

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

THIẾT KẾ WEBSITE QUẢN LÝ LỊCH THI

Sinh viên thực hiện: TRẦN TUẤN ANH – ĐT08
 TRƯƠNG VĂN KHÁNH – ĐT07
Giảng viên hướng dẫn: TS. NGUYỄN THANH BÌNH
Cán bộ phản biện:

Hà Nội, 8-2023

ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho giảng viên hướng dẫn)

Tên giảng viên đánh giá: **TS. Nguyễn Thanh Bình**.....

Họ và tên Sinh viên 1: **Trần Tuấn Anh**MSSV: **20182375**

Họ và tên Sinh viên 2: **Trương Văn Khánh**MSSV: **20182606**

Tên đồ án: **Thiết kế Website quản lý thi học kỳ**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4	5
Kỹ năng viết quyền đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5

Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)		
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế	5
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.	2
10c	Không có thành tích về nghiên cứu khoa học	0
Điểm tổng		/50
Điểm tổng quy đổi về thang 10		

***Nhận xét khác** (về thái độ và tinh thần làm việc của sinh viên)*

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

Người nhận xét

(Ký và ghi rõ họ tên)

ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho cán bộ phản biện)

Giảng viên đánh giá:.....

Họ và tên sinh viên: **Trần Tuấn Anh**..... MSSV: **20182364**

Họ và tên sinh viên: **Trương Văn Khánh**..... MSSV: **20182606**

Tên đồ án: **Thiết kế Website quản lý thi học kỳ**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4	5
Kỹ năng viết quyền đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5

Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)		
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế	5
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.	2
10c	Không có thành tích về nghiên cứu khoa học	0
Điểm tổng		/50
Điểm tổng quy đổi về thang 10		

Nhận xét khác của cán bộ phản biện

.....

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

Người nhận xét

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại công nghệ hiện nay, việc sử dụng công nghệ thông tin vào các quy trình nghiệp vụ trong cuộc sống đã trở nên rất phổ biến. Có thể thấy các mô hình dịch vụ như quán café, nhà hàng, khách sạn ... đều được số hóa và quản lý bằng công nghệ. Trong giáo dục cũng vậy, việc áp dụng công nghệ cũng là một xu hướng không thể thiếu.

Nhờ vào tiến bộ của công nghệ thông tin, ứng dụng các phần mềm cần thiết giúp mang lại những cải thiện đáng kể trong việc nâng cao chất lượng giảng dạy và học tập. Từ cáo ứng dụng mobile cho đến phần mềm quản lý học tập cũng như các công cụ học trực tuyến như Team, Zoom, ... giáo viên và học sinh có thể có một môi trường học linh hoạt và sáng tạo. Và đó là lí do vì sao Website quản lý thi học kỳ ra đời và nhận được sự quan tâm đặc biệt.

Website Quản lý thi học kỳ làm một nền tảng đáng tin cậy và tiện ích, nhằm tối ưu hóa quá trình quản lý thi trong môi trường đại học. Với các tính năng đa dạng và linh hoạt nó mang lại một cách quản lý thực sự hiệu quả hơn so với sổ sách truyền thống.

Website nhằm giúp Phòng đào tạo quản lý các thông tin về sinh viên, giáo viên, môn học và tổ chức lịch thi một cách dễ dàng. Sinh viên cũng có thể truy cập trang web để tra cứu lịch thi của mình đảm bảo việc tổ chức thi được tiến hành một cách hiệu quả nhất.

LỜI CAM ĐOAN

Chúng tôi là Trần Tuấn Anh - mã số sinh viên 20182375 và Trương Văn Khánh – mã số sinh viên 20182606, khóa 63, viện Điện tử viễn thông. Người hướng dẫn là TS. Nguyễn Thanh Bình. Chúng tôi xin cam đoan toàn bộ nội dung được trình bày trong đồ án “*Thiết kế Website quản lý thi học kỳ*” là kết quả của quá trình nghiên cứu và tìm hiểu của tôi. Các dữ liệu được nêu ra trong đồ án là hoàn toàn trung thực, phản ánh đúng với kết quả thực hiện và đo đạc thực tế. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ, các tài liệu tham khảo được liệt kê rõ ràng. Chúng tôi xin chịu hoàn toàn trách nhiệm đối với những nội dung được viết trong đồ án này.

Hà Nội, ngày 15 tháng 07 năm 2023

Người cam đoan

Trần Tuấn Anh

Trương Văn Khánh

TÓM TẮT ĐỒ ÁN

Đồ án "Thiết kế Website quản lý thi học kỳ" tập trung vào việc thiết kế một hệ thống quản lý thi học kỳ sử dụng các công nghệ web hiện đại. Website bao gồm các chức năng như thêm, sửa, xóa người dùng, học sinh, môn học, giáo viên cùng với tạo và xem lịch thi. Hệ thống hỗ trợ giáo viên đăng ký lịch thi môn mình giảng dạy và dựa trên đó admin có thể tạo lịch thi phù hợp.

Website cho phép quản trị viên (admin) quản lý người dùng bao gồm giáo viên và học sinh. Admin có thể thêm mới hoặc sửa đổi, xóa các tài khoản người dùng. Học sinh và giáo viên thì có thể đăng nhập vào hệ thống với quyền của mình và sử dụng những chức năng phù hợp.

Hệ thống cho phép giáo viên đăng ký các môn thi và thời gian thi tương ứng. Khi giáo viên đăng ký thi, hệ thống sẽ kiểm tra tính hợp lệ của lịch thi, bao gồm xung đột thời gian và địa điểm. Từ đó admin đưa ra một lịch thi hợp lệ, hệ thống sẽ ghi nhận lại và hiển thị lịch thi cho sinh viên. Sinh viên cũng có thể lọc ra lịch thi của chính mình và xuất ra dưới dạng file Excel.

Trong đồ án cũng đề cập đến một số kỹ thuật và công nghệ được sử dụng để xây dựng Website, bao gồm: HTML, CSS, JavaScript, ReactJS, Antd, MySQL, MongoDB.

Tóm lại, đồ án "Thiết kế Website quản lý thi học kỳ" nhằm tạo ra một hệ thống quản lý thi toàn diện và tiện ích, hệ thống giúp cải thiện quá trình quản lý trong môi trường giáo dục, tăng cường tính chính xác và đáng tin cậy của dữ liệu, tạo điều kiện thuận lợi cho công tác tổ chức thi.

PROJECT SUMMARY

The project "Design of a Semester Examination Management Website" focuses on designing a system for managing semester examinations using modern web technologies. The website includes functions such as adding, editing, and deleting users, students, courses, and teachers, as well as creating and viewing examination schedules. The system supports teachers in registering examination schedules for the courses they teach, and based on that, the admin can generate appropriate examination schedules.

The website allows the administrator (admin) to manage users, including teachers and students. The admin can add, edit, or delete user accounts. Students and teachers can log into the system with their respective roles and utilize the relevant functions.

The system enables teachers to register courses and corresponding examination schedules. When teachers register for examinations, the system verifies the validity of the schedules, including time and venue conflicts. Once a valid examination schedule is established, the system records and displays it for students. Students can also filter their own examination schedules and export them in Excel format.

The project also discusses various techniques and technologies used to build the website, including HTML, CSS, JavaScript, ReactJS, MySQL, and MongoDB.

In summary, the "Design of a Semester Examination Management Website" project aims to create a comprehensive and convenient examination management system. It improves the management process in an educational environment, enhances the accuracy and reliability of data, and provides a favorable environment for examination organization.

PHÂN CÔNG CÔNG VIỆC

Thời gian	Sinh viên thực hiện	Công việc
4/4	Trần Tuấn Anh	Trao đổi với giáo viên về các đề tài khả thi Đề xuất đề tài Thiết kế website quản lý thi học kỳ
	Trương Văn Khánh	
6/4- 15/4	Trần Tuấn Anh	Tìm hiểu đề tài đưa ra một số giải pháp để thực hiện đề tài (đưa ra công nghệ dùng cho backend, frontend, database...)
	Trương Văn Khánh	
15/4- 15/5	Trần Tuấn Anh	Bước đầu xây dựng các thành phần UI cơ bản: theme, sidebar, component cơ bản, ... Xây dựng controller cho server side
	Trương Văn Khánh	
15/5- 28/5	Trương Văn Khánh	Xây dựng các CRUD cho các thành phần sinh viên, giáo viên, môn học
	Trần Tuấn Anh	
28/5- 14/6	Trương Văn Khánh	Thực hiện phần đăng nhập cho user sử dụng redux. Thực hiện deploy frontend và backend lên các server free để test thử.
	Trần Tuấn Anh	
14/6- 20/7	Trương Văn Khánh	Hoàn thiện báo cáo
	Trần Tuấn Anh	

Danh mục ký hiệu và chữ viết tắt

Từ viết tắt	Tên đầy đủ	Ý nghĩa
UI	User Interface	Giao diện người dùng
DOM	Document Object Model	Mô hình các đối tượng tài liệu
JSX	JavaScript XML	Cú pháp mở rộng cho phép viết HTML trong React
NPM	Node Package Manager	Công cụ tạo và quản lý các thư viện JavaScript cho Node.js
CSDL	Cơ sở dữ liệu	
RDBMS	Relational Database Management System	Hệ quản trị cơ sở dữ liệu quan hệ
CRUD	Create, Read, Update, Delete	
HP	Học phần	
CRC	Class Responsibility Colabolation	Sử dụng để định rõ trách nhiệm và tương tác giữa các lớp trong hệ thống
ID	Identifier	
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
UML	Unified Model Language	Ngôn ngữ đồ họa và kỹ thuật mô hình hóa trong xây dựng phần mềm
BSON	Binary JSON	Định dạng trao đổi dữ liệu máy tính dạng nhị phân
JSON	JavaScript Object Notation	Kiểu dữ liệu mở trong Javascript có dạng text
SQL	Structured Query Language	Ngôn ngữ truy vấn cấu trúc

Danh mục hình vẽ

Hình 1.1 Sơ đồ Document Object Element	2
Hình 1.2 Virtual DOM trong ReactJS	2
Hình 1.3 Cấu trúc công cụ Redux.....	3
Hình 1.4 Ảnh chụp màn hình trang chủ website Node.js	4
Hình 1.5 Khảo sát về 10 framework và công nghệ từ lập trình viên năm 2022	6
Hình 1.6 Sơ đồ kết nối mongoDB và Nodejs	7
Hình 1.7 Hình ảnh MySQL	9
Hình 1.8 Một số ưu điểm của MySQL	11
Hình 4.1 Sơ đồ Usecase	22
Hình 4.2 Sơ đồ các lớp cho hệ thống quản lý lịch thi học kỳ.....	32
Hình 4.3 Sơ đồ Sequence đăng nhập	33
Hình 4.4 Sơ đồ Sequence quản lý người dùng	34
Hình 4.5 Sơ đồ Sequence cập nhật học phần	35
Hình 4.6 Sơ đồ Sequence cập nhật thời gian và hình thức thi.....	36
Hình 4.7 Sơ đồ Sequence download danh sách thi.....	37
Hình 4.8 Sơ đồ Sequence xem lịch thi	38
Hình 5.1 Sơ đồ thực thể liên kết	40
Hình 5.2 Sơ đồ cơ dữ liệu SQL	41
Hình 5.3 Sơ đồ cơ sở dữ liệu NoSQL.....	42
Hình 6.1 Mô hình MVC	43
Hình 6.2 Trang chủ Website.....	44
Hình 6.3 Đăng nhập.....	45
Hình 6.4 Dữ liệu sinh viên trong MongoDB	46
Hình 6.5 Giao diện thay đổi mật khẩu cho tài khoản giáo viên	46
Hình 6.6 Giao diện dashboard admin	47
Hình 6.7 Giao diện thêm user	48
Hình 6.8 Thêm giáo viên	48
Hình 6.9 Danh sách giáo viên.....	49
Hình 6.10 Chỉnh sửa giáo viên	49
Hình 6.11 Thêm sinh viên	50
Hình 6.12 Danh sách sinh viên	50
Hình 6.13 Chỉnh sửa sinh viên	51
Hình 6.14 Danh sách học phần.....	51

Hình 6.15 Danh sách lớp thi	52
Hình 6.16 Sắp xếp lịch thi	52
Hình 6.17 Danh sách lịch thi	53
Hình 6.18 Chính sửa lịch thi.....	53
Hình 6.19 Đăng ký hình thức thi	54
Hình 6.20 Đăng ký hình thức thi	54
Hình 6.21 Xem danh sách thi và in ra	55
Hình 6.22 Xem thông tin lịch thi.....	55

Danh mục bảng biểu

Bảng 1-1 Một số câu lệnh cơ bản mongoDB và MySQL	7
Bảng 1-2 So sánh giữa SQL DB (RDBMS) schema và MongoDB schema	9
Bảng 3-1 Bảng từ điển dữ liệu.....	20
Bảng 4-1 Mô tả usecase đăng nhập ở hình 4.1	22
Bảng 4-2 Bảng mô tả chức năng quản lí người dùng ở hình 4.1	23
Bảng 4-3 Bảng mô tả chức năng quản lí học phần ở hình 4.1	23
Bảng 4-4 Bảng mô tả chức năng quản lí sinh viên ở hình 4.1	24
Bảng 4-5 Bảng mô tả chức năng quản lí giáo viên ở hình 4.1	24
Bảng 4-6 Bảng mô tả chức năng quản lí lớp thi ở hình 4.1	25
Bảng 4-7 Bảng mô tả chức năng quản lí lịch thi ở hình 4.1	25
Bảng 4-8 Bảng mô tả đăng kí hình thức thi.....	25
Bảng 4-9 Bảng mô tả hình thức Download danh sách thi	26
Bảng 4-10 Bảng mô tả hình thức xem lịch thi.....	26
Bảng 4-11 Thẻ CRC cho lớp Person	27
Bảng 4-12 Thẻ CRC cho lớp Giáo vụ	27
Bảng 4-13 Thẻ CRC cho lớp Giáo viên	28
Bảng 4-14 Thẻ CRC cho lớp Sinh viên.....	29
Bảng 4-15 Thẻ CRC cho lớp Học phần.....	29
Bảng 4-16 Thẻ CRC cho lớp Lớp thi	30
Bảng 4-17 Thẻ CRC cho lớp Lịch thi.....	30
Bảng 4-18 Bảng ánh xạ giữa các chức năng và thẻ CRC.....	31

MỤC LỤC

TÓM TẮT ĐỒ ÁN.....	ii
PROJECT SUMMARY	iii
Danh mục ký hiệu và chữ viết tắt.....	v
Danh mục hình vẽ	vi
Danh mục bảng biểu.....	viii
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	1
1.1 Cơ sở lý thuyết	1
1.1.1 Giới thiệu về ReactJS và công cụ Redux.....	1
1.1.2 Giới thiệu về Node.js	4
1.1.3 Giới thiệu về MongoDB và MySQL	6
CHƯƠNG 2: TỔNG QUAN HỆ THỐNG	13
2.1 Tổng quan đề tài.....	13
2.1.1 Lý do chọn đề tài.....	13
2.1.2 Phạm vi của đề tài.....	13
2.1.3 Mục tiêu của đề tài.....	13
2.1.4 Đề xuất giải pháp	14
2.2 Phân tích tính khả thi	14
2.2.1 Khả thi về mặt kỹ thuật.....	14
2.2.2 Khả thi về mặt tổ chức	15
CHƯƠNG 3: THU THẬP YÊU CẦU.....	16
3.1 Khảo sát thực trạng.....	16
3.1.1 Cơ cấu tổ chức của hệ thống.....	16
3.1.2 Quy trình nghiệp vụ của hệ thống hiện tại.....	17
3.1.3 Đề xuất phương án.....	17
3.2 Các yêu cầu của hệ thống.....	18
3.2.1 Yêu cầu chức năng.....	18
3.2.2 Yêu cầu phi chức năng.....	19
3.2.3 Yêu cầu về dữ liệu	19

CHƯƠNG 4: PHÂN TÍCH HỆ THỐNG	21
4.1 Xây dựng kịch bản sử dụng.....	21
4.1.1 Sơ đồ Usecase	21
4.1.2 Mô hình hóa cấu trúc	27
4.1.3 Mô hình hóa hoạt động hệ thống	32
CHƯƠNG 5: THIẾT KẾ HỆ THỐNG	39
5.1 Thiết kế sơ đồ thực thể liên kết (entity- relationship model)	39
5.2 Thiết kế cơ sở dữ liệu SQL	40
5.3 Thiết kế cơ sở dữ liệu NoSQL	41
5.4 So sánh 2 phương pháp thiết kế cơ sở dữ liệu.....	42
CHƯƠNG 6: TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC	43
6.1 Mô hình MVC	43
6.2 Giao diện đăng nhập và giao diện trang chủ	44
6.2.1 Giao diện trang chủ.....	44
6.2.2 Quy trình đăng nhập.....	45
6.2.3 Thay đổi mật khẩu cho các tài khoản.....	46
6.3 Giao diện website của từng user	47
6.2.1 Quy trình làm việc của admin (giáo vụ)	47
6.2.2 Quy trình làm việc của giáo viên	53
6.2.3 Quy trình với tài khoản sinh viên	55
6.3 Kết luận chương	55
KẾT LUẬN	56
Kết quả đạt được	56
Khó khăn gặp phải.....	56
Đề xuất và hướng phát triển.....	56
DANH MỤC TÀI LIỆU THAM KHẢO	57

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Cơ sở lý thuyết

1.1.1 Giới thiệu về ReactJS và công cụ Redux

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

1.1.1.1 Tổng quan về ReactJS

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, nó có thể bao gồm các menu, thành tìm kiếm, sidebar, nút ấn, thẻ ... Giả sử ta cần lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm rồi nhấn vào nút thêm vào giỏ hàng, việc tiếp theo ta phải làm là hiện thị sản phẩm đó đã thêm vào giỏ hàng để user vào có thể thấy, việc này gọi là xử lý tương tác.

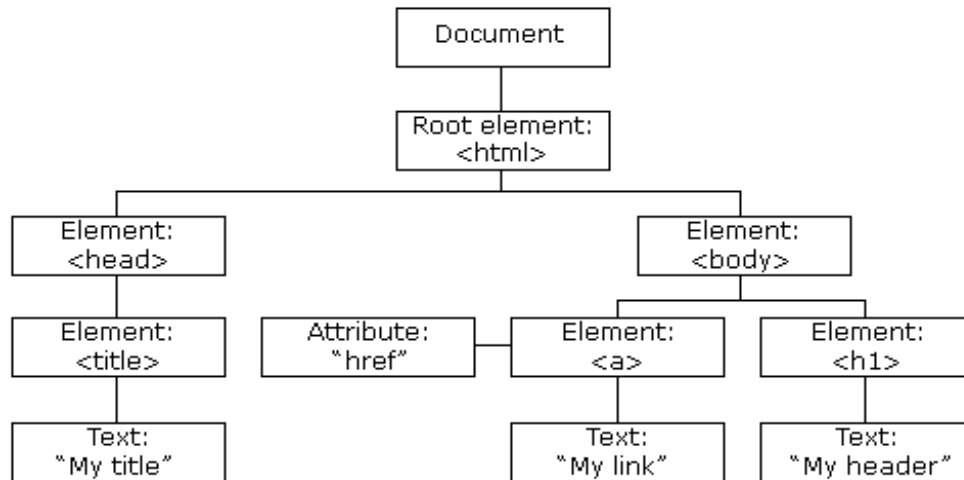
Trước khi có ReactJS, Lập trình viên thường gặp khó khăn khi sử dụng JavaScript thuần và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển sản phẩm diễn ra lâu hơn và xuất hiện nhiều rủi ro hơn. Vậy nên, vào năm 2011, Jordan Walke - một nhân viên Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Để tăng tốc quá trình phát triển và giảm những rủi ro không đáng có, ReactJS cung cấp cho người dùng khả năng Reusable Code.

Trong khi sử dụng React cho việc xây dựng website, có 2 khái niệm cần quan tâm:

- Khái niệm JSX:

Phần trọng tâm của một website bất kỳ là những HTML documents. Khi ta sử dụng trình duyệt để đọc các document này nội dung của website sẽ được hiện trên thiết bị duyệt web đó. Trình duyệt sẽ gọi đến một thứ gọi là Document Object Model (DOM) – một tree đại diện cho các cấu trúc website được hiển thị. Lập trình viên thêm các content động vào website bằng cách dùng JavaScript để thay đổi DOM. JSX nói ngắn gọn là một React extension cho phép chúng ta dễ dàng thay đổi cây DOM bằng các cấu trúc code đơn giản.

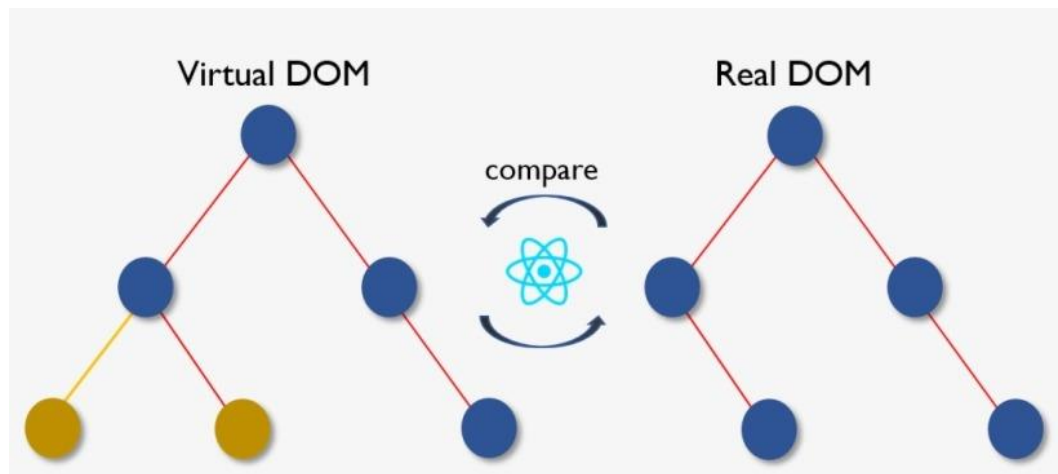


Hình 1.1 Sơ đồ Document Object Element

- **Khái niệm Virtual DOM:**

Đối với các website lớn, đặc biệt là các website thiên về việc xử lý tương tác người dùng nhiều, việc liên tục reload lại cây DOM khiến cho hiệu suất của website trở nên không tốt vì mỗi lần người dùng nhấn vào tính năng yêu cầu phải tải lại trang thì lại phải load lại toàn bộ.

Tuy nhiên nếu ta dùng JSX thì sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS khởi tạo một thứ gọi là Virtual DOM. Đây là bản copy của DOM thật trên trang đó, React dùng nó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi một sự kiện nào đó xảy ra. Nói cách khác, React sẽ giúp bạn scan trên Virtual DOM để tìm ra chỗ cần thay đổi và chỉ thay đổi nó.



Hình 1.2 Virtual DOM trong ReactJS

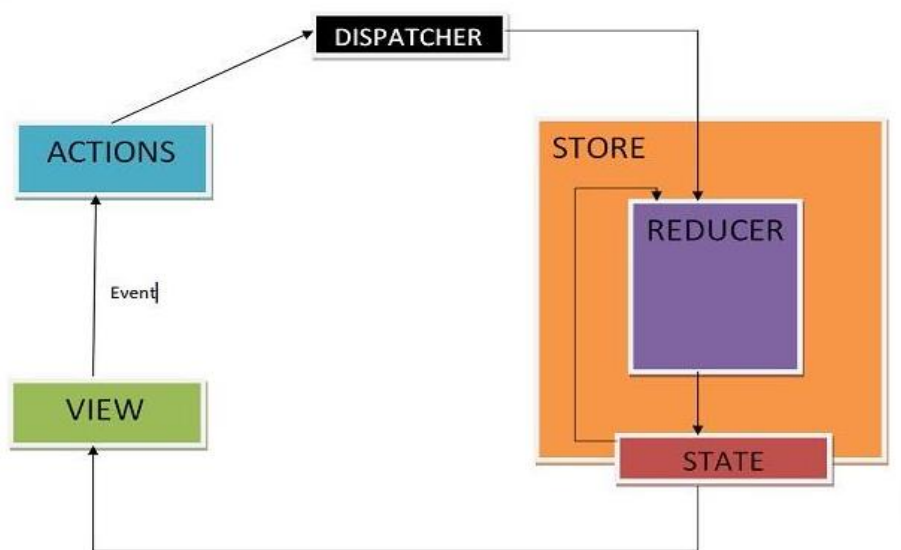
1.1.1.2 Tổng quan về Redux

Redux là một công cụ quản lý trạng thái, nó thường được dùng cùng ReactJS nhưng nó cũng có thể dùng với các thư viện JavaScript khác. Dung lượng của Redux là rất nhẹ nên không sợ việc sử dụng sẽ làm cồng kềnh dự án của chúng ta.

Khi sử dụng Redux, trạng thái của chúng ta được giữ trong một “store” và mỗi thành phần có thể truy cập bất kỳ trạng thái nào mà nó cần từ “store” này. “Store” trung tâm này chứa toàn bộ trạng thái của ứng dụng, mỗi thành phần có thể truy cập trạng thái được lưu trong store mà không cần phải truyền từ thành phần này sang thành phần khác.

Trong Redux có 3 thành phần cần được quan tâm:

- **Actions:**
Action ở đây chính là sự kiện xảy ra, đây là cách để ta có thể gửi dữ liệu của mình đến “store” Redux.
Các hành động này được gửi đến bằng phương thức `store.dispatch()`. Các action này cần có thuộc tính loại để chỉ ra loại hành động cần thực hiện cũng như có “payload” có chứa thông tin cần được xử lý.
Hành động được tạo thông qua Action Creator.
- **Reducers:**
Là các hàm thuần túy lấy trạng thái hiện tại của ứng dụng, thực hiện một hành động và sau đó trả về một trạng thái mới. Các trạng thái này xác định trạng thái của ứng dụng thay đổi như thế nào khi các action được thực hiện.
- **Store:**
Là nơi để giữ trạng thái của ứng dụng. Lưu ý, chỉ có một “store” trong bất kỳ ứng dụng redux nào. Các hành động được thực hiện trên trạng thái luôn trả về một trạng thái mới. Vì vậy, quản lý trạng thái là rất dễ dàng và có thể dự đoán.



Hình 1.3 Cấu trúc công cụ Redux

1.1.2 Giới thiệu về Node.js

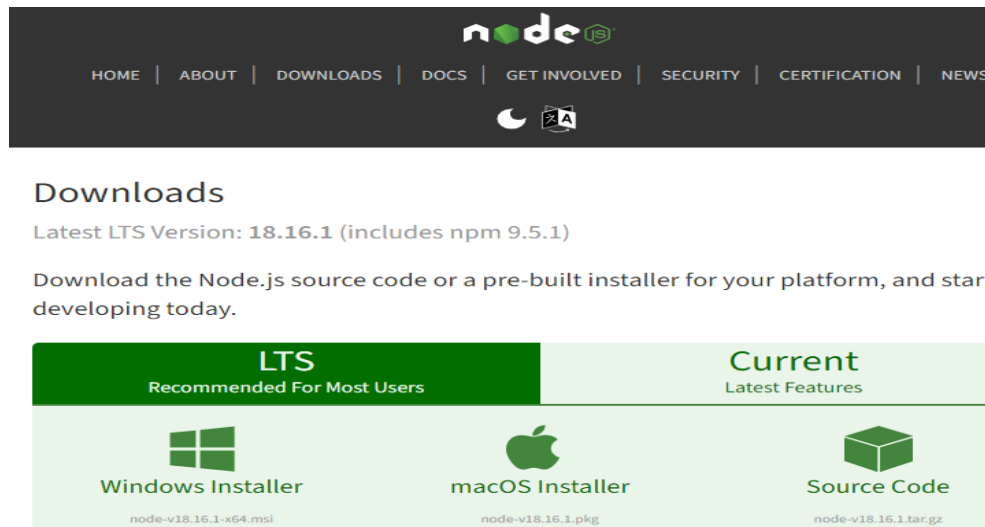
Node.js là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng một cách nhanh chóng và dễ dàng mở rộng.

Ưu điểm của Node.js:

- I/O hướng sự kiện không đồng bộ, cho phép các developer xử lý nhiều sự kiện cùng một lúc.
- Node.js sử dụng ngôn ngữ là JavaScript, đây là một ngôn ngữ dễ học và quen thuộc với nhiều lập trình viên.
- Có thể cùng chia sẻ code ở cả phía server và client.
- NPM (Node Package Manager) và các module Node ngày càng phát triển.
- Có cộng đồng người dùng lớn.

Nhược điểm của Node.js:

- Node.js hạn chế khả năng mở rộng, do đó các nhà phát triển không thể tận dụng các phần cứng cấp server với lợi thế về mô hình đa lõi hiện nay.
- Node.js tỏ ra khá khó thao tác với các cơ sở dữ liệu quan hệ.
- Mỗi Callback lại cần phải đi kèm với các Callback lồng nhau.
- Node.js không thực sự phù hợp với các thao tác cần nhiều CPU.



Hình 1.4 Ảnh chụp màn hình trang chủ website Node.js

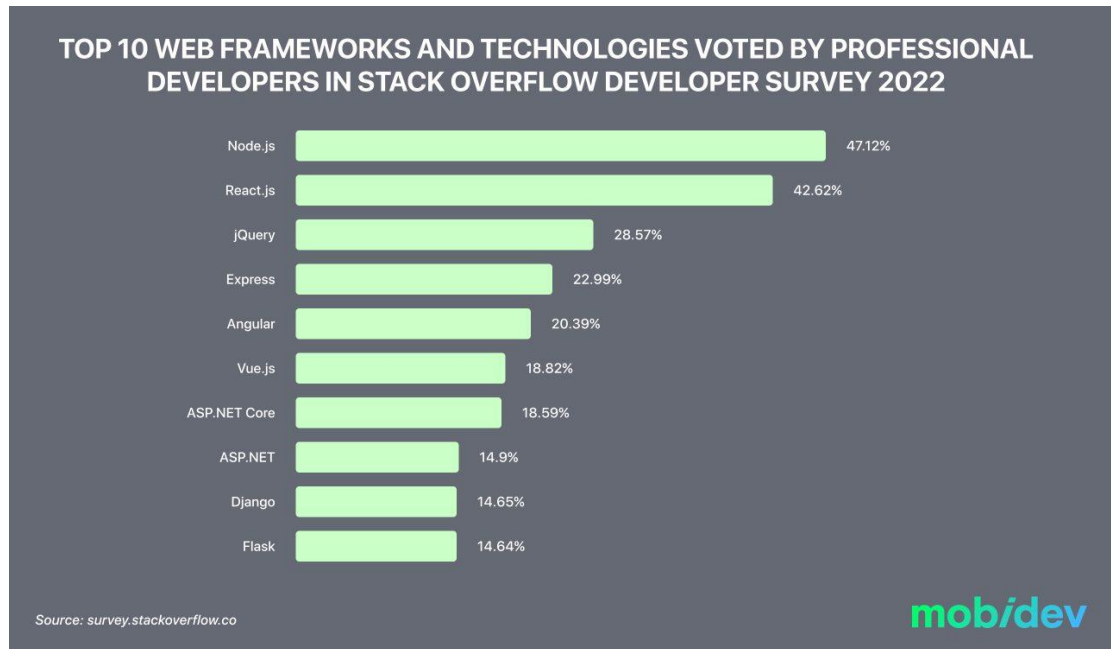
Những ứng dụng nên viết bằng Node.js:

Có thể thấy không phải Node.js có thể làm tốt mọi thứ, nếu chúng ta cần một hệ thống với tính ổn định cao, logic phức tạp thì các ngôn ngữ như PHP hay Java, Ruby... là một sự lựa chọn đáng tin cậy. Tuy nhiên, bên cạnh đó cũng có những ứng dụng nên viết bằng Node.js.

- Websocket server: các máy chủ web socket như là Game Server chẳng hạn
- Fast File Upload Client: là các chương trình upload file tốc độ cao
- Ứng dụng web realtime: Nodejs được sử dụng rộng rãi cho các ứng dụng web realtime như chat applications, livestream do có khả năng xử lý đồng thời tốt.
- Microservices: là lựa chọn phổ biến cho việc xây dựng microservices, với Node, ta có thể triển khai các dịch vụ nhỏ độc lập, cho phép phát triển và mở rộng ứng dụng.
- IoT (Internet of Things) applications: được sử dụng trong việc phát triển các ứng dụng IoT nhờ vào khả năng xử lý sự kiện và giao tiếp realtime.

Thuận lợi khi sử dụng Nodejs cho việc viết backend website:

- Hiệu suất cao và khả năng xử lý đồng thời: Node sử dụng kiến trúc đơn luồng sự kiện (event-driven) và không chặn (non-blocking), giúp xử lý nhiều yêu cầu cùng một lúc mà không làm gián đoạn các yêu cầu khác.
- Tích hợp dễ dàng: Nodejs có hệ sinh thái module đa dạng, cho phép ta sử dụng những module bên ngoài để xây dựng ứng dụng một cách thuận tiện nhất, việc cùng sử dụng ngôn ngữ JavaScript – là ngôn ngữ dùng cho cả frontend và backend giúp việc chia sẻ mã giữa các thành phần của dự án dễ dàng hơn.
- Tốc độ phát triển dự án: Nodejs có cú pháp dễ học và là môi trường linh hoạt cho phát triển ứng dụng. Ta có thể nhanh chóng xây dựng các APIs đơn giản và các ứng dụng không quá phức tạp một cách nhanh chóng, bên cạnh đó còn có nhiều công cụ và framework như Express.js, Nest.js giúp đơn giản hóa việc phát triển và tạo ra ứng dụng nhanh chóng.
- Cộng đồng và tài liệu tham khảo: cộng đồng lớn và phong phú, tài liệu dễ tìm kiếm.



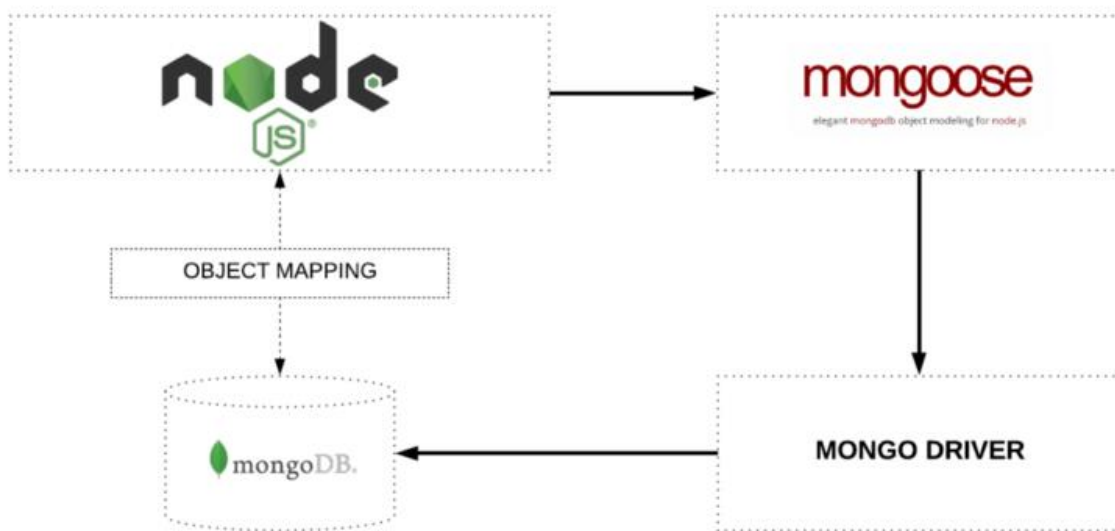
Hình 1.5 Khảo sát về 10 framework và công nghệ từ lập trình viên năm 2022

1.1.3 Giới thiệu về MongoDB và MySQL

1.1.3.1. MongoDB

MongoDB là phần mềm cơ sở dữ liệu mã nguồn mở NoSQL hỗ trợ đa nền tảng được thiết kế theo hướng đối tượng. Các bảng trong MongoDB có cấu trúc linh hoạt cho phép dữ liệu không cần tuân theo dạng cấu trúc nào. Ở trong project này, chúng em sử dụng MongoDB cho phần tài khoản user đăng nhập.

MongoDB có thể dùng để lưu trữ dữ liệu có cấu trúc phức tạp và đa dạng. Nó lưu dữ liệu vào collection theo hướng tài liệu kiểu JSON thay vì bảng nên có hiệu suất cao và tính khả dụng cũng cao.



Hình 1.6 Sơ đồ kết nối mongoDB và Nodejs

Trong khái niệm MongoDB ta có nhắc đến NoSQL, đây là thuật ngữ viết tắt của “Not Only SQL” và đề cập đến các loại dữ liệu không quan hệ (non-relational) hoặc không dùng SQL như cách truy vấn và quản lý dữ liệu. NoSQL được phát triển để đáp ứng các yêu cầu của ứng dụng hiện nay, các website với quy mô lớn và đảm bảo tính mở rộng cao. Trong cơ sở dữ liệu quan hệ truyền thống, dữ liệu được tổ chức theo các bảng, hàng và cột, cũng như được quản lý bằng ngôn ngữ truy vấn SQL. Với NoSQL, nó giới thiệu các mô hình dữ liệu không cấu trúc hoặc bán cấu trúc khác nhau.

Một số câu lệnh cơ bản trên MongoDB:

Bảng 1-1 Một số câu lệnh cơ bản mongoDB và MySQL

CSDL	MySQL	MongoDB
Tạo csdl	CREATE DATABASE test;	use test;
Tạo bảng	CREATE TABLE students (ten_cot - kieu_du_lieu);	db.createCollection ('students');
Tạo bản ghi	INSERT INTO studetns ('name','gender') VALUES	d.sbtudents.insert({ name:'thanh',gender: 'male'});

	('thanh', 'male');	
Cập nhật	UPDATE students SET name = 'thanh update' WHERE id = 1;	db.students.update({ _id: 1 }, { \$set: { name: 'thanh update' } });
Xóa bản ghi	DELETE FROM students Where id = 1;	db.students.remove({ _id: 1 });
Tìm kiếm all	SELECT * FROM students;	db.students.find({ });
Tìm kiếm	SELECT * FROM students WHERE name = 'thanh';	db.students.find({ name: 'thanh' });

Ưu điểm của MongoDB:

Việc sử dụng MongoDB mang lại nhiều lợi ích cho người dùng.

- Tính linh hoạt lưu trữ dữ liệu theo kích cỡ khác nhau, dữ liệu lưu dưới dạng JSON nên có thể chèn vào bất cứ thông tin nào chúng ta muốn.
- Khác với RDBMS, dữ liệu trong đây không có sự ràng buộc và không có yêu cầu tuân thủ theo khuôn khổ nhất định, điều này giúp tiết kiệm thời gian hơn trong việc chèn, xóa, cập nhật hay thay đổi dữ liệu.
- Dễ dàng mở rộng hệ thống bằng cách thêm node vào cluster- cụm các node chứa dữ liệu giao tiếp với nhau.
- Tốc độ truy vấn của MongoDB nhanh hơn nhiều so với hệ quản trị cơ sở dữ liệu RDBMS do dữ liệu truy vấn được cached lên bộ nhớ RAM để lượt truy vấn sau diễn ra nhanh hơn mà không cần đọc từ ổ cứng.
- Trường dữ liệu “_id” luôn được tự động đánh địa chỉ để đạt hiệu suất tốt nhất.

Nhược điểm của MongoDB:

Ngoài các ưu điểm vượt trội, hệ quản trị dữ liệu này cũng có những mặt hạn chế riêng:

- Dữ liệu trong MongoDB không bị ràng buộc như RDBMS nhưng người sử dụng lưu ý cẩn thận mọi thao tác để không ảnh hưởng đến dữ liệu.
- Tốn nhiều bộ nhớ do dữ liệu được lưu ở dạng key-value, trong khi các collection chỉ khác nhau về value nên sẽ lặp lại key dẫn đến thừa dữ liệu.
- Thông thường, dữ liệu thay đổi từ RAM xuống ổ cứng phải qua 60s thì chương trình mới thực hiện hoàn tất, đây là nguy cơ bị mất dữ liệu nếu bất ngờ xảy ra tính huống mất nguồn trong 60s đó.

Bảng 1-2 So sánh giữa SQL DB (RDBMS) schema và MongoDB schema

SQL DB schema	Mongo DB schema
Table	Collection
Row	Document
Column	Field
Joins	Embedded documents, linking
Primary key	Primarykey (mặc định là _id)

1.1.3.2 Hệ quản trị cơ sở dữ liệu MySQL

MySQL là 1 hệ thống quản trị về cơ sở dữ liệu với mã nguồn mở (được gọi tắt là RDBMS) và đang hoạt động theo mô hình dạng client-server. Đối với RDBMS – Relational Database Management System thì MySQL đã được tích hợp apache và PHP.



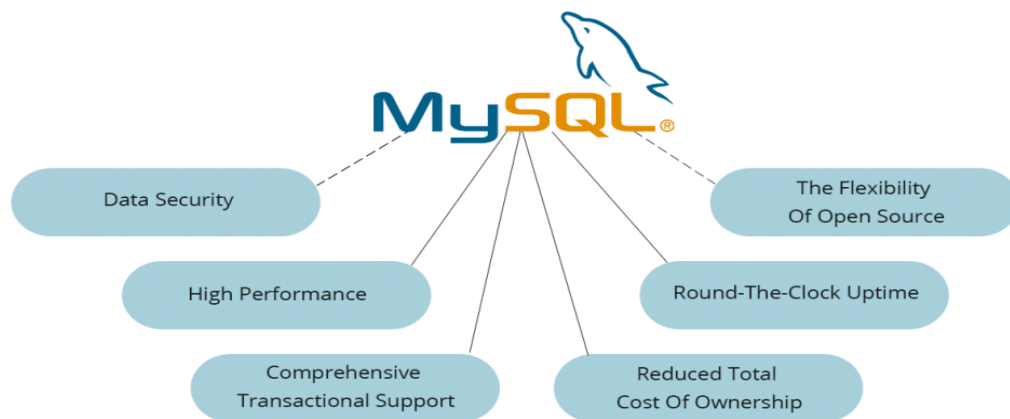
Hình 1.7 Hình ảnh MySQL

Quá trình hình thành của MySQL:

- Năm 1979: Michael Widenius, người sáng lập MySQL, bắt đầu phát triển một phiên bản đầu tiên của MySQL khi ông còn là sinh viên tại Đại học Helsinki, Phần Lan. MySQL ban đầu chỉ là một dự án cá nhân và được viết bằng ngôn ngữ C.

- Năm 1995: Michael Widenius và David Axmark thành lập công ty MySQL AB tại Thụy Điển. Họ chuyển đổi MySQL từ dự án cá nhân thành một sản phẩm phần mềm thương mại, nhưng vẫn giữ bản quyền mã nguồn mở.
- Năm 1996: MySQL 1.0 được phát hành là phiên bản đầu tiên của MySQL. Nó đã nhận được sự chú ý và sự phổ biến nhanh chóng nhờ vào tính nhỏ gọn, tốc độ xử lý nhanh và khả năng mở rộng.
- Năm 2000: Phiên bản MySQL 3.23 được phát hành với tính năng nâng cao bao gồm hỗ trợ transaction và khóa ngoại (foreign key). Điều này làm tăng khả năng sử dụng MySQL trong các ứng dụng lớn và yêu cầu tính toàn vẹn dữ liệu cao.
- Năm 2008: Sun Microsystems mua lại MySQL AB với mục tiêu mở rộng sự phát triển và sự phổ biến của MySQL.
- Năm 2010: Oracle Corporation mua lại Sun Microsystems và tiếp quản MySQL từ đó.
- Năm 2011: MySQL 5.5 được phát hành, đưa ra nhiều cải tiến về hiệu suất, bảo mật và khả năng xử lý.
- Năm 2013: MySQL 5.6 được phát hành, với những cải tiến như InnoDB storage engine được tích hợp sẵn, khả năng xử lý thời gian chờ và nâng cao hiệu suất truy vấn.
- Năm 2015: MySQL 5.7 ra mắt, mang đến nhiều cải tiến về hiệu suất, bảo mật, và tính năng mới như JSON support và nhóm ký tự mở rộng.
- Hiện tại, MySQL tiếp tục phát triển với các phiên bản mới và có một cộng đồng rộng lớn, được hỗ trợ và đóng góp từ cộng đồng người dùng trên toàn thế giới.

Qua quá trình phát triển, MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các ứng dụng web, ứng dụng di động và hệ thống quản lý dữ liệu.



Hình 1.8 Một số ưu điểm của MySQL

Có nhiều lợi ích khi sử dụng MySQL như một hệ quản trị cơ sở dữ liệu. Dưới đây là một số lý do nên sử dụng MySQL:

- Hiệu suất và tốc độ: MySQL được tối ưu hóa để cung cấp hiệu suất cao và tốc độ xử lý nhanh. Nó có thể xử lý hàng ngàn truy vấn mỗi giây và hỗ trợ khả năng mở rộng theo quy mô ngang, cho phép bạn tăng cường khả năng xử lý bằng cách thêm máy chủ.
- Tin cậy và ổn định: MySQL đã được sử dụng và phát triển trong nhiều năm, với một cộng đồng lớn và đội ngũ phát triển chuyên nghiệp. Nó đã được kiểm tra và chứng minh tính ổn định và đáng tin cậy trong môi trường sản xuất.
- Dễ sử dụng và quản lý: MySQL có giao diện dễ sử dụng và cung cấp các công cụ quản lý như phpMyAdmin để quản lý cơ sở dữ liệu. Nó có một ngôn ngữ truy vấn đơn giản (SQL) và hỗ trợ các tính năng quản lý dữ liệu phổ biến như backup, khôi phục và sao lưu dữ liệu.
- Đa nền tảng: MySQL có sẵn cho nhiều hệ điều hành và nền tảng khác nhau, bao gồm Windows, macOS và các bản phân phối Linux. Điều này giúp tăng tính di động và linh hoạt trong việc triển khai và phát triển ứng dụng.
- Hỗ trợ chuẩn SQL: MySQL tuân thủ chuẩn SQL, cho phép bạn sử dụng các truy vấn và câu lệnh SQL phổ biến. Điều này giúp dễ dàng chuyển đổi giữa các hệ quản trị cơ sở dữ liệu khác hoặc tích hợp với các công cụ và thư viện hỗ trợ chuẩn SQL.

Tóm lại, MySQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ, ổn định và dễ sử dụng, được sử dụng rộng rãi trong các ứng dụng web và hệ thống quản lý dữ liệu. Nó cung cấp hiệu suất cao, tính tin cậy, và hỗ trợ chuẩn SQL, đồng thời có một cộng đồng lớn và tài liệu phong phú để hỗ trợ việc phát triển và quản lý cơ sở dữ liệu.

CHƯƠNG 2: TỔNG QUAN HỆ THỐNG

2.1 Tổng quan đề tài

2.1.1 Lý do chọn đề tài

Trong thời đại ngày nay, việc ứng dụng các công nghệ vào trong thi cử là điều đang được thực hiện một cách rộng rãi. Quy trình thi diễn ra từ việc tập hợp dữ liệu các lớp cần thi, hình thức và thời gian thi, đề thi sau đó đưa ra lịch thi, tổ chức thi, tổ chức chấm điểm đánh giá và trả về kết quả. Đó là một quá trình đòi hỏi sự chính xác và hiệu quả cao để đánh giá một cách công bằng năng lực của sinh viên. Ở mỗi giai đoạn trong công tác này đều có các công cụ hỗ trợ để đạt hiệu quả tốt nhất.

Trong phạm vi đề tài này, chúng em tập trung vào việc đưa ra lịch thi cho sinh viên, từ các dữ liệu đang có về sinh viên, lớp học, học phần và các đề xuất thi của giáo viên có thể tạo ra lịch thi để sinh viên có thể tham khảo sớm nhất có thể. Phần mềm quản lý lịch thi là một công cụ vô cùng cần thiết và hữu ích trong môi trường học tập hiện đại. Phần mềm hứa hẹn mang lại trải nghiệm tốt hơn cho người dùng cũng như tiết kiệm thời gian hơn cho quá trình xuất ra lịch thi.

2.1.2 Phạm vi của đề tài

Đề tài được xây dựng cho trường đại học Bách Khoa, cũng như các trường đại học có quá trình lên lịch thi đang là thủ công.

Đối tượng sử dụng gồm có Giáo viên, Sinh viên và các cán bộ quản lý như giáo vụ.

Đề tài tập trung chủ yếu vào khâu lên lịch thi, một trong những giai đoạn quan trọng của quá trình tổ chức thi và trả kết quả.

2.1.3 Mục tiêu của đề tài

Mục tiêu của hệ thống quản lý lịch thi là tạo ra một giải pháp hiệu quả và tự động quy trình lập lịch thi, nhằm đảm bảo tính chính xác, minh bạch và công bằng trong công tác tổ chức thi cho sinh viên và giảng viên.

Đảm bảo tối ưu hóa lịch thi, tránh sự xung đột và trùng lặp, giảm thiểu tối đa thời gian chờ đợi và tăng cường hiệu suất sử dụng tài nguyên học tập.

Hệ thống cung cấp giao diện thân thiện và dễ sử dụng cho cả sinh viên và giáo viên, giúp họ dễ dàng truy cập và hiểu thông tin về lịch thi một cách nhanh chóng.

2.1.4 Đề xuất giải pháp

Hệ thống quản lý lịch thi là cần thiết vì nó mang lại nhiều lợi ích và giải quyết vấn đề quan trọng trong quá trình tổ chức thi.

Giải pháp giúp hạn chế sử dụng đến giấy tờ, giảm thiểu việc in ấn, bảo vệ môi trường và tiết kiệm chi phí cho doanh nghiệp cũng như tiết kiệm thời gian và công sức hơn.

Cải thiện hiệu quả lập lịch, dù giai đoạn vẫn còn chưa được tự động hoàn toàn nhưng việc số hóa các thông tin và đơn giản hóa các thao tác liên quan đến thiết lập lịch thi giúp cho giáo vụ dễ dàng truy cập và tìm kiếm thông tin cần thiết.

Trong quá trình lên lịch, nếu có 2 thời khóa biểu thi trùng lịch nhau sẽ được hệ thống cảnh báo ngay, điều này giúp giảm thiểu sai sót và tránh xung đột lịch. Khi nhận được thông báo trùng lịch, người quản lý có thể kịp thời giải quyết một cách nhanh chóng.

hiệu quả. Các thông tin cần thiết về Lịch thi, Môn học, Giảng viên và Sinh viên được tổ chức và duy trì một cách chặt chẽ

Hệ thống sẽ bao gồm hai thành phần cơ bản là UI (giao diện người dùng): Phần này sẽ sử dụng thư viện ReactJS cho việc phát triển frontend bên cạnh các thành phần cơ Hệ thống này cũng cho phép dễ dàng quản lý dữ liệu liên quan đến lịch thi một các bản như HTML, CSS sử dụng các thư viện hỗ trợ như Material UI hỗ trợ tại giao diện thân thiện người dùng. Thành phần backend (server side) sẽ sử dụng NodeJS-một môi trường phát triển backend sử dụng javascript cùng với thư viện Express và Axios cho việc giao tiếp với frontend. Database chúng em sử dụng MySQL cho lưu trữ thông tin sinh viên, giáo viên, ... và dùng MongoDB cho việc đăng nhập.

2.2 Phân tích tính khả thi

2.2.1 Khả thi về mặt kỹ thuật

Các công nghệ sử dụng:

- Ngôn ngữ: HTML/CSS, JavaScript
- Nền tảng: NodeJs
- Thư viện: ReactJS (frontend) và Express (backend)
- Database: MongoDB, MySQL
- Mô hình phần mềm: MVC (Model-View-Controller)

Ứng dụng của hệ thống:

- Triển khai nền tảng là một website, truy cập được từ mọi nơi qua trình duyệt với kết nối internet.

- Giao diện dùng các công nghệ hiện đại, đẹp mắt và dễ sử dụng.
- Đảm bảo được tính bảo mật thông tin

2.2.2 Khả thi về mặt tổ chức

Hệ thống dễ sử dụng và cũng dễ vận hành, với đội ngũ nhân lực hiện tại có thể nhanh chóng làm quen sử dụng chỉ sau một thời gian training ngắn.

Thời gian để cài đặt và triển khai hệ thống ước tính phù hợp với các đối tượng hướng đến ở trên.

Hệ thống có nghiệp vụ không thay đổi nhiều so với nghiệp vụ truyền thống nên tính tương thích với quy trình hiện tại không phải là vấn đề lớn.

CHƯƠNG 3: THU THẬP YÊU CẦU

3.1 Khảo sát thực trạng

3.1.1 Cơ cấu tổ chức của hệ thống

Xem xét về cơ cấu tổ chức của các thành phần tham gia trong việc tạo ra lịch thi, đây là phần quan trọng của quá trình quản lý thi cử:

Về giáo vụ:

- Giáo vụ là bộ phận chịu trách nhiệm tổ chức và xây dựng lịch thi cho các kỳ thi và đưa nó ra cho sinh viên tham khảo.
- Các nhân viên trong giáo vụ phải phối hợp với giáo viên và các bộ phận khác trong trường để thu thập thông tin về môn học, giảng viên, sinh viên, định lượng số lượng sinh viên tham gia thi và các yêu cầu đặc biệt khác.
- Giáo vụ phải đảm bảo rằng lịch thi được xây dựng ra sao cho không có sự xung đột về thời gian giữa các kỳ thi, đồng thời cân nhắc sự thoải mái và cân bằng cho sinh viên tham gia thi.

Về ban giám hiệu và lãnh đạo trường học:

- Ban giám hiệu và lãnh đạo trường có trách nhiệm giám sát và duyệt các kế hoạch và quyết định liên quan đến lịch thi.
- Họ đảm bảo rằng quy trình tạo lịch thi đáp ứng các tiêu chuẩn và yêu cầu của trường học, đảm bảo tính minh bạch.

Về giáo viên giảng dạy:

- Giáo viên cung cấp thông tin về kế hoạch giảng dạy và yêu cầu về thi cử cho bộ phận giáo vụ.
- Họ cũng tham gia và xây dựng đề thi, thảo luận và đề xuất các nội dung thi phù hợp với môn học.

Về sinh viên:

- Sinh viên thường chỉ có nhiệm vụ là xem lịch thi và tham gia theo đúng lịch thi theo quy định sẵn của từng học phần.

- Một số trường hợp sinh viên có thể phản ánh lên giáo vụ một vài vấn đề liên quan lịch thi để giáo vụ có thể cơ cấu lại lịch thi cho phù hợp với yêu cầu của đa số sinh viên

3.1.2 Quy trình nghiệp vụ của hệ thống hiện tại

Chu trình nghiệp vụ để tạo ra lịch thi là quá trình thực hiện các bước và hoạt động từ khi bắt đầu lập lịch cho đến khi công bố lịch thi chính thức.

- Thu thập thông tin: đầu tiên sẽ phải thu thập thông tin liên quan đến môn học, giảng viên và số lượng sinh viên tham gia, yêu cầu về thi cử cũng như các yêu cầu đặc biệt khác. Thông tin này cần được thu thập từ các bộ phận liên quan trong trường, giáo viên và các hệ thống thông tin liên quan.
- Xây dựng kế hoạch lịch thi: Dựa vào thông tin thu thập được, giáo vụ và các bộ phận liên quan xây dựng kế hoạch lịch thi với các thông tin cụ thể như thời gian và địa điểm tổ chức thi, mã môn học, số lượng sinh viên tham gia, đề thi và phương thức thi ...
- Xác nhận và xét duyệt: Kế hoạch lịch thi cần được xác nhận và duyệt bởi các bộ phận chịu trách nhiệm như Ban giám hiệu, Phòng đào tạo. Các sửa đổi và điều chỉnh sẽ được thực hiện nếu như cần thiết.
- Xây dựng lịch thi: Lịch thi chính thức được xây dựng dựa trên kế hoạch đã được xác nhận. Các yếu tố quan trọng như thời gian diễn ra môn thi, số lượng sinh viên trong mỗi phòng thi, sự trùng lịch giữa các môn học được cân nhắc.
- Kiểm tra và duyệt lịch thi: Lịch thi được kiểm tra và duyệt bởi các bộ phận có thẩm quyền để đảm bảo tính chính xác và công bằng.
- Công bố lịch thi: sau khi đã được xét duyệt, lịch thi sẽ được công bố chính thức để thông báo cho sinh viên về thời gian và địa điểm tổ chức thi. Lịch thi được công bố một các minh bạch và dễ truy cập cho sinh viên.
- Nếu lịch thi có vấn đề xảy ra hoặc nhận được khiếu nại từ sinh viên, có thể quay lại điều chỉnh thêm bớt trong lịch thi, thực hiện kiểm duyệt lại và công bố phiên bản mới nhất.

3.1.3 Đề xuất phương án

Những vấn đề gặp phải của quy trình truyền thống:

- Mất nhiều thời gian ở khâu thu thập các thông tin cần thiết cho việc lên lịch thi.

- Trong quá trình tạo lịch thi nếu không có các cảnh báo kịp thời sẽ dẫn đến việc trùng lặp, nếu lỡ công bố lịch thi với sai sót sẽ phải tốn công sức quay lại quy trình từ phía trên.
- Lịch thi chỉ được công bố khi được cho là hoàn thành tại thời điểm được xuất, sinh viên phải đợi trên cổng thông tin (vd: ctt.hust.edu.vn) và thực hiện tải các bản cập nhật mới nhất dưới dạng file excel về và tìm kiếm môn mình cần thi.

Phương án xây dựng hệ thống mới:

- Hệ thống Website quản lý lịch thi với các thông tin về sinh viên, mã học phần, giáo viên và lịch thi được lưu sẵn trong hệ thống, việc truy xuất thông tin để lên lịch thi trở nên đơn giản. VD: trong quá trình học có một số sinh viên nghỉ học, các sinh viên này sẽ bị xóa trong database hệ thống trước đó nên không làm ảnh hưởng đến quá trình lên lịch thi.
- Phần mềm cho phép cảnh báo nếu các lịch thi trong bảng lịch thi bị trùng thì sẽ không được thêm lớp thi đã trùng với lớp cũ vào, giúp quá trình lên lịch được chính xác hơn.
- Phần mềm cho phép giáo viên nhanh chóng đưa ra các đề xuất về hình thức và thời gian thi cho giáo vụ thông qua tài khoản giáo viên được cấp.
- Sinh viên có thể theo dõi lịch thi ngay trên hệ thống nhờ tài khoản của mình, có thể thực hiện tải xuống file excel nếu muốn, và lịch được thiết lập đến đâu sinh viên có thể theo dõi đến đấy, giúp ước lượng thời gian ôn luyện tốt hơn thay vì phải đợi xong lịch hoàn toàn mới biết.

3.2 Các yêu cầu của hệ thống

3.2.1 Yêu cầu chức năng

Đối với giáo vụ (admin):

- Giáo vụ được quyền quản lý các tài khoản khác, thêm hoặc sửa các tài khoản khác trong hệ thống
- Thay đổi các thông tin về các thành phần khác như sinh viên, giáo viên, học phần...
- Giáo vụ cũng là người duyệt các đề xuất của giáo viên và tạo lịch thi
- Truy cập tài khoản admin có thể xem được thống kê các thông tin về các đối tượng khác trong hệ thống thông qua dashboard admin.
- Chỉnh sửa các thông tin về lịch thi nếu cần thiết: thời gian, địa điểm thi, hình thức thi ...

Đối với giáo viên:

- Giáo viên được cấp tài khoản truy cập từ giáo vụ
- Có thể thay đổi mật khẩu của tài khoản mình được cấp theo ý muốn.
- Có nhiệm vụ tạo các đề xuất về hình thức thi, thời gian thi cho các môn để giáo vụ có thể duyệt.
- Xem thông tin về lịch thi

Đối với sinh viên:

- Sinh viên được giáo vụ cấp cho tài khoản đăng nhập.
- Đăng nhập dưới quyền sinh viên và xem trạng thái lịch thi đang được xây dựng.
- Có thể tải về file lịch thi nếu muốn dưới dạng excel.
- Tìm kiếm lớp thi trực tuyến bằng chức năng search theo mã lớp thi hoặc mã môn học.

3.2.2 Yêu cầu phi chức năng

Hiệu năng của hệ thống:

- Hệ thống hoạt động với độ ổn định cao.
- Các thao tác trên Website phải mượt mà, nhanh chóng.
- Truy xuất dữ liệu đến database nhanh.

Giao diện hệ thống:

- Giao diện đẹp, thân thiện với người dùng.
- Bố trí đơn giản, dễ sử dụng, dễ thao tác.
- Hệ thống sử dụng ngôn ngữ là Tiếng Việt.

Tính bảo mật:

- Các tài khoản được phân quyền nhất định, không làm xung đột đến nhau.
- Chỉ có tài khoản admin mới được sửa dữ liệu trong database, mật khẩu admin lưu trong DB được mã hóa.

3.2.3 Yêu cầu về dữ liệu

Biểu diễn yêu cầu về dữ liệu dưới dạng bảng từ điển dữ liệu:

Bảng 3-1 Bảng từ điển dữ liệu

Tên dữ liệu	Mô tả
Giáo viên	Mã GV, Tên giáo viên, Ngày sinh, Giới tính, Địa chỉ, Viện, Số điện thoại
Giáo vụ	Mã Giáo vụ, Tên giáo vụ, Ngày sinh, Giới tính, Địa chỉ, Số điện thoại
Sinh viên	MSSV, Tên Sinh viên, Ngày Sinh, Địa chỉ, Lớp, Viện
Học phần	Mã học phần, Tên học phần
Lớp	Mã lớp, Tên lớp, Giáo viên phụ trách, Mã học phần
Lịch thi	Mã Lịch thi, Hình thức thi, Thời gian thi, Ngày thi, Kỳ học, Mã lớp thi, Mã học phần

CHƯƠNG 4: PHÂN TÍCH HỆ THỐNG

Trong chương 4 này, hệ thống sẽ được phân tích một cách chi tiết qua sơ đồ chức năng, các lược đồ usecase cũng như các biểu đồ hoạt động và sơ đồ lớp của UML

4.1 Xây dựng kịch bản sử dụng

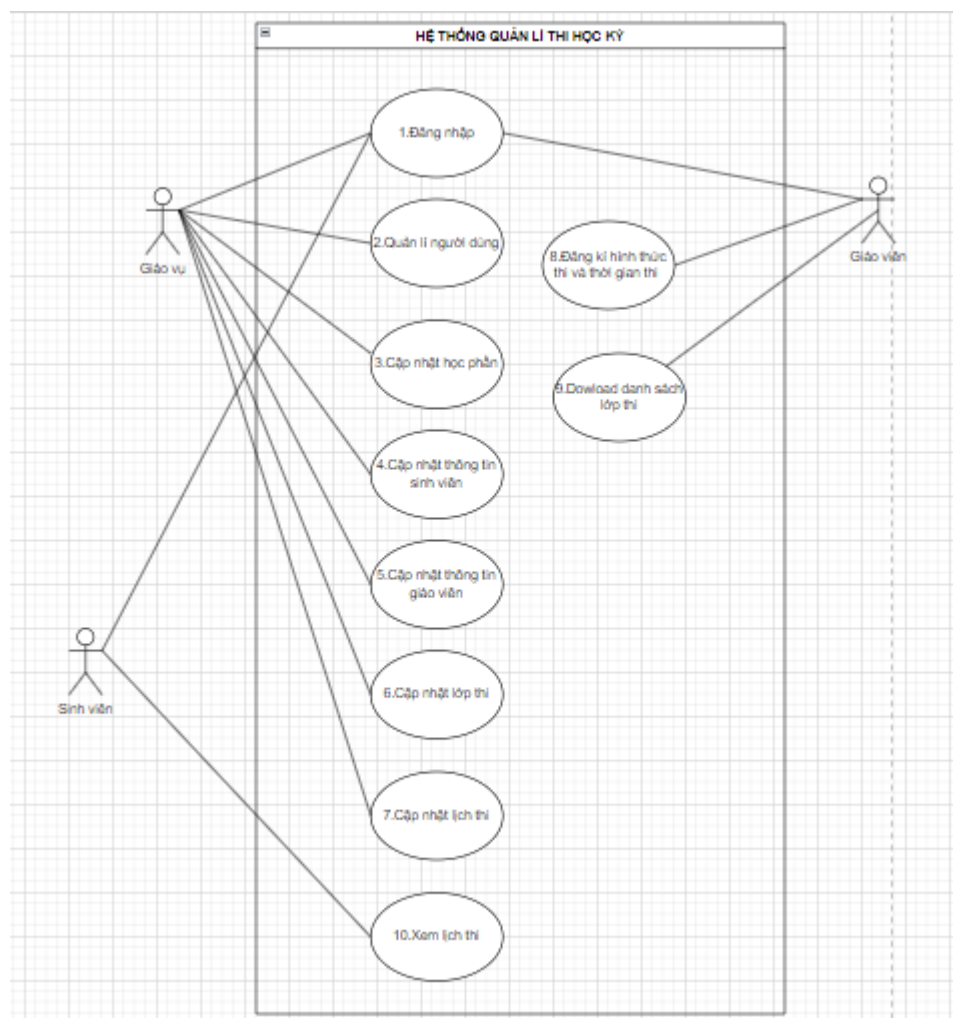
4.1.1 Sơ đồ Usecase

Ngôn ngữ mô hình hóa thống nhất UML (Unified Modeling Language) là một ngôn ngữ mô hình gồm các ký hiệu đồ họa mà các phương pháp hướng đối tượng sử dụng để thiết kế các hệ thống thông tin một cách nhanh chóng.

Cách xây dựng các mô hình trong UML phù hợp mô tả các hệ thống thông tin cả về cấu trúc cũng như hoạt động. Cách tiếp cận theo mô hình của UML giúp ích rất nhiều cho những người thiết kế và thực hiện hệ thống thông tin cũng như những người sử dụng nó; tạo nên một cái nhìn bao quát và đầy đủ về hệ thống thông tin dự định xây dựng. Cách nhìn bao quát này giúp nắm bắt trọn vẹn các yêu cầu của người dùng, phục vụ từ giai đoạn phân tích đến việc thiết kế, thẩm định và kiểm tra sản phẩm ứng dụng công nghệ thông tin. Các mô hình hướng đối tượng được lập cũng là cơ sở cho việc ứng dụng các chương trình tự động sinh mã trong các ngôn ngữ lập trình hướng đối tượng, chẳng hạn như ngôn ngữ C++, Java, ... Phương pháp mô hình này rất hữu dụng trong lập trình hướng đối tượng. Các mô hình được sử dụng bao gồm mô hình đối tượng (mô hình tĩnh) và mô hình động.

UML sử dụng một hệ thống ký hiệu thống nhất biểu diễn các phần tử mô hình (model elements). Tập hợp các phần tử mô hình tạo thành các sơ đồ UML (UML diagrams). Có nhiều loại sơ đồ UML như: sơ đồ lớp (Class Diagram), sơ đồ đối tượng (Object Diagram), sơ đồ tình huống sử dụng (Use case Diagram), sơ đồ trình tự (Sequence Diagram), ...

Đồ án này sẽ trình bày về sơ đồ Use case , Class Diagram và Sequence Diagram.



Hình 4.1 Sơ đồ Usecase

Bảng 4-1 Mô tả usecase đăng nhập ở hình 4.1

Tên Usecase: Đăng nhập	ID: 1
Chức năng	Đăng nhập
Tác nhân	Giáo viên, giáo vụ và sinh viên
Điều kiện trước	<ul style="list-style-type: none"> - Các người dùng đang ở trạng thái chưa đăng nhập - Giáo vụ sẽ được cung cấp tài khoản trước đó - Giáo viên và sinh viên sẽ được giáo vụ cấp tài khoản.

Điều kiện sau	
Mô tả	Người dùng đăng nhập vào trang Web để sử dụng
Kịch bản	<ol style="list-style-type: none"> 1. Vào trang Web đăng nhập 2. Nhập thông tin 3. Gửi yêu cầu đăng nhập

Bảng 4-2 Bảng mô tả chức năng quản lí người dùng ở hình 4.1

Tên Usecase: Quản lí người dùng	ID: 2
Chức năng	Quản lí thông tin người dùng
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ có thể tạo, chỉnh sửa và xóa người dùng trong hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật thông tin người dùng 3. Thực hiện các chức năng tạo, sửa, xóa người dùng.

Bảng 4-3 Bảng mô tả chức năng quản lí học phần ở hình 4.1

Tên Usecase: Cập nhật học phần	ID: 3
Chức năng	Quản lí học phần
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ tạo, chỉnh sửa và xóa các học phần trong hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật danh sách học phần 3. Thực hiện các chức năng tạo, sửa, xóa

	học phần.
--	-----------

Bảng 4-4 Bảng mô tả chức năng quản lí sinh viên ở hình 4.1

Tên Usecase: Cập nhật thông tin sinh viên	ID: 4
Chức năng	Quản lí sinh viên
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ tạo, chỉnh sửa và xóa thông tin sinh viên hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật thông tin sinh viên 3. Thực hiện các chức năng tạo, sửa, xóa thông tin.

Bảng 4-5 Bảng mô tả chức năng quản lí giáo viên ở hình 4.1

Tên Usecase: Cập nhật thông tin giáo viên	ID: 5
Chức năng	Quản lí giáo viên
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ tạo, chỉnh sửa và xóa thông tin giáo viên hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật thông tin giáo viên 3. Thực hiện các chức năng tạo, sửa, xóa thông tin.

Bảng 4-6 Bảng mô tả chức năng quản lí lớp thi ở hình 4.1

Tên Usecase: Cập nhật lớp thi	ID: 6
Chức năng	Quản lí lớp thi
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ tạo, chỉnh sửa và xóa thông tin lớp thi trong hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật danh sách lớp thi 3. Thực hiện các chức năng tạo, sửa, xóa thông tin.

Bảng 4-7 Bảng mô tả chức năng quản lí lịch thi ở hình 4.1

Tên Usecase: Cập nhật lịch thi	ID: 7
Chức năng	Cập nhật lịch thi
Tác nhân	Giáo vụ
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo vụ tạo, chỉnh sửa và xóa thông tin lịch thi trong hệ thống.
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật danh sách lịch thi 3. Thực hiện các chức năng tạo, sửa, xóa thông tin.

Bảng 4-8 Bảng mô tả đăng kí hình thức thi

Tên Usecase: Đăng kí hình thức thi và thời gian thi	ID: 8
--	--------------

Chức năng	Đăng kí hình thức thi
Tác nhân	Giáo viên
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo viên có thể tạo, sửa hình thức và thời gian thi từng môn
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục cập nhật danh sách lịch thi 3. Tìm kiếm lịch thi dựa vào ID lịch thi 4. Thực hiện các chức năng tạo, sửa thông tin.

Bảng 4-9 Bảng mô tả hình thức Download danh sách thi

Tên Usecase: Download danh sách thi	ID: 9
Chức năng	Download danh sách lớp
Tác nhân	Giáo viên
Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Giáo viên có thể download danh sách sinh viên có trong lớp thi
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục danh sách lớp thi 3. Tìm kiếm lớp thi dựa vào mã lớp thi 4. Thực hiện các chức năng download file excel danh sách lớp thi đã tìm kiếm.

Bảng 4-10 Bảng mô tả hình thức xem lịch thi

Tên Usecase: Xem lịch thi	ID: 10
Chức năng	Xem lịch thi
Tác nhân	Sinh viên

Điều kiện trước	Đã đăng nhập thành công
Điều kiện sau	
Mô tả	Sinh viên có thể xem danh sách lịch thi trên hệ thống
Kịch bản	<ol style="list-style-type: none"> 1. Đăng nhập vào hệ thống 2. Chọn mục xem lịch thi 3. Tìm kiếm lịch thi của mình theo MSSV 4. Thực hiện các chức năng download file excel danh sách lịch thi đã tìm kiếm.

4.1.2 Mô hình hóa cấu trúc

3.1.2.1 Thẻ CRC

Sau khi đã liệt kê, khoanh vùng được chức năng của hệ thống ta tiến hành xác định các thẻ CRC để bước đầu vạch ra các lớp sử dụng trong hệ thống.

Bảng 4-11 Thẻ CRC cho lớp Person

Class name: Tài khoản	ID: 1
Use Case: Đăng nhập	
Description: là tài khoản để đăng nhập hệ thống	
Responsibilities: - Đăng nhập	Collaborators: - Giáo viên, sinh viên, giáo vụ
Attributes: - Username (String) - Password (String) - Level (String)	
Relationship: - Generalization: - Aggregation : - Other Associations: Sinh viên, giáo viên, Giáo vụ	

Bảng 4-12 Thẻ CRC cho lớp Giáo vụ

Class name: Giáo vụ	ID: 2
Use case: Quản lí người dùng	
Description: Là người quản lí sinh viên và giáo viên	

Responsibilities: <ul style="list-style-type: none"> - Cập nhật lớp thi - Cập nhật học phần - Cập nhật lịch thi - Quản lí người dùng - Cập nhật thông tin sinh viên - Cập nhật thông tin giáo viên 	Collaborators: <ul style="list-style-type: none"> - Lớp thi - Học phần - Lịch thi - Tài khoản - Sinh viên - Giáo viên
Attributes: <ul style="list-style-type: none"> - Tên (String) - ID (Int) 	
Relationship: <ul style="list-style-type: none"> - Generalization: tài khoản - Aggregation : - Other Associations: Sinh viên, giáo viên, học phần, lớp thi, lịch thi 	

Bảng 4-13 Thẻ CRC cho lớp Giáo viên

Class name: Giáo viên	ID: 3
Use case: Cập nhật thông tin giáo viên	
Description:	
Responsibilities: <ul style="list-style-type: none"> - Cập nhật hình thức thi, thời gian thi - Download danh sách thi 	Collaborators: <ul style="list-style-type: none"> - Lịch thi - Lớp thi
Attributes: <ul style="list-style-type: none"> - ID (Int) - Họ tên (String) - Viện (String) - Ngày sinh (Date) - SĐT (Int) - Địa chỉ (String) - Giới tính (String) 	
Relationship: <ul style="list-style-type: none"> - Generalization: Tài khoản - Aggregation : - Other Associations: Danh sách thi, lịch thi 	

Bảng 4-14 Thẻ CRC cho lớp Sinh viên

Class name: Sinh viên	ID: 4
Use case: Cập nhật thông tin sinh viên	
Description:	
Responsibilities: - Xem lịch thi	Collaborators: - Lịch thi
Attributes: - Họ tên (String) - Ngày sinh (Date) - MSSV (Int) - Giới tính (String) - Lớp (String) - Địa chỉ (String) - Viện (String)	
Relationship: - Generalization: Tài khoản - Aggregation : Lịch thi - Other Associations:	

Bảng 4-15 Thẻ CRC cho lớp Học phần

Class name: Học phần	ID: 5
Use case: Cập nhật học phần	
Description:	
Responsibilities:	Collaborators:
Attributes: - Tên học phần (String) - Mã học phần (Int) - Yêu cầu TN (Boolean) - Viện (String)	
Relationship: - Generalization: - Aggregation : - Other Associations: Lớp thi	

Bảng 4-16 Thẻ CRC cho lớp Lớp thi

Class name: Lớp thi	ID: 6
Use case: Cập nhật lớp thi, download danh sách lớp thi	
Description:	
Responsibilities:	Collaborators:
Attributes: - Tên lớp (Int) - Mã lớp (String) - Mã học phần (Int) - ID Giáo viên (Int)	
Relationship: - Generalization: - Aggregation : - Other Associations: Giáo viên	

Bảng 4-17 Thẻ CRC cho lớp Lịch thi

Class name: Lịch thi	ID: 7
Use case: Cập nhật lịch thi, xem lịch thi, đăng kí hình thức thi và thời gian thi	
Description:	
Responsibilities:	Collaborators:
Attributes: - Mã lịch thi (String) - Ngày thi (Date) - Giờ thi (String) - Hình thức thi (String) - Mã HP (String) - Học kì (Int)	
Relationship: - Generalization: - Aggregation : Học phần, Lớp thi - Other Associations:	

4.1.2.2 Bảng ánh xạ giữa sơ đồ lớp và Use case

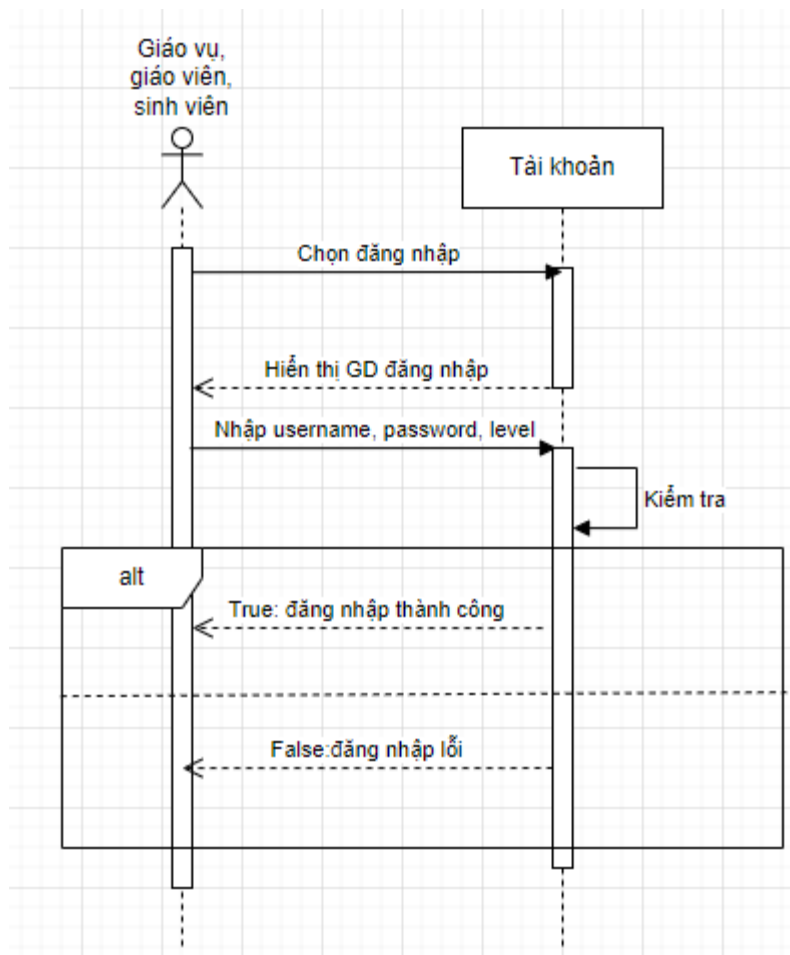
Với nghiệp vụ hệ thống đã được chúng ta phân tích ở phần trên, ta xây dựng được các lớp chịu trách nhiệm cho từng chức năng và sự hợp tác giữa các lớp được thể hiện trong Bảng 2.16 dưới đây.

Bảng 4-18 Bảng ánh xạ giữa các chức năng và thẻ CRC

Thẻ CRC Use case	Tài khoản	Giáo vụ	Sinh viên	Giáo viên	Học phần	Lớp thi	Lịch thi
Đăng nhập	x						
Quản lí người dùng		X					
Cập nhật học phần					x		
Cập nhật thông tin sinh viên			x				
Cập nhật thông tin giáo viên				x			
Cập nhật lớp thi						x	
Cập nhật lịch thi							x
Đăng kí hình thức thi và thời gian thi							x
Download danh sách thi						x	
Xem lịch thi							x

4.1.2.3 Sơ đồ lớp

Dựa vào các thẻ CRC được xây dựng, ta xây dựng được sơ đồ lớp cho hệ thống quản lý thi học kì như trên Hình 2.6 dưới đây.

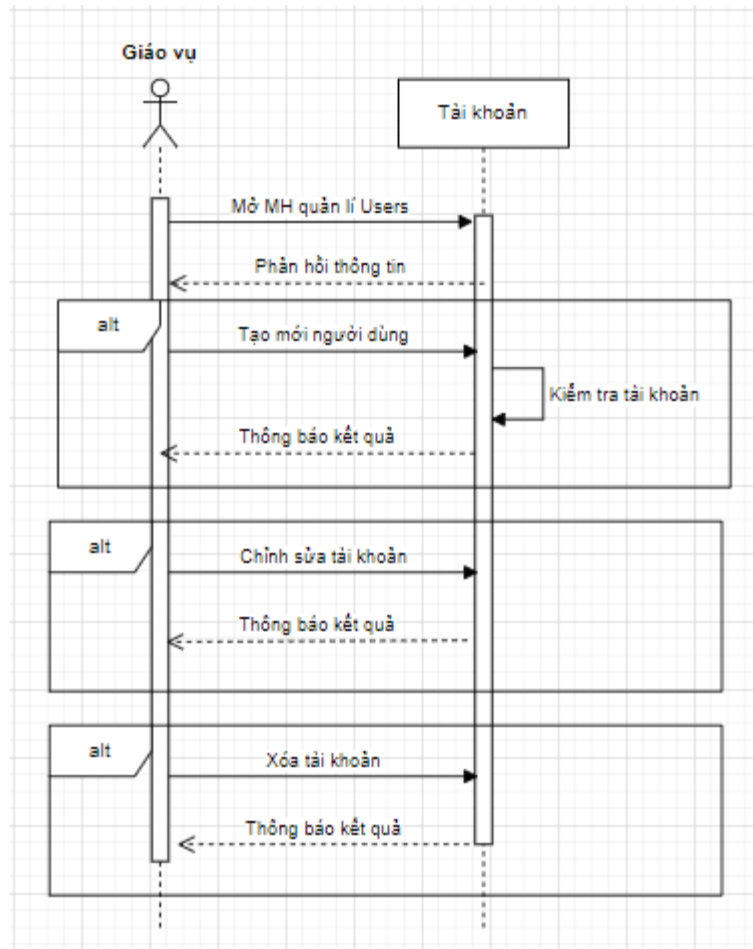


Hình 4.3 Sơ đồ Sequence đăng nhập

Sơ đồ Sequence đăng nhập được diễn giải như sau:

1. Người dùng chọn vào màn hình đăng nhập
2. Hệ thống hiển thị màn hình đăng nhập
3. Người dùng đăng nhập bằng Username, Password và Level
4. Hệ thống sẽ kiểm tra yêu cầu đăng nhập
5. Hệ thống trả về kết quả cho người dùng

4.1.3.3 Sơ đồ Sequence quản lí người dùng

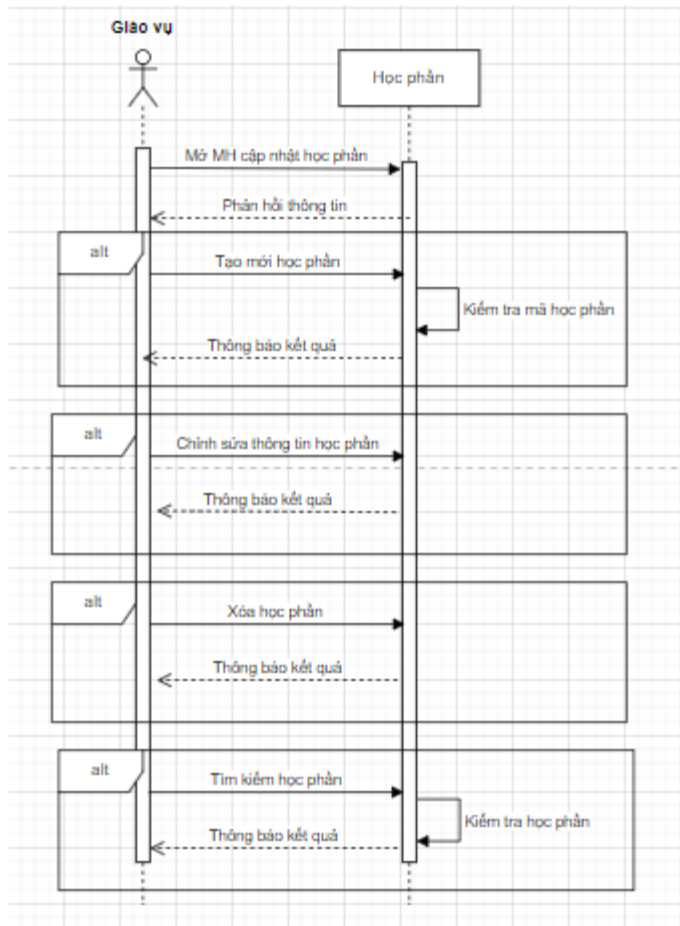


Hình 4.4 Sơ đồ Sequence quản lý người dùng

Sơ đồ Sequence quản lý User được diễn giải như sau:

- Giáo vụ có quyền mở màn hình quản lý người dùng và thực hiện các chức năng CRUD tài khoản người dùng.

4.1.3.4 Sơ đồ Sequence quản lý học phần



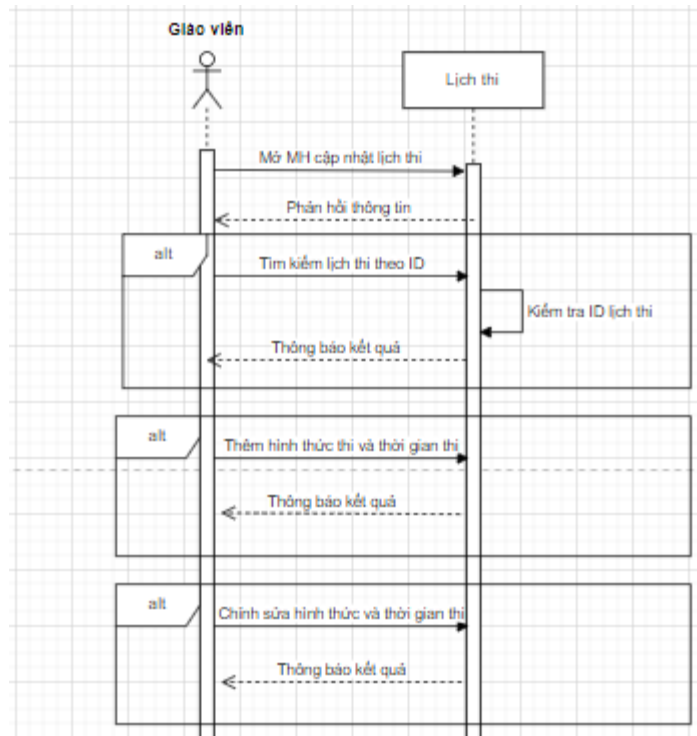
Hình 4.5 Sơ đồ Sequence cập nhật học phần

Sơ đồ Sequence cập nhật học phần được diễn giải như sau:

- Giáo vụ có quyền truy cập vào màn hình cập nhật học phần và xử dụng các chức năng CRUD và tìm kiếm học phần

+ Do các chức năng quản lý sinh viên, giáo viên, lớp thi, lịch thi đều giống chức năng quản lý học phần nên sơ đồ Sequence sẽ tương tự như trên.

4.1.3.5 Sơ đồ Sequence cập nhật thời gian và hình thức thi

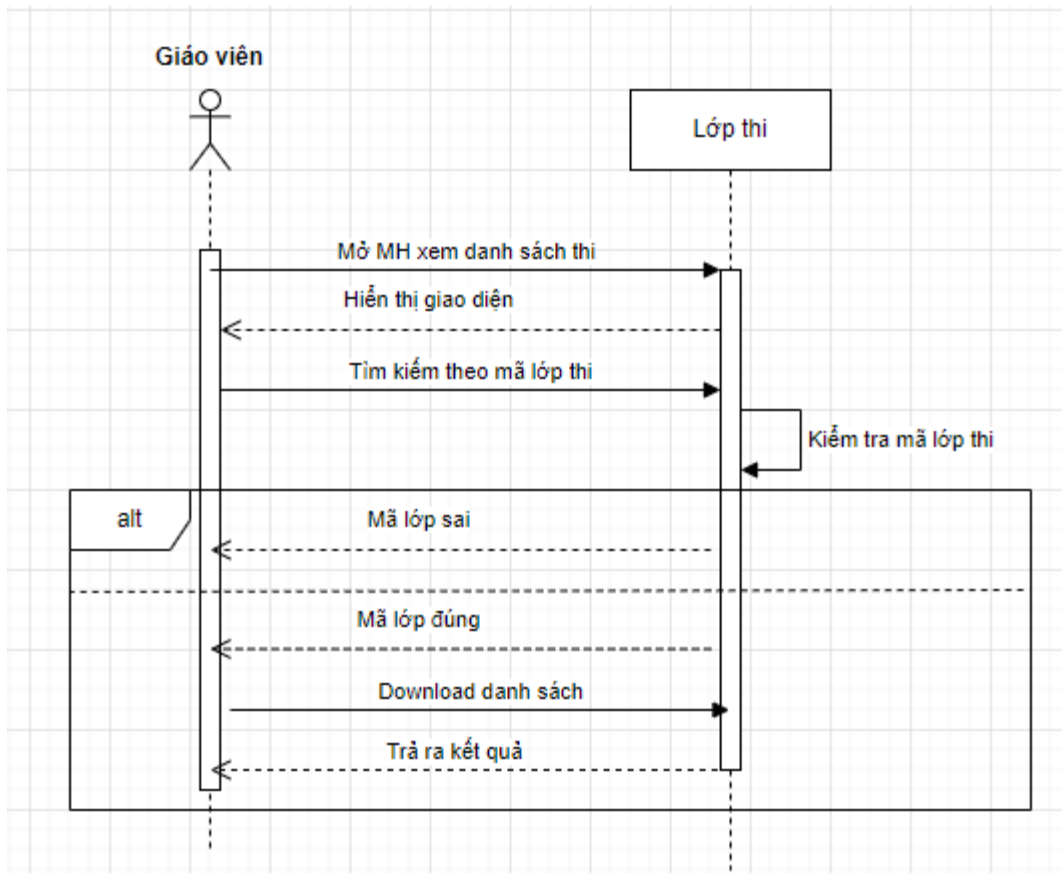


Hình 4.6 Sơ đồ Sequence cập nhật thời gian và hình thức thi

Sơ đồ Sequence cập nhật thời gian và hình thức thi được diễn giải như sau:

- Giáo viên truy cập vào màn hình cập nhật lịch thi ,có thể xử dụng các chức năng tìm kiếm lịch thi, tạo, chỉnh sửa hình thức và thời gian thi

4.1.3.6 Sơ đồ Sequence Download danh sách thi

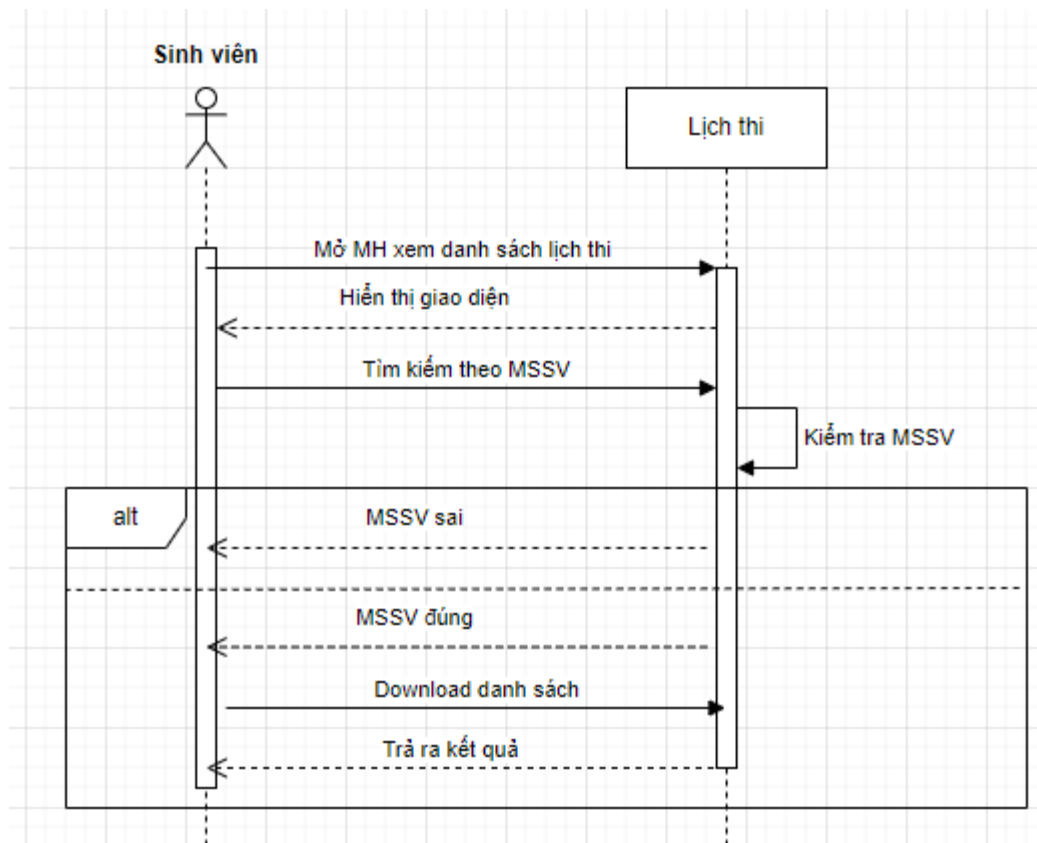


Hình 4.7 Sơ đồ Sequence download danh sách thi

Sơ đồ Sequence download danh sách thi được diễn giải như sau:

1. Giáo viên chọn vào màn hình lớp thi
2. Hệ thống hiển thị màn hình
3. Giáo viên tìm kiếm theo mã lớp thi
4. Hệ thống trả về kết quả
5. Nếu đúng thì hiển thị danh sách sinh viên
6. Giáo viên nhấn nút download để tải danh sách về máy.

4.1.3.7 Sơ đồ Sequence xem lịch thi



Hình 4.8 Sơ đồ Sequence xem lịch thi

Sơ đồ Sequence xem lịch thi được diễn giải như sau:

1. Sinh viên chọn vào màn hình xem lịch thi
2. Hệ thống hiện giao diện
3. Dữ liệu danh sách thi được hiển thị lên màn hình
4. Sinh viên có thể dùng chức năng tìm kiếm theo MSSV
5. Nếu đúng MSSV thì hệ thống sẽ hiển thị danh sách lịch thi của SV
6. Sinh viên có thể chọn nút download để tải danh sách về máy.

CHƯƠNG 5: THIẾT KẾ HỆ THỐNG

5.1 Thiết kế sơ đồ thực thể liên kết (entity- relationship model)

Từ sơ đồ các lớp đã được thiết kế và các yêu cầu về dữ liệu thu thập được liệt kê trong từ điển dữ liệu ở phần thu thập yêu cầu, ta xác định được các thực thể của mô hình thực thể liên kết (ERD) được liệt kê trong Bảng 5.1 dưới đây.

Bảng 5-1 Các thực thể và thuộc tính trong mô hình ERD

<i>STT</i>	<i>Tên thực thể</i>	<i>Các thuộc tính</i>
1	Tài khoản	<u>ID</u> , username, password, level, email, MSSV/ID, Img, Status
2	Giáo viên	<u>ID</u> , Họ tên, Ngày sinh, Giới tính, Địa chỉ, Viện, SĐT
3	Giáo vụ	<u>ID</u> , Họ tên, Ngày sinh, Giới tính, Địa chỉ, SĐT
4	Sinh viên	<u>MSSV</u> , Họ tên, Ngày sinh, Giới tính, Địa chỉ, Lớp, Viện
5	Học phần	<u>Mã học phần</u> , Tên học phần, Yêu cầu thí nghiệm
6	Lớp thi	<u>Mã lớp thi</u> , Tên lớp thi, ID giáo viên, Mã học phần
7	Lịch thi	<u>ID</u> , Hình thức thi, Thời gian thi, Ngày thi, Kì học, Mã lớp thi, Mã học phần
8	Danh sách thi	Mã lớp thi, MSSV

Xác định liên kết giữa các thực thể:

- **Quan hệ 1:1 giữa Tài khoản và Giáo vụ/Giáo viên/Sinh viên:** Một người dùng chỉ có 1 tài khoản duy nhất.

- **Quan hệ 1:N giữa Giáo viên và Lớp thi:** Một Giáo viên có nhiều lớp nhưng một lớp chỉ có một giáo viên

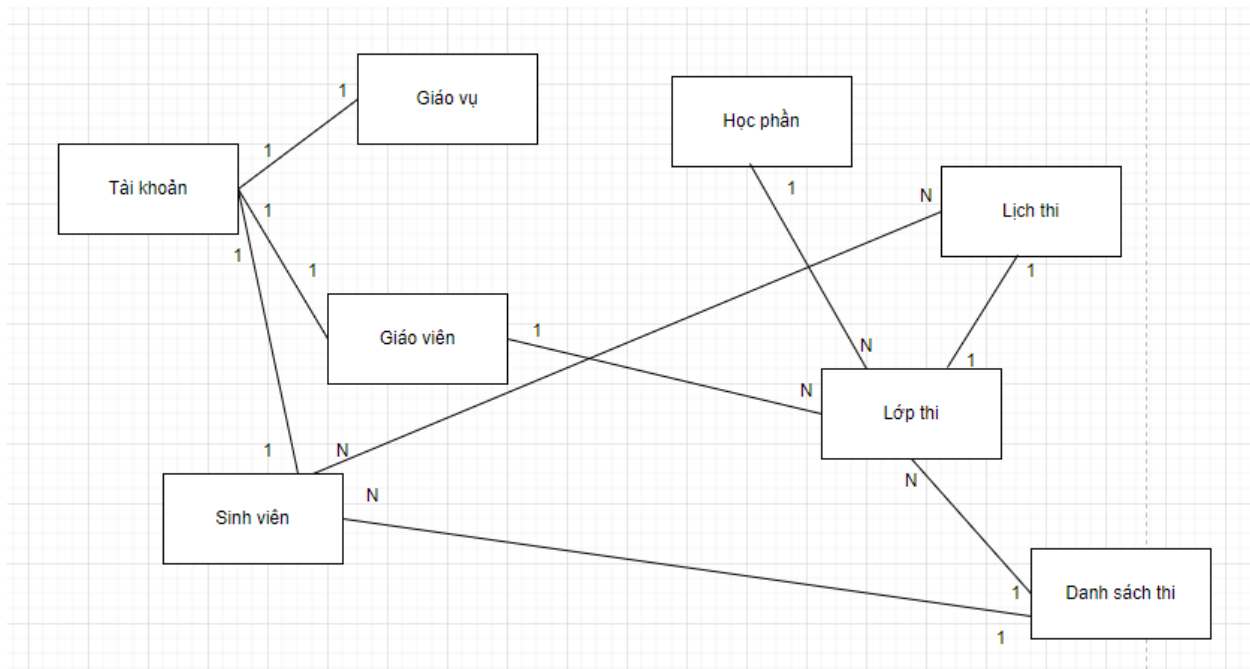
- **Quan hệ 1:N giữa Học phần và Lớp thi :** Một học phần có nhiều lớp nhưng một lớp chỉ có một học phần

- **Quan hệ 1:1 giữa Lịch thi và Lớp thi:** Một lớp thi chỉ có 1 lịch thi duy nhất.

- **Quan hệ 1:N giữa Danh sách và Lớp thi:** Một danh sách thi có nhiều lớp và một lớp chỉ có một danh sách thi.

- **Quan hệ 1:N giữa Danh sách và Sinh viên:** Một danh sách thi có nhiều sinh viên và một sinh viên chỉ trong một danh sách thi.

- **Quan hệ N:N giữa Lịch thi và Sinh viên:** Một lịch thi sẽ dành cho nhiều sinh viên và một sinh viên sẽ có nhiều lịch thi.

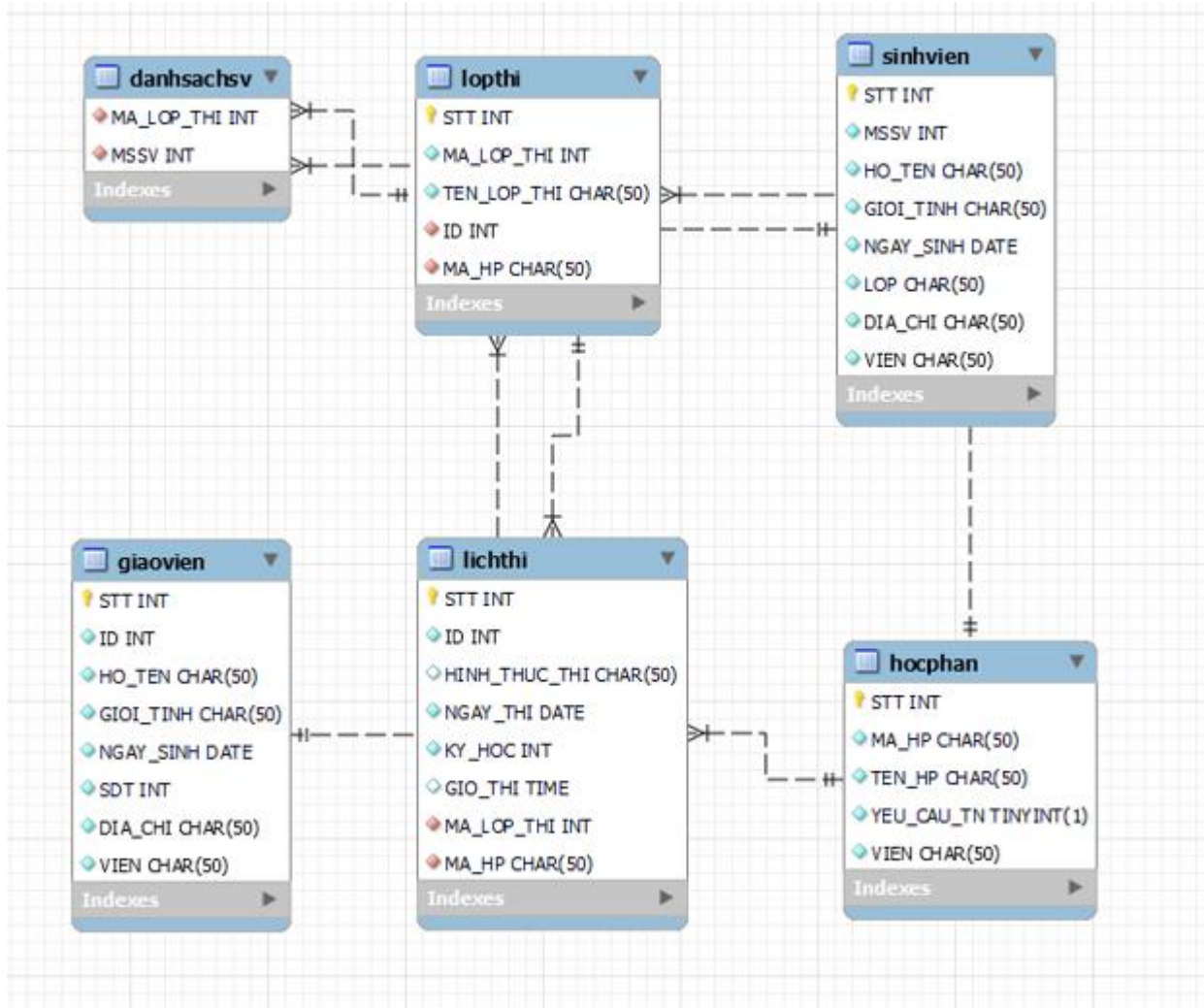


Hình 5.1 Sơ đồ thực thể liên kết

5.2 Thiết kế cơ sở dữ liệu SQL

Sơ đồ cơ sở dữ liệu theo cấu trúc SQL được mô tả như hình 5.2.

Trong hình 5.2 các bảng dữ liệu cần có được lưu trữ dữ liệu dạng SQL, trong mỗi bảng đều có khóa chính, các bảng đều được liên kết với nhau thông qua các khóa ngoại nên dữ liệu ở các bảng được ràng buộc với nhau khá chặt chẽ, theo một trình tự nhất định. Nếu muốn thêm dữ liệu vào bảng cũng cần tuân theo trình tự, bảng nào phải được thêm trước, bảng nào phải thêm dữ liệu sau, để tránh phát sinh lỗi trong quá trình truy vấn dữ liệu sau này.

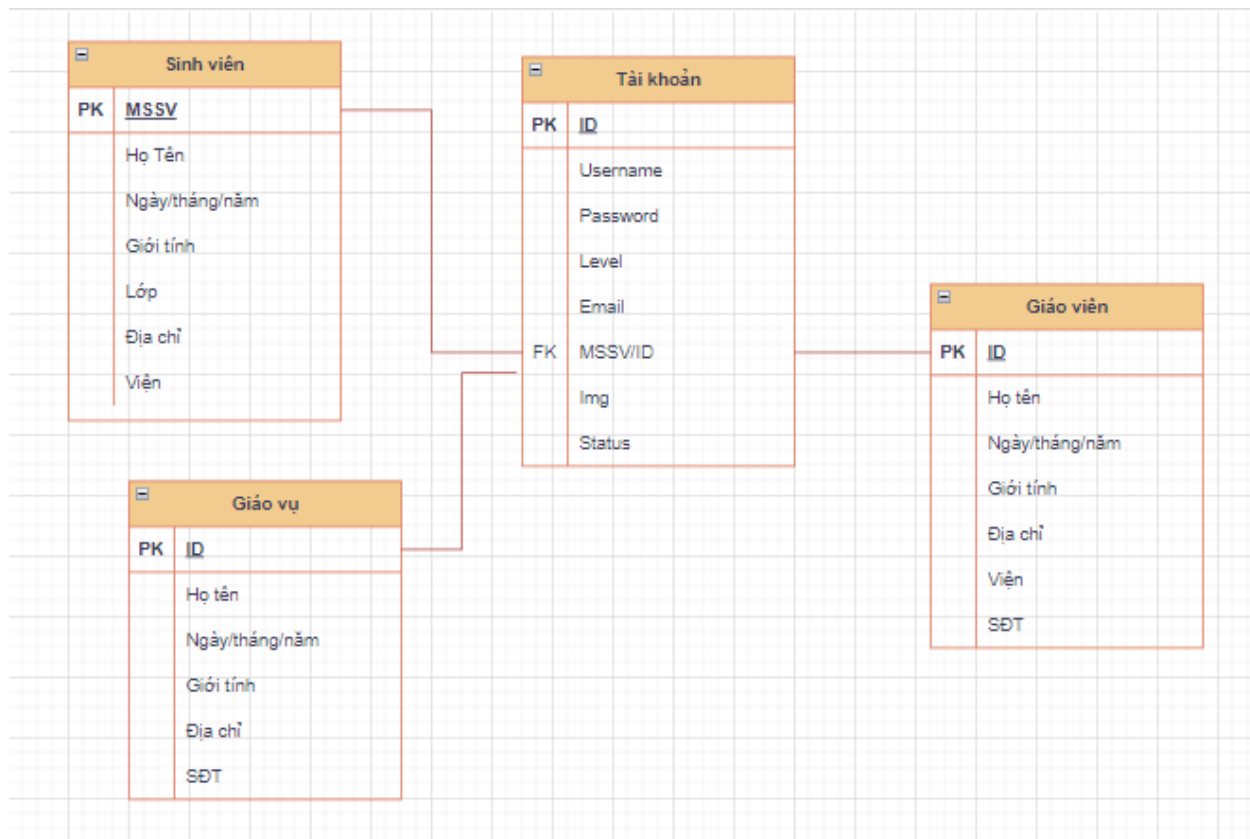


Hình 5.2 Sơ đồ cơ dữ liệu SQL

5.3 Thiết kế cơ sở dữ liệu NoSQL

Cơ sở dữ liệu được tạo bằng MongoDB, được lưu trữ dưới dạng BSON.

- **BSON** là mã hóa nhị phân của các tài liệu giống như JSON mà MongoDB sử dụng khi lưu trữ tài liệu trong các bộ sưu tập. Nó bổ sung hỗ trợ cho các loại dữ liệu như Ngày và nhị phân không được hỗ trợ trong JSON.
- Trong hệ thống này chúng em sử dụng MongoDB để lưu trữ dữ liệu tài khoản User để đăng nhập vào hệ thống.



Hình 5.3 Sơ đồ cơ sở dữ liệu NoSQL

5.4 So sánh 2 phương pháp thiết kế cơ sở dữ liệu

Trong cơ sở dữ liệu SQL, các bảng liên kết với nhau theo khóa chính và khóa ngoại có mối liên hệ chặt chẽ với nhau nên việc truy vấn phức tạp và mất thời gian hơn.

Trong cơ sở dữ liệu NoSQL, để lưu trữ và truy suất cần phải lồng ghép các model dữ liệu với nhau, MongoDB sẽ dùng các **document** thay cho row trong RDBMS và **collection** thay vì bảng

Vì vậy việc sử dụng cơ sở dữ liệu theo cấu trúc NoSQL sẽ không mất nhiều thời gian, hợp với các dự án có dữ liệu lớn.

CHƯƠNG 6: TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC

Dựa trên việc tìm hiểu về yêu cầu hệ thống cũng như phân tích thiết kế hệ thống ở chương trước, chúng em đã có cái nhìn toàn diện về website cần thực hiện. Ở chương này, chúng em từng bước thực hiện trang web với phần UI và backend, database như đã thiết kế với hy vọng mang lại một trang web đơn giản, dễ thao tác, với hiệu quả sử dụng cao.

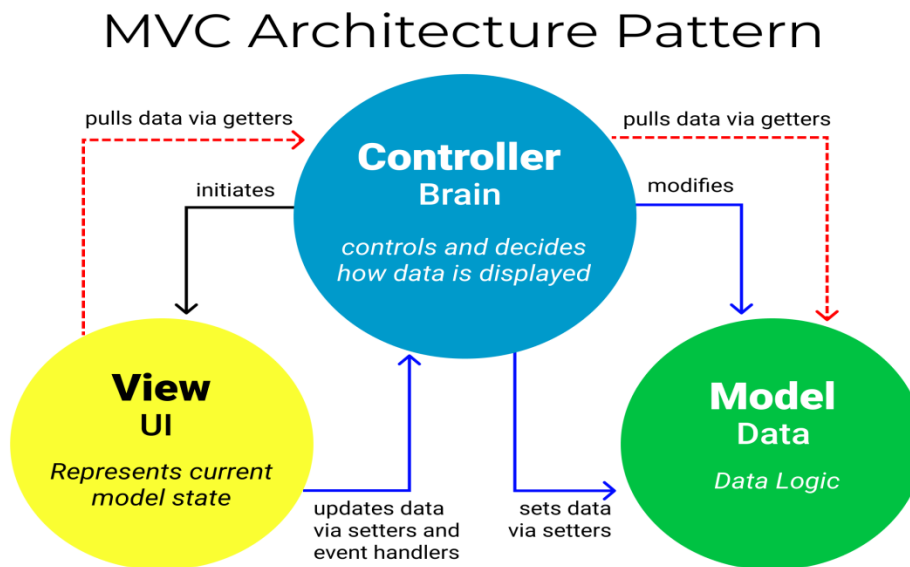
6.1 Mô hình MVC

MVC là viết tắt của cụm từ Model-View-Controller, đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một loại kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia phần mềm của chúng ta thành 3 phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó, độc lập với các thành phần khác. Các thành phần trong MVC gồm:

Model (dữ liệu) : Quản lý việc xử lý các dữ liệu, cấu trúc dữ liệu.

View (giao diện): Hiển thị dữ liệu cho người dùng.

Controller (bộ điều khiển): Điều khiển sự tương tác của 2 thành phần Model và View.



Hình 6.1 Mô hình MVC

Mô hình MVC (MVC pattern) thường được dùng để phát triển các giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

Lường xử lý cơ bản trong mô hình MVC, chúng ta có thể hình dung qua các bước:

- Khi một yêu cầu từ máy khách được gửi đến Server thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.
- Cuối cùng, khi xử lý xong thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

Hiện nay, mô hình MVC được ứng dụng khá phổ biến ở đa dạng các ngôn ngữ lập trình khác nhau, trong đó phổ biến nhất phải kể đến PHP MVC, ASP.NET MVC. Mô hình này đã và đang được ứng dụng nhiều trong lập trình. Nó thể hiện được tính hữu ích khi phát triển tốt sẽ cho phép phát triển front-end, back-end cùng hệ thống mà không cần sự can thiệp, chia sẻ, chỉnh sửa khi một hoặc cả 2 bên vẫn đang làm việc. Một ưu điểm nữa là việc sử dụng MVC trong khi thiết kế web cũng tương đối đơn giản, chỉ cần hiểu rõ về quá trình vận hành và chức năng của từng bộ phận là có thể triển khai được.

6.2 Giao diện đăng nhập và giao diện trang chủ

6.2.1 Giao diện trang chủ

Giao diện đầu tiên khi người dùng truy cập vào website là giao diện trang chủ, với button đăng nhập ở góc phải trên.

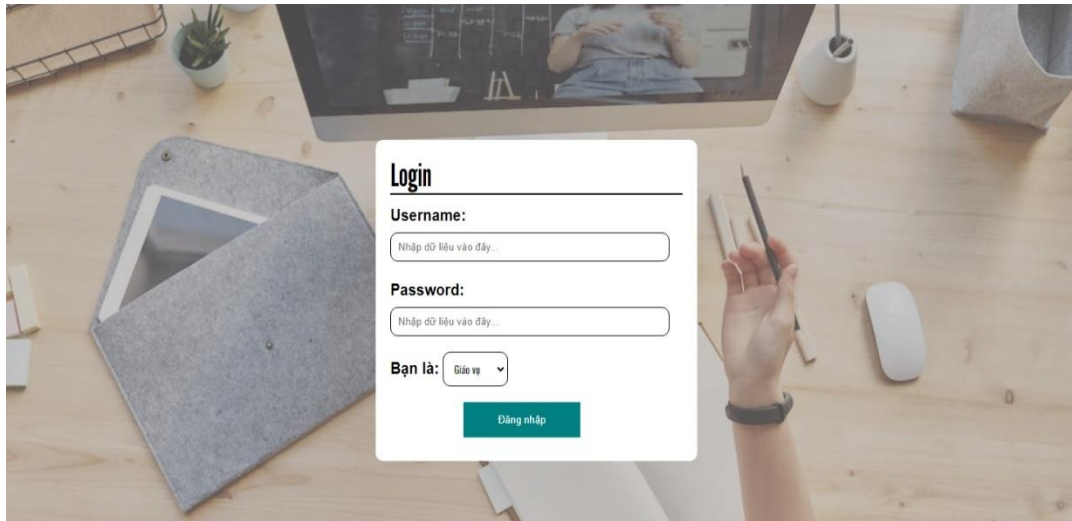


Hình 6.2 Trang chủ Website

6.2.2 Quy trình đăng nhập

Người dùng (ở các quyền khác nhau) sau khi click vào nút đăng nhập sẽ hiển thị giao diện như hình phía dưới.

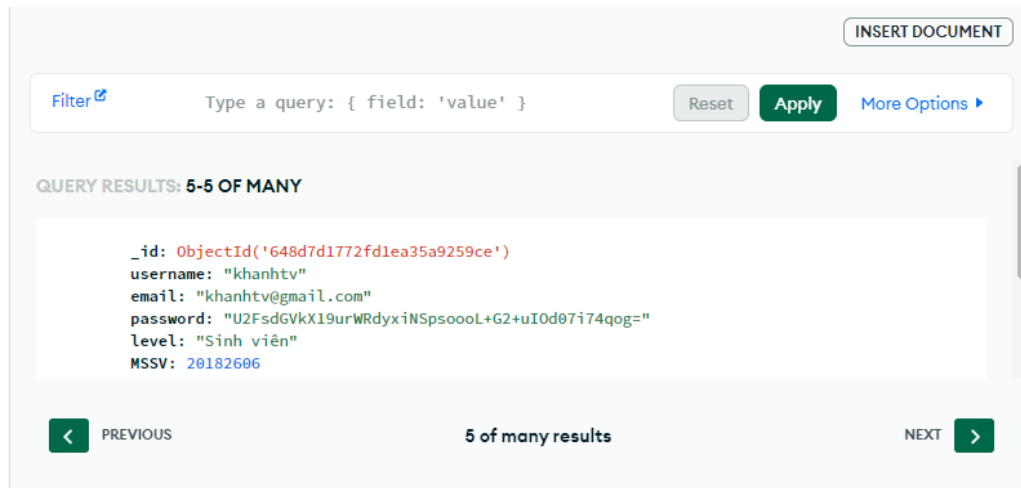
Ở đây, người dùng thực hiện nhập Tên đăng nhập và mật khẩu, ngoài ra cần chọn đúng role của tài khoản mình.



Hình 6.3 Đăng nhập

Sau khi nhận dữ liệu từ View, Controller thực hiện tìm kiếm trong các user với tên đăng nhập và role như yêu cầu, nếu không tìm thấy thì gửi thông báo không thể đăng nhập. Trong trường hợp tìm thấy thì thực hiện so sánh password của người dùng nhập với password của đối tượng được tìm thấy dưới database được model cung cấp, nếu password là đúng thì đăng nhập hoàn tất và cho vào giao diện làm việc.

Phần model ứng với giao diện đăng nhập là model User, tạo bởi sử dụng thư viện mongoose. Model này ứng với bảng sau trong database MongoDB



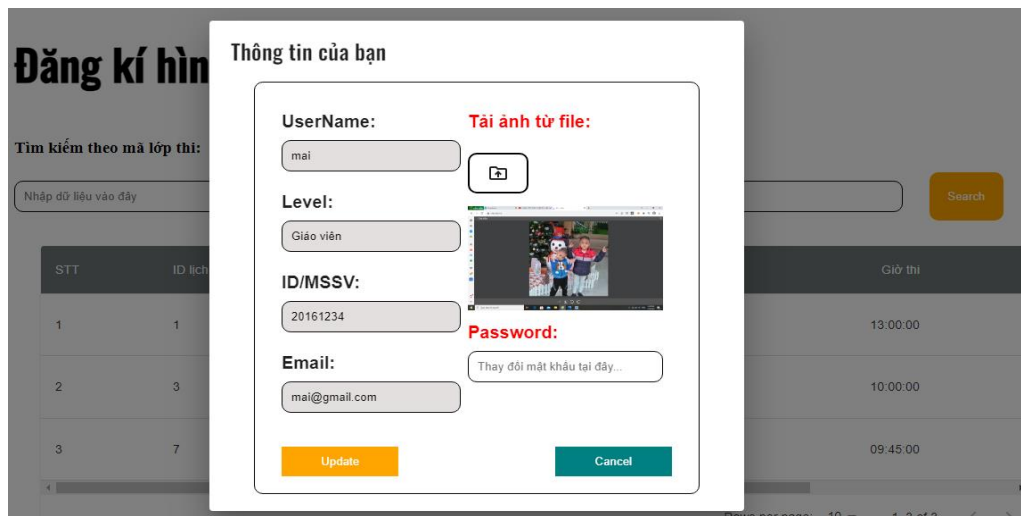
Hình 6.4 Dữ liệu sinh viên trong MongoDB

6.2.3 Thay đổi mật khẩu cho các tài khoản

Ở đây, xét việc đổi mật khẩu cho tài khoản giáo viên (với tài khoản admin hay tài khoản sinh viên ta làm tương tự)

Người dùng chọn vào ảnh đại diện phía góc phải trên màn hình, giao diện form chỉnh sửa như bên dưới sẽ được hiện lên.

Thực hiện chỉnh sửa trường thông tin Password được in màu đỏ ở bên dưới.



Hình 6.5 Giao diện thay đổi mật khẩu cho tài khoản giáo viên

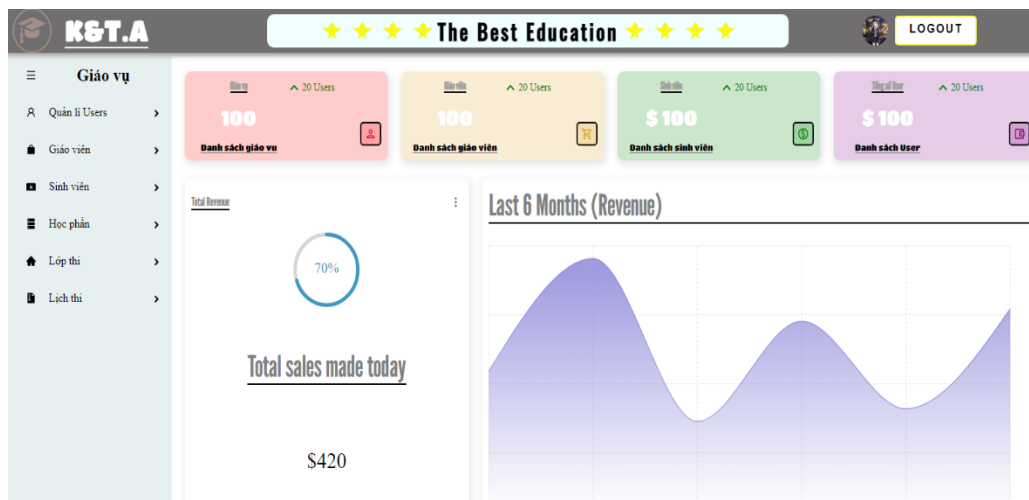
6.3 Giao diện website của từng user

6.2.1 Quy trình làm việc của admin (giáo vụ)

Bước 1: Sau khi đăng nhập

Admin sẽ có thể có các quyền như xem dashboard, thực hiện việc chỉnh sửa các thông tin như Sinh viên, Giáo viên, Học phần, Lớp thi và Lịch thi.

Dashboard cho phép admin theo dõi các thông tin về người dùng như user, giáo viên, sinh viên một cách trực quan và nhanh chóng.



Hình 6.6 Giao diện dashboard admin

Bước 2: Giáo vụ thực hiện mục tiêu cần làm khi đăng nhập, thêm hoặc xóa các tài khoản người dùng hoặc chỉnh sửa các thông tin đang bị sai.

Admin có thể thêm user bằng cách chọn và nút thêm user và thực hiện điền thông tin cho user, set ảnh đại diện cho user. User sau khi được cấp tài khoản có quyền truy cập và hệ thống với các quyền ứng với level của mình.

a) Tác vụ thêm User

K&T.A ★★★★★ The Best Education ★★★★★ **LOGOUT**

Giáo vụ

- Quản lý Users >
- Giáo viên** >
- Sinh viên >
- Học phần >
- Lớp thi >
- Lịch thi >

Thêm User

UserName:

Tải ảnh từ file:

Password:

Level:

ID/MSSV:

Email:

Hình 6.7 Giao diện thêm user

b) Tác vụ thêm, chỉnh sửa giáo viên

Admin (giáo vụ) có thể thực hiện việc thêm giáo viên vào database, thay đổi thông tin của giáo viên đó trong bảng dữ liệu cũng như xóa giáo viên trong trường hợp cần thiết. Hệ thống cung cấp giao diện tương tác một cách dễ dàng: để thêm giáo viên hãy chọn option thêm giáo viên ở sidebar và tiến hành điền dữ liệu, để xóa giáo viên hoặc thay đổi thông tin hãy vào bảng danh sách giáo viên và chọn vào giáo viên cần thao tác (có thể sử dụng công cụ tìm kiếm để đơn giản hóa).

K&T.A ★★★★★ The Best Education ★★★★★ **LOGOUT**

Giáo vụ

- Quản lý Users >
- Giáo viên** >
- Sinh viên >
- Học phần >
- Lớp thi >
- Lịch thi >

Thêm giáo viên

ID:

Ngày sinh:

Họ và tên:

SĐT:

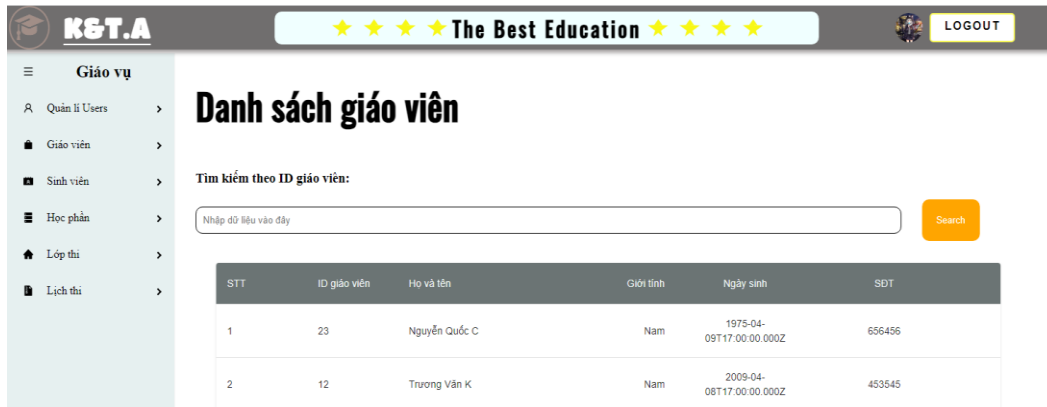
Giới tính:

Địa chỉ:

Viện:

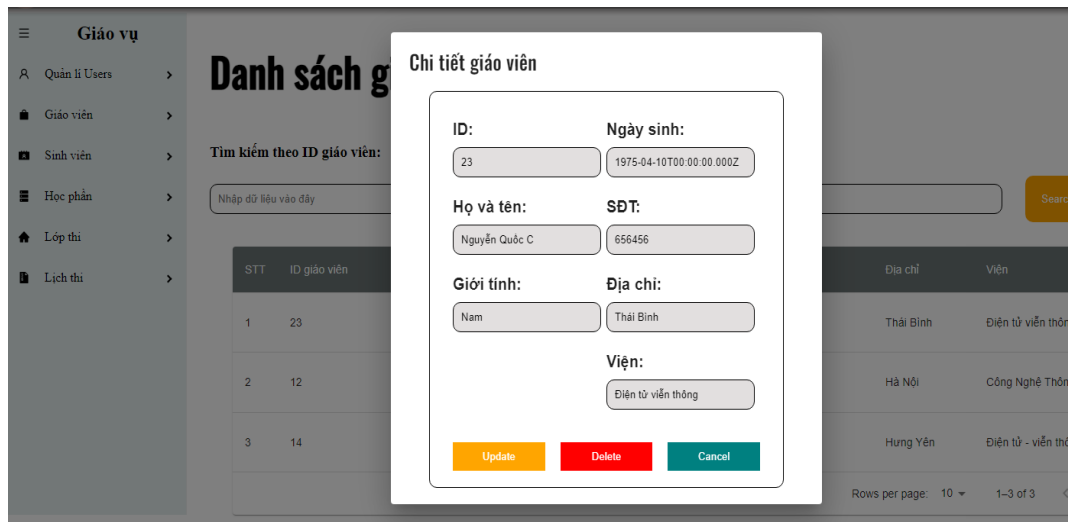
Hình 6.8 Thêm giáo viên

Giáo vụ có thể xem danh sách giáo viên hiện tại, tìm kiếm ra giáo viên cần chỉnh sửa và thực hiện click vào bảng giáo viên để chỉnh sửa giáo viên đó



Hình 6.9 Danh sách giáo viên

Sau khi click vào giáo viên muốn chỉnh sửa trong bảng giáo viên, một form chỉnh sửa sẽ hiện ra như hình dưới, giáo vụ thực hiện sửa các thông tin và submit lưu lại thông tin đã chỉnh sửa. Thông tin đó sẽ được lưu lại vào database (MySQL) cho những lần truy xuất sau.



Hình 6.10 Chỉnh sửa giáo viên

c) Tác vụ thêm và chỉnh sửa sinh viên

K&T.A. ★★★★★ The Best Education ★★★★★ **LOGOUT**

Giáo vụ

- Quản lý Users
- Giáo viên
- Sinh viên
- Học phần
- Lớp thi
- Lịch thi

Thêm sinh viên

MSSV: **Ngày sinh:**

Họ và tên: **Lớp:**

Giới tính: **Địa chỉ:**

Viện:

Submit

Hình 6.11 Thêm sinh viên

Xem danh sách các sinh viên đã được thêm vào:

K&T.A. ★★★★★ The Best Education ★★★★★ **LOGOUT**

Giáo vụ

- Quản lý Users
- Giáo viên
- Sinh viên
- Học phần
- Lớp thi
- Lịch thi

Danh sách sinh viên

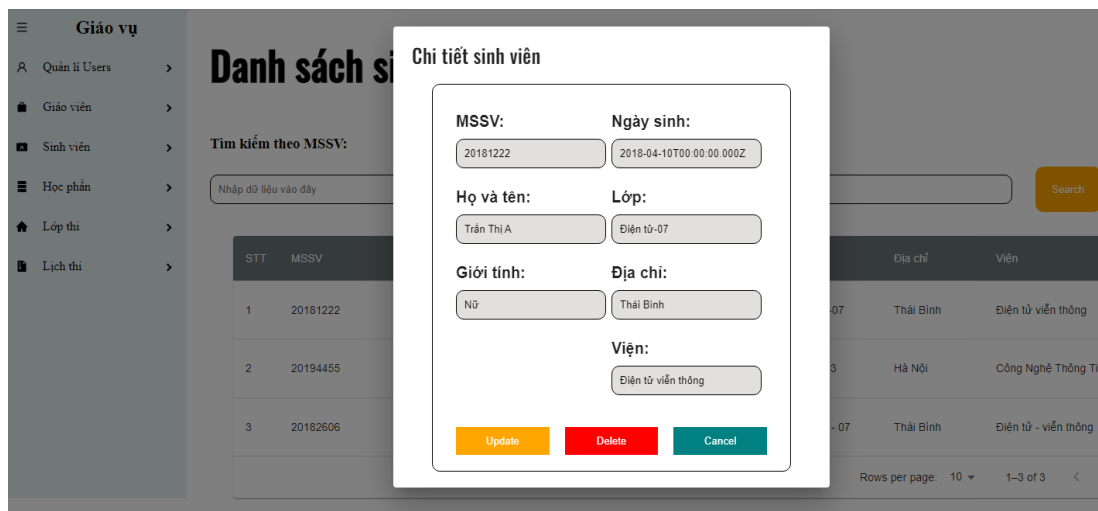
Tim kiếm theo MSSV: **Search**

STT	MSSV	Họ và tên	Giới tính	Ngày sinh	Lớp
1	20181222	Trần Thị A	Nữ	2018-04-09T17:00:00.000Z	Điện tử-07
2	20194455	Phạm Thị Y	Nữ	2016-04-09T17:00:00.000Z	CNTT-03

Rows per page: 10 1-2 of 2

Hình 6.12 Danh sách sinh viên

Giống như với lựa chọn giáo viên, thực hiện chọn sinh viên cần chỉnh sửa trong list sinh viên và form chỉnh sửa trên giao diện sẽ xuất hiện để thực hiện chỉnh sửa.

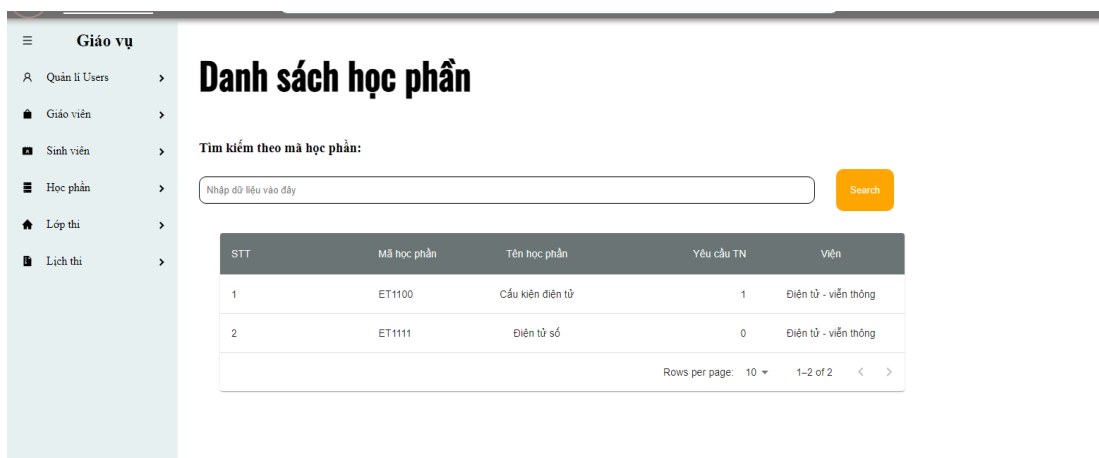


Hình 6.13 Chỉnh sửa sinh viên

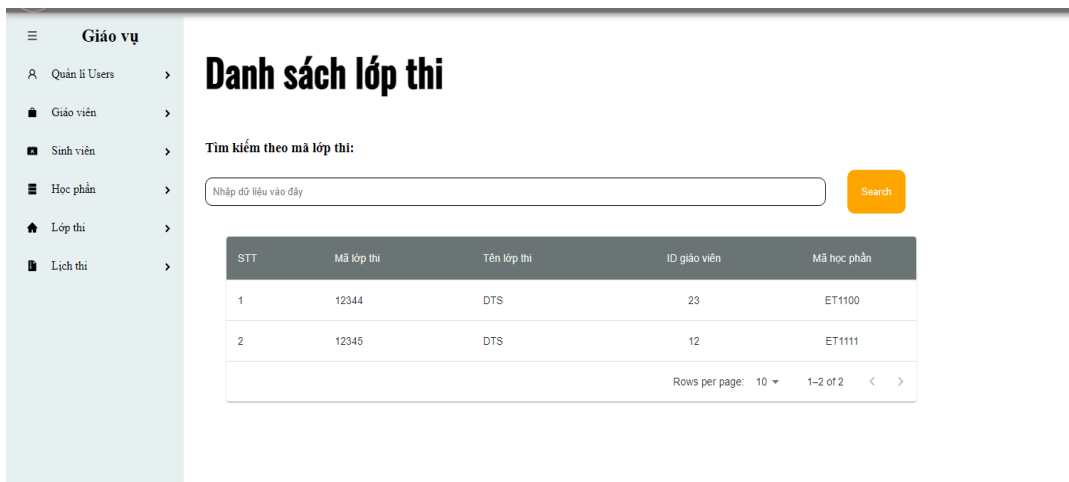
d) Tác vụ chỉnh sửa với các đối tượng thông tin còn lại.

Với giao diện học phần và lớp thi ta cũng thực hiện tương tự với giao diện tương đồng.

Dưới đây là ví dụ về danh sách học phần và danh sách lớp thi đã được thêm vào:



Hình 6.14 Danh sách học phần



Hình 6.15 Danh sách lớp thi

Bước 3: Giáo vụ thực hiện chỉnh sửa lịch thi

Giáo vụ sẽ thực hiện thêm lịch thi vào danh sách lịch thi với các trường thông tin cơ bản như thời gian thi, lớp thi, môn thi ...

Giáo viên sẽ dựa trên đó để có thể đăng ký hình thức thi, cũng như đăng ký giờ thi có thể.

Giáo vụ sau đó duyệt lại một lần nữa các thông tin và hoàn thiện các lớp thi được đăng ký. Dưới đây lần lượt là các giao diện : thêm lịch thi vào danh sách thi, xem danh sách lịch thi hiện tại, chỉnh sửa thông tin của một lớp thi bất kỳ.

Thêm một lớp thi vào danh sách:

Giáo vụ

Quản lý Users

Giáo viên

Sinh viên

Học phần

Lớp thi

Lịch thi

Sắp xếp lịch thi

ID:

Nhập dữ liệu vào đây...

Học kì:

Nhập dữ liệu vào đây...

Ngày thi:

mm/dd/yyyy

Mã lớp thi:

Mã học phần:

Submit

Hình 6.16 Sắp xếp lịch thi

Xem danh sách các lớp hiện tại:

STT	ID lịch thi	Hình thức thi	Ngày thi	Kỳ học	Giờ thi
1	1	Trắc nghiệm	2022-04-09T17:00:00.000Z	20221	13:00:00
2	3	Trắc nghiệm	2017-09-10T17:00:00.000Z	20221	10:00:00

Hình 6.17 Danh sách lịch thi

Thực hiện chỉnh sửa các thông tin lịch thi:

Chi tiết lịch thi

ID: 3 Học kì: 20221

Thời gian thi: 10:00:00 Hình thức thi: Trắc nghiệm

Ngày thi: 2017-09-10T17:00:00.000Z Mã lớp thi: 12345

Mã học phần: ET1111

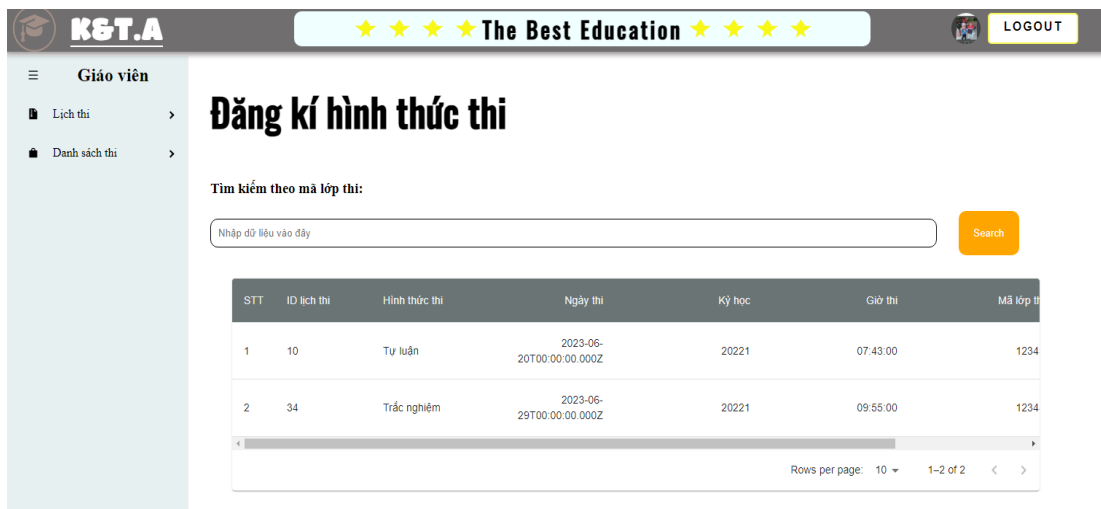
Update Delete Cancel

Hình 6.18 Chỉnh sửa lịch thi

6.2.2 Quy trình làm việc của giáo viên

Sau khi thực hiện đăng nhập vào tài khoản của mình với vai trò là giáo viên, giao diện của giáo viên sẽ được hiển thị như bên dưới.

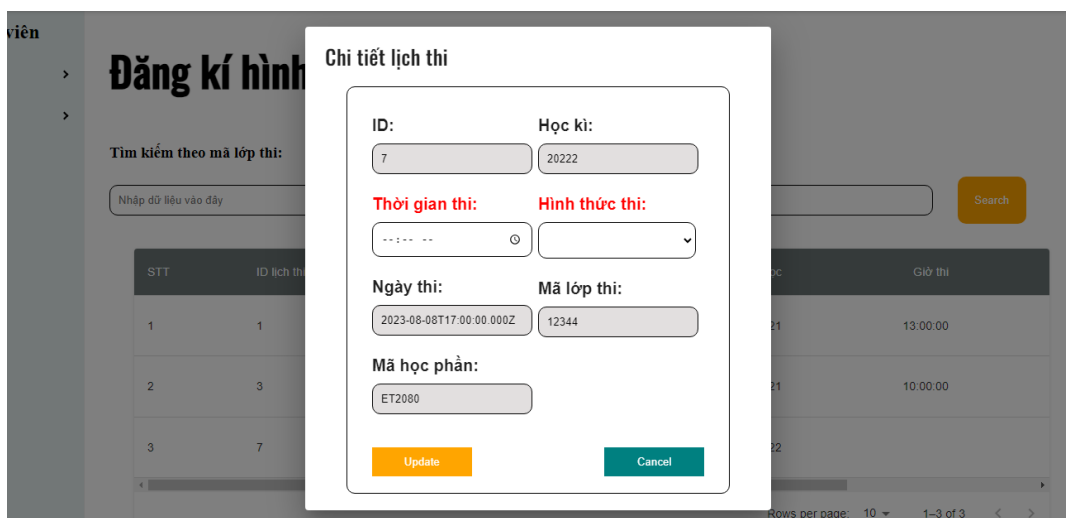
Bước 1: Giáo viên thực hiện đăng nhập vào tài khoản



Hình 6.19 Đăng ký hình thức thi

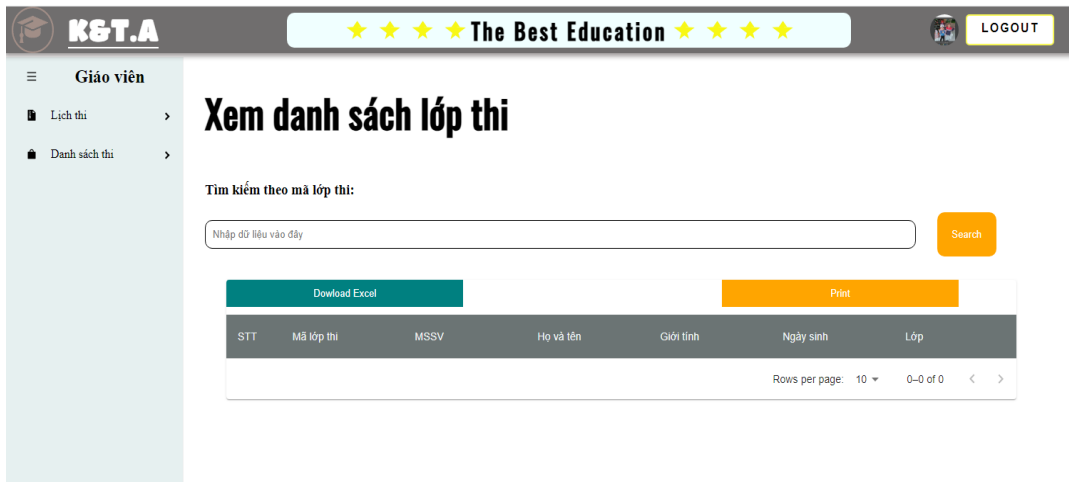
Bước 2: Giáo viên chọn vào lớp thi của mình và đăng ký hình thức thi

Giáo viên điền các thông tin trong ô màu đỏ về hình thức thi (trắc nghiệm hoặc tự luận) và thời gian về giờ thi.



Hình 6.20 Đăng ký hình thức thi

Bước 3: Sau khi đã điền thực hiện update thông tin và có thể xem danh sách lớp thi hiện tại. Nếu không còn hoạt động gì thêm có thể logout hệ thống.

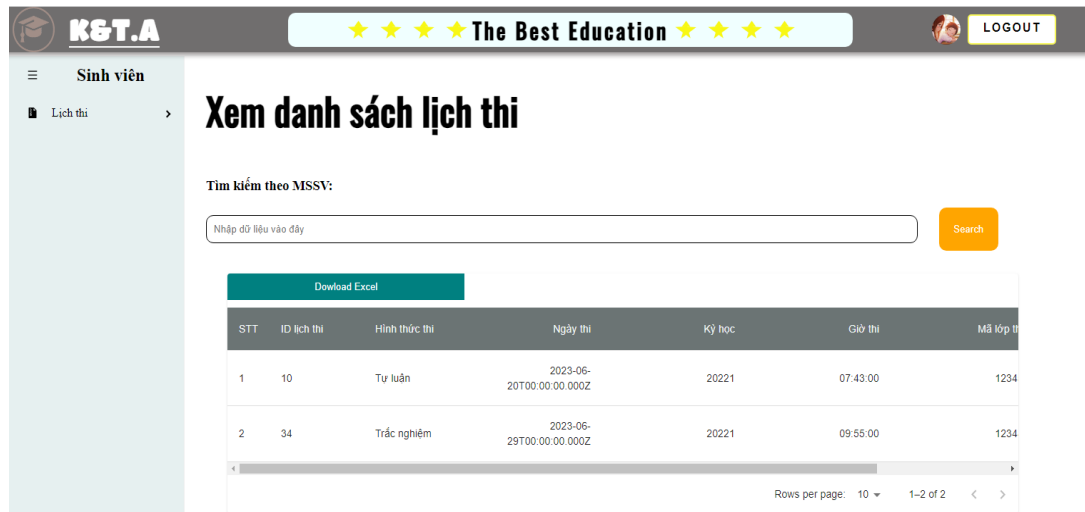


Hình 6.21 Xem danh sách thi và in ra

6.2.3 Quy trình với tài khoản sinh viên

Sinh viên được vào xem lịch thi cũng như sử dụng tính năng tìm kiếm hoặc tải file lịch thi dưới dạng excel.

Hình 6.22 Xem thông tin lịch thi



6.3 Kết luận chương

Trong chương này, chúng em đã trình bày sơ bộ kết quả của quá trình làm việc với đề án “Thiết kế website quản lý thi học kỳ”. Về cơ bản, website đáp ứng được các chức năng đã nêu ra ban đầu trong quy trình phân tích và thiết kế. Chúng em sẽ cố gắng để hệ thống được hoàn thiện hơn nữa trong tương lai gần.

KẾT LUẬN

Kết quả đạt được

Sau quá trình tìm hiểu lý thuyết, nghiên cứu, và tiến hành thực hiện đề tài đồ án tốt nghiệp, chúng em đã tiếp thu được nhiều kiến thức bổ ích, kinh nghiệm thực hành một dự án thực tế.

Đồ án đã thành công tạo ra một website quản lý và tổ chức thi hiệu quả. Chúng em đã xây dựng được một hệ thống linh hoạt và dễ sử dụng, giúp quản lý các thông tin về giáo viên, sinh viên, lịch thi một cách thuận tiện. Qua việc sử dụng các hệ quản trị cơ sở dữ liệu phù hợp, chúng em đảm bảo được tính tin cậy và hiệu suất ổn định trong quá trình lưu trữ và truy xuất dữ liệu.

Đề tài đáp ứng được các mục tiêu và yêu cầu đề ra ban đầu, đồng thời cũng mang lại lợi ích và giá trị nhất định cho người dùng cuối.

Khó khăn gặp phải

Khó khăn là điều tất yếu trong quá trình phát triển. Nắm bắt sự phức tạp của hệ thống, đảm bảo tính ổn định và hiệu suất, học những công nghệ cần thiết cho việc phát triển và đáp ứng yêu cầu người dùng là những thách thức cần vượt qua. Trong quá trình thực hiện đề tài, chúng em đã gặp phải nhiều vấn đề khó giải quyết. Tuy nhiên qua sự hướng dẫn tận tình của TS. Nguyễn Thanh Bình chúng em đã giải quyết được các vấn đề tồn đọng và hoàn thành được đồ án đúng thời hạn. Qua quá trình làm việc này, chúng em cũng đã học hỏi được những kinh nghiệm cũng như học được các làm việc nhóm hiệu quả, chuẩn bị cho khoảng thời gian sau khi tốt nghiệp ra trường.

Đề xuất và hướng phát triển

Trong tương lai, đề tài của chúng em còn cần cải thiện những vấn đề:

- Tăng cường tính năng và mở rộng quy mô: tiếp tục phát triển và cải tiến các chức năng hiện có. Mở rộng khả năng quản lý và tổ chức thi để đáp ứng các yêu cầu phức tạp hơn.
- Tích hợp công nghệ mới: nghiên cứu sử dụng các công nghệ như machine learning, data analytics để tối ưu hóa việc quản lý và sử dụng dữ liệu. có thể thêm các công cụ hỗ trợ tạo lịch tự động từ các đồ án của các bạn sinh viên khác để tối ưu hơn.
- Tạo thêm một app mobile giúp quá trình quản lý được linh hoạt.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Trang chủ ReactJS: <https://legacy.reactjs.org/docs/getting-started.html> [Truy cập lần cuối ngày 16/7/2023]
- [2] Hướng dẫn toàn tập về MySQL: <https://wiki.matbao.net/mysql-la-gi-huong-dan-toan-tap-ve-mysql/> [Truy cập lần cuối ngày 16/7/2023]
- [3] MySQL tutorial: <https://www.mysqltutorial.org/> [Truy cập lần cuối ngày 15/7/2023]
- [4] MongoDB là gì, cơ sở dữ liệu phi quan hệ: <https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgOPl9N> [Truy cập lần cuối ngày 16/7/2023]
- [5] <https://stackoverflow.com/>
- [6] CSS Tutorial: <https://www.w3schools.com/css/> [Truy cập lần cuối ngày 16/7/2023]
- [7] JavaScript Tutorial: <https://www.w3schools.com/js/> [Truy cập lần cuối này 16/7/2023]
- [8] Redux trong React.js-hướng dẫn đầy đủ: <https://techmaster.vn/posts/37636/redux-trong-react-js-huong-dan-day-du> [Truy cập lần cuối ngày 16/7/2023]
- [9] Express.js express.Router() function: <https://www.geeksforgeeks.org/express-js-express-router-function/> [Truy cập lần cuối ngày 16/7/2023]
- [10] Xây dựng website với ReactJS: <https://fullstack.edu.vn/courses/reactjs> [Truy cập lần cuối ngày 16/7/2023]