

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng Blockchain trong kiểm tra chứng chỉ

Nguyễn Văn Hải

Hà Nội, 12/2018

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng Blockchain trong kiểm tra chứng chỉ

Sinh viên thực hiện Nguyễn Văn Hải

Người hướng dẫn ThS. Lê Đức Trung

Hà Nội, 12/2018

Lời cam kết

Họ và tên sinh viên: Nguyễn Văn Hải

Điện thoại liên lạc: 0967135492

Email: hai213k57@gmail.com

Lớp: CNTT-TT 2.03 K57

Hệ đào tạo: Đại học chính quy

Tôi – *Nguyễn Văn Hải* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS.Lê Đức Trung*. 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. Mọi 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 tháng năm

Tác giả ĐATN

Nguyễn Văn Hải

Lời cảm ơn

Lời đầu tiên em được gửi lời cảm ơn chân thành tới các thầy cô giáo trong trường Đại Học Bách Khoa Hà Nội nói chung và các thầy cô giáo trong Viện Công Nghệ Thông Tin và Truyền Thông cũng như bộ môn Công Nghệ Phần Mềm nói riêng đã tận tình truyền đạt những kiến thức, kỹ năng bổ ích cho em trong suốt những năm học qua. Đặc biệt em cũng xin gửi lời cảm ơn chân thành đến **ThS. Lê Đức Trung**, người đã trực tiếp hướng dẫn, tận tình chỉ bảo em trong suốt quá trình nghiên cứu, thực hiện đồ án tốt nghiệp. Cuối cùng em xin gửi lời cảm ơn chân thành tới gia đình, anh chị, bạn bè đã quan tâm động viên, đóng góp ý kiến và giúp đỡ trong suốt quá trình học tập và rèn luyện tại nhà trường và công ty để hoàn thành đồ án tốt nghiệp này.

Tóm tắt

Ngày nay, với sự phát triển mạnh của công nghệ thông tin trong thời đại của cuộc cách mạng công nghiệp 4.0 thì việc chống làm giả các chứng chỉ đào tạo càng gặp nhiều khó khăn khi mà nền tảng công nghệ chưa thể hỗ trợ. Các hệ thống hiện nay vẫn lưu trữ dữ liệu tập trung, nếu một ai đó có thể đăng nhập được vào cơ sở dữ liệu của hệ thống thì họ hoàn toàn có thể chỉnh sửa dữ liệu đó theo hướng có lợi cho mình, vì thế để có được một hệ thống có độ bảo mật cao và phát hiện được các trường hợp gian lận gặp nhiều khó khăn và thách thức khi mà công nghệ Blockchain chưa được công bố.

Với việc phát triển hệ thống trợ giúp quản lý chứng chỉ đào tạo trên nền tảng công nghệ Blockchain thì các yêu cầu về nghiệp vụ trước gặp nhiều khó khăn và chưa thể giải quyết được thì nó đã trở lên không quá phức tạp và khó triển khai. Hệ thống được em xây dựng gồm các phần chính sau:

- Phát triển nền tảng web client cho phép người dùng có thể xem được trạng thái hiện tại của chứng chỉ, xem được lịch sử cập nhật chứng chỉ.
- Phát triển nền tảng web admin cho phép người quản trị hệ thống có thể quản trị được hệ thống thông qua giao diện web.
- Phát triển Server để phục vụ cho việc xử lý và trả về các yêu cầu được gửi đến từ các thành phần của hệ thống: web admin, web client, smart contracts.
- Phát triển hợp đồng thông minh để có thể ghi lại các lịch sử cập nhật lên hệ thống Blockchain.

Kết quả đạt được của ĐATN là một hệ thống có tính hoàn thiện cao đầy đủ các thành phần: Web client, Web admin, Server, Smart Contracts và có thể mở rộng cho nhiều bài toán hơn nữa. Và đây là một ứng dụng mã nguồn mở, toàn bộ mã

nguồn của hệ thống đã được đưa lên github, nên bất kỳ ai quan tâm đến đề tài này đều có thể phát triển tiếp, cũng như bổ sung thêm các tính năng mới.

Mục lục

Lời cam kết.....	3
Lời cảm ơn	4
Tóm tắt	5
Mục lục.....	7
Danh mục hình vẽ	12
Danh mục thuật ngữ	18
Chương 1 Giới thiệu đề tài.....	19
1.1 Đặt vấn đề.....	19
1.2 Mục tiêu và phạm vi đề tài	19
1.3 Định hướng giải pháp	20
1.4 Bố cục đồ án	21
Chương 2 Khảo sát và Phân tích yêu cầu	22
2.1 Khảo sát yêu cầu.....	22
2.2 Biểu đồ use case	23
2.2.1 Biểu đồ use case tổng quan.....	23
2.2.2 Biểu đồ use case phân rã cho chức năng quản lý tài khoản.....	24
2.2.3 Biểu đồ use case phân rã cho chức năng phân quyền.....	25

2.2.4	Biểu đồ use case phân rã cho chức năng quản lý sinh viên.....	26
2.2.5	Biểu đồ use case phân rã cho chức năng quản lý lớp học	27
2.2.6	Biểu đồ use case phân rã cho chức năng quản lý chứng chỉ	28
2.3	Quy trình nghiệp vụ.....	30
2.3.1	Quy trình nghiệp vụ phân quyền cho các tài khoản.....	30
2.3.2	Quy trình nghiệp vụ phát hành chứng chỉ	31
2.4	Đặc tả use case.....	32
2.4.1	Đặc tả use case quản lý tài khoản	32
2.5	Yêu cầu phi chức năng	35
2.5.1	Hiệu năng của hệ thống	35
2.5.2	Tính tin cậy của hệ thống	35
2.5.3	Tính mở rộng của hệ thống	35
Chương 3 Công nghệ sử dụng.....		36
3.1	Blockchain	36
3.1.1	Giới thiệu về Blockchain	36
3.1.2	Kiến trúc của Blockchain.....	38
3.1.3	Ưu và nhược điểm của Blockchain.....	39
3.1.4	Ứng dụng.....	41
Sản xuất.....		41
Y tế.....		41

Giáo dục	42
Tài chính.....	42
Thương mại điện tử.....	42
3.2 AngularJS	43
3.2.1 Giới thiệu AngularJS	43
3.2.2 Các thành phần của AngularJS	43
3.2.3 Ưu điểm và nhược điểm của AngularJS.....	45
3.3 NodeJS	45
3.3.1 Giới thiệu NodeJS.....	45
3.3.2 Ưu điểm và nhược điểm của NodeJS	46
Chương 4 Phát triển và triển khai ứng dụng	48
4.1 Thiết kế kiến trúc	48
4.1.1 Lựa chọn kiến trúc phần mềm	48
4.1.2 Thiết kế tổng quan	50
4.1.3 Thiết kế chi tiết gói	52
4.2 Thiết kế chi tiết	56
4.2.1 Thiết kế giao diện	56
4.2.2 Thiết kế lớp.....	59
4.2.3 Thiết kế cơ sở dữ liệu	64
4.3 Xây dựng ứng dụng	70

4.3.1	Thư viện và công cụ sử dụng	70
4.3.2	Kết quả đạt được.....	72
4.3.3	Minh họa các chức năng chính.....	73
4.4	Kiểm thử	77
4.5	Triển khai.....	78
4.5.1	Yêu cầu hệ thống.....	78
4.5.2	Cách triển khai hệ thống.....	79
Chương 5 Các giải pháp và đóng góp nổi bật.....		81
5.1	Phát triển web client.....	81
5.1.1	Đặt vấn đề.....	81
5.1.2	Giải pháp.....	81
5.1.3	Kết quả đạt được	82
5.2	Phát triển server.....	84
5.2.1	Đặt vấn đề	84
5.2.2	Giải pháp	84
5.2.3	Kết quả đạt được.....	85
5.3	Phát triển web admin.....	87
5.3.1	Đặt vấn đề	87
5.3.2	Giải pháp.....	87
5.3.3	Kết quả đạt được	88

5.4 Phát triển hợp đồng thông minh.....	90
5.4.1 Đặt vấn đề.....	90
5.4.2 Giải pháp	90
5.4.3 Kết quả đạt được.....	91
Chương 6 Kết luận và hướng phát triển.....	94
6.1 Kết luận.....	94
6.2 Hướng phát triển.....	95
Tài liệu tham khảo.....	96

Danh mục hình vẽ

Hình 1: Biểu đồ Use Case tổng quát	23
Hình 2: Biểu đồ use case phân rã cho chức năng quản lý tài khoản.....	24
Hình 3: Biểu đồ use case phân rã cho chức năng phân quyền	26
Hình 4: Biểu đồ use case phân rã cho chức năng quản lý sinh viên	27
Hình 5: Biểu đồ use case phân rã cho chức năng quản lý lớp học.....	28
Hình 6: Biểu đồ use case phân rã cho chức năng quản lý chứng chỉ.....	29
Hình 7: Biểu đồ luồng hoạt động cho nghiệp vụ phân quyền tài khoản	30
Hình 8: Biểu đồ hoạt động cho nghiệp vụ phát hành chứng chỉ.....	31
Hình 9 : Các phiên bản Blockchain.....	37
Hình 10: Hình ảnh mô tả các thành phần cơ bản trong AngularJS.....	44
Hình 11: Kiến trúc mô hình MVC.	49
Hình 12: Biểu đồ package tổng quan	50
Hình 13: Hình ảnh thiết kế chi tiết gói User Interface.	52
Hình 14: Hình ảnh thiết kế chi tiết gói Presentation Logic.....	54
Hình 15: Hình ảnh thiết kế chi tiết gói Application Facade.....	55
Hình 16: Hình ảnh thiết kế chi tiết gói Data Access	56

Hình 17 : Hình ảnh giao diện của web client.....	58
Hình 18: Hình ảnh mô tả web admin	58
Hình 19: Thiết kế chi tiết của lớp AccountModel.....	59
Hình 20: Thiết kế chi tiết của lớp CetificateCategoryModel.....	60
Hình 21 : Hình ảnh thiết kế chi tiết của lớp CetificateListModel.....	61
Hình 22 :Hình ảnh đặc tả chi tiết của lớp ClassroomModel	62
Hình 23 : Thiết kế chi tiết của lớp StudentModel.....	63
Hình 24 : Hình ảnh thực thể liên kết cơ sở dữ liệu.	64
Hình 25 : Giao diện của web client.....	73
Hình 26: Hình ảnh hiển thị trạng thái chứng chỉ.....	74
Hình 27: Hiển thị lịch sử cập nhật của chứng chỉ	74
Hình 28: Hình ảnh trang đăng nhập của admin	75
Hình 29: Màn hình quản lý tài khoản	75
Hình 30: Màn hình quản lý sinh viên.....	76
Hình 31:Quản chức vụ của các tài khoản.....	76
Hình 32: Lịch sử cập nhật chứng chỉ	77
Hình 33 : Mô hình kiến trúc của hệ thống.	79
Hình 34: Giao diện trang chủ của web client.....	82
Hình 35: Thông tin chi tiết trạng thái của chứng chỉ.	83
Hình 36: Thông tin chi tiết lịch sử cập nhật chứng chỉ.	84

Hình 37: Danh sách các API của hệ thống.....	87
Hình 38: Màn hình đăng nhập vào hệ thống	88
Hình 39Màn hình quản lý tài khoản.....	88
Hình 40: Màn hình quản lý chức vụ.....	89
Hình 41: Màn hình quản lý sinh viên.....	89
Hình 42: Màn hình hiển thị lịch sử cập nhật chứng chỉ của hệ thống.	90

Danh mục bảng

Bảng 1 :Bảng đặc tả use case	32
Bảng 2: Đặc tả use case phân quyền	33
Bảng 3: Đặc tả use case quản lý sinh viên	33
Bảng 4: Đặc tả use case quản lý lớp học.....	34
Bảng 5: Đặc tả use case quản lý chứng chỉ.....	35
Bảng 6:Bảng mô tả chi tiết các gói package	51
Bảng 7: Mô tả danh sách các độ phân giải của màn hình	57
Bảng 8: Mô tả chi tiết các thuộc tính và phương thức của AccountModel.	60
Bảng 9: Đặc tả chi tiết thuộc tính và phương thức của lớp CetificateCategoryModel	61
Bảng 10 : Đặc tả chi tiết của lớp CetificateListModel.....	61
Bảng 11 :Đặc tả chi tiết của lớp ClassroomModel.	62
Bảng 12 : Đặc tả chi tiết của class StudentModel.....	63
Bảng 13 : Bảng cơ sở dữ liệu account.....	65
Bảng 14 : Bảng cơ sở dữ liệu position	65
Bảng 15: Bảng cơ sở dữ liệu account_position.....	66
Bảng 16: Bảng cơ sở dữ liệu classroom.....	66
Bảng 17: Bảng cơ sở dữ liệu student	67

Bảng 18: Bảng cơ sở dữ liệu student_classroom	67
Bảng 19: Bảng cơ sở dữ liệu cetificate_list	68
Bảng 20: Bảng cơ sở dữ liệu cetificate_category.....	68
Bảng 21: Bảng cơ sở dữ liệu classroom_management	69
Bảng 22Bảng cơ sở dữ liệu support	69
Bảng 23: Cấu trúc dữ liệu của Certificate.....	70
Bảng 24: Author.....	70
Bảng 25: Danh sách các thư viện và công cụ sử dụng.....	71
Bảng 26: Thông tin về ứng dụng.....	72
Bảng 27: Các trường hợp kiểm thử.....	78
Bảng 28: Đặc tả yêu cầu của hệ thống	78

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
EUD	End-User Development Phát triển ứng dụng người dùng cuối
GWT	Google Web Toolkit Công cụ lập trình Javascript bằng Java của Google
HTML	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
CNTT	Công nghệ thông tin
ĐATN	Đồ án tốt nghiệp
SV	Sinh viên

Danh mục thuật ngữ

Blockchain

Công nghệ chuỗi khối

Block

Là thành phần bên trong Blockchain và có dữ liệu

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Hiện nay, với sự bùng nổ của công nghệ thông tin trong thời đại của cuộc cách mạng công nghiệp 4.0, thì bài toán chống làm giả các loại chứng chỉ đào tạo càng gặp nhiều khó khăn và thách thức: Không thể biết được ai là người chỉnh sửa thông tin dữ liệu của chứng chỉ ? Thời gian chỉnh sửa là khi nào ? Nội dung chỉnh sửa gồm những gì ?. Với những thách thức đó thì em đã nghiên cứu công nghệ Blockchain để có thể giải quyết được những thách thức trong bài toán chống gian lận chứng chỉ.

Với việc ứng dụng công nghệ Blockchain vào giải quyết bài toán, thì việc phát hiện và truy xuất lịch sử cập nhật chứng chỉ trở lên dễ dàng hơn rất nhiều. Dữ liệu sau khi được ghi vào Blockchain thì không ai có quyền chỉnh sửa hay xóa nó bởi vậy nó là an toàn và minh bạch. Khi đó các vấn đề: Ai là người cập nhật thông tin ? Thời gian cập nhật là khi nào ? Nội dung cập nhật gồm những thông tin gì ? thì chúng ta rất dễ dàng tìm ra được do dữ liệu trên Blockchain được tổ chức theo dạng chuỗi, một block thì có chứa hash đại diện cho chính block đó và một hash để liên kết đến block trước nó. Với cách tổ chức dữ liệu như này làm tăng tính bảo mật cho hệ thống và những kẻ tấn công không thể thay đổi được dữ liệu của hệ thống.

1.2 Mục tiêu và phạm vi đề tài

Khi mà công nghệ Blockchain chưa được công bố thì việc giải quyết bài toán chống làm giả chứng chỉ đào tạo gặp không ít khó khăn: Dữ liệu được lưu trữ một cách tập trung vì vậy nếu ai đó có thể đăng nhập được vào hệ thống quản trị cơ sở

dữ liệu thì họ dễ dàng chỉnh sửa dữ liệu đó mà không ai biết được,... Với việc ứng dụng công nghệ Blockchain thì tập hệ thống tập trung vào giải quyết 3 vấn đề cốt lõi đó là:

- Biết được ai là người cập người cập nhật ?
- Biết được nội dung cập nhật là gì ?
- Biết được thời gian cập nhật là khi nào ?

Với việc ứng dụng công nghệ Blockchain thì làm cho hệ thống trở lên bảo mật và trong suốt, dữ liệu sau khi được ghi vào block thì không ai có quyền chỉnh sửa hay xóa nó đi và khi có bất kỳ một sự thay đổi nào thì nó đều được ghi lại. Để có thể giải quyết được 3 vấn đề chính trên thì chúng ta sẽ phát triển một hợp đồng thông minh và để có thể thực thi được hợp đồng thông minh này thì chúng ta sử dụng nền tảng ethereum.

1.3 Định hướng giải pháp

Với các yêu cầu đặt ra của bài toán thì chúng ta sẽ sử dụng công nghệ Blockchain với công nghệ Blockchain thì chúng ta có một hệ thống có độ bảo mật cao và có tính minh bạch. Để giải quyết được bài toán chống làm giả chứng chỉ thì chúng ta sẽ phát triển một hợp đồng thông minh, hợp đồng này sẽ được thực thi trên nền tảng của ethereum blockchain.

Hợp đồng thông minh (Smart contracts) về cơ bản nó là một thuật toán khi có điều kiện cho trước được đáp ứng đầy đủ thì các điều khoản hợp đồng sẽ được thực thi một cách tự động mà không cần sự can thiệp của con người hay nói cách khác nó là những ứng dụng được lưu trữ trong blockchain. Công nghệ này ra đời cho phép “cuốn sổ cái” theo dõi và ghi lại mọi hoạt động giao dịch, thỏa thuận liên quan. Với hợp đồng thông minh chúng ta sẽ giải quyết được các vấn đề: Ai là người cập nhật dữ liệu của chứng chỉ ? Nội dung cập nhật là gì ? Thời gian cập nhật là khi nào ?

1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp được tổ chức như sau.

Chương 2 trình bày về quá trình khảo sát các yêu cầu từ người dùng, đánh giá về các ứng dụng tương tự hiện có. Qua đó xác định được yêu cầu đặt ra, ta tóm tắt về các chức năng của ứng dụng. Đi sâu vào việc đặc tả chi tiết từng chức năng với phương pháp phân tích thiết kế hướng đối tượng.

Chương 3 giới thiệu về các công nghệ, nền tảng đã có sẵn. Được tích hợp, và áp dụng trong việc phát triển ứng dụng. Các yêu cầu đặt ra được giải quyết nhờ các công nghệ này, và cũng như lý do lựa chọn công nghệ đó. Và hướng dẫn việc tích hợp và cách triển khai các công nghệ đó trong ứng dụng.

Chương 4 tôi đưa ra kiến trúc phần mềm được sử dụng, thiết kế chi tiết và quá trình phát triển ứng dụng. Sau đó là các kết quả đạt được, các tính năng thú vị nhất. Và cuối cùng là việc kiểm thử trước khi triển khai ứng dụng đến người dùng cuối.

Chương 5 trình bày về các nội dung đóng góp mà tôi thấy tâm đắc nhất, cũng như các vấn đề khó khăn gặp phải mà tôi từng bước giải quyết được trong quá trình phát triển ứng dụng. Và đặc biệt là thư viện cửa sổ nổi trên màn hình được xây dựng, là bàn đạp cho các tính năng thú vị nhất của ứng dụng.

Chương 6 tôi sẽ tổng kết lại các kết quả mà mình đạt được trong quá trình thực hiện ĐATN, cũng như những việc chưa làm được để rút ra bài học kinh nghiệm cho bản thân. Và hướng phát triển trong tương lai của ứng dụng để hoàn thiện ứng dụng ở mức cao nhất có thể.

Chương 2 Khảo sát và Phân tích yêu cầu

2.1 Khảo sát yêu cầu

Hiện nay, với sự phát triển của công nghệ thông tin thì việc chống và phát hiện các trường hợp làm giả các chứng chỉ đào tạo càng gặp nhiều khó khăn hơn, cùng với đó là hàng loạt các cơ sở đào tạo giáo dục được mở ra mà không đảm bảo được chất lượng đầu ra. Gần đây cũng đã xảy ra rất nhiều vụ việc liên quan đến làm giả chứng chỉ đào tạo, gian lận thi cử như: Gian lận thi cử trong kỳ thi Tốt Nghiệp Trung Học Phổ Thông ở tỉnh Hà Giang, Sơn La, Hòa Bình,...

Khi đứng trên góc độ của một nhà tuyển dụng thì họ sẽ căn cứ vào các chứng chỉ đào tạo mà ứng viên của họ đã đạt được để từ đó tìm ra được những người phù hợp, nhưng mà khi việc làm giả các chứng chỉ đó ngày một tinh vi hơn thì các nhà tuyển dụng đã gặp khó khăn trong việc xác thực, nó gây cho quá trình tuyển chọn được những ứng viên phù hợp càng gặp nhiều khó khăn hơn.

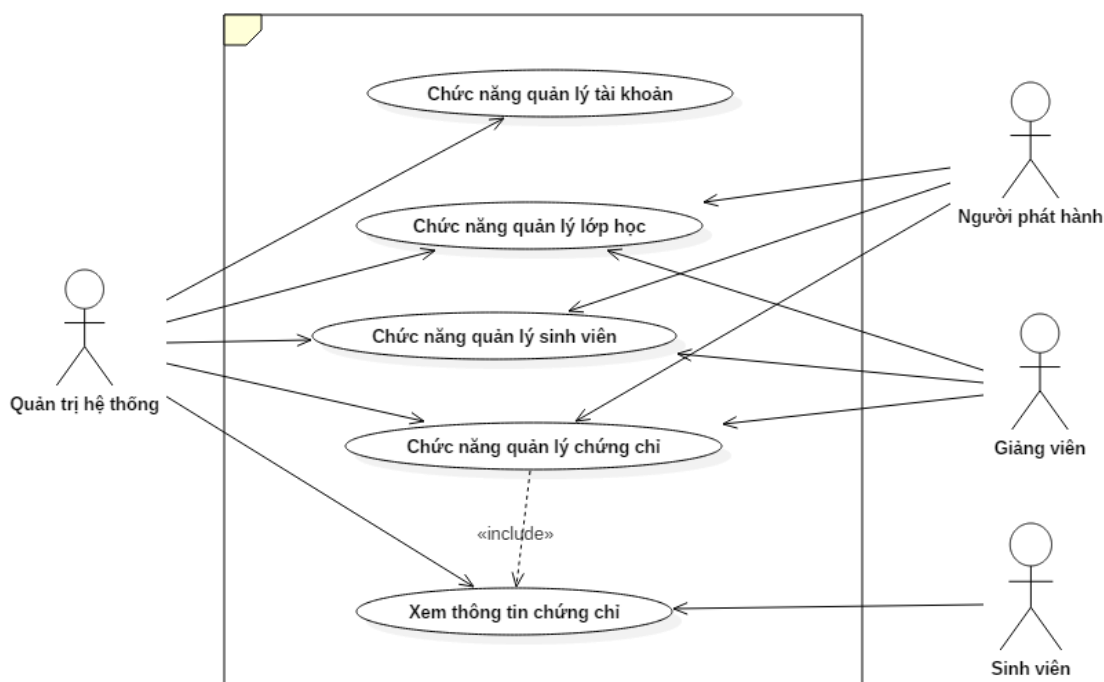
Từ những vấn đề: làm sao có thể chống làm giả các chứng chỉ, làm sao phát hiện được ai đã chỉnh sửa các nội dung của chứng chỉ, lịch sử chỉnh sửa các chứng chỉ đó,...thì em đã nghiên cứu và phát triển hệ thống trợ giúp quản lý và phát hành chứng chỉ đào tạo trên nền tảng Blockchain. Hệ thống tập chung vào các chức năng chính như sau:

- Chống việc làm giả chứng chỉ
- Truy xuất được lịch sử cập nhật chứng chỉ: ai là người cập nhật, nội dung cập nhật, thời gian cập nhật,...
- Người dùng có thể truy xuất được thông tin về chứng chỉ khi biết mã của chứng chỉ, từ đó là căn cứ giúp các nhà tuyển dụng rút ngắn được quá trình tuyển dụng và tìm ra được những ứng viên phù hợp nhất.

2.2 Biểu đồ use case

2.2.1 Biểu đồ use case tổng quan

Hình sau mô tả các use case chính của hệ thống, với mỗi use case chính là các chức năng chính của hệ thống và có 4 tác nhân là: quản trị hệ thống, người phát hành, giảng viên, sinh viên.



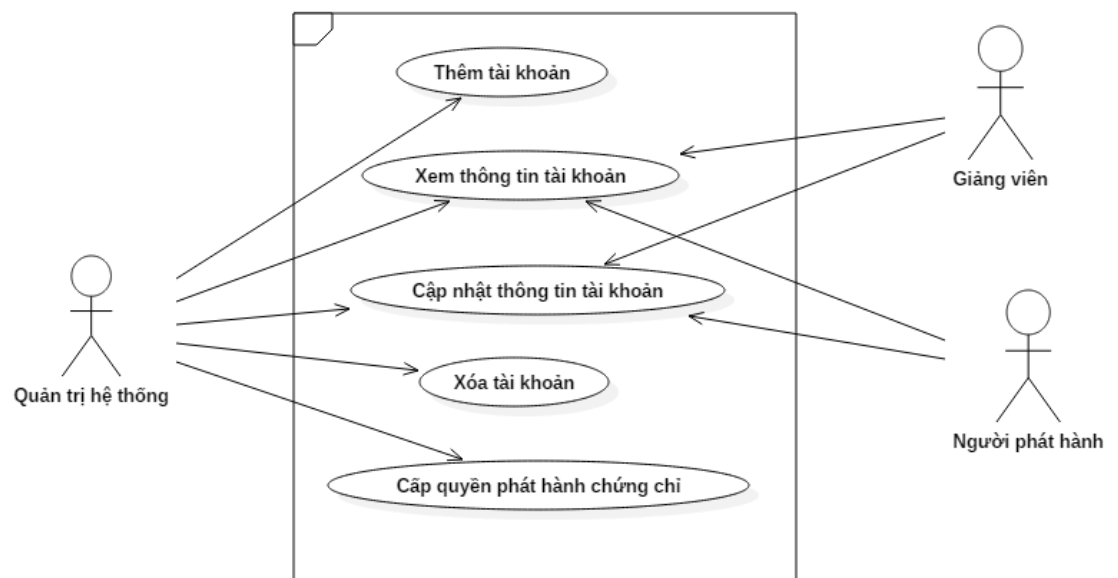
Hình 1: Biểu đồ Use Case tổng quát

Các tác nhân tham gia vào hệ thống:

- Quản trị hệ thống: là người quản trị của hệ thống, có nhiệm vụ quản lý tài khoản, phân quyền cho tài khoản, quản lý sinh viên, quản lý lớp học, quản lý chứng chỉ.
- Người phát hành: có quyền phát hành chứng chỉ và xem các thông tin về giảng viên, sinh viên, lớp học
- Giảng viên: có quyền xem các thông tin về chứng chỉ, lịch cập nhật, thông tin giảng viên, thông tin sinh viên, thông tin về lớp học
- Sinh viên: tra cứu các thông tin về chứng chỉ như: thông tin chi tiết về chứng chỉ, lịch sử cập nhật

2.2.2 Biểu đồ use case phân rã cho chức năng quản lý tài khoản

Hình sau mô tả use case cho chức năng lý tài khoản gồm có các chức năng chính: thêm tài khoản, xem thông tin của tài khoản, cập nhật thông tin của tài khoản, xóa tài khoản, cấp quyền phát hành chứng chỉ.



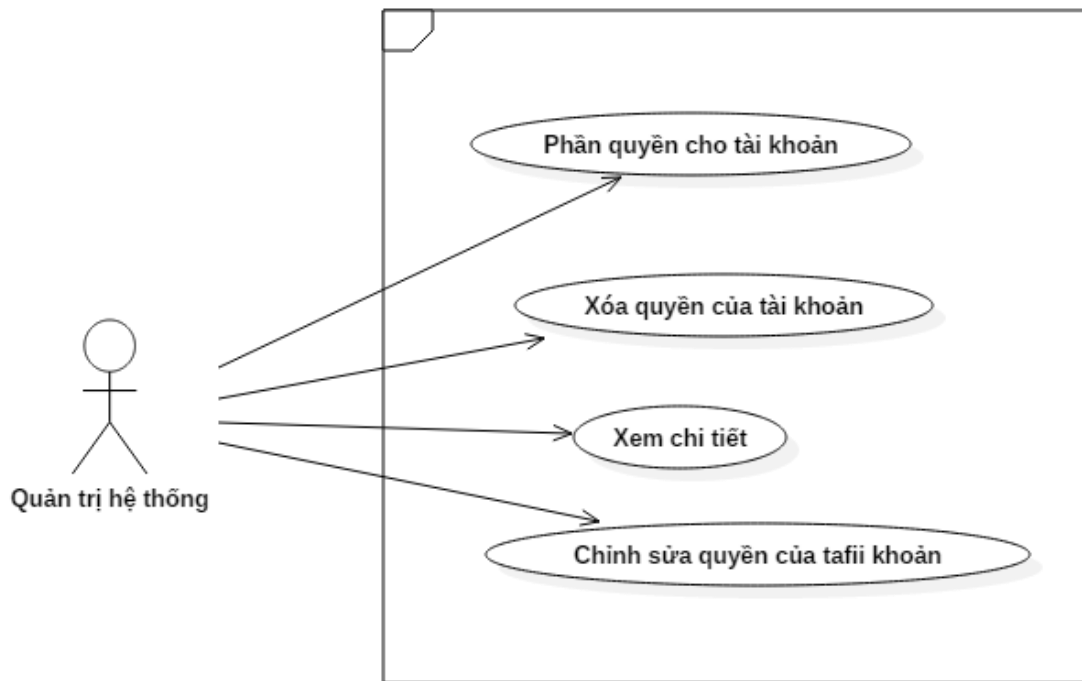
Hình 2: Biểu đồ use case phân rã cho chức năng quản lý tài khoản.

Các tác nhân tham gia vào hệ thống:

- Quản trị hệ thống: Người có quyền cao nhất đối với hệ thống và có thể: thêm tài khoản, xem thông tin chi tiết của các tài khoản có trong hệ thống, cập nhật thông tin của tài khoản, xóa tài khoản, cấp quyền phát hành chứng chỉ cho một tài khoản.
- Người phát hành: Ở đây người phát hành được hiểu là người mà quản trị hệ thống cấp quyền được phép phát hành chứng chỉ và có thể: xem thông tin tài khoản, cập nhật thông tin, phát hành chứng chỉ.
- Giảng viên: Là một tác nhân tham gia vào hệ thống và có quyền: cập nhật thông tin về tài khoản, xem chi tiết thông tin của tài khoản.

2.2.3 Biểu đồ use case phân rã cho chức năng phân quyền

Hình sau mô tả biểu đồ use case cho chức năng phân quyền gồm có các chức năng chính: phân quyền cho tài khoản đăng nhập vào hệ thống, xóa quyền của một tài khoản trên hệ thống, xem chi tiết quyền của một tài khoản, chỉnh sửa quyền đăng nhập của một tài khoản.



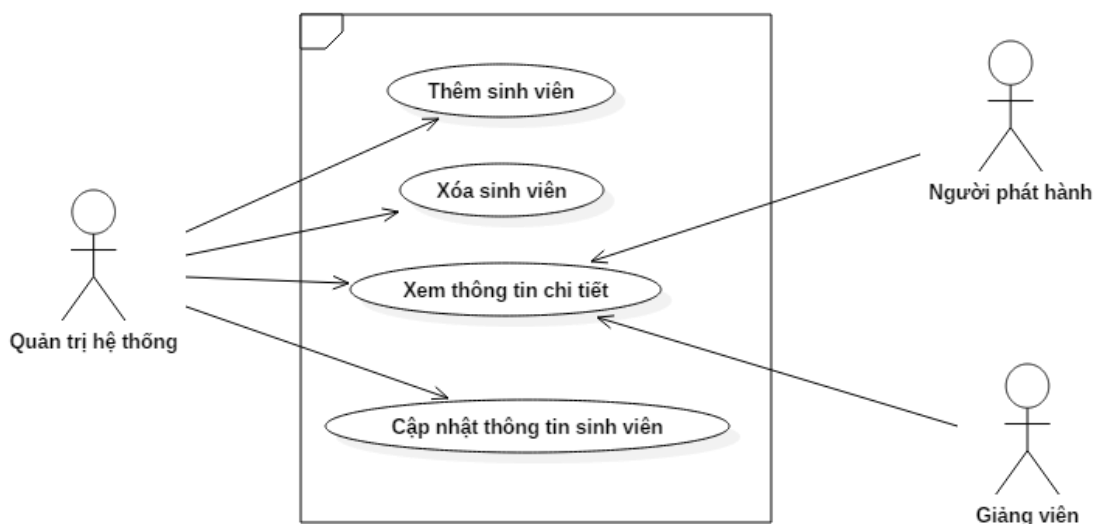
Hình 3: Biểu đồ use case phân rã cho chức năng phân quyền

Các tác nhân tham gia vào hệ thống:

- Người quản trị hệ thống: Ở use case này thì chỉ có duy nhất người quản trị của hệ thống là người được phép: cấp quyền, thu hồi quyền, xóa quyền, xem chi tiết, chỉnh sửa quyền của các tài khoản có trên hệ thống.

2.2.4 Biểu đồ use case phân rã cho chức năng quản lý sinh viên

Hình sau mô tả biểu đồ use case phân rã cho chức năng quản lý sinh viên gồm các chức năng chính: thêm sinh viên, xóa sinh viên, xem thông tin chi tiết về một sinh viên, cập nhật thông tin của sinh viên.



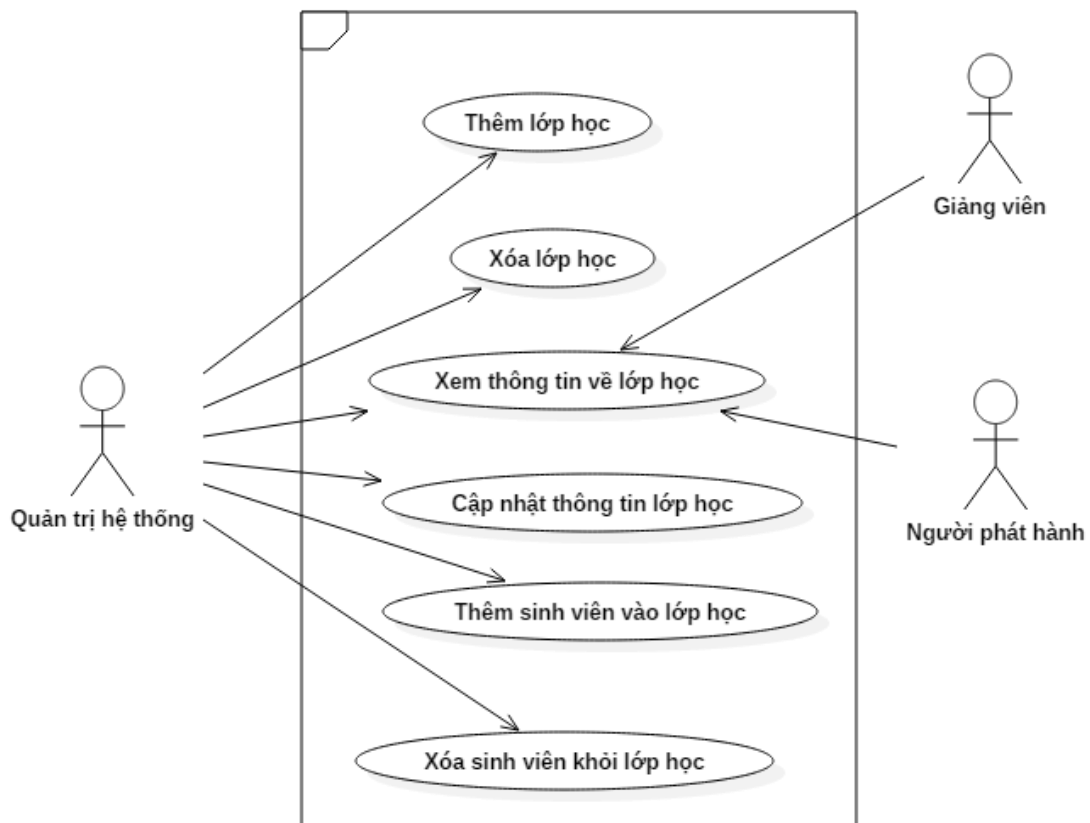
Hình 4: Biểu đồ use case phân rã cho chức năng quản lý sinh viên

Các tác nhân tham vào hệ thống:

- Quản trị hệ thống: Là người có quyền sử dụng đầy đủ các chức năng: thêm sinh viên, xóa sinh viên, xem thông tin chi tiết, cập nhật thông tin sinh viên.
- Người phát hành: Chỉ được quyền xem thông tin của một sinh viên.
- Giảng viên: Chỉ được quyền xem thông tin của một sinh viên.

2.2.5 Biểu đồ use case phân rã cho chức năng quản lý lớp học

Hình sau mô tả biểu đồ use case phân rã cho chức năng quản lý lớp học gồm các chức năng chính: thêm lớp học, xóa lớp học, xem thông tin chi tiết về lớp học, cập nhật thông tin lớp học, thêm sinh viên vào lớp học, xóa sinh viên khỏi lớp học.



Hình 5: Biểu đồ use case phân rã cho chức năng quản lý lớp học.

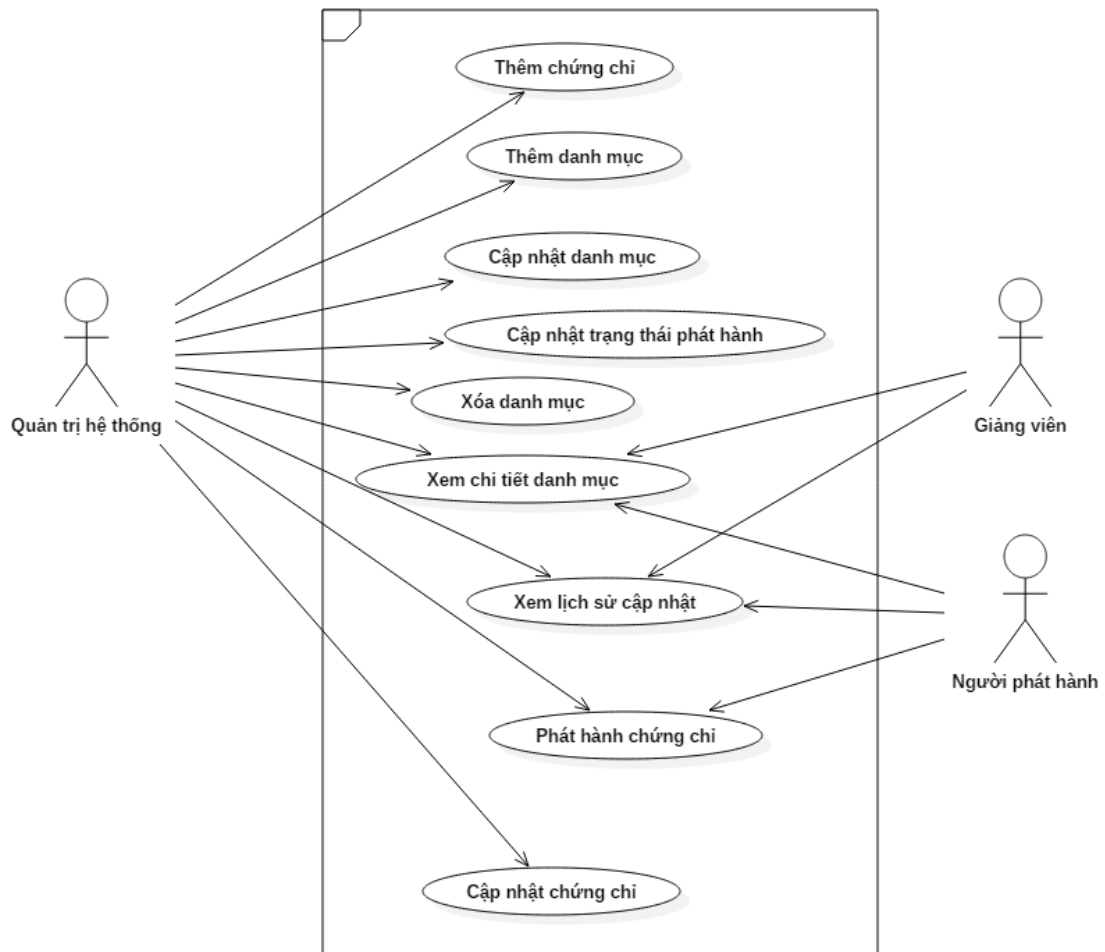
Các tác nhân tham gia vào hệ thống:

- Quản trị hệ thống: Là người có đầy đủ quyền để có thể sử dụng các chức năng: thêm lớp học, xóa lớp học, xem thông tin chi tiết về lớp học, cập nhật thông tin về lớp học, thêm sinh viên vào lớp học, xóa sinh viên khỏi lớp học.
- Người phát hành: Chỉ được phép xem thông tin chi tiết về lớp học.
- Giảng viên: Chỉ được phép xem thông tin chi tiết về lớp học.

2.2.6 Biểu đồ use case phân rã cho chức năng quản lý chứng chỉ

Hình sau mô tả biểu đồ use case phân rã cho chức năng quản lý chứng chỉ gồm có các chức năng chính: thêm danh mục, cập nhật danh mục, xóa danh mục, xem chi tiết danh mục, thêm chứng chỉ, cập nhật chứng chỉ, cập nhật trạng thái phát

hành chứng chỉ, xem lịch sử cập nhật chứng chỉ, xem chi tiết thông tin của chứng chỉ.



Hình 6: Biểu đồ use case phân rã cho chức năng quản lý chứng chỉ

Các tác nhân tham gia vào hệ thống:

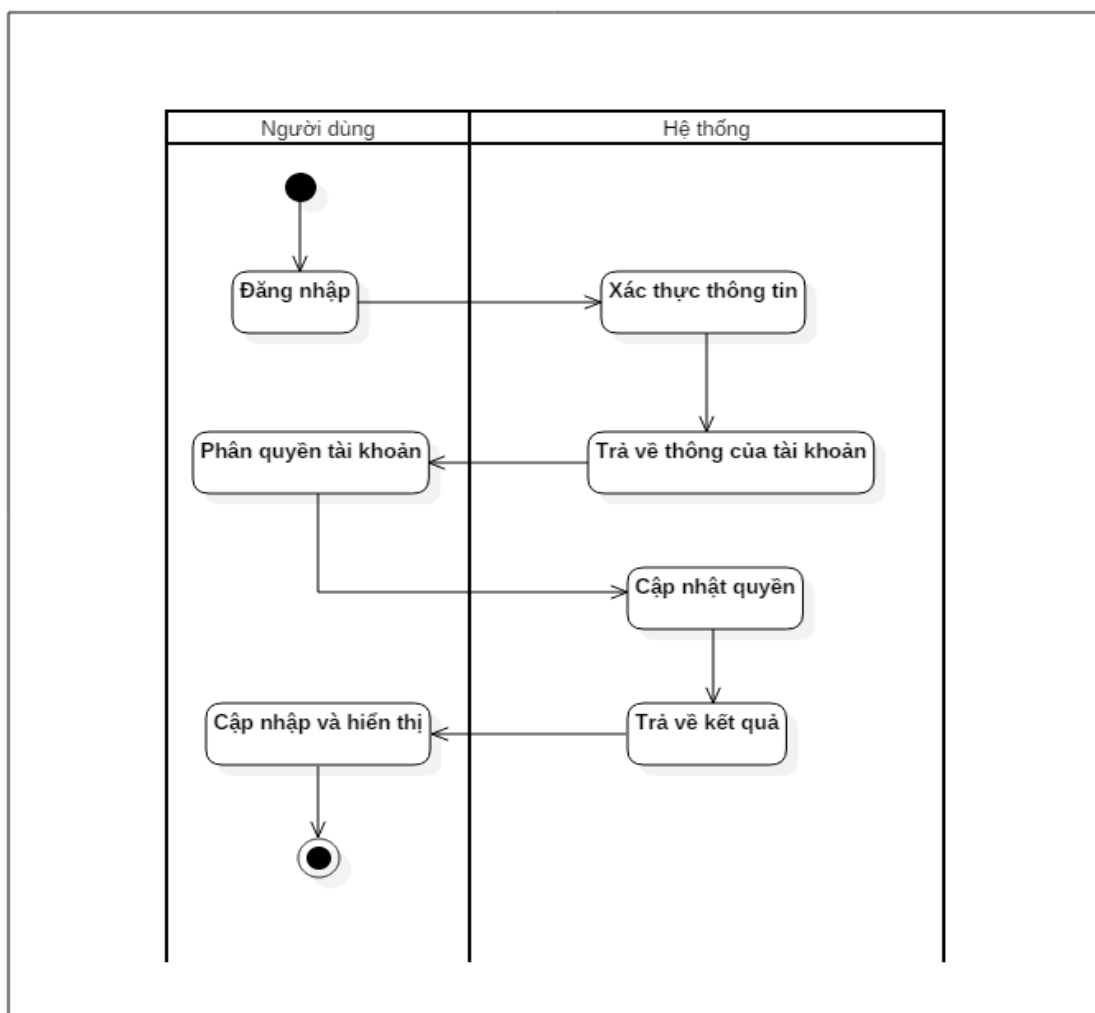
- **Quản trị hệ thống:** Tác này tham gia vào hệ thống và có đầy đủ chức năng: Thêm danh mục của chứng chỉ, cập nhật danh mục của chứng chỉ, xem chi tiết danh mục của chứng chỉ, thêm chứng chỉ, xem chi tiết thông tin về chứng chỉ, cập nhật chứng chỉ, xem chi tiết thông tin cập nhật, cập nhật trạng thái phát hành.
- **Người phát hành:** Có quyền xem thông tin về danh mục, xem thông tin về chứng chỉ, xem lịch sử cập nhật chứng chỉ, cập nhật trạng thái phát hành chứng chỉ.

- Giảng viên: Có quyền: xem thông tin về danh mục, xem thông tin chứng chỉ, xem lịch sử cập nhật của chứng chỉ.

2.3 Quy trình nghiệp vụ

2.3.1 Quy trình nghiệp vụ phân quyền cho các tài khoản

Hình sau mô tả biểu đồ hoạt động cho quy trình nghiệp vụ phân quyền cho các tài khoản trong hệ thống

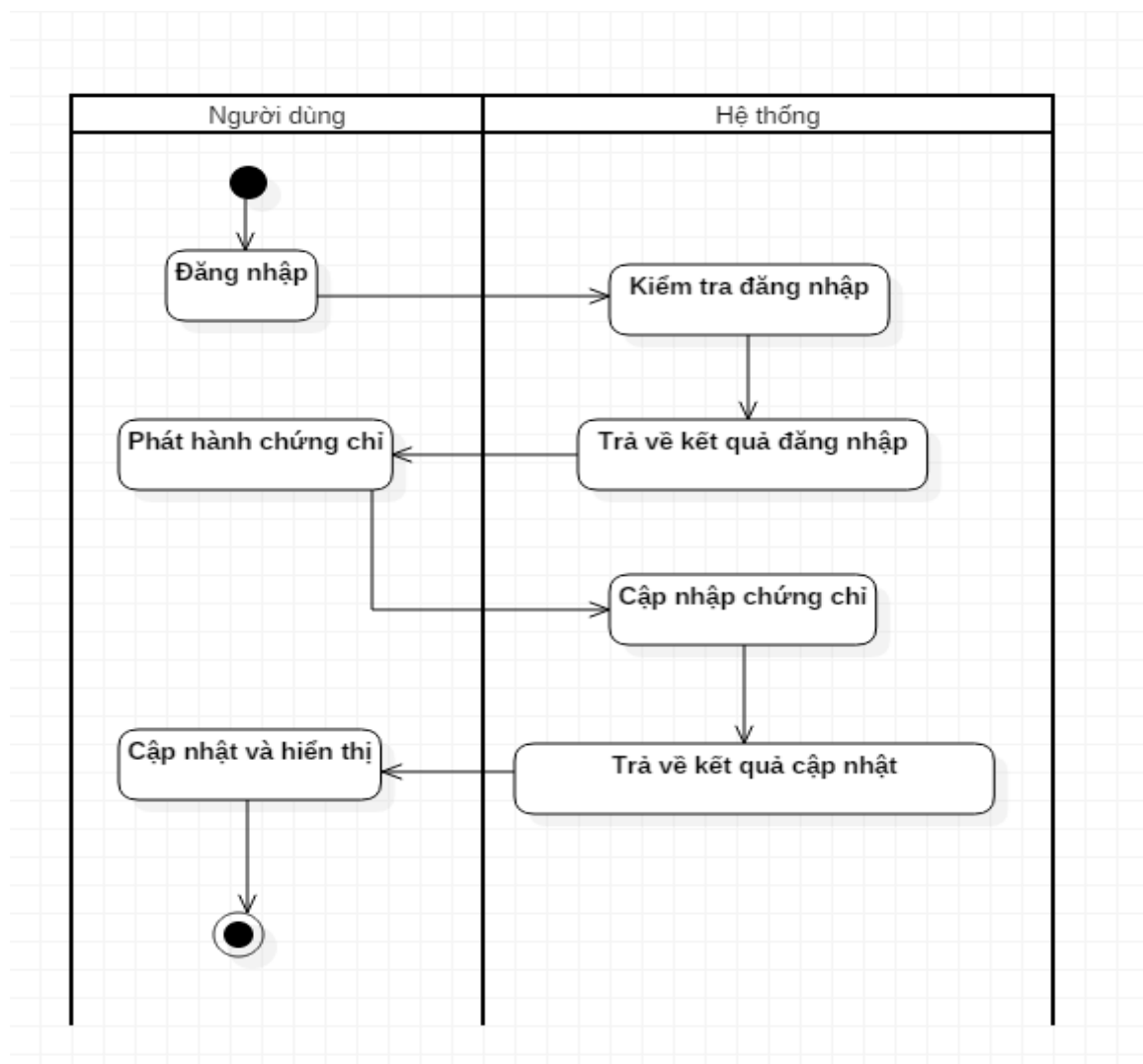


Hình 7: Biểu đồ luồng hoạt động cho nghiệp vụ phân quyền tài khoản

Khi mà người dùng đăng nhập vào hệ thống và gửi thông tin đăng nhập cho Server thì Server sẽ xác thực các thông tin và trả về thông tin của tài khoản sau khi đăng nhập thành công. Nếu người dùng được người quản trị cấp quyền để có thể phân quyền cho các tài khoản trong hệ thống thì sẽ gửi dữ liệu phân quyền lên cho Server và sau khi Server đã cập nhật thành công trạng thái phân quyền thì sẽ trả về kết quả cho người dùng.

2.3.2 Quy trình nghiệp vụ phát hành chứng chỉ

Hình sau mô tả biểu đồ hoạt động cho quy trình nghiệp vụ phát hành chứng chỉ



Hình 8: Biểu đồ hoạt động cho nghiệp vụ phát hành chứng chỉ

Khi mà người dùng đăng nhập vào hệ thống và gửi các thông tin đăng nhập lên cho Server, Server sẽ kiểm tra và trả về kết quả sau. Nếu mà người dùng đó được người quản trị cấp quyền cho phép phát hành chứng chỉ thì sẽ gửi thông tin chứng chỉ lên cho Server, lúc nào Server sẽ cập nhật chứng chỉ đó vào hệ thống Blockchain. Sau khi hoàn thành quá trình cập nhật Server sẽ trả về kết quả cập nhật cho người dùng, lúc nào người dùng sẽ thấy được thông tin về chứng chỉ được hiển thị trên giao diện web.

2.4 Đặc tả use case

2.4.1 Đặc tả use case quản lý tài khoản

Tên	Use case quản lý tài khoản
Mô tả	Biểu đồ use case này mô tả cho chức năng quản lý tài khoản, người quản trị sẽ là người có quyền sử dụng chức năng này để phân quyền cho người dùng trong hệ thống.
Tác nhân	1. Người quản trị hệ thống 2. Người phát hành (Người có quyền phát hành chứng chỉ) 3. Giảng viên
Xảy ra khi nào	Khi người quản trị muốn phân quyền cho các tài khoản có trong hệ thống
Điều kiện	Yêu cầu người dùng đăng nhập
Dòng sự kiện	1. Người dùng đăng nhập vào hệ thống. 2. Có thể thực hiện các chức năng: thêm tài khoản, xóa tài khoản, cập nhật thông tin tài khoản, xem chi tiết thông tin tài khoản, phân quyền. 3. Cập nhật và hiển thị dữ liệu lên giao diện web.
Ngoại lệ	Dữ liệu được cập nhật, nhưng không hiển thị trên giao diện của người dùng.

Bảng 1 :Bảng đặc tả use case

Tên	Use case phân quyền
Mô tả	Chức năng phân quyền giúp người quản trị hệ thống có thể giới hạn được chức năng truy cập của một số tài khoản với hệ thống.
Tác nhân	1. Người quản trị hệ thống 2. Người phát hành (Người có quyền phát hành chứng chỉ) 3. Giảng viên
Xảy ra khi nào	Khi người quản trị hệ thống muốn phân quyền cho các tài khoản có trong hệ thống.
Điều kiện	Yêu cầu đăng nhập
Dòng sự kiện	1. Người quản trị hệ thống đăng nhập vào hệ thống 2. Người quản trị sau khi đăng nhập thì sẽ có thể sử dụng một số chức năng: xem thông tin chi tiết quyền, xóa quyền, phân quyền cho người dùng. 3. Cập nhật và hiển thị lên giao diện web.
Ngoại lệ	Dữ liệu được Server cập nhật nhưng không cập nhật nó lên giao diện của web.

Bảng 2: Đặc tả use case phân quyền

Tên	Use case quản lý sinh viên
Mô tả	Người quản trị viên sẽ sử dụng các chức năng: thêm, cập nhật, sửa thông tin của sinh viên có trong hệ thống và có thể thêm sinh viên vào lớp học.
Tác nhân	1. Người quản trị hệ thống 2. Người phát hành hệ thống 3. Giảng viên
Xảy ra khi nào	Người quản trị hệ thống sẽ sử dụng chức năng này để quản lý các sinh viên có trong hệ thống.
Điều kiện	Yêu cầu đăng nhập
Dòng sự kiện	1. Đăng nhập 2. Người quản trị sau khi đăng nhập vào có thể sử dụng chức năng quản lý sinh viên: thêm, cập nhật, xóa sinh viên và có thể thêm sinh viên vào lớp học. 3. Cập nhật và hiển thị dữ liệu lên giao diện
Ngoại lệ	Dữ liệu không hiển thị lên giao của người dùng sau khi cập nhật thông tin của sinh viên

Bảng 3: Đặc tả use case quản lý sinh viên

Tên	Use case quản lý lớp học
Mô tả	Biểu đồ use case quản lý lớp học mô tả chức năng quản lý lớp học của người quản trị hệ thống. Người quản trị hệ thống có thể: thêm, cập nhật, xóa thông tin về lớp học, thêm sinh viên vào lớp học.
Tác nhân	1. Người quản trị hệ thống 2. Người phát hành (Người có quyền phát hành chứng chỉ) 3. Giảng viên
Xảy ra khi nào	Người quản trị muốn: thêm, cập nhật, xóa thông tin về lớp học.
Điều kiện	Yêu cầu đăng nhập vào hệ thống và được cấp quyền để có thể sử dụng chức năng.
Dòng sự kiện	1. Người dùng đăng nhập vào hệ thống 2. Người dùng sau khi đăng nhập vào hệ thống thì có thể sử dụng chức năng: thêm, cập nhật, xóa, thêm sinh viên vào lớp học. 3. Cập nhật và hiển thị dữ liệu lên giao diện của người dùng.
Ngoại lệ	Người dùng không thể sử dụng được chức năng này do server có thể không hoạt động.

Bảng 4: Đặc tả use case quản lý lớp học

Tên	Use case quản lý chứng chỉ
Mô tả	Biểu đồ use case quản lý chứng chỉ mô tả cho chức năng quản lý chứng chỉ: phát hành, thu hồi bản phát hành, cập nhật nội dung của chứng chỉ.
Tác nhân	1. Người quản trị hệ thống 2. Người phát hành (Người được ủy quyền để có thể phát hành chứng chỉ). 3. Giảng viên
Xảy ra khi nào	Khi người quản trị hệ thống muốn cập nhật các thông tin về chứng chỉ của sinh viên có trong hệ thống.
Điều kiện	Yêu cầu người dùng đăng nhập và được ủy quyền để có thể sử dụng chức năng này
Dòng sự kiện	1. Đăng nhập vào hệ thống 2. Sau khi đăng nhập vào hệ thống thì người dùng có thể: thêm, cập nhật, xóa chứng chỉ của một sinh viên. Các thao tác chỉnh sửa thông tin về chứng chỉ đều được lưu viết trên Blockchain. 3. Cập nhật và hiển thị lên giao diện của người dùng.
Ngoại lệ	Không thể sử dụng được chức năng này do server không hoạt

	động.
--	-------

Bảng 5: Đặc tả use case quản lý chứng chỉ

2.5 Yêu cầu phi chức năng

2.5.1 Hiệu năng của hệ thống

Hệ thống sử dụng mô hình client-server vì vậy độ trễ từ phía server trả về là không thể tránh khỏi, vì thế hệ thống cần đảm bảo được độ trễ kết quả gửi về cho client ở trong mức cho phép. Công nghệ Blockchain được ứng dụng vào trong hệ thống, vì thế cần giảm thời gian xác thực một block vào Blockchain để có thể giảm thời gian xử lý của hệ thống và giảm độ trễ thời gian trả kết quả về cho client.

2.5.2 Tính tin cậy của hệ thống

Hệ thống cần đáp ứng được: Ai là người đã chỉnh sửa dữ liệu của chứng chỉ, thời gian chỉnh sửa, nội dung đã chỉnh sửa để từ đó có thể phát hiện ra được các trường hợp làm giả chứng chỉ. Khi áp dụng công nghệ Blockchain thì hệ thống có tính tin cậy, bảo mật cao và dễ dàng phát hiện ra các trường hợp làm giả chứng chỉ.

2.5.3 Tính mở rộng của hệ thống

Hệ thống cần được thiết kế để có tính mở rộng, không chỉ giải quyết được bài toán chống làm giả chứng chỉ mà còn có thể tích hợp giải pháp cho vấn đề: chống gian lận điểm thi, học bạ điện tử và rộng hơn nữa là được ứng dụng rộng vào trong giáo dục. Ứng dụng công nghệ Blockchain vào trong bài toán chống làm giả chứng chỉ là tiền đề để có thể ứng dụng công nghệ này vào các lĩnh vực: y tế, giáo dục, xuất-nhập khẩu, tài chính ngân hàng,...

Chương 3 Công nghệ sử dụng

3.1 Blockchain

3.1.1 Giới thiệu về Blockchain

Blockchain là một sổ cái kỹ thuật số phân tán hay có thể hiểu Blockchain như là cơ sở dữ liệu trong một mạng. Sổ cái được chia sẻ cho những người tham gia vào mạng lưới. Điều này cho thấy rằng trong toàn bộ hệ thống không phải chỉ có một vị trí duy nhất, một tài liệu có thể làm căn cứ đáng tin duy nhất, vì những lần sao chép cùng một phiên bản sổ cái được đặt ở nhiều nơi.

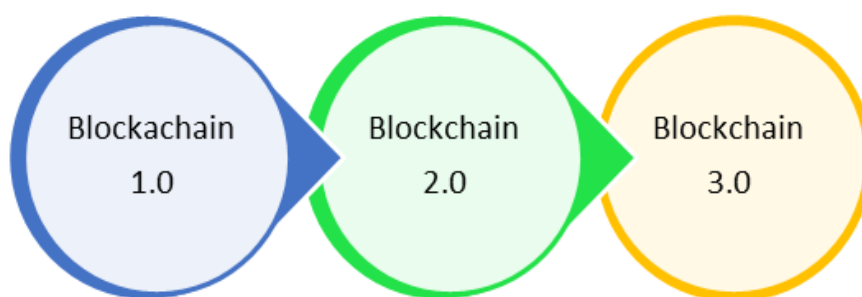
Tất cả các bản sao này được cập nhật khi dữ liệu hoặc giao dịch mới được ghi vào blockchain thông qua sự đồng thuận của tất cả mọi người tham gia. Người đào có trách nhiệm phê duyệt các giao dịch và giám sát mạng bằng cách giải quyết các công thức toán học với sự trợ giúp của máy tính. Nó là một hệ thống ngang hàng P2P, loại bỏ tất cả mọi khâu trung gian, làm tăng cường an ninh, minh bạch và sự ổn định cũng như giảm thiểu chi phí và lỗi do con người gây ra.

Các đặc điểm nổi bật của công nghệ Blockchain:

- Không thể làm giả, không thể phá hủy các chuỗi blockchain. Theo như lý thuyết thì chỉ có máy tính lượng tử mới có thể can thiệp vào và giải mã chuỗi blockchain và nó chỉ bị phá hủy hoàn toàn khi không còn internet trên toàn cầu

- Bất biến một khi những giao dịch hoặc dữ liệu đã được ghi bởi người nắm giữ private key (mã khóa bí mật - chỉ riêng người khởi tạo blockchain mới có) dữ liệu đó không thể sửa chữa nó sẽ lưu lại mãi mãi
- Bảo mật Dữ liệu: Các thông tin, dữ liệu về các chuỗi blockchain được phân tán và an toàn tuyệt đối chỉ có người nắm giữ private key mới có quyền truy xuất dữ liệu đó
- Minh bạch: Ai cũng có thể theo dõi được đường đi của blockchain từ địa chỉ này tới địa chỉ khác và có thể thống kê toàn bộ lịch sử trên địa chỉ đó.
- Hợp đồng thông minh: là các kỹ thuật số được nhúng bởi một đoạn code if-this-then-that (IFTTT), cho phép chúng tự thực thi. Trong thực tế, một bên trung gian bảo đảm rằng tất cả các bên liên quan đều tuân thủ các điều khoản. Blockchain không cần bên thứ ba, nhưng nó cũng bảo đảm rằng tất cả các bên tham gia đều biết được chi tiết hợp đồng và các điều khoản sẽ được tự động thực hiện một khi các điều kiện được bảo đảm.

Các phiên bản Blockchain:



Hình 9 : Các phiên bản Blockchain

- Blockchain 1.0 (Tiền thuật toán): Ở phiên bản này nó cho phép thực hiện các giao dịch trên nền tảng blockchain và ứng dụng điển hình là bitcoin.
- Blockchain 2.0 (Hợp đồng thông minh): Ngoài khả năng trao đổi tiền thuật toán online thì còn cung cấp nền tảng cho người dùng tự định

nghĩa các trường hợp phán xử/phán xét và quyết định khi có các yếu tố xảy ra (Hợp đồng thông minh), như thế sẽ bỏ qua bên thứ 3.

- Blockchain 3.0 (Ứng dụng phi tập trung): Cung cấp nền tảng để phát triển ứng dụng và các ứng dụng này sẽ được kích hoạt để thực hiện 1 nhiệm vụ cụ thể do chính người dùng định nghĩa ra.

3.1.2 Kiến trúc của Blockchain

Blockchain là một chain chứa các block, bên trong mỗi block thì có chứa các thông tin. Block đầu tiên trong blockchain được gọi là genesis block. Mỗi một block mới được tạo ra thì sẽ được liên kết với block trước đó. Blockchain được cấu tạo bằng ba thành phần chính.

- Block: là bản danh sách các giao dịch được ghi vào sổ cái trong một khoảng thời gian. Kích thước, thời gian và các sự kiện là tùy theo từng kiểu Blockchain khác nhau. Không phải Blockchain nào cũng ghi nhận và bảo toàn quá trình dịch chuyển tiền tệ số như là mục tiêu đầu tiên. Nhưng tất cả các Blockchain đều ghi nhận sự di chuyển token. Đơn giản hãy coi mỗi một sự trao đổi tiền tệ chỉ là ghi dữ liệu thôi. Áp giá trị vào dữ liệu chỉ là cách để định nghĩa dữ liệu đó nghĩa là gì (ví dụ như áp giá trị tiền tệ vào token trong trường hợp chuyển tiền).
- Chain: là một hash nối các block với nhau bằng thuật toán. Đây chính là sự kỳ diệu kết nối các block lại với nhau, cho phép kiến tạo ra niềm tin từ toán học. Hash trong Blockchain được tạo từ dữ liệu có sẵn từ block trước. Hash chính là dấu vân tay của dữ liệu này và kết nối, bảo mật cũng như khóa các block lại với nhau theo trật tự và thời gian. Tuy Blockchain là một sáng tạo khá mới mẻ nhưng hash thì khá cũ. Được sáng tạo hơn 30 năm trước, phát kiến này tạo hàm một chiều không thể bị phá mã. Một hash tạo ra một thuật toán ghép bản đồ dữ liệu với với bất cứ độ lớn nào vào một chuỗi dữ liệu với độ lớn cố định. Thông thường một chuỗi dữ liệu có độ dài 32 ký tự đại diện cho một bộ dữ liệu được hash. Thuật toán bảo mật hash (Secure Hash Algorithm - SHA) là một trong những hash mật mã được dùng trong

Blockchain. SHA-256 là một thuật toán thường được dùng khác có thể tạo ra một hash gần như độc nhất với độ lớn cố định là 256 bit (32 byte). Để dễ hiểu chúng ta có thể coi một hash là một dấu vân tay của dữ liệu được mã hóa trong Blockchain.

- **Mạng lưới:** là một tổ hợp của các Nút đầy đủ. Đó là các máy tính chạy thuật toán để bảo toàn cả mạng lưới. Mỗi nút chứa toàn bộ bản ghi đầy đủ của mọi giao dịch trên toàn hệ thống Blockchain. Các nút được định vị trên toàn cầu và có thể được vận hành bởi bất cứ ai. Việc vận hành một nút đầy đủ là khó khăn, tốn công sức, thời gian, tiền bạc và vì thế mà một nút phải được trả công. Thuật toán nền tảng của Blockchain sẽ trả công cho các nút này vì cung cấp nền tảng bằng tiền tệ số giống như trong trường hợp của Bitcoin. Và việc cung cấp nút cho hệ thống Blockchain Bitcoin vận hành được gọi là Đào Bitcoin.

3.1.3 Ưu và nhược điểm của Blockchain

Blockchain là một công nghệ mới và đang có xu hướng phát triển mạnh, nó có thể được áp dụng vào một số lĩnh vực như: tài chính-ngân hàng, xuất nhập khẩu, y tế ... và để có thể áp dụng được thì cần hiểu rõ được ưu, nhược điểm của công nghệ này.

Ưu điểm:

- **Tính không thể đảo ngược:** Dữ liệu giao dịch sau khi vào sổ xác nhận đủ lâu sẽ không thể đảo ngược vì các block trong sổ cái được liên kết lồng nhau, trong mã ID của một block có ‘dấu vết’ của toàn bộ các giao dịch từ đầu cộng thêm các yếu tố ngẫu nhiên.
- **Tính bền vững:** Khi đã phát triển đủ số lượng nút mạng blockchain trở nên không thể bị phá vỡ, không có nút cổ chai trên blockchain dẫn đến việc tấn công DDoS từ chối dịch vụ rất khó xảy ra.
- **Tính sẵn sàng cao:** Các máy trạm blockchain thường rải rác nhiều khu vực địa lý, hoạt động sinh block thường được hưởng tiền thưởng nên

luôn có nhiều máy online liên tục cho phép xử lý giao dịch, việc tham gia mạng không đòi hỏi cấp phép của một đơn vị quản lý tập trung thường dẫn tới số nút mạng đông đảo.

- Tính cộng đồng: Mỗi nút chỉ cần đóng góp lượng nhỏ tài nguyên là có thể tạo ra lượng tài nguyên khổng lồ về: tính toán, lưu trữ, kiểm tra / xử lý.

Nhược điểm

- Thiếu tính riêng tư: Xuất phát từ tính phi tập trung, blockchain là một mạng lưới trong đó tất cả các giao dịch trong mạng lưới đó đều có thể được truy xuất ngược bởi bất kỳ thành viên nào. Nếu bạn thanh toán hàng hóa tại một cửa hàng mà mạng thanh toán tại cửa hàng được sử dụng công nghệ blockchain, chủ cửa hàng có thể thấy giao dịch đó trên blockchain và dễ dàng truy xuất xem khoản tiền đó được gửi từ ví nào. Thậm chí, họ còn có thể kiểm tra số dư tài khoản đó cũng như toàn bộ các giao dịch gửi, rút tiền hoặc thanh toán khác từ tài khoản của bạn
- Tài nguyên của hệ thống: Mỗi node cần liên lạc với những node khác để nhận giao dịch về, xác thực giao dịch và công bố kết quả kiểm tra giao dịch. Những nhiệm vụ này làm tốn băng thông mạng, có thể ảnh hưởng lớn tới mạng Internet trong khu vực.
- Tốc độ xử lý các giao dịch: Thuật toán bằng chứng xử lý (Proof of Work) mà nhiều mạng lưới blockchain sử dụng yêu cầu chứng thực rằng các nguồn lực và công suất tính toán được đóng góp vào mạng lưới trước khi một khối được bổ sung vào mạng. Bằng chứng này được thể hiện dưới dạng lời giải cho một mảnh ghép được thêm vào khối để mạng lưới xác nhận xem có chính xác hay không.

3.1.4 Ứng dụng

Với những đặc thù của Blockchain, các chuyên gia cho rằng, công nghệ Blockchain sẽ mở ra một xu hướng ứng dụng tiềm năng cho nhiều lĩnh vực như tài chính ngân hàng, bán lẻ, vận chuyển hàng hóa, sản xuất, viễn thông... Việc áp dụng Blockchain vào cuộc sống sẽ đem lại lợi ích thật sự cho cộng đồng và toàn xã hội.

Sản xuất

Diễn hình như áp dụng đặc điểm không thể làm giả, không thể phá hủy của Blockchain vào ngành công nghiệp sản xuất sẽ giúp người tiêu dùng truy xuất được nguồn gốc xuất xứ của sản phẩm đang được chào bán. Ví dụ: Nếu một doanh nghiệp sản xuất sữa áp dụng Blockchain vào quản lý chất lượng sản phẩm thì nhà quản lý người tiêu dùng có thể truy xuất được các thông tin rất minh bạch.

Đối với nhà sản xuất họ có thể thống kê và lưu trữ toàn bộ những hộp sữa đó trên thị trường để biết được những hộp sữa đó đã tiêu thụ chưa, tiêu thụ được bao nhiêu, bao nhiêu hộp còn hạn sử dụng và bao nhiêu hộp đã hết hạn sử dụng. Đối với người tiêu dùng: Công có thể kiểm tra thông tin hộp sữa đó có phải hàng chính hãng hay không sẽ ngăn chặn toàn bộ những sản phẩm nhái, hàng giả trên thị trường.

Y tế

Khi người bệnh đi khám hay xét nghiệm, mọi kết quả của họ sẽ được lưu trữ sử dụng công nghệ blockchain sẽ giúp người bệnh bảo mật toàn bộ thông tin và chỉ số xét nghiệm của mình. Trong trường hợp người bệnh có nhu cầu chuyển sang bệnh viện khác ở bất kỳ đâu trên thế giới, họ chỉ cần truy xuất thông tin và kết quả chỉ số xét nghiệm của mình trên chuỗi blockchain mà cho dù hai bệnh viện (nơi khám ban đầu và nơi chữa bệnh mới) không cùng ngôn ngữ hay sử dụng phần mềm khác nhau.

Việc này giúp người bệnh giảm thiểu chi phí xét nghiệm lại khi đến các bệnh viện mới cũng như góp phần giúp nơi tiếp nhận bệnh nhân mới có thể truy xuất tiền

sử bệnh tật, phác đồ điều trị hay các phản ứng phụ đối với các loại thành tố thuốc trước đây của bệnh nhân. Để từ đó giúp chuẩn đoán và đưa ra liệu trình điều trị phù hợp, mang lại hiệu quả cao cho người bệnh.

Giáo dục

Việc thẩm định bằng cấp, chứng chỉ là một vấn đề phức tạp và chưa bao giờ hết mới đối với nhiều nước trên thế giới. Khi tìm kiếm trên google, chúng ta có thể dễ dàng nhận thấy việc mua bán bằng cấp, chứng chỉ giả ở nhiều website trên thế giới.

Việc quản lý các chứng chỉ, bằng cấp của các trường đại học nói chung hay các cơ sở đào tạo nghề nói riêng nếu được áp dụng công nghệ Blockchain sẽ góp phần minh bạch hóa hồ sơ học viên cũng như giúp các nhà tuyển dụng dễ dàng truy xuất nguồn gốc cơ sở đào tạo hay quá trình học tập của các ứng viên từ thấp đến cao.

Tài chính

Blockchain được xem như là một cách để cắt giảm chi phí và thời gian thanh toán bù trừ giao dịch liên ngân hàng, cũng như tạo ra hệ thống an toàn hơn, bảo mật cao với chi phí thấp. Tại thời điểm này, nhiều tổ chức tài chính đang có cuộc chiến tranh giành nhau nhằm hình thành các liên minh mới để thương mại hóa công nghệ blockchain.

Thương mại điện tử

Những thách thức lớn đó của thương mại điện tử có thể được xử lý bằng các hợp đồng thông minh khi ứng dụng công nghệ Blockchain. “Với công nghệ Blockchain, những người tham gia yên tâm khi ký các hợp đồng thông minh và đưa giải pháp thanh toán vào website. Từ đây tất cả mọi người có thể dễ dàng kinh doanh và hợp tác với nhiều doanh nghiệp trong và ngoài nước trong một thời gian ngắn hơn và thủ tục đơn giản hơn nhiều lần.

3.2 AngularJS

3.2.1 Giới thiệu AngularJS

AngularJS là một dự án mã nguồn mở được phát triển đầu tiên bởi Misko Hevery một nhân viên của Google. Hevery bắt đầu nghiên cứu và phát triển dự án vào năm 2009 và phiên bản 1.0 được cho ra mắt vào năm 2012. Do sự hữu ích của dự án này nên Google quyết định là công ty chính thức đứng đằng sau hỗ trợ sự phát triển của AngularJS. Phiên bản hiện tại của AngularJS vào thời điểm này là 1.6.6.

AngularJS là một Framework được viết bằng Javascript. AngularJS không chỉ đơn giản là một thư viện mà nó còn là một framework. Khác với các thư viện Javascript như jQuery, AngularJS đưa ra hướng dẫn cụ thể hơn cách cấu trúc mã lệnh HTML và Javascript. AngularJS là một Framework được viết bằng Javascript. AngularJS không chỉ đơn giản là một thư viện mà nó còn là một framework. Khác với các thư viện Javascript như jQuery, AngularJS đưa ra hướng dẫn cụ thể hơn cách cấu trúc mã lệnh HTML và Javascript. Trong AngularJS có các chỉ dẫn: ng-app, ng-model, ng-name. Các chỉ dẫn này có nghĩa:

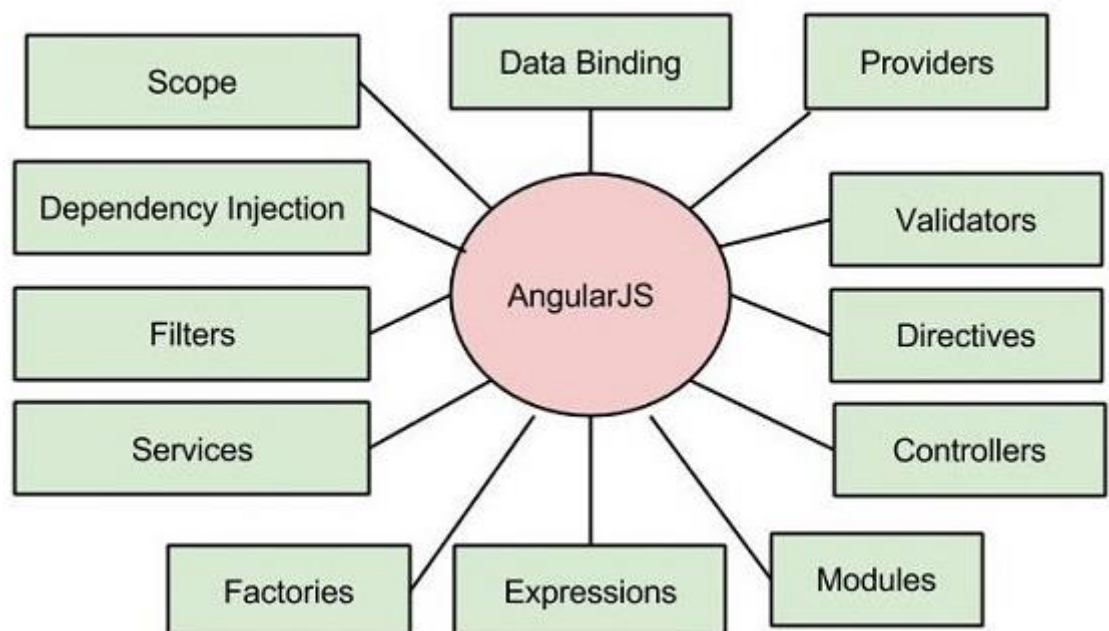
- Chỉ dẫn ng-app="" được dùng để giúp AngularJS đánh dấu ứng dụng sẽ được bắt đầu từ bằng nào trong HTML.
- Chỉ dẫn ng-model="name" được dùng để kết nối giá trị của trường <input> với giá trị của biến name trong ứng dụng AngularJS
- Chỉ dẫn ng-bind="name" dùng để gắn giá trị của biến name ở trên trở thành giá trị innerHTML của thẻ

3.2.2 Các thành phần của AngularJS

AngularJS gồm những thành phần chính sau:

- Data-binding: (liên kết dữ liệu) tự động đồng bộ dữ liệu giữa model và view.

- Module được sử dụng để phân biệt phần xử lý logic (service), controller và application ... và giúp code trở nên rõ ràng.
- Scope: (Phạm vi) Đây là những đối tượng kết nối giữa Controller và View.
- Controller: Đây là những hàm javascript xử lý kết hợp với bộ điều khiển Scope.
- Service: Như tôi đề cập ở trên, AngularJS sử dụng các API được xây dựng từ các web service (PHP, ASP) để thao tác với DB.
- Filters: Bộ lọc lọc ra các thành phần của một mảng và trả về mảng mới. *
Directives: đánh dấu vào các yếu tố của DOM, nghĩa là sẽ tạo ra các thẻ HTML tùy chỉnh.
- Templates: hiển thị thông tin từ controller, đây là một thành phần của views.
- Routing: chuyển đổi giữa các action trong controller.
- MVC: Mô hình chia thành phần riêng biệt thành Model, View, Controller.
- Deep Linking: Liên kết sâu, cho phép bạn mã hóa trạng thái của ứng dụng trong các URL để nó có thể đánh dấu được với công cụ tìm kiếm.
- Dependency Injection: Angular giúp các nhà phát triển tạo ứng dụng dễ dàng hơn để phát triển, hiểu và thử nghiệm dễ dàng.



Hình 10: Hình ảnh mô tả các thành phần cơ bản trong AngularJS

3.2.3 Ưu điểm và nhược điểm của AngularJS

Ưu điểm:

- Cung cấp khả năng tạo ra các Single Page Application dễ dàng.
- Cung cấp khả năng data binding tới HTML, khiến cho người dùng cảm giác linh hoạt, thân thiện.
- Dễ dàng để thực hiện việc kiểm tra và phát hiện page
- Dễ dàng tái sử dụng component
- Giúp lập trình viên viết code ít hơn với nhiều chức năng hơn.
- Chạy được trên các loại trình duyệt, trên cả PC lẫn mobile.

Nhược điểm:

- Không an toàn : được phát triển từ javascript cho nên ứng dụng được viết bởi AngularJS không an toàn. Nên có sự bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
- Nếu người sử dụng ứng dụng của vô hiệu hóa JavaScript thì sẽ chỉ nhìn thấy trang cơ bản

3.3 NodeJS

3.3.1 Giới thiệu NodeJS

NodeJS là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine), được phát triển bởi Ryan Dahl năm 2009 và phiên bản cuối cùng là v0.10.36. Nó là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường NodeJS trên hệ điều hành Window, Linux... Ngoài ra, nó cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng NodeJS với các phần mở rộng. Nó có một số đặc điểm

- Không đồng bộ và Phát sinh sự kiện (Event Driven): Tất cả các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dữ liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về Sự kiện của NodeJS giúp Server nhận đưa phản hồi từ các API gọi trước đó.
- Chạy rất nhanh: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.
- Các tiến trình đơn giản nhưng hiệu năng cao: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Nodejs sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.
- Không đệm: Ứng dụng Node.js không lưu trữ các dữ liệu buffer.
- Có giấy phép: Node.js được phát hành dựa vào MIT License.

3.3.2 Ưu điểm và nhược điểm của NodeJS

Ưu điểm của NodeJS:

- JSON APIs: Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Web làm bằng JSON.
- Ứng dụng trên 1 trang: Nếu chúng ta viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.
- Shelling tools unix: NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu suất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

- Streaming Data (Luồng dữ liệu): Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả sử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
- Ứng dụng Web thực.

Nhược điểm của NodeJS

- Ứng dụng nặng tốn tài nguyên: Nếu chúng ta cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này chúng ta hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa
- NodeJS và ngôn ngữ khác: NodeJS vẫn còn là một công nghệ mới và cộng đồng lập trình viên vẫn còn hạn chế.

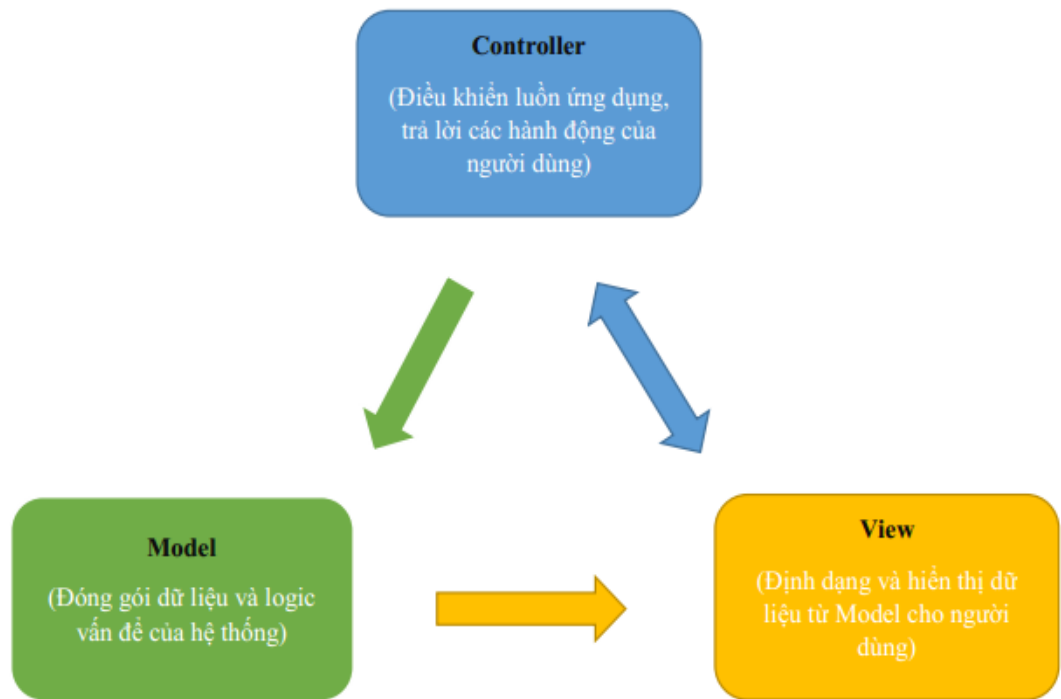
Chương 4 Phát triển và triển khai ứng dụng

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Ứng dụng sử dụng kiến trúc ba lớp Model-View-Controller, mỗi lớp có nhiệm vụ riêng độc lập với nhau. Đây là một kiến trúc đơn giản, quen thuộc với hầu hết mọi người. Trong đó nhiệm vụ của các lớp như sau:

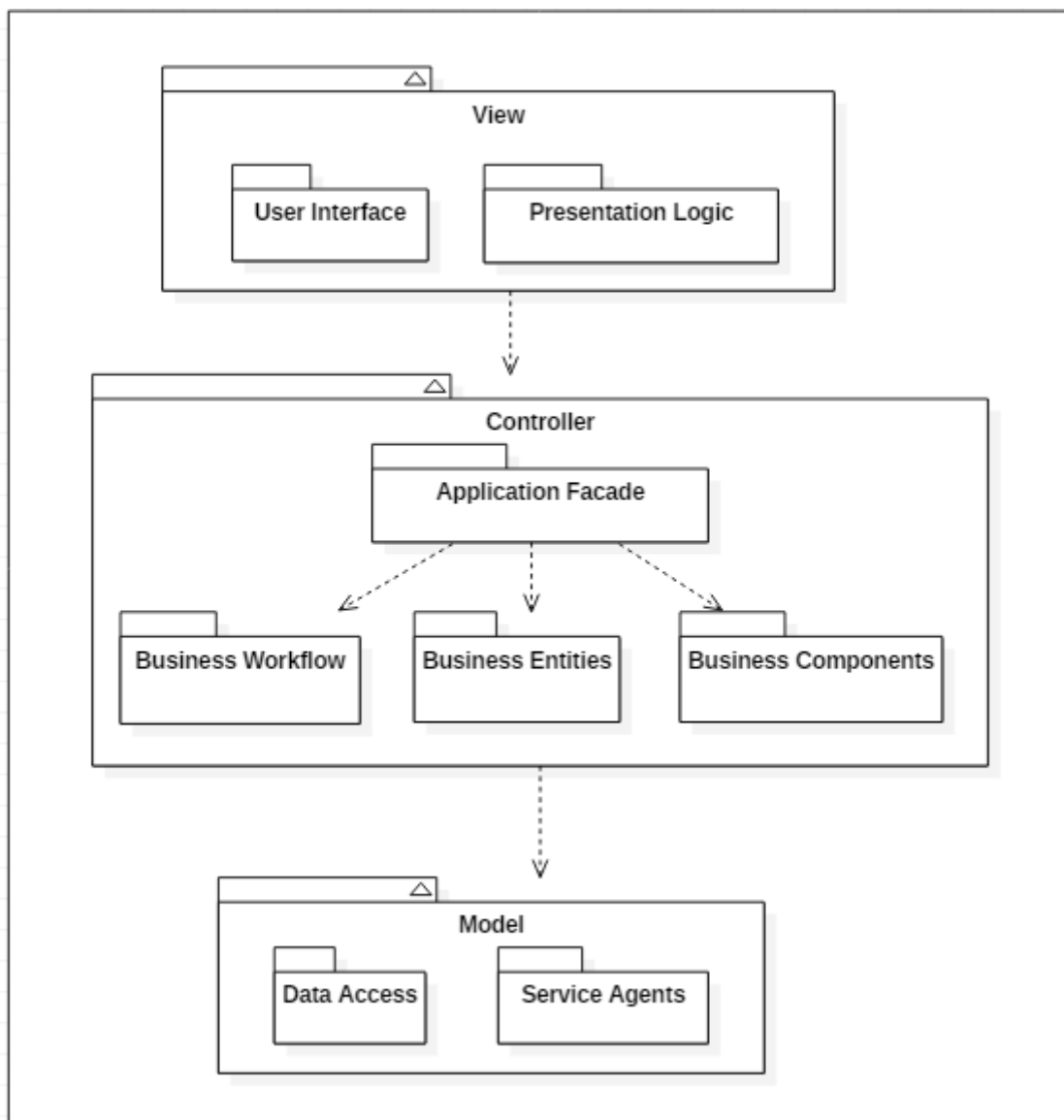
- Model: bao gồm dữ liệu, nghiệp vụ logic
- View: các thành phần giao diện người dùng (UI Components) mà người dùng tương tác, thể hiện trực quan của Model
- Controller: trung gian, điều phối viên giữa Model và View



Hình 11: Kiến trúc mô hình MVC.

Model-View-ViewModel được ra đời sau hay chuẩn thiết kế Architecture Components được Google đưa ra vào năm 2017 sau gần 9 năm hệ điều hành Android được ra mắt. Mà kiến trúc Model-View-Controller được sử dụng cho ứng dụng này là bởi vì đây là một ứng dụng được xây dựng và phát triển bởi cá nhân, không có quy mô lớn. Cũng như phần Model trong ứng dụng thường ở dạng tĩnh, ít thay đổi, không cần cập nhật qua lại giữa View-Model quá nhiều. Nên kiến trúc Model-View-Controller rất phù hợp cho ứng dụng từ điển này, mặc dù kiến trúc này ở Android không đúng chuẩn Model-View-Controller như ở các nền tảng khác.

4.1.2 Thiết kế tổng quan



Hình 12: Biểu đồ package tổng quan

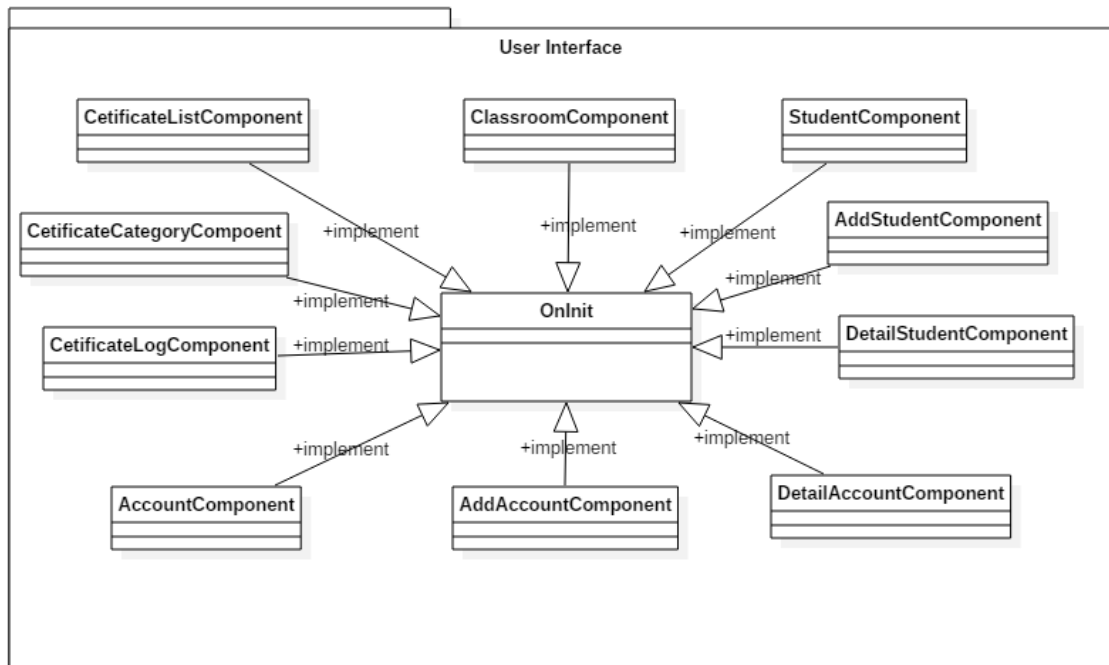
Tên gói	Nhiệm vụ/ Mục đích
User Interface	Gói package này thuộc về View, chứa các giao diện của người dùng: giao diện web client, web của phần quản trị hệ thống.
Presentation Logic	Gói package này thuộc về View, chứa các logic để xử lý các

	dữ liệu từ View gửi đến Controller và từ Controller trả về cho View.
Business Workflow	Gói package này thuộc về Controller, chứa các logic nghiệp vụ của hệ thống: các luồng hoạt động của hệ thống, xử lý các yêu cầu từ View.
Business Entities	Gói package này thuộc về Controller, chứa các class để tương tác với Model.
Business Components	Gói này thuộc về Controller, nhận các yêu cầu từ phía client và trả về kết quả cho client.
Data Access	Gói package này thuộc về Model, chứa các thành phần để có thể tương tác được với cơ sở dữ liệu.
Service Agents	Gói package này thuộc về Model, nhận các yêu cầu từ phía Controller và xử lý các yêu cầu đó, sau đó trả về kết quả cho Controller khi đã thực hiện xong.

Bảng 6: Bảng mô tả chi tiết các gói package

4.1.3 Thiết kế chi tiết gói

Gói User Interface



Hình 13: Hình ảnh thiết kế chi tiết gói User Interface.

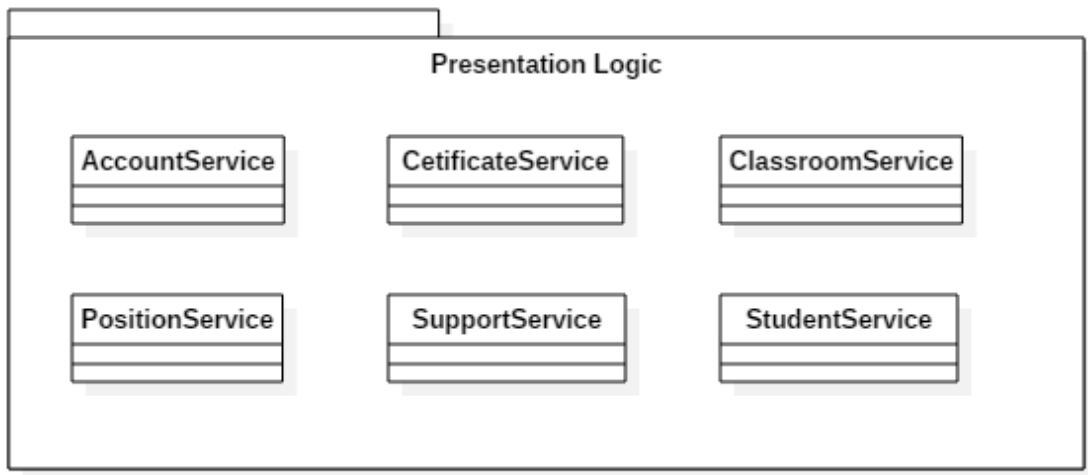
Các class trong gói đều được mở rộng từ class **OnInit**, class mở rộng từ class này đều phải định nghĩa lại hàm **ngInit**.

- **AccountComponent**: Class này chứa giao diện của chức năng quản lý các tài khoản có trong hệ thống, hiển thị các danh sách tài khoản có trong hệ thống.
- **AddAccountComponent**: Class này chứa giao diện của chức thêm tài khoản của hệ thống.
- **DetailAccountComponent**: Class này chứa giao diện của chức năng xem chi tiết một tài khoản có trong hệ thống.
- **CetificateLogComponent**: Class này chứa giao diện của chức năng hiển thị lịch sử cập nhật chứng chỉ trong hệ thống.
- **CetificateCategoryCompoent**: Class này chứa giao diện của chức năng quản lý danh mục chứng chỉ có trong hệ thống.
- **CetificateListComponent**: Class này chứa giao diện của chức năng hiển thị

các chứng chỉ có trong hệ thống.

- ClassroomComponent: Class này chứa giao diện quản lý các lớp học có trong hệ thống.
- StudentComponent: Class này chứa giao diện quản lý sinh viên: thêm, cập nhật, xóa sinh viên có trong hệ thống.
- AddStudentComponent: Class này chứa giao diện của chức năng thêm sinh viên trong hệ thống.
- DetailStudentComponent: Class này chứa giao diện của chức năng hiển thị chi tiết thông tin của một sinh viên có trong hệ thống.

Gói Presentation Logic

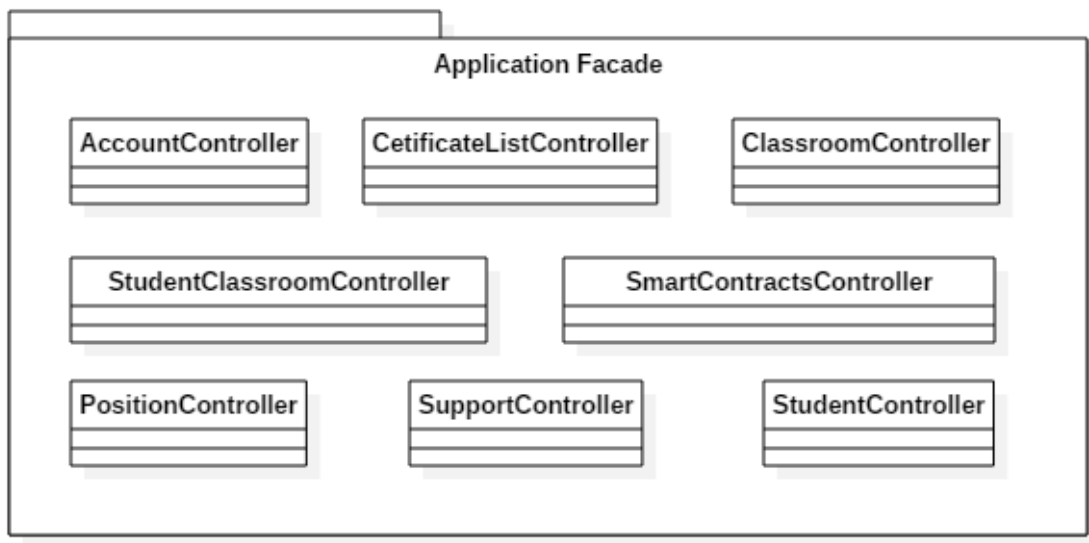


Hình 14: Hình ảnh thiết kế chi tiết gói Presentation Logic

Các class trong gói Presentation Logic gồm:

- AccountService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý tài khoản.
- CetificateService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý chứng chỉ.
- ClassroomService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý lớp học.
- PositionService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý chức vụ của các tài khoản có trong hệ thống.
- SupportService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng thông tin về hệ thống.
- StudentService: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý sinh viên có trong hệ thống.

Gói Application Facade

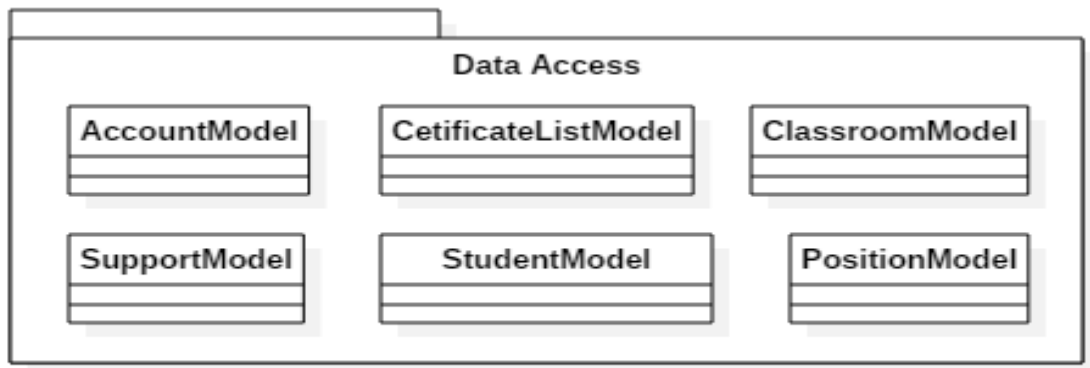


Hình 15: Hình ảnh thiết kế chi tiết gói Application Facade

Các class có trong gói Application Facade gồm:

- AccountController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý tài khoản của hệ thống.
- CetificateController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý chức chứng chỉ của hệ thống.
- ClassroomController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý lớp học của hệ thống.
- StudentClassroomController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý sinh viên của lớp học trong hệ thống.
- SmartContractController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý chứng chỉ của hệ thống.
- PositionController: Class này sẽ xử lý các yêu cầu nghiệp vụ liên quan đến chức năng quản lý chức vụ của các tài khoản có trong hệ thống.
- SupportController: Class này sẽ xử lý các yêu cầu liên quan thông tin về hệ thống: trả về thông tin của hệ thống, thông tin người hỗ trợ,..
- StudentController: Class này sẽ xử lý các yêu cầu liên quan đến chức năng quản lý sinh viên có trong hệ thống.

Gói Data Access



Hình 16: Hình ảnh thiết kế chi tiết gói Data Access

Các class trong gói Data Access bao gồm:

- AccountModel: Class này sẽ cung cấp các hàm thuộc về chức năng quản lý tài khoản để controller có thể tương tác với hệ thống.
- CetificateListModel: Class này sẽ cung cấp các hàm thuộc về chức năng quản lý chứng chỉ để controller có thể tương tác với hệ thống.
- ClassroomModel: Class này sẽ cung cấp các hàm thuộc về chức năng quản lý lớp học để controller có thể tương tác được với hệ thống.
- SupportModel: Class này sẽ cung cấp các hàm thuộc về chức năng quản lý thông của hệ thống
- StudentModel: Class này sẽ cung cấp các hàm thuộc về chức năng quản lý sinh viên để controller có thể tương tác được với hệ thống.
- PositionModel: Class này sẽ cùng cấp các hàm thuộc về chức năng quản lý chức vụ để controller có thể tương tác được với hệ thống

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Ứng dụng hỗ trợ chạy trên các thiết bị desktop, bao gồm tất cả các độ phân giải và kích thước màn hình hình hiện nay. Thiết bị desktop được chia làm 2 loại:

máy tính để bàn và máy tính xách tay. Dưới đây là bảng liệt kê danh sách các màn hình.

Thiết bị	Kích thước màn hình	Độ phân giải
Máy tính để bàn	Màn hình LCD tỷ lệ chuẩn 19 inch	1280 × 1024
	Màn hình LCD tỷ lệ chuẩn 20 inch	1600 × 1200
	Màn hình LCD rộng 20 và 22 inch	1680 × 1050
	Màn hình LCD rộng 24 inch	1920 × 1200
Máy tính xách tay	Màn hình máy tính xách tay tỷ lệ chuẩn 13 đến 15 inch	1400 × 1050
	Màn hình máy tính xách tay rộng 13 đến 15 inch	1280 × 800
	Màn hình máy tính xách tay rộng 17 inch	1680 × 1050

Bảng 7: Mô tả danh sách các độ phân giải của màn hình

Giao diện của ứng dụng chạy tốt nhất trên màn hình máy tính để bàn có kích thước “Màn hình LCD rộng 24 inch” và độ phân giải “1920x1200”, trên thiết bị máy tính xách tay là “Màn hình máy tính xách tay rộng 17 inch” và độ phân giải “1680x1050”. Dưới đây là một số hình ảnh minh họa của hệ thống.

Mozilla

← → ↻

Hệ Thống Quản Lý Chứng Chỉ

Lịch sử ▼

Liên Hệ Với Quản Trị Viên

☐ Nguyễn Văn A
Quản trị viên

☐ Nguyễn Văn A
Quản trị viên

☐ Nguyễn Văn A
Quản trị viên

☐ Nguyễn Văn A
Quản trị viên

[Bản quyền 2018 bởi Nguyễn Văn Hải, Đại Học Bách Khoa Hà Nội](#)

Hình 17 : Hình ảnh giao diện của web client.

Mozilla

← → ↻

Hệ thống quản lý chứng chỉ

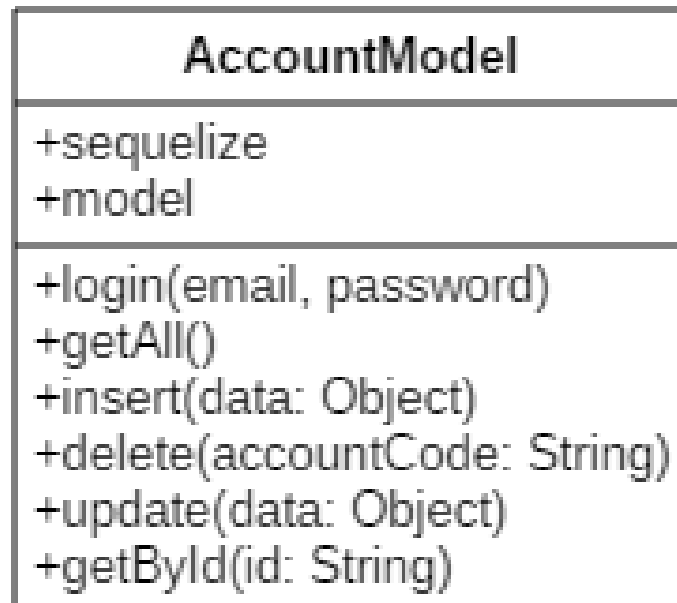
[Quản lý tài khoản](#)
[Quản lý lớp học](#)
[Quản lý sinh viên](#)
[Quản lý chứng chỉ](#)
[Thông tin hệ thống](#)

▼ ID	▼ Name	▼ Trạng thái
1	Nguyen Van A	Hoạt động
2	Nguyen Van B	Hoạt động
3	Nguyen Van C	Ngừng hoạt động
4	Nguyen Van D	Ngừng hoạt động

Hình 18: Hình ảnh mô tả web admin

4.2.2 Thiết kế lớp

Dưới đây là thiết kế chi tiết của một số class, các lớp này sẽ cung cấp các hàm để các lớp controller có thể tương tác được với cơ sở dữ liệu. Các bảng trong cơ sở dữ liệu sẽ tương ứng với các class model và trong các class này sẽ cung cấp các hàm: thêm, cập nhật, xóa, lấy thông tin,...



Hình 19: Thiết kế chi tiết của lớp AccountModel

Thuộc tính/Phương thức	Mô tả
sequelize	Định nghĩa kết nối đến cơ sở dữ liệu.
model	Định nghĩa cấu trúc bảng.
login(email:String,password:String)	Phương thức này dùng cho chức năng đăng nhập.
getAll()	Phương thức này trả về tất cả danh sách tài khoản có trong hệ thống.
insert(data:Object)	Phương thức này dùng để thêm tài khoản

	vào trong hệ thống.
delete(accountCode:String)	Phương thức này dùng để xóa tài khoản khỏi hệ thống.
update(in data:Object)	Phương thức này dùng để cập nhật thông tin tài khoản có trong hệ thống.
getById(in id:String)	Phương thức này dùng để lấy thông tin của một tài khoản theo id của tài khoản.

Bảng 8: Mô tả chi tiết các thuộc tính và phương thức của AccountModel.

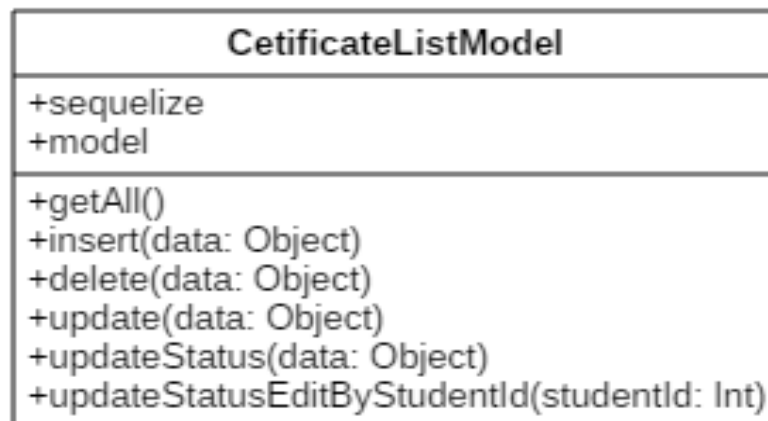
CetificateCategoryModel
+sequelize +model
+getAll() +insert(data: Object) +delete(id: Int) +update(data: Object) +getById(id: Int)

Hình 20: Thiết chi tiết của lớp CetificateCategoryModel.

Thuộc tính/Phương thức	Mô tả
sequelize	Định nghĩa kết nối đến cơ sở dữ liệu
model	Định nghĩa cấu trúc bảng.
getAll()	Phương thức này sẽ trả về tất cả các loại chứng chỉ có trong hệ thống.
insert(data: Object)	Phương thức này sẽ dùng để thêm loại chứng chỉ vào hệ thống.
delete(id: Int)	Phương thức này sẽ dùng để xóa loại chứng chỉ khỏi hệ thống.
update(data: Object)	Phương thức này sẽ dùng để cập nhật loại chứng chỉ của hệ thống.
getById(id: Int)	Phương thức này sẽ trả về thông tin chi tiết của một chứng

	chi.
--	------

Bảng 9: Đặc tả chi tiết thuộc tính và phương thức của lớp CetificateCategoryModel



Hình 21 : Hình ảnh thiết kế chi tiết của lớp CetificateListModel.

Thuộc tính/Phương thức	Mô tả
sequelize	Định nghĩa kết nối đến cơ sở dữ liệu
model	Định nghĩa cấu trúc bảng
getAll()	Phương thức này trả về tất cả các loại chứng chỉ có trong hệ thống.
insert(data: Object)	Phương thức này dùng để thêm dữ liệu chứng chỉ vào hệ thống.
delete(data: Object)	Phương thức này dùng để xóa chứng chỉ khỏi hệ thống.
update(data: Object)	Phương thức này dùng để cập nhật chứng chỉ có trong hệ thống.
updateStatus(data: Object)	Phương thức này dùng để cập nhật trạng thái của chứng chỉ.
updateStatusEditByStudentId(studentId: Int)	Phương thức này dùng để cập nhật trạng thái cập nhật của chứng chỉ.

Bảng 10 : Đặc tả chi tiết của lớp CetificateListModel.

ClassroomModel
+sequelize +model
+insert(data: Object) +update(data: Object) +updateStatusById(data: Object) +getAll() +getClassroomActive() +getStudentOfClassroomById(classroomId: Int)

Hình 22 :Hình ảnh đặc tả chi tiết của lớp ClassroomModel

Phương thức/Thuộc tính	Mô tả
sequelize	Định nghĩa kết nối đến cơ sở dữ liệu
model	Định nghĩa cấu trúc bảng.
insert(data: Object)	Phương thức này dùng để thêm lớp học vào trong hệ thống.
update(data: Object)	Phương thức này dùng để cập nhật dữ liệu của lớp học vào hệ thống.
updateStatusById(data: Object)	Phương thức này dùng để cập nhật trạng thái theo id của lớp học.
getAll()	Phương thức này trả về tất cả các lớp học có trong hệ thống.
getClassroomActive()	Phương thức này trả về tất cả các lớp học đang hoạt động trong hệ thống.
getStudentOfClassroomById(classroomId: Int)	Phương thức này trả về danh sách sinh viên của một lớp học.

Bảng 11 :Đặc tả chi tiết của lớp ClassroomModel.

StudentModel
+sequelize +model
+insert(data: Object) +update(data: Object) +updateStatusById(data: Object) +getAllStudent() +getStudentById(id: Int) +getStudentAvailable() +getStudentBySign(sign: String) +isEditHash(id: Int, name: String, dateOfBirth: String)

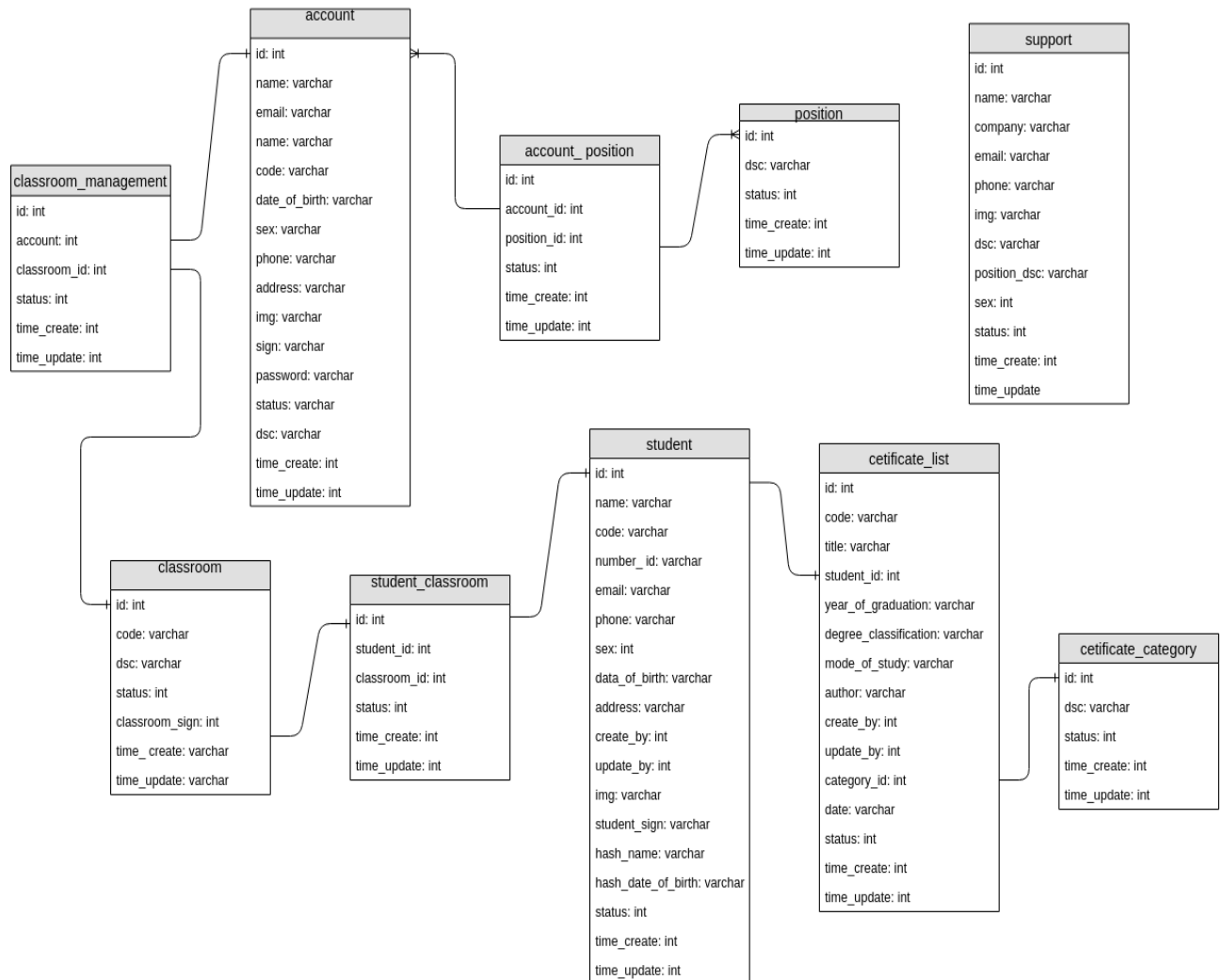
Hình 23 : Thiết kế chi tiết của lớp StudentModel.

Thuộc tính/Phương thức	Mô tả
sequelize	Định nghĩa kết nối đến cơ sở dữ liệu
model	Định nghĩa cấu trúc bảng
insert(data: Object)	Phương thức này dùng để thêm sinh viên vào trong hệ thống.
update(data: Object)	Phương thức này dùng để cập nhật dữ liệu của sinh viên có trong hệ thống.
updateStatusById(data: Object)	Phương thức này dùng để cập nhật trạng thái theo id của sinh viên.
getAllStudent()	Phương thức này sẽ trả về toàn bộ dữ liệu sinh viên có trong hệ thống.
getStudentById(id: Int)	Phương thức này sẽ trả về thông tin của sinh viên theo id của sinh viên.
getStudentAvailable()	Phương thức này sẽ trả về danh sách sinh viên chưa thuộc vào lớp học nào.
getStudentBySign(sign:String)	Phương thức này sẽ trả về thông tin của sinh viên theo chữ ký của sinh viên.
isEditHash(id: Int, name:String, dateOfBirth:String)	Phương thức này sẽ kiểm tra xem dữ liệu của một sinh viên có cập nhật hay không.

Bảng 12 : Đặc tả chi tiết của class StudentModel.

4.2.3 Thiết kế cơ sở dữ liệu

Ứng dụng sử dụng cơ sở dữ liệu MySQL để lưu trữ dữ liệu của hệ thống, trong đó có các bảng account, class_management, account_position, position, support, student, classroom, student_classroom, cetificate_list, cetificate_category.



Hình 24 : Hình ảnh thực thể liên kết cơ sở dữ liệu.

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR	Tên của tài khoản
email	VARCHAR	Địa chỉ email
code	VARCHAR	Mã tài khoản
date_of_birth	VARCHAR	Ngày sinh của tài khoản
sex	INT	Giới tính
phone	VARCHAR	Số điện thoại
address	VARCHAR	Địa chỉ của tài khoản
img	TEXT	Hình ảnh đại diện của tài khoản
sign	TEXT	Chữ ký đại diện cho tài khoản
password	VARCHAR	Mật khẩu
status	INT	Trạng thái
dsc	TEXT	Mô tả
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 13 : Bảng cơ sở dữ liệu account

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
dsc	VARCHAR	Mô tả
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 14 : Bảng cơ sở dữ liệu position

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
account_id	INT	Khóa ngoại của bảng account
position_id	INT	Khóa ngoại của bảng position
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 15: Bảng cơ sở dữ liệu account_position

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
code	VARCHAR	Mã lớp học
dsc	VARCHAR	Mô tả
status	INT	Trạng thái
classroom_sign	VARCHAR	Chữ ký đại diện cho lớp học
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 16: Bảng cơ sở dữ liệu classroom

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR	Tên sinh viên
code	VARCHAR	Mã sinh viên

number_id	VARCHAR	Số chứng minh thư
email	VARCHAR	Địa chỉ email
phone	VARCHAR	Số điện thoại
sex	INT	Giới tính
date_of_birth	VARCHAR	Ngày sinh
address	VARCHAR	Địa chỉ
create_by	INT	ID người tạo
update_by	INT	ID người cập nhật
img	TEXT	Hình ảnh đại diện
student_sign	VARCHAR	Chữ ký để đại diện cho sinh viên
hash_name	VARCHAR	Mã hóa tên
hash_date_of_birth	VARCHAR	Mã hóa ngày sinh
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 17: Bảng cơ dữ liệu student

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
student_id	INT	Khóa ngoại
classroom_id	INT	Khóa ngoại
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 18: Bảng cơ sở dữ liệu student_classroom

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
code	VARCHAR	Mã số chứng chỉ
title	VARCHAR	Tên chứng chỉ
student_id	INT	Khóa ngoại
year_of_graduation	VARCHAR	Năm tốt nghiệp
degree_classification	VARCHAR	Loại chứng chỉ
mode_of_study	VARCHAR	Hình thức đào tạo
author	VARCHAR	Người cấp
create_by	INT	Người tạo
update_by	INT	Người cập nhật
category_id	INT	Khóa ngoại
date	VARCHAR	Ngày cấp
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 19: Bảng cơ sở dữ liệu cetificate_list

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
dsc	VARCHAR	Mô tả
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 20: Bảng cơ sở dữ liệu cetificate_category

Tên trường	Kiểu dữ liệu	Mô tả
------------	--------------	-------

id	INT	Khóa chính
account_id	INT	Khóa ngoại
classroom_id	INT	Khóa ngoại
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 21: Bảng cơ sở dữ liệu classroom_management

Tên trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR	Tên người hỗ trợ
company	VARCHAR	Tên công ty
email	VARCHAR	Địa chỉ email
phone	VARCHAR	Số điện thoại
img	TEXT	Hình ảnh đại diện
dsc	TEXT	Mô tả
position_dsc	TEXT	Mô tả chức vụ
sex	INT	Giới tính
status	INT	Trạng thái
time_create	INT	Ngày tạo
time_update	INT	Ngày cập nhật

Bảng 22Bảng cơ sở dữ liệu support

Ngoài việc sử dụng cơ sở dữ liệu MySQL thì hệ thống sẽ ghi lại toàn bộ lịch sử cập nhật chứng chỉ lên Blockchain. Việc ghi lại để có thể truy xuất được các thông tin về chứng chỉ: Ai là người cập nhật ?, Ngày cập nhật ?, Nội dung cập nhật ?,....

Tên trường	Kiểu dữ liệu	Mô tả
code	BYTES32	Mã chứng chỉ

title	BYTES32	Tên chứng chỉ
studentName	BYTES32	Tên sinh viên
dateOfBirth	BYTES32	Ngày sinh
yearOfGraduation	BYTES32	Năm tốt nghiệp
degreeClassification	BYTES32	Loại bằng
modeOfStudy	BYTES32	Hình thức đào tạo
date	BYTES32	Ngày phát hành
author	BYTES32	Người cấp bằng
updateBy	BYTES32	Người cập nhật
status	BYTES32	Trạng thái chứng chỉ
timeUpdate	UINT	Ngày cập nhật
timeCreate	UINT	Ngày tạo

Bảng 23: Cấu trúc dữ liệu của Certificate

Tên trường	Kiểu dữ liệu	Mô tả
id	STRING	ID của người cấp
name	STRING	Tên của người cấp
sign	STRING	Chữ ký đại diện cho người cấp
status	UINT	Trạng thái

Bảng 24: Author

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Công cụ	Mục đích	Địa chỉ URL
VS Code	IDE lập trình	https://code.visualstudio.com/
Angular 6	Framework để lập trình web	https://angularjs.org/

NodeJS	Ngôn ngữ lập trình	https://nodejs.org/en/
MySQL	Cơ sở dữ liệu	https://www.mysql.com/
Blockchain	Ghi lại lịch sử cập nhật chứng chỉ	

Bảng 25: Danh sách các thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

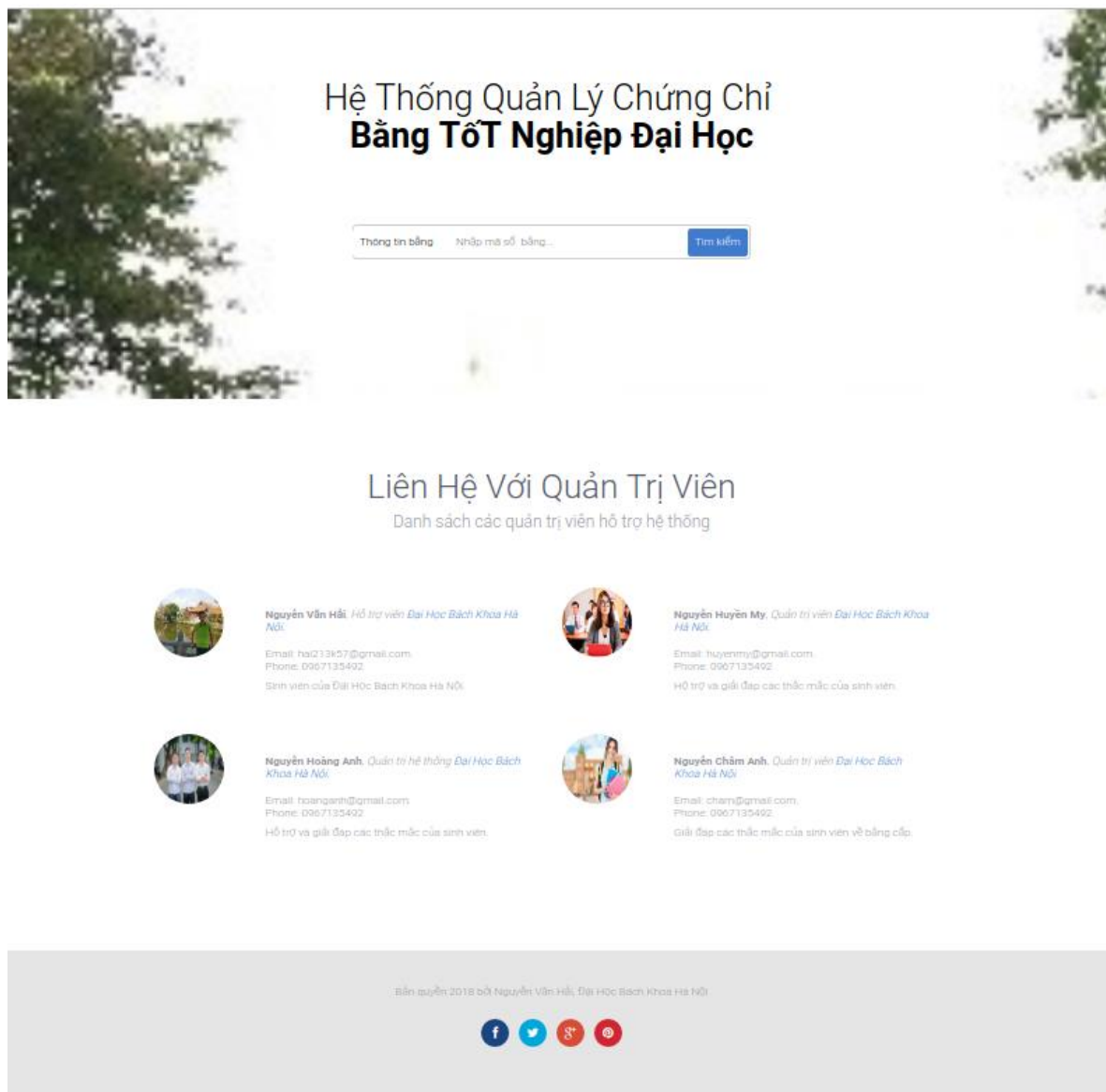
Kết quả của quá trình làm đồ án là một hệ thống hoàn chỉnh để có thể truy xuất được lịch sử cập nhật chỉ mà đã ứng dụng công nghệ Blockchain. Với các chức năng chính đã đưa ra ở mục 2.2 và đặc biệt với việc ứng dụng công nghệ Blockchain vào giải quyết bài toán thì đã trả lời được các câu hỏi: Ai là người cập nhật ?, Thời gian cập nhật là khi nào ?, Nội dung cập nhật là gì ?,....

Hệ thống được đóng gói thành các phần riêng biệt, có thể triển khai trực tiếp lên các Server chạy hệ điều hành Linux, đi cùng với các thành phần mã nguồn là một nền tảng Blockchain dùng để lưu trữ lịch sử cập nhật.

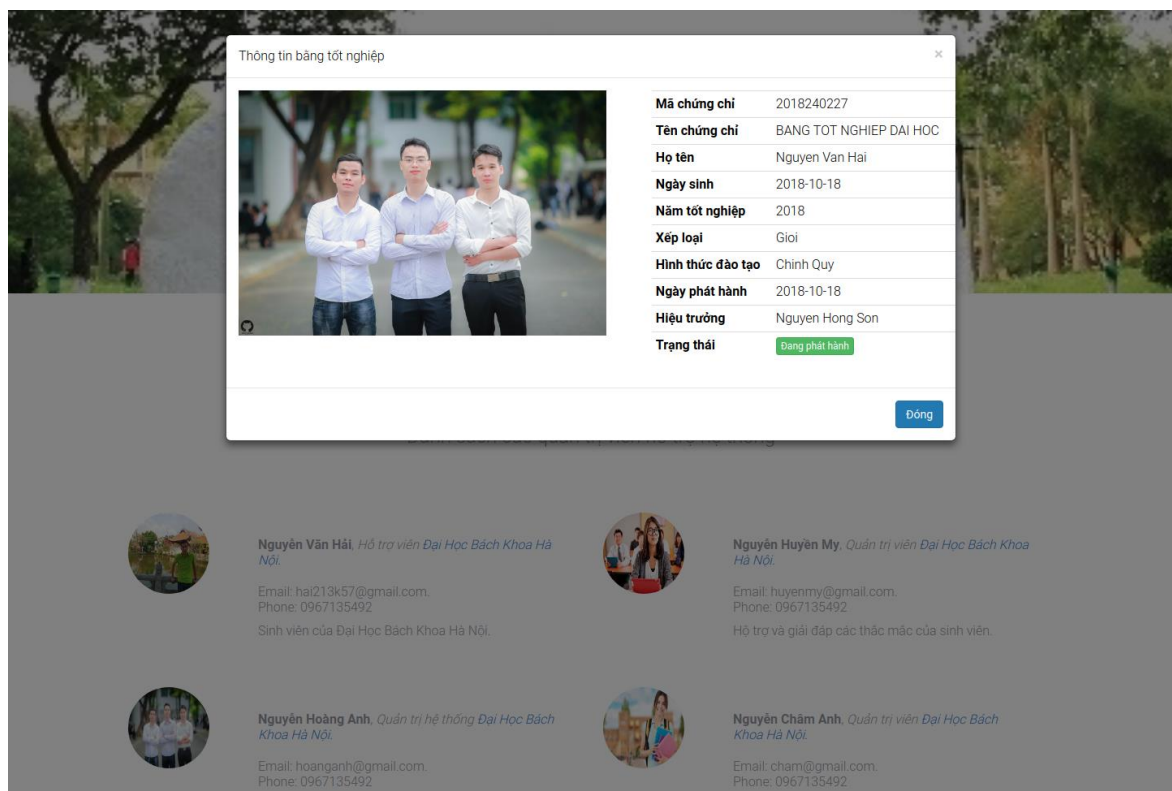
Thông tin	Số lượng	Chú thích
Số gói	14	
Số lớp	75	Gồm các Interface, Abstract Class, Smart Contract.
Số dòng code	10000	Chỉ tính số dòng code thực sự, không tính các ghi chú và dòng trống
Dung lượng mã nguồn	20MB	Mã nguồn của ứng dụng
Số file	950	Tất cả các file dùng trong hệ thống

Bảng 26: Thông tin về ứng dụng

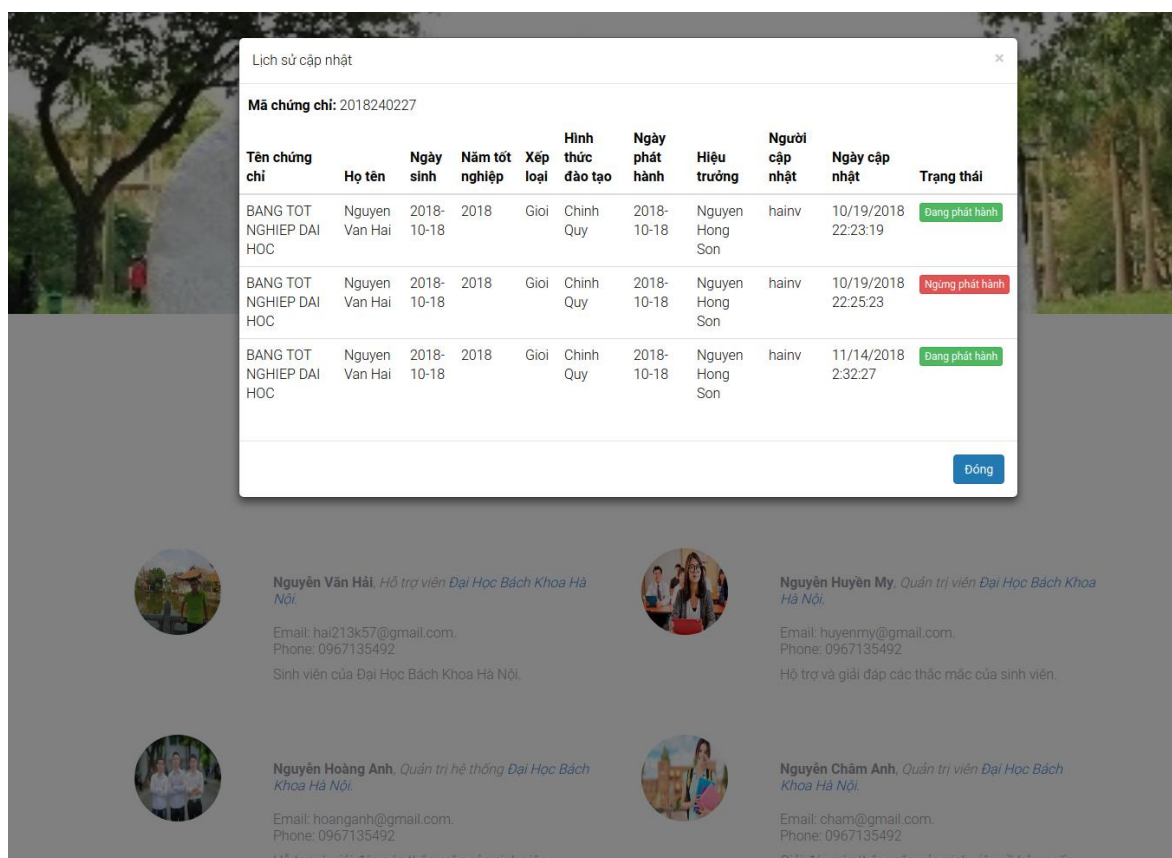
4.3.3 Minh họa các chức năng chính



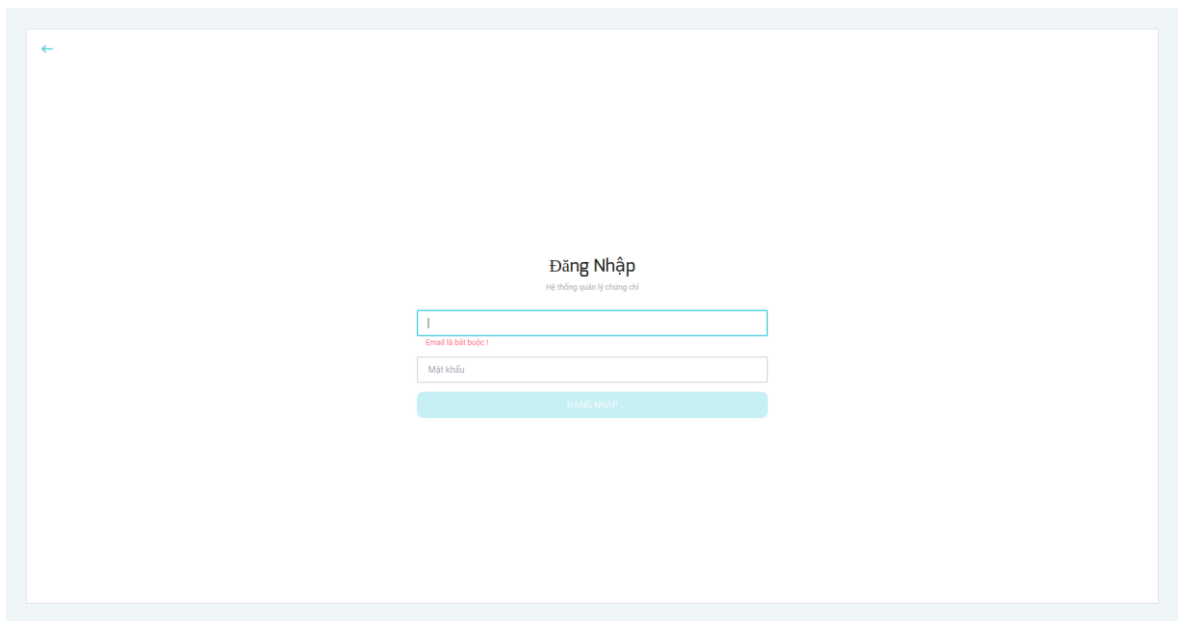
Hình 25 : Giao diện của web client



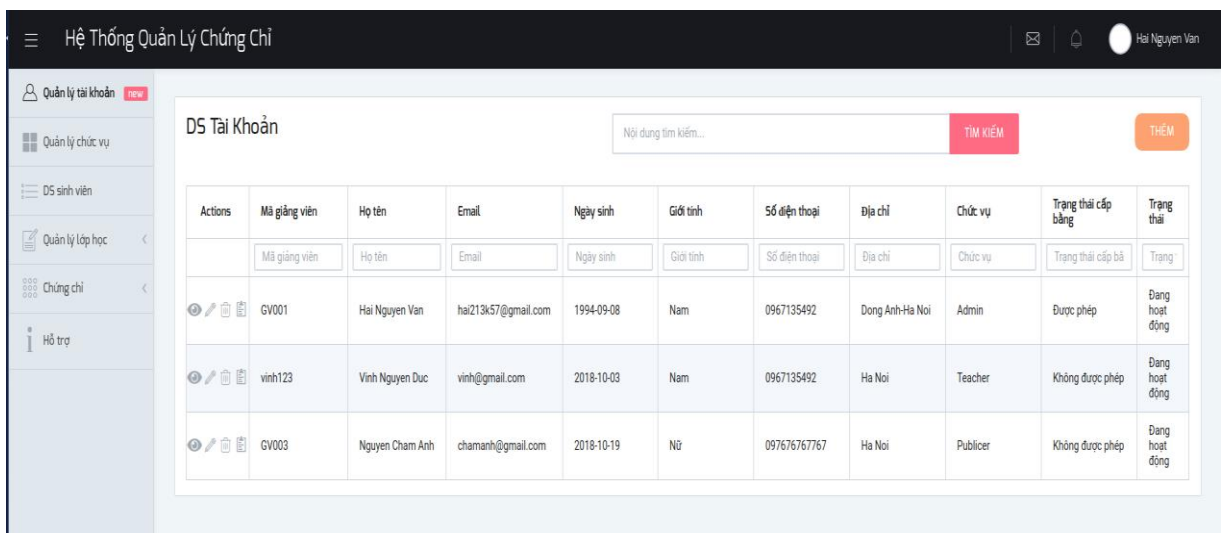
Hình 26: Hình ảnh hiển thị trạng thái chứng chỉ



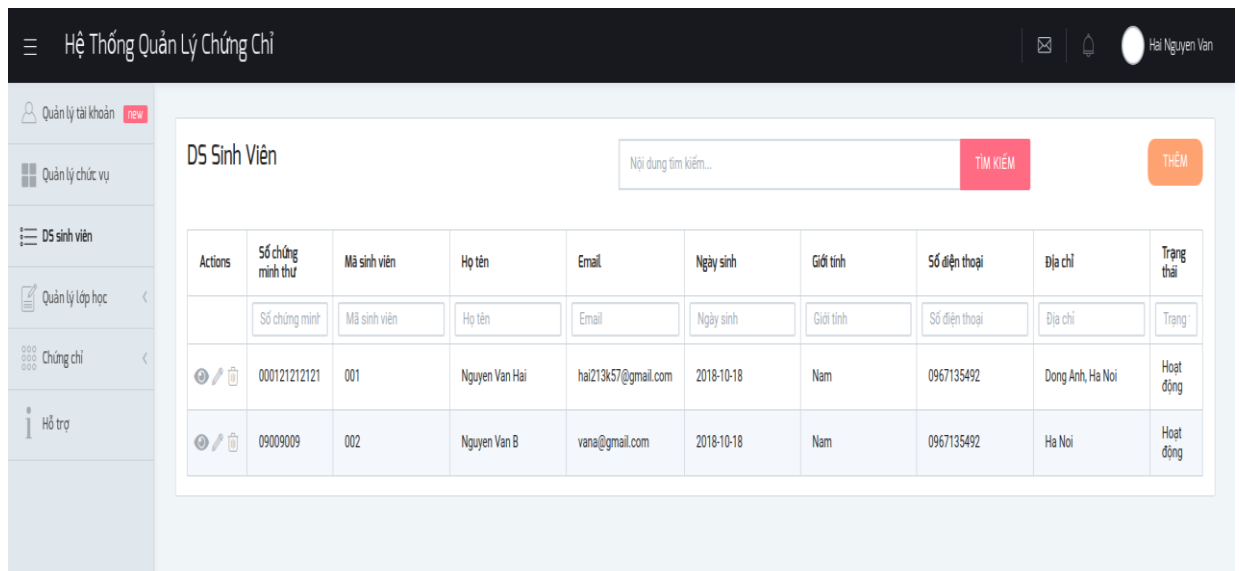
Hình 27: Hiển thị lịch sử cập nhật của chứng chỉ



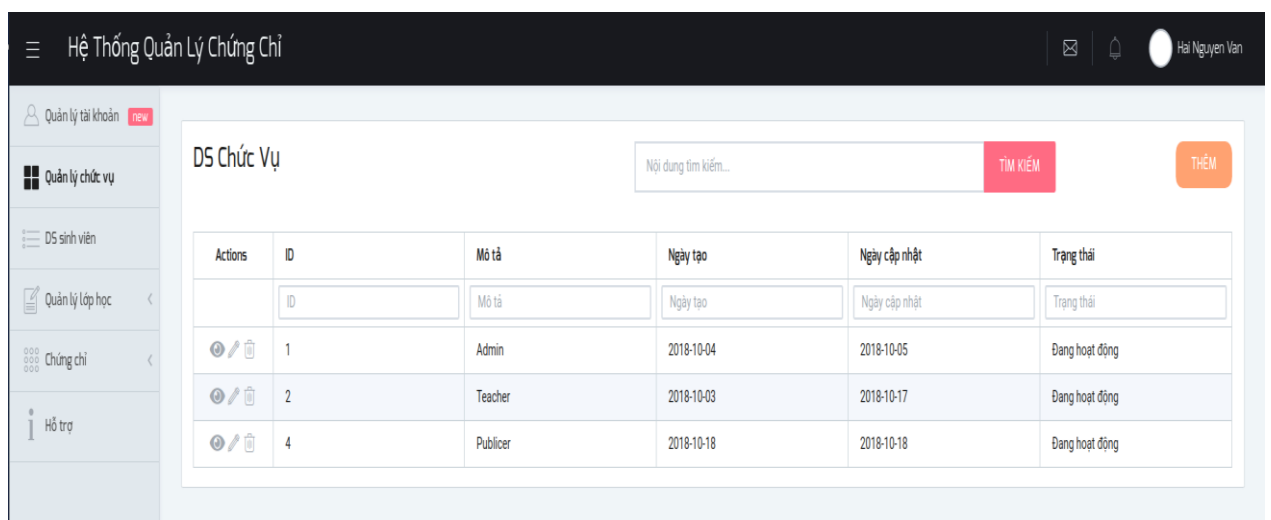
Hình 28: Hình ảnh trang đăng nhập của admin



Hình 29: Màn hình quản lý tài khoản



Hình 30: Màn hình quản lý sinh viên



Hình 31: Quản chức vụ của các tài khoản

<div> <div>Hệ Thống Quản Lý Chứng Chỉ</div> <div> <div></div> <div></div> <div>Hai Nguyen Van</div> </div> </div>												
<div> <div>Quản lý tài khoản</div> <div>Quản lý chức vụ</div> <div>Danh sách lớp học</div> <div>Quản lý đào tạo</div> <div>Chứng chỉ</div> <div>Danh mục chứng chỉ</div> <div>Danh sách chứng chỉ</div> <div>Lịch sử cập nhật</div> <div>Hỗ trợ</div> </div>	<div> <div>Lịch Sử Chứng Cập Nhật</div> <div> <div>Nội dung tìm kiếm...</div> <div>TÌM KIẾM</div> </div> </div>											
	Mã chứng chỉ	Tên chứng chỉ	Chức vụ	Tên sinh viên	Ngày sinh	Năm tốt nghiệp	Xếp loại	Hình thức đào tạo	Hiệu trưởng	Ngày phát hành	Trạng thái	Mã người cập nhật
	Mã chứng chỉ	Tên chứng chỉ	Chức vụ	Tên sinh viên	Ngày sinh	Năm tốt nghiệp	Xếp loại	Hình thức đào tạo	Hiệu trưởng	Ngày phát hành	Trạng thái	Mã người cập nhật
	2018240227	BANG TOT NGHIỆP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Đang phát hành	GV001
	2018240227	BANG TOT NGHIỆP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Ngừng hoạt động	GV001
	2018240227	BANG TOT NGHIỆP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Đang phát hành	GV001

Hình 32: Lịch sử cập nhật chứng chỉ

4.4 Kiểm thử

Ở phần này ta sẽ sử dụng kỹ thuật phân tích giá trị biên và phân vùng tương đương trong kiểm thử hộp đen, ở các chức năng tra cứu trạng thái chứng chỉ, lịch sử cập nhật chứng chỉ, cập nhật chứng chỉ, đăng nhập,...

STT	Chức năng	Điều kiện tiên quyết	Các bước thực hiện	Kết quả mong đợi	Kết quả thực tế
1	Tra cứu trạng thái chứng chỉ	Yêu cầu máy tính có kết nối mạng	1. Truy cập vào địa chỉ URL của hệ thống. 2. Chọn nhãn thông tin bằng. 3. Nhấn vào nút tìm kiếm.	Hiển thị thông tin của chứng chỉ	Pass
2	Tra cứu lịch sử cập nhật chứng chỉ		1. Truy cập vào địa chỉ URL của hệ thống. 2. Chọn nhãn lịch sử cập nhật. 3. Nhấn vào nút tìm kiếm.	Hiển thị thông tin lịch sử cập nhật của chứng chỉ	Pass
3	Cập nhật thông tin	Yêu cầu máy tính có	1. Truy cập vào địa chỉ URL của hệ thống.	Hệ thống trả về thông báo	Pass

	chứng chỉ	kết nối mạng. Tài khoản được cấp quyền để có thể cập nhật thông tin chứng chỉ	2. Đăng nhập vào hệ thống. 3. Cập nhật chứng chỉ	“Đã cập nhật chứng chỉ thành công”	
4	Đăng nhập	Yêu cầu máy tính có kết nối mạng.	1. Truy cập vào địa chỉ URL của hệ thống. 2. Nhập đầy đủ thông tin email và mật khẩu. 3. Đăng nhập vào hệ thống.	Đăng nhập được vào hệ thống	Pass
5	Cập nhật thông tin lớp học	Yêu cầu máy tính có kết nối mạng. Tài khoản được cấp quyền để có thể cập nhật thông tin lớp học.	1. Truy cập vào địa chỉ URL của hệ thống. 2. Đăng nhập vào hệ thống. 3. Cập nhật thông tin lớp học.	Hệ thống trả về thông báo “Đã cập nhật lớp học thành công”	Pass

Bảng 27: Các trường hợp kiểm thử

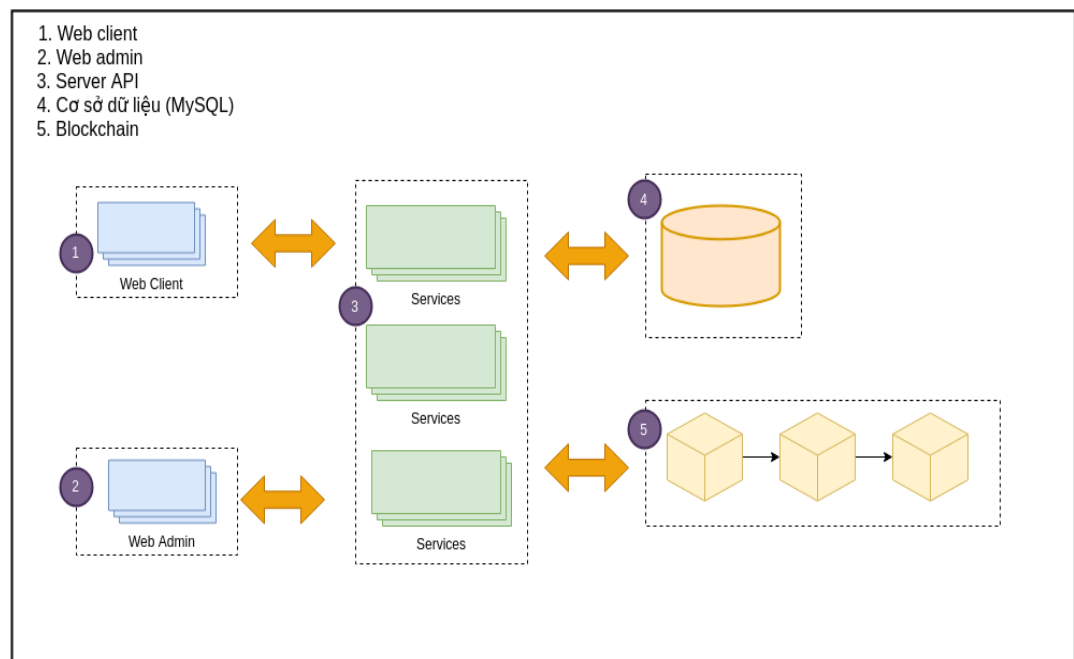
4.5 Triển khai

4.5.1 Yêu cầu hệ thống

Yêu cầu	Đặc tả
Hệ điều hành	Linux (CentOS 7 hoặc Ubuntu Server 16.04)
NodeJS	Phiên bản v8.11.3
PM2	Phiên bản mới nhất
Docker	Phiên bản 18.x.x
Angular 6	Phiên bản 6
GIT	Phiên bản mới nhất

Bảng 28: Đặc tả yêu cầu của hệ thống

4.5.2 Cách triển khai hệ thống



Hình 33 : Mô hình kiến trúc của hệ thống.

Cách triển khai thành phần web client của hệ thống, chúng ta thực hiện tuần tự các bước sau:

- `$ git clone https://github.com/hainguyenan/Do-An.git`
- `$ cd ROOT_SOURCE/source/web-client`
- `$ npm install`
- `$ npm start`

Cách triển khai thành phần web admin của hệ thống, chúng ta thực hiện tuần tự các bước sau:

- `$ git clone https://github.com/hainguyenan/Do-An.git`
- `$ cd ROOT_SOURCE/source/web-admin`
- `$ npm install`
- `$ npm start`

Cách triển khai thành phần server api của hệ thống, chúng ta thực hiện tuần tự các bước sau:

- `$ git clone https://github.com/hainguyenan/Do-An.git`
- `$ cd ROOT_SOURCE/source/api`
- `$ npm install`
- `$ npm start`

Cách triển khai thành phần Blockchain của hệ thống, chúng ta thực hiện tuần tự các bước sau:

- \$ git clone https://github.com/hainguyenvan/Do-An.git
- \$ cd ROOT_SOURCE/source/ganache-cli
- \$ docker-compose up -d

Chương 5 Các giải pháp và đóng góp nổi bật

5.1 Phát triển web client

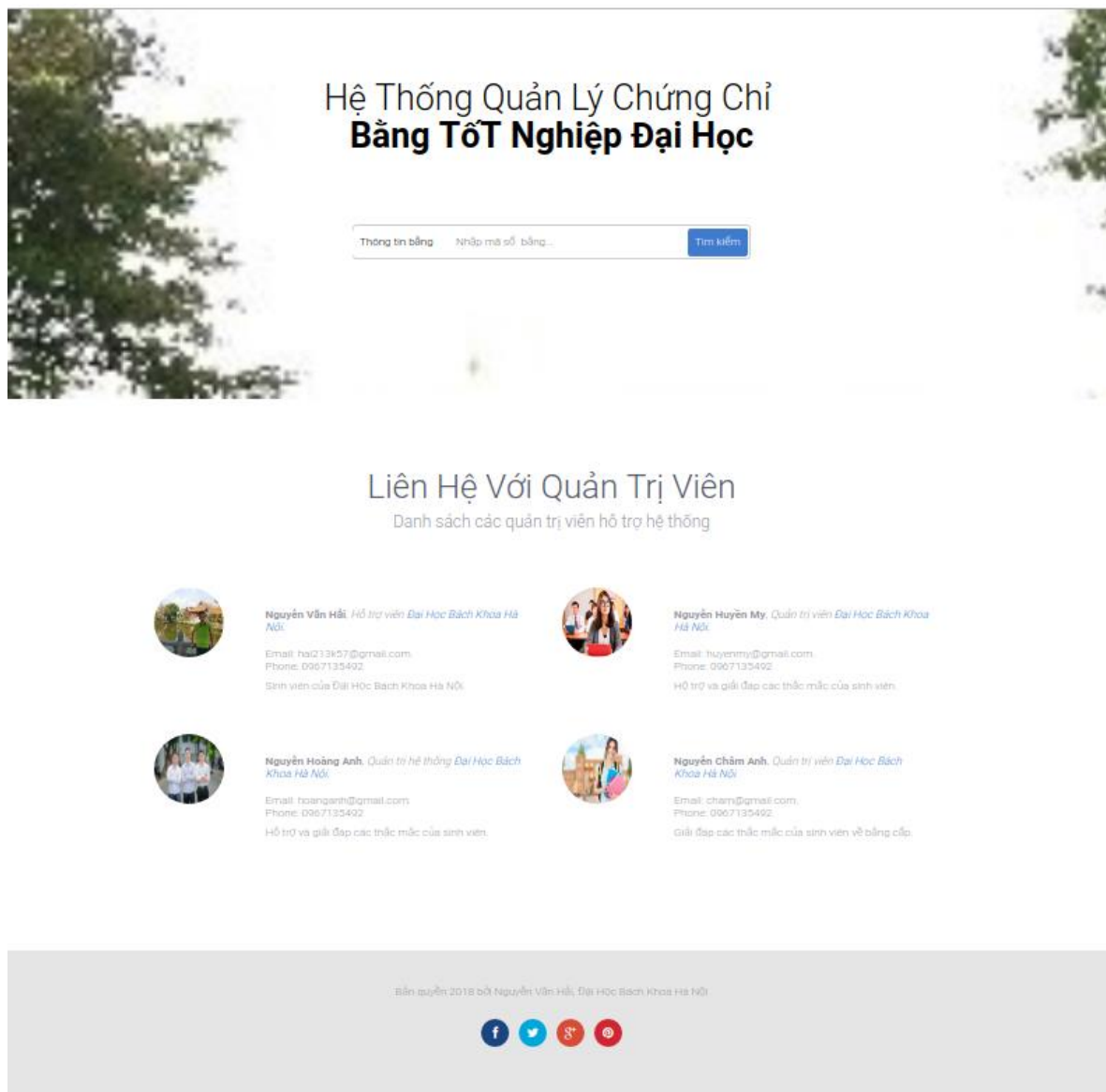
5.1.1 Đặt vấn đề

Để người dùng có thể xem được thông tin trạng thái hiện tại của chứng chỉ, lịch sử cập nhật chứng chỉ thì việc xây dựng một web site để hiển thị thông tin là khách quan nhất. Với web site thì người dùng chỉ việc truy cập vào địa chỉ URL của hệ thống và sau đó nhập mã chứng chỉ, chọn kiểu truy vấn: Thông tin chứng chỉ, Lịch sử cập nhật chứng chỉ thì hệ thống sẽ trả về thông tin và hiển thị nó lên giao diện web.

5.1.2 Giải pháp


Với web client thì chúng ta sử dụng: NodeJS, HTML, CSS, JQuery, Angular, để phát triển. Ngôn ngữ NodeJS sẽ xử lý các yêu cầu từ phía client trước khi nó được gửi đến Server và xử lý dữ liệu mà Server trả về để hiển thị nó lên giao diện web, HTML + CSS + Angular dùng để xử lý giao diện web của hệ thống. Chức năng chính của web site: Danh sách người hỗ trợ khi cần, Tra cứu lịch sử cập nhật chứng chỉ, Tra cứu trạng thái hiện tại của chứng chỉ.

5.1.3 Kết quả đạt được




Hình 34: Giao diện trang chủ của web client.

Thông tin bằng tốt nghiệp




Mã chứng chỉ	2018240227
Tên chứng chỉ	BANG TOT NGHIEP DAI HOC
Họ tên	Nguyen Van Hai
Ngày sinh	2018-10-18
Năm tốt nghiệp	2018
Xếp loại	Gioi
Hình thức đào tạo	Chính Quy
Ngày phát hành	2018-10-18
Hiệu trưởng	Nguyen Hong Son
Trạng thái	Đang phát hành


Đóng




Nguyễn Văn Hải, *Hỗ trợ viên Đại Học Bách Khoa Hà Nội.*
 Email: hai213k57@gmail.com,
 Phone: 0967135492
 Sinh viên của Đại Học Bách Khoa Hà Nội.



Nguyễn Huyền My, *Quản trị viên Đại Học Bách Khoa Hà Nội.*
 Email: huyenmy@gmail.com,
 Phone: 0967135492
 Hỗ trợ và giải đáp các thắc mắc của sinh viên.

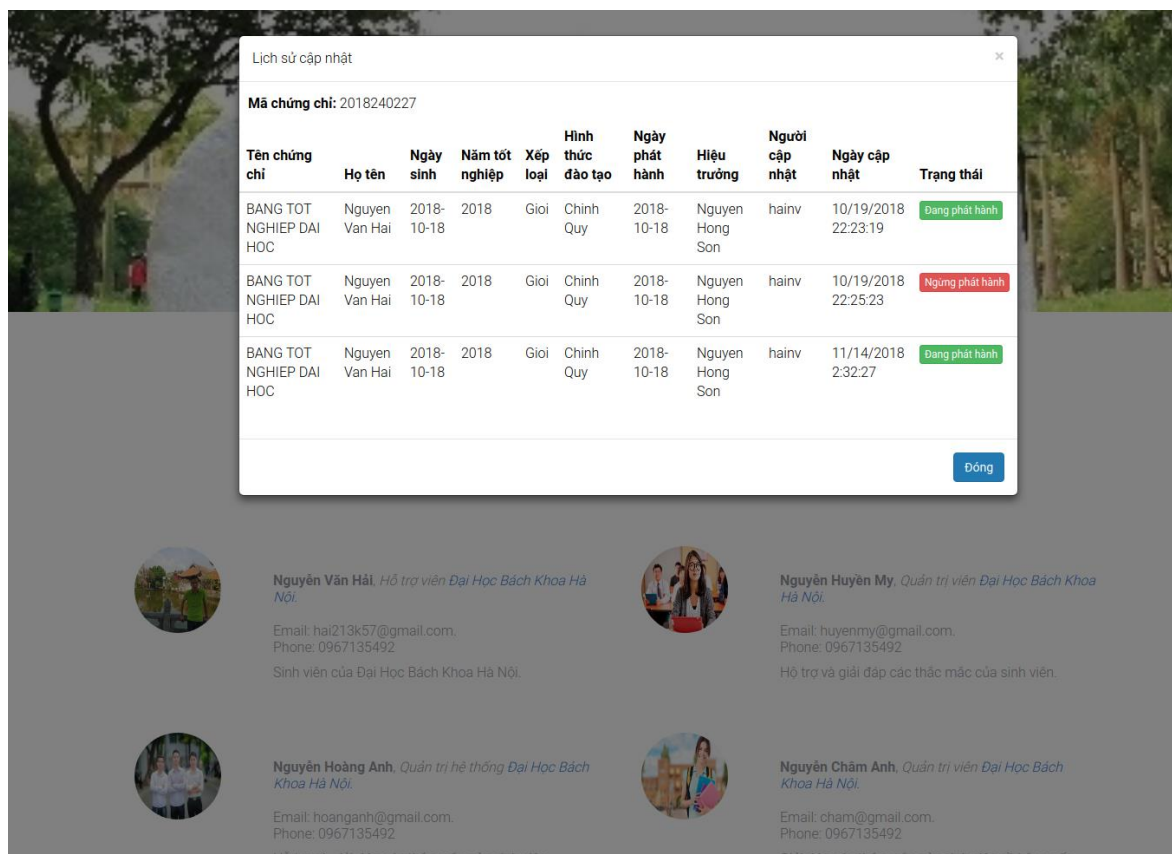


Nguyễn Hoàng Anh, *Quản trị hệ thống Đại Học Bách Khoa Hà Nội.*
 Email: hoanganh@gmail.com,
 Phone: 0967135492



Nguyễn Chăm Anh, *Quản trị viên Đại Học Bách Khoa Hà Nội.*
 Email: cham@gmail.com,
 Phone: 0967135492

Hình 35: Thông tin chi tiết trạng thái của chứng chỉ.



Hình 36: thông tin chi tiết lịch sử cập nhật chứng chỉ.

5.2 Phát triển server

5.2.1 Đặt vấn đề

Để có thể xử lý được các yêu cầu từ phía web client hay web admin thì chúng ta cần phát triển một Server để xử lý các yêu cầu gửi đến. Server này sẽ tương tác trực tiếp với cơ sở dữ liệu để: thêm, sửa, xóa cơ sở dữ liệu, ngoài ra Server sẽ phải kết nối đến hệ thống Blockchain để thực thi các hợp đồng thông minh. Hệ thống Server là cầu nối để web admin và web client có thể tương tác được với nhau và chia sẻ dữ liệu giữa cơ sở dữ liệu với hệ thống web site.

5.2.2 Giải pháp

Để có thể phát triển được hệ thống Server mà đáp ứng được các yêu cầu nêu ra thì việc chọn lựa một ngôn ngữ hay một nền tảng công nghệ gặp nhiều khó khăn,

bởi vì hiện nay có rất nhiều lựa chọn cho chúng ta. Để có thể chọn ra một ngôn ngữ phù hợp thì chúng ta dựa trên một số tiêu chí: xử lý các yêu cầu gửi đến nhanh, có thể xử lý được nhiều yêu cầu gửi đến trong một thời điểm,... Với tiêu chí đó thì NodeJS là một giải pháp tối ưu, NodeJS thường được một số các công ty lớn sử dụng khá nhiều.

5.2.3 Kết quả đạt được

STT	Tên API	Mô tả
1	/login	Đăng nhập
2	/getInfoAccount	Lấy thông tin của tài khoản
3	/getAllAccount	Lấy danh sách các tài khoản có trong hệ thống
4	/addAccount	Thêm tài khoản vào hệ thống
5	/deleteAccount	Xóa tài khoản
6	/updateAccount	Cập nhật thông tin tài khoản
7	/getAllPosition	Lấy về tất cả các chức vụ có trong hệ thống
8	/insertPosition	Thêm chức vụ vào hệ thống
9	/updatePosition	Cập nhật chức vụ
10	/deletePosition	Xóa chức vụ
11	/getAllCetificateCategory	Lấy danh mục các chứng chỉ
12	/insertCetificateCategory	Thêm danh mục chứng chỉ
13	/updateCetificateCategory	Cập nhật danh mục chứng chỉ
12	/deleteCetificateCategory	Xóa danh mục chứng chỉ
13	/getAllCetificateList	Lấy tất cả danh sách chứng chỉ
14	/insertCetificateList	Thêm chứng chỉ
15	/updateCetificateList	Cập nhật chứng chỉ
16	/deleteCetificateList	Xóa chứng chỉ

17	/getCertificateByCode	Lấy thông tin chứng chỉ theo mã chứng chỉ
18	/smartcontracts/getAuthorList	Lấy về danh sách người phát hành
19	/smartcontracts/addAuthor	Thêm người phát hành chứng chỉ
20	/smartcontracts/updateStatusAuthor	Cập nhật thông tin người phát hành chứng chỉ
21	/smartcontracts/getCertificateList	Lấy về danh sách chứng chỉ trên blockchain
22	/smartcontracts/addCertificate	Thêm chứng chỉ vào blockchain
23	/smartcontracts/updateCertificate	Cập nhật chứng chỉ trên blockchain
24	/smartcontracts/getDataChanegs	Lấy về lịch sử cập nhật chứng chỉ
25	/insertStudent	Thêm sinh viên
26	/updateStudent	Cập nhật thông tin sinh viên
27	/updateStudentStatus	Cập nhật trạng thái sinh viên
28	/getAllStudents	Lấy danh sách sinh viên có trong hệ thống
29	/getStudentAvailable	Lấy danh sách sinh viên chưa thuộc lớp nào
30	/getStudentActive	Lấy danh sách sinh viên đang hoạt động
31	/insertClassroom	Thêm lớp học
32	/updateClassroom	Cập nhật thông tin lớp học
33	/updateStatusClassroom	Cập nhật trạng thái của lớp học
34	/getAllClassroom	Lấy danh sách của lớp học
35	/getClassroomActive	Lấy về danh sách lớp học đang hoạt động
36	/getStudentOfClassroom	Lấy về danh sách của sinh viên thuộc một lớp học
37	/addStudyManager	Thêm người quản lý lớp học
38	/updateStudyManager	Cập nhật thông tin người quản lý lớp học
39	/destroyStudentOfClassroom	Xóa sinh viên khỏi lớp học
40	/addSupport	Thêm người hỗ trợ
41	/updateSupport	Cập nhật thông tin người hỗ trợ

42	/getSupportByStatus	Lấy về trạng thái của người hỗ trợ
43	/deleteSupport	Xóa người hỗ trợ
44	/upload	Upload hình ảnh

Hình 37: Danh sách các API của hệ thống

5.3 Phát triển web admin

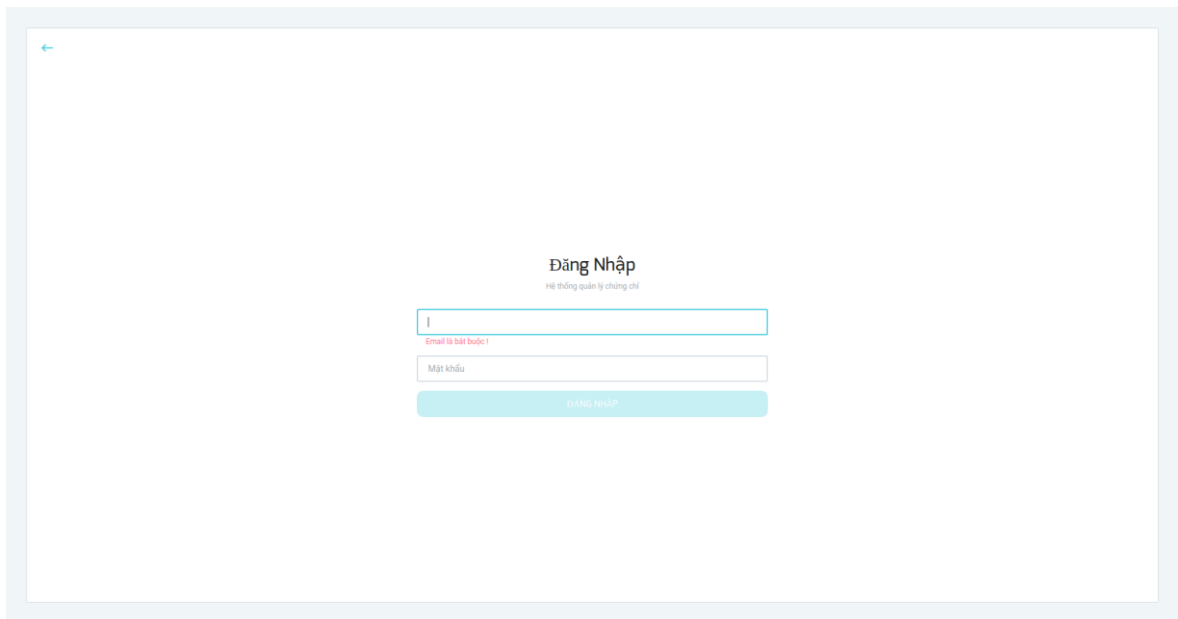
5.3.1 Đặt vấn đề

Để giúp người quản trị hệ thống có thể dễ dàng trong việc quản trị thì việc cung cấp một giao diện web site là một cách trực quan nhất, với giao diện họ có thể sử dụng các chức năng của hệ thống. Từ giao diện các yêu cầu sẽ được gửi đến Server và Server sẽ xử lý các yêu cầu đó, trả về kết quả của việc thực hiện các yêu cầu.

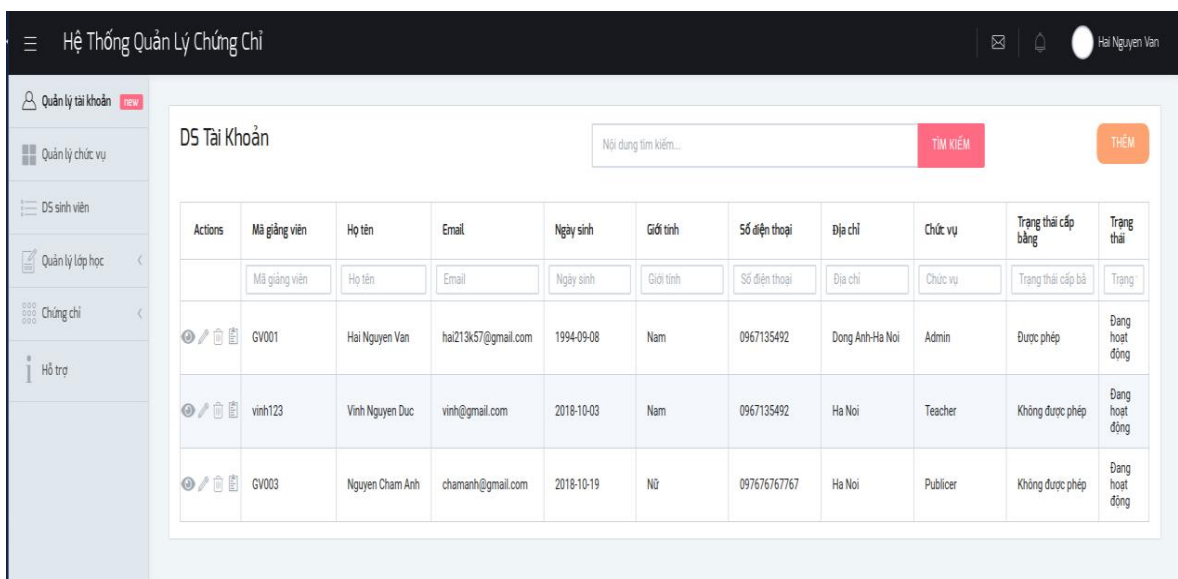
5.3.2 Giải pháp










Để có thể phát triển triển được web admin thì chúng ta sử dụng một số ngôn ngữ và framework: Angular 6, HTML, CSS, JQuery, Angular, để phát triển. Với Angular 6 thì chúng ta có thể có dễ dàng trong việc tạo ra một giao diện web admin và tập trung vào xử lý các yêu logic. Web admin xây dựng để giúp người dùng tương tác với hệ thống dễ hơn và cung cấp một số chức năng cơ bản: Quản lý sinh viên, Quản lý chứng chỉ, Quản lý lớp học, Quản lý tài khoản,...

5.3.3 Kết quả đạt được

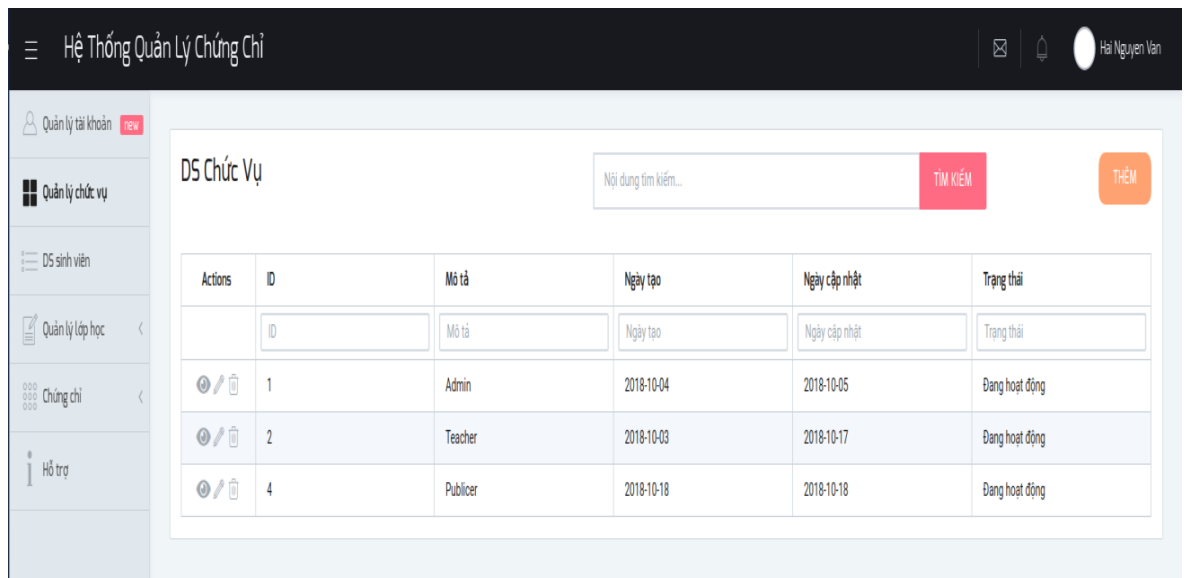


Hình 38: Màn hình đăng nhập vào hệ thống

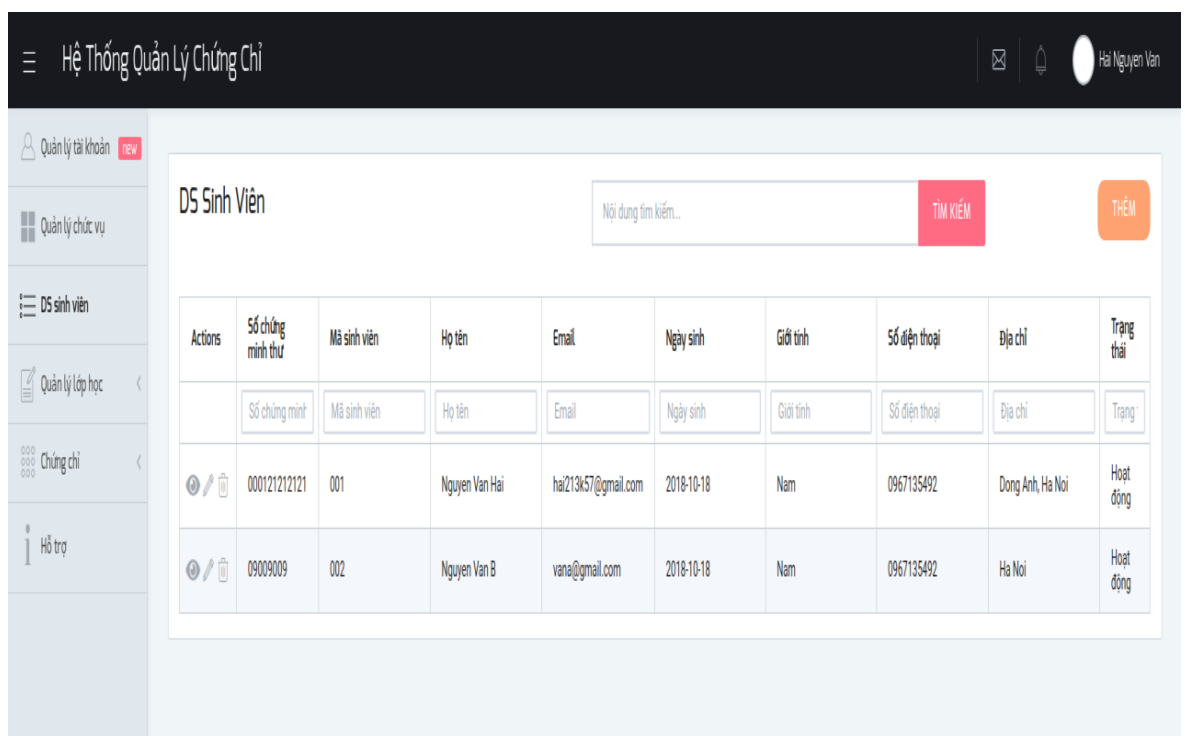


Actions	Mã giảng viên	Họ tên	Email	Ngày sinh	Giới tính	Số điện thoại	Địa chỉ	Chức vụ	Trạng thái cấp bằng	Trạng thái
	Mã giảng viên	Họ tên	Email	Ngày sinh	Giới tính	Số điện thoại	Địa chỉ	Chức vụ	Trạng thái cấp bằng	Trạng
  	GV001	Hai Nguyen Van	hai213k57@gmail.com	1994-09-08	Nam	0967135492	Dong Anh-Ha Noi	Admin	Được phép	Đang hoạt động
  	vinh123	Vinh Nguyen Duc	vinh@gmail.com	2018-10-03	Nam	0967135492	Ha Noi	Teacher	Không được phép	Đang hoạt động
  	GV003	Nguyen Cham Anh	chamanh@gmail.com	2018-10-19	Nữ	09767676767	Ha Noi	Publicer	Không được phép	Đang hoạt động

Hình 39Màn hình quản lý tài khoản



Hình 40: Màn hình quản lý chức vụ



Hình 41: Màn hình quản lý sinh viên

Hệ Thống Quản Lý Chứng Chỉ													
<div> <div> <div>Quản lý tài khoản</div> <div>Quản lý chức vụ</div> <div>D5 sinh viên</div> <div>Quản lý lớp học</div> <div>Danh sách lớp học</div> <div>Quản lý đào tạo</div> <div>Chứng chỉ</div> <div>Danh mục chứng chỉ</div> <div>D5 chứng chỉ</div> <div>Lịch sử cập nhật</div> <div>Hỗ trợ</div> </div> <div> <div>Lịch Sử Cập Nhật</div> <div>Nội dung tìm kiếm...</div> <div>TÌM KIẾM</div> </div> </div>													
Mã chứng chỉ	Tên chứng chỉ	Chữ ký	Tên sinh viên	Ngày sinh	Năm tốt nghiệp	Xếp loại	Hình thức đào tạo	Hiệu trưởng	Ngày phát hành	Trạng thái	Mã người cập nhật	Người cập nhật	Thời gian sửa
Mã chứng chỉ	Tên chứng chỉ	Chữ ký	Tên sinh viên	Ngày sinh	Năm tốt nghiệp	Xếp loại	Hình thức đào tạo	Hiệu trưởng	Ngày phát hành	Trạng thái	Mã người cập nhật	Người cập nhật	Thời gian sửa
2018240227	BANG TOT NGHIEP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Đang phát hành	GV001	hainv	10/19/2018 22:23:19
2018240227	BANG TOT NGHIEP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Ngừng hoạt động	GV001	hainv	10/19/2018 22:25:23
2018240227	BANG TOT NGHIEP DAI HOC	2018412426	Nguyen Van Hai	2018-10-18	2018	Giỏi	Chính Quy	Nguyen Hong Son	2018-10-18	Đang phát hành	GV001	hainv	11/14/2018 2:32:27

Hình 42: Màn hình hiển thị lịch sử cập nhật chứng chỉ của hệ thống.

5.4 Phát triển hợp đồng thông minh

5.4.1 Đặt vấn đề

Để có thể giải quyết được các vấn đề: Ai là người cập nhật ?, Được cập nhật khi nào ?, Nội dung cập nhật gồm những thông tin gì ? với các vấn đề được đưa ra thì việc nghiên cứu và phát triển hệ thống giải quyết các yêu cầu trên là một vấn đề còn gặp nhiều khó khăn khi mà công nghệ Blockchain chưa phát triển mạnh. Với nền tảng công nghệ Blockchain thì dữ liệu trên đó là bảo mật mà không ai có quyền có thể xóa dữ liệu một khi nó đã được ghi vào block.

5.4.2 Giải pháp

Việc ứng dụng công nghệ Blockchain để giải quyết vấn đề bảo mật dữ liệu và truy xuất được lịch sử cập nhật chứng chỉ không gặp quá nhiều khó khăn. Dữ liệu sau khi được ghi vào Block thì không thể ai có quyền xóa, chỉnh sửa nó. Các Block thì có chứa một mã hash đại diện cho Block đó và một mã hash để liên kết với Block trước đó, với cách quản lý này thì việc tấn công vào Block số 5 thì kẻ tấn công bắt buộc phải tấn công vào tất cả các Block trước và sau của Block số 5 vì vậy

nó làm cho quá trình tấn công trở lên không thể thực hiện được. Trong bài toán này chúng ta sử dụng nền tảng Ethereum để thực thi các hợp đồng thông minh.

5.4.3 Kết quả đạt được

Với bài toán phát hiện làm giả chứng chỉ, truy xuất được lịch sử cập nhật của chứng chỉ thì chúng ta phát triển một hợp đồng thông minh và nó được thực thi trên nền tảng Ethereum. Dưới đây là nội dung của hợp đồng thông minh:

```
5  pragma solidity ^0.4.24;
6
7  contract Cetification {
8
9      // Cấu trúc của một bằng tốt nghiệp
10     struct Certificate {
11         bytes32 code;
12         bytes32 title; // Tên bằng tốt nghiệp
13         bytes32 studentName; // Tên sinh viên
14         bytes32 dataOfBirth; // Ngày tháng năm sinh
15         uint yearOfGraduation; // Năm tốt nghiệp
16         bytes32 degreeClassification; // Loại bằng
17         bytes32 modeOfStudy; // Hình thức đào tạo
18         bytes32 date; // Ngày phát hành
19         bytes32 author; // Người cấp bằng
20         uint updateBy; // Id của người sửa dữ liệu
21         uint status; // Trang thai bang,
22         bytes32 timeUpdate;
23         bytes32 studentSign;
24     }
25
26     // 0: Active, -1: Delete
27     struct Author {
28         string id;
29         string name;
30         string sign;
31         string status;
32     }
33
34     // Lưu trữ số lượng người có quyền phát hành bằng tốt nghiệp
35     uint public authorsCount;
36
37     // Lưu trữ số lượng cetification
38     uint public certificatesCount;
39
40     // Mảng các bằng tốt nghiệp
41     mapping (uint => Certificate) public certificates;
42
43     // Mảng những người được cấp quyền để phát hành các chứng chỉ
44     mapping (uint => Author) public authors;
45
46     // Constructor
47     constructor () public {
```

```

48         addAuthor("GV001", "hainv", "0x00000000001121212121212121212121");
49     }
50
51     event eventUpdateCertificate(
52         bytes32 _code,
53         bytes32 _title,
54         bytes32 _studentName,
55         bytes32 _dataOfBirth,
56         uint _yearOfGraduation,
57         bytes32 _degreeClassification,
58         bytes32 _modeOfStudy,
59         bytes32 _date,
60         bytes32 _author,
61         uint _updateBy,
62         uint _status,
63         bytes32 _timeUpdate,
64         bytes32 _studentSign
65     );
66
67     // Add author
68     function addAuthor(string _id,string _name, string _sign) public
69     returns (bool) {
70         authorsCount ++;
71         authors[authorsCount] = Author(_id, _name, _sign,"0");
72         return true;
73     }
74
75     // Delete author
76     function updateStatusAuthor(uint index,string status) public returns
77     (bool){
78         authors[index].status = status;
79     }
80
81     // Add cetificates
82     function addCertificate(bytes32 code, bytes32 title,
83         bytes32 studentName, bytes32 dataOfBirth ,
84         uint yearOfGraduation, bytes32 degreeClassification,
85         bytes32 modeOfStudy, bytes32 date,
86         bytes32 author, uint updateBy,bytes32 timeUpdate, bytes32
87         studentSign) public
88     {
89         certificatesCount ++;
90         certificates[certificatesCount] = Certificate(code, title,
91         studentName, dataOfBirth , yearOfGraduation, degreeClassification,
92         modeOfStudy, date, author, updateBy,1,timeUpdate, studentSign);
93         emit eventUpdateCertificate(code, title, studentName, dataOfBirth ,
94         yearOfGraduation, degreeClassification, modeOfStudy, date, author,
95         updateBy,1,timeUpdate, studentSign);
96     }
97
98     // Update cetificates
99     function updateCertificate(uint index, bytes32 title,
100         bytes32 studentName, bytes32 dataOfBirth ,
101         uint yearOfGraduation, bytes32 degreeClassification,
102         bytes32 modeOfStudy, bytes32 date,
103         bytes32 author, uint updateBy, uint status,bytes32 timeUpdate,
104         bytes32 studentSign) public
105     {

```

```

98         certificates[index].title = title;
99         certificates[index].studentName = studentName;
100        certificates[index].dataOfBirth = dataOfBirth;
101        certificates[index].yearOfGraduation = yearOfGraduation;
102        certificates[index].degreeClassification = degreeClassification;
103        certificates[index].modeOfStudy = modeOfStudy;
104        certificates[index].date = date;
105        certificates[index].author = author;
106        certificates[index].updateBy = updateBy;
107        certificates[index].status = status;
108        certificates[index].timeUpdate = timeUpdate;
109        certificates[index].studentSign = studentSign;
110        emit eventUpdateCertificate(certificates[index].code, title,
studentName, dataOfBirth, yearOfGraduation, degreeClassification,
modeOfStudy, date, author, updateBy, status, timeUpdate, studentSign);
111    }
112 }

```

Thuộc tính/Phương thức	Kiểu dữ liệu	Mô tả
Certificate	struct	Là một đối tượng đại diện cho Certificate
Author	struct	Là một đối tượng đại diện cho Author
authorsCount	uint	Số lượng những người được quyền cập nhật chứng chỉ
certificatesCount	uint	Số lượng các chứng chỉ có trong hệ thống.
certificates	mapping	Chứa các chứng chỉ có trong hệ thống.
authors	mapping	Chứa các tác thông của người cấp phát chứng chỉ.
eventUpdateCertificate	event	Ghi lại lịch sử cập nhật chứng chỉ.
addAuthor	function	Hàm thêm người cập nhật chứng chỉ.
updateStatusAuthor	function	Cập nhật trạng thái người phát hành chứng chỉ.
addCertificate	function	Hàm thêm chứng chỉ của hệ thống.
updateCertificate	function	Hàm cập nhật thông tin của chứng chỉ.

Bảng 30 Đặc tả chi tiết các thuộc tính và phương thức của Smart Contracts

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Hệ thống trợ giúp quản lý chứng chỉ trên nền tảng Blockchain đáp ứng được các yêu cầu, chức năng đưa ra, có tính hoàn thiện và sau khi phát triển sẽ giải quyết được bài toán chống làm giả chứng chỉ đào tạo. Giải pháp công nghệ Blockchain không chỉ dừng ở việc giải quyết bài toán chống làm giả chứng chỉ đào tạo, mà nó có thể được ứng rộng một cách rộng hơn, trong nhiều lĩnh vực: Y tế, Giáo dục, Thương mại điện tử,..Tuy nhiên, trong khoảng thời gian có hạn và trình độ, hiểu biết của em đối với nền tảng Blockchain còn hạn nên hệ thống vẫn còn tồn đọng một số vấn đề cần cải thiện thêm.

Sau một thời gian nghiên cứu và phát triển hệ thống dưới sự hướng dẫn, chỉ bảo của **Th.S Lê Đức Trung**, em đã hoàn thành các yêu cầu ban đầu đặt ra và trong quá trình thực hiện ĐATN thì em đã thu được nhiều kiến thức bổ ích, có được hiểu biết sâu hơn về nền tảng công nghệ Blockchain. Biết cách khắc phục vấn đề, giải quyết các bài toán được đặt ra. Không chỉ dừng lại ở các vấn đề lập trình, mà bên cạnh đó là việc tạo ra được một hệ thống có thể ứng dụng được vào trong thực tế.

6.2 Hướng phát triển

Để hệ thống không chỉ dừng lại ở việc trợ giúp quản lý chứng chỉ và chống làm giảm chứng chỉ, trong tương lai em sẽ tiếp tục phát triển hệ thống để có thể giải quyết được các vấn đề:

- Nghiên cứu thuật toán đồng thuận và cải tiến để có thể giảm thời gian ghi block vào Blockchain.
- Nghiên cứu và phát triển, tích hợp giải pháp vào hệ thống để có thể phát hiện các trường hợp sửa điểm thi.
- Nghiên cứu và phát triển để hệ thống có thể là một giải pháp cho học bạ điện tử.

Tài liệu tham khảo

- [1] MySQL Cookbook, 3rd Edition
- [2] Building Blockchain Projects
- [3] Mastering Blockchain
- [4] Pro Node.js for Developers
- [5] Cơ sở dữ liệu MYSQL: <https://www.mysql.com/>
- [6] AngularJS: <https://angularjs.org/>.
- [7] NodeJS: <https://nodejs.vn/>
- [8] Tài liệu Ethereum: <http://www.ethdocs.org/en/latest/>
- [9] Tài liệu về Solidity:
https://ethereumbuilders.gitbooks.io/guide/content/en/solidity_tutorials.html