biệt ở các thành phần[6].

2.2.1.1. Model

Model bao gồm một tập các lớp mô hình rõ ràng và dễ dàng gắn kết chúng với cơ sở dữ liệu. Laravel hỗ trợ rất tốt cho việc tương tác với nhiều loại cơ sở dữ liệu bởi nó sử dụng SQL thô, đồng thời sở hữu trình tạo truy vấn thông thạo và Eloquent ORM. Hiện tại, Laravel cung cấp hỗ trợ bốn cơ sở dữ liệu: MySQL, PostgreSQL, SQLite, SQL Server.

Ví dụ về một lớp Model:

```php
public function up()
{
    Schema::create( table: 'lops', function (Blueprint $table) {
        $table->string( column: 'malop')->primary();
        $table->string( column: 'makhoa');
        $table->string( column: 'magvcn');
        $table->string( column: 'siso')->default( value: 0);
        $table->string( column: 'nienkhoa')->nullable();
        $table->timestamps();
    });
}
```

Ảnh 8: Hàm tạo lập model

```
Protected $table = "users";
```

Ảnh 9: Cách gọi và sử dụng model

2.2.1.2. Controller

Controller định tuyến và xử lý các request của người dùng và truy xuất dữ liệu, bằng cách tận dụng các Model.

Ví dụ về một Controller:

```php
namespace App\Http\Controllers;

use App\User;
use Illuminate\Support\Facades\Hash;

use Illuminate\Http\Request;

class AdminController extends Controller
{
    public function loginAdmin()
    {
        if (auth()->check()) {
            return redirect()->to( path: '/admin/home');
        }
        return view( view: 'login');
    }
}
```

Ảnh 10: Ví dụ về Controller AdminController

2.2.1.3. View

View trong laravel cung cấp một cách đơn giản để tạo ra HTML.

```php
<!DOCTYPE html>
<html>
  <head>
    <title>Car {{ $car->id }}</title>
  </head>
  <body>
    <h1>Car {{ $car->id }}</h1>
    <ul>
      <li>Make: {{ $car->make }}</li>
      <li>Model: {{ $car->model }}</li>
      <li>Produced on: {{ $car->produced_on }}</li>
    </ul>
  </body>
</html>
```

Ảnh 11: Ví dụ về View trong Laravel

2.2.2. Phân hệ ứng dụng di động

Về phát triển ứng dụng di động cho phép phụ huynh tra cứu thông tin học tập của sinh viên sẽ sử dụng Flutter. Khung làm việc Flutter cung cấp các phương pháp dễ dàng và đơn giản để bắt đầu xây dựng các ứng dụng dành cho thiết bị di động và máy tính để bàn đẹp mắt với một bộ thiết kế material design và widget phong phú[7].

Đặc điểm của Flutter:

- **Code nguồn mở (Open-Source):** Flutter là một framework code nguồn mở và miễn phí để phát triển các ứng dụng di động.
- **Đa nền tảng (Cross-platform):** Tính năng này cho phép Flutter viết code một lần, duy trì và có thể chạy trên các nền tảng khác nhau.
- **Tải lại nóng (Hot Reload):** Bất cứ khi nào nhà phát triển thực hiện thay đổi trong code, thì những thay đổi này có thể được nhìn thấy ngay lập tức với Tải lại nóng.
- **Code tối thiểu (Minimal code):** Ứng dụng Flutter được phát triển bởi ngôn ngữ lập trình Dart, sử dụng biên dịch JIT và AOT để cải thiện thời gian khởi động tổng thể, hoạt động và tăng tốc hiệu suất.
- **Widget:** framework công tác Flutter cung cấp các widget có khả năng phát triển các thiết kế cụ thể có thể tùy chỉnh. Quan trọng nhất, Flutter có hai bộ widget: Material Design và các widget Cupertino giúp mang lại trải nghiệm không có trục trặc trên tất cả các nền tảng.



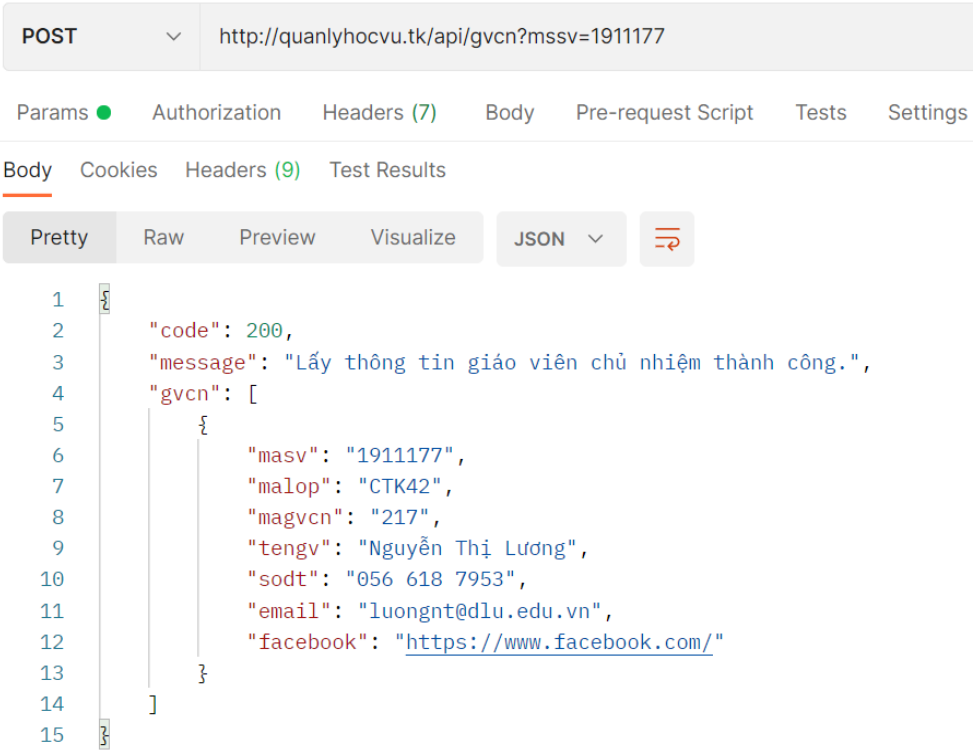
Ảnh 12: Khả năng chạy đa nền tảng của flutter

2.2.3. Phân hệ API

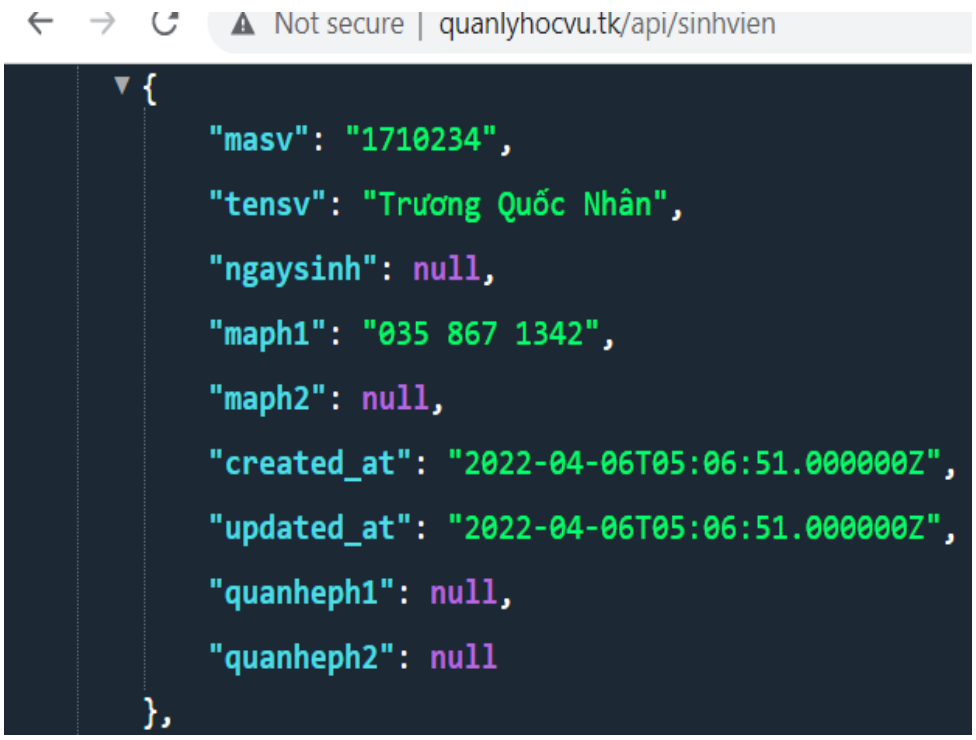
Để cung cấp dữ liệu cần thiết dùng cho ứng dụng di động web server cần cung cấp một số API để ứng dụng di động có thể sử dụng. API cung cấp thông tin về đăng nhập phụ huynh, thông tin bảng điểm, thời khóa biểu, điểm rèn luyện cho sinh viên.

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource[8]. Dưới đây là một số RESTfull API web server cung cấp cho ứng dụng di động để lấy thông tin:



Ảnh 13: Lấy thông tin gvcn với ứng dụng Postman



Ảnh 14: Thông tin sinh viên được API trả về

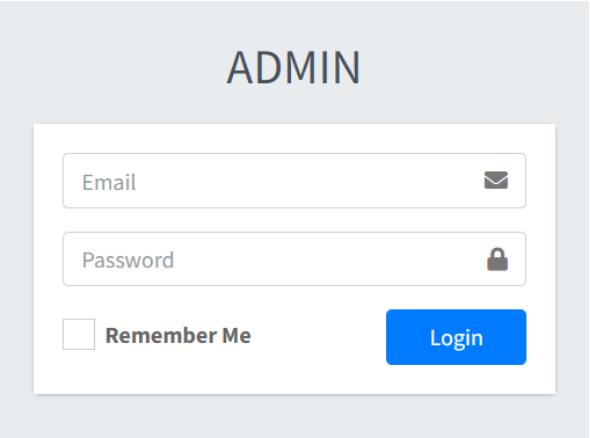
2.3. Kết quả

2.3.1. Phân hệ ứng dụng web

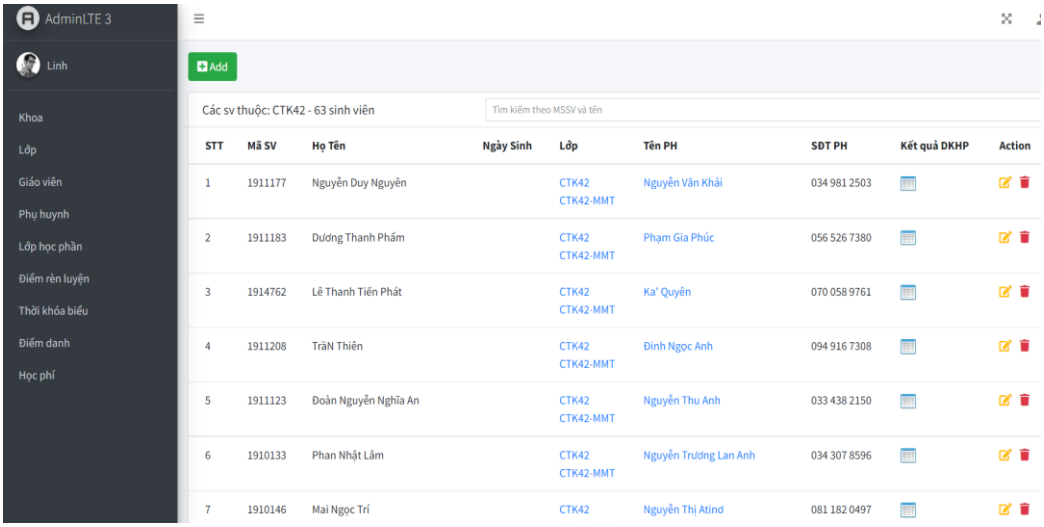
Web quản lý hiện đang được triển khai thử nghiệm tại địa chỉ sau: http://quanlyhocvu.tk/.

Sau khi đăng nhập vào hệ thống giáo vụ có thể thực hiện các công việc như thêm thông tin phụ huynh, giáo viên chủ nhiệm, sinh viên, điểm thi, thời khóa biểu, điểm danh,…

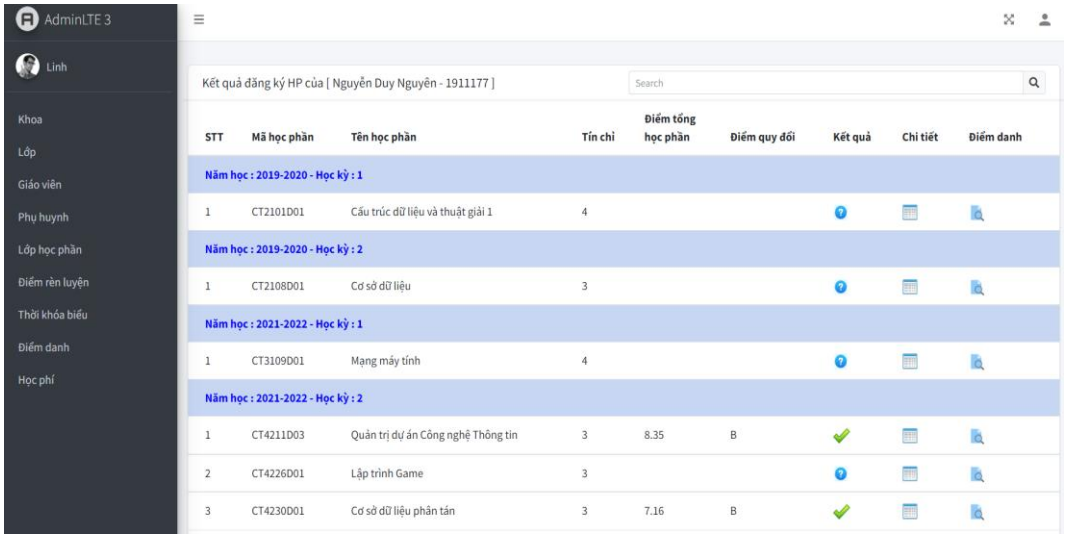
Một số hình ảnh của trang quản trị:



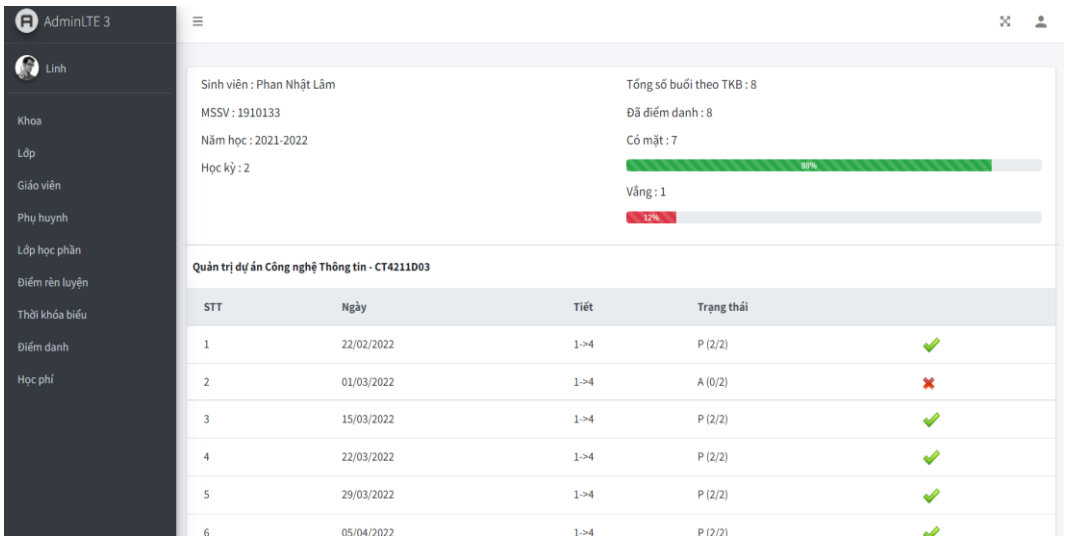
Ảnh 15: Đăng nhập



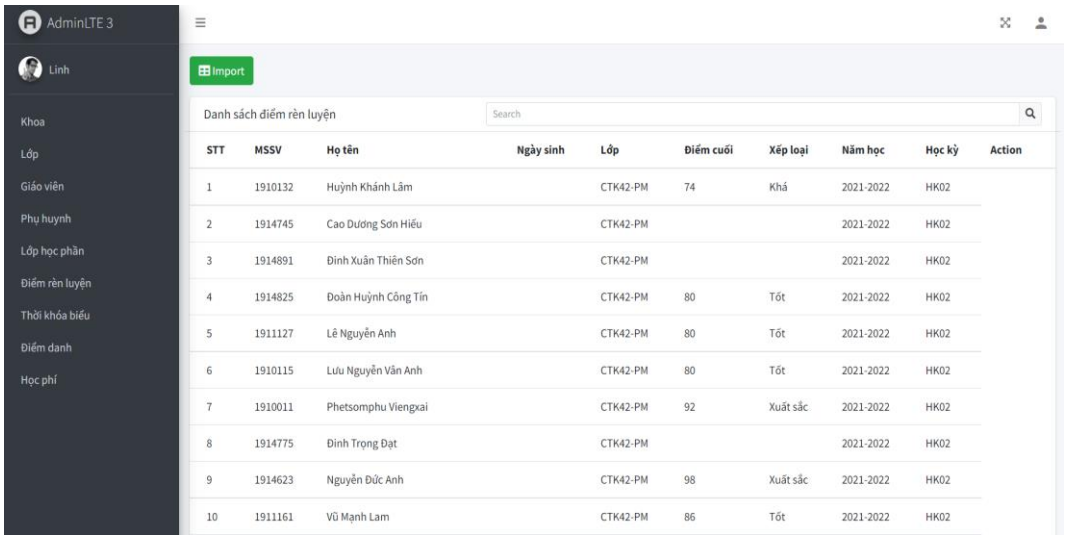
Ảnh 16: Tổng quan website quản lý



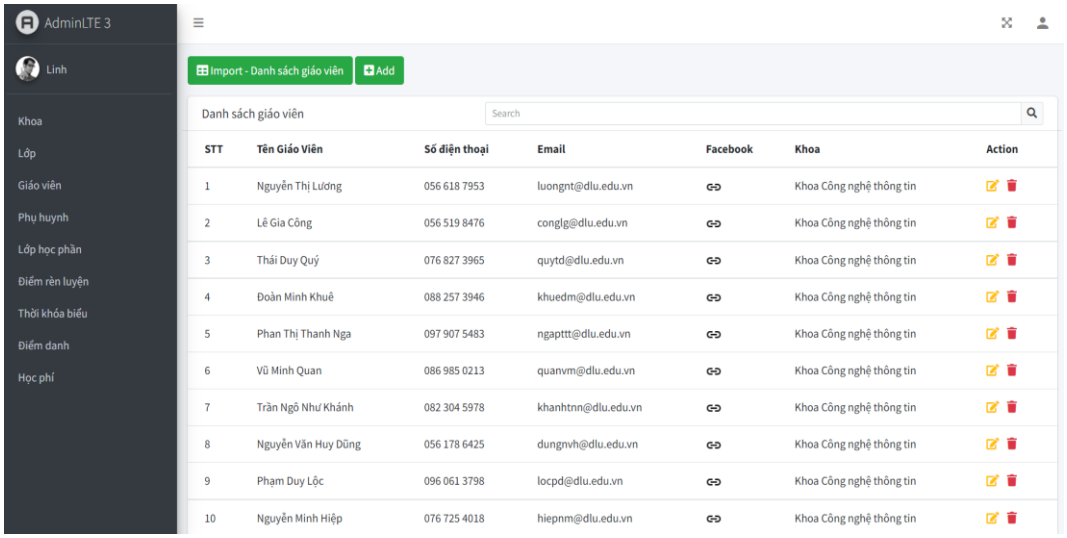
Ảnh 17: Màn hình thông tin đăng ký học phần



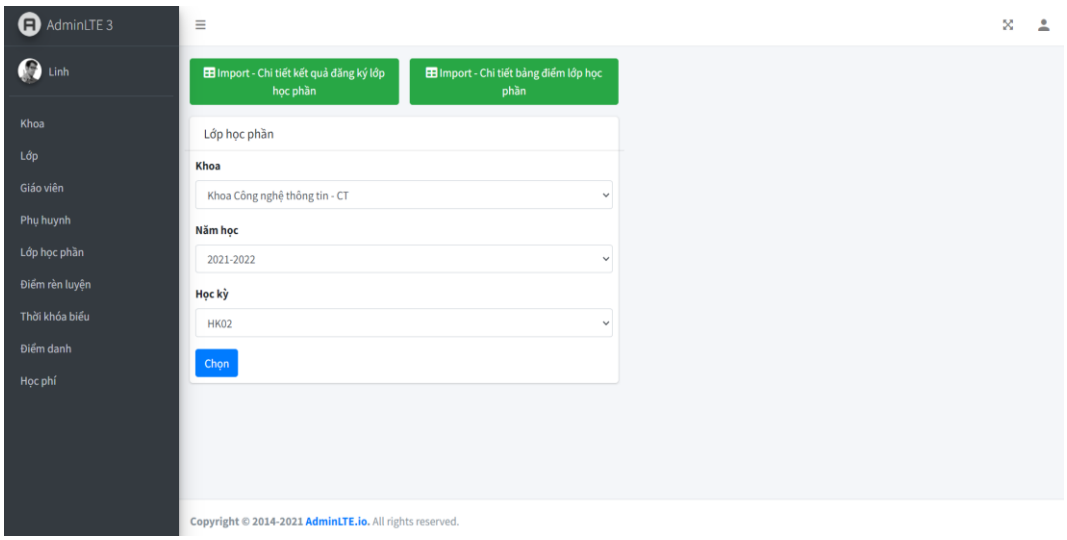
Ảnh 18: Màn hình chi tiết điểm danh sinh viên



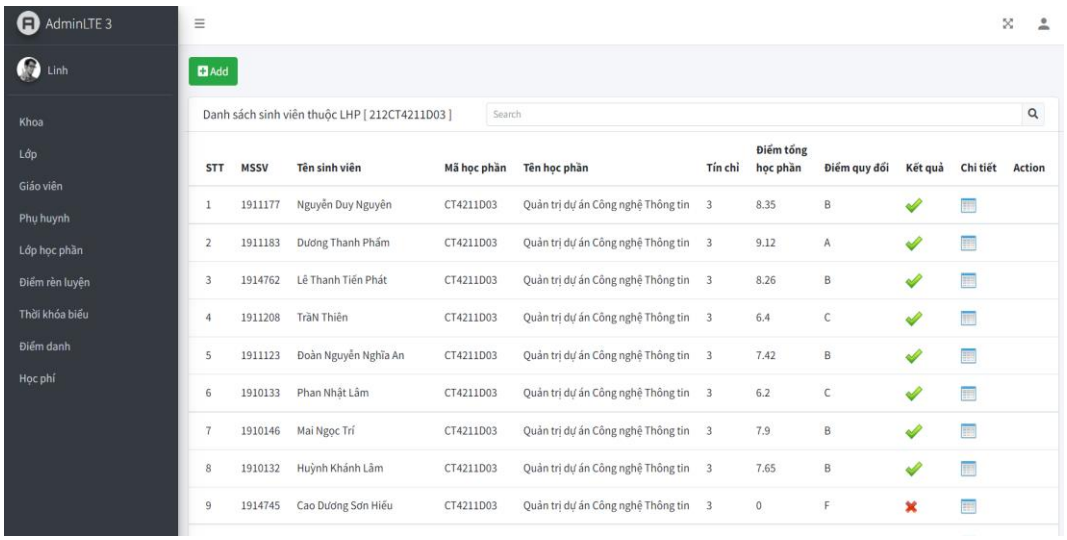
Ảnh 19: Màn hình chi tiết điểm rèn luyện



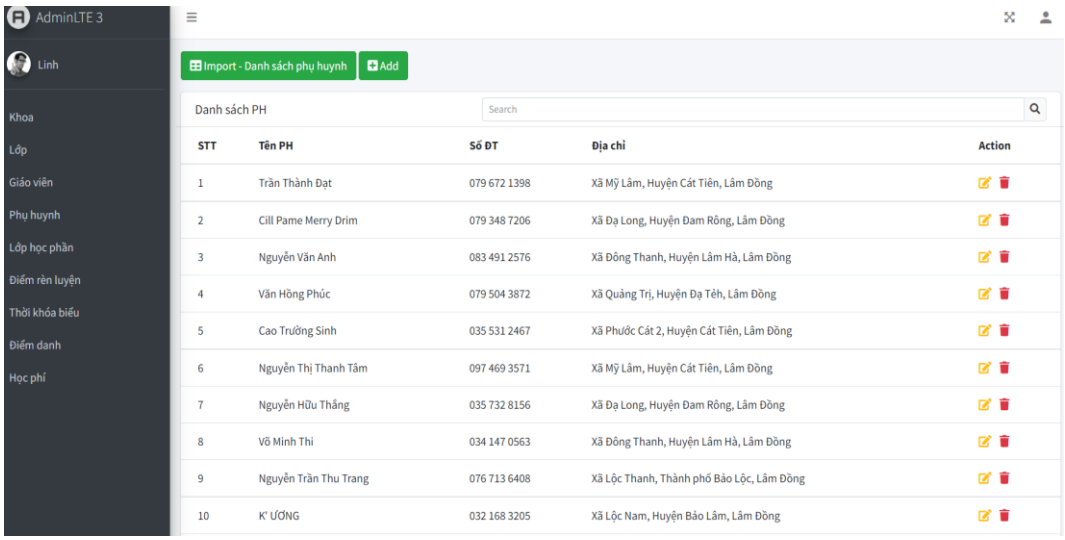
Ảnh 20: Màn hình chi tiết giáo viên



Ảnh 21: Chọn lớp học phần



Ảnh 22: Màn hình chi tiết lớp học phần



Ảnh 23: Màn hình chi tiết phụ huynh

2.3.2. Phân hệ ứng dụng di động

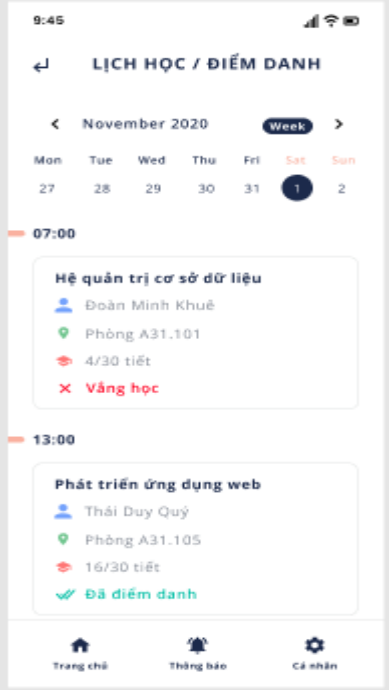
Chức năng chính của ứng dụng này là hỗ trợ phụ huynh tra cứu thông tin học tập của sinh viên với các chức năng như: xem kết quả học tập, xem đánh giá điểm rèn luyện của sinh viên theo học kỳ, xem thời khóa biểu, kết quả điểm danh của các môn học và thông tin liên hệ của giáo viên chủ nhiệm.



Ảnh 24: Màn hình đăng nhập chính



Ảnh 25: Màn hình quản lý của phụ huynh



Ảnh 26: Màn hình hiển thị lịch học điểm danh của sinh viên



Ảnh 27: Màn hình hiển thị điểm thi của sinh viên



Ảnh 28: Màn hình hiển thị học phí của sinh viên

CHƯƠNG 3: KẾT LUẬN

3.1. Kết quả đạt được

Trong quá trình thực hiện xây dựng ứng dụng, nhóm đã đạt được một số kết quả sau:

- **Về lý thuyết:**
  - Nắm rõ cơ bản về phân tích thiết kế hệ thống
  - Thảo luận thiết kế giao diện dễ nhìn, phù hợp người dùng
  - Quy trình phần mềm: lên yêu cầu phần mềm, thiết kế, tiến hành xây dựng ứng dụng, kiểm thử, công bố.
- **Về thực nghiệm:** Đã triển khai thử nghiệm thành công hệ thống gồm các phân hệ khác nhau nhằm làm tăng sự gắn kết của nhà trường với gia đình trong việc quản lý sinh viên. Hệ thống có tính khả thi cao, có thể áp dụng thực nghiệm vào Khoa Công nghệ thông tin Trường Đại học Đà Lạt.

Hệ thống đã được phát triển gồm các phân hệ sau:

- *Phân hệ trang web quản lý* cung cấp các chức năng hỗ trợ giáo vụ và giáo viên chủ nhiệm quản lý thông tin sinh viên.
- *Phân hệ ứng dụng di động* giúp phụ huynh tra cứu thông tin học tập của sinh viên một cách nhanh chóng và được cập nhật kịp thời.

Tuy nhiên, ngoài các kết quả đạt được nhóm còn một số hạn chế sau:

- Do chưa có kinh nghiệm làm ứng dụng thực tế nên một số chức năng còn lý thuyết nhiều, chưa phù hợp với người dùng
- Do ảnh hưởng của dịch COVID-19 quá trình thảo luận trao đổi nhóm diễn ra khó khăn, không kịp thời xử lý những vấn đề phát sinh.
- Tuy ứng dụng đã tương đối hoàn thiện nhưng còn phải bổ sung một số chức năng và dữ liệu để có thể sử dụng trong thực tế.

Tóm lại, việc ứng dụng công nghệ vào công tác giảng dạy

3.2. Hướng phát triển

Từ các kết quả và khó khăn đạt được, đề tài dự kiến sẽ mở rộng phát triển theo hướng:

- Bước đầu áp dụng thực tế trong khoa Công nghệ thông tin Trường Đại học Đà Lạt. Nhằm nâng cao sự gắn kết giữa gia đình và nhà trường trong việc giải quyết vấn đề bỏ học của sinh viên hiện nay tại khoa.

- Kêu gọi nguồn lực để có thể tiếp tục phát triển, duy trì, mở rộng hệ thống.
- Khắc phục, cải thiện các chức năng của hệ thống và thiết kế giao diện phù hợp hơn với người dùng.

TÀI LIỆU THAM KHẢO

[1] B. VOV, "Lượng người dùng smartphone ở Việt Nam đứng trong top 10 toàn cầu," 2021. [Online]. Available: https://vov.vn/cong-nghe/sanh-dieu/luong-nguoi-dung-smartphone-o-viet-nam-dung-trong-top-10-toan-cau-863220.vov.

[2] H. L. Lan, "Application of Quality Management Tools in Student Retention: A Case Study from International School, Duy Tan University," 2021.

[3] T. T. V. Hồng, "Giải pháp giảm tỷ lệ sinh viên bỏ học tại trường Cao đẳng nghề công nghiệp Thanh Hóa," 2016. [Online]. Available: https://dlib.hust.edu.vn/handle/HUST/21119.

[4] M. A. Miller *et al.*, "A RESTful API for access to phylogenetic tools via the CIPRES science gateway," *Evolutionary Bioinformatics*, vol. 11, p. EBO. S21501, 2015.

[5] J. Deacon, "Model-view-controller (mvc) architecture," *Online][Citado em: 10 de março de 2006.] http://www. jdl. co. uk/briefings/MVC. pdf*, 2009.

[6] Z. Subecz, "Web-Development with Laravel Framework," *Gradus*, vol. 8, no. 1, pp. 211-218, 2021.

[7] W. Wu, "React Native vs Flutter, Cross-platforms mobile application frameworks," 2018.

[8] J. Ofoeda, R. Boateng, and J. Effah, "Application programming interface (API) research: A review of the past to inform the future," *International Journal of Enterprise Information Systems (IJEIS)*, vol. 15, no. 3, pp. 76-95, 2019.

PHỤ LỤC THUẬT NGỮ

STT	Từ tiếng Anh	Nghĩa tiếng Việt	Chú giải
1	Client – Server	Mô hình mạng Client – Server	Mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.
2	Request	Yêu cầu	Thông tin từ client gửi đến server, yêu cầu server làm một việc gì đó
3	Response	Lời phản hồi	Kết quả server trả về cho client
4	Back-end System	Hệ thống phụ trợ	Một hệ thống back-end là bất kỳ hệ thống hỗ trợ back-office ứng dụng. Những hệ thống này được sử dụng như một phần của quản trị doanh nghiệp và họ làm việc bằng cách lấy đầu vào người sử dụng và thu thập ý kiến từ các hệ thống khác để cung cấp sản lượng đáp ứng.
5	RESTful API	dạng chuyển đổi cấu trúc dữ liệu	Sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy
6	Widget	Phụ tùng	Trong Flutter Widget là những khối cơ bản dùng để xây dựng và tái sử dụng. Có hai loại Widget là Stateless và Stateful Widget.
7	Dart	Ngôn ngữ Dart	Ngôn ngữ tĩnh, theo hướng đối tượng (OOP), functional programming và lexical scoped, được xác định theo lớp, với cơ chế garbage-collected, sử dụng cú pháp kiểu C để dịch mã tùy ý sang

			JavaScript.