

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

PROJECT III
Xây dựng website xem phim trực tuyến

NGUYỄN ĐỨC THÀNH
thanh.nd183991@sis.hust.edu.vn

Ngành: Công nghệ thông tin
Chuyên ngành: Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Lê Đức Trung _____

Bộ môn: Công nghệ phần mềm
Trường: Trường Công nghệ thông tin và Truyền thông

HÀ NỘI, 01/2022

Lời cam kết

Họ và tên sinh viên: Nguyễn Đức Thành.

Điện thoại liên lạc: 0386851140. Email: thanh.nd183991@sis.hust.edu.vn

Lớp: Công nghệ thông tin 01. Hệ đào tạo: Chính quy.

Tôi –*Nguyễn Đức Thành*– cam kết Project III là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của Thạc sỹ Lê Đức Trung. Các kết quả nêu trong Project III là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong Project III – 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 08 tháng 01 năm 2022

Tác giả Project III

Thành

Nguyễn Đức Thành

Lời cảm ơn

Trước hết em xin chân thành cảm ơn thầy Thạc Sĩ Lê Đức Trung - Bộ môn Công nghệ phần mềm – Trường Công nghệ Thông tin và Truyền thông – Trường Đại học Bách khoa Hà Nội, người đã hướng dẫn em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành học phần này. Đồng thời, em cũng xin chân thành cảm ơn các thầy cô trong bộ môn cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành tốt học phần này.

Em xin gửi lời cảm ơn đến các thành viên lớp Công nghệ thông tin IT3-01, những người bạn đã luôn ở bên cạnh động viên, tạo điều kiện thuận lợi giúp em hoàn thành tốt học phần.

Sau cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã tạo mọi điều kiện để em xây dựng thành công học phần này.

Mặc dù bản thân em đã cố gắng nhưng do thời gian, kiến thức và kinh nghiệm có hạn nên sản phẩm và báo cáo của em còn có thể có nhiều thiếu sót...Vì vậy, em rất mong nhận được sự thông cảm và nhận xét từ thầy để sản phẩm cũng như báo cáo của em được hoàn thiện hơn.

Tóm tắt

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kỹ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin. Ở nước ta, cùng với sự phát triển của nền kinh tế, tin học đang ngày được ứng dụng rộng rãi trong công tác quản lý và từng bước khẳng định sức mạnh cũng như vị trí quan trọng của mình.

Trong ngành CNTT, các phần mềm được thiết kế dựa nhu cầu sử dụng, có 3 mô hình lập trình phần mềm là: Mô hình máy đơn, client/server và peer-to-peer. Trong đó, mô hình client/server là mô hình được sử dụng rộng rãi vì nó có thể cho phép người dùng ở bất kì đâu, chỉ cần có kết nối Internet là dễ dàng truy cập vào và sử dụng. Có rất nhiều lĩnh vực ứng dụng trên nền tảng client-server như là quản lý nhân sự, thương mại điện tử, y tế, giáo dục và đào tạo, vui chơi giải trí, ... Trong đó học phần này, với mục đích xây dựng một hệ thống xem phim trực tuyến, em chọn mô hình thiết kế client-server để xây dựng 1 trang web cho phép người dùng có thể truy cập vào để xem phim trực tuyến thông qua mạng Internet. Bên phía client, hệ thống sử dụng thư viện ReactJS để xây dựng giao diện cho người dùng, phía server hệ thống sử dụng NodeJS và cơ sở dữ liệu phi quan hệ NoSQL MongoDB. Hệ thống tạo ra môi trường giúp người dùng trải nghiệm phim một cách tốt nhất, giúp người điều hành hệ thống quản lý phim một cách hiệu quả. Hệ thống được thử nghiệm và bước đầu hoạt động ổn định trên mạng LAN.

Abstract

Nowadays, information technology applications have been applied in all fields of scientific and technical research as well as in life, computers have become effective aids for people in storing, analyzing and information processing. In our country, along with the general development of the economy, informatics is being widely applied in management work and gradually asserting its strength and important position.

In the IT industry, software is designed based on user needs, there are 3 software programming models: single machine, client/server and peer-to-peer model. In which the client / server model is the most preferred model because it can allow users anywhere, just an Internet connection is easy to access and use. There are many application areas on the client-server platform such as human resource management, e-commerce, healthcare, education and training, entertainment, ... In which this module, with the aim of For the purpose of building an online movie viewing system, I chose a client-server design model to build a website that allows users to access to watch movies online via the Internet. On the client side, the system uses the ReactJS library to build the user interface, the server side uses NodeJS and the NoSQL MongoDB non-relational database. The system helps operators manage movies and series effectively, helping users to search, watch movies, receive notifications from movies. The system has been tested and initially works stably on the LAN.

Mục lục

Lời cam kết	ii
Lời cảm ơn	iii
Tóm tắt.....	iv
Abstract.....	v
Mục lục.....	vi
Danh mục hình vẽ.....	ix
Danh mục bảng.....	xi
Danh mục công thức.....	xii
Danh mục các từ viết tắt.....	xiii
Danh mục thuật ngữ.....	xiv
Chương 1 Giới thiệu đề tài.....	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp	1
1.4 Bố cục đồ án.....	2
Chương 2 Khảo sát và phân tích yêu cầu	3
2.1 Khảo sát hiện trạng.....	3
2.2 Tổng quan chức năng.....	3
2.2.1 Biểu đồ use case tổng quan	3
2.2.2 Biểu đồ use case phân rã “Quản lý người dùng”	5

2.2.3 Biểu đồ use case phân rã “Quản lý phim”	6
2.2.4 Biểu đồ use case phân rã “Quản lý series”	7
2.2.5 Quy trình nghiệp vụ.....	8
2.3 Đặc tả chức năng	9
2.3.1 Đặc tả use case “Đăng nhập”	9
2.3.2 Đặc tả use case “Thêm phim”	10
2.3.3 Đặc tả use case “Sửa phim”	11
2.3.4 Đặc tả usecase “Tạo series”	12
2.3.5 Đặc tả use case “Tìm phim”	13
2.3.6 Đặc tả use case “Thích phim”	13
2.4 Yêu cầu phi chức năng.....	14
Chương 3 Công nghệ sử dụng.....	15
Chương 4 Phát triển và triển khai ứng dụng	20
4.1 Thiết kế kiến trúc.....	20
4.1.1 Lựa chọn kiến trúc phần mềm.....	20
4.1.2 Thiết kế tổng quan	21
4.1.3 Thiết kế chi tiết gói.....	24
4.2 Thiết kế chi tiết.....	27
4.2.1 Thiết kế giao diện	27
4.2.2 Thiết kế lớp	29
4.2.3 Thiết kế cơ sở dữ liệu	34
4.3 Xây dựng ứng dụng	37
4.3.1 Thư viện và công cụ sử dụng	37
4.3.2 Kết quả đạt được.....	38
4.3.3 Minh hoạ các chức năng chính.....	39
4.4 Kiểm thử	46
4.5 Triển khai	49

Chương 5 Các giải pháp và đóng góp nổi bật	51
Chương 6 Kết luận và hướng phát triển	58
6.1 Kết luận.....	58
6.2 Hướng phát triển.....	58
Tài liệu tham khảo.....	59
Phụ lục	1

Danh mục hình vẽ

Hình 1 Sơ đồ Usecase tổng quan.....	3
Hình 2 Phân rã Usecase “Quản lý người dùng”	5
Hình 3 Phân rã Usecase “Quản lý phim”	6
Hình 4 Phân rã Usecase “Quản lý series”	7
Hình 5 Quy trình nghiệp vụ sửa thông tin phim.....	8
Hình 6 Cấu trúc JWT	17
Hình 7 JWT hoạt động.....	18
Hình 8 Mô hình MVC.....	20
Hình 9 Mô hình MVVM	21
Hình 10 Tổng quan gói	22
Hình 11 Thiết kế gói tổng quan Client	24
Hình 12 Thiết kế gói chi tiết người dùng Client	24
Hình 13 Thiết kế gói chi tiết movie Client.....	25
Hình 14 Thiết kế tổng quan gói Admin	25
Hình 15 Thiết kế chi tiết gói Auth Admin	26
Hình 16 Thiết kế chi tiết gói quản lý Admin	26
Hình 17 Thiết kế chi tiết gói API	27
Hình 18 Thiết kế giao diện trang chủ client	28
Hình 19 Thiết kế giao diện admin	29
Hình 20 Chi tiết lớp MoviesController	29
Hình 21 Biểu đồ tuần tự “UpdateMovie”	31

Hình 22 Biểu đồ tuần từ “GetAllGenres”	31
Hình 23 Thiết kế chi tiết lớp SeriesController	32
Hình 24 Biểu đồ tuần từ “CreateSeries”	33
Hình 25 Biểu đồ tuần từ “DeleteSeries”	33
Hình 26 Sơ đồ thực thể liên kết.....	34
Hình 27 Giao diện Trang chủ Người dùng	39
Hình 28 Giao diện Trang Tìm kiếm phim	40
Hình 29 Giao diện chi tiết phim	41
Hình 30 Giao diện xem phim	42
Hình 31 Giao diện Quản lý người dùng.....	42
Hình 32 Giao diện Quản lý phim	43
Hình 33 Giao diện sửa phim	43
Hình 34 Quản lý Series phim	44
Hình 35 Giao diện tạo series phim	44
Hình 36 Giao diện sửa series.....	45
Hình 37 Modal sửa series phim.....	45
Hình 38 Mô hình triển khai của hệ thống	49
Hình 39 chữ ký trong JWT.....	53
Hình 40 Cách thức hoạt động JWT	54

Danh mục bảng

Bảng 1 Đặc tả usecase “Đăng nhập”	10
Bảng 2 Đặc tả use case “Thêm phim”	11
Bảng 3 Đặc tả use case “Sửa phim”	12
Bảng 4 Đặc tả use case “Thêm series”	13
Bảng 5 Đặc tả use case “Tìm phim”	13
Bảng 6 Đặc tả use case “Thích phim”	14
Bảng 7 Mô tả tổng quan gói	23
Bảng 8 Mô tả thuộc tính của lớp MovieController.....	30
Bảng 9 Mô tả phương thức của lớp MovieController.....	30
Bảng 10 Mô tả thuộc tính của lớp SeriesController	32
Bảng 11 Mô tả phương thức của lớp SeriesController	33
Bảng 12 Đặc tả ERD.....	36
Bảng 13 Danh sách thư viện và công cụ sử dụng	37
Bảng 14 Kiểm thử email	46
Bảng 15 Kiểm thử tên người dùng	47
Bảng 16 Kiểm thử mật khẩu người dùng.	47
Bảng 17 Kiểm thử thông tin mô tả về phim.	48
Bảng 18 Kiểm thử năm sản xuất phim.....	49
Bảng 19 Cấu hình máy triển khai ứng dụng.....	50
Bảng 20 Kết quả triển khai.....	50

Danh mục công thức

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
IT	Information Technology Công nghệ thông tin
UI	User Interface Giao diện người dùng
HTML	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
CNTT	Công nghệ thông tin
CSS	Cascading Style Sheets Là ngôn ngữ định dạng văn bản
SV	Sinh viên
JWT	JSON Web Tokens Chuỗi xác thực người dùng
SSR	SERVER-SIDE RENDERING Cơ chế hiển thị trang website

Danh mục thuật ngữ

Browser	Trình duyệt
Cache memory	Bộ nhớ đệm
Compiler	Trình biên dịch
Javascript	Là ngôn ngữ lập trình
ReactJS	Một thư viện viết bằng javascript, dùng để xây dựng giao diện người dùng.
MUI	Thư viện hỗ trợ ReactJS xây dựng giao diện.
NodeJS	Là nền tảng được xây dựng dựa trên “V8 Javascript engine”(giúp phân tích và thông dịch javascript thành mã máy) được viết bằng C++
MongoDB	Cơ sở dữ liệu phi quan hệ
Server	Là một máy chủ hay máy phục vụ là một phần của máy tính phần cứng hoặc phần mềm mà cung cấp chức năng cho các chương trình khác.
JWT	Giao thức xác thực người dùng trên internet
Redux	Công cụ quản lý trạng thái của giao diện

RSA	Một thuật toán mật mã hóa khóa công khai.
MVC	Mô hình kiến trúc ứng dụng
MVVM	Mô hình kiến trúc ứng dụng
Client	Máy khách
JSON	JavaScript Object Notation Một định dạng trao đổi dữ liệu nhẹ.
base64	Phương thức chuyển đổi dạng mã hóa 2 chiều từ binary sang string để có thể gửi đi được trong network một cách dễ dàng
Public key	Khóa công khai khi mã hóa
Private key	Khóa bí mật

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Trong thời đại công nghiệp hóa hiện đại hóa ngày nay, cuộc sống con người ngày càng đa dạng về nhu cầu tận hưởng và giải trí. Mọi người tìm kiếm các dịch vụ giải trí đáp ứng nhu cầu của họ một cách nhanh chóng và hiệu quả. Những năm gần đây, các nền tảng ứng dụng xem phim trực tuyến có sự phát triển mạnh mẽ. Điều đó đã dần được đông đảo khán giả tiếp cận nhằm thỏa mãn nhu cầu giải trí.

Tuy nhiên trên thực tế, nhiều ứng dụng website hiện nay có tốc độ chậm, độ tin cậy không cao gây khó chịu cho người dùng.

Hiểu được điều đó, em đã chọn đề tài “Xây dựng website xem phim trực tuyến” để giúp cho người dùng có những trải nghiệm xem phim tốt nhất, độ tin cậy cao. Đồng thời, hệ thống giúp người điều hành quản lý phim, người dùng một cách nhanh chóng và hiệu quả.

1.2 Mục tiêu và phạm vi đề tài

Công nghệ để xây dựng em chọn là công nghệ web do những ưu điểm của chúng. Những chức năng chính cho hệ thống là: (i) quản lý đăng nhập, tài khoản cá nhân, (ii) xem phim, thích, bình luận phim, (iii) quản lý người dùng, (iv) quản lý phim, series.

Hệ thống cung cấp giao diện đẹp mắt giúp người dùng xem phim. Tuy nhiên, hệ thống còn ít chức năng, có thể chưa đáp ứng được nhu cầu từ phía người dùng. Hệ thống sẽ phát triển thêm hệ thống ủng hộ (donate), và thêm chức năng xem sau, lịch sử xem giúp tối ưu trải nghiệm của người dùng.

1.3 Định hướng giải pháp

Hệ thống sẽ được triển khai trên nền tảng website trực tuyến nơi mọi người có thể kết nối dễ dàng và được xây dựng theo mô hình client-server. Về phía client, hệ thống sử dụng thư viện ReactJS, cùng với HTML/CSS, redux, MUI, để xây dựng giao diện. Về phía server, hệ thống sử dụng NodeJS (được viết bằng ngôn ngữ Javascript), cơ sở dữ liệu phi quan hệ MongoDB.

1.4 Bố cục đồ án

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

Chương 2 trình bày về Khảo sát và phân tích yêu cầu của bài toán được phân tách thành 2 yêu cầu chính là từ phía người dùng và người quản trị hệ thống, đồng thời cũng nêu lên những yêu cầu phi chức năng của hệ thống.

Trong Chương 3, sau khi đã phân tích yêu cầu bài toán từ chương 2, em bắt đầu chọn lựa những công nghệ sử dụng phù hợp với đặc tả của bài toán trên 3 yếu tố: front-end, back-end và cơ sở dữ liệu

Ở Chương 4, em tiến hành lựa chọn kiến trúc mô hình, thiết kế chi tiết gói mà hệ thống sẽ sử dụng ở phần 4.1 và thiết kế chi tiết lớp, giao diện, thiết kế cơ sở dữ liệu của hệ thống trong phần 4.2. Sau đó, minh họa giao diện chương trình kiểm thử và cách triển khai hệ thống.

Trong Chương 5, trình bày đóng góp chính của đồ án, đó là một nền tảng xem phim trực tuyến cho phép người dùng trải nghiệm kho phim một cách tối ưu. Nền tảng website được phát triển dựa trên khái niệm thiết kế kiến trúc client/server theo mô hình MVVM, sử dụng bảo mật, sử dụng các thư viện giúp hiển thị giao diện một cách trực quan và sinh động với người sử dụng.

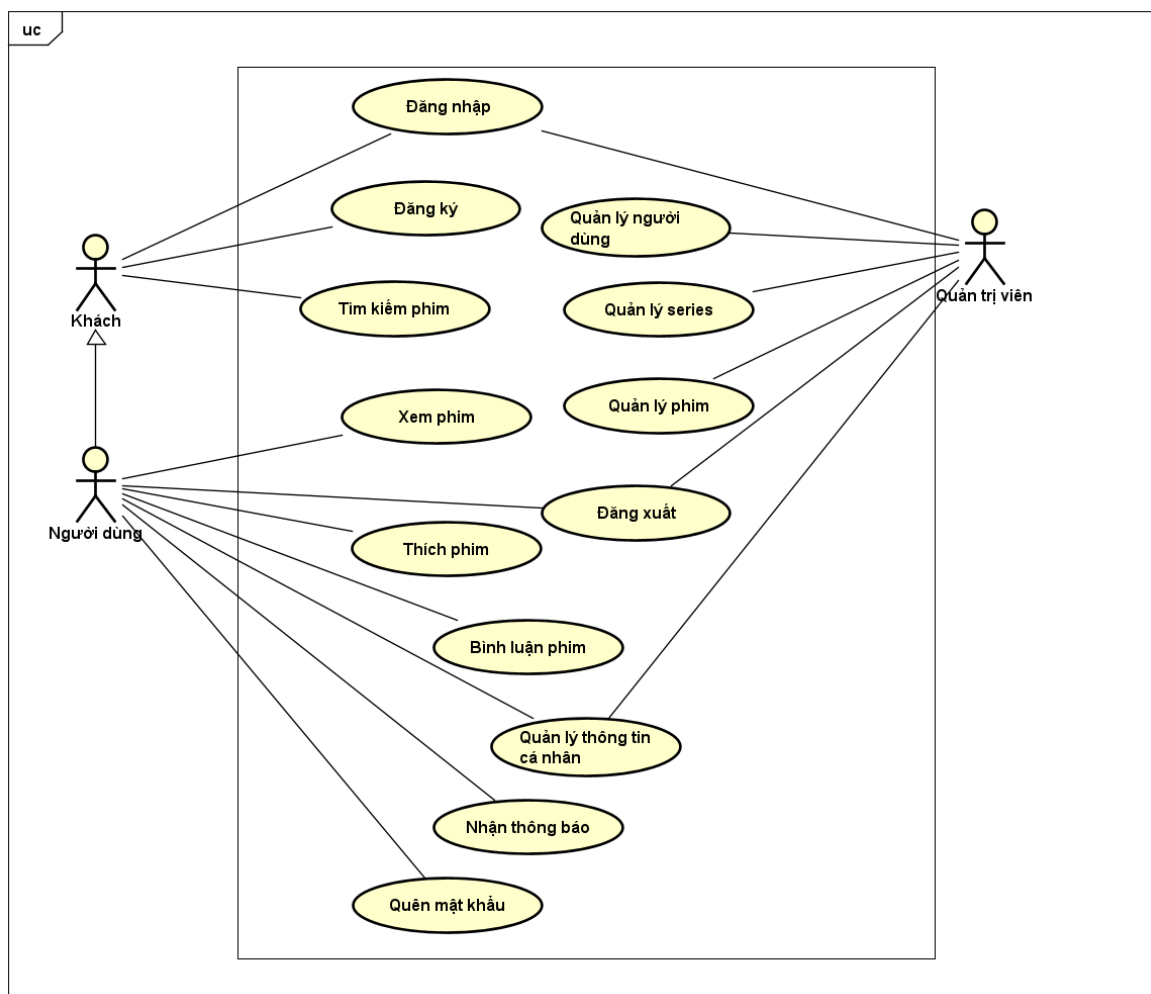
Chương 6 nói về kết luận những gì đã đạt được trong học phần này và hạn chế cần phải giải quyết trong hệ thống. Từ đó, nêu lên hướng phát triển tiếp theo của hệ thống.

Chương 2 Khảo sát và phân tích yêu cầu

2.1 Khảo sát hiện trạng

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan



Hình 1 Sơ đồ Usecase tổng quan

Danh sách tác nhân:

STT	Tên tác nhân	Mô tả tác nhân
-----	--------------	----------------

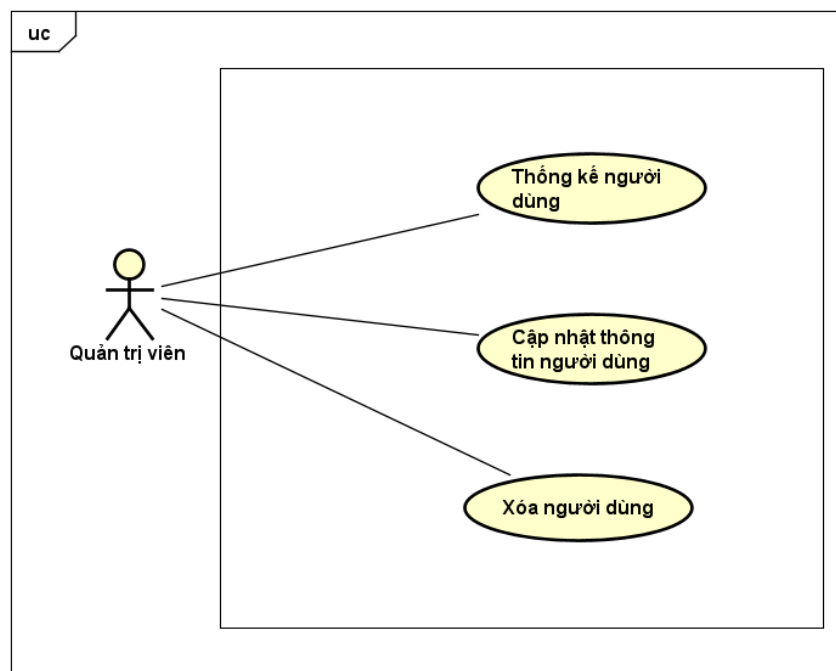
1	Khách	Có quyền đăng ký tài khoản để thành người dùng
2	Người dùng	Là người sử dụng hệ thống, có thể thực hiện các hoạt động như: đăng xuất, xem phim ...
3	Quản trị viên	Người quản trị hệ thống, thực hiện nghiệp vụ: quản lý người dùng, quản lý phim, quản lý series.

Mô tả các usecase chính

STT	Mã Usecase	Tên Usecase	Mô tả Usecase	Tác nhân tương tác
1	UC01	Đăng nhập	Cho phép người dùng có tài khoản truy cập vào hệ thống	Người dùng, quản trị viên
2	UC02	Đăng xuất	Thoát khỏi hệ thống	Người dùng, quản trị viên
3	UC03	Đăng ký	Cho phép khách đăng ký sử dụng hệ thống	Khách
4	UC04	Quản lý người dùng	Cho phép quản trị viên quản lý được các thông tin về người dùng, xóa, chỉnh sửa thông tin về người dùng	Quản trị viên
5	UC05	Quản lý phim	Cho phép quản trị viên quản lý thông tin phim, thêm sửa xóa phim	Quản trị viên
6	UC06	Quản lý series	Cho phép quản trị viên quản lý thông tin series, thêm sửa xóa series	Quản trị viên
7	UC07	Quản lý tài khoản cá nhân	Cho phép người dùng có thể xem sửa thông tin cá nhân	Người dùng, quản trị viên

8	UC08	Tìm kiếm phim	Cho phép khách có thể tìm kiếm bộ phim	Người dùng, khách
9	UC09	Xem phim	Cho phép người dùng xem phim	Người dùng
10	UC10	Thích phim	Cho phép người dùng thích phim	Người dùng
11	UC11	Bình luận phim	Cho phép người dùng bình luận phim	Người dùng
12	UC12	Nhận thông báo	Cho phép người dùng nhận thông báo về phim đã tương tác(thích, bình luận).	Người dùng
13	UC13	Quên mật khẩu	Cho phép người dùng cập nhật lại khi mất mật khẩu	Người dùng

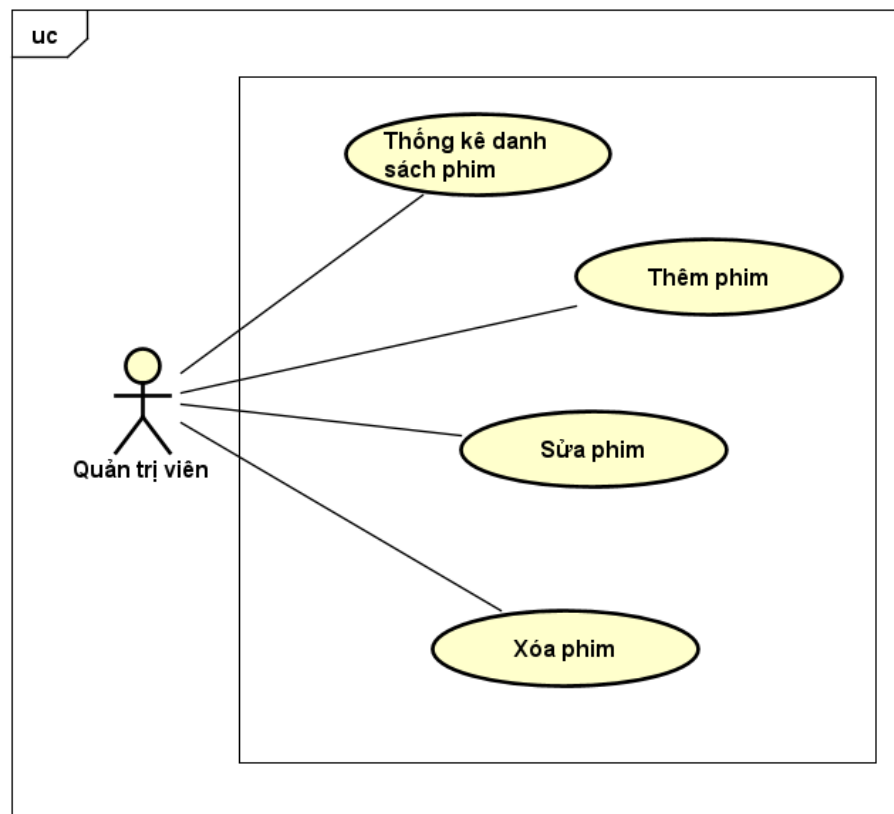
2.2.2 Biểu đồ use case phân rã “Quản lý người dùng”



Hình 2 Phân rã Usecase “Quản lý người dùng”

STT	Mã Usecase	Tên UseCase	Mô tả Usecase	Tác nhân tương tác
1	UC04-1	Thống kê người dùng	Thông kê những người dùng hiện có lọc người dùng	Quản trị viên
2	UC04-2	Cập nhật thông tin người dùng	Sửa lại những thông tin người dùng không phù hợp (liên quan đến tôn giáo, sắc tộc,...)	Quản trị viên
3	UC04-3	Xóa người dùng	Xóa người dùng khỏi hệ thống	Quản trị viên

2.2.3 Biểu đồ use case phân rã “Quản lý phim”

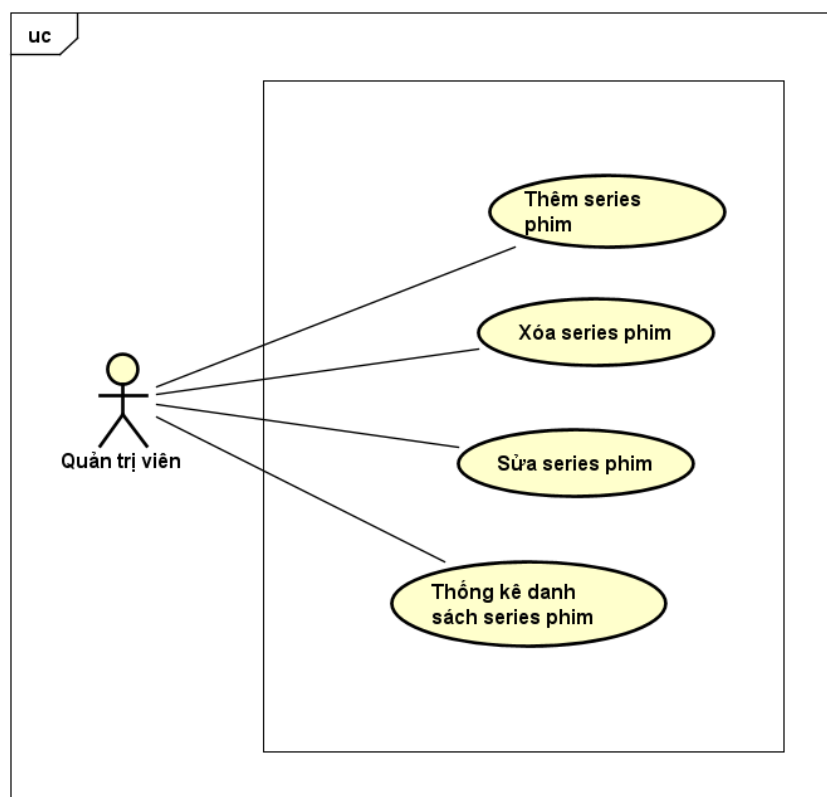


Hình 3 Phân rã Usecase “Quản lý phim”

STT	Mã Usecase	Tên UseCase	Mô tả Usecase	Tác nhân tương tác
-----	------------	-------------	---------------	--------------------

1	UC05-1	Thống kê phim	Thông kê những danh sách phim hiện có và lọc phim	Quản trị viên
2	UC05-2	Thêm phim	Thêm mới một phim vào hệ thống	Quản trị viên
3	UC05-3	Sửa phim	Sửa một bộ phim	Quản trị viên
4	UC05-4	Xóa phim	Xóa một bộ phim	Quản trị viên

2.2.4 Biểu đồ use case phân rã “Quản lý series”

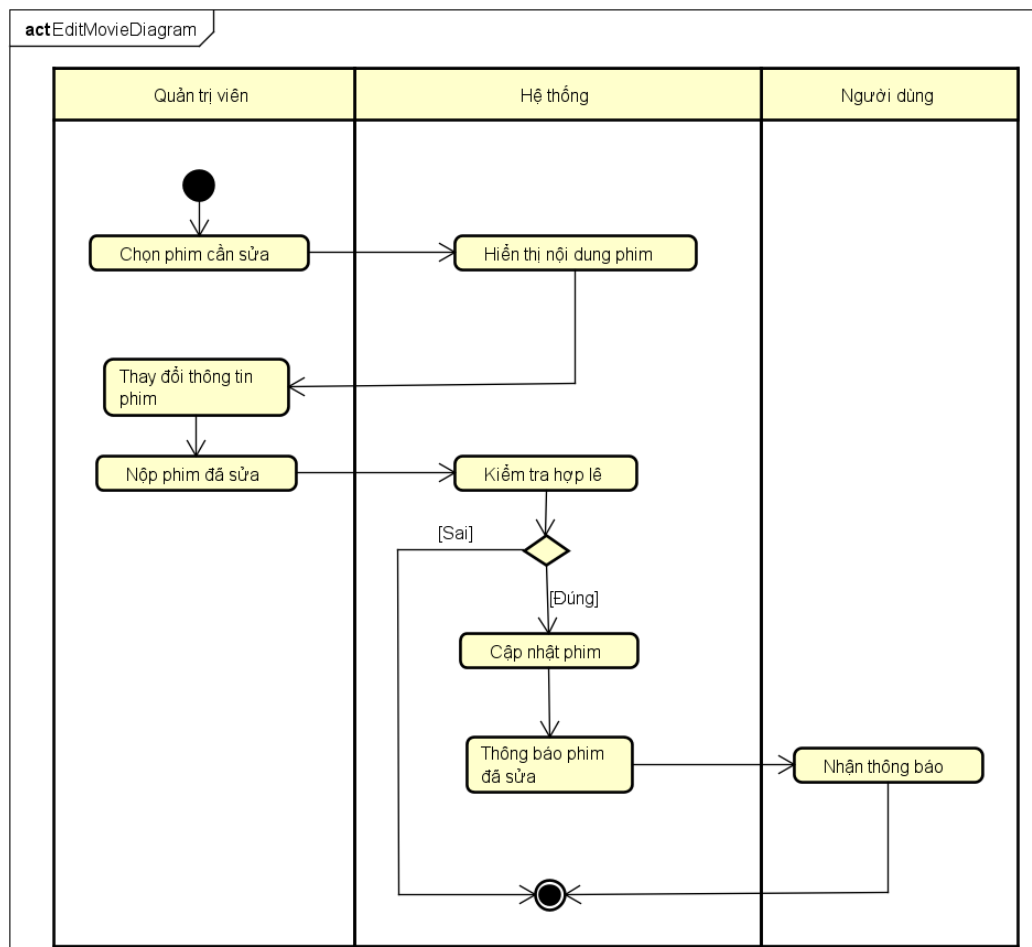


Hình 4 Phân rã Usecase “Quản lý series”

STT	Mã Usecase	Tên UseCase	Mô tả Usecase	Tác nhân tương tác
1	UC06-1	Thống kê series	Thống kê danh sách series phim hiện có	Quản trị viên

2	UC06-2	Thêm series	Thêm mới một series phim từ danh sách các phim hiện có	Quản trị viên
3	UC06-3	Sửa series	Thêm, sửa, xóa các bộ phim trong series	Quản trị viên
4	UC06-4	Xóa series	Xóa series phim được chọn	Quản trị viên

2.2.5 Quy trình nghiệp vụ



Hình 5 Quy trình nghiệp vụ sửa thông tin phim

Mô tả quy trình sửa thông tin phim: Quản trị viên khi sửa thông tin một bộ phim thì sau khi sửa xong thì phải thông báo đến các người dùng đã tương tác với bộ phim (thích, bình luận) về sự thay đổi đó.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case “Đăng nhập”

Mã Use case	UC01	Tên Use case	Đăng nhập
Tác nhân	Khách		
Tiền điều kiện	Đã vào website		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Khách	Chọn chức năng đăng nhập
	2	Hệ thống	Hiển thị giao diện trang đăng nhập
	3	Khách	Nhập thông tin cần thiết để đăng nhập (mô tả ở phía dưới *)
	4	Khách	Gửi yêu cầu đăng nhập lên hệ thống
	5	Hệ thống	Kiểm tra xem khách đã nhập đủ các trường thông tin bắt buộc hay chưa
	6	Hệ thống	Kiểm tra tính hợp lệ của các thông tin trong yêu cầu
	7	Hệ thống	Chuyển đến màn hình chính với vai trò “admin” hoặc “user”
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: cần nhập đầy đủ các trường thông tin bắt buộc
	6a	Hệ thống	Thông báo lỗi: thông tin về tài khoản hoặc mật khẩu không có hoặc không tồn tại
	6b	Hệ thống	Thông báo lỗi: tài khoản hiện tại đang

	bị khóa
Hậu điều kiện	Hệ thống tạo và ghi lại phiên đăng nhập của người dùng (khách sau khi đăng nhập thành công vào hệ thống)

Bảng 1 Đặc tả usecase “Đăng nhập”

2.3.2 Đặc tả use case “Thêm phim”

Mã Use case	UC05-2	Tên Use case	Thêm phim
Tác nhân	Quản trị viên		
Tiền điều kiện	Đang ở trang thêm phim		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Quản trị viên	Chọn chức năng thêm phim
	2.	Hệ thống	Hiển thị giao diện thêm phim(gồm các trường tên phim, tác giả, năm sáng tác, ...)
	3.	Quản trị viên	Điền thông tin của phim
	4.	Quản trị viên	Yêu cầu thêm phim
	5.	Hệ thống	Kiểm tra xem các trường thông tin bắt buộc đã được nhập đầy đủ chưa
	6.	Hệ thống	Kiểm tra xem các trường thông tin có hợp lệ
	7.	Hệ thống	Lưu lại thông tin của phim
	8.	Hệ thống	Thông báo thêm phim thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a.	Hệ thống	thông báo lỗi: Hiển thị thông báo các trường thông tin còn thiếu

	6a.	Hệ hống	thông báo lỗi: Hiển thị thông báo các trường thông tin không hợp lệ
	7a.	Hệ thống	thông báo lỗi: Thông báo đã có lỗi xảy ra trong quá trình lưu
Hậu điều kiện		Cập nhật lại danh sách phim	

Bảng 2 Đặc tả use case “Thêm phim”

2.3.3 Đặc tả use case “Sửa phim”

Mã Use case	UC05-3	Tên Use case	Sửa phim
Tác nhân	Quản trị viên		
Tiền điều kiện	Đang xem danh sách phim		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Quản trị viên	Chọn phim cần cập nhật thông tin
	2.	Hệ thống	Hiển thị giao diện cập nhật thông tin phim
	4.	Quản trị viên	Điền đầy đủ thông tin cần cập nhật
	5.	Quản trị viên	Nhấn nút “Lưu”
	6.	Hệ thống	Kiểm tra xem các trường thông tin nhập vào đã hợp lệ hay chưa
	.	Hệ thống	Cập nhật thông tin của phim lên hệ thống
	8.	Hệ thống	Trả lại thông báo cập nhật thành công cho quản trị viên
	9.	Hệ thống	Gửi thông báo phim đã cập nhật đến người dùng tương tác

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	7a.	Hệ thống	thông báo lỗi: Hiển thị thông báo các trường thông tin còn thiếu hoặc không hợp lệ
	8a.	Hệ thống	thông báo lỗi: Xảy ra lỗi khi cập nhật thông tin xe lên hệ thống
Hậu điều kiện	Cập nhật danh sách các phim		

Bảng 3 Đặc tả use case “Sửa phim”

2.3.4 Đặc tả usecase “Tạo series”

Mã Use case	UC06-2	Tên Use case	Sửa phim
Tác nhân	Quản trị viên		
Tiền điều kiện	Đang xem danh sách series		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Quản trị viên	Nhấn nút “Tạo series phim”
	2	Hệ thống	Hiển thị giao diện tạo series phim
	3.	Quản trị viên	Điền đầy đủ thông tin về series phim
	4.	Quản trị viên	Nhấn nút “Nộp”
	5.	Hệ thống	Kiểm tra xem các trường thông tin nhập vào đã hợp lệ hay chưa
	6.	Hệ thống	Cập nhật phim đã được chọn trong bộ series
	7.	Hệ thống	Tạo series từ các phim đã chọn
	8.	Hệ thống	Hiển thị thông báo tạo series thành công

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a.	Hệ thống	thông báo lỗi: Hiển thị thông báo các trường thông tin còn thiếu hoặc không hợp lệ
	6a.	Hệ thống	thông báo lỗi: Lỗi cập nhật
	7a.	Hệ thống	thông báo lỗi: Lỗi tạo series
Hậu điều kiện	Cập nhật lại danh sách series		

Bảng 4 Đặc tả use case “Thêm series”

2.3.5 Đặc tả use case “Tìm phim”

Mã Use case	UC08	Tên Use case	Thích kiểm phim
Tác nhân	Khách, người dùng		
Tiền điều kiện	Đã vào website		
Luồng sự kiện chính (Thành công)	STT	Thực hi n bởi	Hành động
	1.	Khách, người dùng	Chọn hình thức tìm kiếm (theo thanh tìm kiếm, theo thể loại)
	2.	Hệ thống	Truy vấn cơ sở dữ liệu
	3.	Hệ thống	Hiển thị kết quả phim tìm thấy
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
Hậu điều kiện	Không		

Bảng 5 Đặc tả use case “Tìm phim”

2.3.6 Đặc tả use case “Thích phim”

Mã Use case	UC10	Tên Use case	Thích phim
Tác nhân	Người dùng		

Tiền điều kiện	Đang xem phim		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhấn nút thích phim
	2.	Hệ thống	Kiểm tra quyền có đủ không?
	3.	Hệ thống	Cập nhật lại lượt thích của bộ phim
	4.	Hệ thống	Gửi thông báo đến những người dùng đã tương tác với bộ phim
	5.	Hệ thống	Hiện thị kết quả của thích phim
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	thông báo lỗi: Không đủ quyền
Hậu điều kiện	Không		

Bảng 6 Đặc tả use case “Thích phim”

2.4 Yêu cầu phi chức năng

Hệ thống phải đảm bảo hoạt động tốt trên bất cứ nền tảng website nào (Google Chrome, Cốc cốc,...), tự động cập nhật database. Hệ thống phải hoạt động liên tục 24/7, hoạt động 200 giờ liên tục không lỗi. Hệ thống có thể hoạt động trở lại bình thường trong vòng 2 giờ sau khi xảy ra lỗi. Thời gian đáp ứng tối đa của hệ thống (nếu không được nêu rõ) là 1 giây lúc bình thường hoặc 2 giây lúc cao điểm. Hỗ trợ 100 người truy suất cơ sở dữ liệu cùng lúc. Thời gian đáp ứng các chức năng không quá một giây.

Hệ thống phải đảm bảo các tác nhân phải chạy đúng quyền được giao, không được phép thay đổi cơ sở dữ liệu mà chúng không có đủ thẩm quyền. Đồng thời, cũng phải bảo mật thông tin người dùng.

Hệ thống phải đảm bảo giao diện dễ sử dụng với người dùng, không được sử dụng từ ngữ hình ảnh liên quan đến tôn giáo, chính trị, sắc tộc.

Chương 3 Công nghệ sử dụng

Hệ thống được thiết kế dựa trên mô hình client/server.

Phía client, công nghệ được sử dụng là ReactJS, MUI, Redux, Formik, HTML, CSS... để tạo dựng vào xây dựng giao diện cho các tác nhân: khách, người dùng, quản trị viên.

React (hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó khuyến khích việc tạo ra các thành phần giao diện người dùng có thể tái sử dụng, hiển thị dữ liệu thay đổi theo thời gian. Rất nhiều người sử dụng React làm V trong MVC. React trừu tượng hóa DOM khỏi bạn, cung cấp mô hình lập trình đơn giản hơn và hiệu suất tốt hơn. React cũng có thể hiển thị trên máy chủ bằng Node và nó có thể cung cấp năng lượng cho các ứng dụng gốc bằng React Native. React triển khai luồng dữ liệu phản ứng một chiều, điều này giúp giảm bớt phần soạn sẵn và dễ lý luận hơn so với ràng buộc dữ liệu truyền thống. React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo. Có hai khái niệm cơ bản trong React là: state và props. Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI. Về cơ bản, việc xây dựng một ứng dụng MVC phía client với ràng buộc dữ liệu 2 chiều (2 way data-binding) là khá đơn giản. Tuy nhiên nếu dự án ngày càng mở rộng, nhiều tính năng hơn, làm cho việc bảo trì dự án gặp khó khăn, đồng thời hiệu năng cũng bị giảm. Bạn cũng có thể giải quyết vấn đề đó bằng các thư viện khác như Backbone.js hay Angular.js, tuy nhiên bạn sẽ thấy các hạn chế của chúng khi dự án của bạn ngày càng lớn. React ra đời sau AngularJS, nó sinh ra để dành cho các ứng dụng lớn dễ dàng quản lý và mở rộng. Mục tiêu chính của React là nhanh, đơn

giản, hiệu năng cao và dễ dàng mở rộng. Trong hệ thống này, React sẽ giúp phân tách các thành phần giao diện để dễ quản lý tổ chức và tái sử dụng.

Thư viện MUI: vào tháng 9 năm 2021, công ty trước đây có tên là Material-UI đã đổi tên thành MUI. Thay đổi này chủ yếu xảy ra do nhiều người không thể phân biệt Material-UI với Material Design (một hệ thống thiết kế). MUI cung cấp một thư viện mạnh mẽ, có thể tùy chỉnh và có thể truy cập được gồm các thành phần cơ bản và nâng cao. Trong hệ thống của mình, em sử dụng MUI để xây dựng ứng dụng React nhanh hơn.

Formik được sử dụng trong hệ thống làm nhiệm vụ nhận các giá trị trong và ngoài trạng thái biểu mẫu, xác thực và thông báo lỗi, xử lý gửi biểu mẫu, bằng cách sắp xếp tất cả những điều trên vào một nơi, Formik sẽ giữ mọi thứ có tổ chức - giúp việc kiểm tra, cấu trúc lại và lập luận về các biểu mẫu của bạn trở nên dễ dàng. Yup được sử dụng kết hợp với Formik giúp xác thực các trường trong biểu mẫu một cách dễ dàng.

Hệ thống có sử dụng Redux(cụ thể là Redux toolkit) để quản lý các state trong hệ thống. Hệ thống cũng có thể không sử dụng Redux. Nhưng bây giờ hãy tưởng tượng những gì xảy ra khi một trạng thái phải được chia sẻ giữa các thành phần cách xa nhau trong cấu trúc thành phần. Trạng thái phải được truyền từ thành phần này sang thành phần khác cho đến khi đến nơi cần thiết. Về cơ bản, trạng thái sẽ phải được nâng lên thành phần cha gần nhất và tiếp theo cho đến khi nó đến một thành phần chung cho cả hai thành phần cần trạng thái, và sau đó nó được truyền lại. Điều này làm cho trạng thái khó duy trì và dữ liệu được truyền đến các thành phần không cần sử dụng nó! Rõ ràng rằng việc quản lý trạng thái trở nên lộn xộn khi ứng dụng trở nên phức tạp. Đây là lý do tại sao bạn cần một công cụ quản lý trạng thái như Redux giúp duy trì các trạng thái này dễ dàng hơn. Cách Redux hoạt động rất đơn giản. Có một "store" trung tâm chứa toàn bộ trạng thái của ứng dụng. Mỗi thành phần có thể truy cập trạng thái được lưu trữ mà không phải gửi từ thành phần này sang thành phần khác. Có ba phần xây dựng: actions, store, and reducers. Actions nói một cách đơn giản, action là sự kiện. Chúng là cách duy nhất bạn có thể gửi dữ liệu từ ứng dụng của mình đến "store" Redux. Dữ liệu có thể là từ các tương tác của người dùng, các lệnh gọi API hoặc là gửi form. Reducers là các hàm thuần túy lấy trạng thái hiện tại của ứng dụng, thực hiện một hành động và trả về trạng thái mới. Các trạng thái này được lưu trữ dưới dạng đối tượng và chúng xác định trạng thái của ứng dụng thay đổi như thế nào để đáp ứng với hành động được gửi đến "store".Store giữ trạng thái ứng dụng. Chỉ có một "store" trong bất kỳ ứng dụng Redux nào. Bạn có thể truy cập trạng thái được lưu trữ, cập nhật trạng thái và đăng ký hoặc hủy đăng ký "listeners" thông qua các phương thức trợ giúp.

Hệ thống sử dụng thư viện moment để tùy chỉnh định dạng ngày tháng.

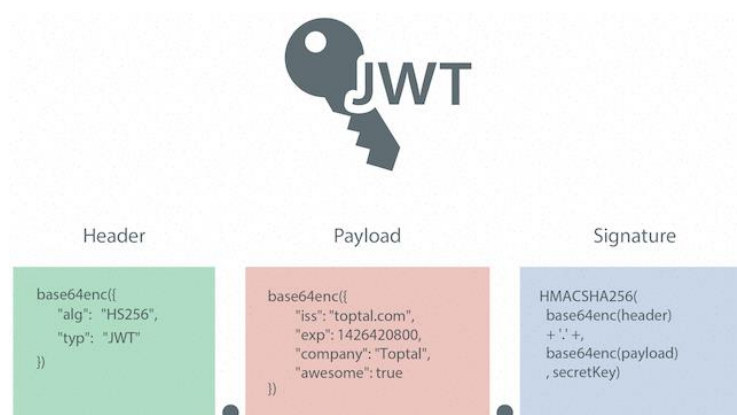
Phía server, hệ thống sử dụng NodeJS, express server, multer, jwt, argon2, redis,... và cơ sở dữ liệu phi quan hệ để xử lý các yêu cầu và phản hồi yêu cầu của người dùng.

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng. NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux. Có thể sử dụng các ngôn ngữ C++, C#, Python nhưng NodeJS đem lại những ưu điểm. NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh. Nếu bạn cần hỗ trợ gì về NodeJS, sẽ nhanh chóng có người hỗ trợ bạn. Tốc độ xử lý nhanh. Nhờ cơ chế xử lý bất đồng bộ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào. Dễ dàng mở rộng. Nếu bạn có nhu cầu phát triển website thì tính năng dễ dàng mở rộng của NodeJS là một lợi thế cực kỳ quan trọng.

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng. Thiết lập các lớp trung gian để trả về các HTTP request. Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL. Cho phép trả về các trang HTML dựa vào các tham số.

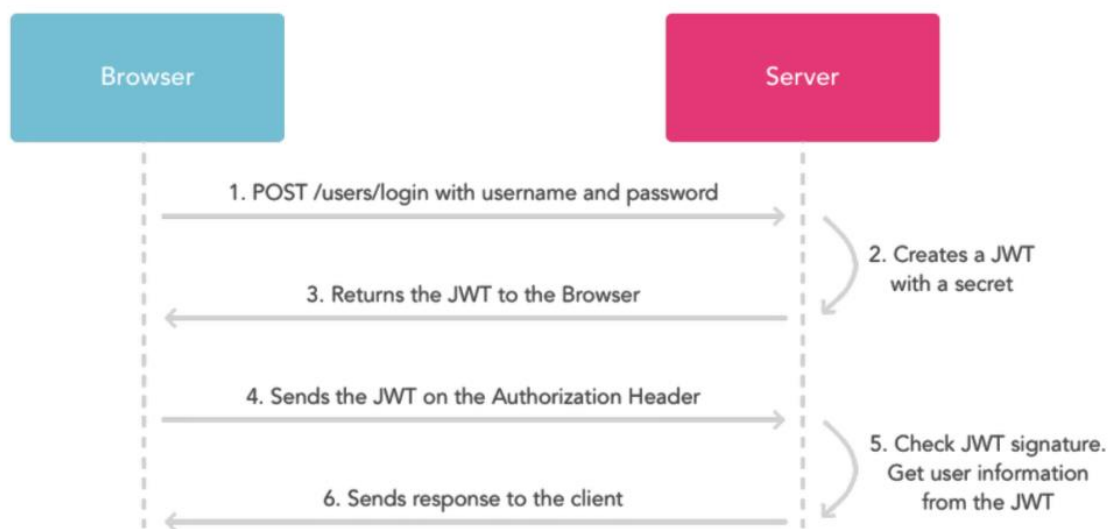
Multer là một middleware cho Express và Nodejs giúp dễ dàng xử lý dữ liệu multipart/form-data khi người dùng upload file.

Hệ thống sử dụng JWT với 2 quyền user và admin giúp để phân biệt yêu cầu gửi từ client đến server xác thực quyền và bảo mật các tài nguyên của hệ thống. JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA. JWT là sự kết hợp (bởi dấu .) một Object Header dưới định dạng JSON được encode base64, một payload object dưới định dạng JSON được encode base64 và một Signature cho URI cũng được mã hóa base64.



Hình 6 Cấu trúc JWT

Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA). Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims. Reserved claims: Đây là một số metadata được định nghĩa trước, trong đó một số metadata là bắt buộc, số còn lại nên tuân theo để JWT hợp lệ và đầy đủ thông tin: iss (issuer), iat (issued-at time) exp (expiration time), sub (subject), aud (audience), jti (Unique Identifier cho JWT.) ... Public Claims - Claims được cộng đồng công nhận và sử dụng rộng rãi. Private Claims - Claims tự định nghĩa (không được trùng với Reserved Claims và Public Claims), được tạo ra để chia sẻ thông tin giữa 2 parties đã thỏa thuận và thống nhất trước đó. Signature chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload.



Hình 7 JWT hoạt động

Argon2 là một dạng mã băm để mã hóa mật khẩu của người dùng trong hệ thống. Argon2 là giành chiến thắng trong cuộc thi Password Hashing Competition vào than 6 năm 2015. Nó được thiết kế bởi Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich từ đại học Luxembourg. Argon2 nhận vào 6 tham số: password, salt, memory cost (the memory usage of the algorithm), time cost (the execution time of the algorithm and the number of iterations), parallelism factor (the number of parallel threads), hash length.

Redis (Remote Dictionary Server) là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker. Nó là hệ thống lưu trữ dữ liệu với dạng KEY-VALUE rất mạnh mẽ và phổ biến hiện nay. Redis nổi bật bởi việc hỗ trợ nhiều cấu trúc dữ liệu cơ bản như: hash, list, set, sorted set,

string... Tất cả dữ liệu được ghi và lưu trên ram, do đó tốc độ đọc ghi dữ liệu rất là nhanh. Trong hệ thống này em sử dụng Redis (redis chạy trên docker) để làm 2 nhiệm vụ: caching (bộ nhớ đệm) để lưu trữ danh sách các thể loại của phim không cần phải truy xuất qua database giúp tăng hiệu suất cho chương trình. Counter (bộ đếm) để lưu trữ các token khi người dùng reset mật khẩu.

Hệ thống sử dụng Socket.IO để realtime cho việc tiếp nhận và truyền đi thông báo, tương tác của người dùng với hệ thống. Các kết nối sẽ được thiết lập với WebSocket, cung cấp một kênh giao tiếp chi phí thấp giữa máy chủ và máy khách. Trong trường hợp kết nối WebSocket không thể thực hiện được, nó sẽ trở lại trạng thái bỏ phiếu dài HTTP. Và nếu kết nối bị mất, máy khách sẽ tự động cố gắng kết nối lại. Mở rộng quy mô đến nhiều máy chủ và gửi sự kiện đến tất cả các máy khách được kết nối một cách dễ dàng.

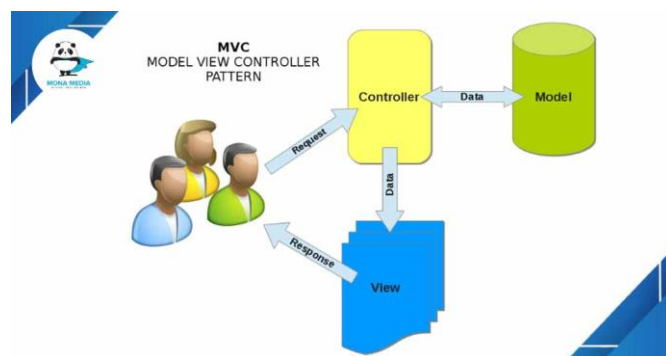
Chương 4 Phát triển và triển khai ứng dụng

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

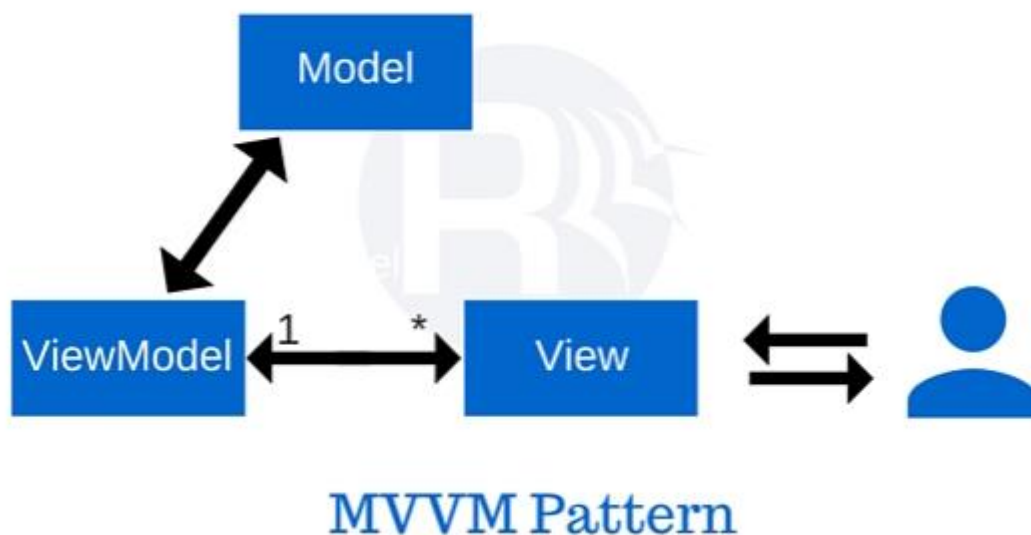
Hệ thống được triển khai theo kiến trúc client/server dựa theo mô hình 3 lớp MVVM. Trước hết, cần hiểu MVC là gì vì MVVM là dẫn xuất của MVC

MVC từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình. Model (M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,... View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi. Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller. Controller (C): Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.



Hình 8 Mô hình MVC

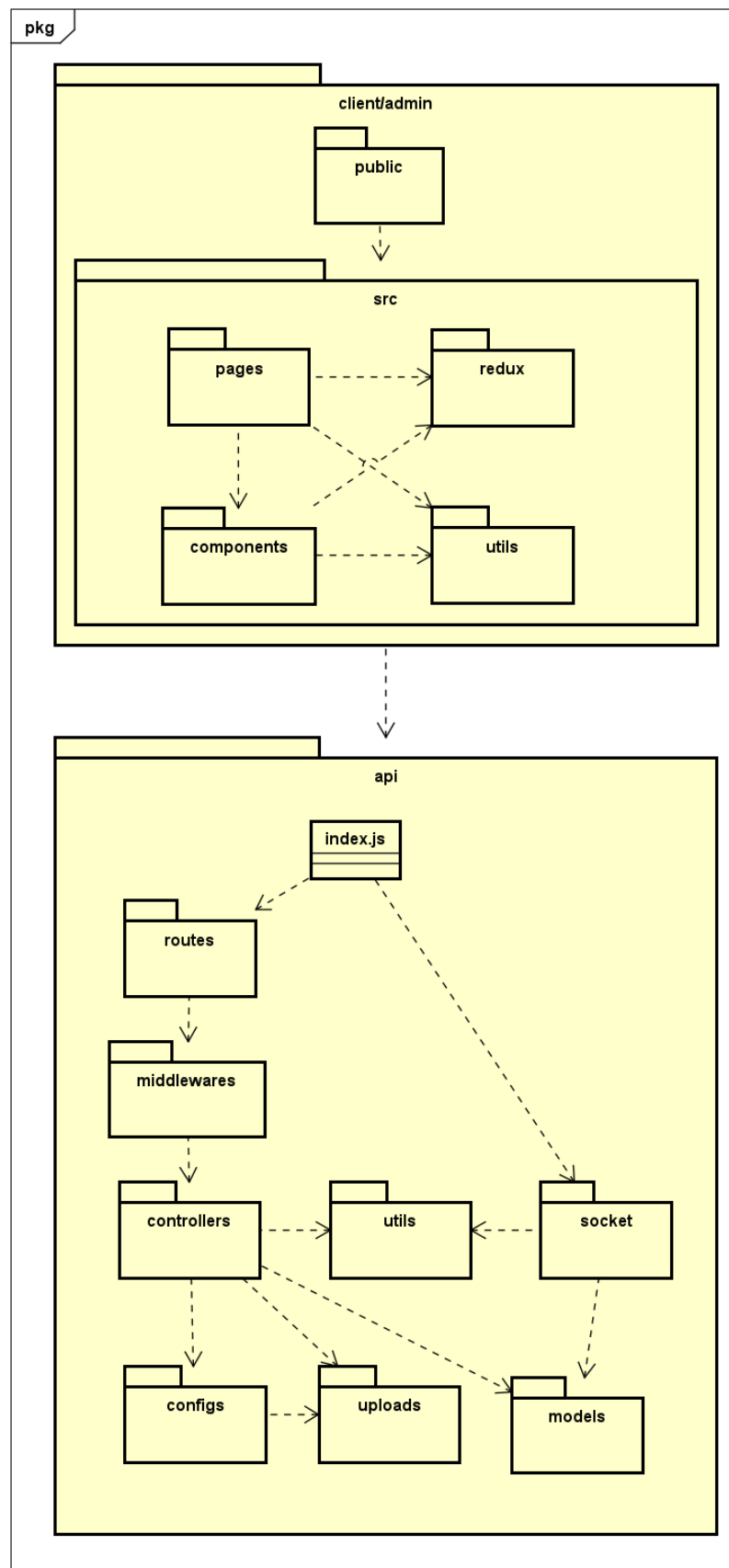
Đối với MVVM, MVVM hỗ trợ cơ chế two-way binding giữa View và ViewModel. Điều này cho phép tự động đồng bộ các thay đổi giữa 2 thành phần. Nhìn chung, ViewModel tối ưu sử dụng pattern Observer để thông báo sự thay đổi trong ViewModel sang Model. View: Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command. Model: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự. ViewModel: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command. ViewModel không hề biết gì về View, một ViewModel có thể được sử dụng cho nhiều View (one-to-many). ViewModel sử dụng Observer design pattern để liên lạc với View (thường được gọi là binding data, có thể là 1 chiều hoặc 2 chiều tùy nhu cầu ứng dụng).



Hình 9 Mô hình MVVM

Trong hệ thống này, phần V (View) sử dụng ReactJS chia ứng dụng thành pages (các trang) được tách riêng đảm bảo tính coupling. Trong một page (trang) được chia thành các component (thành phần) để dễ tái sử dụng. Phần VM (ViewModel) bao gồm các gói routes, controllers, socket thực hiện việc tiếp nhận dữ liệu và phản hồi dữ liệu cho người dùng. Phần M (Model) nằm trong gói model cung cấp các phương thức CRUD được gọi bởi tầng VM (ViewModel).

4.1.2 Thiết kế tổng quan

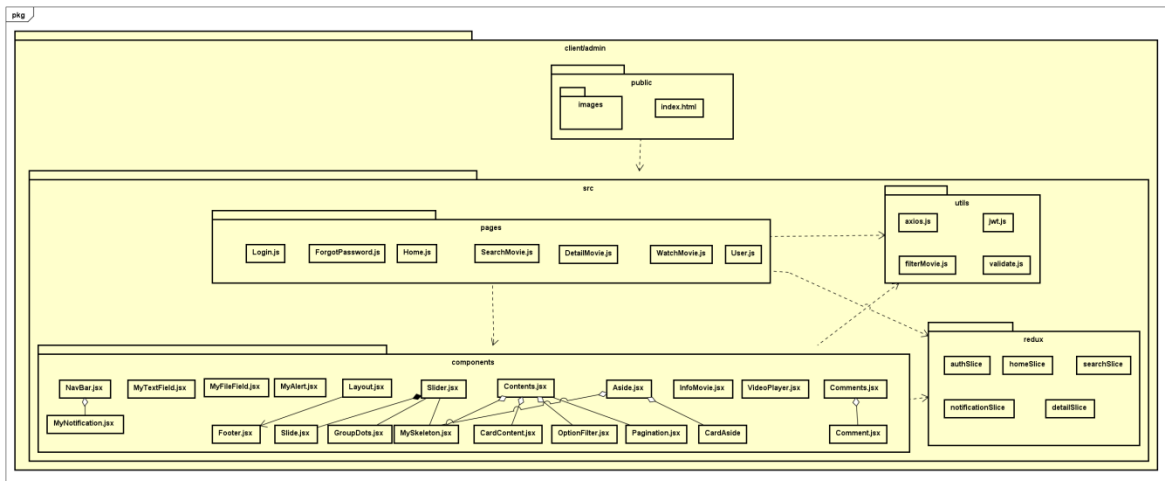


Hình 10 Tổng quan gói

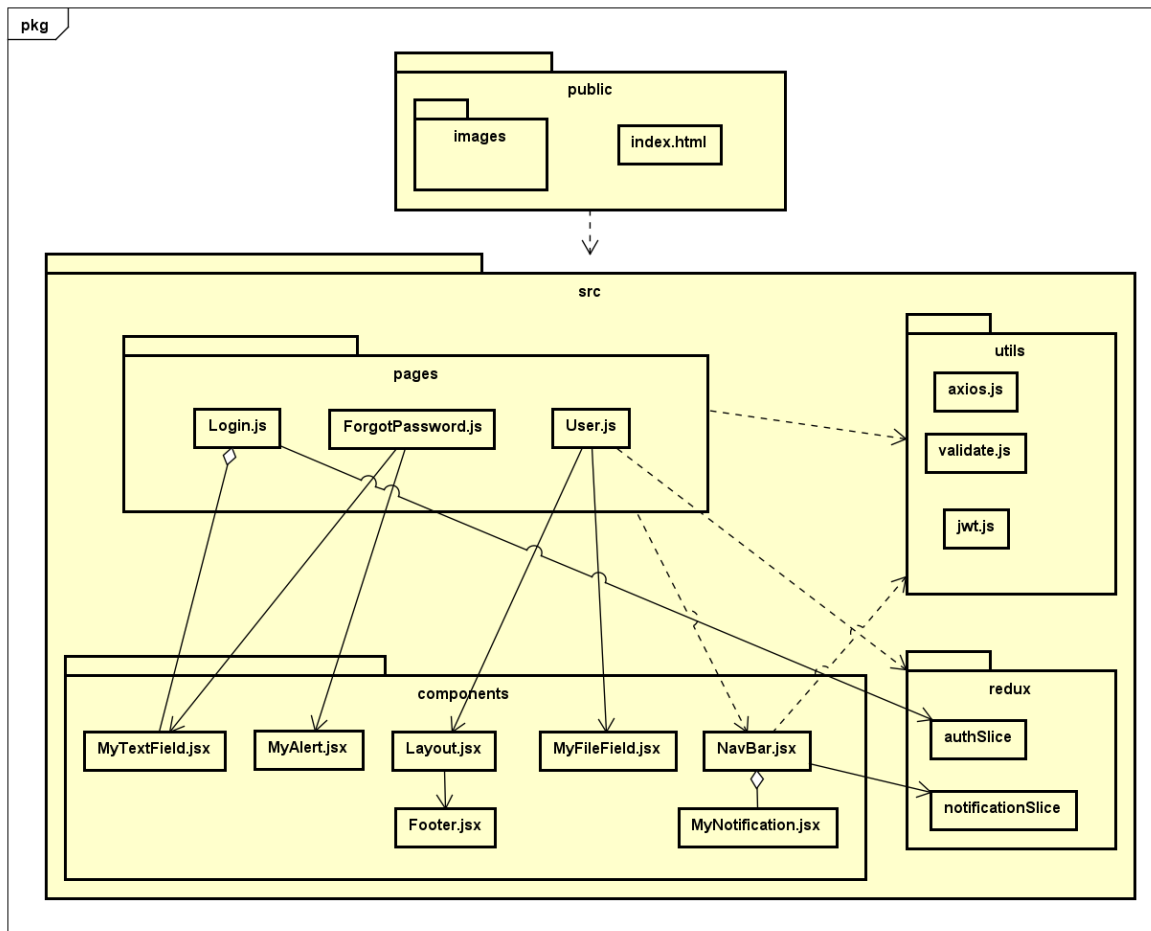
STT	Gói	Mô tả
1	public	Chứa file khởi tạo và các thư mục công khai trong hệ thống
2	pages	Chứa các trang của hệ thống
3	components	Chứa các thành phần của hệ thống có thể tái sử dụng
4	redux	Chứa các state(trạng thái) chung của hệ thống
5	utils	Chứa các tiện ích: xác thực, xác thực quyền, xử lý lỗi, khởi tạo axios
6	routes	Xử lý các yêu cầu đầu vào
7	middlewares	Xác thực người dùng
8	controllers	Xử lý logic, nghiệp vụ
9	socket	Xử lý socket khi người dùng realtime
10	utils	Chứa các tiện ích phía server
11	configs	Khởi tạo lưu file, redis
12	models	Chứa các đối tượng tham chiếu đến database
13	uploads	Chứa các file upload từ người dùng

Bảng 7 Mô tả tổng quan gói

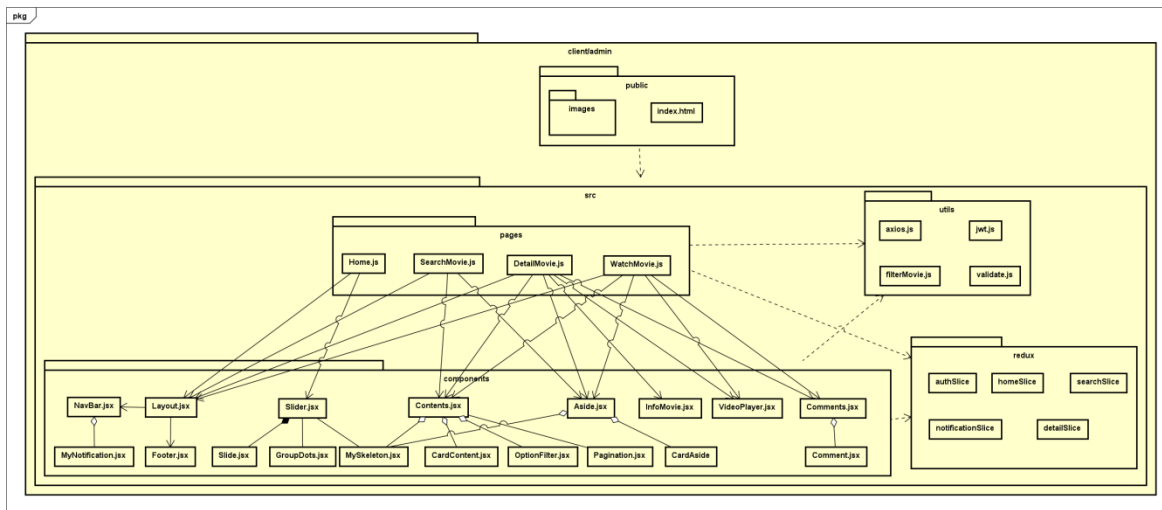
4.1.3 Thiết kế chi tiết gói



Hình 11 Thiết kế gói tổng quan Client

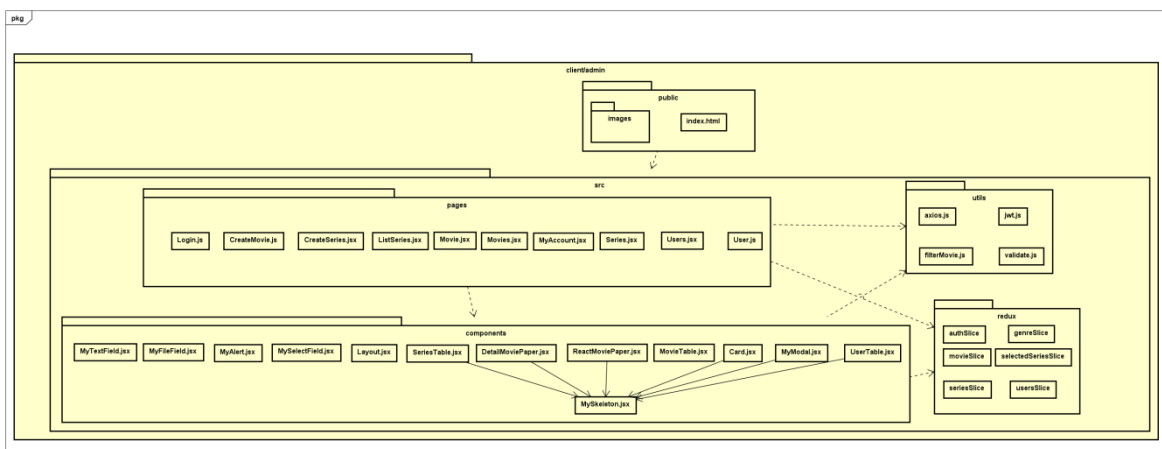


Hình 12 Thiết kế gói chi tiết người dùng Client

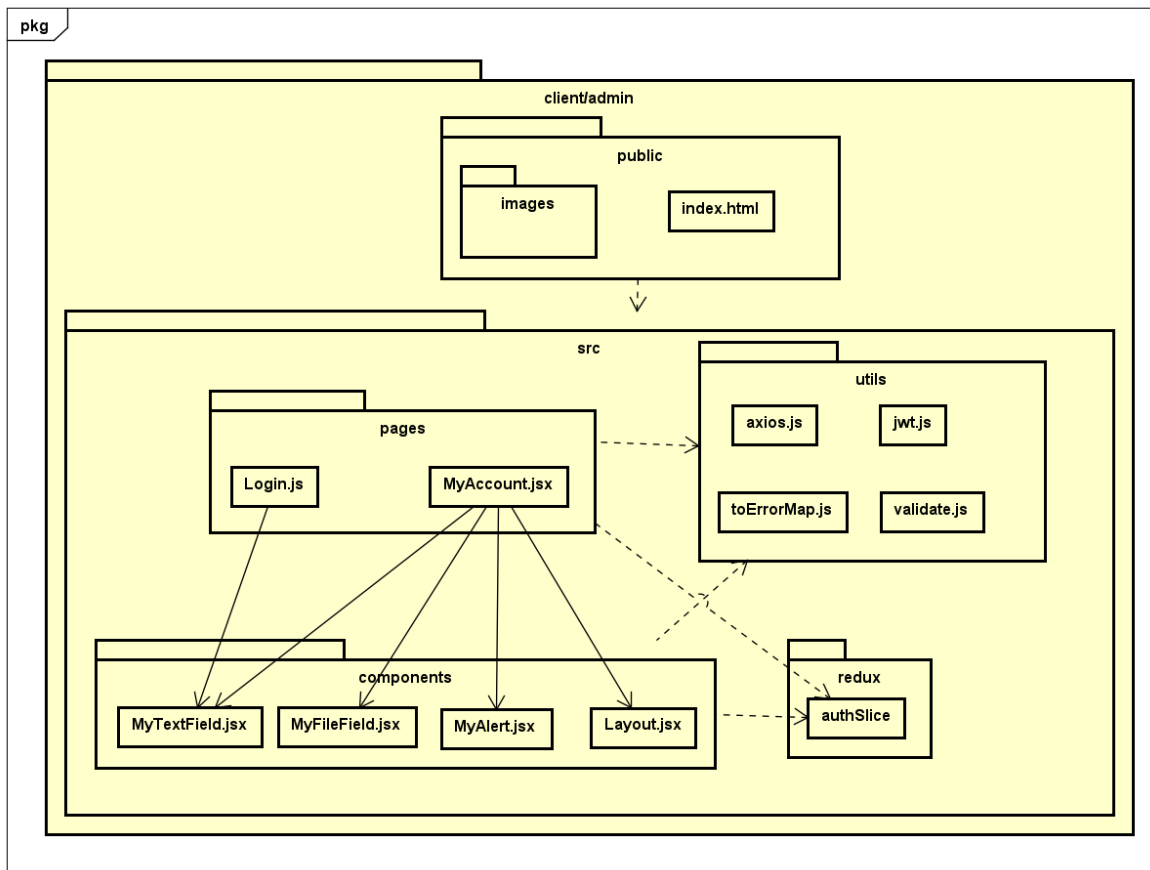


Hình 13 Thiết kế gói chi tiết movie Client

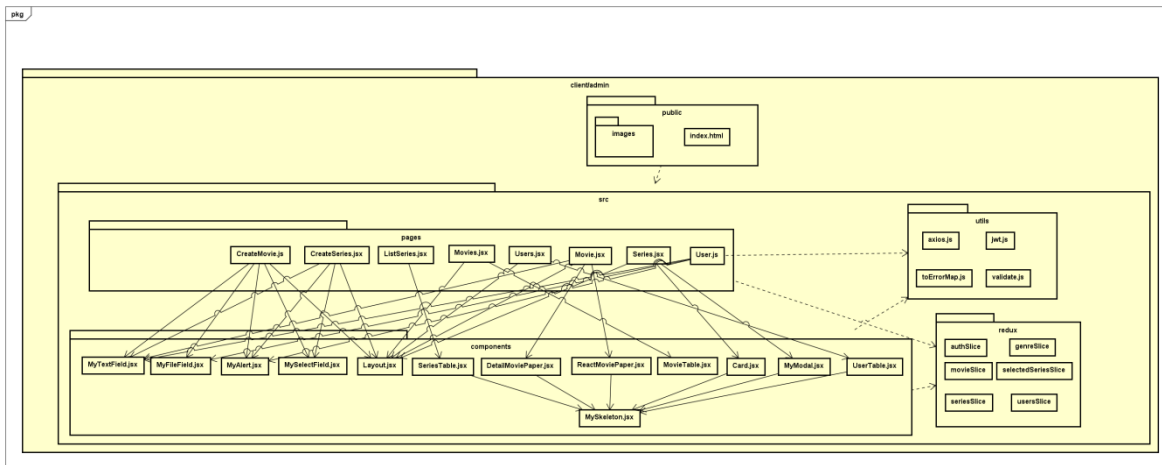
Phần thiết kế giao diện với tác nhân Khách và Người dùng: được phân thành các trang tương ứng với các class trong gói pages. Mỗi class trong gói pages này sẽ lấy 1 hay 1 số thành phần trong gói components để xây dựng lên mỗi trang mà người dùng yêu cầu. Hai gói pages và components được bổ trợ thêm 2 gói: utils(chịu trách nhiệm khởi tạo lời gọi axios gửi cho server, xác thực quyền và xác thực biểu mẫu) và gói redux lưu trữ các state(trạng thái hiện thời) của hệ thống.



Hình 14 Thiết kế tổng quan gói Admin



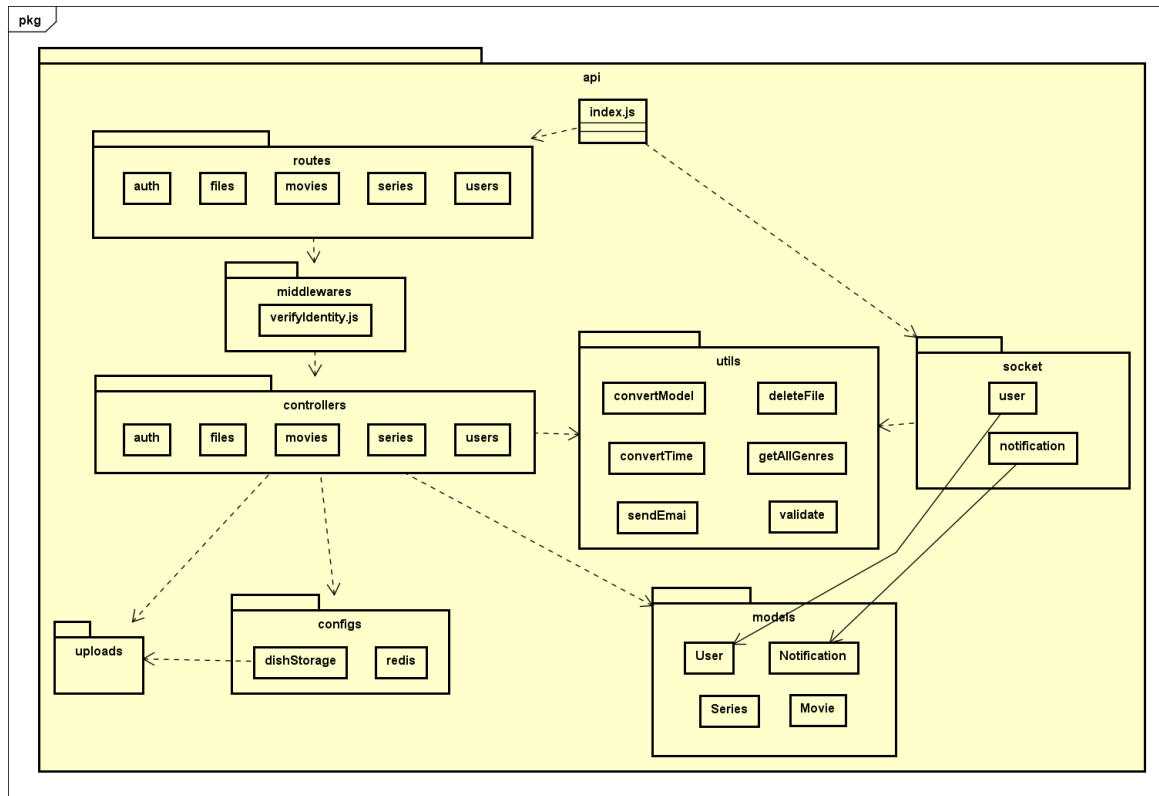
Hình 15 Thiết kế chi tiết gói Auth Admin



Hình 16 Thiết kế chi tiết gói quản lý Admin

Phần thiết kế giao diện với tác nhân Quản trị viên: được phân thành các trang tương ứng với các class trong gói `pages`. Mỗi class trong gói `pages` này sẽ lấy 1 hay 1 số thành phần trong gói `components` để xây dựng lên mỗi trang mà người dùng yêu cầu. Hai gói `pages` và `components` được bổ trợ thêm 2 gói: `utils` (chịu trách nhiệm khởi tạo lời gọi axios

gửi cho server, xác thực quyền và xác thực biểu mẫu) và gói redux lưu trữ các state (trạng thái hiện thời) của hệ thống



Hình 17 Thiết kế chi tiết gói API

Mô tả gói API: Khi nhận được một request từ phía client, đầu tiên sẽ được đi qua gói routes có nhiệm vụ định tuyến đường đi, tiếp đến sẽ đi qua các middleware trong gói middlewares nhằm xác thực quyền hạn của người dùng có được phép truy cập tài nguyên mà client yêu cầu không. Sau đó, gói controllers sẽ làm nhiệm vụ xử lý logic lấy dữ liệu từ gói models và trả về phản hồi cho client. Nhiệm vụ xử lý logic này được các gói utils, configs và uploads hỗ trợ.

Gói socket đảm nhận nhiệm vụ thực hiện realtime cho hệ thống thực hiện việc tiếp nhận tương tác từ người dùng, quản trị viên với bộ phim (thích, bình luận, sửa) và gửi thông báo đến cho các người dùng khác.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Hệ thống hỗ trợ màn hình laptop (1280 x 720), iPad (834 x 1194),... Độ phân giải 720 pixel, 1080 pixel và lớn hơn.

Đối với giao diện người dùng, hệ thống hỗ trợ giao diện tối, màu nền #323335, màu chữ trắng, các nút có màu nền là #BC2222, màu chữ là #FFFFFF, vị trí hiển thị đoạn thông báo cho người dùng ở góc trái bên dưới,...

Đối với giao diện quản trị viên, hệ thống hỗ trợ giao diện sáng màu, màu nền nút button #5743D3, chữ #FFFFFF,...

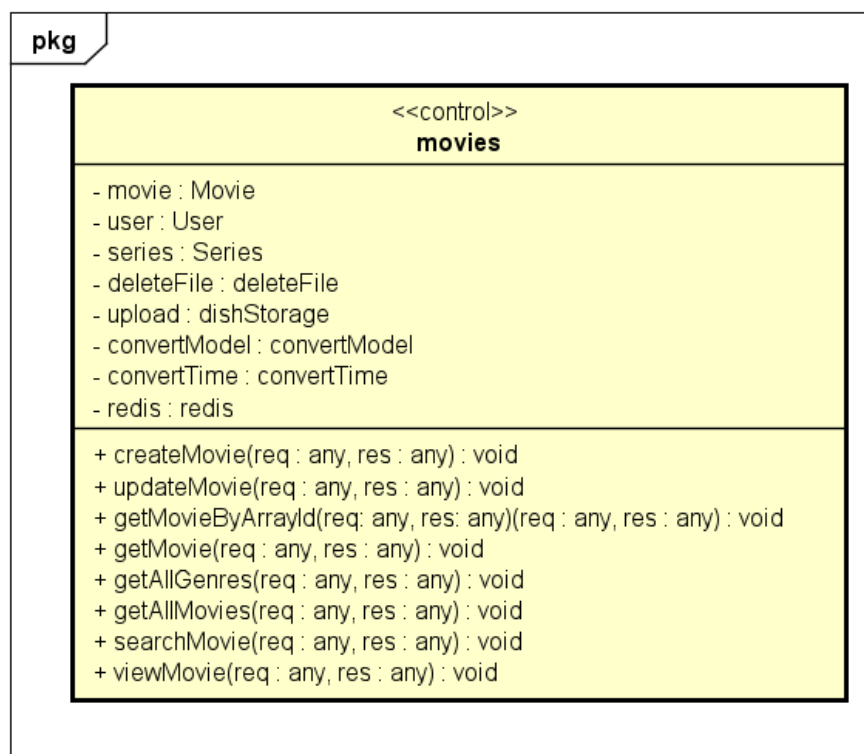


Hình 18 Thiết kế giao diện trang chủ client

Hình 19 Thiết kế giao diện admin

Có thể xem chi tiết về thiết kế giao diện client [tại đây](#) và admin [tại đây](#).

4.2.2 Thiết kế lớp



Hình 20 Chi tiết lớp MoviesController

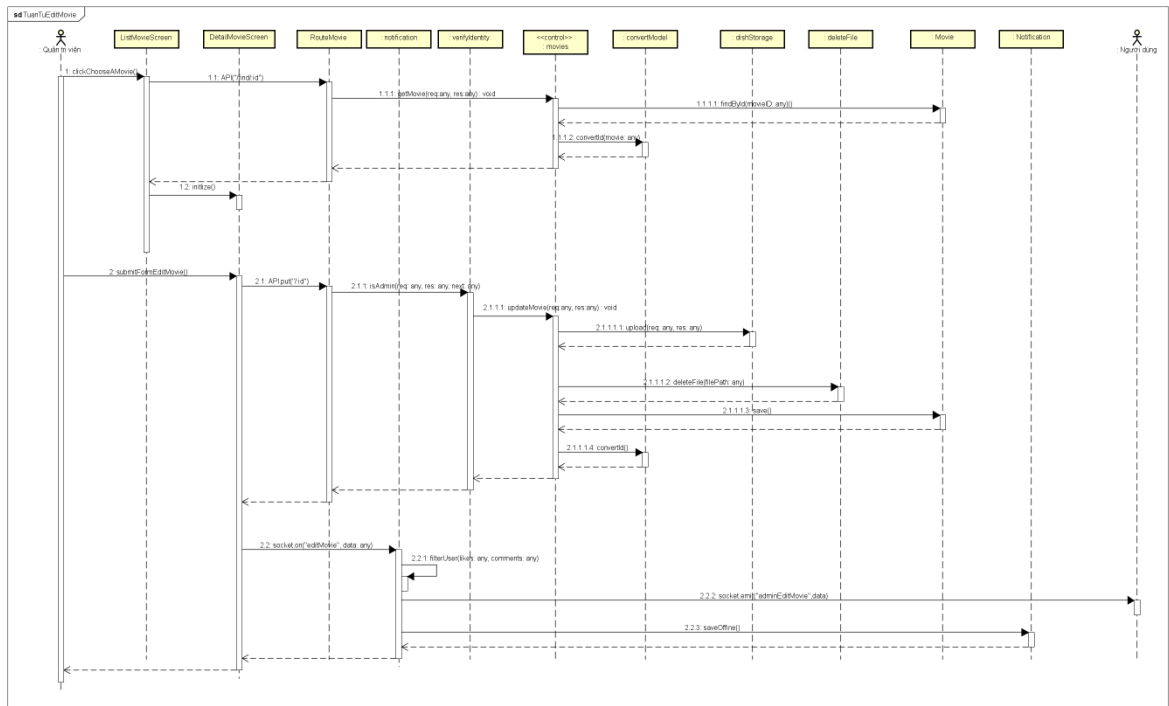
STT	Thuộc tính	Kiểu	Mô tả
1	movie	Movie	Một đối tượng của MovieModel chứa phương thức

			thao tác với bảng Movie
2	user	User	Một đối tượng của UserModel chứa phương thức thao tác với bảng User
3	series	Series	Một đối tượng của SeriesModel chứa phương thức thao tác với bảng Series
4	deleteFile	deleteFile	Xóa file theo đường dẫn
5	upload	diskStorage	Lưu file được tải lên từ phía client
6	convertModel	convertModel	Chuyển đổi một tượng hay một mảng đối tượng từ _id → id
7	convertTime	convertTime	Chuyển đổi thời gian sang giờ phút giây
8	redis	redis	Đối tượng thao tác với csdl redis

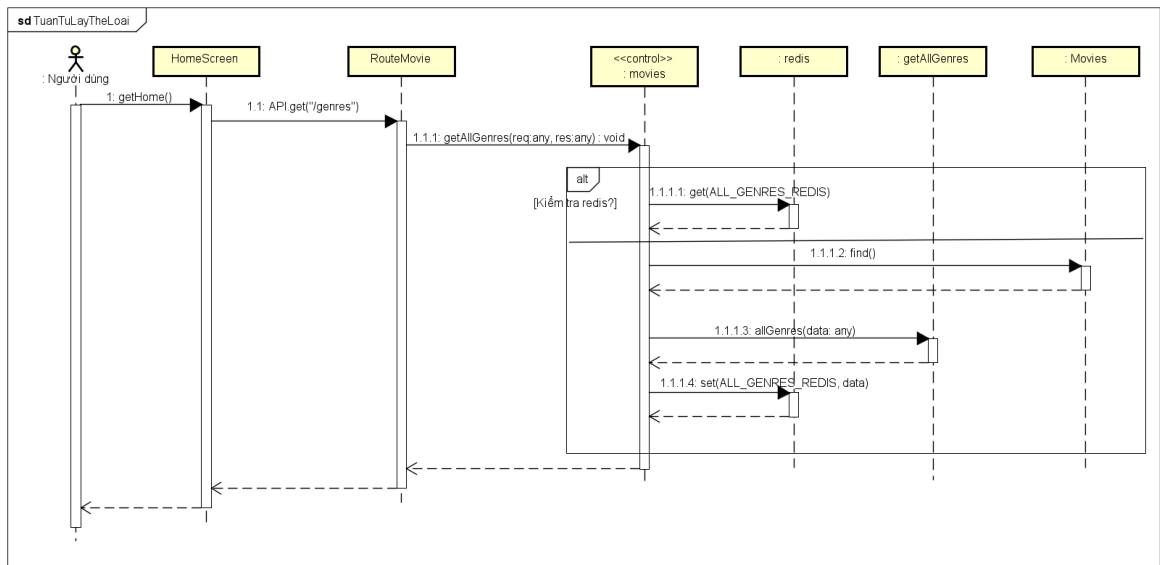
Bảng 8 Mô tả thuộc tính của lớp MovieController

STT	Phương thức	Kiểu trả về	Mô tả
1	createMovie	Void	Tạo mới một phim
2	updateMovie	Void	Sửa một bộ phim
3	getMovieByArrayId	Void	Lấy danh sách phim từ một mảng id
4	getMovie	Void	Lấy 1 bộ phim dựa theo id phim
5	getAllGenres	Void	Lấy danh sách các thể loại phim
6	searchMovie	Void	Search bộ phim theo thể loại, query, ...
7	viewMovie	Void	Tăng lượt xem của một bộ phim

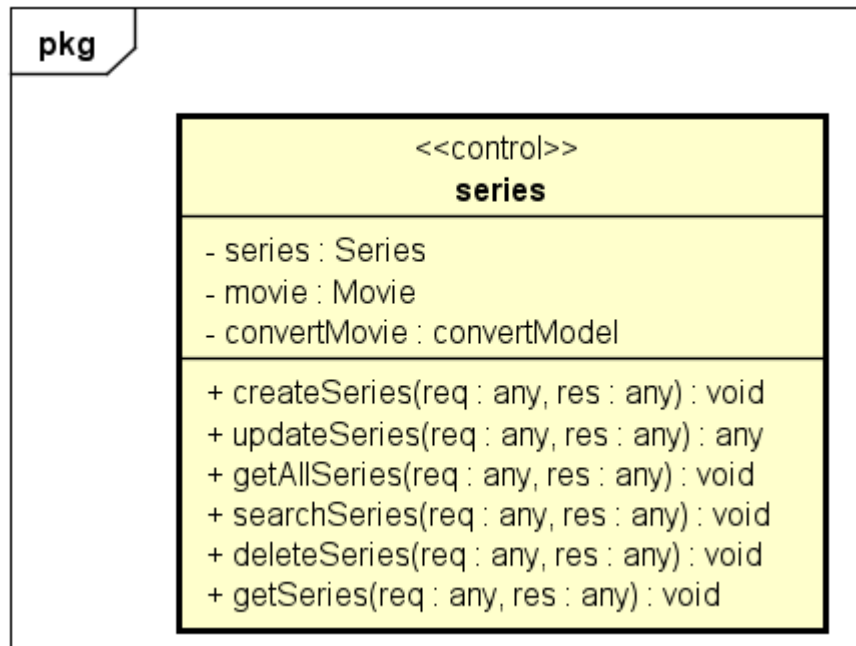
Bảng 9 Mô tả phương thức của lớp MovieController



Hình 21 Biểu đồ tuần tự “UpdateMovie”



Hình 22 Biểu đồ tuần tự “GetAllGenres”



Hình 23 Thiết kế chi tiết lớp SeriesController

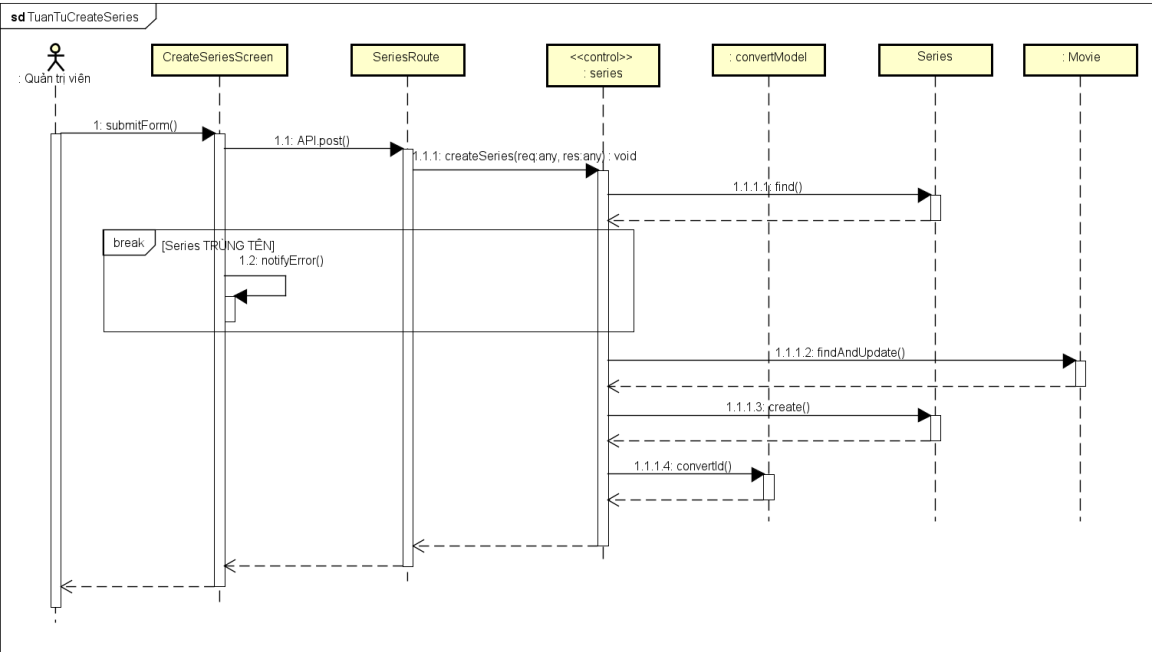
STT	Thuộc tính	Kiểu	Mô tả
1	movie	Movie	Một đối tượng của MovieModel chứa phương thức thao tác với bảng Movie
2	series	Series	Một đối tượng của SeriesModel chứa phương thức thao tác với bảng Series
3	convertModel	convertModel	Chuyển đổi một tượng hay một mảng đối tượng từ _id → id

Bảng 10 Mô tả thuộc tính của lớp SeriesController

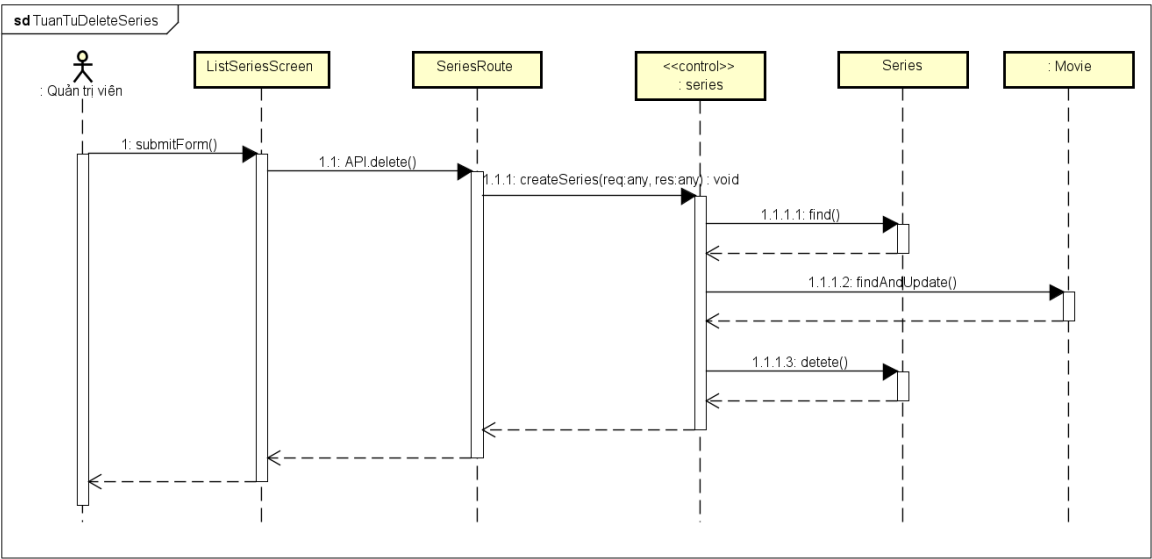
STT	Phương thức	Kiểu trả về	Mô tả
1	createSeries	Void	Tạo mới một series
2	updateSeries	Void	Sửa một series
3	getAllSeries	Void	Lấy tất cả danh sách series

4	searchSeries	Void	Tìm kiếm series
5	deletesSeries	Void	Xóa một series
6	getSeries	Void	Lấy thông tin một series

Bảng 11 Mô tả phương thức của lớp SeriesController

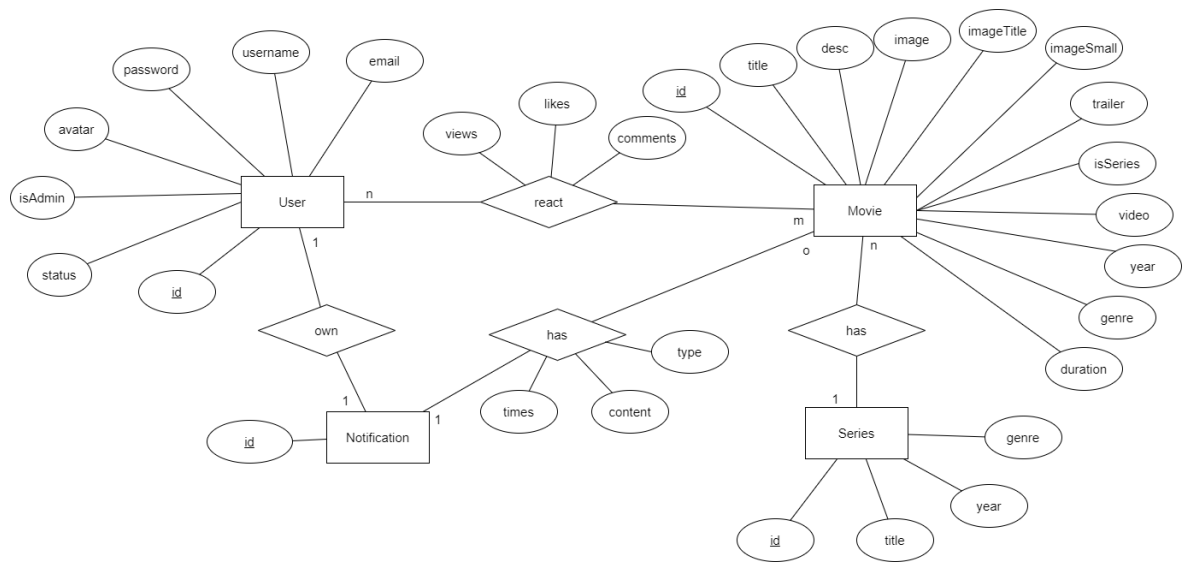


Hình 24 Biểu đồ tuần tự “CreateSeries”



Hình 25 Biểu đồ tuần tự “DeleteSeries”

4.2.3 Thiết kế cơ sở dữ liệu



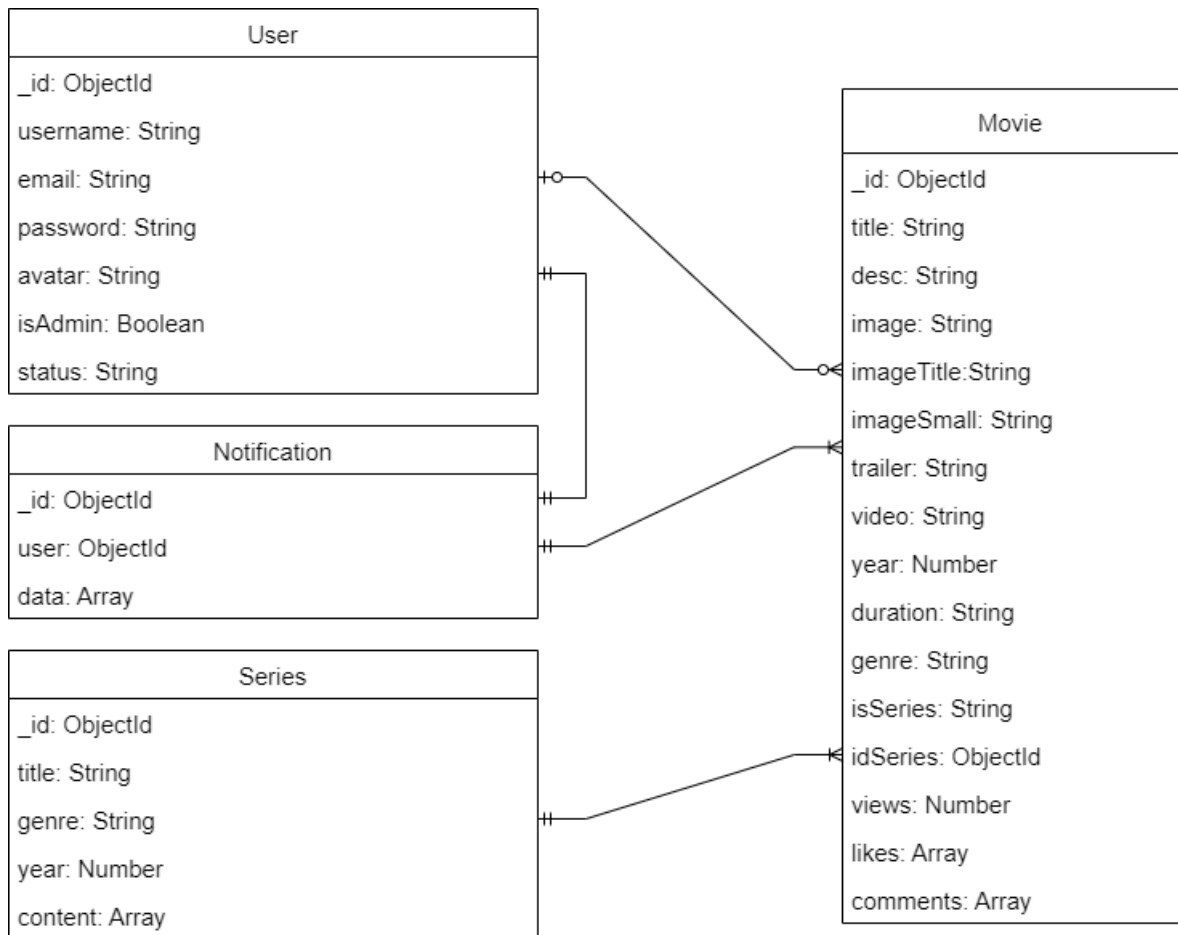
Hình 26 Sơ đồ thực thể liên kết

STT	Thực thể/ Quan hệ	Mô tả																
1	User	Lưu trữ thông tin người dùng																
		<table><tr><th>Thuộc tính</th><th>Mô tả</th></tr><tr><td>id</td><td>Khóa chính</td></tr><tr><td>email</td><td>Email của người dùng</td></tr><tr><td>username</td><td>Tên người dùng</td></tr><tr><td>password</td><td>Mật khẩu người dùng</td></tr><tr><td>avatar</td><td>ảnh đại diện người dùng</td></tr><tr><td>isAdmin</td><td>Quyền người dùng</td></tr><tr><td>status</td><td>Trạng thái người dùng</td></tr></table>	Thuộc tính	Mô tả	id	Khóa chính	email	Email của người dùng	username	Tên người dùng	password	Mật khẩu người dùng	avatar	ảnh đại diện người dùng	isAdmin	Quyền người dùng	status	Trạng thái người dùng
		Thuộc tính	Mô tả															
		id	Khóa chính															
		email	Email của người dùng															
		username	Tên người dùng															
		password	Mật khẩu người dùng															
		avatar	ảnh đại diện người dùng															
		isAdmin	Quyền người dùng															
status	Trạng thái người dùng																	
2	Movie	Lưu trữ thông tin về bộ phim																

		Thuộc tính	Mô tả
		id	Khóa chính
		title	Nhan đề của phim
		desc	Thông tin mô tả của phim
		image	Ảnh nền của phim
		imageTitle	ảnh phim kích thước vừa
		imageSmall	ảnh phim kích thước nhỏ
		trailer	Trailer của phim
		video	Video của phim
		isseries	Kiểm tra xem phim có thuộc series không
		year	Ngày sản xuất bộ phim
		genre	Thể loại của phim
		duration	Thời lượng của bộ phim
3	React	Phản hồi người dùng với 1 bộ phim	
		Thuộc tính	Mô tả
		Views	Lượt xem phim
		Likes	Số lượt thích phim
		Comments	Số lượt bình luận của phim
4	Series	Thông tin về series	
		Thuộc tính	Mô tả

		Id	Khóa chính của thực thể
		Title	Nhan đề series
		Year	Năm phát hành series
		Genre	Thể loại của series
5	Notification	Thông tin về thông báo bộ phim khi người dùng offline	
		Thuộc tính	Mô tả
		id	Khóa chính của thực thể
		user	Người dùng muốn thông báo
		times	Số lần thông báo của bộ phim
		content	Nội dung của thông báo
		type	Loại thông báo (thích, bình luận, sửa,...)

Bảng 12 Đặc tả ERD



4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 13 Danh sách thư viện và công cụ sử dụng

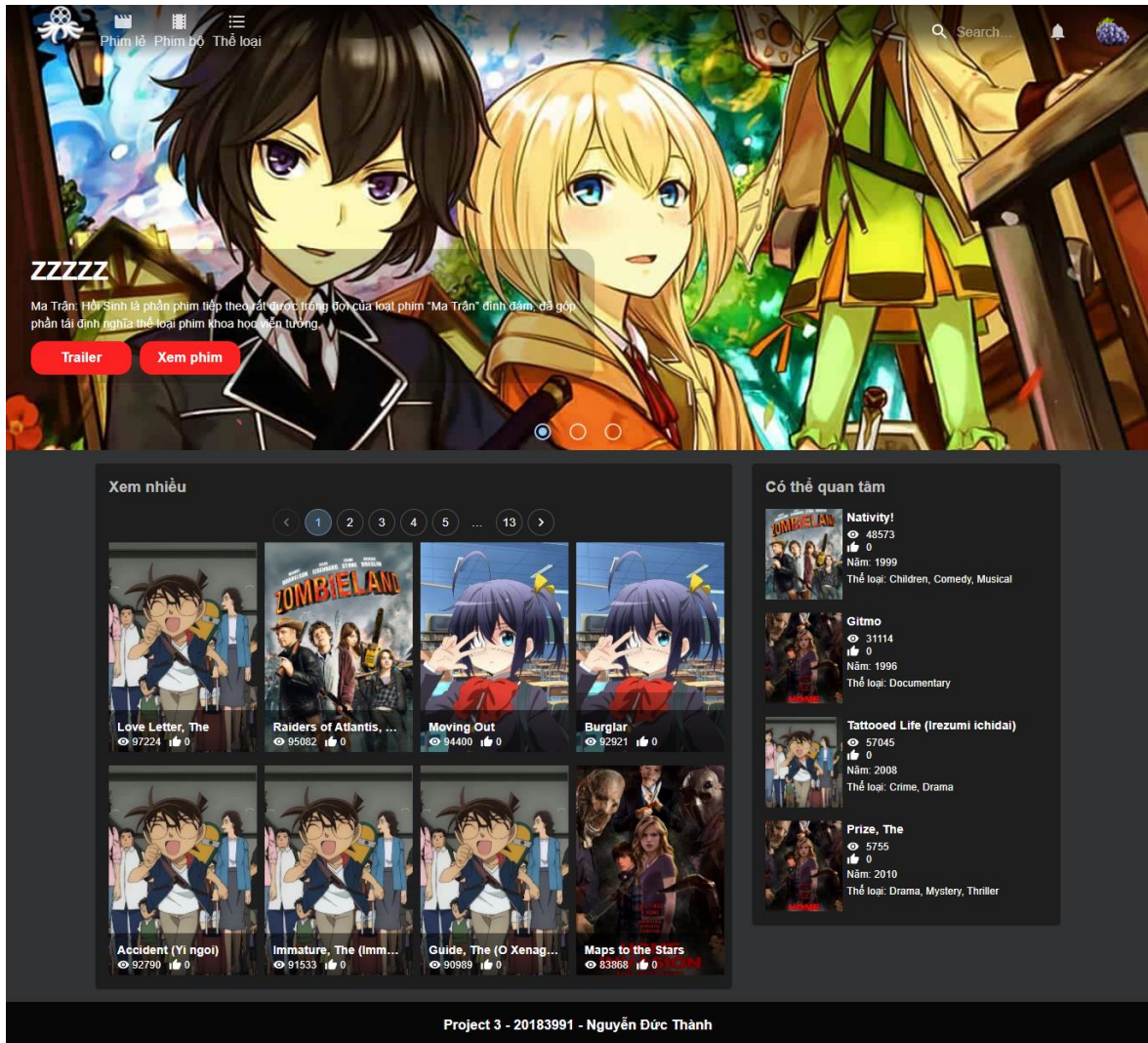
Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
Lập trình Front-end	React (v17.0.2)	https://reactjs.org/
Lập trình Back-end	NodeJS (v14.17.0)	https://nodejs.org/en/
Tạo giao diện	MUI (v5.2.0)	https://mui.com/
Gọi API	Axios (v0.24.0)	https://axios-http.com/
Tạo biểu mẫu	Formik (v2.2.9)	https://formik.org/
Tạo quyền	Jsonwebtoken (v8.5.1)	https://jwt.io/

Định dạng ngày	Moment (v2.29.1)	https://momentjs.com/
Quản lý trạng thái ứng dụng	@reduxjs/toolkit(v1.6.2)	https://redux-toolkit.js.org/
Chat biểu cảm	React-input-emoji(v4.0.9)	https://www.npmjs.com/package/react-input-emoji
Realtime	socket (v4.4.1)	https://socket.io/
Sinh số ngẫu nhiên	uuid (v8.3.2)	https://www.npmjs.com/package/uuid
Xác thực biểu mẫu	Yup (v0.32.11)	https://www.npmjs.com/package/yup
Băm mật khẩu	Argon2 (v0.28.2)	https://www.npmjs.com/package/argon2
Cho phép kết nối khác tên miền	Cors (v2.8.5)	https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
Cho phép đọc file bảo mật	dote nv (v10.0.0)	https://www.npmjs.com/package/dotenv
Chạy server phía back-end	Express (v4.17.1)	https://expressjs.com/
Lấy thời gian từ video	get-video-duration(v4.0.0)	https://www.npmjs.com/package/get-video-duration
Truy cập redis	Ioredis (v4.28.0)	https://redis.io/
Driver mongo	Mongoose (v5.13.2)	https://mongoosejs.com/
Lưu file	Multer (v1.4.3)	https://www.npmjs.com/package/multer

4.3.2 Kết quả đạt được

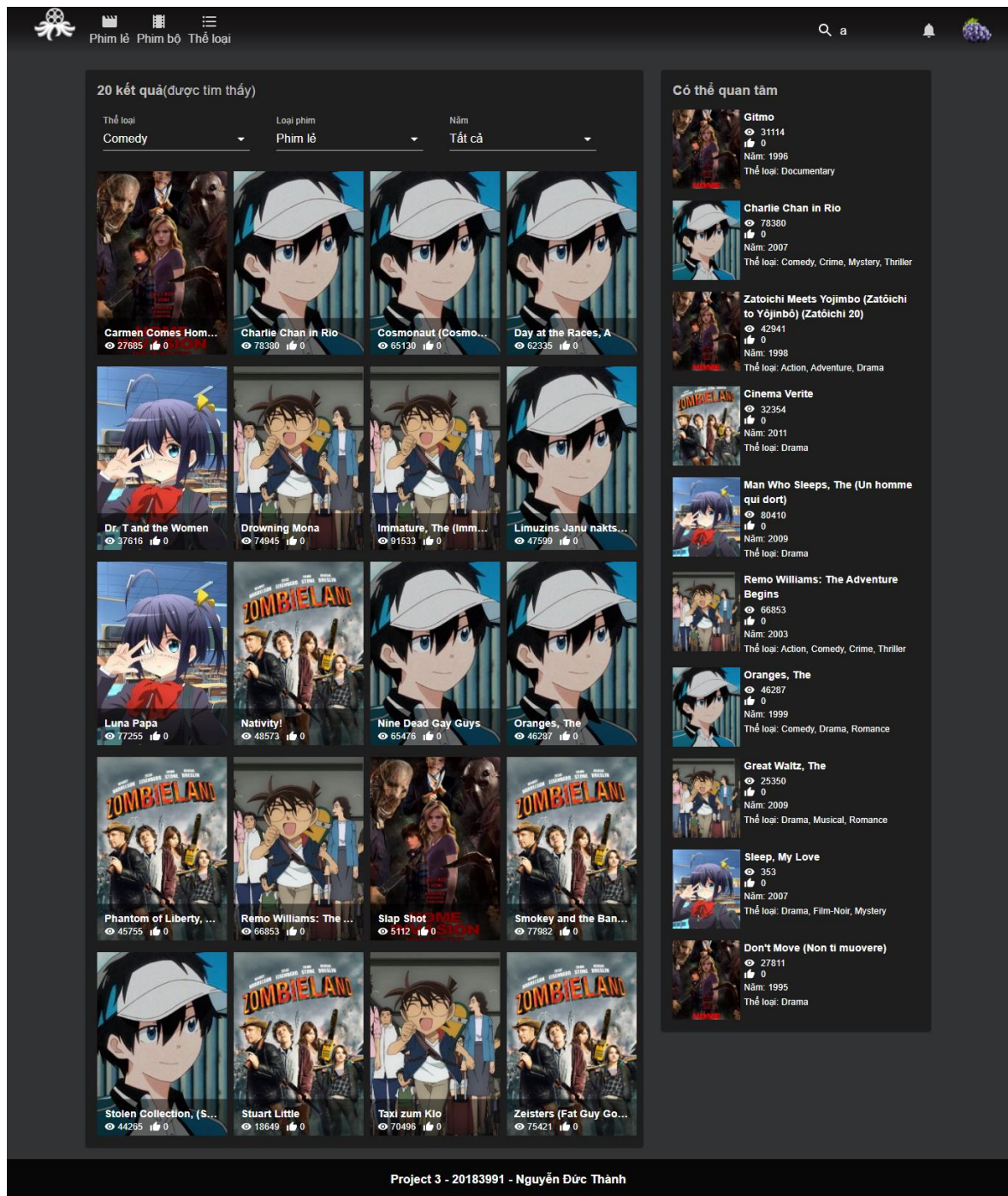
Hệ thống đã đi vào hoạt động với thành phần chính là: giao diện phía người dùng giúp người dùng có trải nghiệm tốt nhất khi xem phim, giao diện phía quản trị viên giúp quản trị viên quản lý website của mình một cách tốt, hiệu quả nhất, một server để nhận các yêu cầu từ client và một cơ sở dữ liệu phi quan hệ để lưu trữ dữ liệu một cách bền vững.

4.3.3 Minh hoạ các chức năng chính



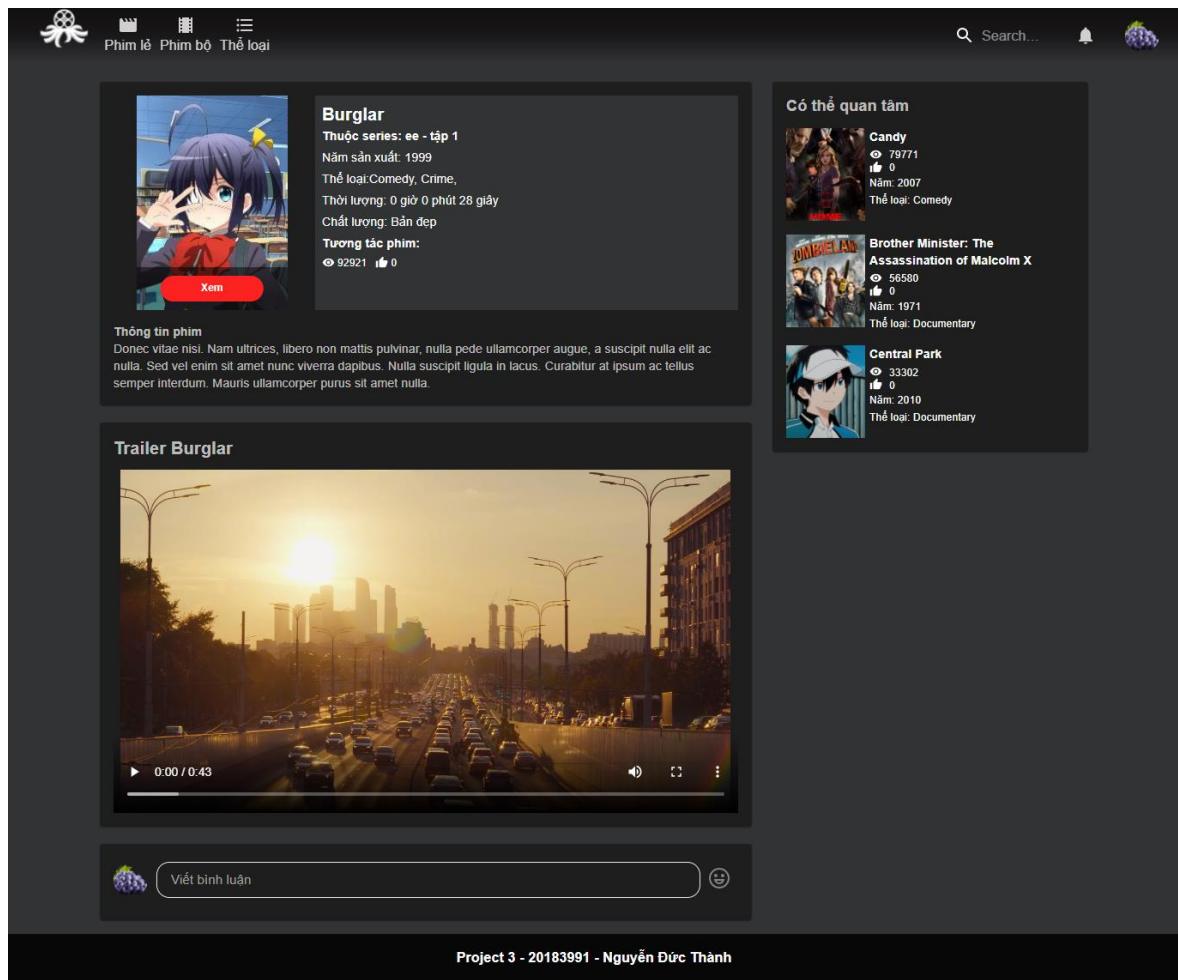
Hình 27 Giao diện Trang chủ Người dùng

Trong giao diện home được chia thành 4 phần chính: Layout (Navbar + Footer), Slider, Contents (hiển thị theo số lượt xem của phim) và Aside (chứa các phim liên quan).



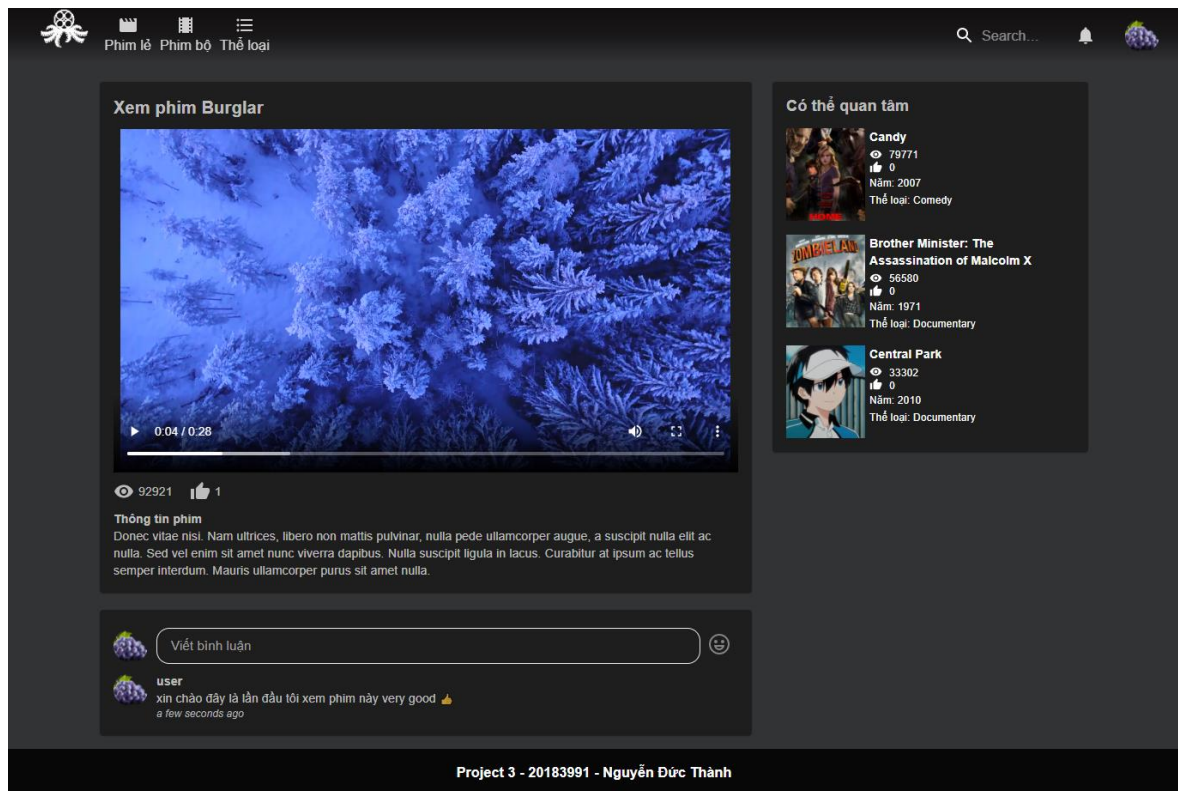
Hình 28 Giao diện Trang Tìm kiếm phim

Trong giao diện trang tìm kiếm phim được chia thành 3 phần chính: Layout (Navbar + Footer), Contents và bộ lọc phim (kết quả ứng với truy vấn phim) và Aside (danh sách các phim ngẫu nhiên).



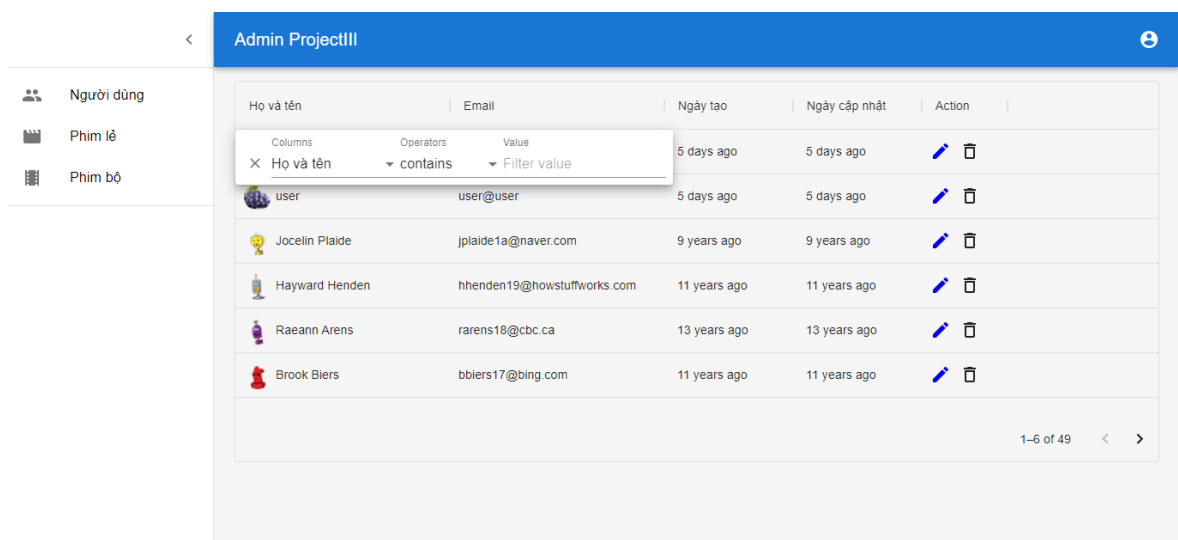
Hình 29 Giao diện chi tiết phim

Trong giao diện chi tiết phim được chia thành 5 phần chính: Layout (Navbar + Footer), InfoMovie (thông tin về phim), VideoPlayer (trailer của phim), Comments (các hình luận phim) và Aside (các phim liên quan) nếu là series thì sẽ hiển thị các bộ trong series đó nếu không sẽ tìm ngẫu nhiên một thể loại có trong bộ phim đang được chọn.















Hình 30 Giao diện xem phim

Trong giao diện chi tiết phim được chia thành 4 phần chính: Layout (Navbar + Footer), VideoPlayer (trailer của phim), Comments (các hình luận phim) và Aside (các phim liên quan) nếu là series thì sẽ hiển thị các bộ trong series đó nếu không sẽ tìm ngẫu nhiên một thể loại có trong bộ phim đang được chọn.






Hình 31 Giao diện Quản lý người dùng

Trong quản lý người dùng được chia thành 2 phần chính: Layout, UserTable có thể lọc người dùng tìm kiếm người dùng, thực hiện sửa, xóa,... người dùng.

Admin ProjectIII								
TAO PHIM MỚI								
Tên phim	Năm sản xuất	Thể loại	Lượt xem	Lượt thích	Lượt bin...	Ngày tạo	Ngày cập nhật	Hành động
phim này được chỉnh : 2020		Action, Adventure, Ani	1	0	1	14 hours ago	14 hours ago	 
ZZZ	2021	Adventure, Animation, 2		1	3	13 hours ago	13 hours ago	 
ZZZZZ	2020	Animation, Children, A	1	0	2	13 hours ago	13 hours ago	 
Carmen Comes Home	2009	Comedy	27685	0	0	10 years ago	10 years ago	 
Cavalcade	2009	Drama, Romance, Wa	82471	0	0	9 years ago	9 years ago	 
Charlie Chan in Rio	2007	Comedy, Crime, Myste	78380	0	0	10 years ago	10 years ago	 
1-6 of 87 < >								

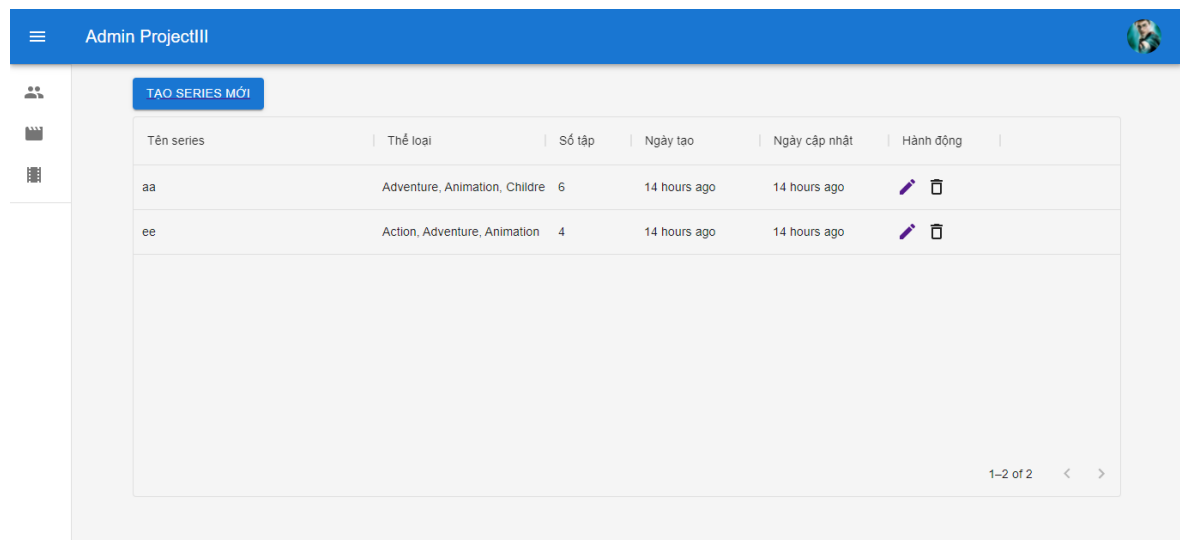
Hình 32 Giao diện Quản lý phim

Trong giao diện quản lý phim quản trị viên có thể thêm, sửa, xóa phim, lọc phim theo tiêu đề, năm,...





Admin ProjectIII	
<ul style="list-style-type: none"> Người dùng Phim lẻ Phim bộ 	<div> <div> <div>Tên phim</div> <div>Burglar</div> </div> <div> <div>Mô tả</div> <div>Donec vitae nisi. Nam ultrices, libero non mattis pulvinar, nulla pede ullamcorper augue, a suscipit nulla elit ac nulla. Sed vel enim sit amet nunc viverra dapibus. Nulla suscipit ligula in lacus. Curabitur at ipsum ac tellus semper interdum. Mauris ullamcorper purus sit amet nulla.</div> </div> <div> <div>Năm sản xuất</div> <div>1999</div> </div> <div> <div>Thời lượng</div> <div>0 giờ 0 phút 28 giây</div> </div> <div> <div>Thể loại</div> <div>Comedy Crime</div> </div> <div> <div>  <div> <div>Ảnh nền</div> <div>Trailer phim</div> </div> </div> <div>  <div> <div>Ảnh phim</div> <div>Nội</div> </div> </div> <div>  <div> <div>Ảnh phim kích thước nhỏ</div> <div>Video phim</div> </div> </div> </div> <div> <div>Đã thích(2)</div> <div> <div>ducthanh</div> <div>user</div> </div> </div> <div> <div>Bình luận(5)</div> <div> <div>ducthanh 5</div> <div>user 4</div> <div>ducthanh 2</div> <div>ducthanh 1</div> <div>user</div> </div> </div> </div>

Hình 33 Giao diện sửa phim

Trong giao diện sửa phim quản trị viên có thể sửa thông tin về bộ phim và xem được lượt thích bình luận về bộ phim

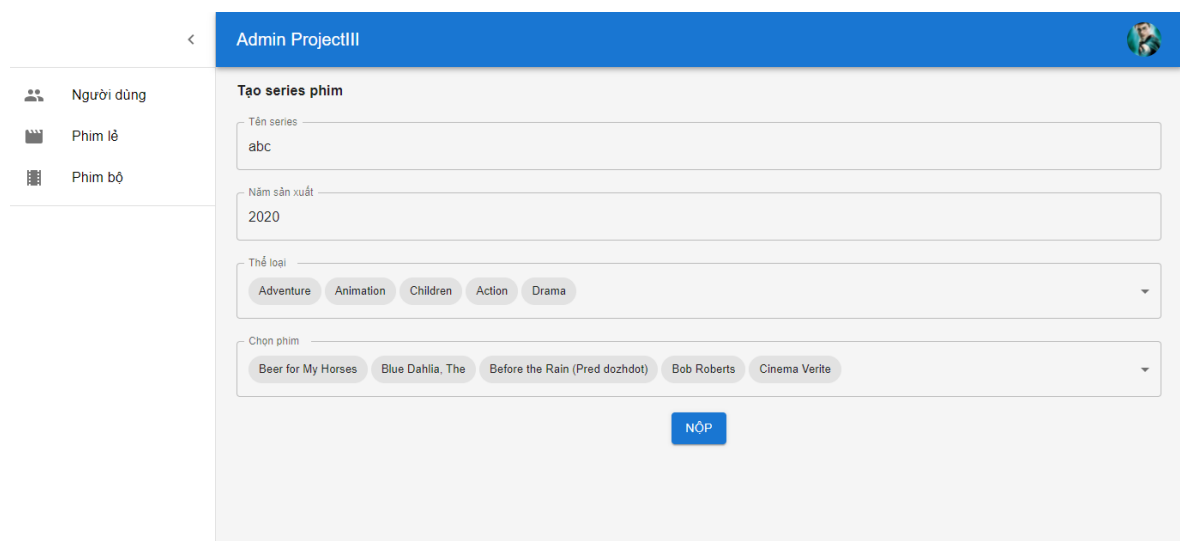


The screenshot shows the 'Admin ProjectIII' interface. At the top, there is a blue header with a menu icon and a user profile icon. Below the header, there is a sidebar with icons for users, movies, and series. The main content area has a button 'TAO SERIES MỚI' and a table with the following columns: Tên series, Thể loại, Số tập, Ngày tạo, Ngày cập nhật, and Hành động. The table contains two rows of data.

Tên series	Thể loại	Số tập	Ngày tạo	Ngày cập nhật	Hành động
aa	Adventure, Animation, Childre	6	14 hours ago	14 hours ago	 
ee	Action, Adventure, Animation	4	14 hours ago	14 hours ago	 

Hình 34 Quản lý Series phim

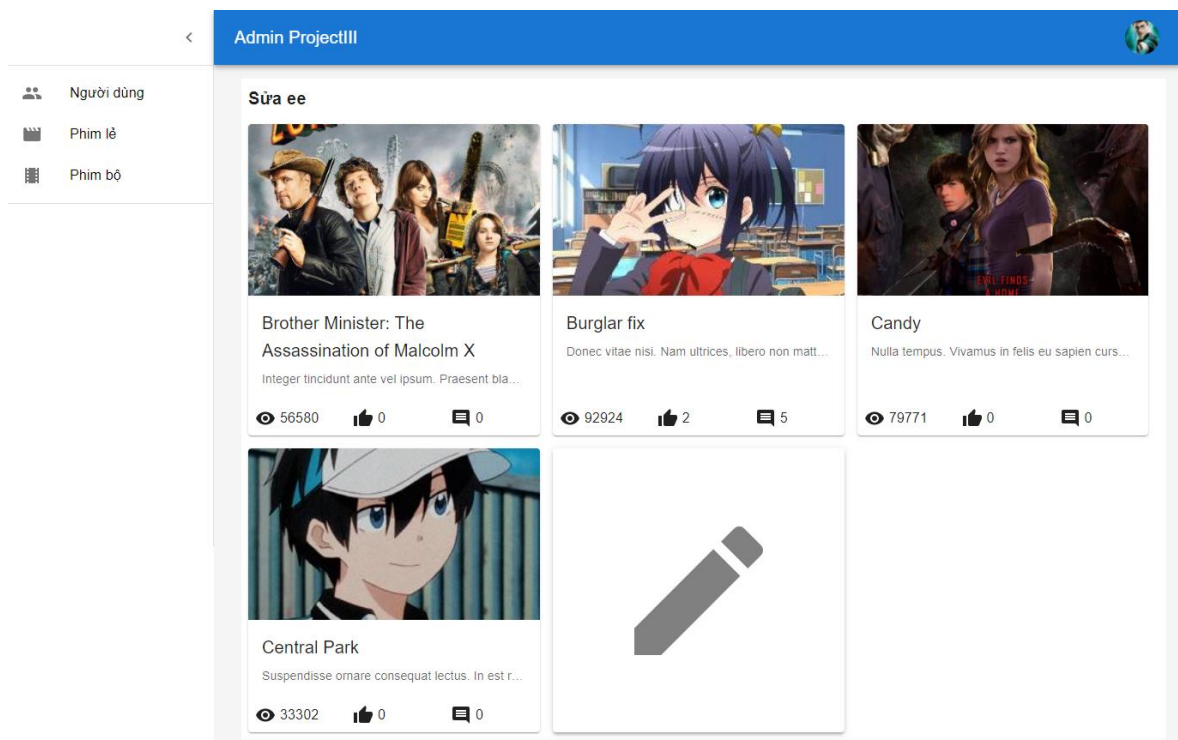
Trong giao diện sửa phim quản trị viên có thể thêm sửa xóa series lọc các thông tin về series theo tên series ngày tạo thể loại,...



The screenshot shows the 'Admin ProjectIII' interface with the 'Tạo series phim' form. The form has a sidebar with icons for users, movies, and series. The main content area has a form with the following fields: Tên series (abc), Năm sản xuất (2020), Thể loại (Adventure, Animation, Children, Action, Drama), and Chọn phim (Beer for My Horses, Blue Dahlia, The, Before the Rain (Pred dozhdot), Bob Roberts, Cinema Verite). There is a 'NỘP' button at the bottom.

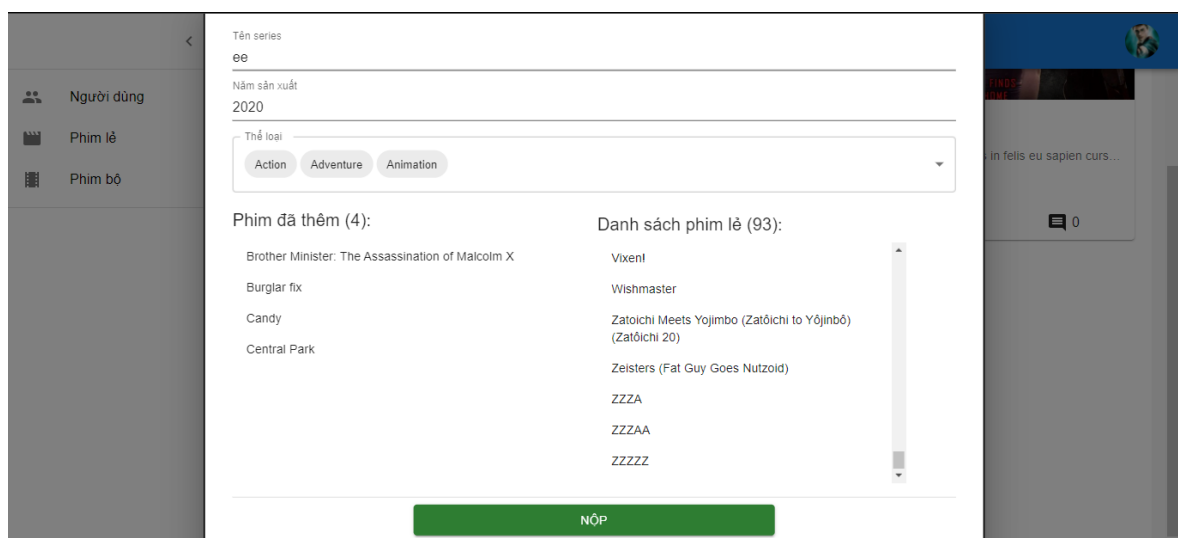
Hình 35 Giao diện tạo series phim

Trong giao diện tạo mới series quản trị viên có thể thêm mới một bộ series từ các phim lẻ.



Hình 36 Giao diện sửa series

Trong giao diện sửa series phim quản trị viên sẽ nhìn thấy toàn bộ thông tin của phim có trong series phim, khi click vào một bộ phim trong danh sách này thì sẽ chuyển đến trang sửa phim, khi nhấn vào biểu tượng icon bút chì sẽ hiện ra màn hình để quản trị viên có thể thêm, xóa phim, thay đổi nhan đề phim trong series.



Hình 37 Modal sửa series phim

4.4 Kiểm thử

Xác thực email người dùng

Mô tả: Phương thức này nhằm email của người dùng.

Yêu cầu: email không được trống, có độ dài lớn hơn 3, có chứa ký tự @ trong email

Kỹ thuật kiểm thử: Sử dụng kỹ thuật phân vùng tương đương, phân tích giá trị biên (trong kiểm thử hộp đen), ta có 2 equivalent class:

Valid: có độ dài chuỗi > 3 ký tự, có chứa ký tự @ trong chuỗi.

Invalid: có ít hơn 3 ký tự trong, không có ký tự nào là @

ID	Test Case	Input	Expected Result
1	Email trống	Null	False
2	Email ít hơn 3 ký tự	12	False
3	Email lớn hơn 3 ký tự và không chứa @	1234	False
4	Email có lớn hơn 3 ký tự và chứa @	<u>thanh@gmail.co</u> <u>m</u>	True
5	Email có 3 ký tự vào không chứa @	123	false

Bảng 14 Kiểm thử email

Xác thực tên người dùng

Mô tả: Phương thức này nhằm xác thực tên người dùng nhập vào có hợp lệ.

Yêu cầu: Tên người dùng không được trống, lớn hơn 2 ký tự và nhỏ hơn 20 ký tự.

Kỹ thuật kiểm thử: Sử dụng kỹ thuật phân vùng tương đương, phân tích giá trị biên (trong kiểm thử hộp đen), ta có 2 equivalent class:

Valid: chuỗi có độ dài ≥ 2 và nhỏ hơn ≤ 20 ký tự.

Invalid: chuỗi trống, chuỗi có độ dài < 2 ký tự và > 20 ký tự.

ID	Test Case	Input	Expected Result
1	Tên người dùng trống	Null	False

2	Tên người dùng có độ dài < 2 ký tự	1	Fasle
3	Tên người dùng có độ dài 2 ký tự	12	False
4	Tên người dùng có độ dài 3 ký tự	123	True
5	Tên người dùng có độ dài 19 ký tự	aa...a(19 ký tự)	True
6	Tên người dùng có độ dài 20 ký tự	aa...a(20 ký tự)	True
7	Tên người dùng có độ dài 21 ký tự	aa...a(21 ký tự)	False

Bảng 15 Kiểm thử tên người dùng

Xác thực mật khẩu người dùng

Mô tả: Phương thức này nhằm mật khẩu của người dùng.

Yêu cầu: mật khẩu không được trống, có độ dài lớn hơn 2 ký tự.

Kỹ thuật kiểm thử: Sử dụng kỹ thuật phân vùng tương đương, phân tích giá trị biên (trong kiểm thử hộp đen), ta có 2 equivalent class:

Valid: có độ dài chuỗi ≥ 2 ký tự.

Invalid: có độ dài chuỗi < 1 ký tự.

ID	Test Case	Input	Expected Result
1	Mật khẩu trống	Null	False
2	Có ít hơn 2 ký tự	1	Fasle
3	Có 2 ký tự	12	False
4	Có 3 ký tự	123	True

Bảng 16 Kiểm thử mật khẩu người dùng.

Xác thực thông tin mô tả phim

Mô tả: phương thức này nhằm mục đích kiểm tra thông tin mô tả về phim khi quản trị viên tạo mới 1 bộ phim.

Yêu cầu: không được trống, có lớn hơn 50 ký tự.

Kỹ thuật kiểm thử: Sử dụng kỹ thuật phân vùng tương đương, phân tích giá trị biên (trong kiểm thử hộp đen), ta có 2 equivalent class:

Valid: có độ dài chuỗi ≥ 50 ký tự.

Invalid: có độ dài chuỗi < 50 ký tự.

ID	Test Case	Input	Expected Result
1	Thông tin mô tả phim trống	Null	False
2	Thông tin mô tả phim có nhỏ hơn 50 ký tự	1234	False
3	Thông tin mô tả phim có 49 ký tự	Aaa...a(49 ký tự)	False
4	Thông tin mô tả phim có 50 ký tự	Aaa...a(50 ký tự)	True
5	Thông tin mô tả phim có 51 ký tự	Aaa...a(51 ký tự)	True

Bảng 17 Kiểm thử thông tin mô tả về phim.

Xác thực năm phim sản xuất

Mô tả: phương thức này nhằm mục đích xác thực thông tin năm nhập vào của phim là hợp lệ hay không

Yêu cầu: không được trống, thuộc kiểu số, có giá trị ≥ 1800 và ≤ 2022 (năm hiện tại).

Kỹ thuật kiểm thử: Sử dụng kỹ thuật phân vùng tương đương, phân tích giá trị biên (trong kiểm thử hộp đen), ta có 2 equivalent class:

Valid: chuỗi số nằm trong khoảng ≥ 1800 và nhỏ hơn ≤ 2022 .

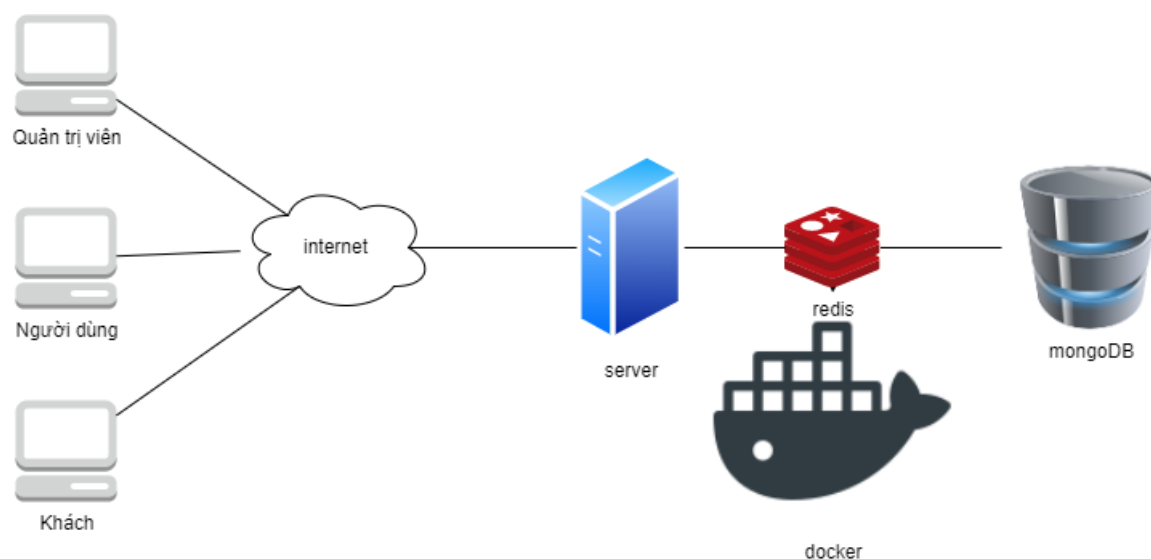
Invalid: chuỗi trống, chuỗi có ký tự chữ, chuỗi số có giá trị < 1800 và > 2022

ID	Test Case	Input	Expected Result
1	Năm sản xuất của phim trống	Null	False
2	Năm sản xuất của phim chứa giá trị không phải là số	1a2	False
3	Năm của phim có giá trị nhỏ hơn 1800	1799	False

4	Năm của phim có giá trị 1800	1800	True
5	Năm của phim có giá trị lớn hơn 1800 và nhỏ hơn 2022	1801	True
6	Năm của phim có giá trị 2021	2021	True
7	Năm của phim có giá trị 2022	2022	True
8	Năm của phim có giá trị lon hơn 2022	2023	False

Bảng 18 Kiểm thử năm sản xuất phim

4.5 Triển khai



Hình 38 Mô hình triển khai của hệ thống

Trong thử nghiệm, hệ thống được chạy ở localhost với cổng máy khách, người dùng ở cổng 3000, cổng quản trị viên là 3001, server chạy ở cổng 5000, redis được chạy thông qua docker ở cổng 6379. Hệ quản trị cơ sở dữ liệu không quan hệ mongoDB chạy ở cổng 27017.

Thông tin cấu hình	Mô tả
Laptop	ASUS S551LN
Hệ điều hành	Windows 10 Pro 64-bit operating system, x64-based processor

RAM	8G
Màn hình	1366 x 768 (32bit) (60Hz)

Bảng 19 Cấu hình máy triển khai ứng dụng

Thông tin	Kết quả
Số lượng người dùng	3 tác nhân (người dùng, quản trị viên, khách)
Số lượng truy cập	4 (2 người dùng, 1 quản trị viên, 1 khách)
Thời gian phản hồi	Không quá 2s
Khả năng chịu tải	Tốt
Kết quả chung	Tốt

Bảng 20 Kết quả triển khai

Chương 5 Các giải pháp và đóng góp nổi bật

Hệ thống được đặt ra phải có khả năng chống quá tải mạng, hỗ trợ, giúp chúng ta có thể trải nghiệm trên bất kì một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp các nhà sản xuất tích hợp lên nhiều sản phẩm khác nhau mà không gặp phải khó khăn gì. Giúp hỗ trợ người dùng nhiều dịch vụ đa dạng và sự tiện dụng bởi khả năng truy cập từ xa mà các mô hình cũ không có. Hệ thống đảm bảo được sự toàn vẹn dữ liệu khi có sự cố xảy ra, dễ dàng mở rộng, xây dựng hệ thống mạng. Hệ thống này cung cấp một nền tảng lý tưởng cho phép tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS)...Người dùng có thể truy cập dữ liệu từ xa, thực hiện các thao tác gửi, nhận file hay tìm kiếm thông tin đơn giản. Có thể có nhiều server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính.

Chính vì những do đó hệ thống được xây dựng và tổ chức theo kiến trúc client/server dựa theo mô hình MVVM. Mô hình MVVM được ra đời sau và được kế thừa từ mô hình MVC truyền thống mang đầy đủ ưu điểm của MVC, phù hợp với các trang web xây dựng theo SPA.

Đầu tiên, MVC là viết tắt của Model View Controller – một mô hình thiết kế được sử dụng rộng rãi trong những phần mềm. Ban đầu, mô hình được thiết kế bởi Trygve Reenskaug trong khi ông đang làm việc ở Smalltalk-80 vào năm 1979. Trong thời gian đầu mô hình này được gọi là View Controller Editor. Cho đến 1994, MVC được đưa vào cuốn sách “Design Patterns: Elements of Reusable Object- Oriented Software” và trở nên phổ biến. Mô hình có 3 thành phần chính: Model (M) – chịu trách nhiệm chính cho các logic nghiệp vụ (business logic) của ứng dụng và quản lý các trạng thái của ứng dụng. Có thể nói rõ chức năng của Model là đọc, ghi dữ liệu trạng thái của ứng dụng một cách liên tục, và có thể xử lý các tác vụ liên quan đến dữ liệu như mạng và xác thực dữ liệu. View (V) – hiển thị dữ liệu và xử lý tương tác Controller (C) – là bộ điều khiển có trách nhiệm tương tác giữa Model và View.

MVVM viết tắt là Model View ViewModel – được phát triển bởi Microsoft và sử dụng cho Windows Presentation Foundation (WPF) và Silverlight. Đến 2005, được John Grossman công bố bằng một bài blog về Avalon (mật mã của Windows Presentation

Foundation). Mô hình MVVM phát triển dựa trên cả MVC và MVP, MVVM được phát triển để tách biệt giao diện UI khỏi logic nghiệp vụ và hành vi trong một ứng dụng. Thành phần trong mô hình MVVM. Model (M) – tương tự như MVC và MVP, Model bao gồm dữ liệu cơ bản để chạy được phần mềm. View (V) – là một giao diện đồ họa giữa người dùng và mô hình tương tự MVC. Dùng để hiển thị kết quả sau khi xử lý dữ liệu. View-Model (VM) – Là lớp trung gian của Model và View. Ưu điểm MVVM: (i)Người dùng có thể thực hiện unit testing mà không phụ thuộc vào View. (ii)Khi test không cần phải tạo mockup như MVP chỉ cần xác nhận biến observable thích hợp. (iii)Sử dụng XAML cho View, vì vậy có thể chỉnh sửa giao diện, không gây ảnh hưởng đến code. (iv)Phân rất rõ ràng ba phần, vì vậy sẽ rất dễ dàng lập trình và kiểm tra sửa lỗi code.

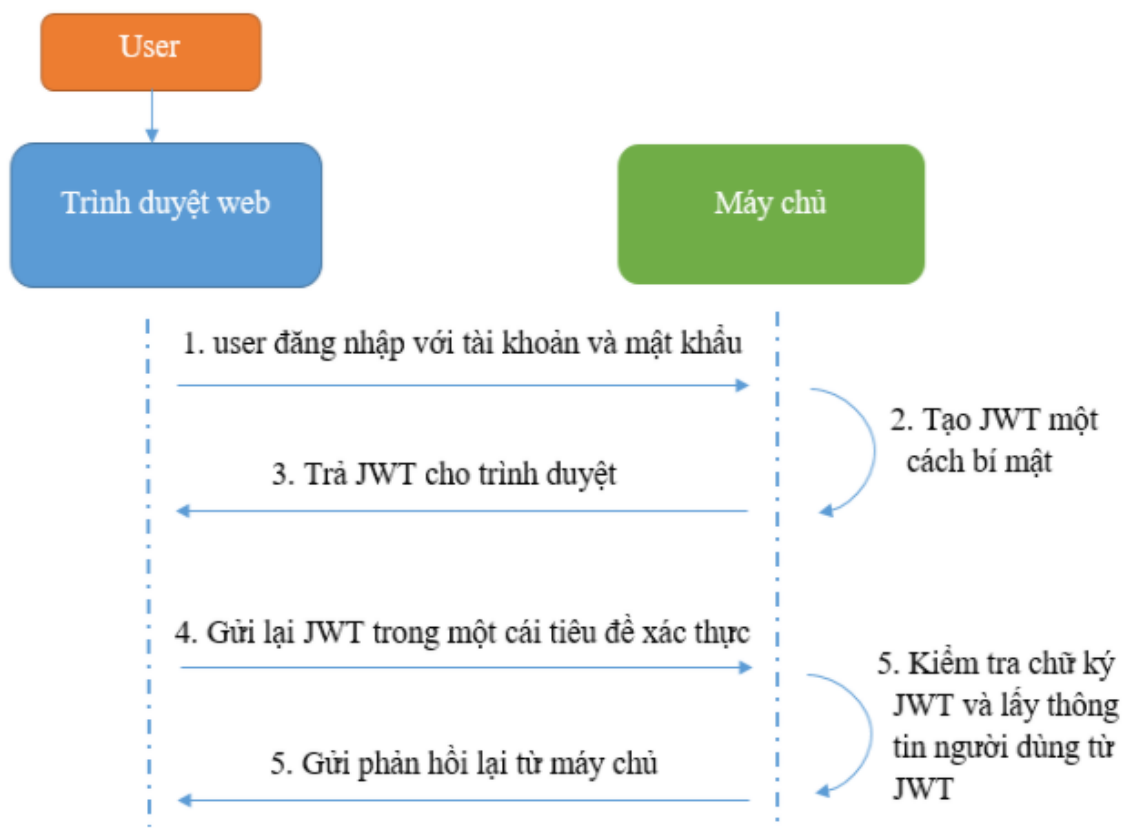
Hệ thống được trên khai trên mô hình client/sever lên vấn đề bảo mật là điều kiện tiên quyết để đảm bảo hệ thống được triển khai đúng mong muốn. Hệ thống có thể bảo mật bằng session cookie. Một session bắt đầu khi client gửi request đến server, nó tồn tại xuyên suốt từ trang này đến trang khác trong ứng dụng web và chỉ kết thúc khi hết thời gian timeout hoặc khi bạn đóng ứng dụng. Giá trị của session sẽ được lưu trong một file trên server. Ví dụ khi bạn đăng nhập vào một trang web và đăng nhập với tài khoản đã đăng ký trước đó. Server sau khi xác thực được thông tin bạn cung cấp là đúng thì nó sẽ sinh ra một tập tin chứa dữ liệu cần lưu trữ của người dùng. Với mỗi session sẽ được cấp phát một định danh duy nhất SessionID. Khi kết thúc một phiên làm việc và bắt đầu một phiên mới, dĩ nhiên bạn sẽ được cấp một SessionID khác với trước đó. Bạn có thể tùy ý quyết định xem nên lưu trữ những thông tin nào vào Session. Nhưng thông thường chúng ta chỉ nên lưu những thông tin tạm thời trong session. Giống khái niệm session là gì, khái niệm cookie cũng được dùng để lưu những thông tin tạm thời. Nhưng tập tin cookie sẽ được truyền từ server tới browser và được lưu trữ trên máy tính của bạn khi bạn truy cập vào ứng dụng. Cookie thường được tạo ra khi người dùng truy cập một website, cookie sẽ ghi nhớ những thông tin như tên đăng nhập, mật khẩu, các tùy chọn do người dùng lựa chọn đi kèm. Các thông tin này được lưu trong máy tính để nhận biết người dùng khi truy cập vào một trang web. Khi người dùng truy cập đến một trang web có sử dụng cookie, web server của trang đó sẽ tự động gửi cookie đến máy tính của người dùng. Khi truy cập đến các trang web sử dụng được cookie đã lưu, những cookie này tự động gửi thông tin của người dùng về cho chủ của nó (người tạo ra cookie). Tuy nhiên những thông tin do cookie ghi nhận không được tiết lộ rộng rãi, chỉ có website chứa cookie mới có thể xem được những thông tin này. Mỗi cookie thường có khoảng thời gian timeout nhất định do lập trình viên xác định trước. Những thông tin được lưu vào cookie ví dụ như thông tin đăng nhập, thao tác người dùng, tần suất ghé thăm website, thời gian truy cập... Tất cả chúng đều là những thông tin mang tính tạm thời và được lưu trong 1 khoảng thời gian.

Việc sử dụng cookie hay session có những hạn chế trong việc mở rộng hệ thống cũng như bảo mật hệ thống. Đối với cookie, dữ liệu cookie được lưu trữ ở phía client nên dữ liệu cookie dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client. Session không được lưu trữ trên trình duyệt. Dữ liệu session được lưu ở phía server, Dữ liệu session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. Do session được lưu ở trên máy chủ lên chúng cũng rất khó để mở rộng hệ thống, nếu hệ thống có 3, 4. ... máy chủ dẫn đến tình trạng phải có 1 cơ chế để đồng bộ các phiên session trong hệ thống giữa các máy chủ, hoặc có thể thêm cache trung gian như redis, ... để làm nhiệm vụ xác thực session của người dùng.

Chính vì vậy, trong hệ thống này em có sử dụng JWT để xác thực quyền của người dùng. Token-based authentication là phương thức xác thực bằng chuỗi mã hóa. Một hệ thống sử dụng Token-based authentication cho phép người dùng nhập user/password (hoặc tương tự) để nhận về 1 chuỗi mã token. Mã này được sử dụng để "xác minh" quyền truy cập vào tài nguyên mà không cần phải cung cấp lại username/password. JSON Web Token (JWT) là 1 tiêu chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA. JWT là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server, các thông tin trong chuỗi JWT được định dạng bằng JSON. Trong đó chuỗi Token phải có 3 phần là header, phần payload và phần signature được ngăn bằng dấu “.”. Header bao gồm hai phần chính: loại token và thuật toán đã dùng để mã hóa. Trong đó loại token có thể mặc định là JWT, một loại thông tin mà cho biết đoạn mã là một token JWT. Thuật toán có thể là HMAC SHA256 – HS256 hoặc RSA. Payload chứa các “Claims”. Claims là một khối thông tin về một thực thể chẳng hạn người dùng là ai và một số metadata bắt buộc, số còn lại tuân theo về JWT hợp lệ và đầy đủ thông tin: iss (issuer), iat (issued-at time) exp (expiration time), sub (subject), aud (audience), ... độ trễ phải hỏi lại từ máy chủ khi tiếp nhận là do độ dài của Payload. Signature là chữ ký trong JWT hay một chuỗi đã được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret)
```

Hình 39 chữ ký trong JWT



Hình 40 Cách thức hoạt động JWT

Như lược đồ ở trên, chuỗi JWT được thực hiện theo một chu trình sau: (i)Người dùng (user) sử dụng trình duyệt đăng nhập vào một miền nào đó mà yêu cầu đăng nhập với tên đăng nhập và mật khẩu. (ii)Máy chủ sẽ nhận được yêu cầu của người dùng, đồng thời kiểm tra thông tin tên đăng nhập và mật khẩu. (iii)Máy chủ sau khi kiểm tra thông tin người dùng, nếu đúng sẽ trả một JWT về cho người dùng, nếu không quay lại bước (i). (iv)Người dùng sẽ sử dụng mã JWT để tiếp tục sử dụng cho các yêu cầu kế tiếp trên miền của máy chủ. (v)Máy chủ sẽ không cần phải kiểm tra lại thông tin người dùng mà chỉ cần kiểm tra đúng JWT đã được cấp từ đó tăng tốc độ sử dụng trên miền giảm thời gian truy vấn. (vi)Máy chủ trả phản hồi phù hợp cho người dùng. Qua quá trình được nêu, rất rõ ràng JWT được tạo ra như một cách để trao đổi thông tin xác thực an toàn giữa hai bên. Vì vậy, JWT không ẩn, không làm mờ, không che giấu dữ liệu mà chỉ nhằm chứng minh dữ liệu được tạo ra bởi một nguồn xác thực.

Như vậy, JWT không phụ thuộc vào bất kỳ điều kiện nào ràng buộc đối với cookie và session. Người dùng không thể nào sửa đổi được chuỗi token, lúc đó server có thể biết được token đã bị sửa đổi và đưa ra thông báo từ chối yêu cầu. JWT không bị ràng buộc bởi số lượng máy server có trong hệ thống, mọi thứ hoạt động chỉ phụ thuộc vào chuỗi token và khóa bí mật (private key) mà server sở hữu. Để đảm bảo tính bảo mật, hệ thống lên thường xuyên thay đổi khóa bí mật của hệ thống,...

Hệ thống sử dụng JPA với thư viện ReactJS mang lại cảm giác tối ưu cho người sử dụng. Single Page Application (hay còn được biết đến với cái tên viết tắt: SPA), là một kiểu lập trình ứng dụng web rất thân thiện với thiết bị di động và bạn sẽ không cần phải load lại trang trong quá trình sử dụng ngay trên trình duyệt. Trước đây, khi chưa có Single Page Application, việc lập trình website sẽ sử dụng mô hình MVC. Phương thức hoạt động sẽ tập trung về phía server: client gửi request cho server, server xử lý và response lại client. Thông thường, server sẽ trả lại kết quả là cả 1 trang web hoàn chỉnh với HTML, CSS và cả JavaScript. Trong khi đó, client gần như không phải làm gì cả. Thao tác phản hồi đầy đủ này của server được gọi là SSR – Server Side Rendering. Đến giai đoạn tiếp theo, website ngày càng phức tạp hơn và yêu cầu của người dùng cao hơn. Điều này buộc các nhà phát triển phải tập trung vào trải nghiệm người dùng. SSR không còn đáp ứng được yêu cầu như trước đây nữa. Nếu muốn tối ưu hóa trải nghiệm người dùng, ứng dụng/ website sẽ buộc phải giảm tải bớt công việc trên server và tập trung xử lý ở client để gia tăng tốc độ, gia tăng trải nghiệm. jQuery xuất hiện như một “cứu tinh” để hỗ trợ cho “việc ai người ấy làm” giữa server và client. Tuy nhiên, giai đoạn này, jQuery vẫn còn thô sơ và chưa đáp ứng được những thao tác xử lý phức tạp tại client. Và Single Page Application ra đời như một hệ quả tất yếu cho việc lập trình web nhằm để gia tăng những thao tác xử lý tại client, giảm tải cho server. Single Page Application là một công nghệ khiến công việc lập trình web trở nên phức tạp hơn khi phân tách frontend và backend ra. Lợi ích của Single Page Application: (i) Hạn chế truy vấn lên server, đây là một ưu điểm nổi bật ví dụ bên trên. Single Page Application chỉ hiển thị đúng những gì bạn cần đồng nghĩa với việc bạn có thể tiết kiệm rất nhiều chi phí, tài nguyên khi không phải tải lại cả 1 trang. (ii) Thân thiện với thiết bị di động. Thiết bị di động đang và sẽ vẫn phát triển bền vững tăng dần đều trong tương lai. Vì thế, những ứng dụng thân thiện với thiết bị di động sẽ “chiếm được cảm tình” của người dùng hơn. Nhược điểm của các trang web SPA code khó điều khiển, bảo trì hệ thống. Redux được ra đời để giải quyết một phần vấn đề này.

Redux được hiểu là một thư viện javascript giữ vai trò tạo một lớp quản lý trạng thái của các ứng dụng và giúp viết các ứng dụng đó một cách nhất quán và dễ dàng thử nghiệm trong nhiều môi trường khác nhau. Redux bao gồm: Actions: Actions hay còn được hiểu là các events. Nó được xem là nơi mang những thông tin mà bạn gửi dữ liệu từ ứng dụng đến cửa hàng redux. Mỗi thông tin là một object mô tả những gì đã xảy ra. Các dữ liệu này có thể là từ sự tương tác của người dùng với ứng dụng, API calls hoặc là từ các form submission. Reducers: Reducers là nơi giúp xác định được sự thay đổi của State. Nó là các function nguyên thủy có khả năng lấy state hiện tại của ứng dụng để thực hiện một action và trả về một state hoàn toàn mới. Các state đó được lưu trữ như những object và dễ dàng xác nhận được sự thay đổi trong phản hồi của action gửi đến store. Store: Store là nơi lưu trạng thái của ứng dụng và là duy nhất trong một ứng dụng redux bất kỳ nào. Nhiệm vụ của store chính là quản lý, access các state được lưu, cho phép truy cập state thông qua

Getstate, update state thông qua dispatch và đăng ký hoặc hủy các listeners thông qua helper methods. Chính vì vậy, redux giúp: cải thiện quản lý trạng thái: tương tự như một bản replay có thể retro hoặc undo các trạng thái của ứng dụng. Phát triển tăng tốc: Khi redux kết hợp với Hot Module Replacement, nó sẽ tạo thành bộ đôi ăn ý giúp bạn có thể dễ dàng viết code và debug. Ứng dụng offline: Mọi thao tác của người sử dụng sẽ được lưu vào một cây trạng thái. Khi có kết nối internet, cây trạng thái này sẽ được đồng bộ lên server bởi các sự kiện. Khả năng dự báo về kết quả. (Predictability of outcome) Khả năng maintain bảo trì: Nghiêm ngặt trong cấu trúc làm cho code trở nên dễ dàng bảo trì hơn. Server rendering: Nó rất hữu ích đặc biệt là cho việc khởi tạo render, người dùng sẽ có trải nghiệm tốt hơn hoặc cho việc tối ưu bộ máy tìm kiếm. Developer tools: Developer có thể theo dõi mọi thứ xảy ra trong ứng dụng từ các actions đến sự thay đổi của state.

Sử dụng NodeJS một Framework xuất sắc giành vị trí số 1 với số lượng người dùng lên tới 50% theo thống kê của Stackoverflow. NodeJS có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Chúng ta có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất. NodeJS có những ưu điểm: giúp dễ dàng mở rộng khi có nhu cầu phát triển website. Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử lý Nodejs nhanh hơn rất nhiều. NodeJS có khả năng xử lý nhiều Requests cùng một lúc trong thời gian ngắn nhất. NodeJS có khả năng xử lý hàng ngàn process(chu trình) cho hiệu suất đạt mức tối ưu nhất. Phù hợp để xây dựng những ứng dụng thời gian thực như các ứng dụng chat, mạng xã hội ...

Đối với cơ sở dữ liệu, chúng ta phải biết đến những khái niệm các bảng, các cơ sở dữ liệu quan hệ. Thậm chí, chúng ta còn phải dùng đến các bảng để lưu dữ liệu. Đến với MongoDB chúng ta sẽ chỉ dùng khái niệm Collection thay vì bảng như trước. Nếu như RDBMS có collection ứng với table, document ứng với row. Thì khi sử dụng Mongo các sẽ dùng các document thay cho row trong RDBMS. Các collection trong MongoDB được cấu trúc linh hoạt. Các bạn có thể sử dụng nó để cho phép các dữ liệu lưu trữ. Mà chúng ta không cần phải tuân theo một cấu trúc nhất định như ban đầu. Những thông tin liên quan đến nhau sẽ cùng được lưu trữ để truy vấn nhanh hơn. Nhờ có ngôn ngữ truy vấn MongoDB các bạn có thể thực hiện những điều này.

Ưu điểm của MongoDB so với những hệ cơ sở dữ liệu khác:

MongoDB có nhiều ưu điểm hơn so với những loại khác. Ưu điểm đầu tiên của MongoDB chính là sử dụng lưu trữ dữ liệu dưới dạng Document JSON. Nhờ có nó nên mỗi một collection đều sẽ có các kích cỡ và các Document khác nhau. Sự linh hoạt trong việc lưu trữ dữ liệu của MongoDB là rất hữu dụng. Chính vì vậy, chúng ta hoàn toàn có thể sử dụng MongoDB để Insert dữ liệu bất cứ lúc nào.

Ưu điểm thứ hai của MongoDB đó chính là nó không có sự ràng buộc lẫn nhau trong dữ liệu. Chúng ta sẽ không cần phải join như trong RDBMS. Nên khi sử dụng insert hay xóa, update sẽ không cần tốn nhiều thời gian. Chúng ta cũng không cần phải chờ xem nó có thỏa mãn các ràng buộc dữ liệu hay không để tiến hành insert...

Ngoài ra, sử dụng MongoDB còn hỗ trợ hiệu năng cao cho người dùng. Ví dụ như tốc độ truy vấn find, update, insert hay delete. Tất cả đều được tối ưu nhanh hơn so với các hệ thống quản trị dữ liệu quan hệ khác. Từ những thử nghiệm cho thấy, tốc độ của MongoDB có thể nhanh gấp 100 lần so với MySQL.

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Đứng trước xu hướng phát triển của công nghệ thông tin như vũ bão, việc áp dụng những công nghệ mới vào lập trình là vô cùng cần thiết giúp tăng trải nghiệm người dùng. Ứng dụng “xây dựng website xem phim trực tuyến” đã giải quyết được vấn đề đó

Sản phẩm đã tạo ra một môi trường giúp cho người dùng có thể xem những bộ phim mình yêu thích, tăng trải nghiệm người dùng khi sử dụng những công nghệ làm website mới thay vì sử dụng các mô hình kiểu cũ gây quá tải cho server.

Tuy nhiên, do thời gian thực hiện đề tài tương đối hạn chế nên chương trình không thể tránh được những thiếu sót nhất định. Đồng thời vẫn còn có một số form chưa hoàn thành đầy đủ và chính xác và các ràng buộc về dữ liệu. Hơn nữa, trong quá trình vừa viết chương trình vừa tìm hiểu về cơ sở dữ liệu MongoDB nên chưa thể tránh khỏi những thiếu sót.

6.2 Hướng phát triển

Để chương trình có thể hoạt động hiệu quả hơn cần dữ liệu phải được lưu trữ bằng hệ quản trị cơ sở dữ liệu có tính bảo mật hơn. Cần có sự giao tiếp song song giữa quản trị viên và người dùng một cách chặt chẽ và nhiều hơn để tối ưu trải nghiệm của người dùng.

Hệ thống cần bổ sung thêm một số chức năng như:

Xem lịch sử, xem sau: thêm 2 Collections: History và WatchLater trong cơ sở dữ liệu.

Mở rộng chức năng đăng nhập với facebook, google,... : sử dụng PassportJS thực hiện đăng nhập với các bên thứ 3 (facebook, google,...).

Cải thiện hiệu suất hệ thống: tối ưu câu truy vấn trong hệ thống, sử dụng tối đa tham chiếu đến các collection.

Tài liệu tham khảo

Trang đặc tả JWT của tổ chức auth0: <https://jwt.io/introduction>

Mẫu mô hình MVC, MVVM của tổ chức TINO: <https://wiki.tino.org/mvc-mvp-mvvm-lagi/>

...

Phụ lục

Toàn bộ source code, hình ảnh, file astah, báo cáo,... của học phần môn Project III có địa chỉ: <https://github.com/thanhnd183991/projectIII>