

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống quản lý điểm danh trên ứng dụng di động

LÊ TRUNG ĐÔNG

`dong.lt150959@sis.hust.edu.vn`

Ngành Công nghệ thông tin

Chuyên ngành Kỹ thuật máy tính

Giảng viên hướng dẫn: ThS. Lê Bá Vui

Bộ môn:

Kỹ thuật máy tính

Viện:

Công nghệ thông tin – Truyền thông

HÀ NỘI, 6/2020

Lời cam kết

Họ và tên sinh viên:

Điện thoại liên lạc: Email:

Lớp : Hệ đào tạo :

Tôi – *Lê Trung Đông* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS Lê Bá Vui*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 20 tháng 06 năm 2020

Tác giả ĐATN

Lê Trung Đông

Họ và tên sinh viên

Lời cảm ơn

Lời đầu tiên, em xin chân thành cảm ơn thầy cô, bạn bè ở trường Đại Học Bách Khoa Hà Nội nói chung, các thầy cô Viện Công nghệ thông tin và truyền thông nói riêng và các thầy cô bộ môn Kỹ thuật máy tính. Em cảm ơn thầy cô vì đã nhiệt tình, tâm huyết chỉ dạy chúng em, giúp chúng em tự tin hơn khi bước ra ngoài cánh cổng Đại Học Bách Khoa.

Em xin gửi lời cảm ơn chân thành nhất đến PGS.TS Cao Tuấn Dũng – thầy giáo trực tiếp hướng dẫn, hỗ trợ rất tận tình em trong quá trình thực hiện đồ án tốt nghiệp của mình.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã hỗ trợ em trong quá trình học tập và hoàn thiện đồ án tốt nghiệp.

Xin chân thành cảm ơn !

Tóm tắt

Ngày nay, điện thoại thông minh ngày càng phổ biến và có ảnh hưởng lớn đến cuộc sống của mỗi người. Chỉ cần có chiếc điện thoại thông minh ta có thể làm được rất nhiều việc từ giải trí đến mua sắm quần áo đồ dùng, đặt thức ăn, hay quản lý cửa hàng, cập nhật thông tin... Sinh viên hiện tại hầu như ai cũng có một chiếc điện thoại để giải trí còn cập nhật các thông tin mới nhất nhà trường của lớp thông qua các ứng dụng. Mà em thấy các thầy cô trong trường quản lý lớp của mình thông qua các file excel, và điểm danh cách thủ công nên em xây dựng ứng dụng giúp các giáo viên quản lý lớp và điểm danh sinh viên dễ dàng, nhanh chóng hơn.

Hệ thống gồm web admin giúp người quản lý phân lớp cho giáo viên, tạo lịch học cho sinh viên, ứng dụng mobile dành cho giáo viên để xem lịch dạy theo dõi điểm danh của sinh viên, gửi thông báo cho sinh viên và ứng dụng mobile cho sinh viên để xem lịch học, điểm danh nhận thông báo từ giáo viên.

Em thấy đồ án của em đã đáp ứng được nhu cầu cơ bản về các chức năng quản lý lớp có thể triển khai trong thực tế.

Trong tương lai em có định hướng phát triển thêm tính năng tạo các bài đăng các cuộc bình luận trong các lớp học để sinh viên và giáo viên có thể trao đổi kiến thức và kênh chat giữa giáo viên với sinh viên. Nâng cấp tính năng điểm danh bằng nhận dạng khuôn mặt để điểm danh chính xác hơn.

Qua đồ án, em đã học thêm nhiều kiến thức về thiết kế cơ sở dữ liệu, xây dựng web và ứng dụng cho thiết bị di động.

Mục lục

Nội dung

Lời cam kết	2
Lời cảm ơn	3
Tóm tắt.....	4
Mục lục	5
Danh mục hình vẽ	8
Danh mục bảng	10
Danh mục thuật ngữ	11
CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP.....	12
1.1. Đặt vấn đề	12
1.2 Các vấn đề hệ thống cần giải quyết	12
1.3 Định hướng giải pháp	12
CHƯƠNG 2. CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG.....	14
2.1. Web admin.....	14
2.1.1 Nodejs.....	14
2.1.2 Express	14
2.1.3 Pub.....	14
2.1.4 Sequelize.....	15
2.2 Cơ sở dữ liệu.....	15
2.3 RESTfull API	15
2.4 App Mobile.....	16

2.4.1 React native	16
2.4.2 Redux	16
CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG	17
3.1. Phân tích quy mô, đối tượng sử dụng	17
3.2 Tổng quan chức năng	18
3.2.1 Biểu đồ use case tổng quan	18
3.2.2 Biểu đồ phân giả use case quản lý sinh viên	19
3.2.3 Biểu đồ phân giả use case quản lý sinh viên.....	20
3.2.4 Biểu đồ phân giả use case quản lý môn học	21
3.2.5 Biểu đồ phân giả use case quản lý lớp	21
3.2.6 Biểu đồ phân giả usecase quản lý điểm danh	22
3.2 Đặc tả chức năng.....	23
3.3.1 Đặc tả use case quản lý sinh viên	23
3.3.2 Đặc tả use case quản lý giáo viên.....	25
3.3.3 Đặc tả use case quản lý lớp học.....	27
3.3.4 Đặc tả use case điểm danh lớp học.....	30
3.3.6 Đặc tả use case điểm danh của sinh viên	32
Chương 4 Thiết kế và triển khai ứng dụng.....	34
4.1 Thiết kế kiến trúc.....	34
4.1.1 Lựa chọn kiến trúc phần mềm.....	34
4.1.2 Thiết kế tổng quan	36
4.2 Thiết kế cơ sở dữ liệu	37
4.2.1 Sơ đồ thực thể liên kết	37
4.2.2 Thiết kế chi tiết bảng	38

4.3	Triển khai.....	43
4.3.1	Web admin.....	43
4.3.2	Ứng dụng mobile	44
4.4	Thiết kế giao diện.....	44
4.4.1	Web admin	44
4.4.2	Ứng dụng di động	45
Chương 5	Kết luận và hướng phát triển	48
5.1	Kết luận	48
6.2	Định hướng phát triển.....	48
Tài liệu tham khảo		49

Danh mục hình vẽ

Hình 2.1 Mô hình xử lý luồng truy cập trong Nodejs.....	14
Hình 2.2 Mô hình RESTfull API	15
Hình 3.1 Biểu đồ usecase tổng quan	18
Hình 3.2 Biểu đồ phân rã cho chức năng quản lý sinh viên.....	19
Hình 3.3 Biểu đồ phân rã chức năng quản lý giáo viên	20
Hình 3.4 Biểu đồ phân rã chức năng quản lý môn học	21
Hình 3.5 Biểu đồ phân rã chức năng quản lý lớp học	21
Hình 3.6 Biểu đồ phân rã chức năng quản lý điểm danh	22
Hình 3.7 Biểu đồ hoạt động mô tả quy trình quản lý sinh viên.....	24
Hình 3.8 Biểu đồ hoạt động mô tả quy trình quản lý giáo viên.....	26
Hình 3.9 Biểu đồ hoạt động mô tả quy trình quản lý lớp học	28
Hình 3.10 Biểu đồ hoạt động mô tả quy trình thêm sinh viên vào lớp học	29
Hình 3.11 Biểu đồ hoạt động mô tả quy trình tạo điểm danh của lớp	31
Hình 3.12 Biểu đồ hoạt động mô tả quy trình tạo điểm danh của sinh viên	33
Hình 4.1 Sơ đồ mô hình MVC.....	34
Hình 4.2 Kiến trúc mô hình redux saga.....	35
Hình 4.3 Sơ đồ thực thể liên kết	37
Hình 4.4 Chạy lệnh yarn.....	43
Hình 4.5 Chạy lệnh yarn start	43
Hình 4.6 Giao diện đăng nhập	44
Hình 4.7 Giao diện màn chính	44

Hình 4.8 Giao diện danh sách lớp học	45
Hình 4.9 Màn danh sách lớp học của ứng dụng giáo viên	45
Hình 4.10 Màn danh sách học sinh của lớp trên ứng dụng giáo viên	46
Hình 4.11 Màn danh sách điểm danh của các lớp của ứng dụng giáo viên	46
Hình 4.12 Màn điểm danh của sinh viên	47
Hình 4.13 Màn danh sách điểm danh của ứng dụng sinh viên	47

Danh mục bảng

Bảng 3.1 Đặc tả use case quản lý sinh viên.....	23
Bảng 3.2 Bảng dữ liệu đầu vào của use case quản lý sinh viên	24
Bảng 3.3 Đặc tả use case quản lý giáo viên.....	25
Bảng 3.4 Bảng dữ liệu đầu vào của use case quản lý giáo viên	26
Bảng 3.5 Đặc tả use case quản lý lớp học	27
Bảng 3.6 Bảng dữ liệu đầu vào của use case quản lý lớp học	28
Bảng 3.7 Đặc tả use case điểm danh lớp học.....	30
Bảng 3.8 Bảng dữ liệu đầu vào của use case quản lý lớp học	30
Bảng 3.9 Đặc tả use case điểm danh lớp học.....	32
Bảng 3.10 Bảng dữ liệu đầu vào của use case quản lý lớp học	33
Bảng 4.1 Bảng User trong cơ sở dữ liệu	38
Bảng 4.2 Bảng Student trong cơ sở dữ liệu	38
Bảng 4.3 Bảng Teacher trong cơ sở dữ liệu	39
Bảng 4.4 Bảng Notification trong cơ sở dữ liệu	39
Bảng 4.5 Bảng Class trong cơ sở dữ liệu	40
Bảng 4.6 Bảng Subject trong cơ sở dữ liệu	40
Bảng 4.7 Bảng Schedule trong cơ sở dữ liệu.....	40
Bảng 4.8 Bảng Student_class trong cơ sở dữ liệu	41
Bảng 4.9 Bảng Absent_class trong cơ sở dữ liệu	41
Bảng 4.10 Bảng Absent_student trong cơ sở dữ liệu.....	42

Danh mục thuật ngữ

Thuật ngữ	Giải thích
web service	Dịch vụ web
Non blocking I/O	Là xử lý đa luồng.
API	Viết tắt của Application Programming Interface là giao diện lập trình ứng dụng
HTTP	Viết tắt của HyperText Transfer Protocol là giao thức truyền tải siêu văn bản được sử dụng trong www dùng để truyền tải dữ liệu giữa Web server đến các trình duyệt Web và ngược lại
ORM	Viết tắt của Object Relational Mapping là ánh xạ các dữ liệu trong hệ quản trị cơ sở dữ liệu sang dạng đối tượng
JSON	Viết tắt của JavaScript Object Notation là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định.
URL	Viết tắt của Uniform Resource Locator là đường dẫn hay địa chỉ dùng để tham chiếu đến các tài nguyên trên mạng Internet.

CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

1.1. Đặt vấn đề

Theo như em tìm hiểu được hiện tại ở trường Đại học Bách khoa Hà Nội, ở viện công nghệ thông tin vào đầu mỗi kì các giáo viên sẽ họp và phân công lớp cho các giáo viên. Sau đó viện sẽ sắp xếp lịch học rồi gửi file excel danh sách lớp học và các sinh viên cho các giáo viên. Từ file excel đó giáo viên sẽ tách ra danh sách các lớp và sắp xếp lịch dạy.

Theo như quy chế của nhà trường mỗi môn học gồm có điểm giữa kì và cuối kì từ đó tính điểm cuối cùng của môn học. Điểm giữa kì được tổng hợp từ điểm chuyên cần, điểm bài tập lớn hoặc bài kiểm tra. Để có điểm chuyên cần giáo viên thường điểm danh mỗi giờ học. Có nhiều cách điểm danh như gọi tên từng sinh viên, phát phiếu cho sinh viên điền vào, điểm danh bằng vân tay, hay làm những bài kiểm tra nhỏ. Những lần điểm danh đó thường mất 15 đến 20 phút mỗi giờ và có thể điểm danh hộ. Vào cuối mỗi kì giáo viên cần tổng hợp lại và chấm điểm chuyên cần.

Vì vậy em xây dựng hệ thống quản lý điểm danh này để giúp giáo viên có thể điểm danh và theo dõi điểm danh dễ dàng thuận tiện nhanh chóng.

1.2 Các vấn đề hệ thống cần giải quyết

Vấn đề thứ nhất điểm danh của sinh viên chính xác, để giải quyết vấn đề này em so sánh vị trí thiết bị giáo viên điểm danh và thiết bị của sinh viên. Để lấy vị trí của sinh viên và giáo viên em thông qua công nghệ dịch vụ lấy vị trí của google. Ngoài ra để xác định vị trí của sinh viên có ở trong lớp học không em thông qua danh sách wifi mà thiết bị có thể kết nối của giáo viên và sinh viên để xác định.

Vấn đề thứ hai điểm danh hộ, để giải quyết em sẽ lấy mã thiết bị khi điểm danh lưu lại trên cơ sở dữ liệu và mỗi khi điểm danh em kiểm tra thiết bị đó điểm danh hay chưa, khi đó mỗi một thiết bị chỉ được điểm danh một học sinh.

1.3 Định hướng giải pháp

Đề án hướng đến xây dựng hệ thống quản lý điểm danh, triển khai hệ thống này gồm 02 thành phần website và ứng dụng di động. Với website em sử dụng mô hình MVC (Model-View-Controller) là mô hình phổ biến nhất hiện nay dành cho xây dựng website. Website được dựng trên nền tảng Nodejs với ngôn ngữ JavaScript với framework Express. Em chọn Nodejs vì có ưu điểm truy cập nhanh chóng mạnh mẽ với blocking I/O và bảo mật và hỗ trợ tốt về cả website và API. Với thành phần ứng

dụng di động em sử dụng react native do Facebook phát triển. Vì với react native bằng javascript em có thể sinh ra cả file apk và ipa có thể chạy trên cả hệ điều hành ios và android mà hiệu năng cao và mượt. Phần ứng dụng di động em sử dụng 02 ứng dụng một cho sinh viên và một cho giáo viên với những chức năng của giáo viên và sinh viên.

Phần điểm danh của sinh viên em sẽ sử dụng vị trí hiện tại của thiết bị điện thoại giáo viên và sinh viên để điểm danh vì hiện nay công nghệ định vị lấy vị trí của thiết bị khá là chính xác.

Trong xây dựng phần mềm, vấn đề quan trọng là việc thiết kế giao diện mang lại trải nghiệm tốt nhất cho người dùng cuối. Một phần mềm tốt là phần mềm đáp ứng được yêu cầu của người dùng bao gồm yêu cầu chức năng và cả phi chức năng.

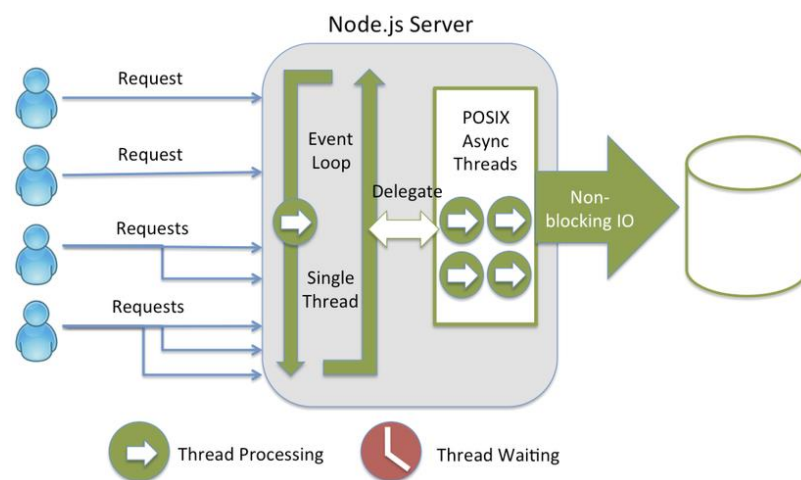
Về mặt kỹ thuật, để xây dựng hệ thống phần mềm cần thiết phải có đó là kiến thức về lập trình. Với các yêu cầu của hệ thống, các kiến thức về xử lý luồng bất đồng bộ cần được áp dụng thành thạo, các kiến thức giao tiếp giữa client và web service hay xử lý các tác vụ chạy ngầm.

CHƯƠNG 2. CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG

2.1. Web admin

2.1.1 Nodejs

Nodejs là nền tảng phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được web server một cách nhanh chóng và dễ dàng mở rộng. Nodejs được chọn là do nodejs có cơ chế xử lý bất đồng bộ nên xử lý non-blocking các tác vụ nhanh và xử lý rất nhiều kết nối cùng một lúc.



Hình 2.1 Mô hình xử lý luồng truy cập trong Nodejs

2.1.2 Express

Express là một framework của nodejs, cung cấp cho nodejs rất nhiều tính năng mạnh mẽ trên nền tảng web. Đó là thiết lập các lớp trung gian trả về các HTTP request, định nghĩa các router với các hành động khác nhau và trả về các trang HTML view.

Ngoài ra kết hợp với các module body-parser để xử lý JSON, mã hóa URL, cookie-parser chuyển đổi header của cookie và phân bố đến các req.cookies.

2.1.3 Pug

Pug is một trong các template engine của nodejs. Pug giúp ta viết HTML nhanh hơn nhờ cú pháp đơn giản, khả năng tái sử dụng lại các đoạn HTML [1].

Pug kết hợp với JavaScript CSS có thể xây dựng giao diện web cách dễ dàng.

2.1.4 Sequelize

Sequelize là một ORM (Object Relational Mapping) dành cho Node.js và io.js. Nó hỗ trợ bạn truy cập một cách dễ dàng đến PostgreSQL, MySQL, MariaDB, SQLite và MSSQL cùng với các tính năng như là relations, transaction, replication [2] ...

2.2 Cơ sở dữ liệu

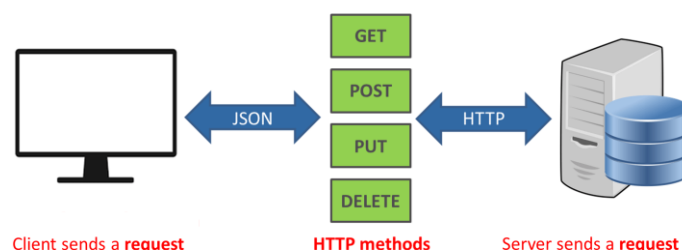
Trong đề án sử dụng hệ quản trị cơ sở dữ liệu postgresql. PostgreSQL là hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system), là hệ thống cơ sở dữ liệu mã nguồn mở mạnh mẽ sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng lưu trữ và mở rộng một cách an toàn các khối lượng công việc dữ liệu phức tạp nhất. PostgreSQL tuân thủ tiêu chuẩn SQL và chạy trên nhiều nền tảng khác nhau.[3]

2.3 RESTfull API

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. RESTfull API gồm hai thành phần API và REST.

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.



Hình 2.2 Mô hình RESTfull API

2.4 App Mobile

2.4.1 React native

React Native là công nghệ được tạo bởi Facebook, cho phép các dev sử dụng JavaScript để làm mobile apps trên cả Android và iOS với cảm nhận và giao diện native. Các ứng dụng react native được tạo viết bằng sự kết hợp giữa javascript và XML-esque (JSX) sau đó được biên dịch ra Objective-C cho IOS và JAVA cho android[4].

Do react native được biên dịch ra ngôn ngữ gốc của IOS và android có hiệu suất cao hơn so với các ứng dụng di động viết trên nền tảng web như hibrid cũng dùng đa nền tảng. Mặt khác react native không chỉ hoạt động trên cả hai hệ điều hành mà còn hoạt động theo cùng một cách trên cả hai nền tảng mà không bị trễ. Do đó rất nhiều công ty khổng lồ như Facebook, Instagram, Skype, ... chuyển hướng thiết kế app bằng react native.

Bên cạnh ưu điểm trên react native hiệu suất không bằng native thuần cũng phải đánh đổi lại khi sinh ra file apk hay ipa có khá nặng.

2.4.2 Redux

Redux là thư viện của javascript giúp quản lý các trạng thái của ứng dụng. Redux được xây dựng dựa trên nền tảng tư tưởng của kiến trúc Flux nên redux với react có thể kết hợp hoàn hảo với nhau[5].

Redux dùng để chứa tất cả mọi thứ thay đổi trong ứng dụng dữ liệu trạng trạng thái giao diện– UI state, được lưu trong đối tượng gọi là state hoặc state tree. Redux sẽ quản lý tất cả dữ liệu các hành động thay đổi dữ liệu như click, menu hay cập nhật dữ liệu từ máy chủ và đồng bộ dữ liệu chung cho màn khác nhau. Các sate được redux chỉ được phép đọc và cách thay đổi duy nhất thông qua các action (một object mô tả những gì xảy ra).

CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

3.1. Phân tích quy mô, đối tượng sử dụng

Hệ thống xây dựng hướng đến đối tượng nhà trường giáo viên quản lý điểm danh, đối tượng sử dụng bao gồm 1 người quản lý admin, giáo viên và sinh viên. Vì là hệ thống nghiệp vụ nội bộ nên sẽ không có đăng ký tài khoản, tài khoản của các sinh viên và giáo viên. Khi tạo các sinh viên và giáo viên admin sẽ tạo luôn tài khoản và mật khẩu cho giáo viên. Với sinh viên tài khoản và mật khẩu mặc định là mã số sinh viên, còn với giáo viên là số điện thoại của giáo viên.

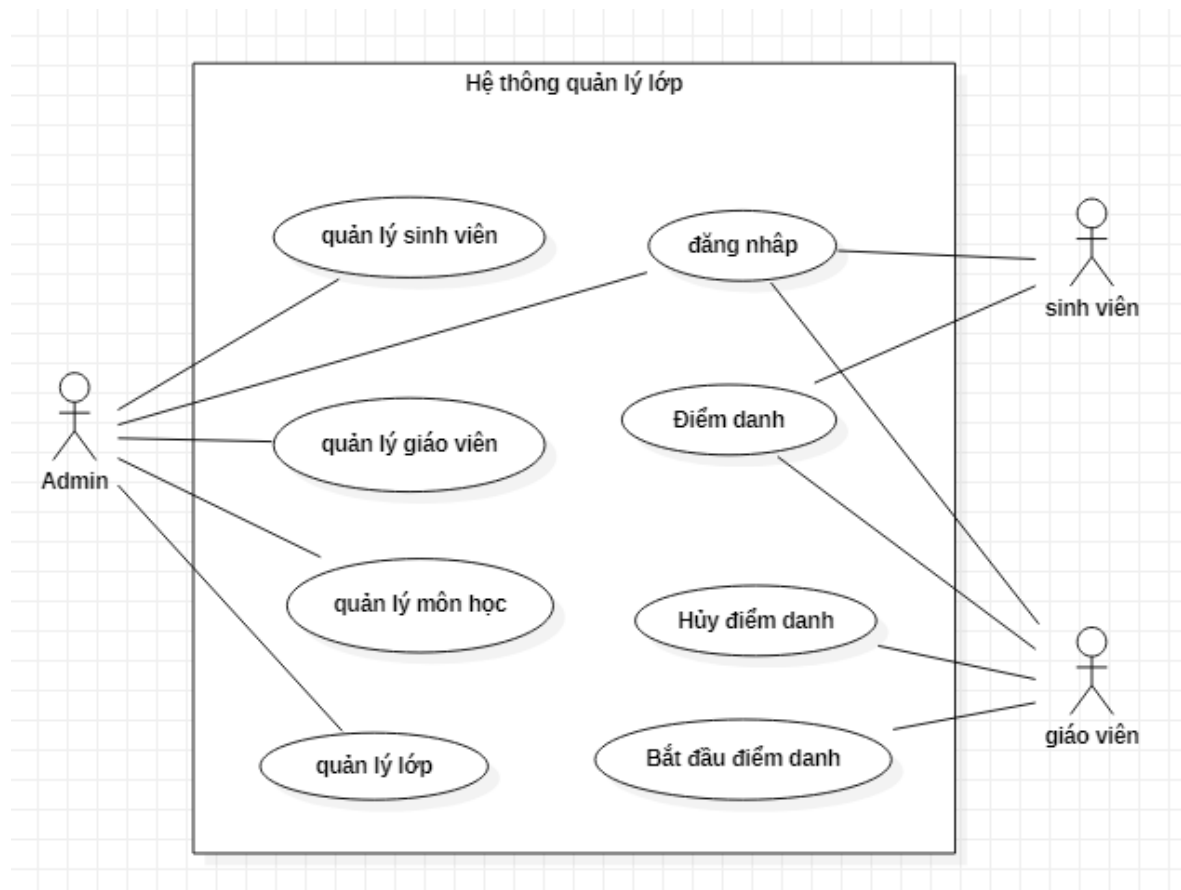
Hệ thống quản lý lớp sẽ cần 03 đối tượng sử dụng hệ thống đó là quản trị viên, giáo viên và sinh viên với các chức năng nghiệp vụ từng đối tượng như sau:

- Quản trị viên:
 - Thêm sửa xóa sinh viên
 - Thêm sửa xóa giáo viên
 - Thêm sửa xóa môn học
 - Tạo lớp chỉnh sửa và thêm sinh viên vào lớp xuất file excel.
- Giáo viên:
 - Xem lịch dạy
 - Kích hoạt, hủy điểm danh cho lớp
 - Nhập điểm cho sinh viên
 - Điểm danh cho sinh viên,
 - Hủy điểm danh của sinh viên
- Sinh viên:
 - Điểm danh
 - Xem lịch học
 - Xem điểm danh của mình

3.2 Tổng quan chức năng

3.2.1 Biểu đồ use case tổng quan

Biểu đồ use case tổng quát các chức năng của hệ thống được biểu diễn ở hình vẽ dưới đây.



Hình 3.1 Biểu đồ usecase tổng quan

Mô tả biểu đồ use case:

Tác nhân: Bao gồm admin, giáo viên, sinh viên

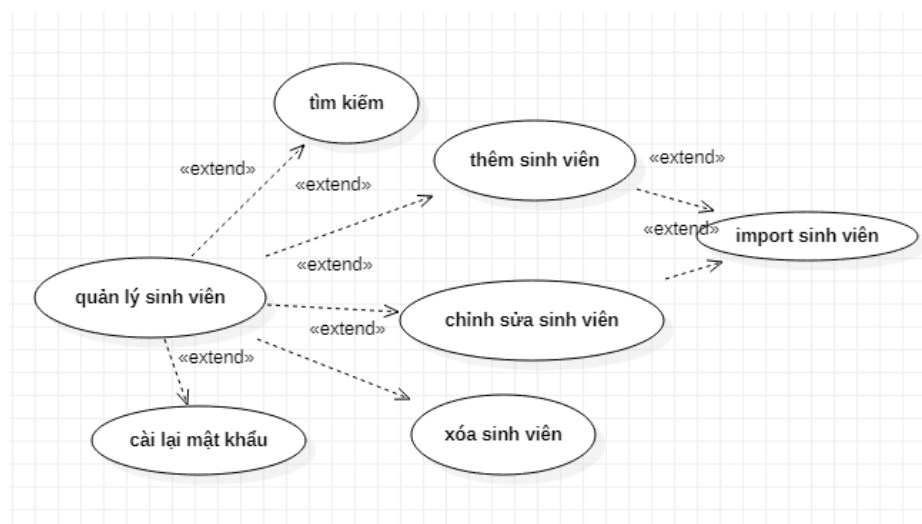
Vai trò:

- *Admin:* Người quản lý lớp.
- *Giáo viên:* Người là quản lý lớp theo dõi sinh viên.
- *Sinh viên:* Người sử dụng ứng dụng.

Mô tả các use case

- *Đăng nhập:* Admin đăng nhập vào web giáo viên đăng nhập ở app giáo viên sinh viên đăng nhập ở app sinh viên.
- *Quản lý sinh viên:* Admin có thể thêm, chỉnh sửa và xóa sinh viên.
- *Quản lý giáo viên:* Admin có thể thêm, chỉnh sửa và xóa giáo viên.
- *Quản lý môn học:* Admin có thể thêm, chỉnh sửa và xóa môn học.
- *Quản lý lớp:* Admin giáo viên thêm lớp chỉnh sửa và thêm sinh viên vào lớp.
- *Bắt đầu điểm danh:* Giáo viên sẽ bắt đầu điểm danh của lớp.
- *Điểm danh:* Giáo viên điểm danh cho sinh viên hoặc sinh viên điểm danh.
- *Hủy điểm danh:* giáo viên có thể hủy điểm danh của sinh viên.

3.2.2 Biểu đồ phần giả use case quản lý sinh viên



Hình 3.2 Biểu đồ phân rã cho chức năng quản lý sinh viên

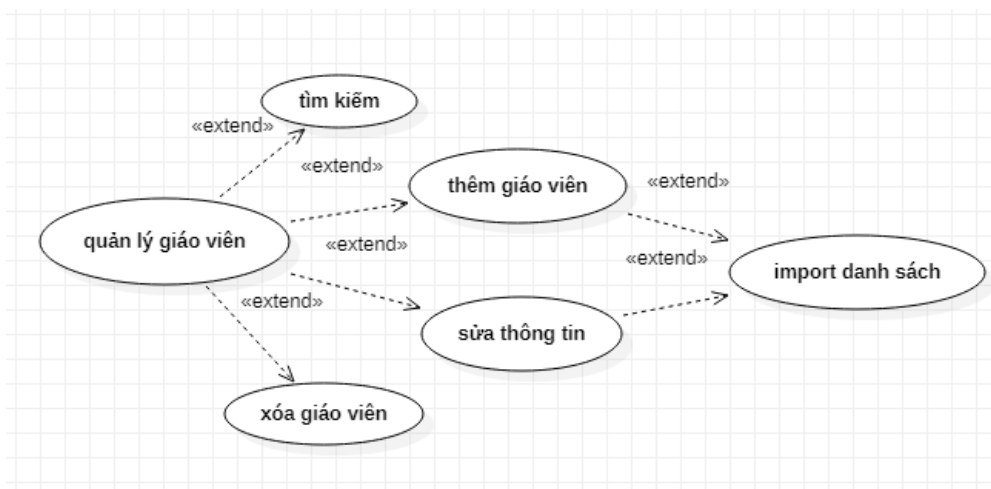
Mô tả biểu đồ use case phân rã

- **Tác nhân:** Admin

Mô tả các use case trong biểu đồ

- *Thêm sinh viên:* admin thêm từng sinh viên hoặc thêm danh sách sinh viên qua import danh sách sinh viên .
- *Chỉnh sửa thông tin sinh viên:* Admin có thể chỉnh sửa thông tin sinh viên bao gồm chỉnh sửa ảnh đại diện, chỉnh sửa tên, số điện thoại...
- *Xóa sinh viên:* Admin có thể xóa sinh viên.
- *Tìm kiếm sinh viên:* Admin có thể tìm kiếm sinh viên theo tên số điện thoại hoặc mã số sinh viên.
- *Đặt lại mật khẩu:* Admin có thể reset lại mật khẩu của sinh viên là mã số sinh viên.
- *Import danh sách sinh viên:* Admin sẽ thêm file excel gồm danh sách sinh viên, nếu sinh viên đã có sẽ sửa thông tin sinh viên nếu chưa thì thêm vào.

3.2.3 Biểu đồ phân giải use case quản lý sinh viên



Hình 3.3 Biểu đồ phân rã chức năng quản lý giáo viên

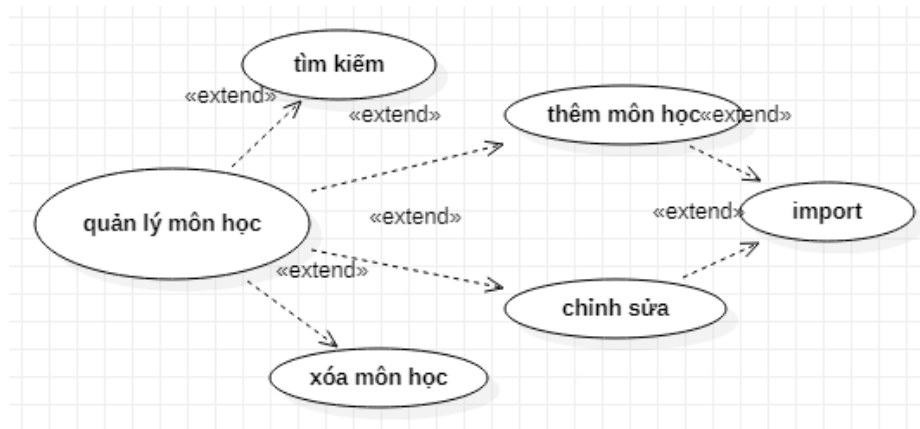
Mô tả biểu đồ use case phân rã

- **Tác nhân:** Admin

Mô tả các use case trong biểu đồ

- *Thêm giáo viên :* admin thêm từng giáo viên hoặc thêm danh sách giáo viên qua import danh sách giáo viên.
- *Chỉnh sửa thông tin giáo viên:* Admin có thể chỉnh sửa thông tin giáo viên bao gồm chỉnh sửa ảnh đại diện, chỉnh sửa tên, số điện thoại , trạng thái.
- *Xóa giáo viên:* Admin có thể xóa giáo viên.
- *Tìm kiếm giáo viên:* Admin có thể tìm kiếm sinh viên theo tên, số điện thoại giáo viên
- *Đặt lại mật khẩu:* Admin có thể reset lại mật khẩu của giáo viên về mặc định là số điện thoại.
- *Import danh sách giáo viên:* Admin sẽ thêm file excel gồm danh sách giáo viên ,nếu giáo viên đã có sẽ sửa thông tin giáo viên nếu chưa thì thêm vào.

3.2.4 Biểu đồ phân rã use case quản lý môn học



Hình 3.4 Biểu đồ phân rã chức năng quản lý môn học

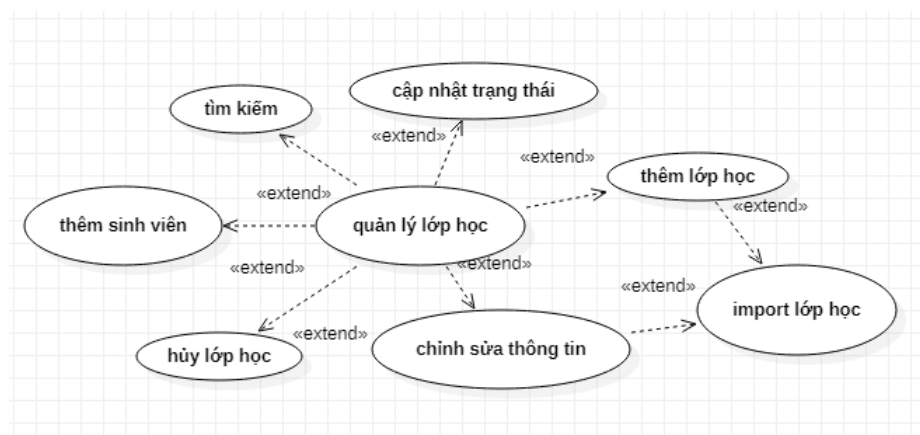
Mô tả biểu đồ use case phân rã

- **Tác nhân:** Admin

Mô tả các use case trong biểu đồ

- *Thêm môn học* : admin thêm từng môn học hoặc thêm danh sách giáo môn qua import danh sách môn học.
- *Chỉnh sửa thông tin môn học*: Admin có thể chỉnh sửa môn học bao gồm chỉnh sửa tên , số tín chỉ...
- *Xóa môn học*: Admin có thể xóa môn học.
- *Tìm kiếm môn học*: Admin có thể tìm kiếm môn học theo tên, mã học phần.
- *Import danh sách môn học*: Admin sẽ thêm file excel gồm danh sách giáo môn học ,nếu môn đã có sẽ sửa thông tin môn học nếu chưa thì thêm vào.

3.2.5 Biểu đồ phân rã use case quản lý lớp học



Hình 3.5 Biểu đồ phân rã chức năng quản lý lớp học

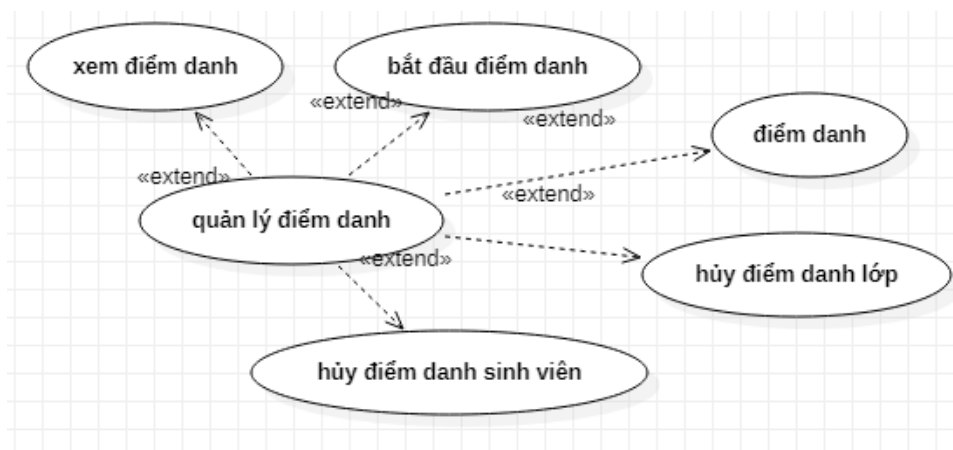
Mô tả biểu đồ use case phân rã

- **Tác nhân:** Admin

Mô tả các use case trong biểu đồ

- *Thêm lớp học* : admin có thể thêm lớp học
- *Chỉnh sửa thông tin lớp học*: Admin có thể chỉnh sửa môn học bao gồm chỉnh môn học, mã lớp học , giáo viên , lịch học
- *Hủy lớp học*: Admin có thể hủy lớp học.
- *Cập nhật trạng thái* :là cập nhật tất cả các lớp đang ở trạng thái đang học sang đã hoàn thành.
- *Tìm kiếm lớp học*: Admin có thể tìm kiếm môn học theo mã lớp, mã học phần.
- *Import danh sách môn học*: Admin sẽ thêm file excel gồm danh sách sinh viên của lớp , giáo viên, môn học lịch học của lớp . Nếu lớp đã có thì thêm sinh viên còn chưa có thì sẽ tạo mới với thông tin trên.
- *Thêm sinh viên vào lớp* :Admin có thể thêm sinh viên vào lớp theo mã số sinh viên.

3.2.6 Biểu đồ phân rã usecase quản lý điểm danh



Hình 3.6 Biểu đồ phân rã chức năng quản lý điểm danh

Mô tả biểu đồ use case phân rã

- **Tác nhân:** giáo viên, sinh viên

Mô tả các use case trong biểu đồ

- *Bắt đầu điểm danh* : Bắt đầu mỗi giờ học giáo viên sẽ vào phần lớp học ở ứng dụng để bắt đầu điểm danh .
- *Hủy điểm danh*: Giáo viên có thể hủy điểm danh của lớp đi và điểm danh đó sẽ không được tính điểm danh của lớp , hoặc giáo viên có thể hủy điểm danh của sinh viên
- *Điểm danh*: khi giáo viên bắt đầu điểm danh của lớp sinh viên sẽ vào lớp điểm danh hoặc giáo viên có thể điểm danh cho sinh viên
- *Xem điểm danh* : Giáo viên có thể vào xem điểm danh của lớp vào thời gian nào hoặc xem điểm danh của từng sinh viên trong lớp . Sinh viên cũng có quyền xem danh sách điểm danh của mình .
- *Hủy điểm danh của sinh viên*: Giáo viên có thể hủy điểm danh của sinh viên đã điểm danh thành công

3.2 Đặc tả chức năng

3.3.1 Đặc tả use case quản lý sinh viên

a, Luồng sự kiện phát sinh

Bảng 3.1 Đặc tả use case quản lý sinh viên

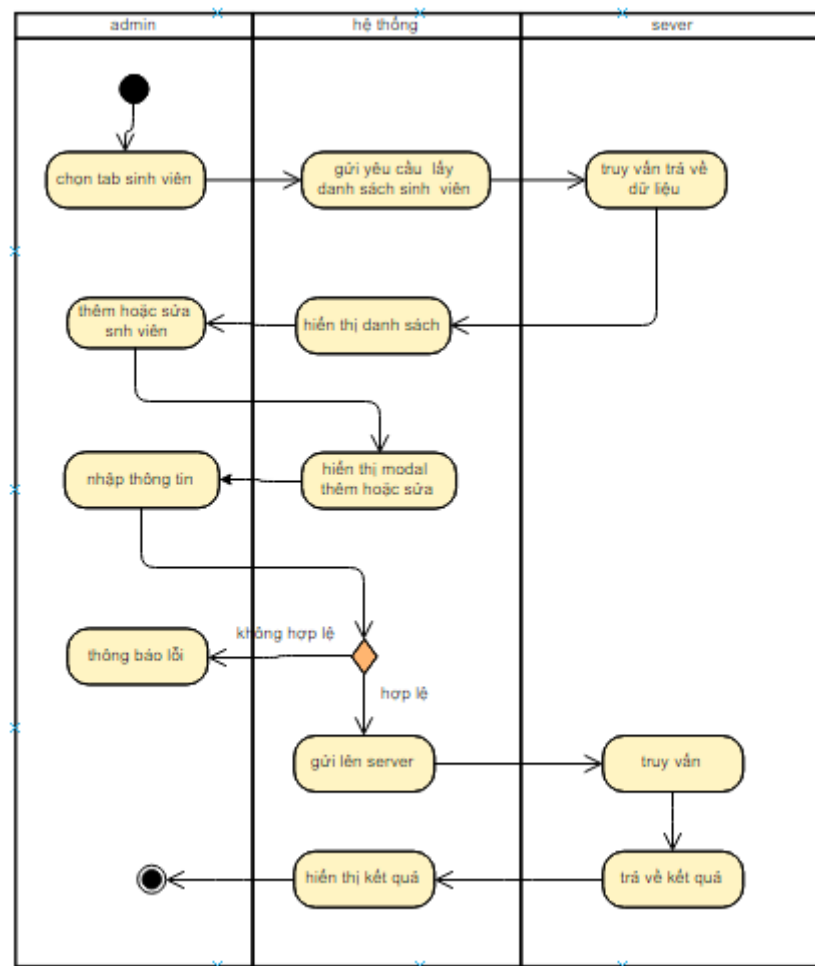
Mã UseCase	UCQLNV	Tên UseCase	Quản lý thông tin sinh viên
Tác nhân	Admin		
Điều kiện trước	Đăng nhập		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Admin	Vào tab Sinh viên	
2	Hệ thống	Hiển thị giao diện danh sách sinh viên	
3	Admin	Chọn chức năng thêm/sửa sinh viên	
4	Hệ thống	Hiển thị modal thêm/sửa sinh viên	
5	Admin	Nhập hoặc chỉnh sửa thông tin sinh viên	
6	Hệ thống	Kiểm tra nhập đầy đủ các thông tin sinh viên	
7	Hệ thống	Kiểm tra mail và số điện thoại đúng định dạng	
8	Hệ thống	Gửi thông tin đã nhập lên server	
9	Server	Kiểm tra mã số sinh viên	
10	Server	Nhận thông tin rồi thêm mới hoặc sửa thông tin của sinh viên trong bảng.	
11	Server	Trả về kết quả cho hệ thống	
12	Hệ thống	Hiển thị kết quả cho admin	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
6b	Hệ thống	Hiển thị thông báo lỗi khi không nhập đầy đủ thông tin của sinh viên.	
7b	Hệ thống	Hiển thị thông báo lỗi khi không nhập đúng định dạng mail hay số điện thoại	
9b	Hệ thống	Hiển thị thông báo lỗi khi mã số sinh viên đã tồn tại.	
Điều kiện sau	Không		

b, Dữ liệu đầu vào

Bảng 3.2 Bảng dữ liệu đầu vào của use case quản lý sinh viên

STT	Trường dữ liệu	Bắt buộc	Điều kiện hợp lệ/ Mô tả	Ví dụ
1	Mã số sinh viên	x	Chuỗi và số	20150959
2	Tên sinh viên	x	Dạng chuỗi	Lê Trung Đông
3	Số điện thoại	x	Dạng số	0329563942
4	Địa chỉ	x	Dạng chuỗi	Bắc Giang
5	Ngày sinh	x	Kiểu ngày	04/03/1997
6	Giới tính	x	Giá trị nguyên	1: Nam, 0: Nữ
7	Email	x	Dạng chuỗi	abc@gmail.com
8	Ảnh		Dạng file	

c, Mô tả bằng biểu đồ hoạt động



Hình 3.7 Biểu đồ hoạt động mô tả quy trình quản lý sinh viên

3.3.2 Đặc tả use case quản lý giáo viên

a, Luồng sự kiện phát sinh

Bảng 3.3 Đặc tả use case quản lý giáo viên

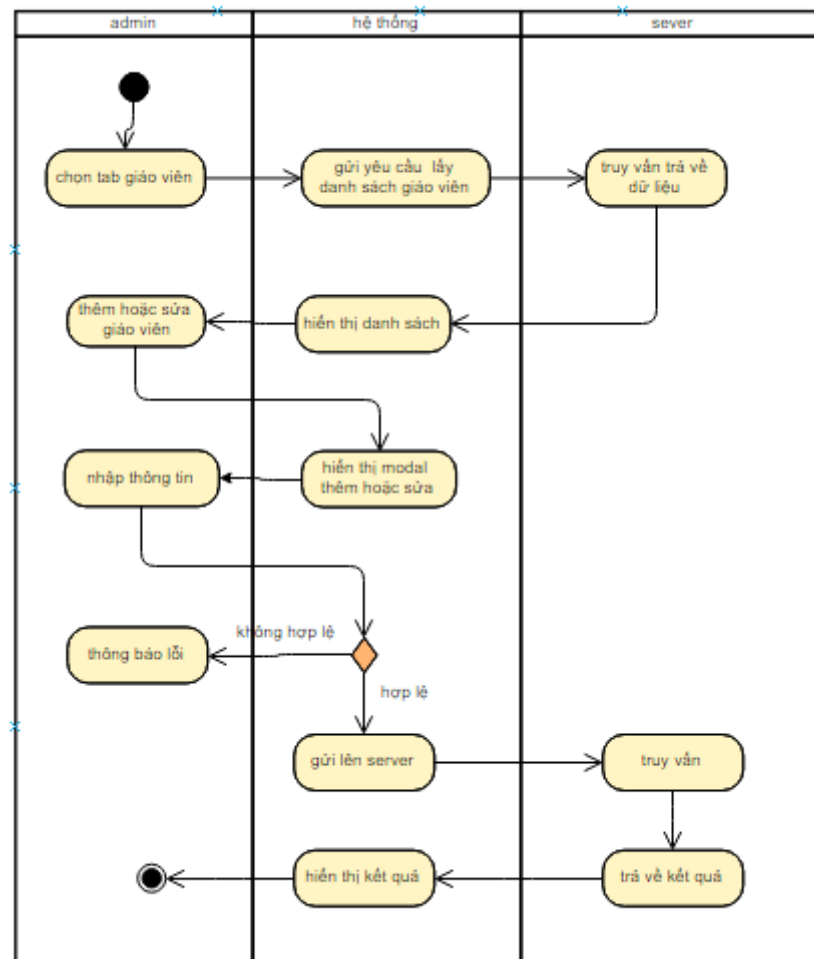
Mã UseCase	UCQLGV	Tên UseCase	Quản lý thông tin giáo viên
Tác nhân	Admin		
Điều kiện trước	Đăng nhập		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Admin	Vào tab giáo viên	
2	Hệ thống	Hiển thị giao diện danh sách giáo viên	
3	Admin	Chọn chức năng thêm/sửa giáo viên	
4	Hệ thống	Hiển thị modal thêm/sửa giáo viên	
5	Admin	Nhập hoặc chỉnh sửa thông tin giáo viên	
6	Hệ thống	Kiểm tra nhập đầy đủ các thông tin giáo viên	
7	Hệ thống	Kiểm tra mail và số điện thoại đúng định dạng	
8	Hệ thống	Gửi thông tin đã nhập lên server	
9	Server	Kiểm tra có số điện thoại giáo viên chưa	
10	Server	Nhận thông tin rồi thêm mới hoặc sửa thông tin của giáo viên trong bảng.	
11	Server	Trả về kết quả cho hệ thống	
12	Hệ thống	Hiển thị kết quả cho admin	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
6b	Hệ thống	Hiển thị thông báo lỗi khi không nhập đầy đủ thông tin của giáo viên.	
7b	Hệ thống	Hiển thị thông báo lỗi khi không nhập đúng định dạng mail hay số điện thoại	
9b	Hệ thống	Hiển thị thông báo lỗi khi số điện thoại đã tồn tại.	
Điều kiện sau	Không		

b, Dữ liệu đầu vào

Bảng 3.4 Bảng dữ liệu đầu vào của use case quản lý giáo viên

STT	Trường dữ liệu	Bắt buộc	Điều kiện hợp lệ/ Mô tả	Ví dụ
1	Số điện thoại giáo viên	x	Số	0978876554
2	Tên giáo viên	x	Dạng chuỗi	Lê Bá Vui
3	Địa chỉ	x	Dạng chuỗi	Hà Nội
4	Ngày sinh	x	Kiểu ngày	04/05/1988
5	Giới tính	x	Giá trị nguyên	1: Nam, 0: Nữ
6	Email	x	Dạng chuỗi	abc@gmail.com
7	Ảnh		Dạng file	

c, Mô tả bằng biểu đồ hoạt động



Hình 3.8 Biểu đồ hoạt động mô tả quy trình quản lý giáo viên

3.3.3 Đặc tả use case quản lý lớp học

a, Luồng sự kiện phát sinh

Bảng 3.5 Đặc tả use case quản lý lớp học

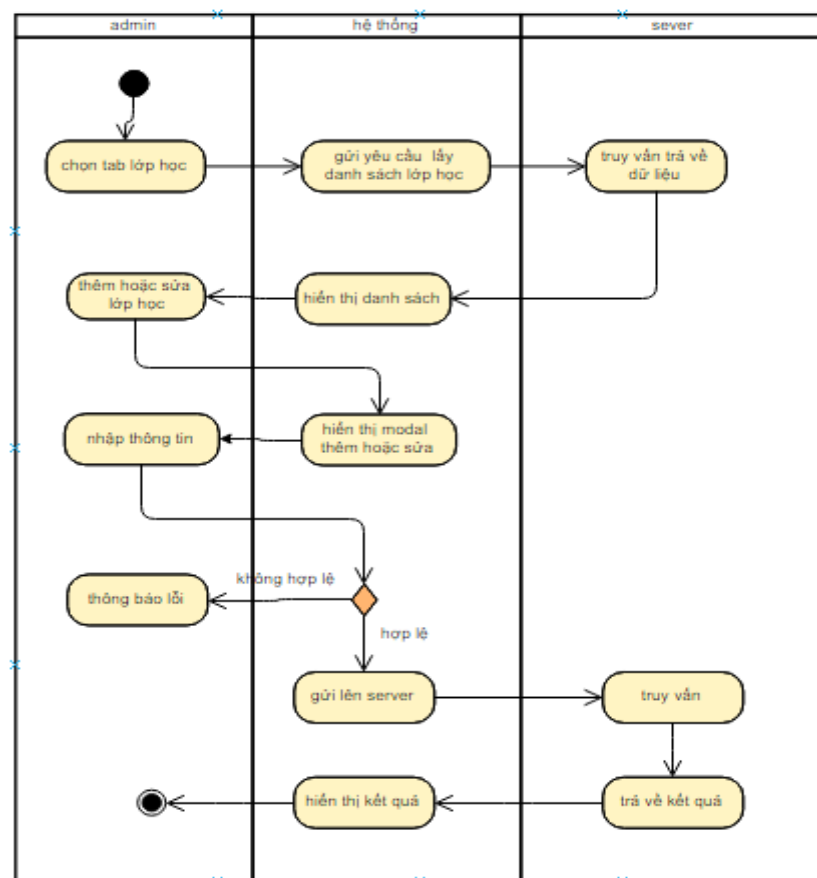
Mã UseCase	UCQLLH	Tên UseCase	Quản lý thông tin lớp học
Tác nhân	Admin		
Điều kiện trước	Đăng nhập		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Admin	Vào tab lớp học	
2	Hệ thống	Hiển thị giao diện danh sách lớp học	
3	Admin	Chọn chức năng thêm/sửa lớp học	
4	Hệ thống	Hiển thị modal thêm/sửa lớp học	
5	Admin	Nhập hoặc chỉnh sửa thông tin lớp học	
6	Hệ thống	Kiểm tra nhập đầy đủ các thông tin lớp học	
7	Hệ thống	Kiểm tra mã lớp học đã tồn tại không	
8	Hệ thống	Kiểm tra số điện thoại giáo viên tồn tại không	
9	Hệ thống	Kiểm tra có mã môn học có tồn tại không	
10	Hệ thống	Gửi thông tin đã nhập lên server	
11	Server	Nhận thông tin rồi thêm mới hoặc sửa thông tin của giáo viên trong bảng.	
12	Server	Trả về kết quả cho hệ thống	
13	Hệ thống	Hiển thị kết quả cho admin	
14	Admin	Chọn vào thông tin lớp học	
15	Admin	Nhập mã số sinh viên	
16	Hệ thống	Kiểm tra mã số sinh viên	
17	Hệ thống	Gửi mã số sinh viên đi	
18	Server	Thêm mới dữ liệu vào bảng student_class trả về kết quả	
19	Hệ thống	Hiển thị kết quả	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
6a	Hệ thống	Hiển thị thông báo lỗi khi không nhập đầy đủ thông tin của lớp học.	
7b	Server	Hiển thị mã lớp đã tồn tại	
8b	Server	Hiển thị giáo viên chưa có	
9b	Server	Hiển thị mã môn học chưa có	
16b	Server	Kiểm tra mã số sinh viên có không	
Điều kiện sau	Không		

b, Dữ liệu đầu vào

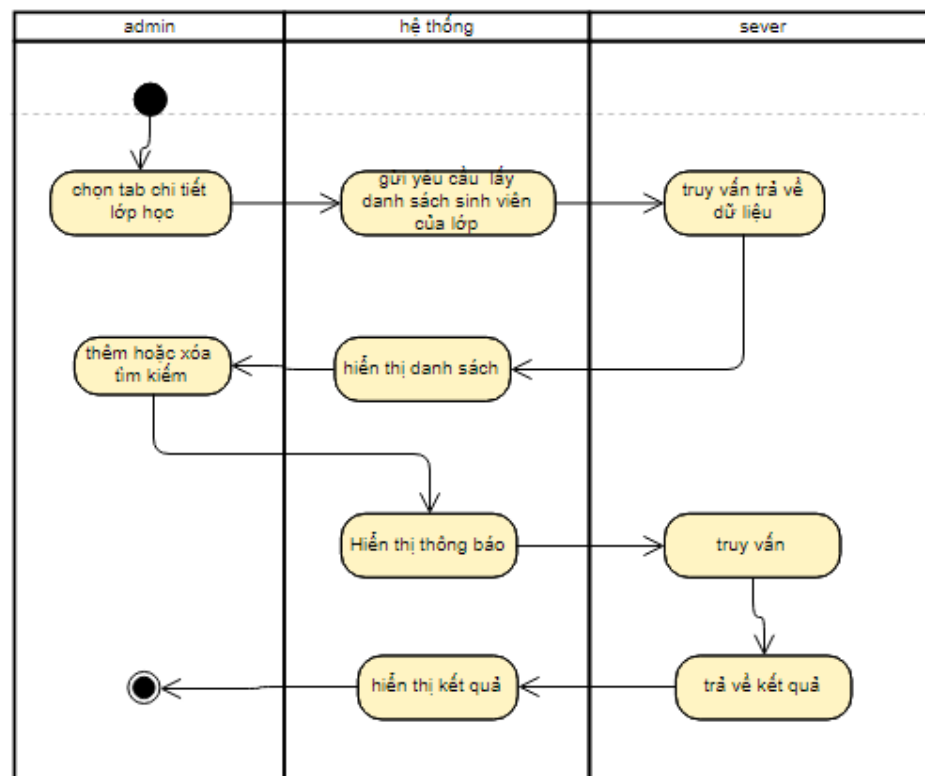
Bảng 3.6 Bảng dữ liệu đầu vào của use case quản lý lớp học

STT	Trường dữ liệu	Bắt buộc	Điều kiện hợp lệ/ Mô tả	Ví dụ
1	Mã lớp học	x	Dạng chuỗi số	234567
2	Mã môn học	x	Dạng chuỗi số	IT1010
4	Số điện thoại giáo viên	x	Số	0978876554
5	Lịch học		Chuỗi	TG:6,15h05-17h30,TC-304
6	Mã ssinh viên	x	Chuỗi số	20150959

c, Mô tả bằng biểu đồ hoạt động



Hình 3.9 Biểu đồ hoạt động mô tả quy trình quản lý lớp học



Hình 3.10 Biểu đồ hoạt động mô tả quy trình thêm sinh viên vào lớp học

3.3.4 Đặc tả use case điểm danh lớp học

a, Luồng sự kiện phát sinh

Bảng 3.7 Đặc tả use case điểm danh lớp học

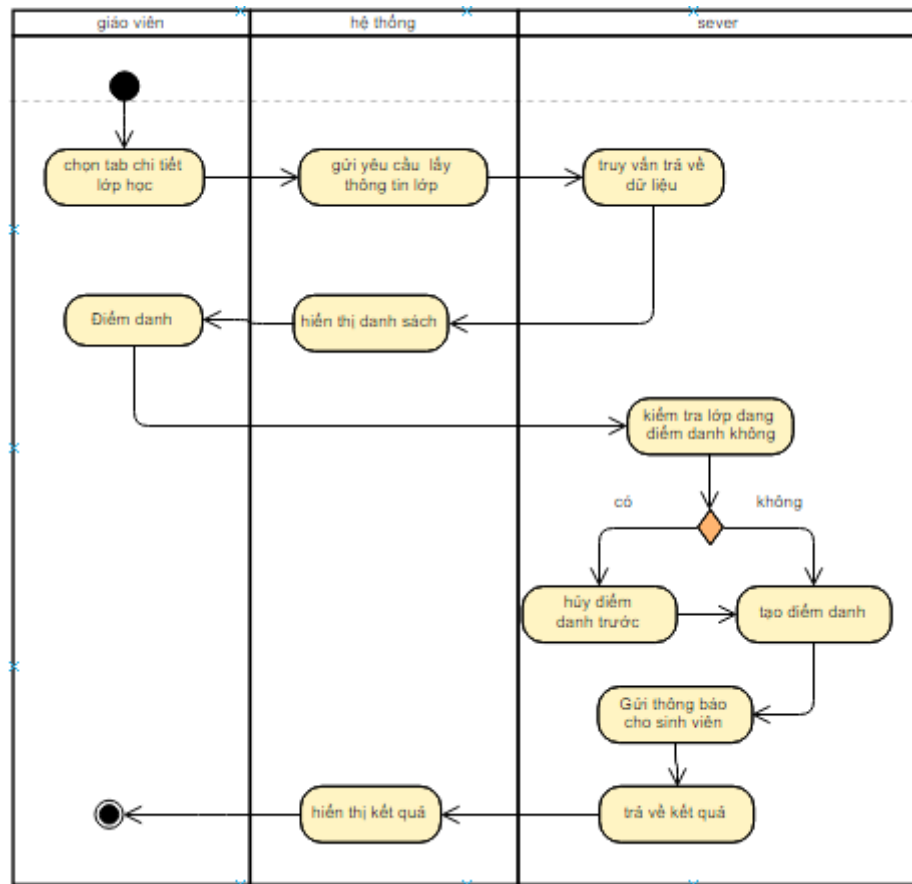
Mã UseCase	UCQLLH	Tên UseCase	Quản lý thông tin lớp học
Tác nhân	Giáo viên		
Điều kiện trước	Đăng nhập		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Giáo viên	Chọn vào lớp cần điểm danh	
2	Hệ thống	Hiển thị giao diện thông tin và danh sách sinh viên của lớp	
3	Giáo viên	Chọn chức năng điểm danh	
4	Server	Kiểm tra lớp hiện tại đang điểm danh không	
5	Server	Tạo điểm danh của lớp và sinh viên	
6	Server	Gửi thông báo cho sinh viên	
7	Server	Trả về kết quả cho hệ thống	
8	Hệ thống	Hiển thị kết quả cho ứng dụng	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
5b	Server	Nếu lớp đó đang điểm danh sẽ hủy điểm danh hiện tại và tạo điểm danh mới	
Điều kiện sau	Không		

b, Dữ liệu đầu vào

Bảng 3.8 Bảng dữ liệu đầu vào của use case quản lý lớp học

STT	Trường dữ liệu	Bắt buộc	Điều kiện hợp lệ/ Mô tả	Ví dụ
1	Mã lớp học	x	Dạng chuỗi số	234567
2	Kinh độ	x	Dạng số	56.99999999
3	Vĩ độ	x	Dạng số	74.83838383
4	Danh sách wifi có thể kết nối	x	Dạng array	[“wifi_hust”, “wifi_student”]

c, Mô tả bằng biểu đồ hoạt động



Hình 3.11 Biểu đồ hoạt động mô tả quy trình tạo điểm danh của lớp

3.3.6 Đặc tả use case điểm danh của sinh viên

a, Luồng sự kiện phát sinh

Bảng 3.9 Đặc tả use case điểm danh lớp học

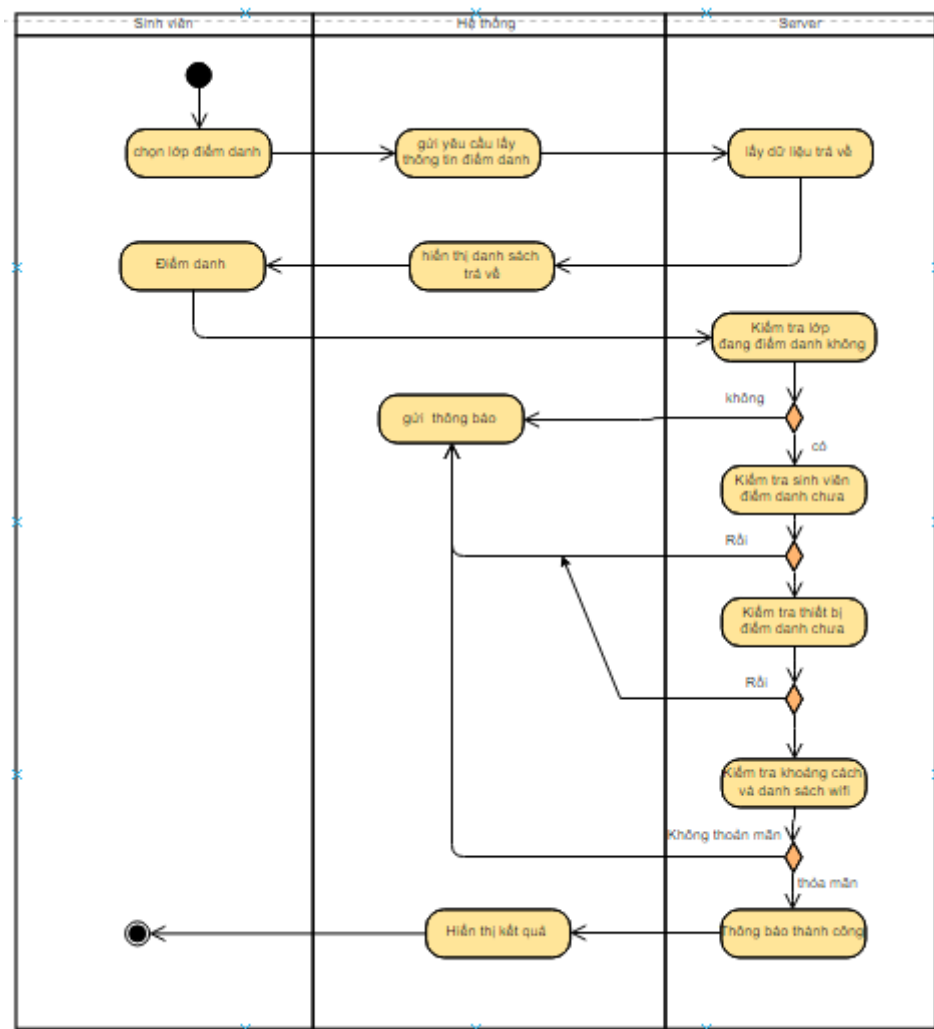
Mã UseCase	UCQLLH	Tên UseCase	Quản lý thông tin lớp học
Tác nhân	Giáo viên, sinh viên		
Điều kiện trước	Đăng nhập		
Luồng thực thi chính			
No.	Thực hiện	Hành động	
1	Sinh viên	Chọn vào lớp có thông báo điểm danh	
2	Hệ thống	Hiển thị danh sách điểm danh của lớp	
3	Sing viên	Chọn chức năng điểm danh	
4	Server	Kiểm tra lớp hiện tại đang điểm danh không	
5	Server	Kiểm tra sinh viên đã điểm danh chưa	
6	Server	Kiểm tra thiết bị đã điểm danh chưa	
7	Server	Kiểm tra vị trí thiết bị gần thiết bị giáo viên không	
8	Server	Kiểm tra danh sách wifi có wifi nào trùng với danh sách wifi của giáo viên	
9	Server	Điểm danh cho sinh viên	
10	Server	Trả về kết quả cho hệ thống	
11	Hệ thống	Hiển thị kết quả cho ứng dụng	
Luồng thực thi mở rộng			
No.	Thực hiện	Hành động	
4b	Server	Nếu lớp đó không trong trạng thái điểm danh thì thông báo về .	
5b	Server	Sinh viên điểm danh rồi thông báo đã điểm danh.	
6b	Server	Mỗi thiết bị chỉ điểm danh cho một sinh viên , nếu thiết bị đã điểm danh rồi thông báo về .	
7b	Server	Thông báo về sinh viên không ở trong lớp.	
8b	Server	Thông báo về sinh viên không ở trong lớp.	
Điều kiện sau	Không		

b, Dữ liệu đầu vào

Bảng 3.10 Bảng dữ liệu đầu vào của use case quản lý lớp học

STT	Trường dữ liệu	Bắt buộc	Điều kiện hợp lệ/ Mô tả	Ví dụ
1	Mã lớp học	x	Dạng chuỗi số	234567
2	Mã sinh viên	x	Dạng số chữ	20150959
3	Kinh độ	x	Dạng số	56.99999999
4	Vĩ độ	x	Dạng số	74.83838383
5	deviceId	x	Dạng chuỗi	d4194038-1130-4ba4-8e09-9ae44b14cc00
6	Danh sách wifi	x	Dạng array	[“wifi_hust”, “wifi_student”]

c, Mô tả bằng biểu đồ hoạt động



Hình 3.12 Biểu đồ hoạt động mô tả quy trình tạo điểm danh của sinh viên

Chương 4 Thiết kế và triển khai ứng dụng

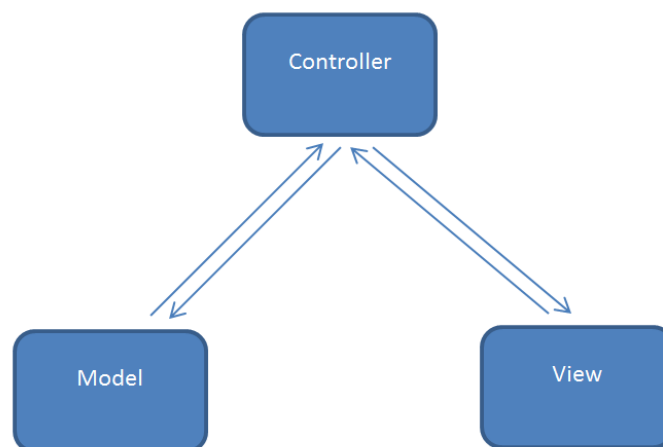
Qua chương 2, em đã trình bày về công nghệ sử dụng làm nền tảng áp dụng triển khai hệ thống. Trong chương này, em sẽ đi vào chi tiết hơn phần thiết kế hệ thống và cách triển khai. Trước hết, em sẽ mô tả chi tiết về thiết kế lớp, thiết kế giao diện và từ đó em đưa ra các kết quả đã đạt được cùng một số hình ảnh minh họa.

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

a, Nền tảng web

Trên nền tảng web, em áp dụng thiết kế theo kiến trúc Model-View-Controller hay còn gọi là MVC. Đây là mô hình được sử dụng rất phổ biến trong xây dựng website. Kiến trúc của MVC được mô tả như hình vẽ dưới đây.



Hình 4.1 Sơ đồ mô hình MVC

Chi tiết các thành phần trong mô hình MVC

- **Model** : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu.
- **View** : Để hiển thị thông tin dữ liệu và nơi chứa các nút hình ảnh ...cho người dùng xem và tương tác.
- **Controller** : là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ

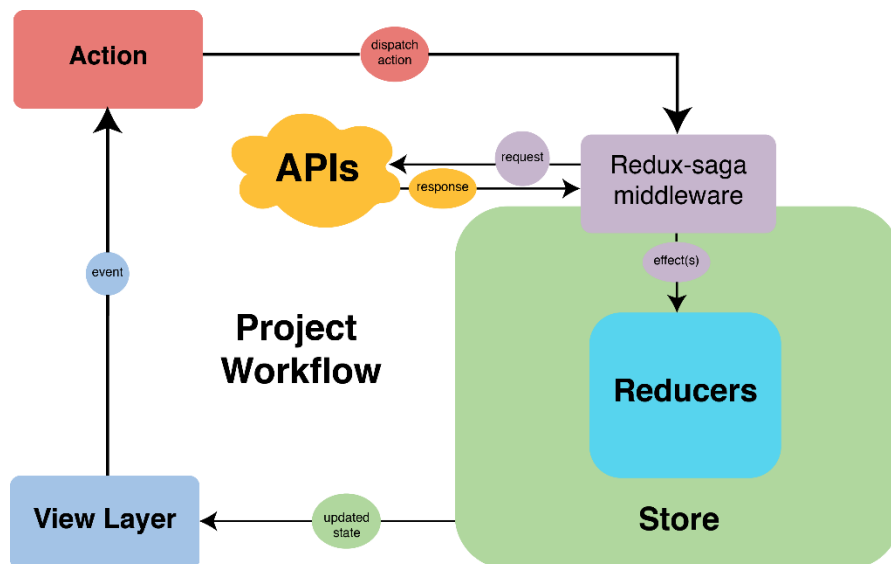
liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

Trong thiết kế web admin em ứng dụng kiến trúc vào để phân ra 3 package như sau:

- **Model** : Bao gồm AbsentStudentModel.js, AbsentClassModel.js, ClassModel.js, StudentModel.js, NotificationModel.js, ...
- **View** : Gồm những file giao diện template engine pug Class.pug, Student.pug, Teacher.pug, ...
- **Controller** : Gồm các file ClassController.js, StudentController.js, TeacherController.js, ...

b, Nền tảng mobile

Trên nền tảng mobile , em áp dụng kiến trúc redux saga để thiết kế ứng dụng. Đây là mô hình phổ biến và phù hợp với react native dùng để lấy dữ liệu về, quản lý dữ liệu hiển thị ra view.



Hình 4.2 Kiến trúc mô hình redux saga

Các thành phần của redux saga:

- **Reducers**: để mô tả các sự thay đổi của trạng thái dữ liệu khi có các action gọi đến.
- **Store** : là nơi lưu trữ trạng thái (state) hay dữ liệu của toàn bộ ứng dụng.
- **View Layer** :là nơi hiển thị dữ liệu và chứa các button thao tác của người dùng.
- **Action**: là để thay đổi các state.
- **Redux-saga**: là một thư viện redux middleware, khi ứng dụng thực hiện hành động gửi request lên server sẽ được gọi đến để lấy dữ liệu về.

Em ứng dụng kiến trúc này để thiết kế ứng dụng mobile của e như sau:

- **Reducers:** chứa các reducer ListClassReducer,ListAbsentReducer...
- **Action:** absent,getlistclass..
- **View Layer :** ClassScreen, UserScreen,LoginScreen...
- **Redux-sage:** gồm NetworkSaga
- **Store:** file index.

4.1.2 Thiết kế tổng quan

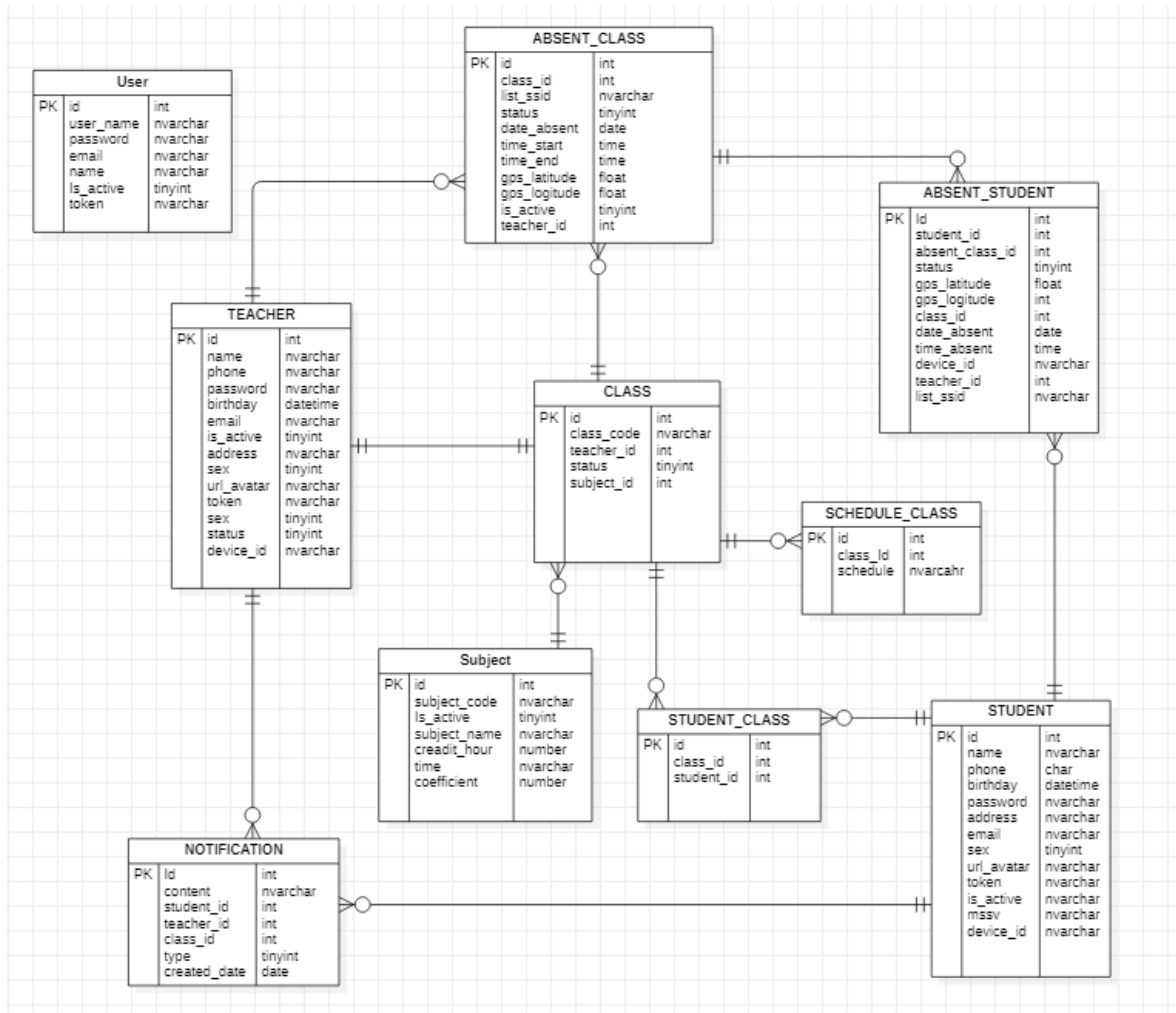
Hệ thống bao gồm 3 thành phần lớn bao gồm:

- **Web admin :** dành cho tác nhân admin .Thành phần này được xây dựng trên ngôn ngữ javascript với Nodejs .Thành phần web admin gồm chức năng quản lý sinh viên , quản lý giáo viên , quản lý môn học , quản lý lớp học.
- **Ứng dụng dành cho giáo viên:** dành cho tác nhân giáo viên. Thành phần này xây dựng trên javascript với react native. Ứng dụng gồm chức năng xem lịch dạy , điểm danh lớp,hủy điểm danh lớp ,điểm danh cho sinh viên , hủy điểm danh sinh viên.
- **Ứng dụng cho sinh viên:** dành cho tác nhân sinh viên. Ứng dụng có chức năng sinh viên điểm danh và xem lịch học.

4.2 Thiết kế cơ sở dữ liệu

4.2.1 Sơ đồ thực thể liên kết

Sau đây là biểu đồ thực thể liên kết cho hệ thống.



Hình 4.3 Sơ đồ thực thể liên kết

4.2.2 Thiết kế chi tiết bảng

Bảng User : dùng để lưu thông tin tài khoản đăng nhập vào web admin.

Bảng 4.1 Bảng User trong cơ sở dữ liệu

User			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID tài khoản
user_name	nvarchar(100)		Tên tài khoản
password	nvarchar(50)		Mật khẩu
email	nvarchar(100)		Email admin
name	nvarchar		Tên admin
Is_active	int	1	Có hoạt động không 1 hoạt động 0 là bị xóa
CreatedDate	datetime	getdate()	Ngày tạo
Token	nvarchar		Mã đăng nhập

Bảng Student: Lưu trữ thông tin sinh viên

Bảng 4.2 Bảng Student trong cơ sở dữ liệu

Student			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID sinh viên
name	nvarchar(100)		Tên Sinh viên
password	nvarchar(50)		Mật khẩu
email	nvarchar(100)		Email admin
phone	nvarchar		Số điện thoại
birthday	date		Ngày sinh
is_active	int		Có hoạt động không 1:hoạt động 0:không hoạt động
address	nvarchar		Địa chỉ
mssv	nvarchar		Mã số sinh viên
sex	int	1	Giới tính 1:nam 0 :nữ
url_avatar	nvarchar		Link ảnh sinh viên
Token	nvarchar		Mã đăng nhập
Device_id	nvarchar		Mã thiết bị đăng nhập

Bảng Teacher :Bảng giáo viên lưu trữ thông tin giáo viên.

Bảng 4.3 Bảng Teacher trong cơ sở dữ liệu

Teacher			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID giáo viên
name	nvarchar(100)		Tên giáo viên
password	nvarchar(50)		Mật khẩu
email	nvarchar(100)		Email admin
phone	nvarchar		Số điện thoại
birthday	date		Ngày sinh
is_active	int	1	Có hoạt động không 1:hoạt động 0:không hoạt động
address	nvarchar		Địa chỉ
sex	int	1	Giới tính 1:nam 0 :nữ
url_avatar	nvarchar		Link ảnh giáo viên
Token	nvarchar		Mã đăng nhập
Device_id	nvarchar		Mã thiết bị đăng nhập
status	int	2	Trạng thái làm việc 0:ngỉ 1:Đang thực tập 2 đang làm

Bảng Notification :Bảng thông báo lưu trữ thông tin các thông báo về cho ứng dụng mobile.

Bảng 4.4 Bảng Notification trong cơ sở dữ liệu

Notification			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID thông báo
content	nvarchar		Nội dung thông báo
Student_id	Int		Id sinh viên nhận thông báo
Teacher_id	Int		id giáo viên nhận thông báo
Class_id	int		id lớp học
type	int		Kiểu thông báo

Bảng Class : lưu thông tin của lớp học

Bảng 4.5 Bảng Class trong cơ sở dữ liệu

Class			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID lớp học
Class_code	nvarchar		Mã lớp học
status	Int	1	Trạng thái lớp học 0: bị hủy 1: đang học 2 hoàn thành
Teacher_id	Int		id giáo viên của lớp
Subject_id	int		Id môn học

Bảng Subject : lưu thông tin của môn học.

Bảng 4.6 Bảng Subject trong cơ sở dữ liệu

Subject			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID môn học
subject_code	nvarchar		Mã môn học
Subject_name	nvarchar		Tên môn học
Teacher_id	Int		id giáo viên của lớp
Credit_hour	int		Số tín chỉ
time	nvarchar		Thời lượng
coefficient	int		Tín chỉ học phí

Bảng Schedule : lưu thông tin của lịch học của lớp.

Bảng 4.7 Bảng Schedule trong cơ sở dữ liệu

Schedule			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính, ID lịch học
class_id	int		Mã lớp học
schedule	nvarchar		Lịch học

Bảng Student_Class : lưu những sinh viên trong lớp

Bảng 4.8 Bảng Student_class trong cơ sở dữ liệu

Student_class			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính
class_id	int		Mã lớp học
Student_id	int		Mã sinh viên

Bảng Absent_class : Lưu dữ thông tin điểm danh của lớp.

Bảng 4.9 Bảng Absent_class trong cơ sở dữ liệu

Absent_class			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính
class_id	int		Id lớp học
list_ssld	nvarchar		Danh sách wifi có thể kết nối
status	int	1	Trạng thái của điểm danh 1:Đang điểm danh 0:Hết thời gian điểm danh
is_active	int	1	Điểm danh hoạt động 1: có 0: bị hủy
date_absent	date		Ngày điểm danh
time_start	time		Thời gian bắt đầu điểm danh
time_end	time		Thời gian kết thúc điểm danh
gps_latitude	float		Vĩ độ của thiết bị điểm danh của giáo viên
gps_longitude	float		Kinh độ của thiết bị điểm danh của giáo viên
teacher_id	int		Id giáo viên

Bảng Absent_Student : Lưu trữ thông tin điểm danh của sinh viên.

Bảng 4.10 Bảng Absent_student trong cơ sở dữ liệu

Absent_student			
Tên trường	Kiểu dữ liệu	Dữ liệu mặc định	Mô tả
id	Int identify		Khóa chính
student_id	int		Id sinh viên
list_ssid	nvarchar		Danh sách wifi có thể kết nối
status	int	0	Trạng thái của điểm danh 1: đã điểm danh 0: Chưa điểm danh
date_absent	date		Ngày điểm danh
time_absent	time		Thời gian điểm danh
gps_latitude	float		Vĩ độ của thiết bị điểm danh của giáo viên
gps_longitude	float		Kinh độ của thiết bị điểm danh của giáo viên
class_id	int		Id lớp học
teacher_id	int		Id giáo viên
device_id	nvarchar		Mã thiết bị điểm danh

4.3 Triển khai

4.3.1 Web admin

Do chưa có điều kiện, em chưa thể triển khai thực tế. Em triển khai website và các dịch vụ web thử nghiệm local ở máy tính cá nhân và ứng dụng mobile ở các thiết bị di động Android. Các bước thực hiện triển khai:

Bước 1 Cài đặt môi trường phát triển:

- Node.js: Để xây dựng ứng dụng phía Client sử dụng Angular, đầu tiên em thực hiện cài đặt Node.js với phiên bản mới nhất v14.4.0, các bước hướng dẫn cài đặt trình bày cụ thể tại <https://nodejs.org/en/>.
- yarn: công cụ quản lý package, thư viện của javascript .Cài đặt theo hướng dẫn trên <https://classic.yarnpkg.com/>.

Bước 2 Chạy project

- Mở project rồi chạy lệnh yarn để tải về các thư viện sử dụng trong project sau đó chạy lệnh yarn start để bắt đầu chạy project.

```
D:\Do_an_tot_nghiep\ApiExpress>yarn
yarn install v1.22.0
[1/4] Resolving packages...
success Already up-to-date.
Done in 0.74s.
```

Hình 4.4 Chạy lên yarn

```
D:\Do_an_tot_nghiep\ApiExpress>yarn start
yarn run v1.22.0
$ nodemon ./bin/www --exec babel-node
[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `babel-node ./bin/www`
Listening on port 3000
```

Hình 4.5 Chạy lệnh yarn start

Bước 3 Truy cập trình duyệt ở địa chỉ <http://localhost:3000/> để xem kết quả.

4.3.2 Ứng dụng mobile

Bạn download file apk rồi cài vào máy điện thoại về hoặc chạy project như sau:

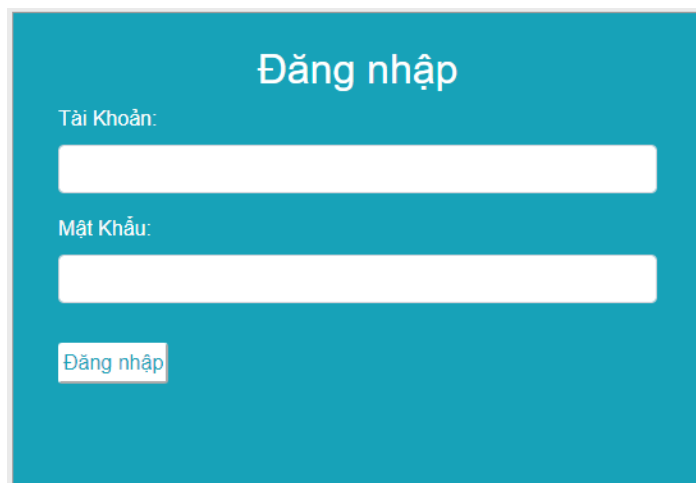
Bước 1 Đầu tiên cài đặt nodejs trước sau đó đặt môi trường react native theo hướng dẫn trên trang <https://reactnative.dev/docs/environment-setup>.

Bước 2 Mở project rồi chạy lệnh yarn để dowload module về sau đó chạy lệnh yarn android để chạy ứng dụng trên hệ điều hành android và yarn ios để chạy ứng dụng trên hệ điều hành ios. Bạn cần kết nối máy tính với thiết bị di động và thiết bị di động cần bật chế độ nhà phát triển.

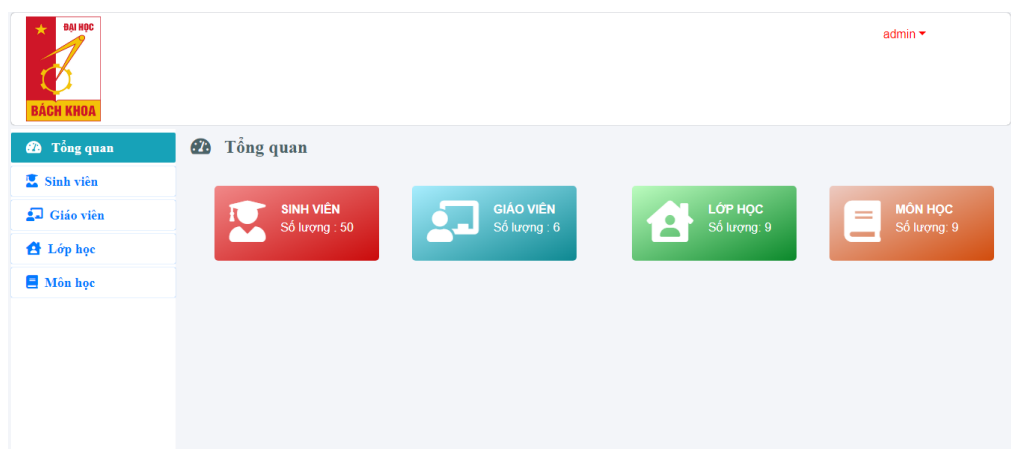
4.4 Thiết kế giao diện

4.4.1 Web admin

Sau đây là một số giao diện web em thiết kế và hoàn thành:



Hình 4.6 Giao diện đăng nhập



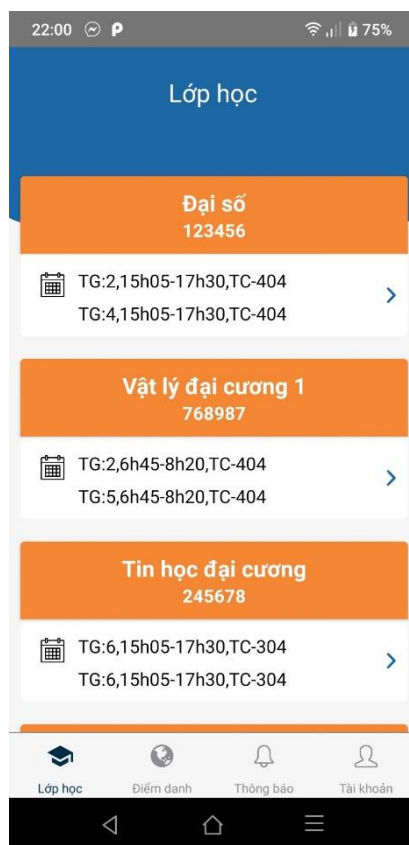
Hình 4.7 Giao diện màn chính

STT	Lịch học	Mã lớp	Mã HP	Tên môn học	Tên Giáo viên	SL	Trạng thái
1	TG:6,15h05-17h30,TC-304 TG:6,15h05-17h30,TC-304	245678	IT1010	Tin học đại cương	Lê Bá Vui	10	Đang học
2	TG:2,6h45-8h20,TC-404 TG:5,6h45-8h20,TC-404	768987	PH1010	Vật lý đại cương 1	Lê Bá Vui	3	Đang học
3	TG:2,6h45-9h05,D3-201 TG:2,6h45-9h05,D3-201	234555	MI1040	Giải tích 2	Nguyễn Thị Mỹ Anh	1	Đã Hoàn thành
4	TG:2,6h45-9h05,D3-201	456789	MI1010	Giải tích 1	Nguyễn Đức Tiến	1	Đang học
5	TG:2,15h05-17h30,TC-404 TG:4,15h05-17h30,TC-404	123456	MI1030	Đại số	Lê Bá Vui	1	Đang học

Hình 4.8 Giao diện danh sách lớp học

4.4.2 Ứng dụng di động

Sau đây giao diện ứng dụng dành cho giáo viên và sinh viên em thiết kế :



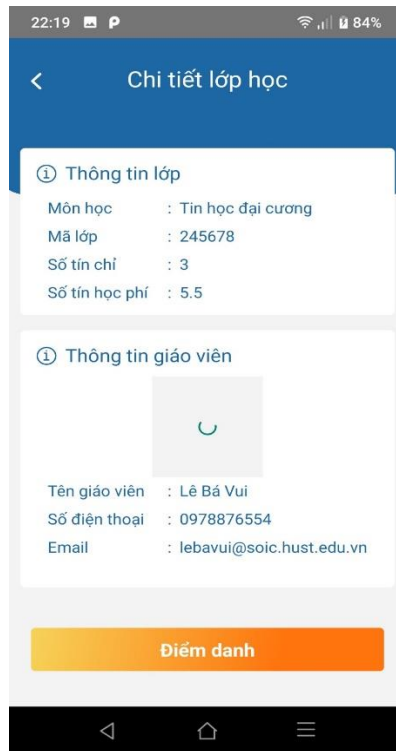
Hình 4.9 Màn danh sách lớp học của ứng dụng giáo viên



Hình 4.10 Màn danh sách học sinh của lớp trên ứng dụng giáo viên



Hình 4.11 Màn danh sách điểm danh của các lớp của ứng dụng giáo viên



Hình 4.12 Màn điểm danh của sinh viên



Hình 4.13 Màn danh sách điểm danh của ứng dụng sinh viên

Chương 5 Kết luận và hướng phát triển

5.1 Kết luận

Sau quá trình làm đồ án tốt nghiệp, em đã hoàn thiện hệ thống quản lý điểm danh. Nhìn chung, em thấy hệ thống đã khá đầy đủ chức năng và có thể ứng dụng vào thực tế.

Qua quá trình thực hiện đồ án, em học hỏi thêm nhiều công nghệ kiến thức và rút ra nhiều kinh nghiệm. Trước tiên là trong phát triển phần mềm, thiết kế đóng vai trò rất quan trọng. Cần tuân thủ các quy trình cũng như quy tắc thiết kế để đảm bảo sự thống nhất. Bài học thứ hai cho em đó là khi bắt tay vào giải quyết một bài toán, cần tập trung hiểu rõ các yêu cầu của bài toán, không nên vội vã làm phung phí thời gian. Và cuối cùng em học được đó là hiệu suất của hệ thống cũng là vấn đề quan trọng cần xử lý.

6.2 Định hướng phát triển

Để hoàn thiện hệ thống này e đã hiểu các kiến thức kỹ năng lập trình web như nắm vững về ngôn ngữ JavaScript cũng như HTML CSS, sử dụng git để quản lý code áp dụng những kiến thức trong quá trình học tập vào đồ án để thiết kế cơ sở dữ liệu, phân tích yêu cầu và thực hiện.

Em nhận thấy đồ án còn một số điểm để cải thiện trong phần điểm danh. Em xin đề xuất hướng phát triển đó là điểm danh bằng công nghệ nhận diện khuôn mặt.

Dưới sự hướng dẫn tận tâm của ThS Lê Bá Vui, em đã hoàn thành hệ thống đáp ứng mục tiêu ban đầu đề ra. Cũng từ đồ án, em đã học được nhiều kiến thức và kinh nghiệm quý báu cho mình.

Tài liệu tham khảo

- [1] Pug <https://pugjs.org/api/getting-started.html>
- [2] Sequelize <https://sequelize.org/master/index.html>
- [3] PostgreSQL <https://www.postgresql.org/>
- [4] React native <https://reactnative.dev/>
- [5] Redux <https://redux.js.org/>

