

TRƯỜNG ĐẠI HỌC SƯ PHẠM VÀ KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



TIÊU LUẬN CHUYÊN NGÀNH

CÔNG NGHỆ PHẦN MỀM

XÂY DỰNG TRANG WEB ĐỂ XUẤT VÀ XEM PHIM CHO NGƯỜI DÙNG
SỬ DỤNG REACTJS, NODEJS, ML.NET, ASP.NET CORE, PYTHON

TÊN SINH VIÊN:

NGUYỄN THANH BẰNG

MÃ SỐ SINH VIÊN:

18110254

TRẦN VĂN HÂN MINH ĐÍNH

18110269

Khóa Học:

2018-2022

GIÁO VIÊN HƯỚNG DẪN:

TS. HUỲNH XUÂN PHUNG

Tp. Hồ Chí Minh, 10/08/2021

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: Nguyễn Thanh Bằng MSSV: 18110254

Họ và tên Sinh viên 2: Trần Văn Hân Minh Đính MSSV: 18110269

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng trang web để xuất và xem phim cho người dùng sử dụng ReactJS, NodeJS, ML.NET, ASP.NET Core, Python.

Họ và tên Giáo viên hướng dẫn: Huỳnh Xuân Phụng

NHẬN XÉT

1. Về nội dung đề tài khối lượng thực hiện:

.....
.....
.....
.....
.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....
.....
.....

3. Khuyết điểm

.....
.....
.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại :

6. Điểm :

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: Nguyễn Thanh Bằng MSSV: 18110254

Họ và tên Sinh viên 2: Trần Văn Hân Minh Đính MSSV: 18110269

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng trang web để xuất và xem phim cho người dùng sử dụng ReactJS, NodeJS, ML.NET, ASP.NET Core, Python.

Họ và tên Giáo viên hướng dẫn: Huỳnh Xuân Phụng

NHẬN XÉT

1. Về nội dung đề tài khởi lượng thực hiện:

.....
.....
.....
.....
.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....
.....
.....
.....

3. Khuyết điểm

.....
.....
.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại :

6. Điểm :

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên phản biện

(Ký & ghi rõ họ tên)

NHIỆM VỤ CỦA TIÊU LUẬN CHUYÊN NGÀNH

Tên sinh viên: Nguyễn Thanh Bằng (Đại trà)

Mã số sinh viên: 18110254

Tên sinh viên: Trần Văn Hân Minh Đính (Đại trà)

Mã số sinh viên: 18110269

Chuyên ngành: Công nghệ phần mềm

Tên đề tài: xây dựng trang web đề xuất và xem phim cho người dùng sử dụng ReactJS, NodeJS, ML.NET, ASP .NET Core

Môi trường làm việc: Github

Hướng dẫn: TS. Huỳnh Xuân Phụng

Nội dung:

Yêu cầu:

- Phân tích, thiết kế, xây dựng hệ thống dự đoán và cho phép người dùng xem phim trực tuyến.
- Đáp ứng đầy đủ các yêu cầu của công nghệ phần mềm.
- An toàn, bảo mật cho người dùng.

Thực hành: Áp dụng cho việc xây dựng, thiết kế hệ thống dự đoán và xem phim.

Thời gian thực hiện: 23/08/2021 – 13/12/2021

Hồ Chí Minh, ngày tháng năm 2021

GIÁO VIÊN XÉT DUYỆT

(Tên và chữ ký)

GIÁO VIÊN HƯỚNG DẪN

(Tên và chữ ký)

ĐÁM BẢO

Chúng tôi cam đoan rằng dự án này là do chúng tôi thực hiện. Chúng tôi không sao chép, sử dụng bất kỳ tài liệu, mã nguồn nào của người khác mà không ghi rõ nguồn gốc. Chúng tôi xin chịu trách nhiệm nếu vi phạm.

Hồ Chí Minh, ngày 27 tháng 12 năm 2021

Trần Văn Hân Minh Đính Nguyễn Thanh Bằng

MỤC LỤC

LỜI CẢM ƠN	1
TÓM TẮT	2
1. CÁC VĂN ĐỀ NGHIÊN CỨU	2
2. CÁC VĂN ĐỀ PHÁT SINH.....	2
3. CÁC PHƯƠNG PHÁP GIẢI QUYẾT VĂN ĐỀ	2
4. KẾT QUẢ ĐẠT ĐƯỢC.....	2
5. CHUA ĐẠT ĐƯỢC.....	2
DANH MỤC VIẾT TẮT	3
PHẦN MỞ ĐẦU	4
1.1. TÍNH CẤP THIẾT VÀ MỤC TIÊU CỦA ĐỀ TÀI.....	4
1.1.1. Tính cấp thiết của đề tài	4
1.1.2. Mục tiêu của đề tài.....	4
1.2. KHẢO SÁT THỰC TRẠNG	4
1.2.1. Khảo sát thực trang xem phim hiện nay	4
1.2.2. Khảo sát các website xem phim	5
1.3. ĐỐI TƯỢNG, PHẠM VI VÀ PHƯƠNG PHÁP NGHIÊN CỨU	8
1.3.1. Đối tượng nghiên cứu	8
1.3.2. Phạm vi nghiên cứu	8
1.4. KẾT QUẢ MONG MUỐN.....	8
PHẦN NỘI DUNG	9
Chương 1: NGUYÊN LÝ HOẠT ĐỘNG	9
1.1. SƠ ĐỒ HOẠT ĐỘNG.....	9
1.2. NGUYÊN LÝ HOẠT ĐỘNG	9
Chương 2: THIẾT KẾ HỆ THỐNG.....	10
2.1. USECASE VÀ ĐẶT TẨ.....	10
2.2. ĐẶC TẨ UI	34
2.3. ACTIVITIES DIAGRAM	52
2.4. SEQUENCE DIAGRAM	61
Chương 3: THUẬT TOÁN DỰ ĐOÁN PHIM LIÊN QUAN.....	68
3.1. TÌM HIỂU CÔNG NGHỆ VÀ THƯ VIỆN	68
3.2. THƯ VIỆN SỬ DỤNG.....	68
3.3. ÁP DỤNG VÀ THUẬT TOÁN XỬ LÝ	69
Chương 4: THUẬT TOÁN DỰ ĐOÁN PHIM CHO TÙNG NGƯỜI DÙNG ..	76

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM	GVHD: TS. Huỳnh Xuân Phụng
4.1. TÌM HIỂU CÔNG NGHỆ VÀ THƯ VIỆN SỬ DỤNG.....	76
4.2. ÁP DỤNG THUẬT TOÁN VÀ XỬ LÝ	76
PHẦN KẾT LUẬN	82

LỜI CẢM ƠN

Trên thực tế, không có thành công nào không gắn với sự hỗ trợ, giúp đỡ hoặc những lần thất bại. Hôm nay chúng em xin gửi lời cảm ơn chân thành này đến TS. Huỳnh Xuân Phụng - người trực tiếp hướng dẫn chúng em hoàn thành đồ án này, đã hỗ trợ cực kỳ nhiệt tình, tỉ mỉ để đưa chúng em đi trên con đường có tỉ lệ thành công cao nhất, dễ dàng nhất và truyền động lực cho chúng em trong quá trình chọn đề tài cũng như là hướng dẫn và nhận xét. Nếu không có sự hướng dẫn và kinh nghiệm thực tế của thầy, nhóm chúng em nghĩ bài báo cáo này sẽ khó hoàn thành đúng thời hạn mà đầy đủ những chức năng cơ bản đến như vậy. Một lần nữa chúng em xin cảm ơn thầy.

Chúng em cũng xin chân thành cảm ơn các thầy cô Khoa công nghệ thông tin trường Đại học Sư phạm Kỹ thuật TP.HCM đã dạy cho nhóm sinh viên báo cáo, những kỹ năng, kiến thức vô cùng quý báu của một người lập trình viên, để chúng em có đủ kiến thức hoàn thành đề tài của chúng em cũng như là hoàn thành được một ước mơ, mục tiêu trong lòng chúng em.

Do thời gian có hạn, kiến thức còn hạn chế và còn nhiều lo lắng khác nên không tránh khỏi những sai sót, rất mong nhận được sự góp ý quý báu của quý thầy cô để bổ sung kiến thức của chúng em hoàn thiện hơn cho sau này. Chúng em xin gửi cảm ơn chân thành và kính trọng to lớn như những bậc sinh thành tới TS. Huỳnh Xuân Phùng nói riêng và quý thầy cô trong khoa nói chung.

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

TÓM TẮT

1. CÁC VẤN ĐỀ NGHIÊN CỨU

- Tìm hiểu về các hệ thống tương tự, phân tích để tìm ra điểm mạnh và phù hợp để áp dụng cho đề tài.
- Tìm hiểu về ReactJS để thiết kế giao diện và xử lý sự kiện.
- Tìm hiểu về NODEJS, ASP .NET Core để giải quyết vấn đề về API cho hệ thống dự đoán phim.
- Tìm hiểu về Recommendation System (Hệ thống gợi ý) để gợi ý một số bộ phim phù hợp với từng người dùng cụ thể là ML.NET.

2. CÁC VẤN ĐỀ PHÁT SINH

- Đánh giá và bình luận không real-time với nhiều người dùng.
- Bảo mật là một trong những yêu cầu quan trọng nhất cho website thực tế.
- MongoDB không cho phép lưu hình ảnh nên phải chuyển ảnh sang binary nên tồn không gian lưu trữ cũng như là truy xuất chậm, làm thời gian tải trang lâu hơn bình thường.

3. CÁC PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ

- Sử dụng thư viện Socket.io để real-time cho bình luận và đánh giá.
- Băm mật khẩu và sử dụng JWT để set cookies để cải thiện yêu cầu bảo mật cho trang web.
- Sử dụng Cloudinary để lưu hình ảnh.

4. KẾT QUẢ ĐẠT ĐƯỢC

- Hệ thống dự đoán và xem phim bao gồm: hệ thống trang web cho phía người dùng cuối (user) và hệ thống trang web cho phía người quản lý (admin).
- Giao diện dễ sử dụng, thân thiện với người dùng.
- Tính cá nhân hóa cho user thông qua việc gợi ý các bộ phim có thể phù hợp.
- Dễ dàng quản lý, sửa chữa và mở rộng.

5. CHƯA ĐẠT ĐƯỢC

- Tăng tổng số lượt xem khi đã xem một khoảng thời gian nhất định cho từng bộ phim.
- Hệ thống dự đoán chưa có độ chính xác cao do thiếu số liệu và số liệu không có độ tin cậy cao.

DANH MỤC VIẾT TẮT

API: Application Programming Interface

JWT: Json Web Token

Hover: Thao tác di chuyển con trỏ chuột vào vị trí chỉ định

Object: Một đối tượng trong lập trình hướng đối tượng

Admin: Người quản lý trang web

Model: Là nơi lưu trữ dữ liệu trong lập trình MVC, hoặc cũng là nơi đào tạo trong mô hình dự đoán

Username (user): Tên đăng nhập

UI: Giao diện

Back-End: Phần xử lý cho một hành động trên giao diện web.

Front-End: Phần giao diện của web.

PHẦN MỞ ĐẦU**1.1. TÍNH CẤP THIẾT VÀ MỤC TIÊU CỦA ĐỀ TÀI****1.1.1. Tính cấp thiết của đề tài**

Điều đầu tiên chúng ta cần làm trước khi xem một bộ phim nào đó chính là chọn phim để xem, nghe thì đơn giản nhưng thật khó để chọn được một bộ phim phù hợp với chính bản thân mình. Bạn thử tưởng tượng xem nếu chọn nhầm một bộ phim thật tệ để xem thì bạn sẽ như thế nào? Vâng! Nếu bạn là người dễ tính thì đó chẳng là một vấn đề gì to tát. Còn nếu bạn là một người không dễ tính thì tôi nghĩ đó là một vấn đề không hề dễ chịu. Thật vậy, cho dù dễ hay không dễ tính thì chọn sai phim sẽ làm người dùng tốn rất là nhiều thời gian, mệt mỏi và bức tức không những thế lại còn làm họ mất đi niềm tin đối với hệ thống web của chúng ta nó làm cho họ càng tin rằng web của chúng ta không có nhiều phim hay để xem. Thật tồi tệ nếu điều đó xảy ra với web của chúng ta.

1.1.2. Mục tiêu của đề tài

Làm hài lòng khách hàng là tiêu chí chung của không ít giới kinh doanh hiện nay bởi một người khách vui vẻ, tin tưởng đón nhận sản phẩm của chúng ta thì có một điều chắc chắn rằng là nếu có cơ hội họ sẽ quay lại ủng hộ chúng ta hoặc giới thiệu chúng ta cho người họ quen biết và ngược lại. Nắm bắt được điều này “hệ thống trang web dự đoán và xem phim” của chúng tôi được đori với mục tiêu làm hài lòng những người dùng đến với chúng tôi. Không để các bạn tò mò hơn nữa chúng tôi xin giới thiệu về mục tiêu cũng như những gì chúng tôi thực hiện để làm hài lòng các bạn sau đây:

- Đầu tiên: bạn có thể xem đánh giá của những người đã trải nghiệm bộ phim đó (về số sao 1-5, đánh giá bằng chữ).
- Tóm tắt ngắn gọn nội dung phim bằng chữ và có cả trailer của bộ phim đó.
- Hệ thống dự đoán phim bạn sẽ thích khác nhau trên từng tài khoản người dùng (Bạn chỉ cần đăng nhập là có thể trải nghiệm ngay).
- Bảng xếp hạng phim được xem nhiều nhất hoặc đánh giá cao nhất.
- An toàn bảo mật, tuyệt đối.

1.2. KHẢO SÁT THỰC TRẠNG**1.2.1. Khảo sát thực trạng xem phim hiện nay**

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM GVHD: TS. Huỳnh Xuân Phụng

Ở Việt Nam, các nền tảng xem phim trực tuyến như: Galaxy Play, FPT Play, K+, VTV Cab On, Zing TV, Danet, POPS... cũng nhanh chóng đáp ứng nhu cầu của khán giả.

Để đáp ứng nhu cầu của khán giả, nhiều nền tảng trực tuyến tại Việt Nam liên tục cho ra mắt các nội dung hấp dẫn, các dự án phim ảnh được đầu tư bài bản, với số vốn lên tới cả trăm tỷ đồng nhằm góp phần “giải cứu phim Việt” khỏi khó khăn mùa Covid-19 cũng như nâng cao chất lượng dòng phim online, mở ra một sân chơi mới đầy tiềm năng trong thời kỳ giải trí số.

Điều này cho thấy, xu hướng, sở thích của nhiều người đã thay đổi, nhất là khi các nền tảng xem phim trực tuyến đang phát triển một cách bài bản và bền vững về cả nội dung và chất lượng.

Các nền tảng xem phim được sử dụng nhiều nhất^[1]:

- Netflix (41,99%): là ứng dụng xem phim hàng đầu trên thế giới với hầu hết tất cả bộ phim, hình ảnh và chất lượng đều được ở mức cao nhất.
- FPT Play (24,27%): FPT Play là ứng dụng xem video, xem tivi trực tuyến rất phổ biến trên điện thoại, máy tính và nay là cả trên tivi.
- VieOn (14.56%): VieON là ứng dụng giải trí hàng đầu của người Việt và cho người Việt, cung cấp bởi Dzones Network trực thuộc công ty Đất Việt VAC. Với kho giải trí khổng lồ có bản quyền cùng sự kết hợp với K+.
- Zing TV (24.27%): ZING TV (tv.zing.vn) là kênh truyền hình trực tuyến theo yêu cầu, với kho nội dung phong phú quy tụ các chương trình truyền hình, hài kịch, phim ảnh, các chương trình giáo dục, thể thao.
- K+ (12.62%)
- Khác (0.24%)

1.2.2. Khảo sát các website xem phim

Tiêu chí đánh giá	Netflix	ZingTV	FPT Pay
Tốc độ tải website	Nhanh	Nhanh	Nhanh

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM **GVHD: TS. Huỳnh Xuân Phụng**

Tốc độ/ Công nghệ	Tốc độ upload, download phim	Nhanh	Nhanh	Nhanh
	Tương thích với các hệ điều hành	Tương thích với mọi hệ điều hành	Tương thích với mọi hệ điều hành	Tương thích với mọi hệ điều hành
	Tương thích với nhiều trình duyệt	Tương thích với nhiều trình duyệt: Google Chrome, Mozilla Firefox, Microsoft Edge...	Tương thích với nhiều trình duyệt: Google Chrome, Mozilla Firefox, Microsoft Edge...	Tương thích với nhiều trình duyệt: Google Chrome, Mozilla Firefox, Microsoft Edge...
Giao diện	Bố cục, cấu trúc các chuyên mục	Bố rõ ràng thành các chuyên mục phim	Bố rõ ràng thành các chuyên mục phim	Bố rõ ràng thành các chuyên mục phim
	Màu sắc	Màu sắc thống nhất, không rối mắt	Màu sắc thống nhất, không rối mắt	Màu sắc thống nhất, không rối mắt
	Kiểu chữ, cỡ chữ	Kiểu chữ đơn giản tối đa	Kiểu chữ đơn giản tối đa	Kiểu chữ đơn giản tối đa
	Thuận tiện cho các thao tác người dùng	Thuận tiện cho các thao tác của người dùng	Thuận tiện cho các thao tác của người dùng	Thuận tiện cho các thao tác của người dùng
	Hỗ trợ truy cập di động	Hỗ trợ trên di động và ipad	Hỗ trợ trên di động và ipad	Hỗ trợ trên di động và ipad
Nội dung	Số lượng phim	Số lượng phim không lò, phải trả phí mới có thể xem những phim hay và mới	Số lượng phim không lò, tuy nhiên những phim cũ thì lại không có	Số lượng phim không lò, quảng cáo trên một lần xem quá nhiều
	Phân loại	Nhiều thể loại	Nhiều thể loại	Nhiều thể loại
	Số lượng server chuyển đổi	1	1	1

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM GVHD: TS. Huỳnh Xuân Phụng

Tính tương tác của website	Tính năng bình luận online	Có	Có	Có
	Tính năng yêu thích phim	Có	Có	Có
Khả năng tìm kiếm	Tìm kiếm nhanh trên website	Nhanh và chính xác	Nhanh và chính xác	Nhanh và chính xác

Nhìn chung với 3 website được khảo sát ở trên đều có tốc độ truy cập nhanh, giao diện thu hút, nội dung khá là phong phú. Song lại có những khuyết điểm về nội dung do các hình thức kinh doanh tăng thu nhập chẵn hạn như chạy quảng cáo, hay nâng cấp tài khoản, ... Đặc biệt là không có nhiều server để chuyển đổi, nếu có số lượng truy cập lớn rất dễ gây giật, đứng khi xem.

Nhận thấy được những khuyết điểm này cũng như là điểm mạnh của từng website nên chúng em đã chọn đề tài này để mang lại cảm giác dễ chịu nhất cho người sử dụng nói chung và người xem phim nói riêng.

1.3. ĐỐI TƯỢNG, PHẠM VI VÀ PHƯƠNG PHÁP NGHIÊN CỨU

1.3.1. Đối tượng nghiên cứu

- Đò án được thực hiện dự trên hai đối tượng trọng tâm là giữa công nghệ với kiến thức thực tế về quản lý và xây dựng trang web đề xuất, xem phim cho người dùng.
- Trong đó đối tượng công nghệ bao gồm sử dụng ASP.NET Core và NodeJS để viết API, RESTful APIs, ReactJS cho giao diện và xử lý sự kiện, ML.NET để dự đoán phim cho từng người dùng. Ngoài ra, còn có các cơ sở dữ liệu như MongoDB, Cloudinary và các dịch vụ để triển khai và vận hành trang web trực tuyến như heroku.
- Kến thức thực tế về quản lý và xây dựng trang web đề xuất, xem phim cho người dùng bao gồm quy trình xử lý cho người dùng sử dụng web và người quản lý web, thống kê và phân tích số liệu.

1.3.2. Phạm vi nghiên cứu

- Phạm vi nghiên cứu được đặt ra ở mức độ tổng hợp là tìm và hiểu được kiến thức chung về các nội dung nghiên cứu và vận dụng được những kiến thức đó cho các vấn đề của đề tài.

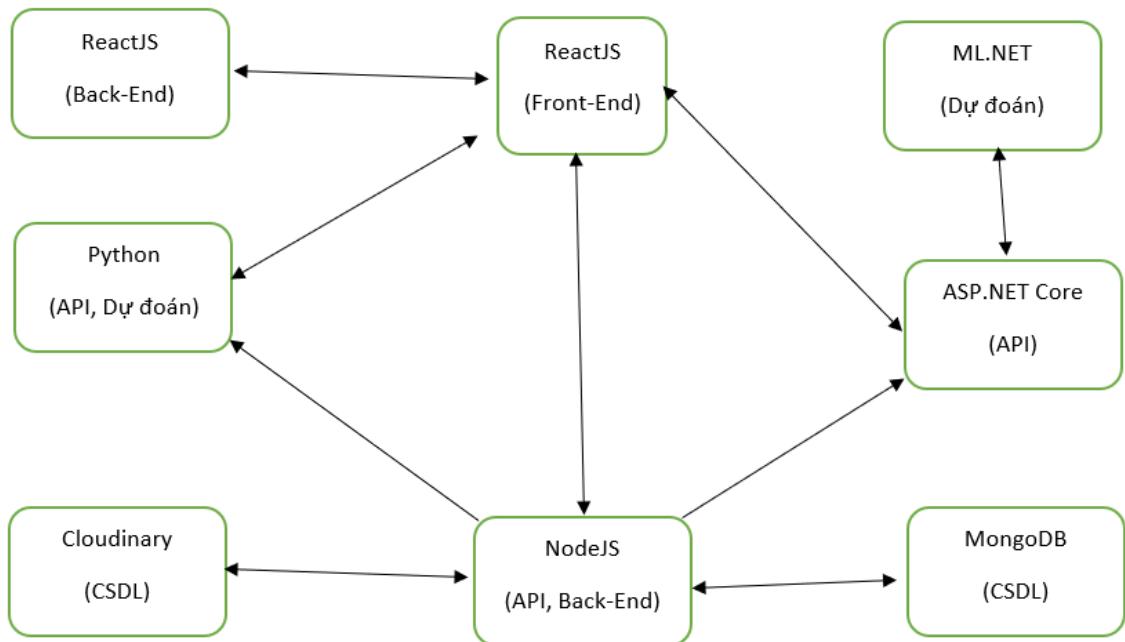
1.4. KẾT QUẢ MONG MUỐN

- Nâng vững kiến thức về ReactJS, NodeJS, ASP.NET Core, ML.NET.
- Xây dựng thành công trang web đề xuất và xem phim cho người dùng.

PHẦN NỘI DUNG

Chương 1: NGUYÊN LÝ HOẠT ĐỘNG

1.1. SƠ ĐỒ HOẠT ĐỘNG



Hình 1.1.1: Nguyên lý hoạt động của hệ thống

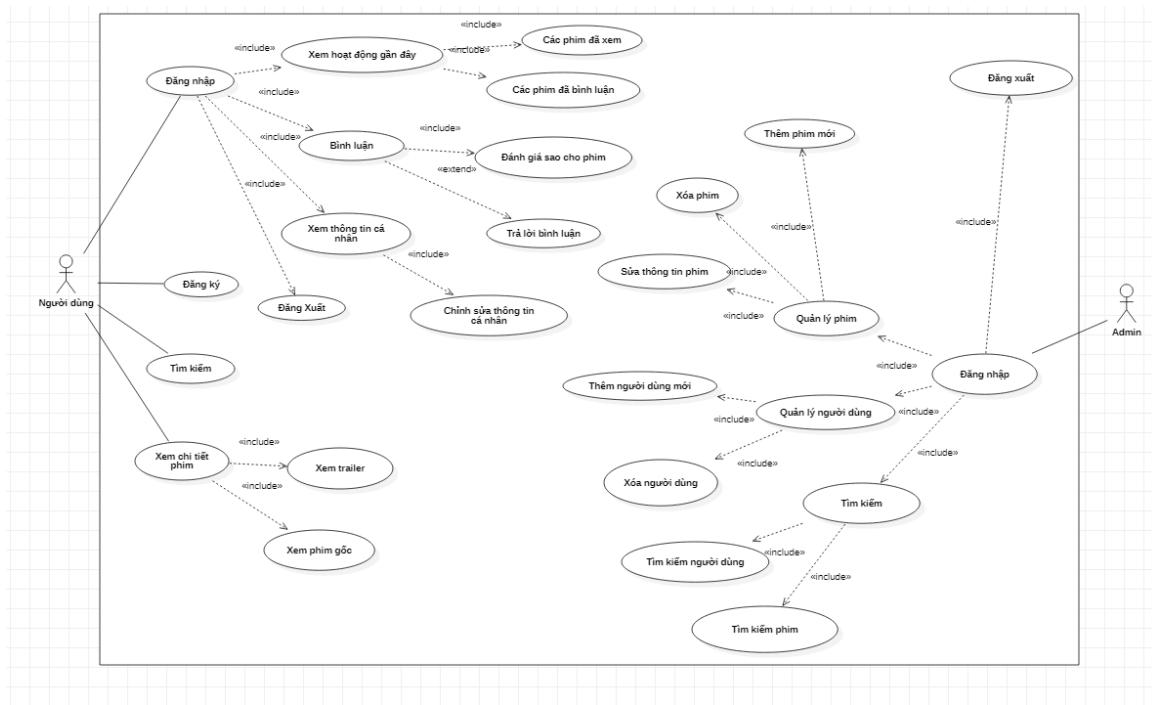
1.2. NGUYÊN LÝ HOẠT ĐỘNG

- ReactJS làm giao diện cho hệ thống khi có tương tác trên giao diện này thì sẽ chuyển cho Back-End xử lý. Cụ thể là những xử lý dữ liệu cho view hay những hoạt động không liên quan đến cơ sở dữ liệu thì được xử lý bằng ReactJS, còn những hoạt động liên quan đến cơ sở dữ liệu hay thay đổi dữ liệu thì sẽ được chuyển cho NodeJS xử lý.
- Nếu có yêu cầu lấy dữ liệu từ Front-End là React thì các API phụ trách chức năng phù hợp với yêu cầu sẽ trả dữ liệu dạng JSON cho Front-End.
- Các dữ liệu dạng chữ, số hoặc mảng sẽ được lưu tại MongoDB. Riêng có dữ liệu hình ảnh là ảnh đại diện của người dùng sẽ được lưu trên Cloudinary do MongoDB không hỗ trợ lưu ảnh.

Chương 2: THIẾT KẾ HỆ THỐNG

2.1. USECASE VÀ ĐẶT TẢ

2.1.1. Usecase



Hình 2.1.1.1: Lược đồ usecase tổng thể của hệ thống

- Lưu ý:
- Tất cả những usecase đều yêu cầu quyền truy cập mạng.
- Có thể vào thẳng usecase mà không cần trình tự thông qua url nhưng với các usecase cần đăng nhập mà chưa đăng nhập thì chuyển sang trang đăng nhập hoặc trang lỗi tương ứng.
- Tất cả các usecase của admin đều yêu cầu đăng nhập nếu chưa đăng nhập thì chuyển sang trang đăng nhập cho admin.
- Usecase “tìm kiếm người dùng” và “Tìm kiếm phim” sẽ được đặt tả chung vào phần đặt tả usecase “tìm kiếm”.

2.1.2. Đặt tả

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

2.1.1.1. Đăng ký

Bảng 2.1.2.1.1: Đặt tả usecase đăng ký

Name	Đăng ký
Brief description	Người dùng tạo tài khoản để đăng nhập vào hệ thống
Actor(s)	Người dùng
Pre-conditions	Người dùng chọn đăng ký. Nhập đầy đủ thông tin cần thiết. Chọn đăng ký.
Post-conditions	Đăng ký thành công: chuyển sang trang đăng nhập Đăng ký thất bại: hiển thị nguyên nhân thất bại.
Flow of events	
Basic flow	<ol style="list-style-type: none">Người dùng chọn đăng ký tại các trang của khách hàng.Hệ thống hiển thị cửa sổ cho người dùng nhập tên đăng nhập, ảnh đại diện, email, họ và tên, mật khẩu và nhập lại mật khẩu.Người dùng nhập đầy đủ thông tin bước 2.Hệ thống kiểm tra thông tin.Thực hiện theo Post-conditions. Nếu thất bại quay lại bước 2.
Alternative flow	
Extension point	Người dùng chọn đăng nhập thì sẽ quay lại trang đăng nhập. Người dùng chọn icon sẽ quay về trang chủ.

2.1.1.2. Đăng nhập

Bảng 2.1.2.2.1: Đặt tả usecase đăng nhập

Name	Đăng nhập
------	-----------

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

Brief description	Người dùng đăng nhập vào hệ thống
Actor(s)	Người dùng, admin
Pre-conditions	<p>Người dùng đã đăng ký tài khoản hoặc admin được cấp tài khoản trước đó.</p> <p>Chọn vào đăng nhập tại thanh header của trang người dùng.</p>
Post-conditions	<p>Đăng nhập thành công:</p> <ul style="list-style-type: none"> + Người dùng: trở về trang trước lúc đăng nhập và set cookies (thời gian lưu 12 tiếng) + Admin: Xác thực vào hệ thống và hiển thị trang quản lý phim <p>Đăng nhập thất bại: Thông báo thông tin tài khoản hoặc mật khẩu sai.</p>
Flow of events	
Basic flow (Thành công)	<p>Use case bắt đầu khi người dùng hoặc admin cần thực hiện chức năng nào đó của hệ thống cần thực hiện đăng nhập.</p> <ol style="list-style-type: none"> 1. Người dùng chọn đăng nhập tại các trang của khách hàng hoặc với admin thì truy cập vào địa chỉ đăng nhập riêng. 2. Hệ thống hiển thị cửa sổ cho người dùng nhập tên đăng nhập và mật khẩu. 3. Người dùng nhập tên đăng nhập và mật khẩu. 4. Hệ thống xác thực tên đăng nhập và mật khẩu. 5. Hệ thống vào trang quản lý phim với admin và mở trang trước trang đăng nhập với người dùng.
Alternative flow (Thất bại)	<p>Nếu người dùng nhập sai tên đăng nhập hay/mật khẩu.</p> <p>Những công việc sau được thực hiện:</p>

	<ol style="list-style-type: none">1. Hệ thống mô tả lý do xác thực bị sai.2. Hệ thống lưu ý người dùng đăng nhập lại.3. Khi người dùng nhập lại tên đăng nhập và mật khẩu, bước 3 ở Basic flow được thực hiện.
Extension point	<p>Người dùng chọn quên mật khẩu thì chuyển sang usecase quên mật khẩu</p> <p>Người dùng chọn hiển thị mật khẩu thì hiện mật khẩu cho người dùng kiểm tra lại.</p>

2.1.1.3. *Tìm kiếm***Bảng 2.1.2.3.1: Đặt tả usecase đăng tìm kiếm**

Name	Tìm kiếm
Brief description	<p>Người dùng tìm kiếm phim mình muốn xem.</p> <p>Admin tìm kiếm phim mình muốn chỉnh sửa hoặc xóa.</p>
Actor(s)	Người dùng, admin
Pre-conditions	<p>Người dùng hoặc admin nhập từ khóa vào ô tìm kiếm.</p> <p>Admin nhập từ khóa hoặc đúng id phim hoặc tên tài khoản hoặc tên người dùng vào ô tìm kiếm.</p>
Post-conditions	<p>Tìm kiếm thành công:</p> <ul style="list-style-type: none"> + Người dùng: Hiển danh sách 13 phim tại ô tìm kiếm để lựa chọn + Admin: Hiển thị danh sách, số kết quả tìm kiếm được và phân trang (mỗi trang 20 phim) <p>Tìm kiếm thất bại:</p> <ul style="list-style-type: none"> + Người dùng: không hiển thị + Admin: Hiển thị số kết quả tìm kiếm 0
Flow of events	
Basic flow (Thành công)	<p>Use case bắt đầu khi người dùng hoặc admin nhập nội dung vào ô tìm kiếm (với admin cần ấn thêm nút tìm kiếm).</p> <ul style="list-style-type: none"> + Admin: hiển thị tổng số kết quả tìm kiếm được và hiển thị 20 kết quả tìm kiếm/ trang (có phân trang). + Người dùng: hiển thị danh sách 13 phim tìm kiếm được.
Alternative flow	<p>Nếu không có kết quả tìm kiếm:</p> <ul style="list-style-type: none"> + Admin: kết quả tìm kiếm là 0

	<p>+ Người dùng: không hiển thị danh sách phim.</p> <p>Khi tìm kiếm xong:</p> <p>+ Người dùng và admin có thể sử dụng các chức năng khác hoặc chọn vào kết để quả tìm kiếm để chuyển sang usecase tiếp theo</p>
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.4. Xem chi tiết phim**Bảng 2.1.2.4.1: Đặt tả usecase chi tiết xem phim**

Name	Xem chi tiết phim
Brief description	Sau khi tìm được phim yêu thích người dùng có thể chuyển sang trang chi tiết để xem thông tin phim và phim hoặc trailer:
Actor(s)	Người dùng
Pre-conditions	Người dùng được chuyển sang trang chi tiết nếu thực hiện một trong những thao tác sau: + Chọn vào các card của phim đó + Chọn xem phim trên card + Chọn tiêu đề phim + Chọn xem ngay + Chọn vào phim lúc tìm kiếm
Post-conditions	Các thông tin cần thiết (id phim, thông tin truy xuất từ API, kết nối,...) hợp lệ thì hiển thị trang chi tiết phim với những thông tin như sau: + Tổng sao đánh giá + View + Xem trailer + Xem phim + Tiêu đề + Tagline + Overview + Doanh thu + Ngày phát hành + Ngân sách + Thời lượng phim

	<ul style="list-style-type: none"> + Thể loại + Diễn viên + Bình luận <p>Nếu thông tin không hợp lệ: Chuyển sang trang lỗi.</p>
Flow of events	
Basic flow (Thành công)	<ul style="list-style-type: none"> + Người dùng thực hiện một trong những thao tác tại Pre-conditions. + Hệ thống bắt đầu kiểm tra các thông tin cần thiết. + Hiển thị trang chi tiết với các nội dung tại Post-conditions tương thích với phim đó.
Alternative flow (Thất bại)	Nếu bước kiểm tra tại Basic flow thất bại hoặc có lỗi xảy ra thì chuyển sang trang lỗi.
Extension point	

2.1.1.5. *Xem Trailer***Bảng 2.1.2.5.1: Đặt tả usecase xem trailer**

Name	Xem Trailer
Brief description	Xem trailer của phim tại trang chi tiết của một phim nào đó.
Actor(s)	Người dùng
Pre-conditions	Người dùng chọn nút xem trailer tại trang chi tiết phim.
Post-conditions	Panel trailer được hiển thị.
Flow of events	
Basic flow	<ul style="list-style-type: none"> + Người dùng chọn nút xem trailer tại trang chi tiết phim. + Panel trailer được hiển thị.
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.6. *Xem phim gốc*Bảng 2.1.2.6.1: **Đặt tả usecase xem phim gốc**

Name	Xem phim gốc
Brief description	Xem phim của phim tại trang chi tiết của một phim nào đó.
Actor(s)	Người dùng
Pre-conditions	Người dùng chọn nút xem phim tại trang chi tiết phim.
Post-conditions	Di chuyển xuống phần xem phim để xem.
Flow of events	
Basic flow	<ul style="list-style-type: none"> + Người dùng chọn nút xem phim tại trang chi tiết phim. + Di chuyển xuống phần xem phim của trang chi tiết phim.
Alternative flow	<ul style="list-style-type: none"> + Nếu không chọn nút xem phim có thể tự di chuyển xuống để xem.
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.7. *Xem các hoạt động gần đây*Bảng 2.1.2.7.1: **Đặt tả usecase xem hoạt động gần đây**

Name	Xem các hoạt động gần đây
Brief description	Xem các hoạt động gần đây như bình luận và phim của tài khoản khi đã đăng nhập.
Actor(s)	Người dùng
Pre-conditions	<p>Người dùng đã đăng nhập.</p> <p>Người dùng di chuyển chuột lên phần tên tài khoản của header.</p> <p>Người dùng chọn hoạt động gần đây.</p>
Post-conditions	Trang hoạt động gần đây với các phim đã xem và những bình luận, đánh giá đã thực hiện được hiển thị.
Flow of events	
Basic flow	<p>Thực hiện lần lượt theo các bước sau:</p> <ul style="list-style-type: none"> + Sau khi đăng nhập thành công người dùng di chuyển chuột lên phần tên tài khoản của thanh header. + Chọn hoạt động gần đây. + Chuyển đến trang hoạt động gần đây. + Hiển thị các phim đã xem gần đây và các bình luận, đánh giá đã thực hiện.
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.8. *Các phim đã xem*Bảng 2.1.2.8.1: *Đặt tả usecase các phim đã xem*

Name	Các phim đã xem
Brief description	Xem danh sách các phim đã xem
Actor(s)	Người dùng
Pre-conditions	Người dùng thực hiện usecase các hoạt động gần đây
Post-conditions	Hiển thị danh sách những phim đã xem
Flow of events	
Basic flow	Sau khi thực hiện usecase các hoạt động gần đây người dùng chọn “đã xem gần đây” tại trang hoạt động gần đây sau đó hiển thị cho người dùng danh sách các phim đã xem.
Alternative flow	+ Tại danh sách các phim đã xem người dùng có thể chọn phim nào đó. + Thực hiện usecase “Xem chi tiết phim”
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.9. *Các phim đã bình luận***Bảng 2.1.2.9.1: Đặt tả usecase các phim đã bình luận**

Name	Các phim đã bình luận
Brief description	Xem danh sách các phim đã bình luận
Actor(s)	Người dùng
Pre-conditions	Người dùng thực hiện usecase các hoạt động gần đây
Post-conditions	Hiển thị danh sách những phim đã bình luận
Flow of events	
Basic flow	Sau khi thực hiện usecase các hoạt động gần đây người dùng chọn “đã bình luận” tại trang hoạt động gần đây sau đó hiển thị cho người dùng danh sách các phim đã bình luận.
Alternative flow	+ Tại danh sách các phim đã bình luận người dùng có thể chọn phim nào đó. + Thực hiện usecase “Xem chi tiết phim”
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.10. Bình luận

Bảng 2.1.2.10.1: Đặt tả usecase bình luận

Name	Bình luận
Brief description	Bình luận, đánh giá với mỗi bộ phim
Actor(s)	Người dùng
Pre-conditions	Người dùng thực hiện usecase xem chi tiết phim. Người dùng đã đăng nhập.
Post-conditions	Thêm bình luận, đánh giá vào phần bình luận và mọi người có thể xem mà không cần tải lại trang.
Flow of events	
Basic flow	<p>Sau khi thực hiện usecase xem chi tiết phim người dùng phải trong trạng thái đăng nhập để sử dụng chức năng này.</p> <ul style="list-style-type: none"> + Người dùng nhập đánh giá tại phần đánh giá (đánh giá sao và nội dung) - Mỗi tài khoản chỉ có một đánh giá với một bộ phim nào đó nhưng có thể có nhiều trả lời đánh giá đối với cùng phim đó. + Người dùng chọn trả lời đánh giá của người khác. + Sau khi nhập thông tin cần thiết người dùng chọn gửi. + Thêm bình luận, đánh giá của người dùng vừa bình luận vào phần bình luận của phim.
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.11. Xem thông tin cá nhân

Bảng 2.1.2.11.1: Đặt tả usecase xem thông tin cá nhân

Name	Xem thông tin cá nhân
Brief description	Xem thông tin tài khoản đã đăng nhập.
Actor(s)	Người dùng
Pre-conditions	<p>Người dùng đã đăng nhập</p> <p>Người dùng di chuyển chuột lên phần tên tài khoản của thanh header.</p> <p>Người dùng chọn “thông tin tài khoản”</p>
Post-conditions	Trang thông tin tài khoản được hiển thị với những thông tin phù hợp với tài khoản đã đăng nhập bao gồm:
	<ul style="list-style-type: none"> + Họ và tên + Tên đăng nhập + Email + Ảnh đại diện + Tổng số bình luận, đánh giá
Flow of events	
Basic flow	Sau khi thực hiện các bước tại Pre-conditions thì tiến hành truy xuất thông tin tài khoản và hiển thị những thông tin cho phần post-conditions.
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.12. Sửa thông tin cá nhân

Bảng 2.1.2.12.1: Đặt tả usecase sửa thông tin cá nhân

Name	Sửa thông tin cá nhân
Brief description	Sửa thông tin tài khoản đã đăng nhập.
Actor(s)	Người dùng
Pre-conditions	Sau khi thực hiện usecase “Xem thông tin cá nhân”. Người dùng chọn “chỉnh sửa”
Post-conditions	Trang thông tin tài khoản được hiển thị với những thông tin đã được chỉnh sửa của tài khoản đã đăng nhập bao gồm: + Họ và tên + Tên đăng nhập + Email + Ảnh đại diện + Mật khẩu
Flow of events	
Basic flow	Sau khi thực hiện các bước tại Pre-conditions thì cho phép người dùng chỉnh sửa những thông tin đó ngoại trừ “tên đăng nhập” Nếu người dùng chọn “Lưu”: lưu những thông tin đã chỉnh sửa. Nếu người dùng chọn “Hủy Bỏ”: không lưu những thông tin đã chỉnh sửa.
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.13. Đăng xuất**Bảng 2.1.2.13.1: Đặt tả usecase đăng xuất**

Name	Đăng xuất
Brief description	Đăng xuất khỏi hệ thống.
Actor(s)	Người dùng, admin
Pre-conditions	Sau khi thực hiện xong usecase đăng nhập. Người dùng hoặc admin chọn đăng xuất.
Post-conditions	Đăng xuất khỏi hệ thống và xóa cookies đăng nhập.
Flow of events	
Basic flow	Người dùng hoặc admin chọn “đăng xuất” + Người dùng: Xóa cookies và đăng xuất khỏi hệ thống. + Admin: đăng xuất khỏi hệ thống.
Alternative flow	+ Người dùng: khi đăng nhập quá 12 tiếng tự động đăng xuất.
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.14. Quản lý người dùng

Bảng 2.1.2.14.1: Đặt tả usecase quản lý người dùng

Name	Quản lý người dùng
Brief description	Quản lý các tài khoản người dùng và admin
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase đăng nhập cho admin. Admin chọn “Users” trên thanh header.
Post-conditions	Trang quản lý người dùng được hiển thị với những yêu cầu sau: + Mỗi trang chứa 20 tài khoản + Chứa các thông tin như: username, name, email, role, avatar, comments
Flow of events	
Basic flow	Thực hiện các bước nhu pre-conditions
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.15. Thêm người dùng mới

Bảng 2.1.2.15.1: Đặt tả usecase thêm người dùng mới

Name	Thêm người dùng mới
Brief description	Thêm một tài khoản admin hoặc người dùng mới
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase quản lý người dùng, Admin tiếp tục chọn add
Post-conditions	Tài khoản mới được tạo với mật khẩu đã băm
Flow of events	
Basic flow	<p>Sau khi thực hiện như Pre-conditions thực hiện lần lượt theo những bước sau:</p> <ul style="list-style-type: none"> + Admin nhập những thông tin cần thiết cho tạo tài khoản (full name, email, password, re-password, username, role, avatar) + Admin chọn “Add” + Hệ thống kiểm tra tài khoản, email đã tồn tại hay chưa và các thông tin cần thiết có đúng không + Nếu đúng tạo tài khoản mới với những thông tin đã nhập + Thông báo tạo thành công
Alternative flow	<ul style="list-style-type: none"> + Tại bước kiểm tra nếu thông tin sai hoặc tài khoản, email đã tồn tại + Thông báo và yêu cầu nhập lại
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.16. Xóa người dùng**Bảng 2.1.2.16.1: Đặt tả usecase xóa người dùng**

Name	Xóa người dùng
Brief description	Xóa tài khoản người dùng, admin đã đăng ký
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase quản lý người dùng, Admin tiếp tục chọn option “delete”
Post-conditions	Tài khoản muốn xóa bị xóa khỏi hệ thống
Flow of events	
Basic flow	Sau khi thực hiện như Pre-conditions thực hiện lần lượt theo những bước sau: + Thông báo xác nhận xuất hiện + Chọn “delete” + Xóa tài khoản tương ứng khỏi hệ thống
Alternative flow	+ Tại bước chọn “delete” mà admin chọn “hủy” + Quay lại usecase quản lý người dùng
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.17. Quản lý phim

Bảng 2.1.2.17.1: Đặt tả usecase quản lý phim

Name	Quản lý phim
Brief description	Quản lý các phim trong hệ thống
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase đăng nhập cho admin hoặc admin chọn “Films” trên thanh header khi đã đăng nhập.
Post-conditions	Trang quản lý người dùng được hiển thị với những yêu cầu sau: + Mỗi trang chứa 20 tài khoản + Chứa các thông tin như: id, title, comment, rate, view, revenue
Flow of events	
Basic flow	Thực hiện các bước như pre-conditions
Alternative flow	
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.18. *Thêm phim mới***Bảng 2.1.2.18.1: Đặt tả usecase thêm phim mới**

Name	Thêm phim mới
Brief description	Thêm phim mới cho hệ thống
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase quản lý phim, Admin tiếp tục chọn option “edit”
Post-conditions	Phim mới được lưu vào hệ thống
Flow of events	
Basic flow	<p>Sau khi thực hiện như Pre-conditions trang chỉnh sửa xuất hiện và admin cần thực hiện lần lượt theo những bước sau:</p> <ul style="list-style-type: none"> + Admin sửa những thông tin cần thiết cho thông tin phim mới (Name, Slogan, link poster, link background, description, revenue, release date, video ID) + Admin chọn “Edit” + Hệ thống kiểm tra các thông tin cần thiết có đúng không + Nếu đúng tạo tài khoản mới với những thông tin đã nhập + Thông báo tạo thành công
Alternative flow	<ul style="list-style-type: none"> + Tại bước kiểm tra nếu thông tin sai + Thông báo và yêu cầu nhập lại
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.19. Xóa phim

Bảng 2.1.2.19.1: Đặt tả usecase xóa phim

Name	Xóa phim
Brief description	Xóa phim khỏi hệ thống
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase quản lý phim, Admin tiếp tục chọn option “delete”
Post-conditions	Phim muốn xóa bị xóa khỏi hệ thống
Flow of events	
Basic flow	<p>Sau khi thực hiện như Pre-conditions thực hiện lần lượt theo những bước sau:</p> <ul style="list-style-type: none"> + Thông báo xác nhận xuất hiện + Chọn “delete” + Xóa phim tương ứng khỏi hệ thống
Alternative flow	<ul style="list-style-type: none"> + Tại bước chọn “delete” mà admin chọn “hủy” + Quay lại usecase quản lý phim
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.1.1.20. Sửa thông tin phim

Bảng 2.1.2.20.1: Đặt tả usecase sửa thông tin phim

Name	Sửa thông tin phim
Brief description	Sửa các thông tin phim đã có trên hệ thống
Actor(s)	Admin
Pre-conditions	Sau khi thực hiện xong usecase quản lý phim, Admin tiếp tục chọn add
Post-conditions	Thông tin phim mới được lưu vào hệ thống
Flow of events	
Basic flow	<p>Sau khi thực hiện như Pre-conditions thực hiện lần lượt theo những bước sau:</p> <ul style="list-style-type: none"> + Admin nhập những thông tin cần thiết cho thêm phim mới (Name, Slogan, link poster, link background, description, revenue, release date, video ID) + Admin chọn “Edit” + Hệ thống kiểm tra các thông tin cần thiết có đúng không + Nếu đúng lưu phim mới với những thông tin đã nhập + Thông báo tạo thành công
Alternative flow	<ul style="list-style-type: none"> + Tại bước kiểm tra nếu thông tin sai + Thông báo và yêu cầu nhập lại
Extension point	Chuyển sang trang lỗi nếu có lỗi xảy ra.

2.2. ĐẶC TÍCH UI

2.2.1. Chức năng đăng nhập

2.2.1.1. Chưa đăng nhập

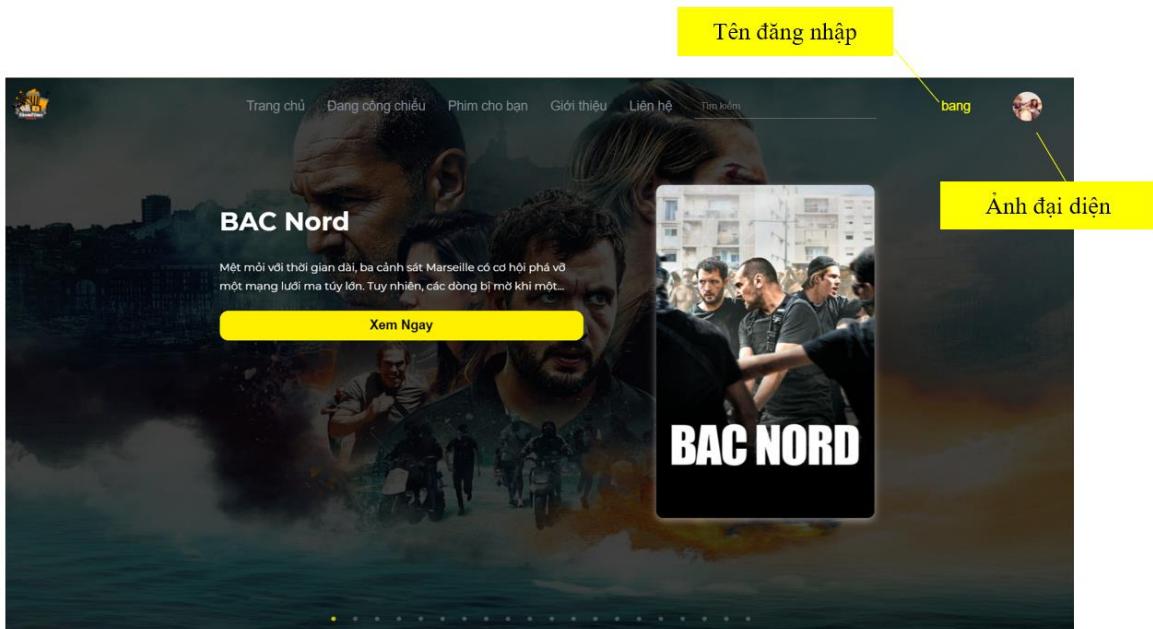


Hình 2.2.1.1 Chưa đăng nhập ở chức năng đăng nhập

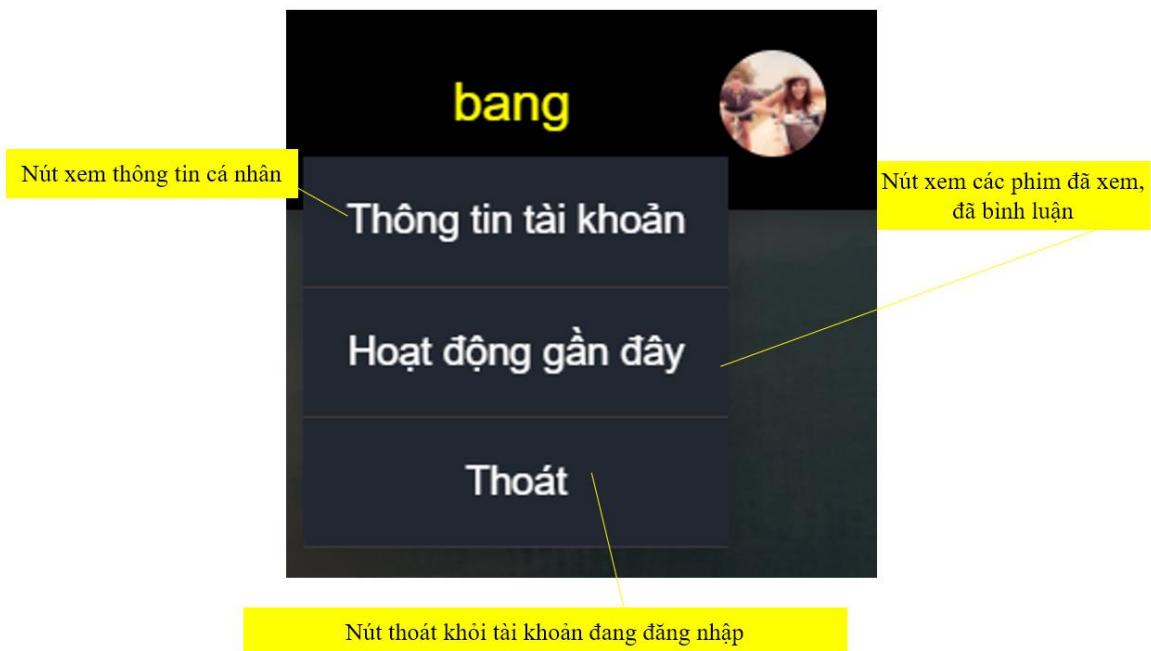
2.2.1.2. Đăng nhập thành công

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

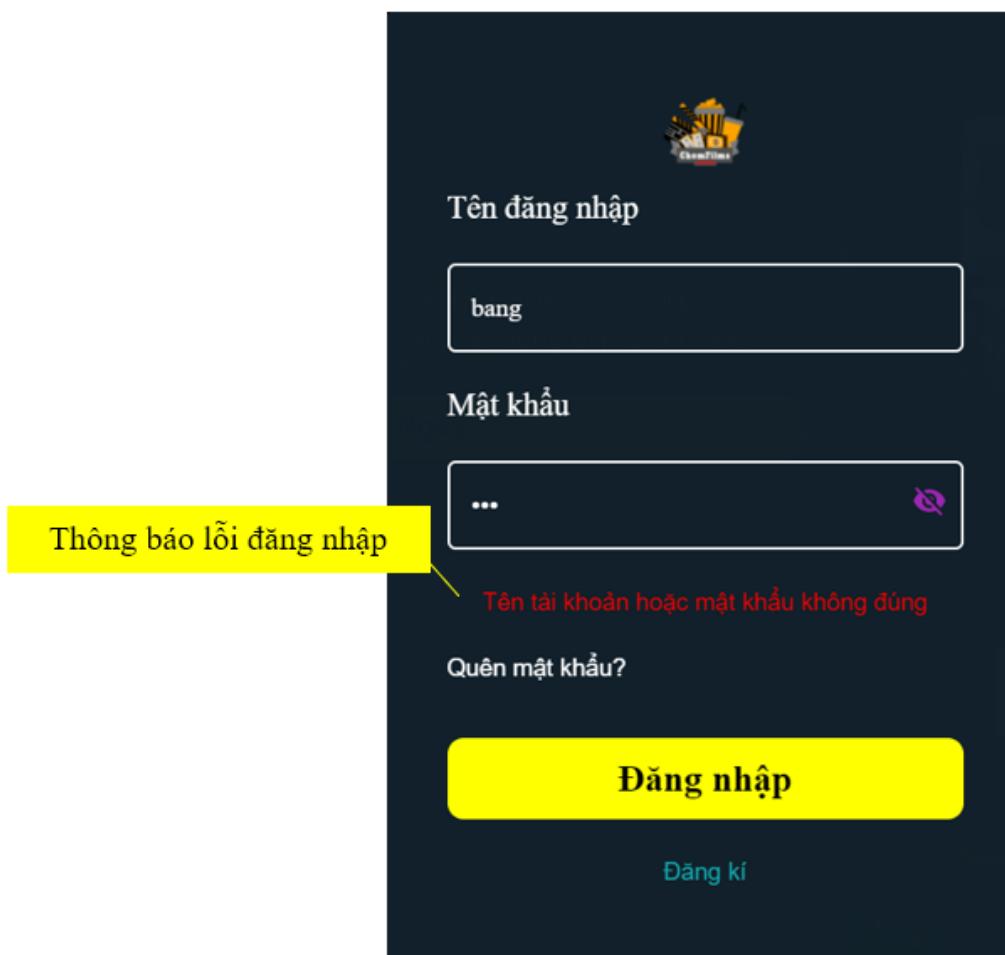


Hình 2.2.1.2.1 Đăng nhập thành công ở trang chủ



Hình 2.2.1.2.2 Hover vào tên đăng nhập sau khi đăng nhập thành công

2.2.1.3. Đăng nhập sai thông tin



Hình 2.2.1.2 Đăng nhập sai thông tin ở chức năng đăng nhập

2.2.1.4. Quên mật khẩu

2.2.1.4.1. Chưa quên mật khẩu



XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

Hình 2.2.1.3.1 Chưa quên mật khẩu ở chức năng quên mật khẩu

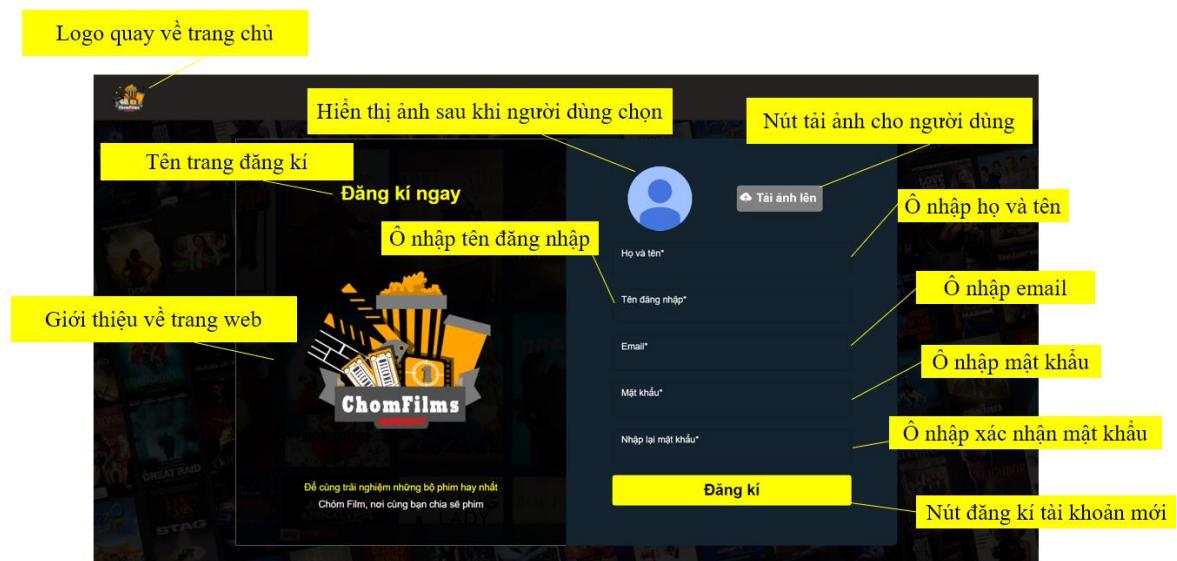
2.2.1.4.2. Quên mật khẩu với thông tin sai



Hình 2.2.1.3.2 Quên mật khẩu với thông tin sai với chức năng quên mật khẩu

2.2.2. Chức năng đăng ký

2.2.2.1. Chưa đăng ký



Hình 2.2.2.1 Chưa đăng kí ở trang đăng kí

2.2.2.2. Đăng kí sai thông tin

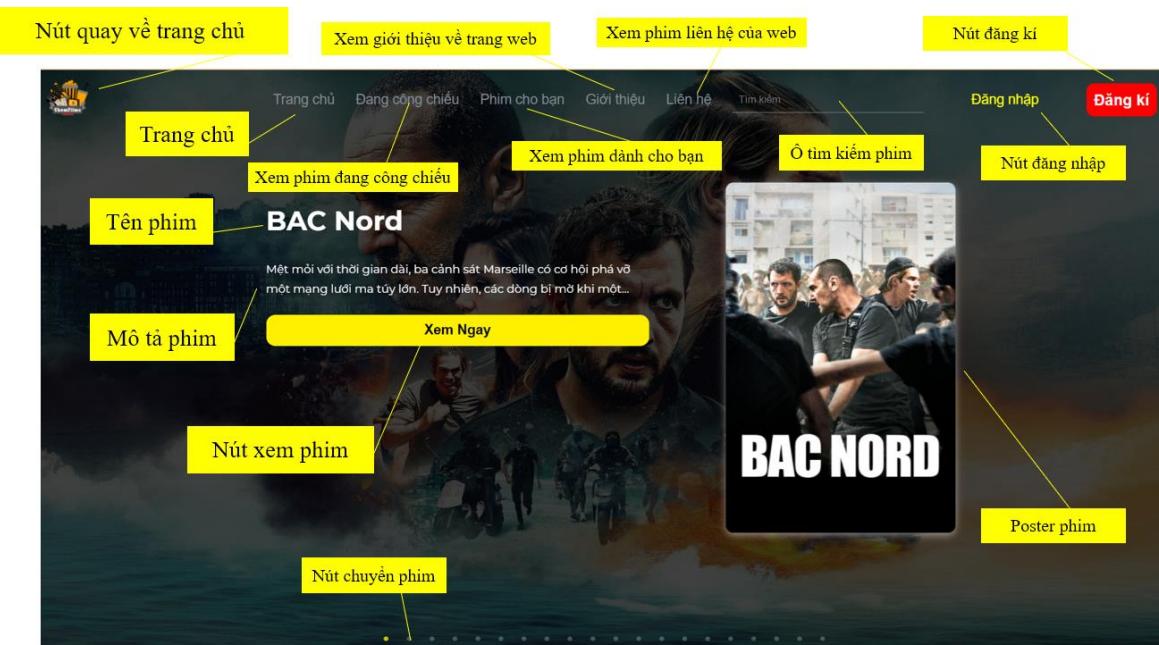
The screenshot shows a registration form with the following fields and errors:

- Họ và tên***: thanh bagn (highlighted in yellow)
- Tên đăng nhập***: nguyenLong (highlighted in yellow)
- Email***: bangnguyen123@gmail.com
- Mật khẩu***: (empty field)
- Please enter your Password**
- Nhập lại mật khẩu***: (empty field)
- Re - Password does not match password**
- Đăng kí** button (highlighted in yellow)

Hình 2.2.2.2 Đăng kí với thông tin sai ở trang đăng kí

2.2.3. Trang chủ

2.2.3.1. Danh sách phim phổ biến



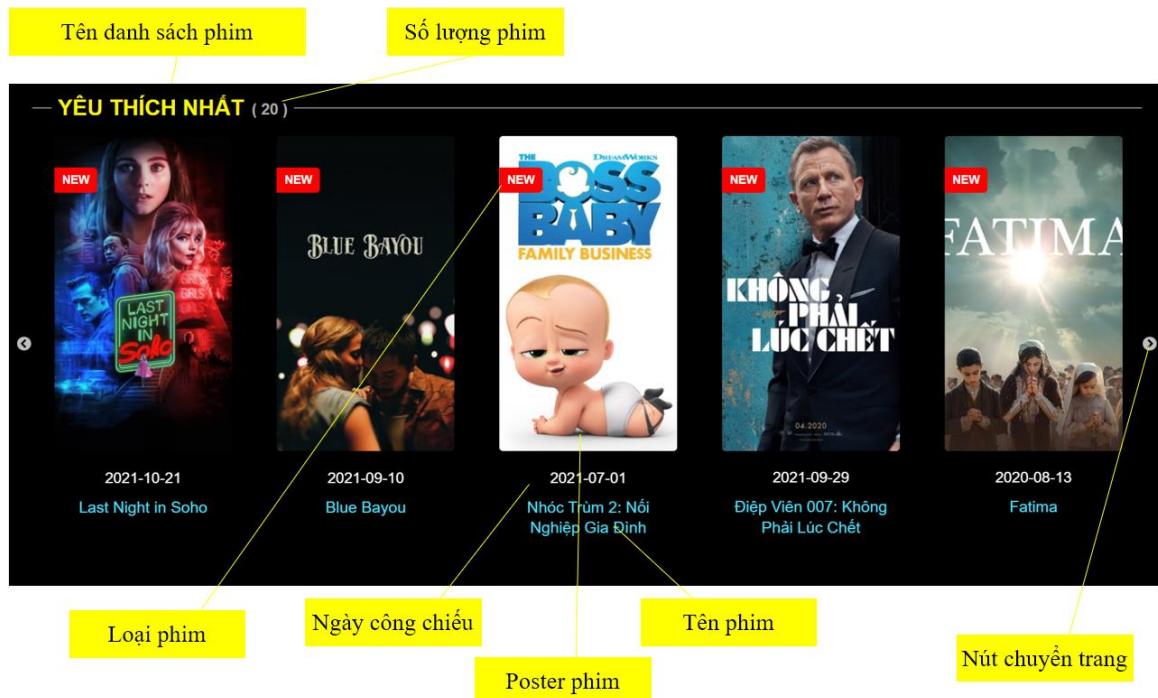
Hình 2.2.3.1 Danh sách phim phổ biến ở trang chủ

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

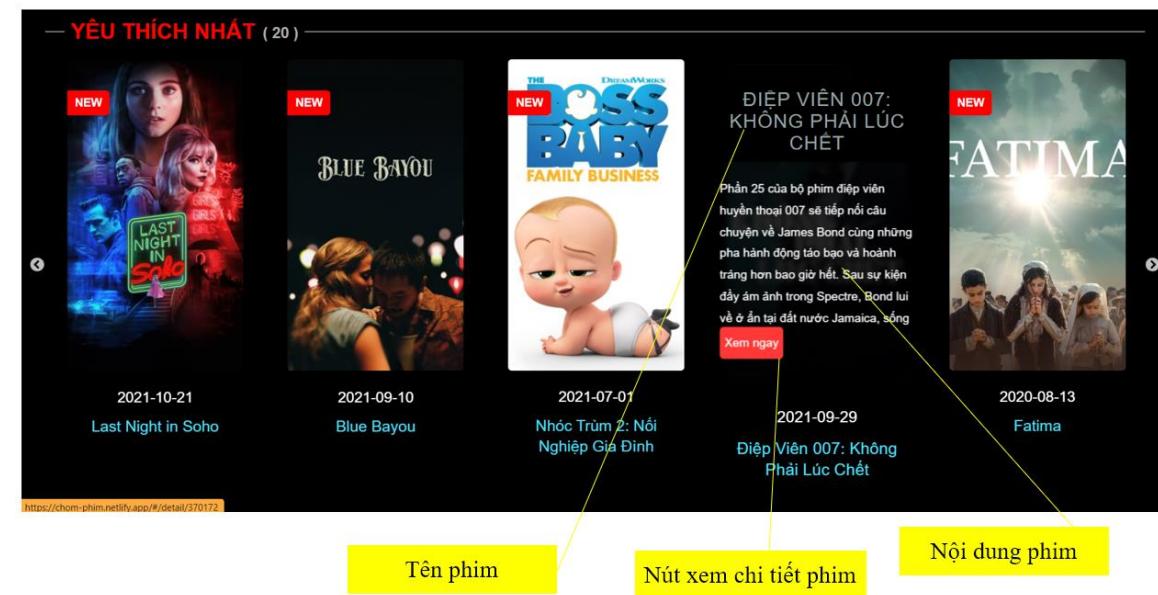
2.2.3.2. Danh sách phim được yêu thích nhất

2.2.3.2.1. Xem danh sách phim



Hình 2.2.3.2.1 Xem danh sách phim yêu thích ở trang chủ

2.2.3.2.2. Hover vào phim



Hình 2.2.3.2.2 Phim khi được hover ở mục phim yêu thích

2.2.3.3. Danh sách phim được gợi ý cho người dùng

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

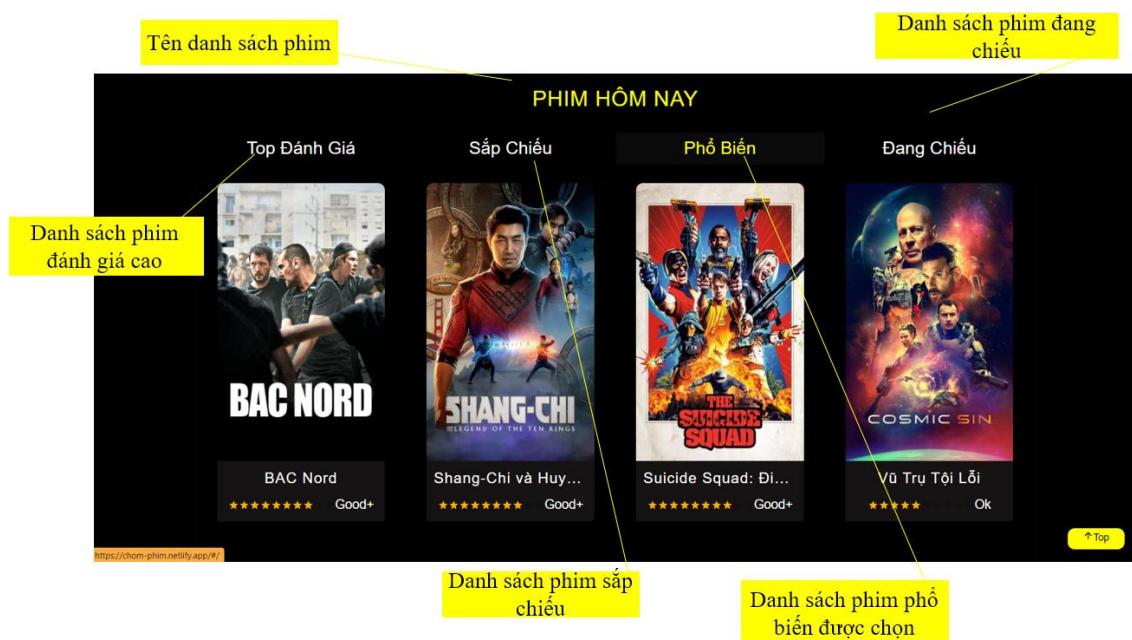
GVHD: TS. Huỳnh Xuân Phụng



Hình 2.2.3.3 Danh mục phim được gợi ý cho người dùng

2.2.3.4. Danh mục phim

2.2.3.4.1. Danh mục mặc định



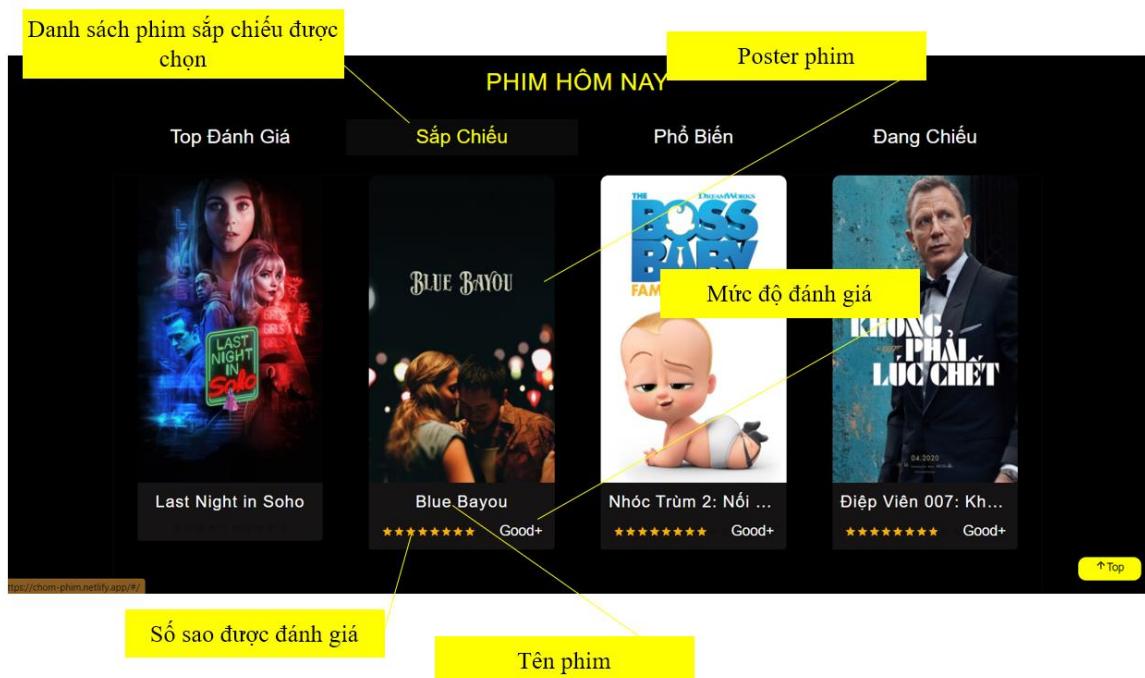
Hình 2.2.3.4.1 Danh mục phim mặc định khi vào trang web

2.2.3.4.2. Top phim đánh giá



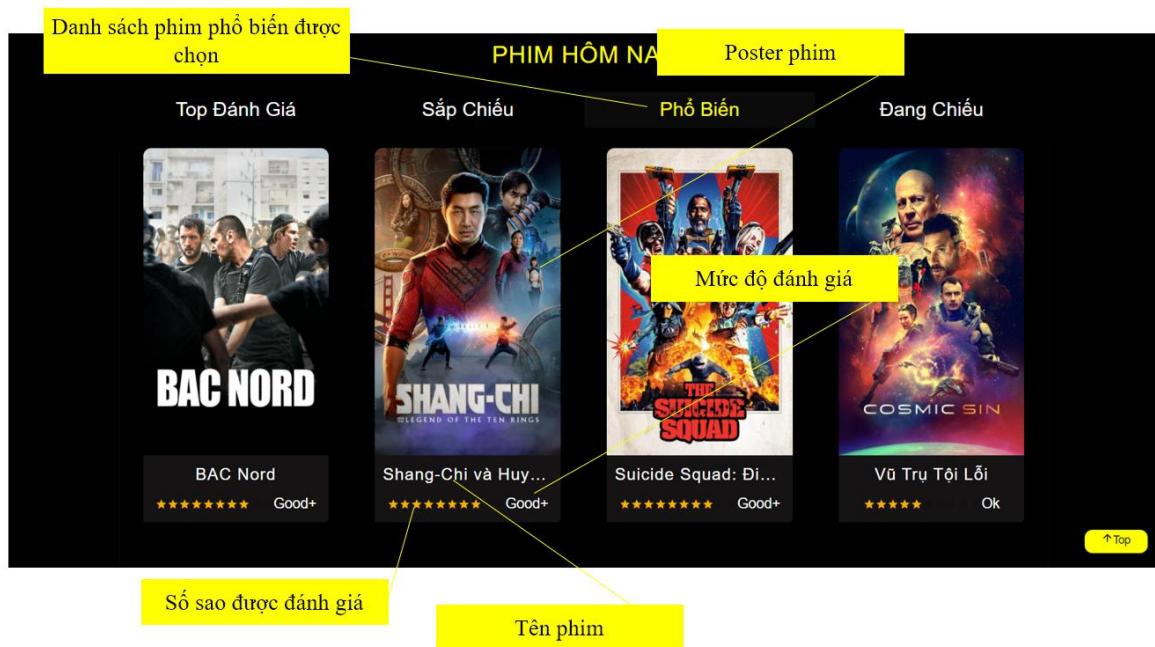
Hình 2.2.3.4.2 Danh sách phim được đánh giá cao ở trang chủ

2.2.3.4.3. Phim sắp chiếu



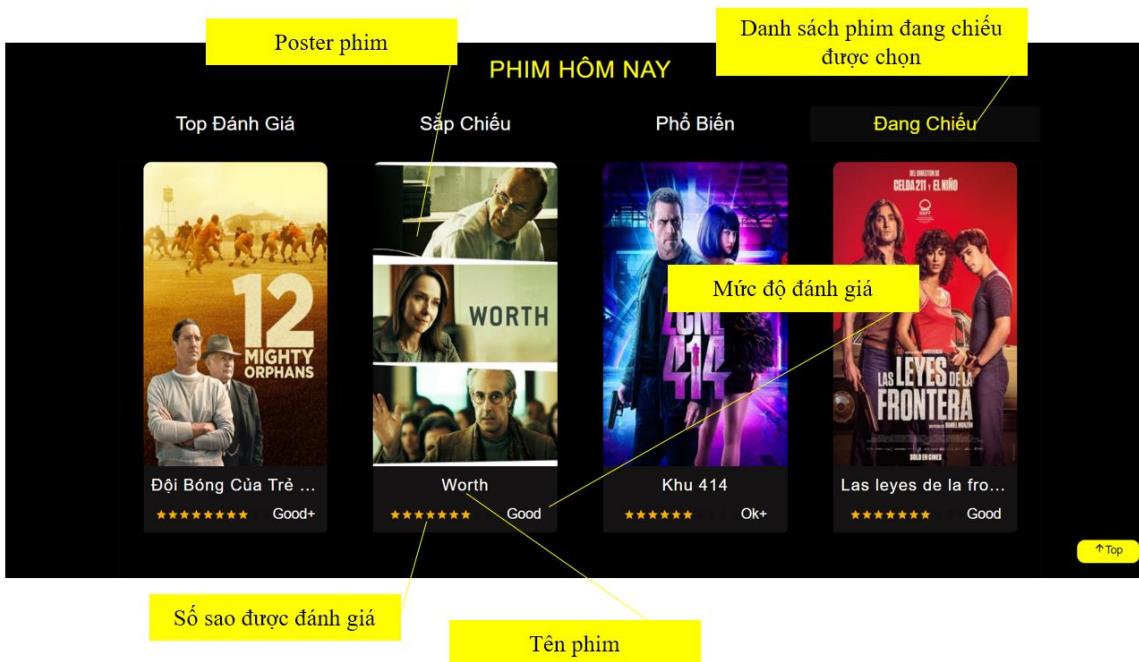
Hình 2.2.3.4.3 Danh sách phim sắp chiếu ở trang chủ

2.2.3.4.4. Phim phô biến



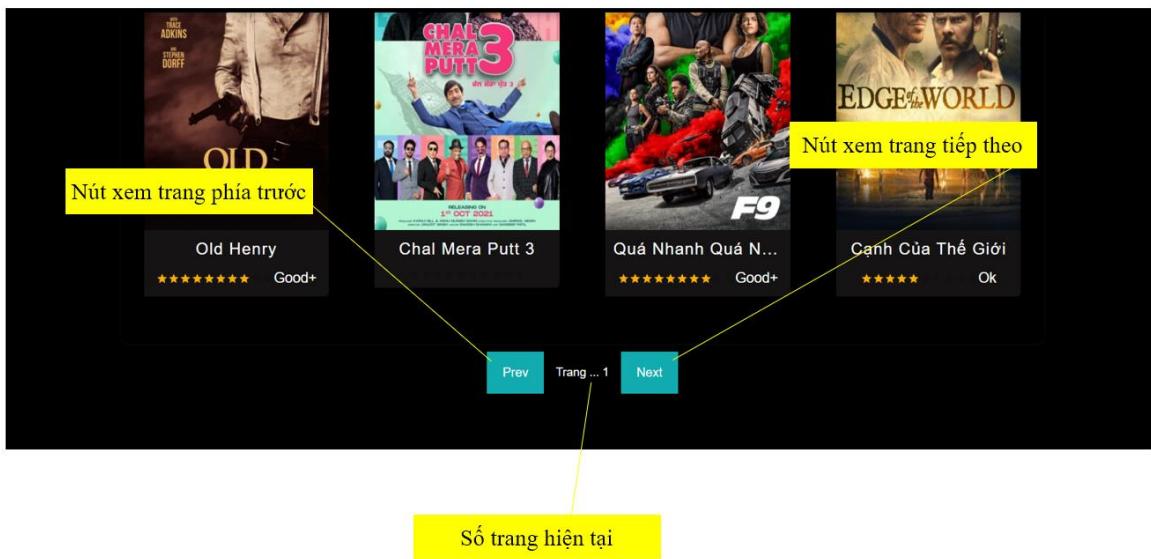
Hình 2.2.3.4.4 Danh sách phim phổ biến ở trang chủ

2.2.3.4.5. Phim đang chiếu



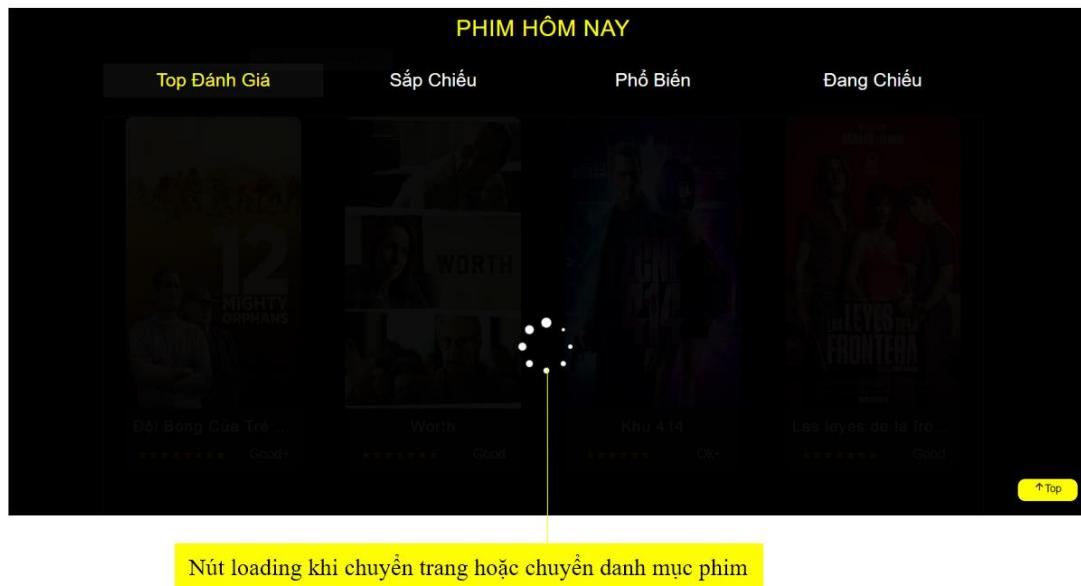
Hình 2.2.3.4.5 Danh sách phim đang chiếu ở trang chủ

2.2.3.4.6. Phân trang



Hình 2.2.3.4.6 Phân trang tại danh mục phim ở trang chủ

2.2.3.4.7. Màn hình chờ khi phân trang hoặc chuyển danh mục



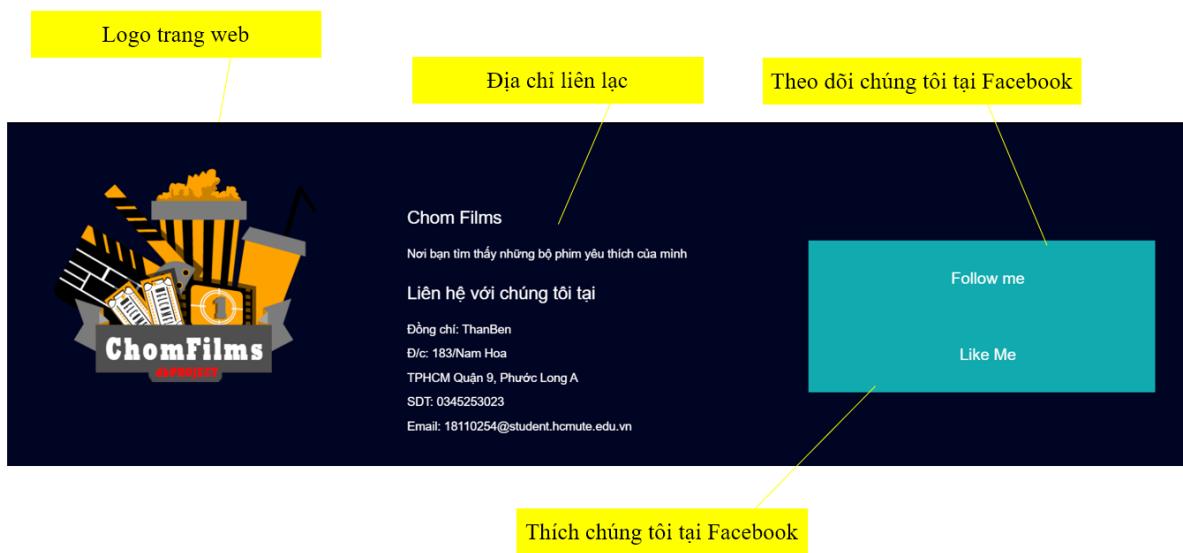
Hình 2.2.3.4.7 Màn hình chờ khi phân trang hoặc chuyển danh mục tại trang chủ

2.2.3.5. Thông tin trang web



Hình 2.2.3.5 Thông tin trang web

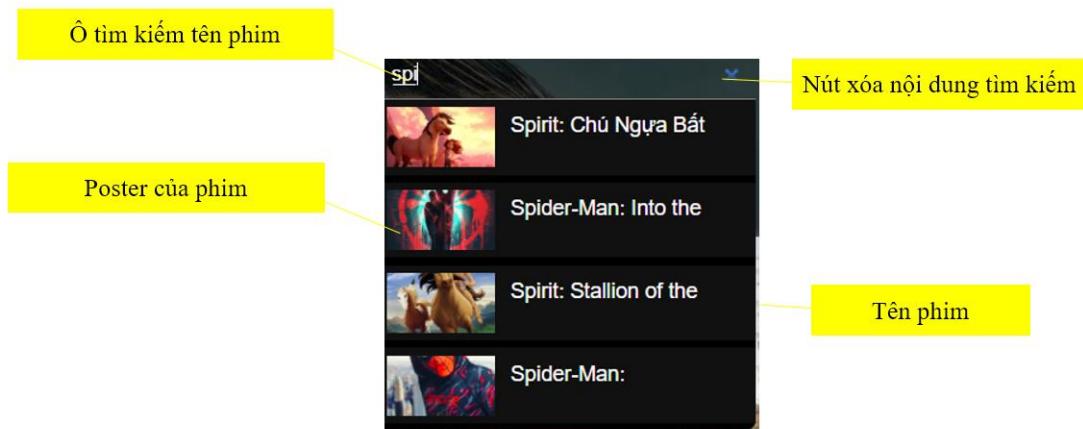
2.2.3.6. *Liên hệ*



Hình 2.2.3.6 Liên hệ về trang web

2.2.3.7. *Chức năng tìm kiếm*

2.2.3.7.1. Tìm kiếm phim tồn tại



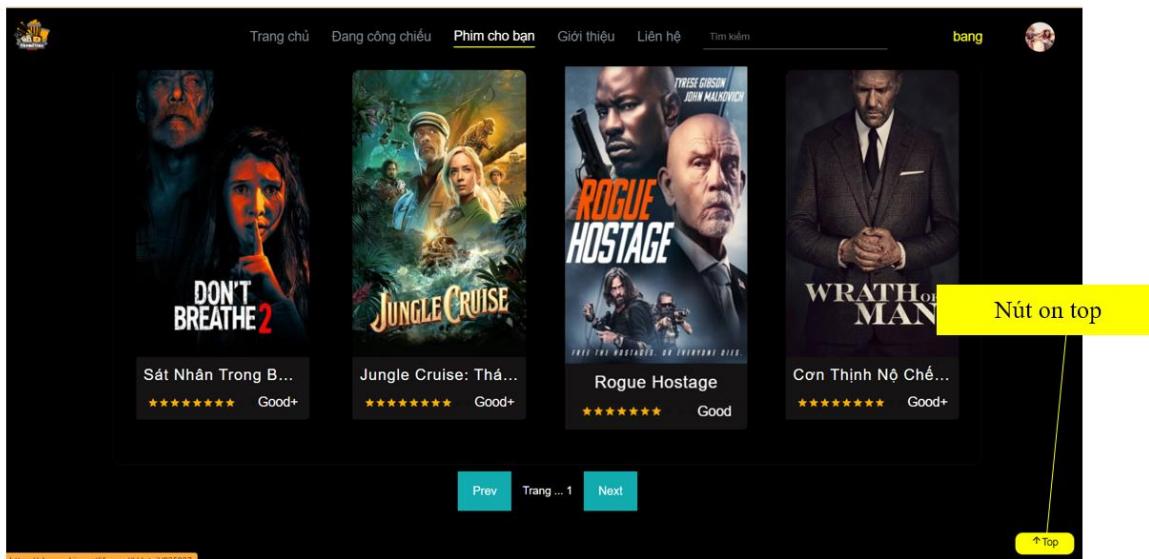
Hình 2.2.3.7.1 *Chức năng tìm kiếm với phim tồn tại*

2.2.3.7.2. *Tìm kiếm phim không tồn tại*



Hình 2.2.3.7.2 *Chức năng tìm kiếm phim không tồn tại*

2.2.3.8. *Nút On Top*

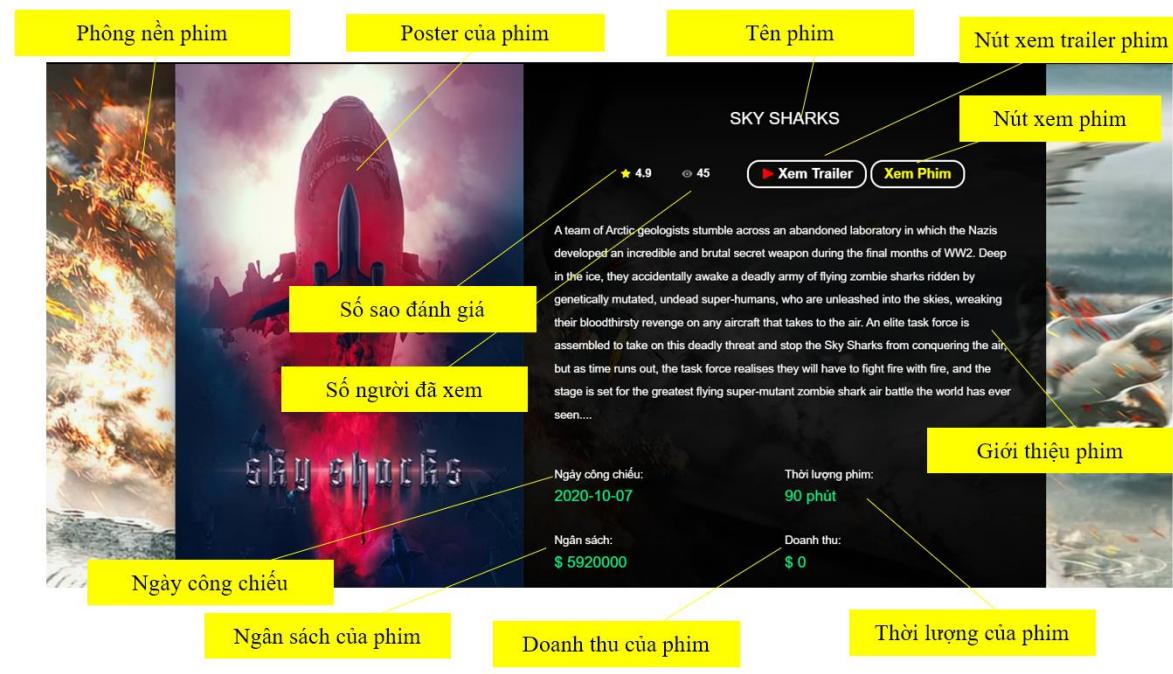


Hình 2.2.3.8 Nút on top ở trang chủ

2.2.4. Trang chi tiết phim

2.2.4.1. Chi tiết phim

2.2.4.1.1. Thông tin phim



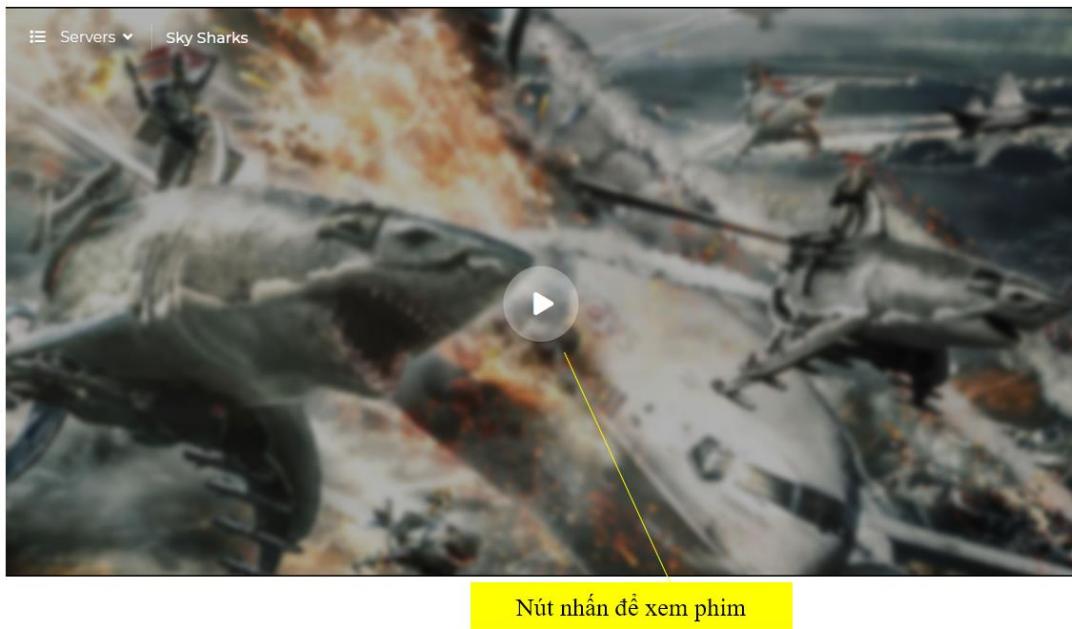
Hình 2.2.4.1.1.1 Thông tin chi tiết của phim

2.2.4.1.2. Danh sách diễn viên



Hình 2.2.4.1.1.2 Danh sách diễn viên chính trong phim

2.2.4.2. Xem phim



Hình 2.2.4.2 Xem phim ở trang chi tiết phim

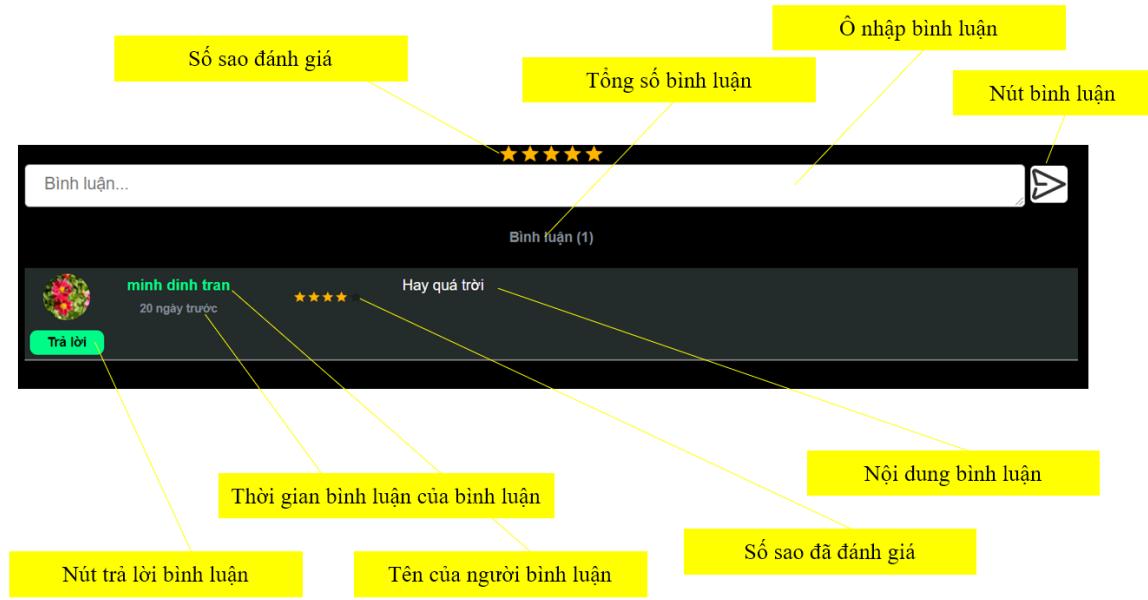
2.2.4.3. Bình luận phim

2.2.4.3.1. Chưa đăng nhập



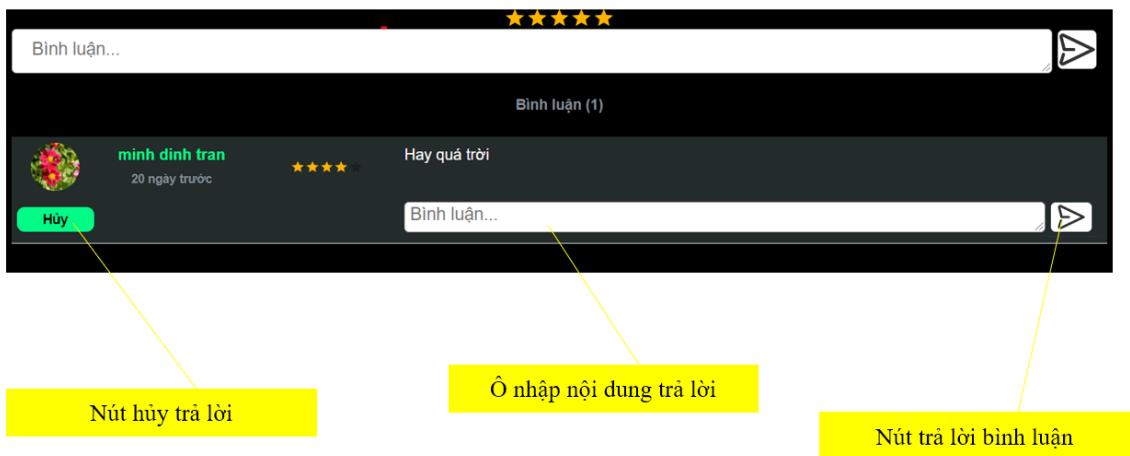
Hình 2.2.4.3.1 Yêu cầu người dùng đăng nhập để được bình luận

2.2.4.3.2. Đã đăng nhập



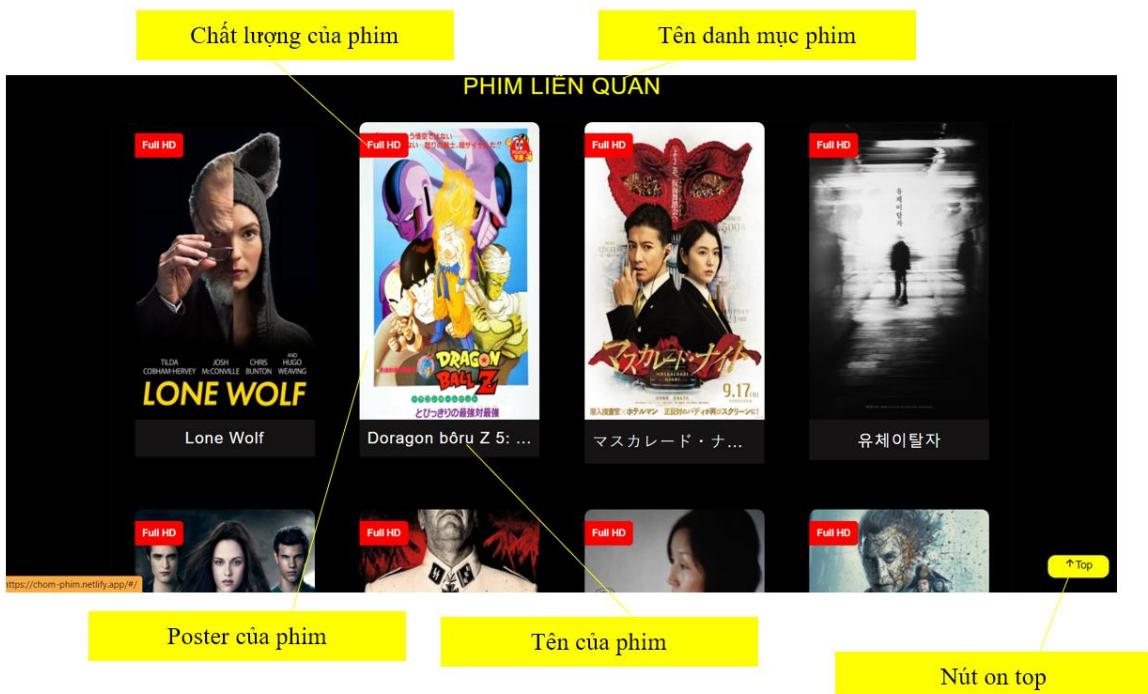
Hình 2.2.4.3.2 Bình luận nếu người dùng đã đăng nhập

2.2.4.3.3. Trả lời bình luận



Hình 2.2.4.3.3 Trả lời bình luận của người khác

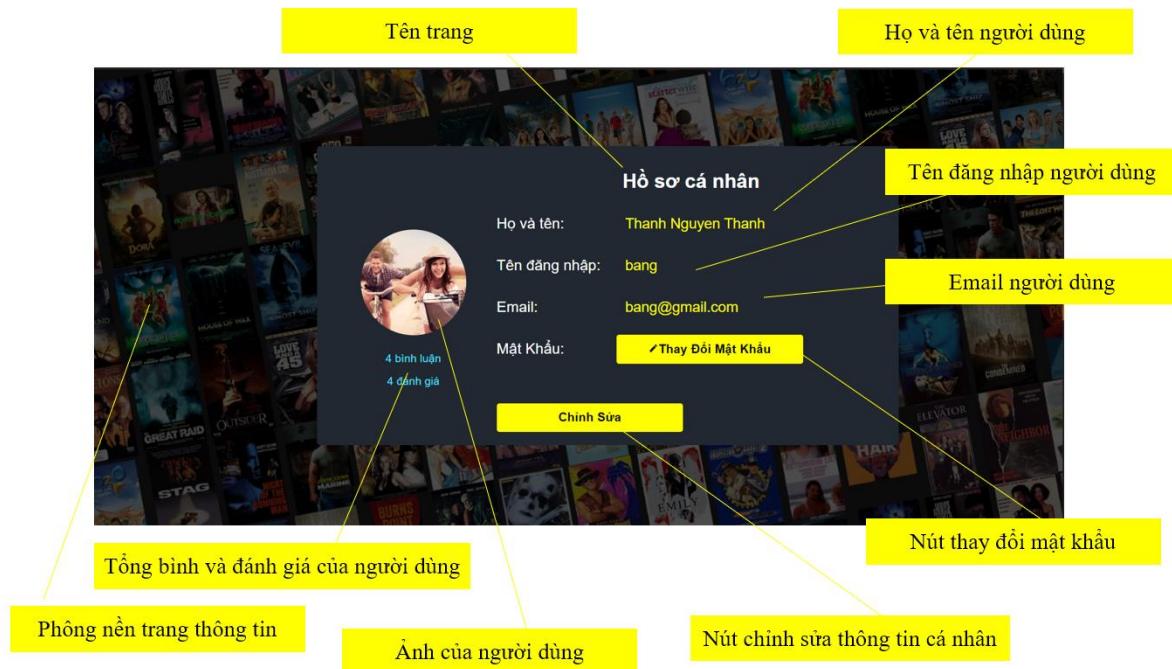
2.2.4.4. Danh sách phim liên quan



Hình 2.2.4.4 Danh sách phim liên quan

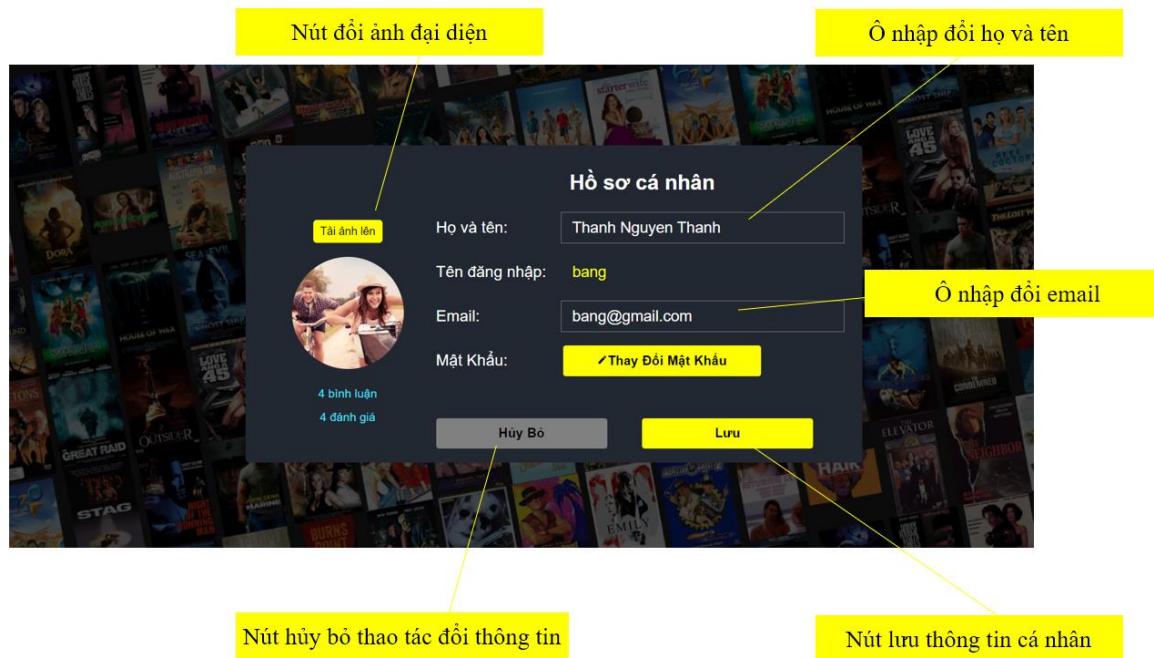
2.2.5. Trang thông tin cá nhân

2.2.5.1. Xem thông tin cá nhân



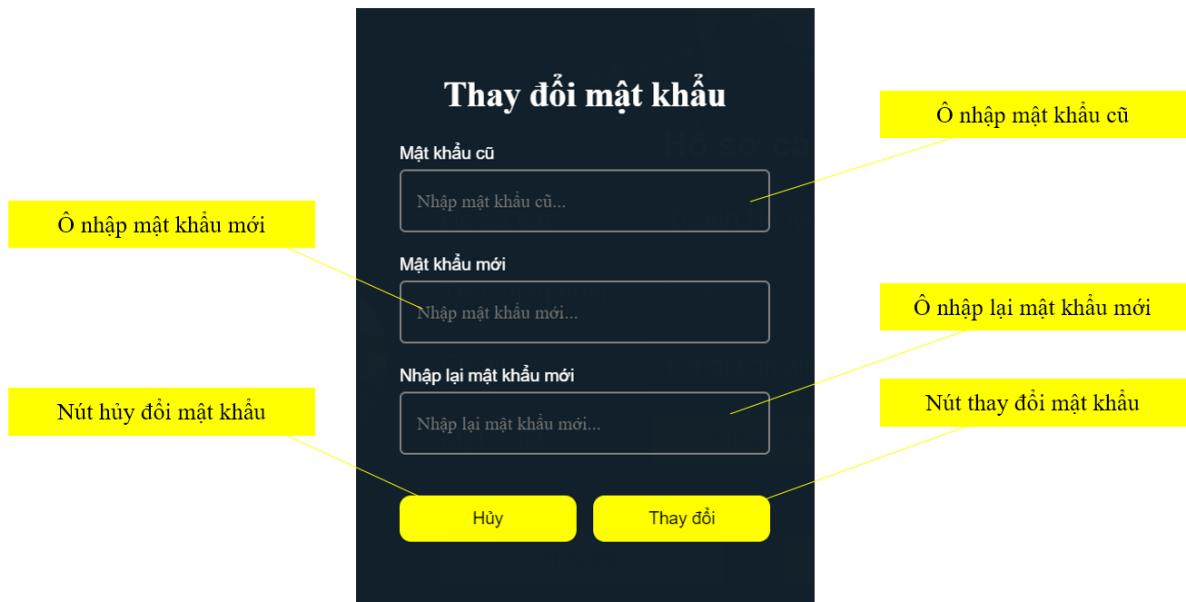
Hình 2.2.5.1 Xem thông tin cá nhân của người dùng

2.2.5.2. *Chỉnh sửa thông tin cá nhân*



Hình 2.2.5.2 Thay đổi thông tin cá nhân

2.2.5.3. *Thay đổi mật khẩu*



Hình 2.2.5.3 Thay đổi mật khẩu của người dùng

2.2.6. Trang hoạt động gần đây

2.2.6.1. Phim đã xem

Số sao đã đánh giá

Thể loại của phim

Tên hoạt động

Poster của phim

Tên của phim

Phim	Tựa đề	Danh giá	Thể loại phim
	Sky Sharks	4.9 ★	Phim Kinh Dị, Phim Hài, Phim Khoa Học Viễn Tưởng,
	Bố Già	8.7 ★	Phim Chính Kịch, Phim Hình Sự,
	Ác Mộng Không Lối Thoát	6.1 ★	Phim Kinh Dị, Phim Gây Cấn, Phim Bi Án,
	Sex Tape	5.4 ★	Phim Hài,
	Sexo con amor	6.5 ★	Phim Hài, Phim Lãng Mạn,
	69	0 ★	
	22 vs. Trái Đất	7.2 ★	Phim Gia Đình, Phim Hài, Phim Giả Tượng, Phim Hoạt Hình,
	13 Minutes	0 ★	Phim Chính Kịch, Phim Gây Cấn,
	Spencer	0 ★	Phim Chính Kịch, Phim Lịch Sử,
	Shang-Chi và Huyền Thoại Thập Luân	7.8 ★	Phim Hành Động, Phim Phiêu Lưu, Phim Giả Tượng,
	Hỗn Số Tử Thần	6.1 ★	Phim Hành Động, Phim Phiêu Lưu, Phim Chính Kịch, Phim Bi Án, Phim Khoa Học Viễn Tưởng,

Hình 2.2.6.1 Xem danh sách phim đã xem

2.2.6.2. Phim đã bình luận

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

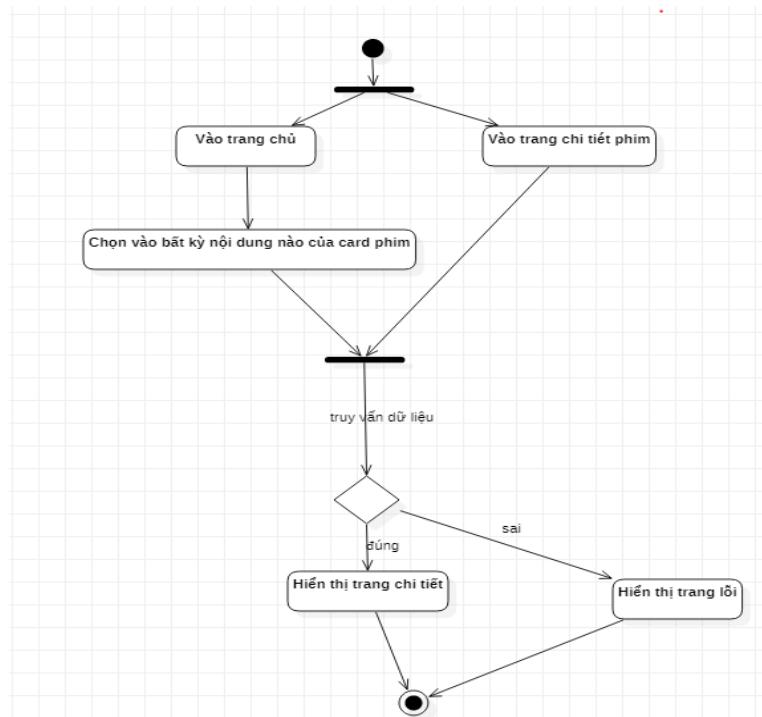
GVHD: TS. Huỳnh Xuân Phụng



Hình 2.2.6.2 Xem danh sách phim đã bình luận

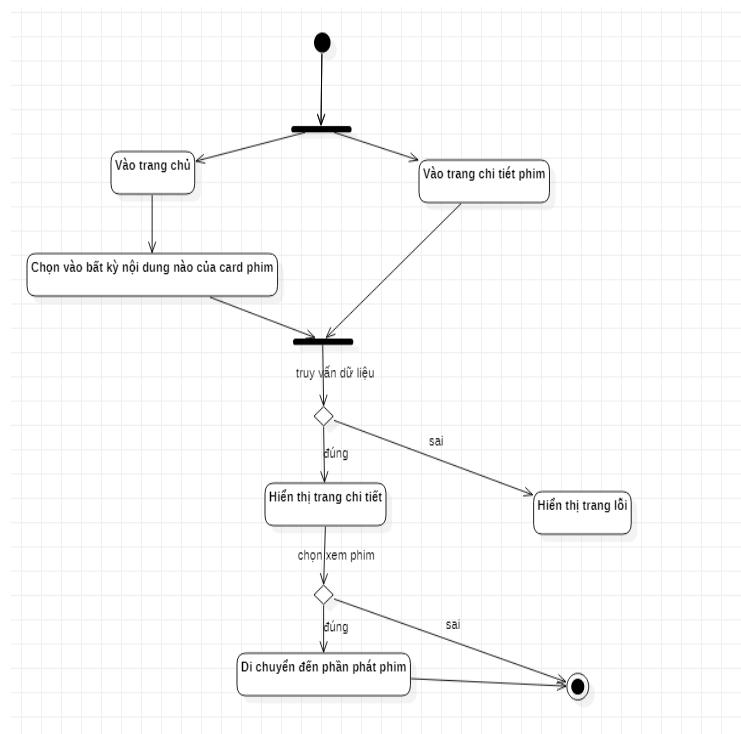
2.3. ACTIVITIES DIAGRAM

2.3.1. Xem chi tiết phim



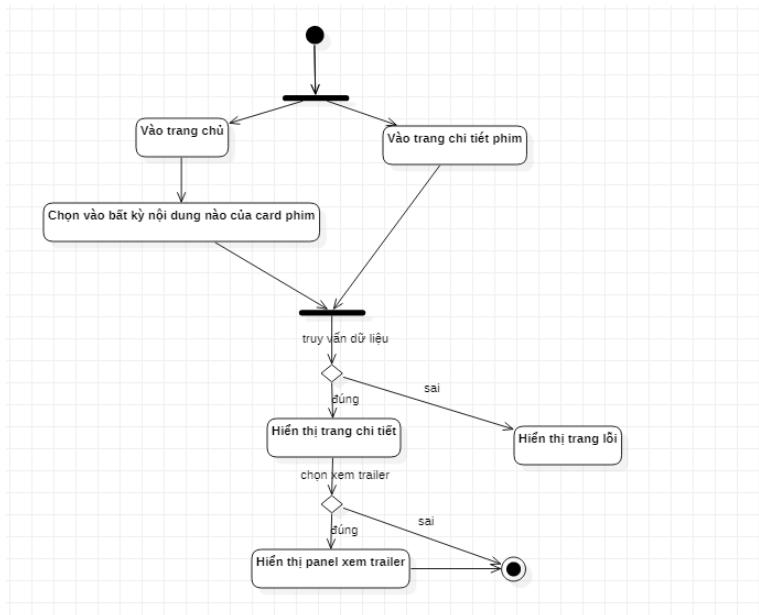
Hình 2.3.1.1: Sơ đồ hoạt động xem chi tiết phim

2.3.2. Xem phim



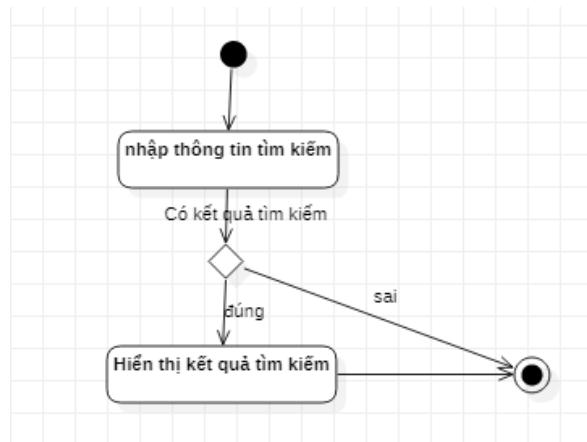
Hình 2.3.2.1: Sơ đồ hoạt động xem phim

2.3.3. Xem trailer



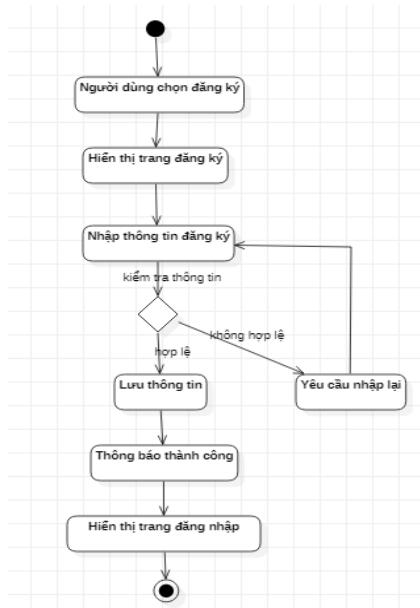
Hình 2.3.3.1: Sơ đồ hoạt động xem trailer

2.3.4. Tìm kiếm



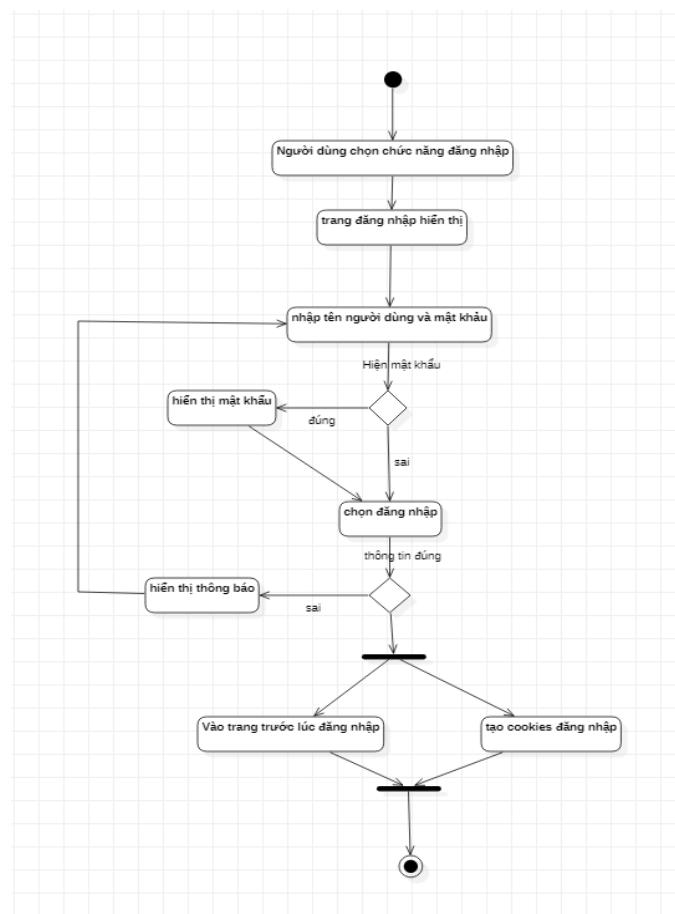
Hình 2.3.4.1: Sơ đồ hoạt động tìm kiếm

2.3.5. Đăng ký



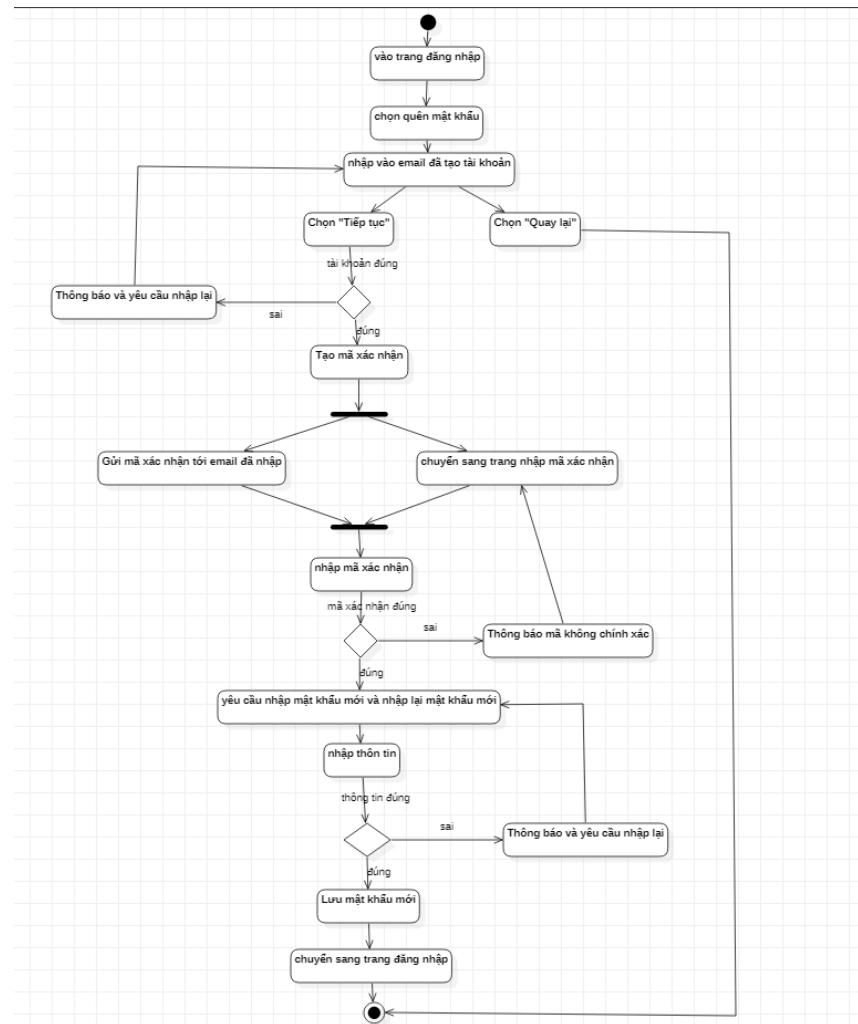
Hình 2.3.5.1: Sơ đồ hoạt động đăng ký

2.3.6. Đăng nhập



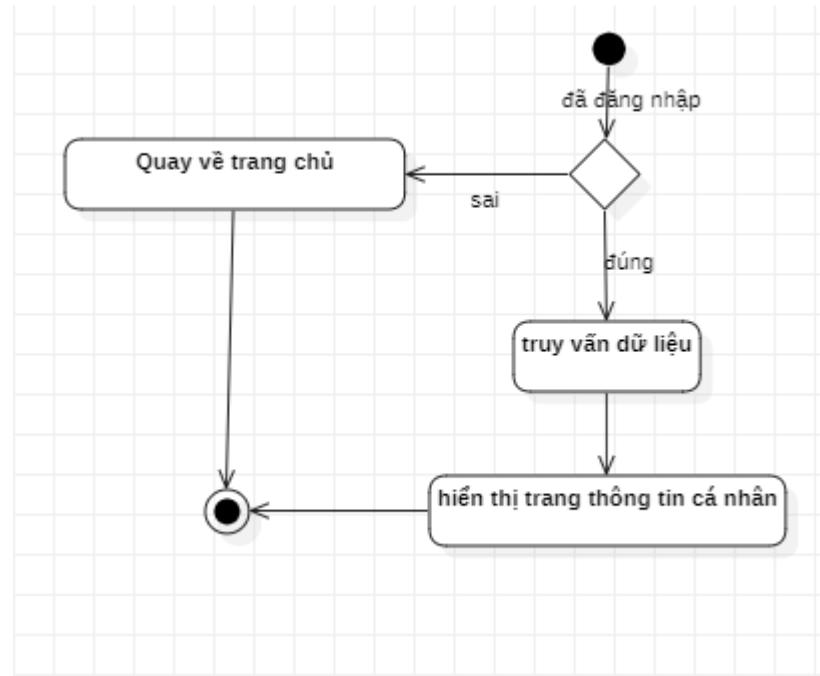
Hình 2.3.6.1: Sơ đồ hoạt động đăng nhập

2.3.7. Quên mật khẩu



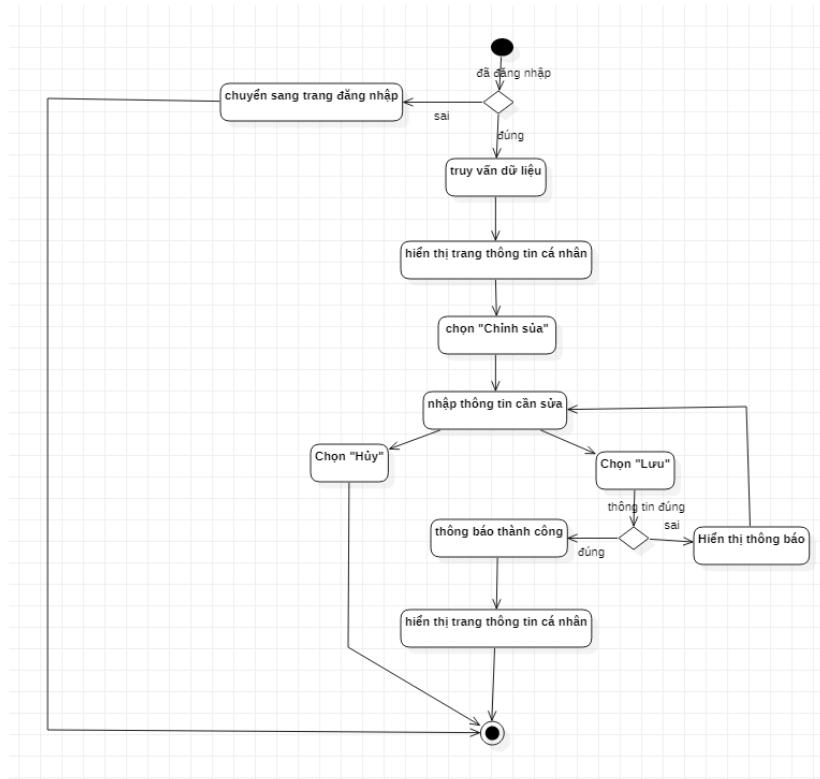
Hình 2.3.7.1: Sơ đồ hoạt động quên mật khẩu

2.3.8. Xem thông tin cá nhân



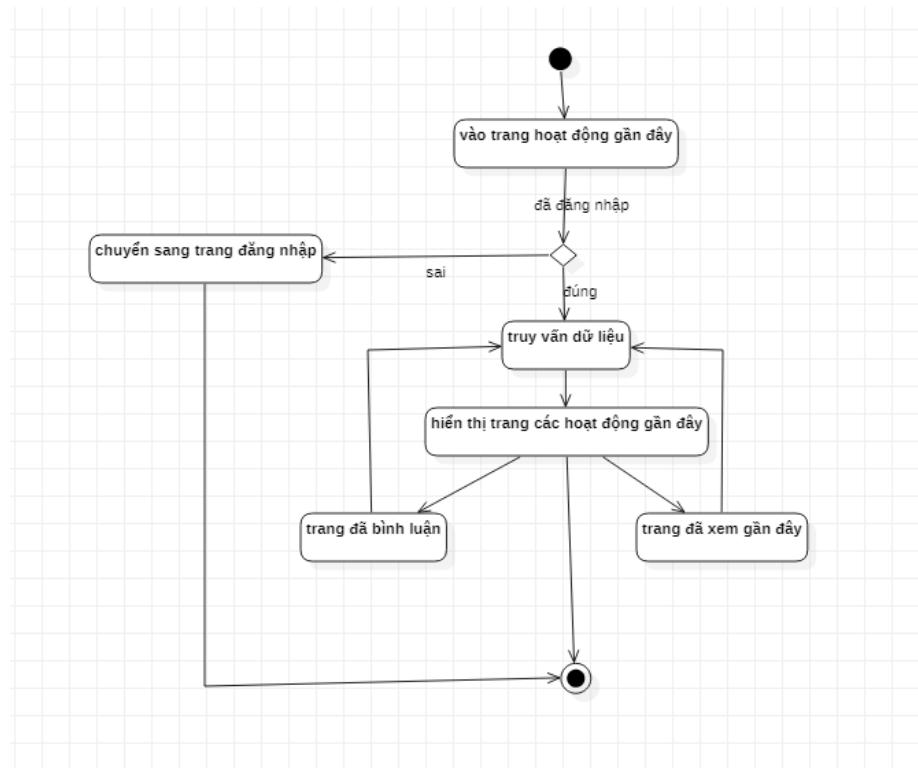
Hình 2.3.8.1: Sơ đồ hoạt động xem thông tin cá nhân

2.3.9. Sửa thông tin cá nhân



Hình 2.3.9.1: Sơ đồ hoạt động sửa thông tin cá nhân

2.3.10. Xem các hoạt động gần đây



Hình 2.3.10.1: Sơ đồ hoạt động xem các hoạt động gần đây

2.3.11. Bình luận

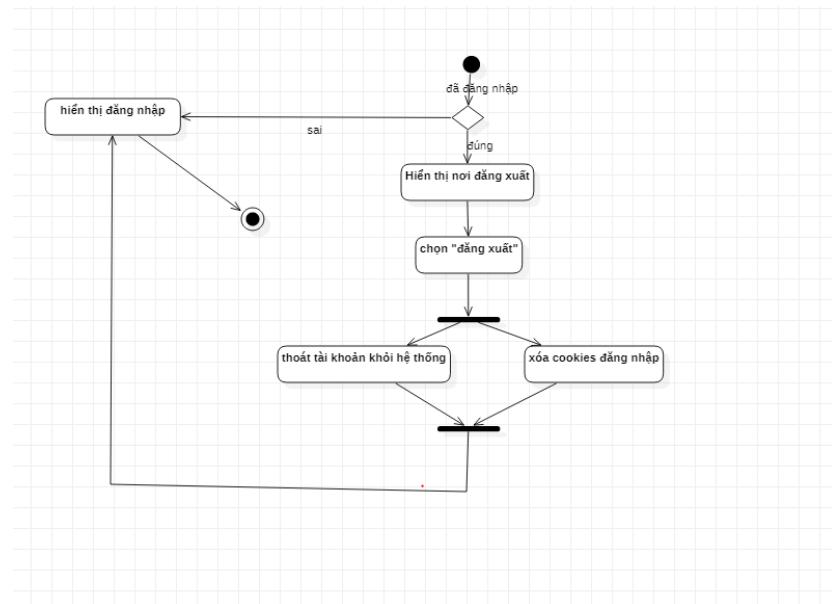


XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

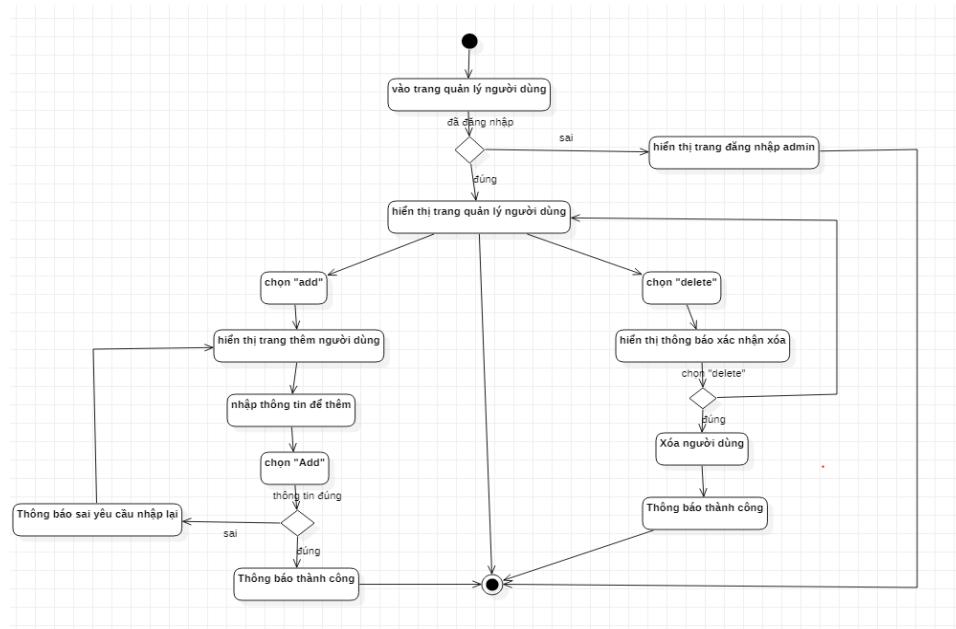
Hình 2.3.11.1: Sơ đồ hoạt động bình luận

2.3.12. Đăng xuất



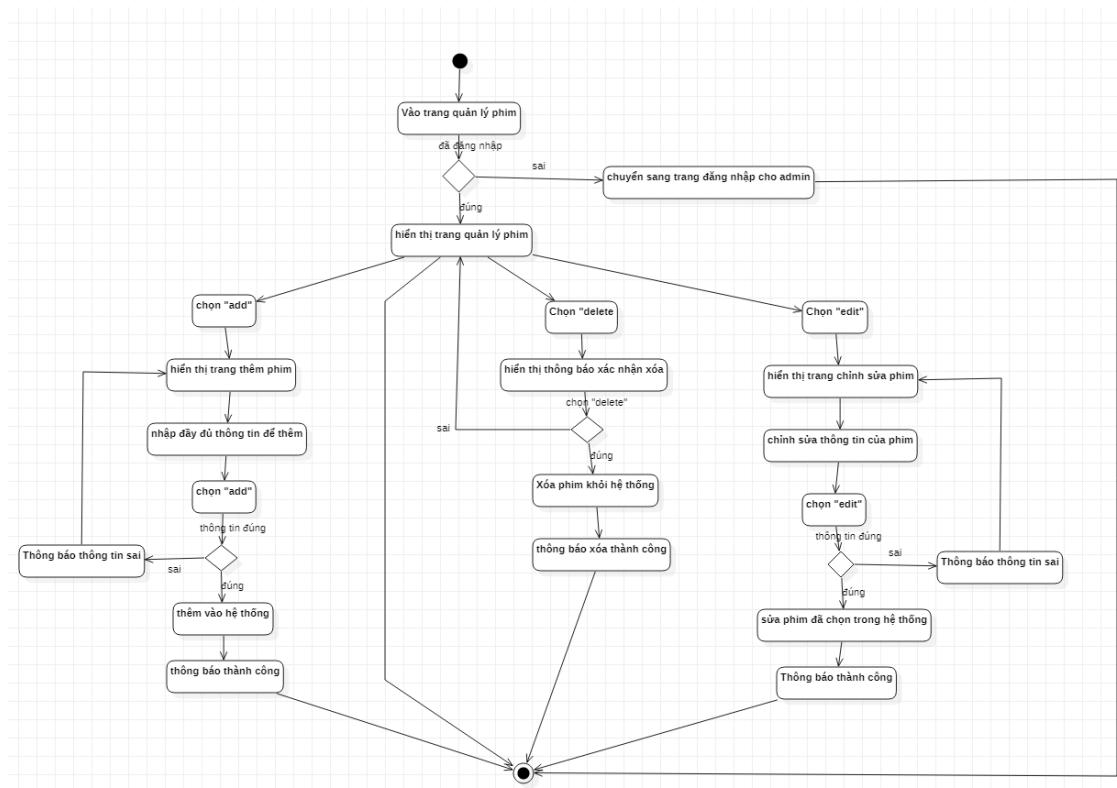
Hình 2.3.12.1: Sơ đồ hoạt động đăng xuất

2.3.13. Quản lý người dùng



Hình 2.3.13.1: Sơ đồ hoạt động quản lý người dùng

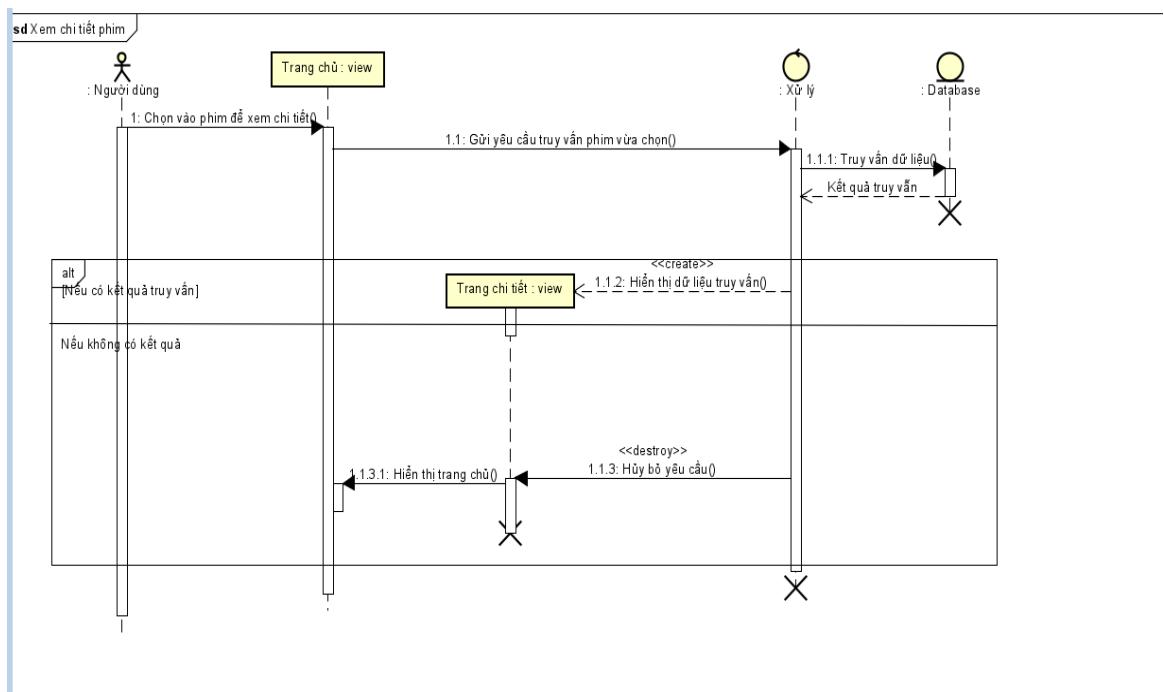
2.3.14. Quản lý phim



Hình 2.3.14.1: Sơ đồ hoạt động quản lý phim

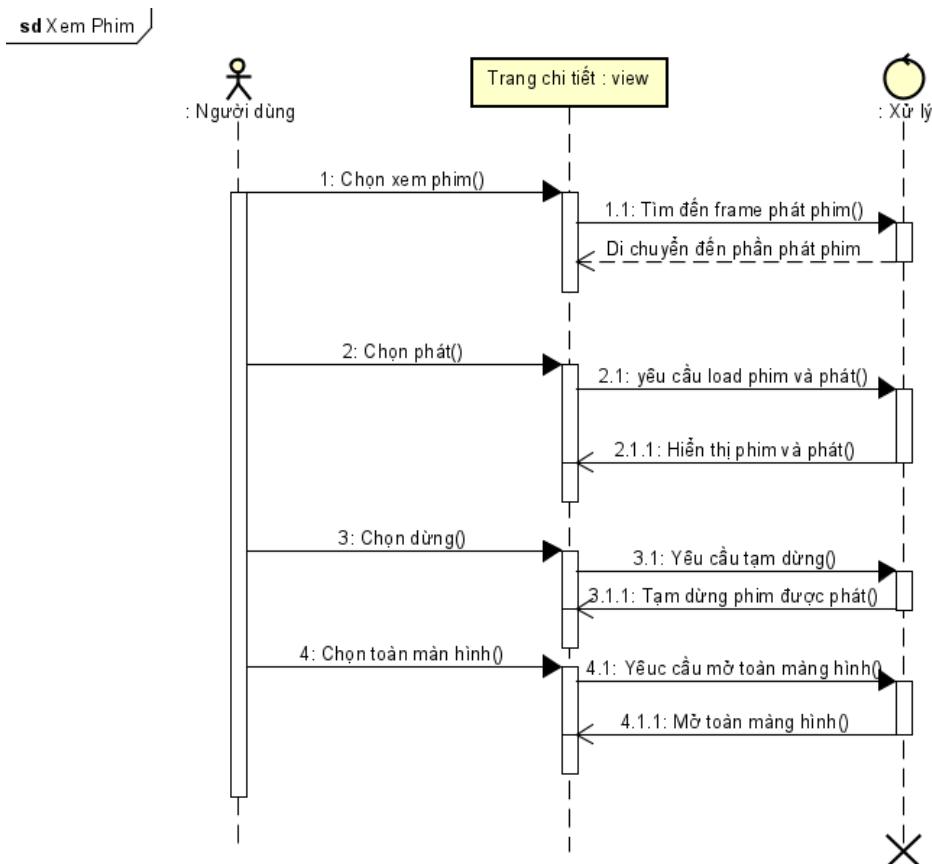
2.4. SEQUENCE DIAGRAM

2.4.1. Xem chi tiết phim



Hình 2.4.1.1: Lược đồ tuần tự xem chi tiết phim

2.4.2. Xem phim

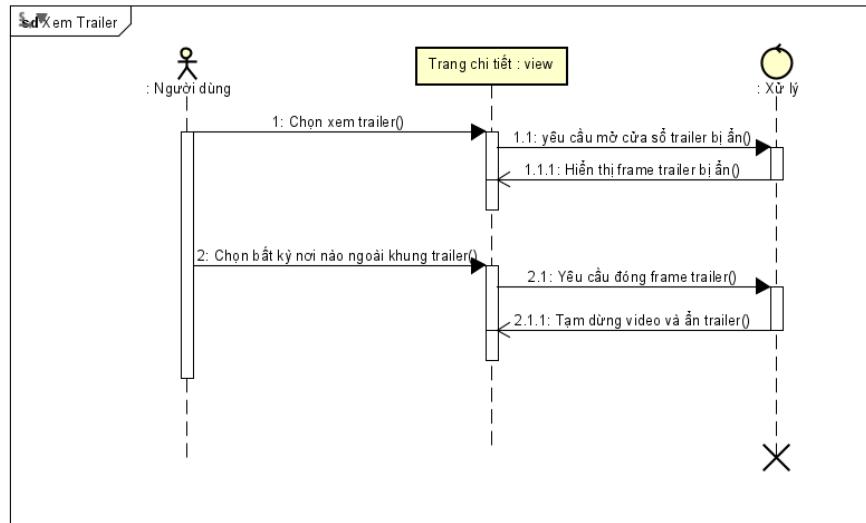


XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

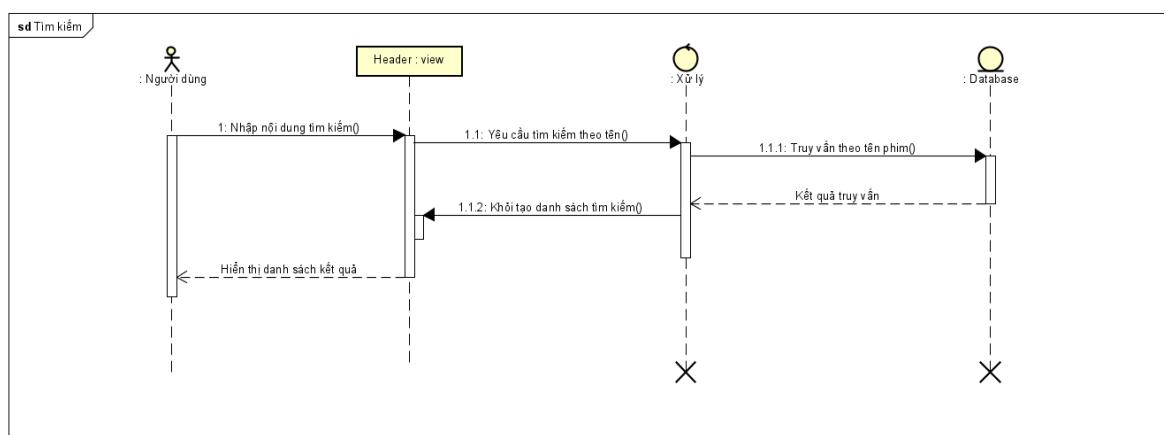
Hình 2.4.2.2: Lược đồ tuần tự xem phim

2.4.3. Xem trailer



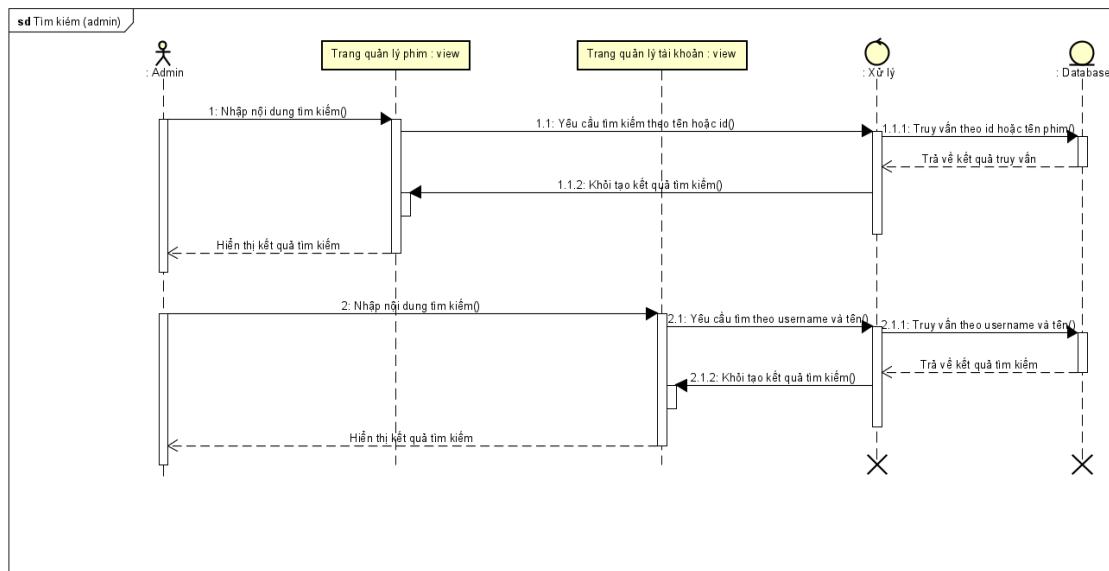
Hình 2.4.3.1: Lược đồ tuần tự xem trailer

2.4.4. Tìm kiếm (người dùng)



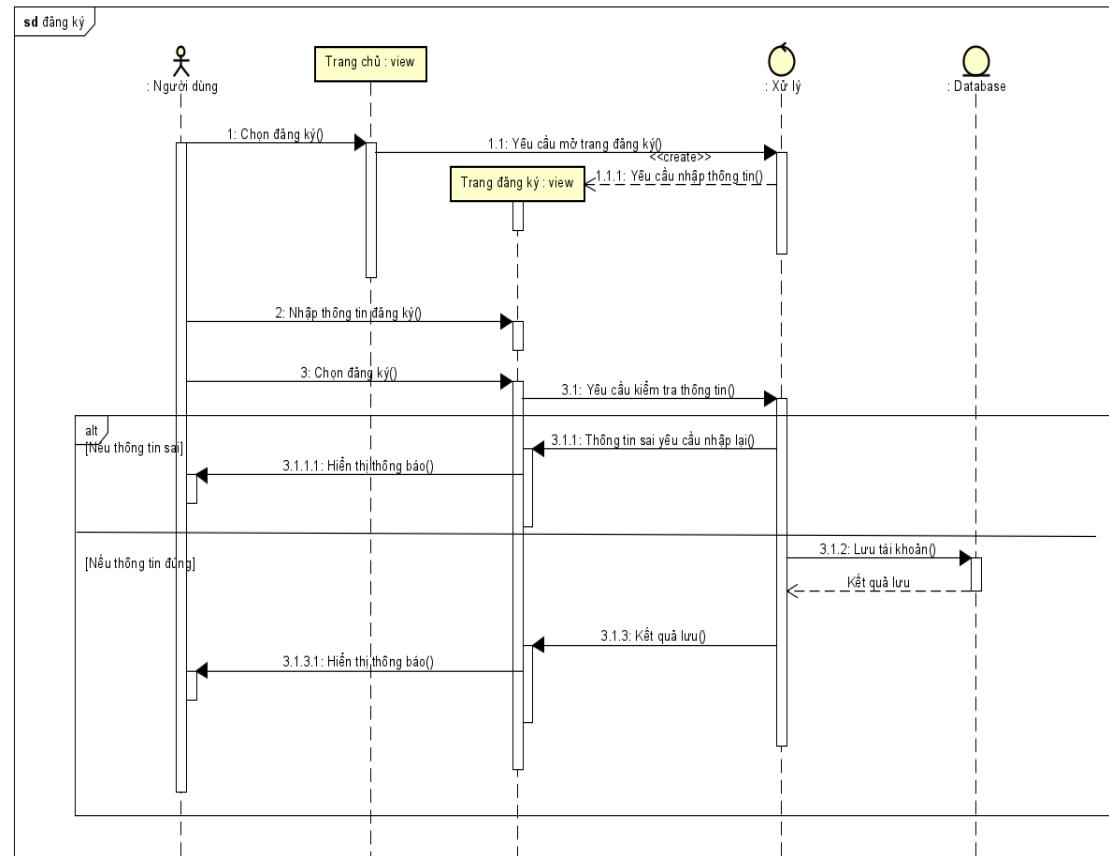
Hình 2.4.4.1: Lược đồ tuần tự tìm kiếm cho người dùng

2.4.5. Tìm kiếm (admin)



Hình 2.4.5.1: Lược đồ tuần tự tìm kiếm cho admin

2.4.6. Đăng ký

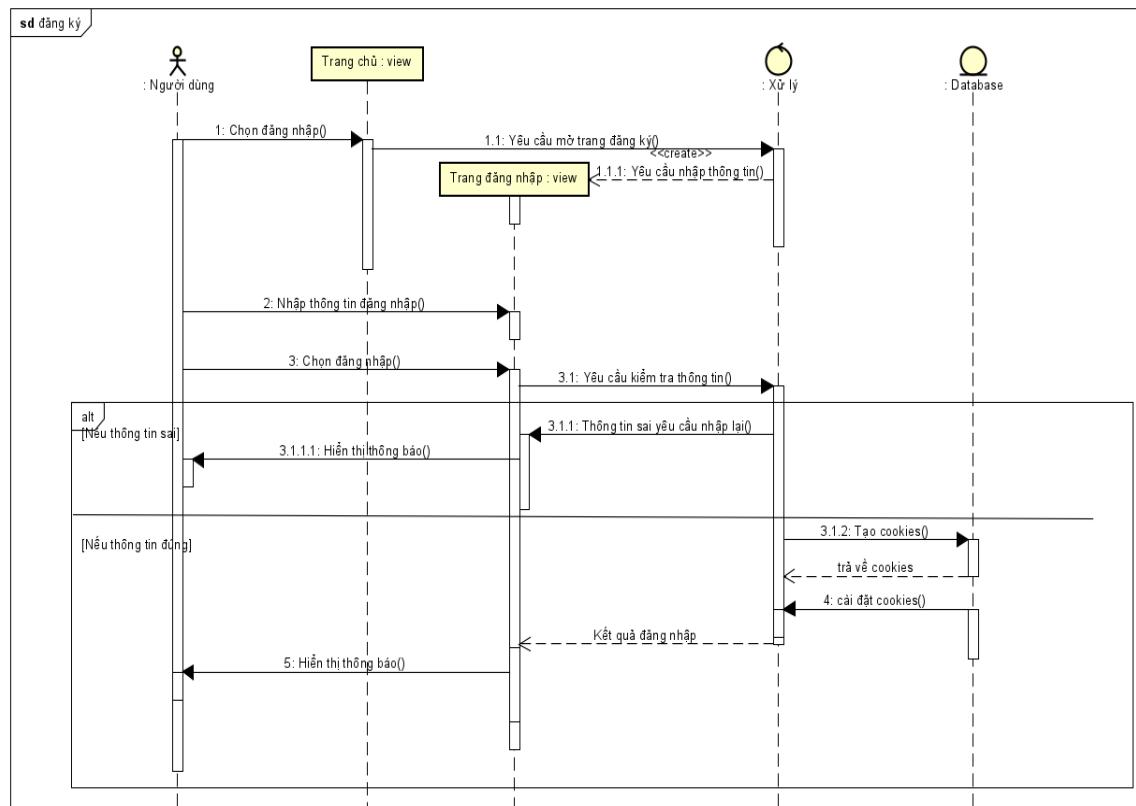


Hình 2.4.6.1: Lược đồ tuần tự đăng ký

2.4.7. Đăng nhập

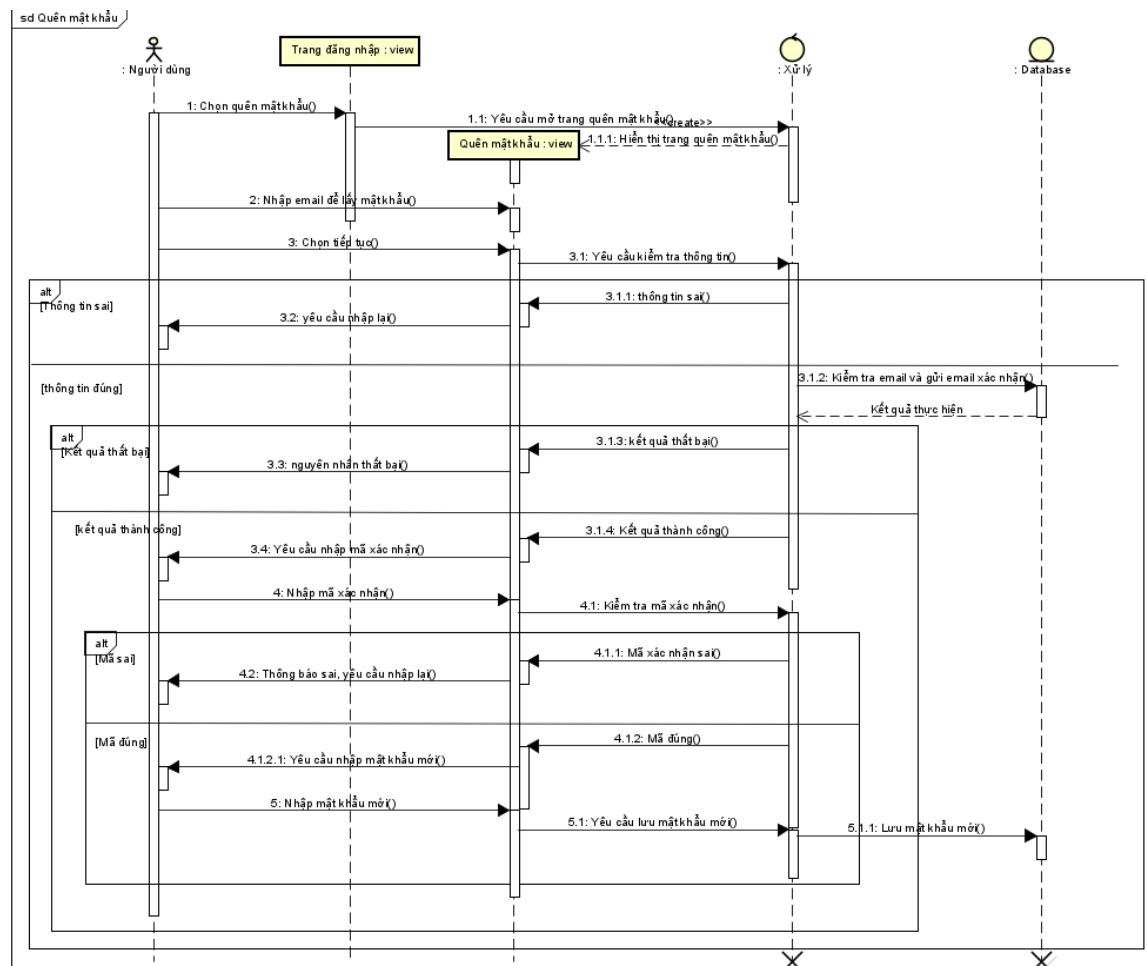
XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng



Hình 2.4.7.1: Lược đồ tuần tự đăng nhập

2.4.8. Quên mật khẩu

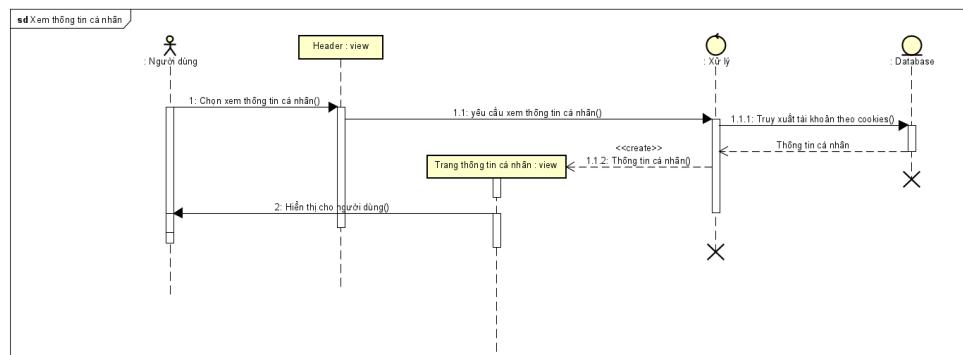


XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

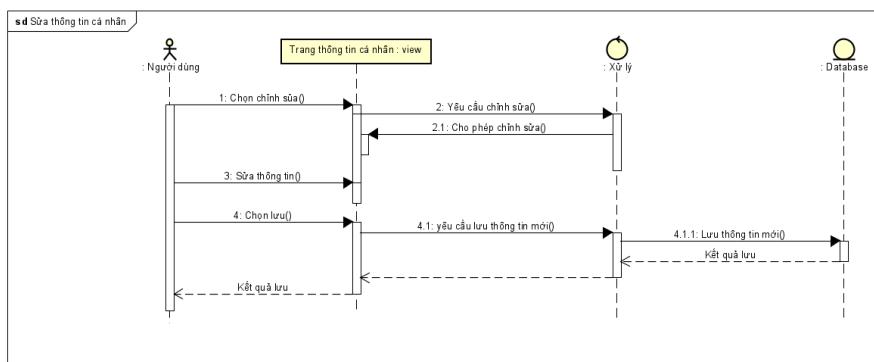
Hình 2.4.8.1: Lược đồ tuần tự quên mật khẩu

2.4.9. Xem thông tin cá nhân



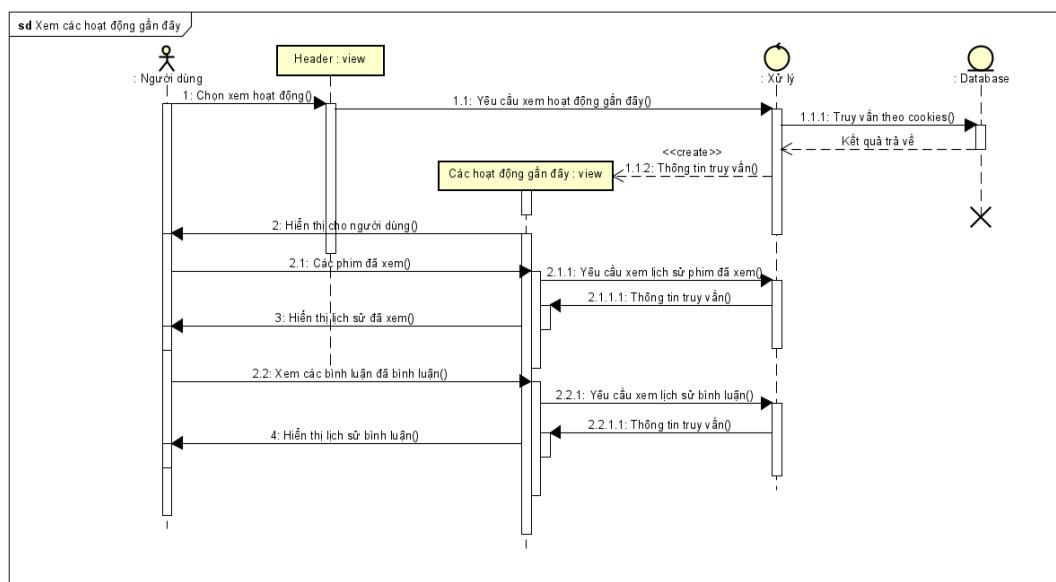
Hình 2.4.9.1: Lược đồ tuần tự xem thông tin cá nhân

2.4.10.Sửa thông tin cá nhân



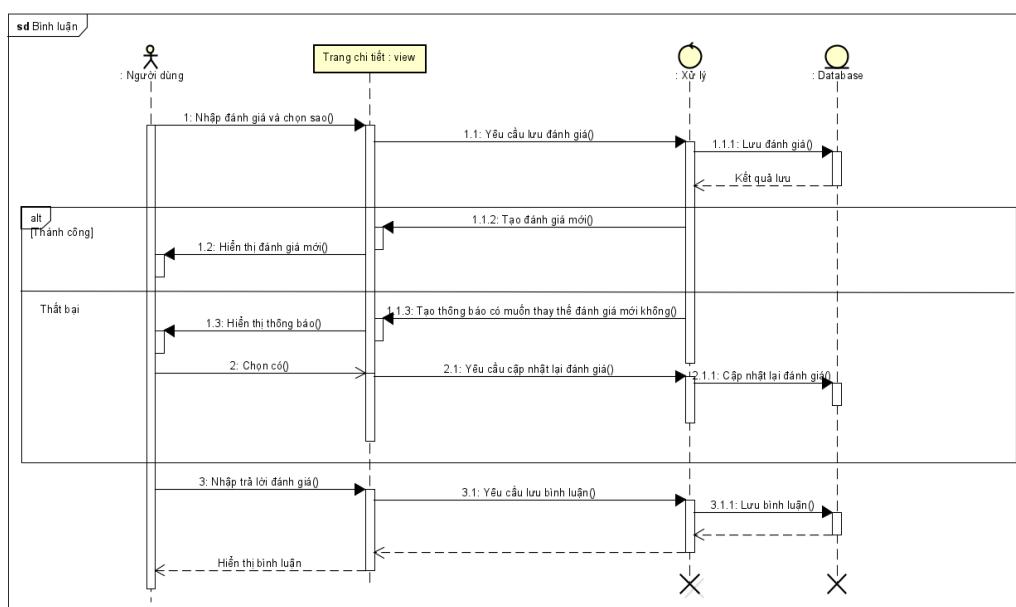
Hình 2.4.10.1: Lược đồ tuần tự sửa thông tin cá nhân

2.4.11.Xem các hoạt động gần đây



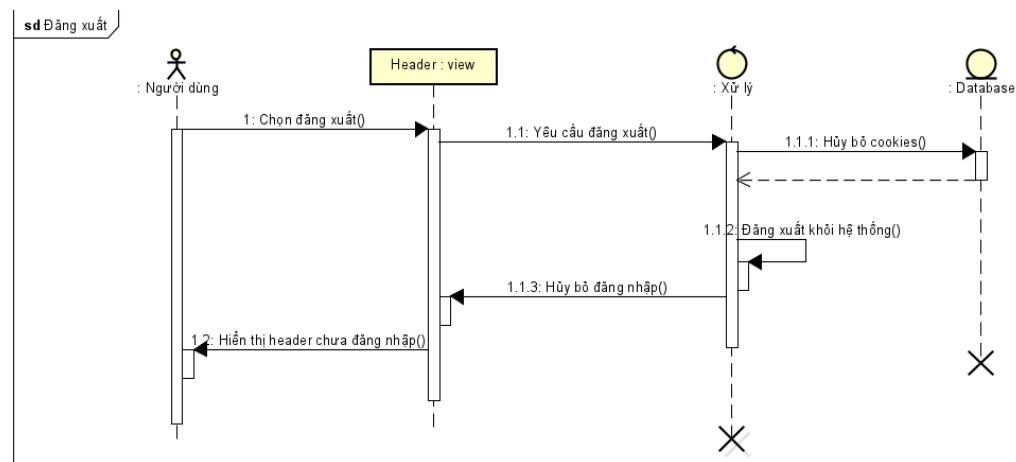
Hình 2.4.11.1: Lược đồ tuần tự xem các hoạt động gần đây

2.4.12.Bình luận



Hình 2.4.7.1: Lược đồ tuần tự bình luận

2.4.13. Đăng xuất



Hình 2.4.7.1: Lược đồ tuần tự đăng xuất

2.4.14. Quản lý người dùng

Chương 3: THUẬT TOÁN DỰ ĐOÁN PHIM LIÊN QUAN

3.1. TÌM HIỂU CÔNG NGHỆ VÀ THƯ VIỆN

3.1.1. Công nghệ sử dụng

- Python là một ngôn ngữ bậc cao cho các mục đích lập trình đa tính năng đặc biệt là trong trí tuệ nhân tạo. Với hình thức sáng sủa, tốc độ xử lý nhanh, hệ thống thư viện lớn và mạnh thuận tiện cho việc lập trình theo hướng trí tuệ nhân tạo.
- Với Python chúng ta sẽ dễ dàng tiếp cận trí tuệ nhân tạo hơn bởi vì không cần hiểu quá sâu để viết ra thuật toán, với python chúng ta có những thư viện chuyên dụng giúp đỡ về điều đó, ở đây chúng ta chỉ cần điều chỉnh những thông số cho phù hợp. Và đây cũng là lý do nhóm chúng em chọn python là công nghệ sử dụng cho phần dự đoán phim liên quan.

3.2. THƯ VIỆN SỬ DỤNG

3.2.1. CountVectorizer

- CountVectorizer là một class của thư viện Sklearn nó giúp chuyển đổi văn bản đầu vào thành một ma trận số của từ khóa.

3.2.2. Cosine_similarity

- Tính độ tương tự cosin giữa 2 ma trận dưới dạng tích số chấm chuẩn hóa của 2 ma trận đó. Tức là, tính góc cosin giữa 2 danh sách vector được vẽ bằng 2 ma trận đầu vào.
- Ví dụ: 2 vector trùng nhau và cùng hướng thì góc hợp bởi chúng là 0 và cosin giữa chúng đạt giá trị lớn nhất là 1. Ngược lại, nếu chúng ngược chiều nhau thì góc hợp bởi chúng là 180 tất nhiên điều này sẽ làm cosin giữa chúng nhỏ nhất là -1.
- Như vậy, chúng ta có thể dễ dàng trả lời câu hỏi “làm thế nào để biết 2 vector tương đồng nhau nhất?” đó chính là giá trị cosin của chúng lớn nhất.

3.2.3. PorterStemmer

- PorterStemmer là một class của thư viện nltk nó giúp loại bỏ các yếu tố phụ của một từ. Ví dụ như Ran sẽ được chuyển thành run, hoặc stating sẽ được chuyển thành state.

3.2.4. Pandas

- Về python khi nói đến xử lý mảng thì không thể không nhắc đến pandas đây là một thư viện phân tích và thao tác dữ liệu nhanh, mạnh, linh hoạt và dễ dàng sử dụng.

3.2.5. Flask

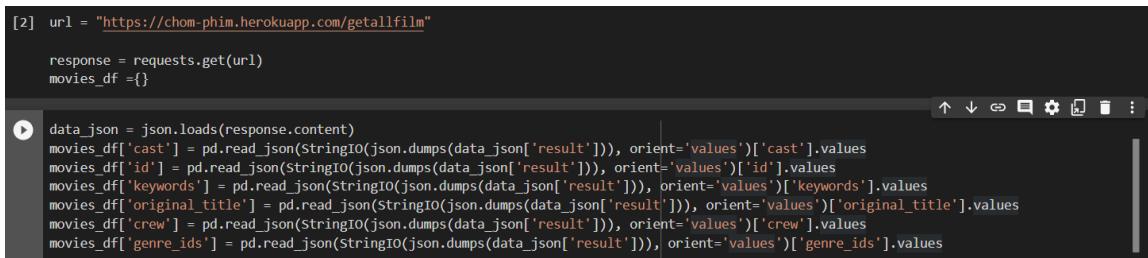
- Flask là một micro framework. Micro không phải là thiếu chức năng, mà micro ở đây là xúc tích, gói gọn. Flask cung cấp cho các lập trình viên khả năng tùy biến khi phát triển ứng dụng web, nó cung cấp cho bạn các công cụ, thư viện và cơ chế cho phép bạn xây dựng một ứng dụng web nhưng nó sẽ không thực thi bất kỳ sự phụ thuộc nào hoặc cho bạn biết dự án sẽ như thế nào.
- Ứng dụng web có thể là blog, trang web thương mại hoặc một số trang web khác, nó vẫn cho phép các lập trình viên cơ hội sử dụng một số tiện ích mở rộng để thêm nhiều chức năng hơn cho ứng dụng web.

3.3. ÁP DỤNG VÀ THUẬT TOÁN XỬ LÝ

- Thuật toán được viết dựa trên “Build a Movie Recommendation System in Python using Machine Learning”^[2]

3.3.1. Xử lý dữ liệu

- Chúng ta cần lấy dữ liệu từ API sau đó lưu những dữ liệu cần thiết như diễn viên, id, từ khóa, tên tiếng anh của phim, ... vào dict của python. Ở đây chúng ta dùng dict để có thể dễ dàng truy xuất dữ liệu hơn vì dict có dạng “key” và “value” với mỗi “key” sẽ ứng với “value” mà chúng đã đặt cho nó.

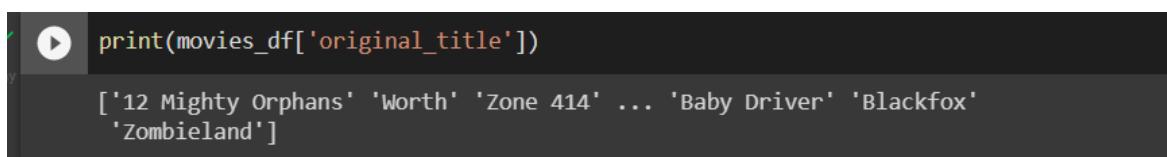


```
[2] url = "https://chom-phim.herokuapp.com/getallfilm"
response = requests.get(url)
movies_df = {}

data_json = json.loads(response.content)
movies_df['cast'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')['cast'].values
movies_df['id'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')[['id']].values
movies_df['keywords'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')['keywords'].values
movies_df['original_title'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')['original_title'].values
movies_df['crew'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')['crew'].values
movies_df['genre_ids'] = pd.read_json(StringIO(json.dumps(data_json['result'])), orient='values')['genre_ids'].values
```

Hình 3.3.1.1: Nhận dữ liệu từ API

- Sau khi thực hiện xong chúng ta sẽ xem thử dữ liệu hiện tại như thế nào:



```
print(movies_df['original_title'])

['12 Mighty Orphans' 'Worth' 'Zone 414' ... 'Baby Driver' 'Blackfox'
 'Zombieland']
```

Hình 3.3.1.2: Hiển thị mảng tên của dữ liệu nhận từ API

- Tuy đã lấy được dữ liệu nhưng hãy chú ý đến cast, crew, keywords và genre_ids chúng là một object gồm nhiều phần tử cho nên chúng ta cần lọc ra để lấy những dữ liệu cần thiết.
- ❖ Khởi tạo director để lưu đạo diễn của phim

```
[8] movies_df["director"] = []
```

Hình 3.3.1.3: Khởi tạo mảng để lưu đạo diễn

- ❖ Hàm để thực hiện trích xuất đạo diễn từ crew mà chúng ta đã nhận được

```
[9] def get_director(x):
    for i in x:
        team_dir = []
        for j in i:
            if j["job"] == "Director":
                team_dir.append(str.lower(j["name"]))
    movies_df['director'].append(team_dir)
```

Hình 3.3.1.4: Hàm trích xuất đạo diễn từ dữ liệu đã lưu

- ❖ Tiến hành trích xuất

```
[11] pd.DataFrame(movies_df["crew"]).apply(get_director)
```

Hình 3.3.1.5: Trích xuất đạo diễn từ dữ liệu đã lưu

- ❖ Bây giờ chúng ta sẽ xem dữ liệu hiện tại chúng ta nhận được như thế nào

```
▶ print(movies_df['director'])
[['ty roberts'], ['sara colangelo'], ['andrew baird'], ['daniel monzón'], ['cédric jimenez'], ['destin daniel crett']]
```

Hình 3.3.1.6: Hiển thị danh sách đạo diễn cho từng phim

- ❖ Vì mỗi bộ phim có thể có nhiều đạo diễn cho nên nó được đặt ở dạng mảng.
- ❖ Tương tự, chúng ta sẽ tiếp tục trích xuất cast, keywords, genre_ids. Nhưng vì chúng có cấu trúc dữ liệu chung nên chúng ta chỉ dùng một hàm để trích xuất.

```
[10] def get_list(x, column):
    movies_df[column] = []
    for i in x[0]:
        temp = []
        if(i != 0):
            for j in i:
                temp.append(str.lower(j["name"]))
    movies_df[column].append(temp)
```

Hình 3.3.1.7: Hàm trích xuất dữ liệu

```
▶ get_list(pd.DataFrame(movies_df["cast"]), column='cast')
get_list(pd.DataFrame(movies_df["keywords"]), column='keywords')
get_list(pd.DataFrame(movies_df["genre_ids"]), column='genre_ids')
```

Hình 3.3.1.8: Trích xuất dữ liệu

XÂY DỰNG TRANG WEB ĐỀ XUẤT PHIM GVHD: TS. Huỳnh Xuân Phụng

GVHD: TS. Huỳnh Xuân Phụng

- Sau khi đã lọc dữ liệu thành công chúng ta sẽ đến với bước tiếp theo là tổng hợp những dữ liệu của một bộ phim lại thành một phần tử của mảng.
 - ❖ Tạo mảng chứa và phần tử cần loại bỏ ở đây chúng ta bỏ từ “phim”.

```
 movies_df['soup'] = []  
stopwords=["phim"]
```

Hình 3.3.1.9: Khởi tạo mảng chứa

- ❖ Tiến hành tổng hợp dữ liệu.

```
[1] def create_soup(x):
    stop = (int)(x.size /len(x))
    for i in range(stop):
        movies_df["soup"].append(' '.join(x[i]['keywords']) + ' ' + ' '.join(x[i]['cast']) + ' ' + ' '.join(x[i]['director']) + ' ' + ' '.join(x[i]['plot']) + ' ' + ' '.join(x[i]['genres']))
```

Hình 3.3.1.10: Hàm tổng hợp dữ liệu

```
[44] create_soup(pd.DataFrame.from_dict(movies_df, orient = 'index'))
```

Hình 3.3.1.11: Tổng hợp dữ liệu

- ❖ Xem kết quả chúng ta thu được như thế nào

```
print(movies_df['soup'])
```

['american football sports coach based on a true story orphan 1930s luke wilson vinessa shaw wayne knight martin sheen robert duvall ja

Hình 3.3.1.12: Kết quả tổng hợp dữ liệu

- ❖ Chúng ta hãy chú ý tất cả dữ liệu này đang là một câu và các từ của nó chưa ở nguyên mẫu nên khi dự đoán nhất định sẽ ảnh hưởng lớn kết quả thu được. Cho nên việc tiếp theo chúng ta cần làm đó chính là tách những câu này ra thành từng từ, đưa nó về nguyên mẫu và loại bỏ những từ không liên quan.

```
ps = PorterStemmer()
movies_df['total'] = []
for words in movies_df['soup']:
    movies_df['total'].append([ps.stem(word) for word in words.split() if not word in set(stopwords)])
```

Hình 3.3.1.13: Tách từ và đưa về nguyên mẫu

- ❖ Khi thực hiện thành công chúng ta hãy xem lại kết quả một lần nữa

```
print(movies_df['total'])  
[['american', 'footbal', 'sport', 'coach', 'base', 'on', 'a', 'true', 'stori', 'orphan', '1930', 'luke',  
4
```

Hình 3.3.1.14: Kết quả tách từ và đưa về nguyên mẫu

3.3.2. Dự đoán

- Tổng hợp các từ lại thành một chuỗi các từ vì thuộc tính fit_transform của class CountVectorizer là chỉ nhận đầu vào là chuỗi mà hiện tại với mỗi id phim chúng ta

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM GVHD: TS. Huỳnh Xuân Phụng

có bộ tổng hợp các từ nằm trong mảng. Cho nên chúng ta sẽ nối những từ đơn lẻ này thành chuỗi các từ.

```
[63] movies_df['train'] = []
    for words in movies_df['total']:
        temp = ""
        for word in words:
            temp += word + " "
        movies_df['train'].append(temp)
```

Hình 3.3.2.1: Nối từ đơn lẻ trong mảng thành chuỗi từ

- Dữ liệu cuối cùng chúng ta có được là như sau:

```
print(movies_df['train'])
['american football sport coach base on a true story orphan 1930 luke wilson vinessa shaw wayn knight martin sheen rober
```

Hình 3.3.2.2: Kết quả nối từ

- Điều tiếp theo chúng ta cần làm đó là đưa bộ dữ liệu này chuyển sang dạng số để có thể vẽ vector của nó.

```
▶ count_vectorizer = CountVectorizer(stop_words="english")
[67] count_matrix = count_vectorizer.fit_transform(movies_df["train"])
```

Hình 3.3.2.3: Vẽ vector với tập dữ liệu vừa nối

- Dữ liệu đầu ra sẽ có dạng như sau (X, Y) A, Với X là index của liệu mà ta đã đưa vào trước đó, Y là giá trị của một từ được tách ra từ chuỗi, A là số lần lặp lại của nó trong câu đó.

```
[68] print(count_matrix)
[(0, 991), 1
 (0, 10359), 1
 (0, 28961), 1
 (0, 6160), 1
 (0, 2558), 1
 (0, 31258), 1
 (0, 29293), 1
 (0, 23083), 1
 (0, 17), 1]
```

Hình 3.3.2.4: Kết quả sau khi vẽ vector

- Tiếp theo chúng ta sẽ tính vector tương đồng của chúng

```
cosine_sim2 = cosine_similarity(count_matrix, count_matrix)
```

XÂY DỰNG TRANG WEB ĐỀ XUẤT PHIM GVHD: TS. Huỳnh Xuân Phung

Hình 3.3.2.5: Tính vector tương đồng

- Chuyển sang ma trận ngang để lấy được index của key

```
[103] movies_df = pd.DataFrame.from_dict(movies_df, orient = 'index').reset_index()  
  
print(movies_df)  
  
          index ... 1973  
0      cast ... [woody harrelson, jesse eisenberg, emma stone,...  
1        id ... 19908  
2   keywords ... [circus, sibling relationship, washington dc, ...  
3 original_title ... Zombieland  
4      crew ... [{adult': False, 'gender': 2, 'id': 561, 'kno...  
5  genre_ids ... [phim hài, phim kinh dị]  
6    director ... [ruben fleischer]  
7       soup ... circus sibling relationship washington dc, usa...  
8      total ... [circu, sibl, relationship, washington, dc,, u...  
9       train ... circu sibl relationship washington dc, usa pos...  
  
[10 rows x 1975 columns]
```

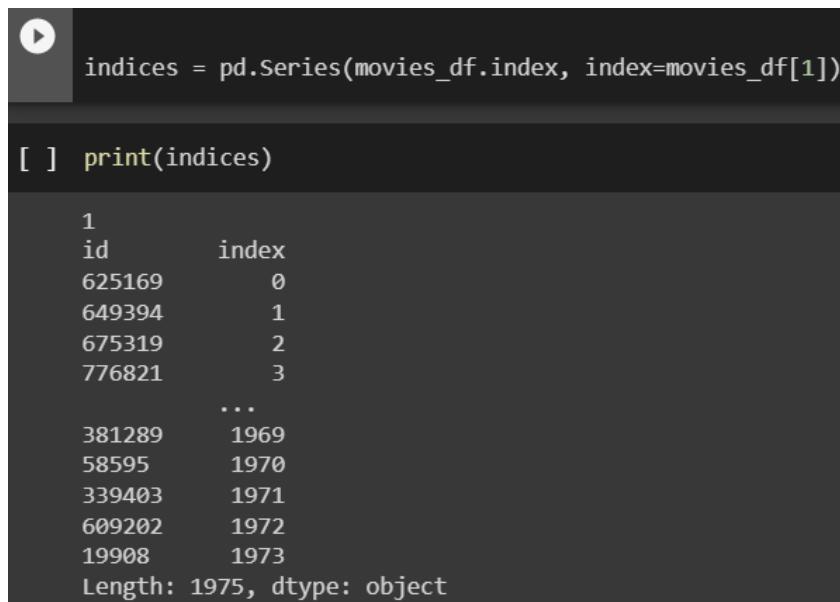
Hình 3.3.2.6: Chuyển sang ma trận ngang và kết quả

- Chuyển sang ma trận doc để dễ dàng truy xuất phần tử cũng như là tham chiếu với phần dự đoán của CountVectorizer

```
[34] movies_df = movies_df.transpose()  
  
[35] print(movies_df)  
  
          0 ... 9  
index                          cast ... train  
0    [luke wilson, vinessa shaw, wayne knight, mart... ... american football sport coach base on a true st...  
1    [michael keaton, amy ryan, stanley tucci, tate... ... base on a true story 9/11 world trade center m...  
2    [guy pearce, matilda anna ingrid lutz, travis ... ... android beat cop synth guy pearce matilda anna ...  
3    [marcos ruiz, begoña vargas, chechu salgado, p... ... robbery base on novel or book 1970 cine quinqu...  
...      ... ... ... ...  
1969  [josh gad, k.j. apa, britt robertson, dennis q... ... base on novel or book human animal relationship ...  
1970  [kristen stewart, chris hemsworth, Charlize Th... ... queen magic imprisonment fairytale immortal forest ...  
1971  [ansel elgort, kevin spacey, lily james, jon h... ... robbery waitress atlanta getaway car crime bosses ...  
1972  [ayaka nanase, haruka tomatsu, you taichi, hir... ... fight ninja school anime futurist city ayaka na...  
1973  [woody harrelson, jesse eisenberg, emma stone,... ... circus sibling relationship washington dc, usa pos...  
  
[1975 rows x 10 columns]
```

Hình 3.3.2.7: Chuyển thành ma trận đọc và kết quả

- Trích xuất lấy index và cột id của movies_df.



```

indices = pd.Series(movies_df.index, index=movies_df[1])

[ ] print(indices)

1
id      index
625169      0
649394      1
675319      2
776821      3
...
381289    1969
58595     1970
339403    1971
609202    1972
19908     1973
Length: 1975, dtype: object

```

Hình 3.3.2.8: Trích xuất dữ liệu cần thiết

- Viết hàm dự đoán. Ở đây chúng ta sẽ trích xuất 20 giá trị từ 1-21 là những giá trị cao nhất được tính từ thuật toán Cosine_similarity chúng ta sẽ bỏ phần tử 0 vì chúng nhất định bằng 1 do nó sẽ trùng với chính nó và đạt giá trị lớn nhất.

```

def get_recommendations(title, cosine_sim):
    """
    in this function,
        we take the cosine score of given movie
        sort them based on cosine score (movie_id, cosine_score)
        take the next 10 values because the first entry is itself
        get those movie indices
        map those indices to titles
        return title list
    """

    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:21]
    # (a, b) where a is id of movie, b is sim_score

    movies_indices = [ind[0] for ind in sim_scores]
    movies = movies_df[1].iloc[movies_indices]
    return movies

```

Hình 3.3.2.9: Hàm dự đoán theo id phim

- Và đây là kết quả chúng ta thu được

```
[137] get_recommendations(666, cosine_sim2)

 1026    744275
 205     12477
 620      348
 898    669363
 737    786302
 281    614587
 1466   846964
 1052   160885
 114    423108
 11      672741
 727    865953
 442    22832
 1470   705991
 1631   23823
 351    459073
 949    876603
 1468   883338
 1031   424694
 867    591275
 1089   724885
Name: 1, dtype: object
```

Hình 3.3.2.10: Dự đoán và kết quả thu được

3.3.3. Kết luận

Chúng em đã hoàn thành được hệ thống đề xuất phim dựa trên đầu vào là id phim và đầu ra là id của 20 phim.

Chương 4: THUẬT TOÁN DỰ ĐOÁN PHIM CHO TÙNG NGƯỜI DÙNG

4.1. TÌM HIỂU CÔNG NGHỆ VÀ THƯ VIỆN SỬ DỤNG

4.1.1. Công nghệ sử dụng

- Asp.NET Core là một nền tảng mã nguồn mở cho việc xây dựng và kết nối các công việc như xây dựng web, Internet of thing, web API

4.1.2. Thư viện sử dụng

- Machine Learning .NET là chiếc cầu nối giữa các thư viện python và .NET để .NET có thể truy cập đến các thư viện như Keras, CNTK, TensorFlow, ... của python.

4.2. ÁP DỤNG THUẬT TOÁN VÀ XỬ LÝ

- Thuật toán được viết dựa trên “Build a movie recommender using matrix factorization with ML.NET”^[3]

4.2.1. Xử lý dữ liệu

- Tạo hàm dựng và các phương thức cho model

```
public FilmsModel(int id, string username, string title, float evaluate, string poster_path, int run_time, string release_date)
{
    this.id = id;
    this.title = title;
    this.username = username;
    this.evaluate = evaluate;
    this.poster_path = poster_path;
    this.run_time = run_time;
    this.release_date = release_date;
}

10 references
public int id { get; set; }

6 references
public string title { get; set; }

6 references
public int run_time { get; set; }

6 references
public string release_date { get; set; }

7 references
public string username { get; set; }

5 references
public float evaluate { get; set; }
6 references
public string poster_path { get; set; }
```

Hình 4.2.1.1: Tạo model lưu trữ cho một phim

- Tạo model để lưu trữ tất cả các phim

```

2 references
public ListFilm(string json)
{
    films = new List<FilmsModel>();
    dynamic myObject = JValue.Parse(json);
    if (myObject.List.GetType().Name == "JArray")
    {
        foreach (dynamic film in myObject.List)
        {
            films.Add(new FilmsModel((int)film.id_film, (string)film.id_info, (string)film.title, (string)film.evaluate, (string)film.poster_path, (int)film.run_time, (string)film.release_date));
        }
    }
}

5 references
public List<FilmsModel> films { get; set; }

```

Hình 4.2.1.2: Tạo model lưu trữ cho danh sách phim

- Lấy dữ liệu từ API sau đó lưu vào model đã tạo

```

mlContext = new MLContext();
client = factory.CreateClient();
fmd = new FilmsModel[1974];

client.BaseAddress = new Uri("https://chom-phim.herokuapp.com");
var response = client.GetAsync("/2000comments").Result;
string jsonData = response.Content.ReadAsStringAsync().Result;

ListFilm films = new ListFilm(jsonData);

int i = -1;
foreach (FilmsModel item in films.films)
{
    i += 1;
    fmd[i] = new FilmsModel(item.id, item.username, item.title, item.evaluate, item.poster_path, item.run_time, item.release_date);
}

```

Hình 4.2.1.3: Nhận dữ liệu từ API

- Tạo dữ liệu cho phần đào tạo model ở đây chúng ta sẽ dùng DataView vì nó có liên kết với DataTable – một DataView có thể được tùy chỉnh để trình bày một tập hợp từ DataTable. Để dễ dàng hình dung dữ liệu của DataView như thế nào hãy xem hình 4.2.1.6

```

public static (IDataView training, IDataView test) LoadData(MLContext mlContext, FilmsModel[] films)
{
    IDataView trainingDataGridView = mlContext.Data.LoadFromEnumerable<FilmsModel>(films);
    IDataView testDataView = trainingDataGridView;

    return (trainingDataGridView, testDataView);
}

```

Hình 4.2.1.4: Hàm tạo dữ liệu cho đào tạo mô hình

```
(trainingDataGridView, testDataView) = LoadData(mlContext, fmd);
```

Hình 4.2.1.5: Tạo dữ liệu cho mô hình

Name	Value
trainingDataGridView	(Microsoft.MLData.DataViewConstructionUtils.StreamingDataGridView<TrainAPI.Models.FilmsModel>)
CanShuffle	false
Host	(Microsoft.MLData.LocalEnvironment.Host)
Schema	(7 columns)
[0]	(id: Int32)
[1]	(title: String)
[2]	(run_time: Int32)
[3]	(release_date: String)
[4]	(username: String)
[5]	(evaluate: Single)
[6]	(poster_path: String)

Hình 4.2.1.6: Dữ liệu DataView của mô hình

4.2.2. Xây dựng và đào tạo model

- Đầu tiên chúng ta sẽ lấy cột username và id của phim để xác định sự biến đổi giữa chúng và chuyển đổi mỗi username ứng với mỗi id phim thành một khóa. Ví dụ, xem hình 4.2.2.1

```
IEstimator<ITransformer> estimator = mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "userIdEncoded", inputColumnName: "username")
    .Append(mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "movieIdEncoded", inputColumnName: "id"));
```

userId	movieId	Label	userIdEncoded	movieIdEncoded
1	1	4	userKey1	movieKey1
1	3	4	userKey1	movieKey2
1	6	4	userKey1	movieKey3

Hình 4.2.2.1: Khung dữ liệu khi biến thành khóa

Nguồn: <https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/movie-recommendation>

- Chọn thuật toán và thêm những từ khóa này vào định nghĩa những giá trị cần thiết. Với thuật toán được chọn ở đây là MatrixFactorizationTrainer là một thuật toán chuyên dùng để dự đoán giá trị trong ma trận bằng cách sử dụng thừa số hóa ma trận. Cụ thể, là chúng ta sẽ tìm ma trận w tương ứng với mỗi sao mà user đó đã cho.

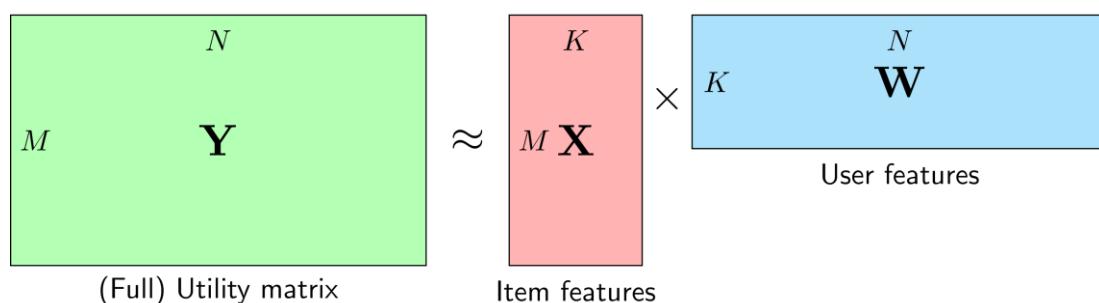
$$\mathbf{Y} \approx \mathbf{xw}$$

Trong đó, Y là dữ liệu đầu vào của chúng ta, x là dữ liệu của username được cho rằng w sẽ sắp xỉ.

Nếu Y là ma trận cấp MxN

Thì x sẽ có cấp MxK (với $M < K \leq N$) và w sẽ cấp KxN (với $M < K \leq N$).

$$\mathbf{Y} \approx \hat{\mathbf{Y}} = \mathbf{XW}$$



Hình 4.2.2.2: Mô hình dự đoán cho ma trận w

Nguồn: <https://machinelearningcoban.com/2017/05/31/matrixfactorization/>

XÂY DỰNG TRANG WEB ĐỂ XUẤT PHIM

GVHD: TS. Huỳnh Xuân Phụng

Sau đây là đoạn chương trình hiện thực những điều trên.

```
var options = new MatrixFactorizationTrainer.Options
{
    MatrixColumnIndexColumnName = "userIdEncoded",
    MatrixRowIndexColumnName = "movieIdEncoded",
    LabelColumnName = "evaluate",
    NumberOfIterations = 800,
    ApproximationRank = 100
};
options.LossFunction = MatrixFactorizationTrainer.LossFunctionType.SquareLossOneClass;
options.Alpha = 0.01;
options.Lambda = 0.025;
var trainerEstimator = estimator.Append(mlContext.Recommendation().Trainers.MatrixFactorization(options)); s 10ms elapsed
```

Hình 4.2.2.3: Cấu hình cho thuật toán

- Khi đào tạo thành công chúng ta sẽ nhận được bộ dữ liệu gồm 1974 phim với 6 username ứng với 6 tài khoản người dùng làm dữ liệu.

model	(Microsoft.ML.Data.TransformerChain<Microsoft.ML.Trainers.Recommender.MatrixFactorizationPredictionTransformer>)
LastTransformer	{Microsoft.ML.Trainers.Recommender.MatrixFactorizationPredictionTransformer}
Model	{Microsoft.ML.Trainers.Recommender.MatrixFactorizationModelParameters}
ApproximationRank	100
LeftFactorMatrix	{float[197400]}
NumberOfColumns	6
NumberOfRows	1974
RightFactorMatrix	{float[600]}
Static members	
Non-Public members	
Static members	
Non-Public members	

Hình 4.2.2.4: Dữ liệu khi đào tạo hoàn tất

- Để đánh giá mô hình chúng ta như thế nào thì có thể thực hiện như sau

```
var prediction = model.Transform(testDataView);

var metrics = mlContext.Regression.Evaluate(prediction, labelColumnName: "evaluate", scoreColumnName: "Score");
```

Hình 4.2.2.5: Đánh giá mô hình khi đã đào tạo

- Khi thực hiện thành công bộ dữ liệu chúng ta nhận được là:

metrics	(Microsoft.ML.Data.RegressionMetrics)
LossFunction	3.5664796433161192
MeanAbsoluteError	1.5261889328347875
MeanSquaredError	3.5664795851807694
RSquared	-1.8826424775721611
RootMeanSquaredError	1.8885125324394247

Hình 4.2.2.6: Kết quả đánh giá

Trong đó có 2 giá trị bạn cần chú ý đó là Rsquared cho biết dữ liệu phù hợp với mô hình như thế nào, giá trị càng gần 1 có nghĩa là mô hình càng khớp chính xác với dữ liệu đầu vào. Đóng nghĩa với độ chính xác càng cao. Giá trị thứ 2 bạn cần chú ý đó là RootMeanSquaredError nó được sử dụng để đo lường sự khác biệt giữa các giá trị mô hình dự đoán và các giá trị quan sát kiểm tra dữ liệu, nó là căn bậc 2 của trung bình các bình phương lỗi, ngược lại với Rsquared thì giá trị này càng thấp thì càng tốt.

4.2.3. Sử dụng mô hình

- Đọc mô hình đã đào tạo vào tạo các mảng dữ liệu cần thiết.

```

var modelPath = Path.Combine(Environment.CurrentDirectory, "Models", "MovieRecommenderModel.zip");
var model = mlContext.Model.Load(modelPath, out schema);
var predictionEngine = mlContext.Model.CreatePredictionEngine<FilmsModel, MovieRatingPrediction>(model);

List<FilmsModel> result = new List<FilmsModel>();
double[] index = new double[1974];

```

Hình 4.2.2.7: Đọc mô hình đã dự đoán và tạo dữ liệu

- Sau khi đọc mô hình bây giờ chúng ta sẽ tìm xem username được dự đoán đã xem nhiều bộ phim của những username đã được đào tạo nhất và dĩ nhiên điều này là xác định ma trận x đã được nhắc ở trên.
- Sau khi đã có được username mà nó tương đồng nhất thì điều còn lại là đem từng bộ phim của toàn bộ phim để dự đoán cho username được dự đoán. Tức là đi tìm từ đối số cho cho ma trận w.

Để làm được việc đó chúng ta phải dùng một mảng tạm là index để lưu những giá trị mà mô hình dự đoán cho chúng ta

```

for (int i = 0; i < 1974; i++)
{
    if (testData[i].username != username)
    {
        var testInput = new FilmsModel(testData[i].id, username, testData[i].title, 0, testData[i].poster_path, testData[i].run_time, testData[i].release_date);
        var movieRatingPrediction = predictionEngine.Predict(testInput);
        index[i] = (100 / (1 + Math.Exp(-Math.Round(movieRatingPrediction.Score, 1))));
    }
    else
    {
        var testInput = new FilmsModel(testData[i].id, username, testData[i].title, testData[i].evaluate, testData[i].poster_path, testData[i].run_time, testData[i].release_date);
        var movieRatingPrediction = predictionEngine.Predict(testInput);
        index[i] = (100 / (1 + Math.Exp(-Math.Round(movieRatingPrediction.Score, 1))));
    }
}

```

Hình 4.2.2.8: Dự đoán tất cả phim cho một người dùng

- Sau khi đã đầy đủ các giá trị việc còn lại chỉ là tìm xem bộ phim nào có khả năng username có khả năng nhất bằng cách tìm phần tử lớn nhất trong index. Ở đây chúng ta sẽ lấy 36 phần tử lớn nhất ứng với 36 bộ phim.

```

double temp;
int tempIndex;
for (int i = 0; i < 36; i++)
{
    temp = index.Max();
    tempIndex = index.ToList().IndexOf(temp);
    result.Add(testData[tempIndex]);
    testData = testData.Where((val, idx) => idx != tempIndex).ToArray();
    index = index.Where((val, idx) => idx != tempIndex).ToArray();
}
return result;

```

Hình 4.2.2.9: Lấy 36 kết quả dự đoán cao nhất

4.3. Kết luận

Dự đoán dựa trên xếp hạng (1-5) mà người dùng đánh giá cho một bộ phim cụ thể và đề xuất bộ phim khác theo những người dùng khác mà đã xem phim đó và có khả năng thích một bộ phim khác nữa. Đó là điều mà nhóm chúng em đã thực hiện được.

Tuy nhiên, do tìm kiếm và khảo sát khó khăn trên bộ dữ liệu lớn (1974 bộ phim) và web vẫn chưa được nhiều người biết đến cũng như là cần sự tương tác lâu dài nên bộ dữ liệu cho phần này là được tạo random không sát với thực tế gây ảnh hưởng nhiều tới kết quả dự đoán.

PHẦN KẾT LUẬN

Với những gì thầy Phụng đã hướng dẫn đến nay trang web của chúng em cung cấp hơn 1900 phim với thể loại vô cùng phong phú cùng với giao diện hợp lý, dễ dàng sử dụng, những chức năng cơ bản của một web phim cần có, đặc biệt là chức năng dự đoán phim liên quan và dự đoán phim cho từng tài khoản khác nhau. Song với đó chúng em còn ưu tiên đến trải nghiệm người dùng khi xem phim nên chúng em quyết định cho phép người dùng xem trailer, mô tả, bình luận, đánh giá kèm theo nếu thấy thích thì họ có thể xem phim để tránh mất thời gian vào một bộ phim mà mình không thích thú.

- Những điều đã đạt được:

Thiết kế trang web thân thiện, dễ sử dụng.

Áp dụng được thuật toán dự đoán vào hệ thống.

Xem được phim trên hệ thống.

Tạo được chức năng đánh giá phim.

Quản lý được người dùng.

- Những điều chưa làm được:

Người dùng không thể trả lời bình luận nhiều lớp.

Chưa responsive tốt cho giao diện điện thoại.

Trailer phim, diễn viên còn ít, chưa phong phú.

Trang quản lý còn đơn giản.

- Hướng phát triển:

Thiết kế lại trang quản lý.

Áp dụng nhiều công nghệ mới.

Cải thiện lại hiệu suất.

Thêm chức năng xem phim trả phí.

Tạo tài khoản VIP cho người dùng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. INFOQ VIỆT NAM, “*KẾT QUẢ NGHIÊN CỨU Khảo sát về các trang bán lẻ trực tuyến*”, <https://cpanel.infoq.vn/common/reports/files/khao-sat-ve-nen-tang-xem-phim-truc-tuyen.pdf>, Tháng 9 2021, 15-16.
- [2]. *Build a Movie Recommendation System in Python using Machine Learning*,
<https://techvidvan.com/tutorials/movie-recommendation-system-python-machine-learning/>
- [3]. *Build a movie recommender using matrix factorization with ML.NET*,
<https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/movie-recommendation>, 29/11/2021.