

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO**

—***—



**THỰC HÀNH PHÁT TRIỂN
HỆ THỐNG TRÍ TUỆ NHÂN TẠO
BÁO CÁO BÀI TẬP LỚN**

**NHÓM 5
Ứng dụng xem phim**

Giảng viên:
TS. Trần Quốc Long
ThS. Nguyễn Quang Minh

Sinh viên:
Đặng Văn Khải 22022550
Lê Việt Hùng 22022666
Đàm Thái Ninh 22022522
Dương Minh Đức 22022606

Contents

1 Mô tả yêu cầu	4
1.1 Phát Biểu Bài Toán	4
1.1.1 Bài toán	4
1.1.2 Vấn đề	4
1.1.3 Mục tiêu	4
1.2 Yêu Cầu Hệ Thống	5
1.2.1 Yêu cầu chức năng (Functional Requirements)	5
1.2.2 Yêu cầu phi chức năng (Non-functional Requirements)	5
2 Mô hình gợi ý phim	7
2.1 Giới thiệu về bài toán gợi ý phim	7
2.2 Chuẩn bị dữ liệu	7
2.3 Thuật toán Neural Collaborative Filtering (NCF)	8
2.4 Dự đoán	9
2.5 Ưu điểm, nhược điểm và phương pháp cải tiến cho mô hình	10
3 Mô tả Use Cases	11
3.1 Biểu đồ	11
3.2 Mô tả chi tiết	12
3.2.1 Đăng ký tài khoản (User Registration)	12
3.2.2 Đăng nhập (User Login)	13
3.2.3 Đăng xuất (User Logout)	13
3.2.4 Quản lý hồ sơ người dùng (User Profile Management)	14
3.2.5 Quản lý danh sách phim yêu thích (Watchlist Management)	15
3.2.6 Browse Movies (Duyệt Danh Sách Phim)	15
3.2.7 View Movie Details (Xem Chi Tiết Phim)	16
3.2.8 Movie Management	16
3.2.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)	17
3.2.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)	18
3.2.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)	18
3.2.12 Review Movie (Đánh Giá Phim)	18
4 Kiến Trúc Dự Án MotChill	20
4.1 Kiến Trúc Tổng Quan	20
4.2 Các Thành Phần Chính	20
4.3 Kiến Trúc Layered	21
4.4 Mô Hình MVC của Django	21
4.5 Quan Hệ Giữa Các Thành Phần	21
4.6 Thiết Kế Hướng Đến Khả Năng Mở Rộng	21
5 Biểu đồ mức phân tích	22
5.1 Đăng ký tài khoản (User Registration)	22
5.2 Đăng nhập (User Login)	24
5.3 Đăng xuất (User Logout)	26
5.4 Quản lý hồ sơ người dùng (User Profile Management)	28
5.5 Quản lý danh sách phim yêu thích (Watchlist Management)	32
5.6 Browse Movies (Duyệt Danh Sách Phim)	34
5.7 View Movie Details (Xem Chi Tiết Phim)	35
5.8 Movie Management	36

5.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)	38
5.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)	39
5.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)	40
5.12 Review Movie (Đánh Giá Phim)	43
6 Biểu đồ mức thiết kế	46
6.1 Đăng ký tài khoản (User Registration)	46
6.2 Đăng nhập (User Login)	48
6.3 Đăng xuất (User Logout)	50
6.4 Quản lý hồ sơ người dùng (User Profile Management)	51
6.5 Quản lý danh sách phim yêu thích (Watchlist Management)	54
6.6 Browse Movies (Duyệt Danh Sách Phim)	57
6.7 View Movie Details (Xem Chi Tiết Phim)	59
6.8 Movie Management	60
6.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)	63
6.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)	64
6.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)	65
6.12 Review Movie (Đánh Giá Phim)	68
7 Databases	71
8 Demo	72
8.1 Video Demo Sản Phẩm	72
8.2 Github	72
9 Đóng góp	74
9.1 Bản tự đánh giá đóng góp của từng thành viên	74
10 Tổng kết	75
10.1 Một số vấn đề ứng dụng giải quyết	75
10.2 Cấu trúc dự án	75
10.3 Điểm tích cực	76
10.4 Các điểm có thể cải thiện	76

Chapter 1

Mô tả yêu cầu

1.1 Phát Biểu Bài Toán

1.1.1 Bài toán

Xây dựng một hệ thống web quản lý nội dung phim MotChill (MovieSubscription), một nền tảng dịch vụ video theo yêu cầu cho phép người dùng:

- Đăng ký tài khoản và đăng nhập vào hệ thống.
- Quản lý hồ sơ cá nhân, bao gồm cập nhật thông tin và xem lịch sử hoạt động.
- Truy cập và tương tác với nội dung phim, bao gồm xem danh sách phim, chi tiết phim, và quản lý danh sách phim yêu thích.
- Đăng ký các gói dịch vụ khác nhau, với tính năng thanh toán an toàn và thuận tiện qua PayPal.
- Để lại đánh giá và xếp hạng cho các bộ phim, góp phần vào việc cải thiện nội dung và gợi ý phim cho người dùng khác.

Hệ thống cần hỗ trợ các chức năng cơ bản như xác thực người dùng, quản lý hồ sơ, xử lý các yêu cầu liên quan đến phim, và đánh giá phim.

1.1.2 Vấn đề

Hiện nay, các nền tảng cung cấp nội dung video thường gặp các vấn đề sau:

- Trải nghiệm người dùng chưa tối ưu:** Người dùng thường gặp khó khăn trong việc tìm kiếm phim phù hợp với sở thích cá nhân, cũng như quản lý danh sách phim yêu thích một cách hiệu quả.
- Bảo mật thanh toán:** Người dùng lo ngại về tính an toàn khi thực hiện các giao dịch thanh toán trực tuyến, đặc biệt là với các phương thức thanh toán qua cổng thứ ba như PayPal.
- Quản lý nội dung phim:** Các quản trị viên gặp khó khăn trong việc quản lý một lượng lớn nội dung phim, bao gồm việc cập nhật thông tin, giá cả, và đánh giá phim từ người dùng.
- Khả năng mở rộng:** Khi số lượng người dùng tăng lên, hệ thống hiện tại không thể mở rộng một cách dễ dàng để đáp ứng nhu cầu sử dụng của người dùng.
- Tích hợp AI chưa hiệu quả:** Hệ thống gợi ý phim chưa thực sự hiệu quả, không mang lại những đề xuất phù hợp với sở thích và hành vi của người dùng.

1.1.3 Mục tiêu

Mục tiêu của bài toán là khắc phục các vấn đề trên bằng cách xây dựng một hệ thống quản lý nội dung phim với trải nghiệm người dùng tối ưu, bảo mật thanh toán cao, khả năng mở rộng tốt, và tích hợp AI hiệu quả để gợi ý phim phù hợp với người dùng. Cụ thể:

- Cung cấp nền tảng dịch vụ video theo yêu cầu:** Cho phép người dùng truy cập vào nội dung phim dựa trên các gói dịch vụ mà họ đăng ký, từ đó tạo ra một trải nghiệm giải trí cá nhân hóa.

- **Cải thiện trải nghiệm người dùng:** Hỗ trợ người dùng trong việc tìm kiếm, xem chi tiết phim, quản lý danh sách phim yêu thích, và theo dõi các bộ phim mới ra mắt.
- **Tích hợp thanh toán an toàn:** Đảm bảo quy trình thanh toán diễn ra an toàn, bảo mật và tiện lợi thông qua tích hợp với dịch vụ thanh toán PayPal.
- **Quản lý và đánh giá phim:** Cho phép người dùng để lại đánh giá và xếp hạng cho các bộ phim, từ đó tạo ra cộng đồng người dùng và cải thiện chất lượng nội dung phim được đề xuất.

1.2 Yêu Cầu Hệ Thống

1.2.1 Yêu cầu chức năng (Functional Requirements)

Quản Lý Sản Phẩm

- **Thêm/Chỉnh sửa/Xóa Sản phẩm:** Quản trị viên có thể quản lý thông tin sản phẩm bao gồm thêm mới, chỉnh sửa, hoặc xóa các sản phẩm khỏi hệ thống.
- **Xem chi tiết Sản phẩm:** Người dùng có thể truy cập và xem thông tin chi tiết của từng sản phẩm, bao gồm tên, giá, mô tả, hình ảnh và đánh giá từ người dùng khác.

Quản Lý Người Dùng

- **Đăng ký/Đăng nhập:** Cung cấp tính năng tạo tài khoản và đăng nhập để người dùng có thể truy cập và sử dụng hệ thống.
- **Quản lý Hồ sơ Người dùng:** Cho phép người dùng cập nhật thông tin cá nhân, xem lịch sử mua hàng, và quản lý các thông tin liên quan đến tài khoản của họ.

Gợi Ý Sản Phẩm

- **Tạo gợi ý dựa trên hành vi Người dùng:** Hệ thống sẽ theo dõi và phân tích hành vi duyệt web và mua hàng của người dùng để đề xuất các sản phẩm phù hợp với sở thích của họ.
- **Cập nhật mô hình gợi ý:** Mô hình gợi ý cần được duy trì và cập nhật thường xuyên dựa trên dữ liệu hành vi mới nhất của người dùng để đảm bảo độ chính xác và hiệu quả.

Giỏ Hàng và Thanh Toán

- **Quản lý Giỏ hàng:** Người dùng có thể thêm, xóa, hoặc cập nhật số lượng sản phẩm trong giỏ hàng của họ một cách dễ dàng.
- **Thanh toán:** Đảm bảo quy trình thanh toán an toàn, hỗ trợ đa dạng phương thức thanh toán, đặc biệt là tích hợp với PayPal để mang đến trải nghiệm mua sắm mượt mà.

1.2.2 Yêu cầu phi chức năng (Non-functional Requirements)

Hiệu Suất

- **Thời gian phản hồi:** Hệ thống cần đáp ứng yêu cầu phản hồi nhanh chóng, đảm bảo thời gian phản hồi dưới 3 giây để tối ưu hóa trải nghiệm người dùng.
- **Khả năng mở rộng:** Hệ thống phải có khả năng mở rộng linh hoạt để đáp ứng tăng trưởng người dùng trong tương lai mà không làm giảm hiệu suất.

Bảo Mật

- **Bảo mật dữ liệu người dùng:** Đảm bảo tuân thủ các tiêu chuẩn bảo mật nghiêm ngặt để bảo vệ dữ liệu cá nhân và giao dịch của người dùng.
- **Xác thực và ủy quyền:** Đảm bảo chỉ những người dùng có quyền mới có thể truy cập vào các chức năng nhất định, tăng cường tính bảo mật và an toàn cho hệ thống.

Độ Tin Cậy

- **Khả năng phục hồi:** Hệ thống phải có khả năng tự phục hồi sau các sự cố kỹ thuật mà không làm mất dữ liệu hoặc gây gián đoạn dịch vụ.
- **Sao lưu:** Dữ liệu cần được sao lưu định kỳ để phòng ngừa mất mát do lỗi hệ thống hoặc các sự cố bất ngờ.
- **Nhật ký (log):** Ghi lại các hoạt động quan trọng của hệ thống để hỗ trợ trong việc gỡ lỗi và giám sát hoạt động.

Khả Năng Sử Dụng

- **Giao diện:** Hệ thống cần có giao diện thân thiện, trực quan, dễ sử dụng cho mọi đối tượng người dùng, từ người mới đến người đã có kinh nghiệm.
- **Đa ngôn ngữ:** Hỗ trợ nhiều ngôn ngữ khác nhau để mở rộng đối tượng người dùng và tiếp cận khách hàng quốc tế.
- **Đa thiết bị:** Hệ thống phải tương thích với nhiều loại thiết bị khác nhau như máy tính, máy tính bảng, và điện thoại di động, đảm bảo trải nghiệm nhất quán trên mọi nền tảng.

Chapter 2

Mô hình gợi ý phim

2.1 Giới thiệu về bài toán gợi ý phim

Giới thiệu: Bài toán gợi ý phim (Movie Recommendation) là một trong những ứng dụng tiêu biểu của hệ thống gợi ý (Recommender Systems), nhằm giúp người dùng khám phá các bộ phim mà họ có thể quan tâm dựa trên sở thích cá nhân và hành vi tương tác của họ với các bộ phim khác. Với sự bùng nổ của các nền tảng xem phim trực tuyến như Netflix, Amazon Prime, và Disney+, việc cung cấp các gợi ý chính xác và cá nhân hóa đã trở thành một yếu tố quan trọng để giữ chân người dùng và nâng cao trải nghiệm của họ.

Phương pháp: Hệ thống gợi ý phim có thể được phát triển bằng nhiều phương pháp khác nhau, bao gồm lọc cộng tác (Collaborative Filtering), lọc theo nội dung (Content-Based Filtering), và các phương pháp kết hợp (Hybrid Methods). Để xây dựng một hệ thống hiệu quả, cần xử lý một khối lượng dữ liệu phong phú, từ thông tin người dùng và đánh giá phim (ratings), đến các đặc điểm chi tiết của từng bộ phim như thể loại, diễn viên, và đạo diễn. Đặc biệt, các thuật toán học máy, nhất là các mô hình học sâu (Deep Learning), đã được ứng dụng để nâng cao độ chính xác và tính cá nhân hóa của các gợi ý, từ đó mang đến trải nghiệm xem phim tối ưu cho người dùng.

Mục tiêu: Xây dựng một hệ thống có khả năng dự đoán và gợi ý những bộ phim mà người dùng có thể yêu thích, từ đó tối ưu hóa trải nghiệm xem phim và nâng cao mức độ hài lòng của người dùng.

2.2 Chuẩn bị dữ liệu

Tập dữ liệu sử dụng cho mô hình gợi ý phim: MovieLens 100k ([Link download](#))

Tổng quan về tập dữ liệu: MovieLens 100k là một tập dữ liệu phổ biến trong lĩnh vực hệ thống gợi ý, được cung cấp bởi GroupLens Research. Đây là một nguồn dữ liệu quý giá cho các nghiên cứu và thực nghiệm trong lĩnh vực gợi ý phim. Dưới đây là tổng quan chi tiết về tập dữ liệu này:

- **Nguồn gốc:** GroupLens Research
- **Số lượng người dùng:** 943
- **Số lượng phim:** 1682
- **Số lượng đánh giá:** 100,000
- **Thời gian thu thập:** tháng 4 năm 1998 đến tháng 3 năm 1999

Cấu trúc tập dữ liệu: Tập dữ liệu MovieLens 100k bao gồm các tệp tin

- **u.data:** Chứa thông tin đánh giá phim
- **u.item:** Chứa thông tin về phim
- **u.user:** Chứa thông tin về người dùng
- **u.genre:** Chứa thông tin về các thể loại phim

Lấy dữ liệu: Để phát triển mô hình gợi ý phim, chúng tôi tập trung vào các thông tin sau:

- Từ tập tin **u.data**, chúng tôi chỉ quan tâm đến:

- **userId** (ID của người dùng)
- **movield** (ID của bộ phim mà người dùng đã đánh giá)
- **rating** (đánh giá của người dùng đối với bộ phim)
- > Lưu **userId**, **movield**, **rating** từ **u.data** vào **ratings.csv**

- Từ tập tin **u.item**, chúng tôi chú ý đến:

- **movield** (ID của bộ phim)
- **title** (tên của bộ phim)
- **genres** (thể loại của bộ phim)
- > Lưu **movield**, **title**, **genres** từ **u.item** vào **movies.csv**

- Các thông tin như tags hay imbdId, tmdbdId cũng được lưu lại nhưng không được sử dụng cho mô hình gợi ý phim. Những thông tin về người dùng như tuổi tác, nghề nghiệp, giới tính không được lưu lại.

Xử lý dữ liệu: Thực hiện theo các bước sau:

1. Đọc dữ liệu từ **ratings.csv** và **movies.csv**, sau đó kết hợp cột **genres** từ **movies.csv** vào **ratings.csv** thông qua **movield**.
2. Chuyển đổi cột **genres** thành các vector one-hot, trong đó mỗi cột đại diện cho một thể loại phim; giá trị 0 biểu thị phim không thuộc thể loại đó, và 1 biểu thị phim thuộc thể loại đó.
3. Chuyển đổi **userId** và **movield** thành các chỉ số index tương ứng, sau đó tổ chức dữ liệu bằng cách đưa **userId**, **movield**, và **genres** vào **X**, còn rating vào **y** tương ứng.
4. Chia dữ liệu thành các batch nhỏ và phân tách thành tập train và validation, chuẩn bị sẵn sàng cho quá trình huấn luyện mô hình.

2.3 Thuật toán Neural Collaborative Filtering (NCF)

Giới thiệu: Neural Collaborative Filtering (NCF) là một kỹ thuật tiên tiến trong hệ thống gợi ý, được thiết kế để cải thiện khả năng dự đoán và gợi ý bằng cách kết hợp sức mạnh của các mạng nơ-ron với phương pháp lọc cộng tác truyền thống. Khác với các phương pháp lọc cộng tác truyền thống chỉ dựa vào các ma trận phân rã, NCF tận dụng các mô hình học sâu để học các yếu tố ẩn và tương tác giữa người dùng và các đối tượng.

Ứng Dụng Thuật Toán NCF Để Gợi Ý Phim Với Bộ Dữ Liệu MovieLens 100k

Thuật toán Neural Collaborative Filtering (NCF) là một phương pháp mạnh mẽ để gợi ý phim dựa trên tương tác giữa người dùng và phim. Trong phần này, chúng ta sử dụng một mô hình mạng nơ-ron dựa trên kỹ thuật embedding để học các yếu tố ẩn liên quan đến người dùng và phim từ bộ dữ liệu MovieLens 100k.

Mô Hình EmbeddingNet:

- **Lớp Embedding:** Mô hình sử dụng các lớp embedding để chuyển đổi **userId** và **movield** thành các vector ẩn, mỗi vector có kích thước là **n_factors**. Điều này giúp mô hình học được các đặc trưng phức tạp của người dùng và phim.
- **Tích hợp các đặc trưng:** Mô hình kết hợp các đặc trưng ẩn từ người dùng và phim cùng với các đặc trưng của thể loại phim (genres) để tạo ra một vector đặc trưng đầu vào cho tầng ẩn.
- **Hidden Layers:** Các hidden layers của mô hình được cấu hình linh hoạt với số lượng lớp và số lượng nơ-ron khác nhau, cùng với các lớp dropout được áp dụng sau mỗi tầng để giảm thiểu hiện tượng overfitting.
- **Output Layer:** Đầu ra của mô hình là một giá trị dự đoán được kích hoạt bởi hàm sigmoid, cho phép dự đoán điểm số rating của người dùng cho mỗi bộ phim. Nếu có giá trị **minmax** cung cấp, đầu ra sẽ được chuẩn hóa lại theo khoảng điểm số của bộ dữ liệu.
- **Khởi tạo trọng số:** việc khởi tạo trọng số đóng vai trò quan trọng để đảm bảo mô hình học hiệu quả và hội tụ nhanh chóng. Các trọng số của mô hình được khởi tạo như sau:

- **Tầng Embedding:** Các trọng số trong các tầng embedding của người dùng (**self.u.weight**) và phim (**self.m.weight**) được khởi tạo ngẫu nhiên trong khoảng từ -0.05 đến 0.05. Điều này giúp đảm bảo rằng các vector ẩn bắt đầu từ các giá trị nhỏ và ngẫu nhiên, giúp mô hình bắt đầu quá trình học với các đặc trưng ẩn đa dạng.
- **Tầng Fully Connected:** Các trọng số được khởi tạo bằng phương pháp **Xavier uniform initialization**. Phương pháp này giúp phân phối các trọng số sao cho chúng nằm trong một khoảng hợp lý, giúp các nơ-ron hoạt động đồng đều từ đầu. Đồng thời, các giá trị bias trong các tầng tuyến tính được khởi tạo với giá trị 0.01 để đảm bảo rằng các nơ-ron có một sự chênh nhẹ khỏi zero ban đầu, giúp quá trình lan truyền ngược diễn ra hiệu quả hơn.

Huấn luyện mô hình:

- **Tối ưu hóa (Optimizer):** Áp dụng optimizer Adam với một learning rate phù hợp để tối ưu hóa quá trình huấn luyện mô hình. Đồng thời, sử dụng weight decay nhằm hạn chế tình trạng overfitting, giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới.
- **Tiêu chí đánh giá (Criterion):** Sử dụng hàm mất mát MSELoss (Mean Squared Error Loss) để đo lường độ sai lệch giữa giá trị dự đoán của mô hình và giá trị thực tế. Điều này giúp mô hình điều chỉnh lại các trọng số để giảm thiểu sai số qua từng bước huấn luyện.
- **Điều chỉnh learning rate (Scheduler):** Triển khai CyclicLR để thay đổi learning rate một cách tuần hoàn trong quá trình huấn luyện. Việc điều chỉnh này giúp mô hình thoát khỏi các điểm cực trị cục bộ và tăng cường khả năng hội tụ đến điểm cực trị toàn cục, cải thiện hiệu suất tổng thể của mô hình.
- **Dừng sớm (Early Stopping):** Kết thúc quá trình huấn luyện khi không thấy sự cải thiện đáng kể trong validation loss. Đồng thời, khôi phục lại trạng thái mô hình tốt nhất đã đạt được trước đó để đảm bảo hiệu suất tối ưu.

2.4 Dự đoán

Với những người dùng chưa đánh giá bộ phim nào

Đối với những người dùng chưa từng đánh giá bộ phim nào, hệ thống gợi ý sẽ áp dụng công thức trung bình trọng số, một phương pháp phổ biến mà IMDB sử dụng để xếp hạng phim. Phương pháp này xác định các bộ phim hàng đầu dựa trên sự kết hợp giữa số lượng đánh giá mà mỗi bộ phim nhận được và điểm đánh giá trung bình của chúng, đảm bảo tính khách quan và công bằng trong việc xếp hạng.

Công thức trung bình trọng số, thường được IMDB sử dụng để xếp hạng phim, được biểu diễn như sau:

$$WR = \frac{v}{v+m} \cdot R + \frac{m}{v+m} \cdot C$$

Trong đó:

- WR : Đánh giá trung bình trọng số (weighted rating).
- R : Đánh giá trung bình của bộ phim (average rating).
- v : Số lượng đánh giá của bộ phim (number of votes).
- m : Ngưỡng tối thiểu các bộ phim phải có để đủ điều kiện xếp hạng (thường là phân vị thứ 90 của số lượng đánh giá).
- C : Đánh giá trung bình toàn cục (mean rating across all movies).

Các bước thực hiện như sau:

- Tính toán đánh giá trung bình toàn cục (C) và phân vị thứ 90 của số lượng đánh giá cho mỗi bộ phim (m).
- Tạo một bảng tổng hợp với số lượng đánh giá (`num_ratings`) và đánh giá trung bình (`avg_rating`) cho mỗi bộ phim.
- Lọc ra những bộ phim có số lượng đánh giá lớn hơn hoặc bằng phân vị thứ 90 (m).
- Sử dụng công thức trung bình trọng số để tính điểm đánh giá cho từng bộ phim được chọn. Công thức này kết hợp giữa đánh giá trung bình toàn cục và đánh giá trung bình của từng bộ phim dựa trên số lượng đánh giá của nó.
- Chọn ra 2^*k bộ phim có điểm đánh giá trung bình theo trọng số cao nhất, sau đó chọn ngẫu nhiên top-k bộ phim từ danh sách này và trả về kết quả.

Với những người dùng đã đánh giá phim

Đối với những người dùng đã có đánh giá về phim, ta áp dụng học gia tăng (incremental learning) để tiếp thu thông tin từ dữ liệu của họ. Phương pháp này cho phép mô hình nhận diện được sự tương đồng giữa người dùng mới và những người dùng cũ đã có sẵn dữ liệu. Từ đó, mô hình có khả năng dự đoán mức độ yêu thích của người dùng đối với các bộ phim họ chưa xem, và đưa ra danh sách những bộ phim có tiềm năng cao được người dùng đánh giá tích cực.

Chúng tôi sẽ không lưu lại mô hình sau mỗi lần học gia tăng, mà thay vào đó sẽ cập nhật mô hình theo chu kỳ định kỳ, chẳng hạn như mỗi tháng một lần, nếu số lượng người dùng mới đủ lớn để tạo ra sự thay đổi đáng kể.

2.5 Ưu điểm, nhược điểm và phương pháp cải tiến cho mô hình

Ưu điểm

- Khả năng học phi tuyến:** Neural Collaborative Filtering (NCF) sử dụng các lớp fully connected phi tuyến, cho phép mô hình học các tương tác phức tạp giữa người dùng và phim, khác với các mô hình tuyến tính như Matrix Factorization (MF).
- Dễ dàng mở rộng:** Mô hình NCF có thể dễ dàng mở rộng để tích hợp nhiều nguồn thông tin khác như thông tin về người dùng (độ tuổi, giới tính,...) và các đặc điểm khác của phim, cho phép tạo ra các dự đoán chính xác hơn.
- Khả năng học liên tục với dữ liệu mới:** Mô hình NCF có thể dễ dàng tiếp tục quá trình huấn luyện với dữ liệu mới mà không cần phải bắt đầu lại từ đầu, điều này giúp nó linh hoạt hơn so với các mô hình truyền thống như Matrix Factorization, vốn yêu cầu tái huấn luyện toàn bộ.
- Tiết kiệm bộ nhớ:** Thay vì phải lưu trữ toàn bộ ma trận trọng số lớn, mô hình NCF chỉ yêu cầu lưu trữ các embeddings và tham số mô hình, giúp giảm thiểu nhu cầu về bộ nhớ và tăng cường khả năng triển khai trong các hệ thống thực tế.

Nhược điểm

- Độ phức tạp tính toán:** Mô hình NCF thường yêu cầu nhiều tài nguyên tính toán hơn so với các mô hình truyền thống như Matrix Factorization, đặc biệt là khi mô hình sâu hơn hoặc có nhiều tham số.
- Cần nhiều dữ liệu hơn:** Mô hình NCF thường cần nhiều dữ liệu để học được các mối tương quan phức tạp giữa người dùng và phim, trong khi các mô hình đơn giản có thể hoạt động tốt hơn với ít dữ liệu.
- Khó khăn trong việc tối ưu hóa:** Do có nhiều lớp và tham số, việc tối ưu hóa mô hình NCF có thể trở nên khó khăn hơn và yêu cầu kỹ thuật tối ưu hóa phức tạp như sử dụng dynamic learning rate, gradient clipping,...
- Dự đoán tồn thời gian:** Trong khi các phương pháp sử dụng ma trận chỉ cần truy xuất trực tiếp giá trị dự đoán, mô hình NCF phải tính toán qua toàn bộ danh sách **movielid** để tìm ra lựa chọn tốt nhất, dẫn đến thời gian dự đoán dài hơn đáng kể.

Các phương pháp cải tiến

- Khai thác đa dạng thông tin cho dự đoán:** Mô hình có khả năng tận dụng thêm các đặc điểm chi tiết của người dùng như giới tính, tuổi tác, nghề nghiệp, học vấn,... để đưa ra dự đoán chính xác và cá nhân hóa hơn. Tuy nhiên điều này yêu cầu phải thu thập nhiều thông tin hơn từ người dùng.
- Kết hợp phương pháp lai (Hybrid Method):** Tích hợp các kỹ thuật để xuất phim khác nhau, tạo ra một mô hình mạnh mẽ và đa chiều, mang lại khả năng dự đoán chính xác và toàn diện hơn.
- Cải tiến mô hình** Nâng cao hiệu suất bằng cách sử dụng Attention Mechanisms để tập trung vào các tính năng quan trọng, Residual Connections để cải thiện học sâu, Ensemble Methods để kết hợp nhiều mô hình, Dynamic Learning Rate để tối ưu hóa quá trình huấn luyện, và thử nghiệm với các hàm loss như Hinge Loss hoặc BPR Loss thay vì MSE. Ngoài ra có thể thay đổi cấu trúc mô hình và tối ưu hóa tham số.

Chapter 3

Mô tả Use Cases

3.1 Biểu đồ

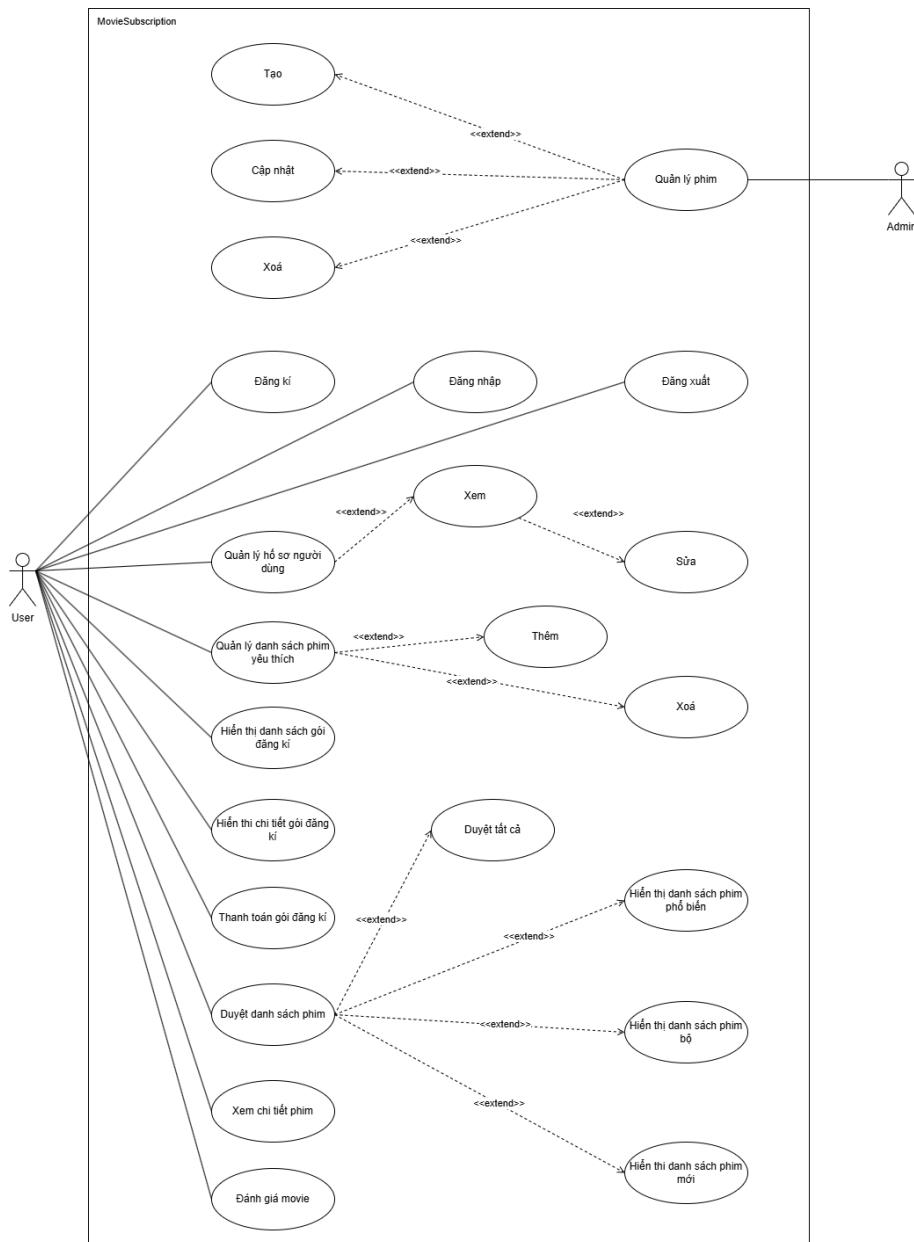


Figure 3.1: Biểu đồ Use Cases

3.2 Mô tả chi tiết

3.2.1 Đăng ký tài khoản (User Registration)

- **Tên Use Case:** Đăng ký tài khoản
- **Tham Chiếu (XRef):**
 - Frontend: user_profile/template/user_profile/signup.html
 - Backend: signup view in user_profile/views.py
- **Kích Hoạt (Trigger):** Người dùng nhấn nút "Sign up" trên trang đăng ký.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã truy cập vào màn hình đăng ký.
- **Luồng Chính (Basic Path):**

1. Người dùng nhập thông tin đăng ký bao gồm:
 - Tên người dùng (username)
 - Email
 - Mật khẩu (password)
 - Xác nhận mật khẩu (repass)
2. Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào:
 - Tên người dùng, email và mật khẩu không được bỏ trống.
 - Email phải có định dạng hợp lệ.
 - Mật khẩu phải đáp ứng các yêu cầu về độ phức tạp.
 - Mật khẩu và xác nhận mật khẩu phải trùng khớp.
3. Nếu thông tin hợp lệ, hệ thống kiểm tra xem email/số điện thoại đã được đăng ký hay chưa.
4. Nếu chưa được đăng ký, hệ thống tạo tài khoản mới cho người dùng:
 - Lưu trữ thông tin người dùng vào cơ sở dữ liệu.
 - Gửi thông báo tài khoản đã được tạo.
5. Hệ thống hiển thị thông báo thành công và chuyển hướng người dùng đến màn hình đăng nhập.

- **Luồng Phụ (Alternative Paths):**

1. Thông tin không hợp lệ:
 - Hệ thống hiển thị thông báo lỗi tương ứng với lỗi của từng trường thông tin.
 - Người dùng có thể sửa thông tin và thực hiện lại bước 2.
2. Email/Số điện thoại đã được đăng ký:
 - Hệ thống hiển thị thông báo lỗi cho người dùng biết email/số điện thoại đã được đăng ký.
 - Người dùng có thể chọn đăng nhập bằng tài khoản đã tồn tại hoặc thay đổi email/số điện thoại để đăng ký.

- **Hậu Điều Kiện (Postcondition):** Người dùng đã có một tài khoản mới trên hệ thống.

- **Đường Dẫn Ngoại Lệ (Exception Paths):**

1. Lỗi kết nối:
 - Hệ thống hiển thị thông báo lỗi kết nối.
 - Người dùng có thể thử lại sau.
2. Lỗi cơ sở dữ liệu:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.
3. Lỗi tạo tài khoản:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.

3.2.2 Đăng nhập (User Login)

- **Tên Use Case:** Đăng nhập
- **Tham Chiếu (XRef):**
 - Frontend: user_profile/template/user_profile/login.html
- **Kích Hoạt (Trigger):** Người dùng nhấn nút "Login" trên trang đăng nhập.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã truy cập vào màn hình đăng nhập.
- **Luồng Chính (Basic Path):**
 1. Người dùng nhập thông tin đăng nhập gồm email/số điện thoại và mật khẩu.
 2. Hệ thống kiểm tra thông tin đăng nhập:
 - Email/Số điện thoại và mật khẩu không được bỏ trống.
 - Email/Số điện thoại phải hợp lệ.
 - Mật khẩu phải khớp với tài khoản đăng ký.
 3. Nếu thông tin hợp lệ, hệ thống cho phép người dùng đăng nhập và chuyển hướng đến trang chủ hoặc trang trước đó.
- **Luồng Phụ (Alternative Paths):**
 1. Thông tin đăng nhập không hợp lệ:
 - Hệ thống hiển thị thông báo lỗi cho người dùng biết rằng thông tin đăng nhập không đúng.
 - Người dùng có thể nhập lại thông tin và thử đăng nhập lại.
- **Hậu Điều Kiện (Postcondition):** Người dùng đã đăng nhập thành công và có thể sử dụng các tính năng của hệ thống.
- **Đường Dẫn Ngoại Lệ (Exception Paths):**
 1. Lỗi kết nối:
 - Hệ thống hiển thị thông báo lỗi kết nối.
 - Người dùng có thể thử lại sau.
 2. Lỗi cơ sở dữ liệu:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.

3.2.3 Đăng xuất (User Logout)

- **Tên Use Case:** Đăng xuất
- **Tham Chiếu (XRef):**
 - Backend: logout view in user_profile/views.py
- **Kích Hoạt (Trigger):** Người dùng nhấn nút "Logout" trên giao diện.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã đăng nhập vào hệ thống.
- **Luồng Chính (Basic Path):**
 1. Người dùng nhấn nút "Logout".
 2. Hệ thống hủy phiên đăng nhập của người dùng.
 3. Hệ thống chuyển hướng người dùng đến màn hình đăng nhập hoặc trang chủ.
- **Hậu Điều Kiện (Postcondition):** Người dùng đã đăng xuất thành công và phiên làm việc của họ đã kết thúc.
- **Đường Dẫn Ngoại Lệ (Exception Paths):**
 1. Lỗi kết nối:
 - Hệ thống hiển thị thông báo lỗi kết nối.
 - Người dùng có thể thử lại sau.
 2. Lỗi cơ sở dữ liệu:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.

3.2.4 Quản lý hồ sơ người dùng (User Profile Management)

Xem thông tin cá nhân (View Profile)

- **Tên Use Case:** Xem thông tin cá nhân
- **Tham Chiếu (XRef):**
 - Frontend: user_profile/detail.html
 - Backend: user_profile_detail view in user_profile/views.py
- **Kích Hoạt (Trigger):** Người dùng nhấp vào liên kết "Profile" trên giao diện.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã đăng nhập vào hệ thống.
- **Luồng Chính (Basic Path):**
 1. Người dùng yêu cầu xem thông tin cá nhân.
 2. Hệ thống truy xuất thông tin cá nhân của người dùng từ cơ sở dữ liệu.
 3. Hệ thống hiển thị thông tin cá nhân của người dùng trên giao diện.
- **Hậu Điều Kiện (Postcondition):** Thông tin cá nhân của người dùng được hiển thị trên giao diện.
- **Đường Dẫn Ngoại Lệ (Exception Paths):**
 1. Lỗi kết nối:
 - Hệ thống hiển thị thông báo lỗi kết nối.
 - Người dùng có thể thử lại sau.
 2. Lỗi cơ sở dữ liệu:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.

Chỉnh sửa thông tin cá nhân (Edit Profile)

- **Tên Use Case:** Chỉnh sửa thông tin cá nhân
- **Tham Chiếu (XRef):**
 - Backend: edit_user_profile view in user_profile/views.py
- **Kích Hoạt (Trigger):** Người dùng nhấp nút "Edit" trên giao diện thông tin cá nhân.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã đăng nhập vào hệ thống.
- **Luồng Chính (Basic Path):**
 1. Người dùng yêu cầu chỉnh sửa thông tin cá nhân.
 2. Hệ thống hiển thị form chỉnh sửa thông tin cá nhân với các thông tin hiện tại.
 3. Người dùng chỉnh sửa thông tin và nhấp nút "Save".
 4. Hệ thống kiểm tra tính hợp lệ của thông tin đã chỉnh sửa.
 5. Nếu hợp lệ, hệ thống cập nhật thông tin người dùng trong cơ sở dữ liệu.
 6. Hệ thống hiển thị thông báo thành công và hiển thị lại thông tin cá nhân đã được cập nhật.
- **Luồng Phụ (Alternative Paths):**
 1. Thông tin không hợp lệ:
 - Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng chỉnh sửa lại thông tin.
- **Hậu Điều Kiện (Postcondition):** Thông tin cá nhân của người dùng đã được cập nhật trong hệ thống.
- **Đường Dẫn Ngoại Lệ (Exception Paths):**
 1. Lỗi kết nối:
 - Hệ thống hiển thị thông báo lỗi kết nối.
 - Người dùng có thể thử lại sau.
 2. Lỗi cơ sở dữ liệu:
 - Hệ thống hiển thị thông báo lỗi chung.
 - Người dùng có thể thử lại sau.

3.2.5 Quản lý danh sách phim yêu thích (Watchlist Management)

- **Tên Use Case:** Quản lý danh sách phim yêu thích
- **Tham Chiếu (XRef):**
 - Backend:
 - * add_to_watchlist view in user_profile/views.py
 - * remove_from_watchlist view in user_profile/views.py
- **Kích Hoạt (Trigger):** Người dùng nhấn nút "Add to Watchlist" hoặc "Remove from Watchlist" trên trang chi tiết phim hoặc trong danh sách yêu thích.
- **Điều Kiện Tiên Quyết (Precondition):** Người dùng đã đăng nhập vào hệ thống.
- **Luồng Chính (Basic Path):**
 1. Người dùng chọn thêm hoặc xóa phim khỏi danh sách yêu thích.
 2. Hệ thống truy xuất thông tin phim từ cơ sở dữ liệu.
 3. Nếu người dùng chọn "Add to Watchlist", hệ thống thêm phim vào danh sách yêu thích của người dùng trong cơ sở dữ liệu.
 4. Nếu người dùng chọn "Remove from Watchlist", hệ thống xóa phim khỏi danh sách yêu thích của người dùng trong cơ sở dữ liệu.
 5. Hệ thống hiển thị thông báo thành công và cập nhật danh sách yêu thích.
- **Luồng Phụ (Alternative Paths):**
 1. Phim đã có trong danh sách yêu thích (khi thêm):
 - Hệ thống hiển thị thông báo rằng phim đã có trong danh sách.
 2. Phim không có trong danh sách yêu thích (khi xóa):
 - Hệ thống hiển thị thông báo rằng phim không có trong danh sách.
- **Hậu Điều Kiện (Postcondition):** Danh sách yêu thích của người dùng đã được cập nhật trong hệ thống.

3.2.6 Browse Movies (Duyệt Danh Sách Phim)

- **Tên Use Case:** Duyệt danh sách phim
- **Tham Chiếu (XRef):**
 - Backend:
 - * movie_list views trong movie/views.py
 - * popular_movies views trong movie/views.py
 - * tv_series views trong movie/views.py
 - * new_movies views trong movie/views.py
 - Frontend:
 - * movie_list.html trong movie/templates/movie
 - * most_popular.html trong movie/templates/movie
 - * tv_series.html trong movie/templates/movie
 - * new_movies.html trong movie/templates/movie
- **Kích Hoạt (Trigger):** Người dùng truy cập trang danh sách phim.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Cơ sở dữ liệu chứa các phim.
 - Người dùng đã đăng nhập (nếu cần).
- **Luồng Chính (Basic Path):**
 1. Người dùng truy cập trang danh sách phim.
 2. Người dùng chọn loại danh sách phim muốn xem (tất cả, phim chiếu rạp, phim bộ, mới).
 3. Hệ thống truy xuất thông tin các phim tương ứng từ cơ sở dữ liệu.
 4. Hệ thống hiển thị danh sách phim với phân trang nếu cần.
- **Hậu Điều Kiện (Postcondition):** Danh sách phim được hiển thị cho người dùng.

3.2.7 View Movie Details (Xem Chi Tiết Phim)

- **Tên Use Case:** Xem chi tiết phim
- **Tham Chiếu (XRef):**
 - Backend: `movie_detail` views trong `movie/views.py`
 - Frontend: `movie_detail.html` trong `movie/templates/movie`
- **Kích Hoạt (Trigger):** Người dùng nhấp vào tên phim từ danh sách phim.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Người dùng đã đăng nhập.
 - Phim tồn tại trong hệ thống.
- **Luồng Chính (Basic Path):**
 1. Người dùng nhấp vào tên phim từ danh sách.
 2. Hệ thống truy xuất thông tin chi tiết của phim từ cơ sở dữ liệu.
 3. Hệ thống hiển thị trang chi tiết phim bao gồm tiêu đề, thể loại, năm phát hành, mô tả, và hình ảnh.
- **Luồng Phụ (Alternative Paths):**
 - Phim yêu cầu đăng ký:
 1. Hệ thống kiểm tra đăng ký của người dùng.
 2. Nếu người dùng chưa đăng ký hoặc đăng ký đã hết hạn, hệ thống chuyển hướng người dùng đến trang đăng ký.
- **Hậu Điều Kiện (Postcondition):** Chi tiết phim được hiển thị cho người dùng.

3.2.8 Movie Management

- **Actor:** Admin
- **Mô tả:** Quản lý danh sách phim bao gồm việc thêm, cập nhật, xóa, và hiển thị thông tin phim.
- **Tham Chiếu (XRef):**
 - Backend: `movie_create`, `movie_update`, `movie_delete` views trong `movie/views.py`

Create Movie (Tạo Phim Mới)

- **Tên Use Case:** Tạo phim mới
- **Kích Hoạt (Trigger):** Admin nhấp vào nút "Tạo phim mới".
- **Điều Kiện Tiên Quyết (Precondition):**
 - Admin đã đăng nhập vào hệ thống.
- **Luồng Chính (Basic Path):**
 1. Admin truy cập trang tạo phim mới.
 2. Admin nhập thông tin chi tiết của phim (tiêu đề, thể loại, năm phát hành, mô tả, hình ảnh, v.v.).
 3. Hệ thống lưu thông tin phim vào cơ sở dữ liệu.
 4. Hệ thống chuyển hướng đến trang danh sách phim và hiển thị phim mới tạo.
- **Hậu Điều Kiện (Postcondition):** Phim mới được thêm vào cơ sở dữ liệu và hiển thị trong danh sách phim.

Update Movie (Cập Nhật Phim)

- **Tên Use Case:** Cập nhật phim
- **Kích Hoạt (Trigger):** Admin nhấp vào nút "Chỉnh sửa" từ trang chi tiết phim hoặc danh sách phim.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Admin đã đăng nhập.
 - Phim tồn tại trong hệ thống.
- **Luồng Chính (Basic Path):**
 1. Admin truy cập trang chỉnh sửa phim.
 2. Admin chỉnh sửa thông tin phim.
 3. Hệ thống lưu thay đổi vào cơ sở dữ liệu.
 4. Hệ thống hiển thị thông báo thành công và chuyển hướng đến trang danh sách phim.
- **Hậu Điều Kiện (Postcondition):** Thông tin phim được cập nhật trong cơ sở dữ liệu.

Delete Movie (Xóa Phim)

- **Tên Use Case:** Xóa phim
- **Kích Hoạt (Trigger):** Admin nhấp vào nút "Xóa" từ trang chi tiết phim hoặc danh sách phim.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Admin đã đăng nhập.
 - Phim tồn tại trong hệ thống.
- **Luồng Chính (Basic Path):**
 1. Admin xác nhận yêu cầu xóa phim.
 2. Hệ thống xóa phim khỏi cơ sở dữ liệu.
 3. Hệ thống hiển thị thông báo thành công và cập nhật danh sách phim.
- **Luồng Phụ (Alternative Paths):**
 - Lỗi trong quá trình xóa:
 1. Hệ thống hiển thị thông báo lỗi và không xóa phim.
- **Hậu Điều Kiện (Postcondition):** Phim bị xóa khỏi cơ sở dữ liệu.

3.2.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)

- **Tên Use Case:** Hiển thị danh sách gói đăng ký
- **Tham Chiếu (XRef):**
 - Backend: `subscriptions_list` view trong `subscription/views.py`
 - Frontend: `subscriptions_list.html` trong `subscription/templates/subscription`
- **Kích Hoạt (Trigger):** Người dùng truy cập trang danh sách gói đăng ký.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Cơ sở dữ liệu chứa các gói đăng ký.
- **Luồng Chính (Basic Path):**
 1. Người dùng truy cập trang danh sách gói đăng ký.
 2. Hệ thống truy xuất thông tin các gói đăng ký từ cơ sở dữ liệu.
 3. Hệ thống hiển thị danh sách gói đăng ký cho người dùng.
- **Hậu Điều Kiện (Postcondition):** Danh sách gói đăng ký được hiển thị cho người dùng.

3.2.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)

- **Tên Use Case:** Hiển thị chi tiết gói đăng ký
- **Tham Chiếu (XRef):**
 - Backend: subscription_detail view trong subscription/views.py
 - Frontend: subscription_detail.html trong subscription/templates/subscription
- **Kích Hoạt (Trigger):** Người dùng chọn một gói đăng ký từ danh sách.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Người dùng đã đăng nhập.
 - Gói đăng ký tồn tại trong cơ sở dữ liệu.
- **Luồng Chính (Basic Path):**
 1. Người dùng chọn một gói đăng ký từ danh sách.
 2. Hệ thống truy xuất thông tin chi tiết của gói đăng ký từ cơ sở dữ liệu.
 3. Hệ thống hiển thị chi tiết gói đăng ký cho người dùng, bao gồm giá cả và các lợi ích.
- **Hậu Điều Kiện (Postcondition):** Chi tiết gói đăng ký được hiển thị cho người dùng.

3.2.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)

- **Tên Use Case:** Thanh toán gói đăng ký
- **Tham Chiếu (XRef):**
 - Backend: checkout view trong subscription/views.py
 - Frontend: checkout.html trong subscription/templates/subscription
- **Kích Hoạt (Trigger):** Người dùng chọn mua một gói đăng ký.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Người dùng đã đăng nhập.
 - Gói đăng ký tồn tại và đang hoạt động.
- **Luồng Chính (Basic Path):**
 1. Người dùng chọn mua một gói đăng ký và chọn khoảng thời gian (tháng, nửa năm, hoặc cả năm).
 2. Hệ thống tính toán giá dựa trên thời gian đã chọn và tạo UserSubscription.
 3. Hệ thống thực hiện thanh toán qua PayPal hoặc cổng thanh toán khác.
 4. Sau khi thanh toán thành công, hệ thống cập nhật thông tin đăng ký cho người dùng.
 5. Hệ thống hiển thị trang xác nhận thanh toán thành công.
- **Hậu Điều Kiện (Postcondition):** Người dùng đã thanh toán thành công và gói đăng ký được kích hoạt.

3.2.12 Review Movie (Đánh Giá Phim)

- **Tên Use Case:** Đánh giá phim
- **Tham Chiếu (XRef):**
 - Backend: Review model trong review/models.py, ReviewForm trong review/forms.py
 - Frontend: review.html trong review/templates/review
- **Kích Hoạt (Trigger):** Người dùng chọn đánh giá một bộ phim.
- **Điều Kiện Tiên Quyết (Precondition):**
 - Người dùng đã đăng nhập.

- Phim tồn tại trong cơ sở dữ liệu.
- **Luồng Chính (Basic Path):**
 1. Người dùng truy cập trang đánh giá phim và chọn phim để đánh giá.
 2. Hệ thống hiển thị form để người dùng nhập đánh giá và chấm điểm cho phim.
 3. Người dùng điền thông tin đánh giá và chấm điểm, sau đó gửi form.
 4. Hệ thống lưu lại đánh giá của người dùng vào cơ sở dữ liệu.
- **Hậu Điều Kiện (Postcondition):** Đánh giá của người dùng được lưu lại và hiển thị trên trang phim.

Chapter 4

Kiến Trúc Dự Án MotChill

4.1 Kiến Trúc Tổng Quan

- Dựa trên Django MVC (Model-View-Controller) framework.

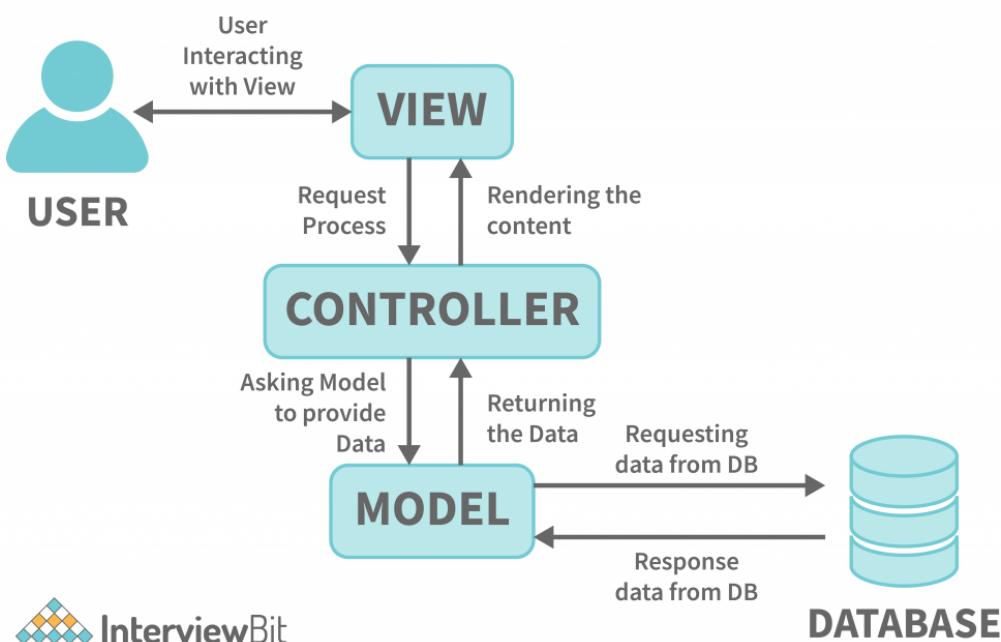


Figure 4.1: Kiến trúc MVC

- Bao gồm các ứng dụng con (apps): `movie`, `user_profile`, `review`, và `subscription`.

4.2 Các Thành Phần Chính

- Django Framework:** Nền tảng cốt lõi cung cấp ORM, quản lý người dùng, định tuyến URL, và quản lý giao diện.
- Apps:**
 - `movie`: Quản lý thông tin phim.
 - `user_profile`: Quản lý hồ sơ người dùng và danh sách theo dõi.
 - `review`: Quản lý đánh giá và bình luận.
 - `subscription`: Quản lý gói đăng ký và thông tin thanh toán.

4.3 Kiến Trúc Layered

- **Presentation Layer:** Giao diện người dùng (templates và views).
- **Business Logic Layer:** Logic ứng dụng (models, views, forms).
- **Data Access Layer:** Truy cập dữ liệu thông qua Django ORM.
- **Integration Layer:** Tích hợp với dịch vụ bên ngoài (ví dụ: PayPal).

4.4 Mô Hình MVC của Django

- **Model:** Định nghĩa trong `models.py` của mỗi app (ví dụ: Movie, Tag, User, Subscription).
- **View:** Xử lý logic trong `views.py` của mỗi app.
- **Template:** Giao diện người dùng trong thư mục `templates`.
- **Controller:** Kết hợp giữa View và URL routing của Django.

4.5 Quan Hệ Giữa Các Thành Phần

- UserProfile - Movie: Nhiều-nhiều (watchlist).
- User - UserProfile: Một-một.
- User - UserSubscription: Một-một.
- User - Review - Movie: Một-nhiều (User-Review và Movie-Review).

4.6 Thiết Kế Hướng Đến Khả Năng Mở Rộng

- Sử dụng các mẫu thiết kế và nguyên tắc SOLID.
- Modular hóa các chức năng thành các app riêng biệt.
- Sử dụng abstraction và interface để dễ dàng thay đổi triển khai.

Chapter 5

Biểu đồ mức phân tích

5.1 Đăng ký tài khoản (User Registration)

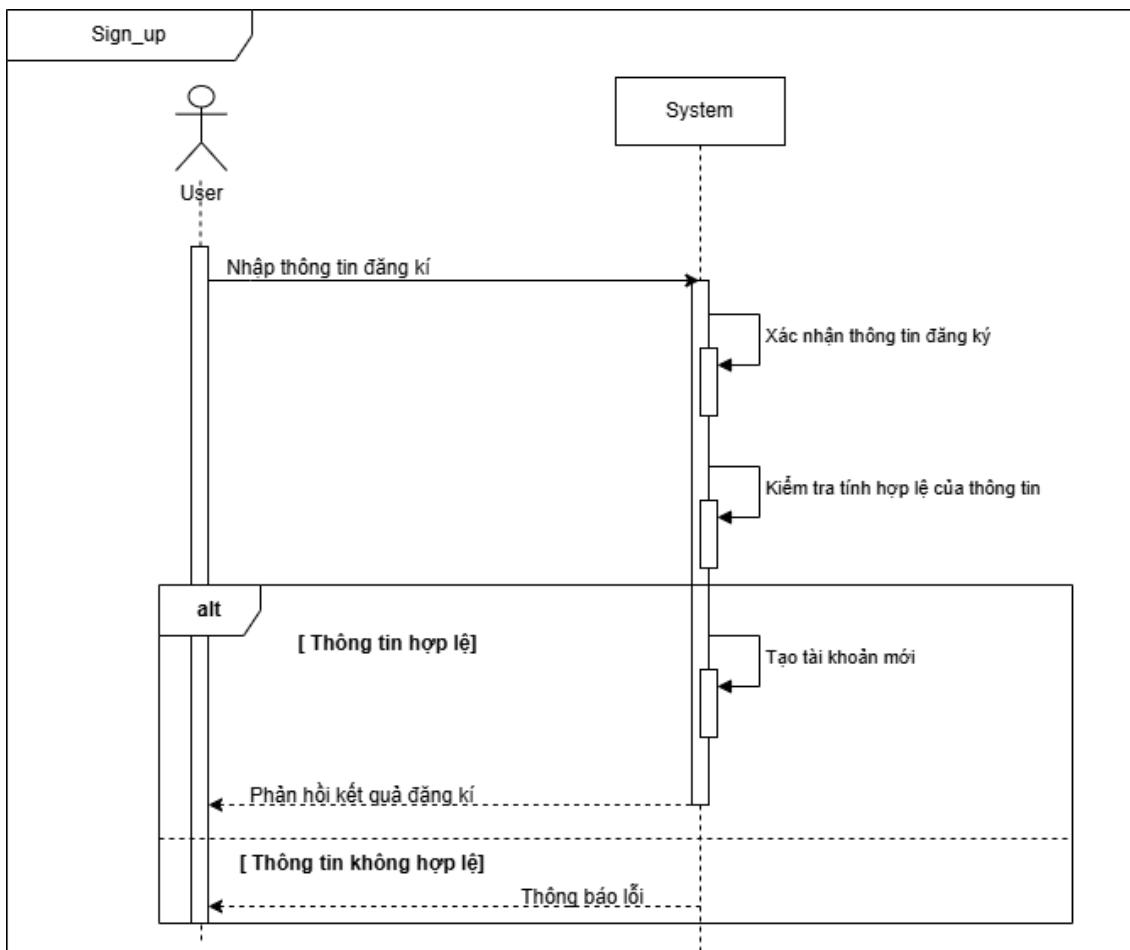


Figure 5.1: Biểu đồ tuần tự cho use case đăng ký (mức phân tích)

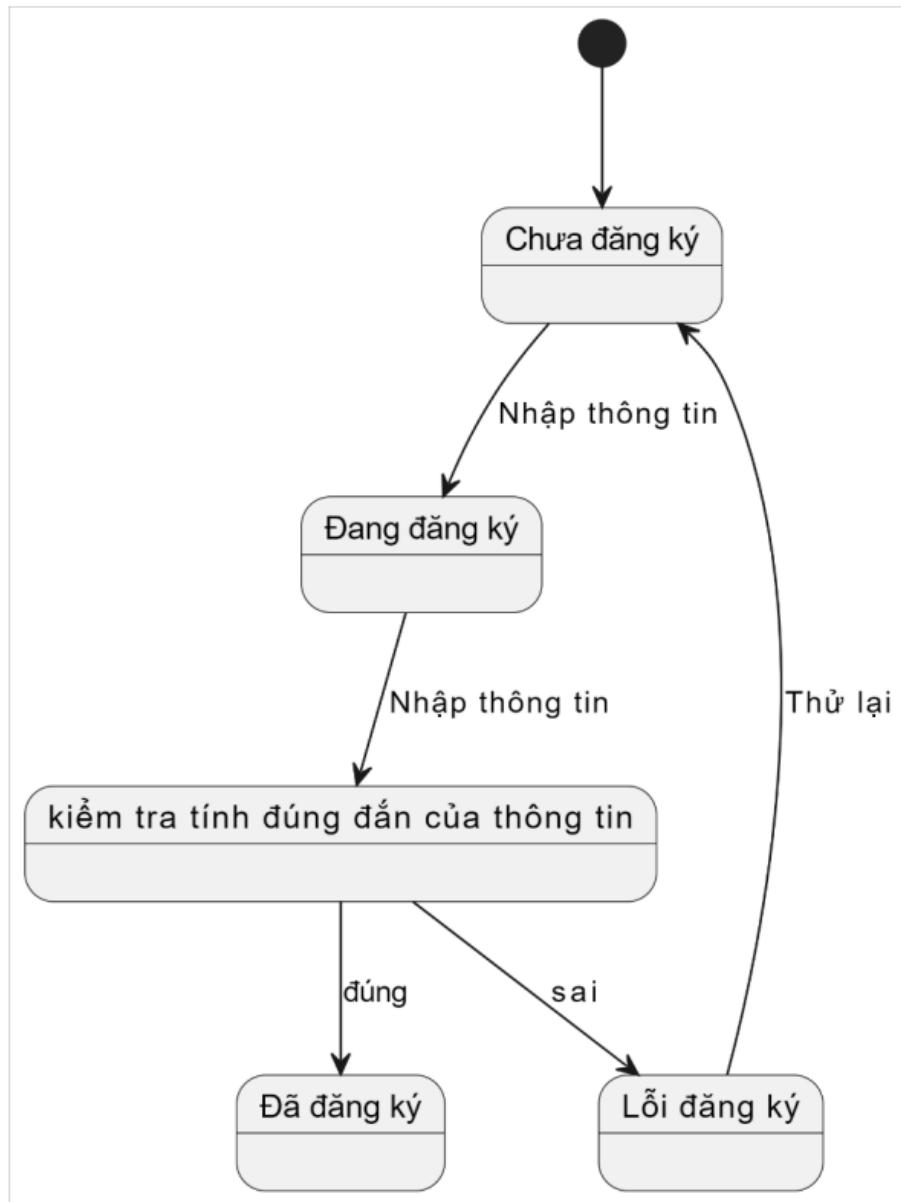


Figure 5.2: Biểu đồ trạng thái cho use case đăng ký (mức phân tích)

5.2 Đăng nhập (User Login)

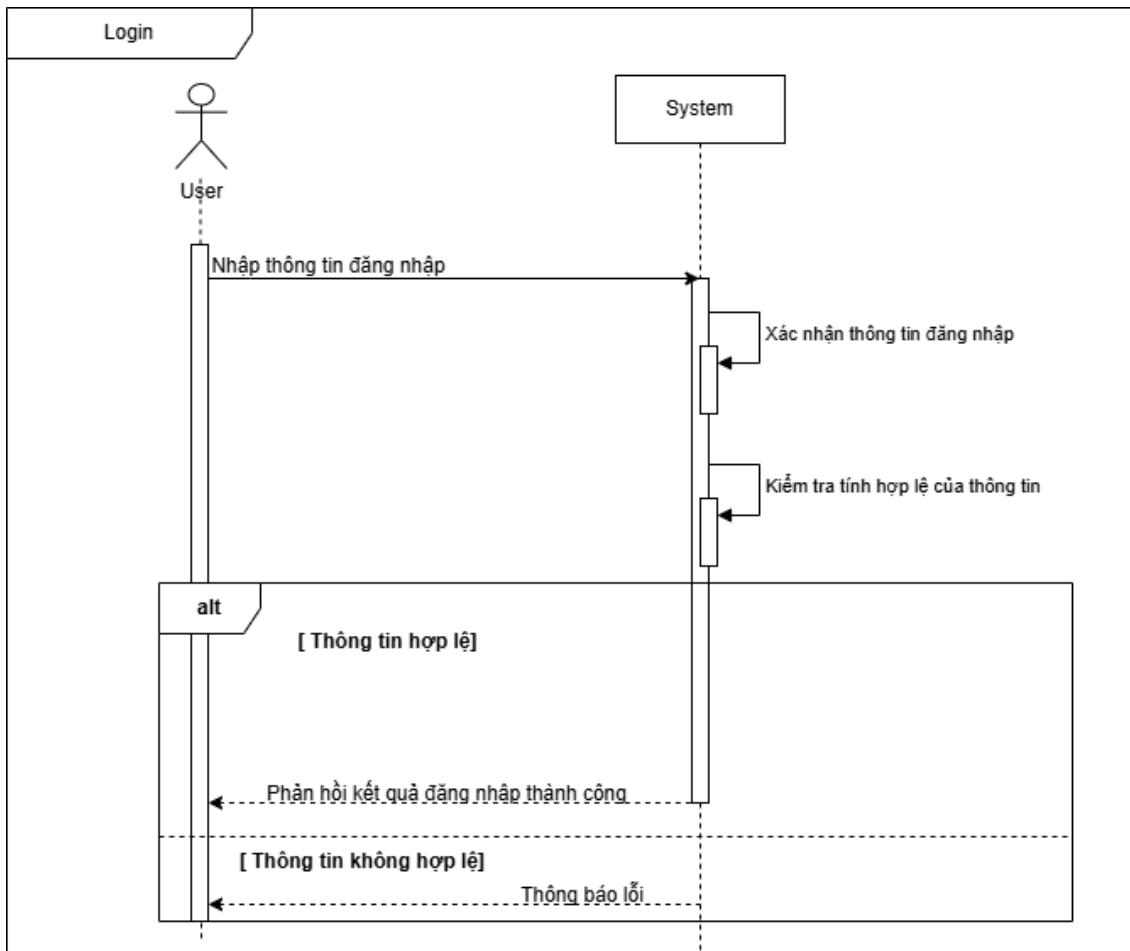


Figure 5.3: Biểu đồ tuần tự cho use case đăng nhập (mức phân tích)

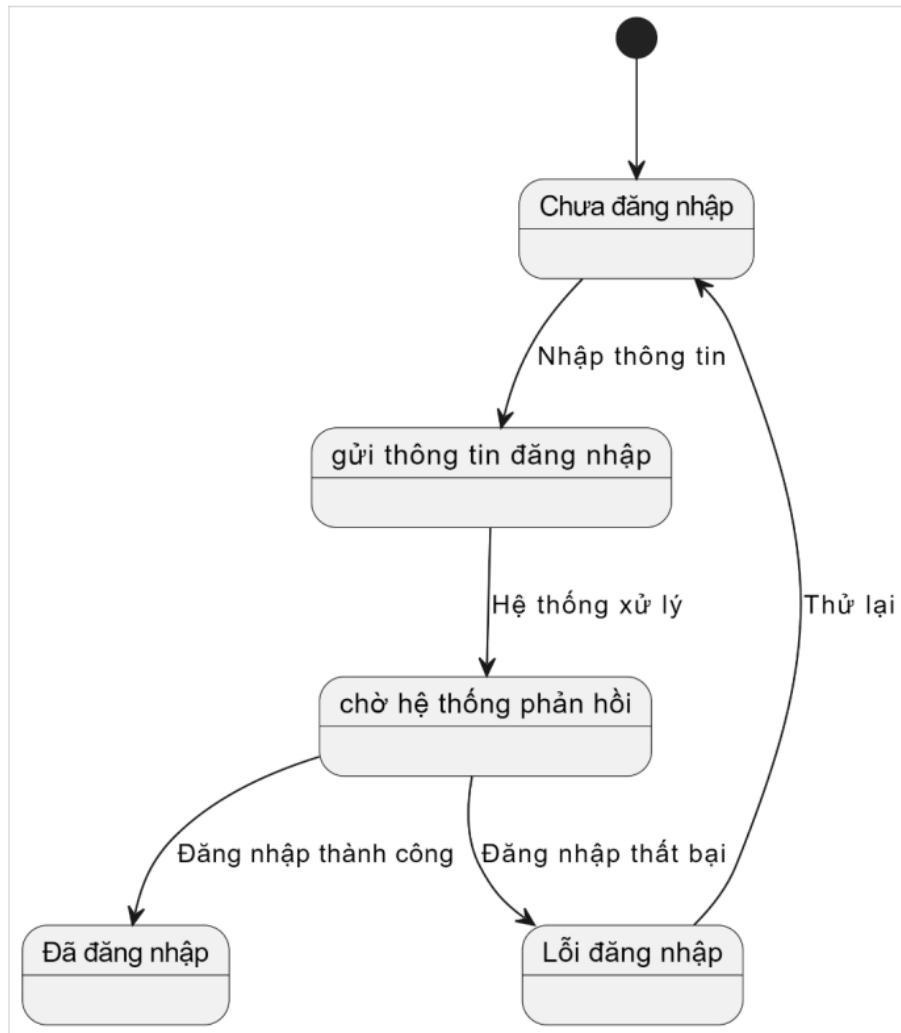


Figure 5.4: Biểu đồ trạng thái cho use case đăng nhập (mức phân tích)

5.3 Đăng xuất (User Logout)

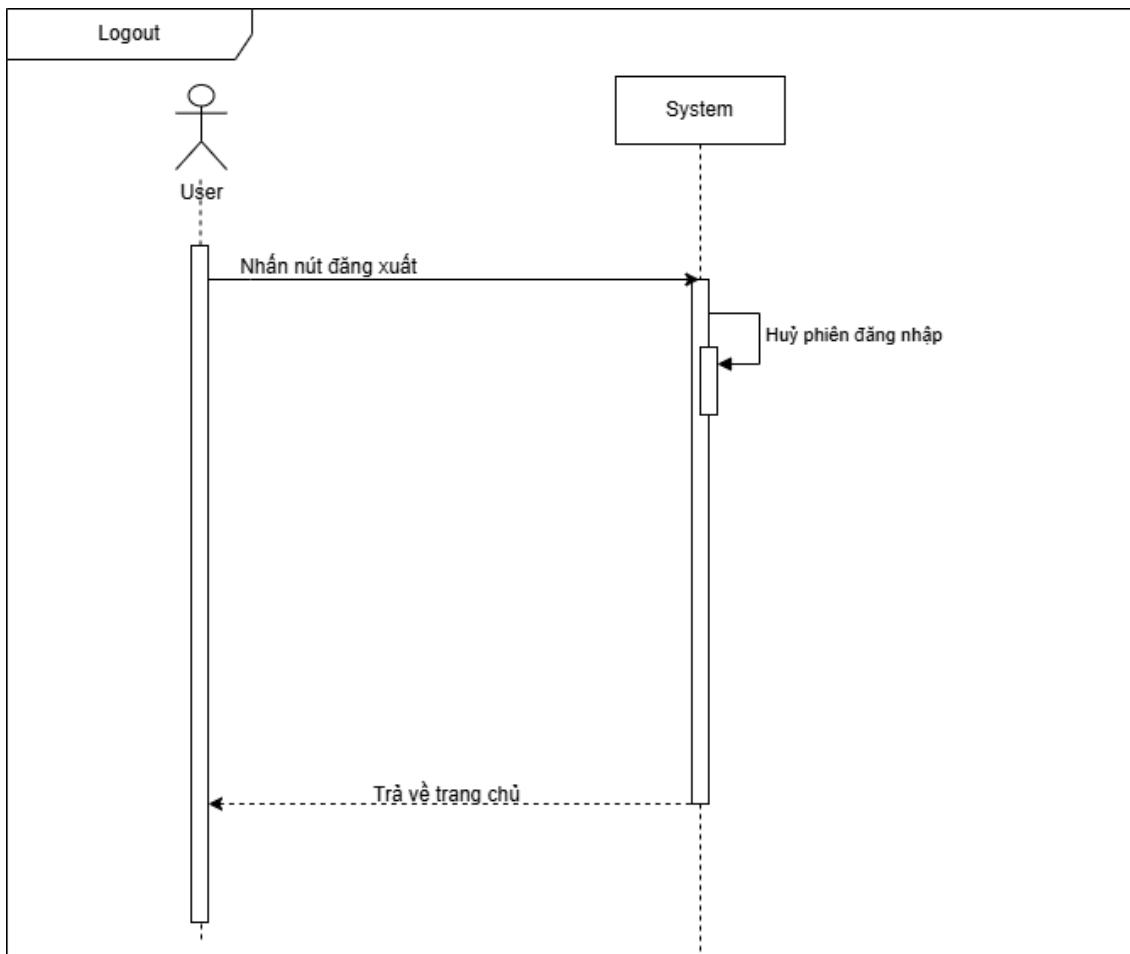


Figure 5.5: Biểu đồ tuần tự cho use case đăng xuất (mức phân tích)

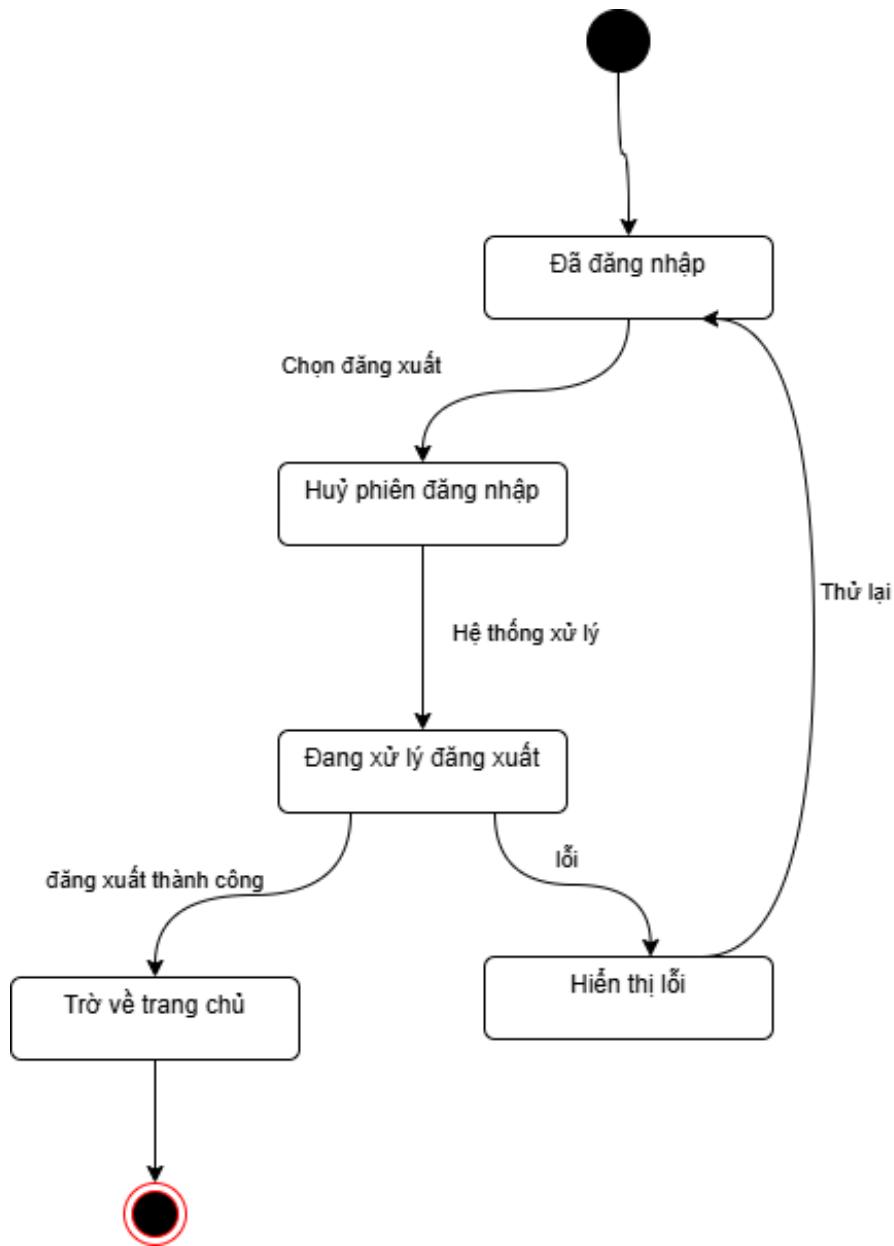


Figure 5.6: Biểu đồ trạng thái cho use case đăng xuất (mức phân tích)

5.4 Quản lý hồ sơ người dùng (User Profile Management)

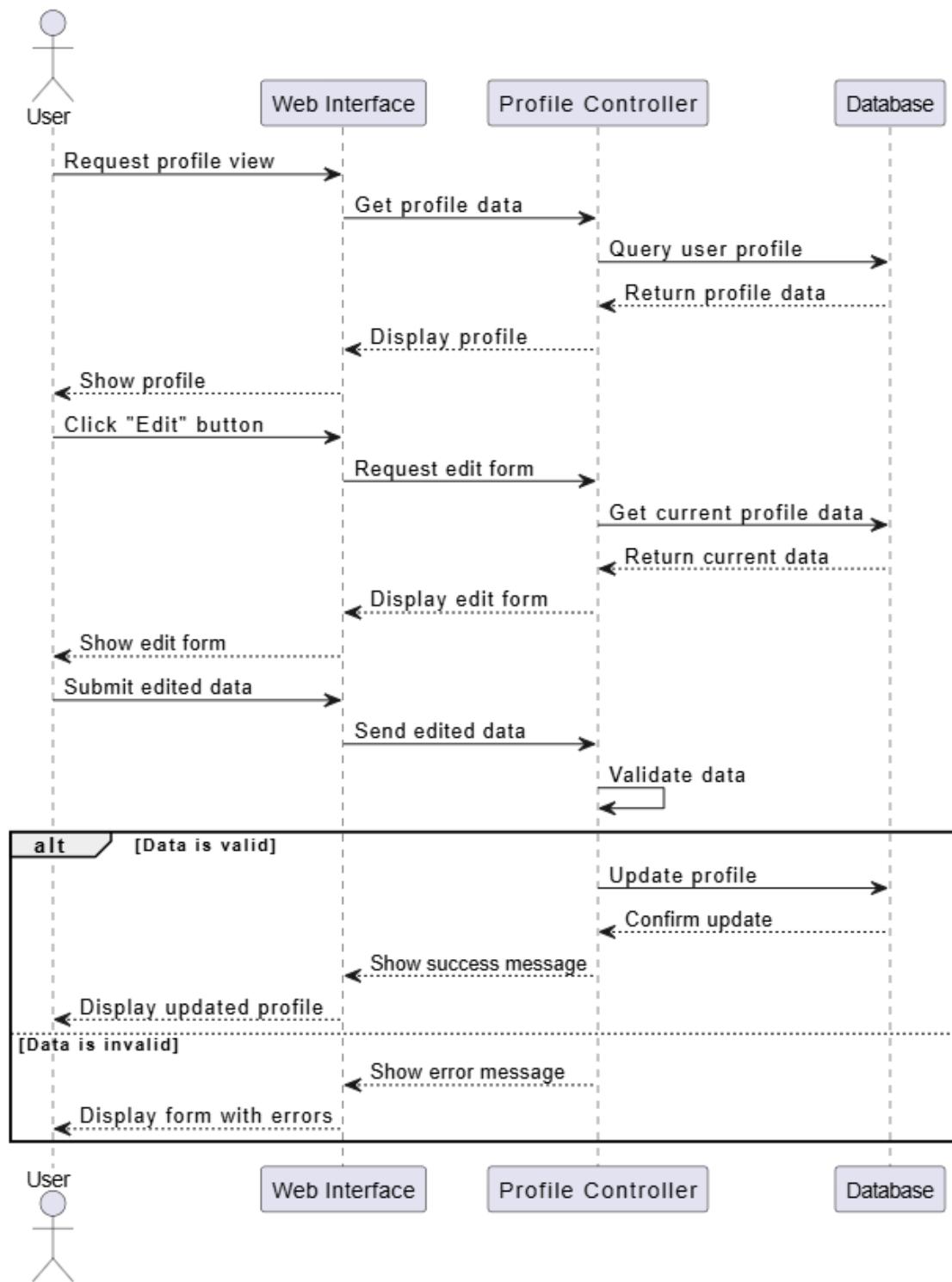


Figure 5.7: Biểu đồ tuần tự cho use case quản lý hồ sơ người dùng(bao gồm xem và sửa) (mức phân tích)

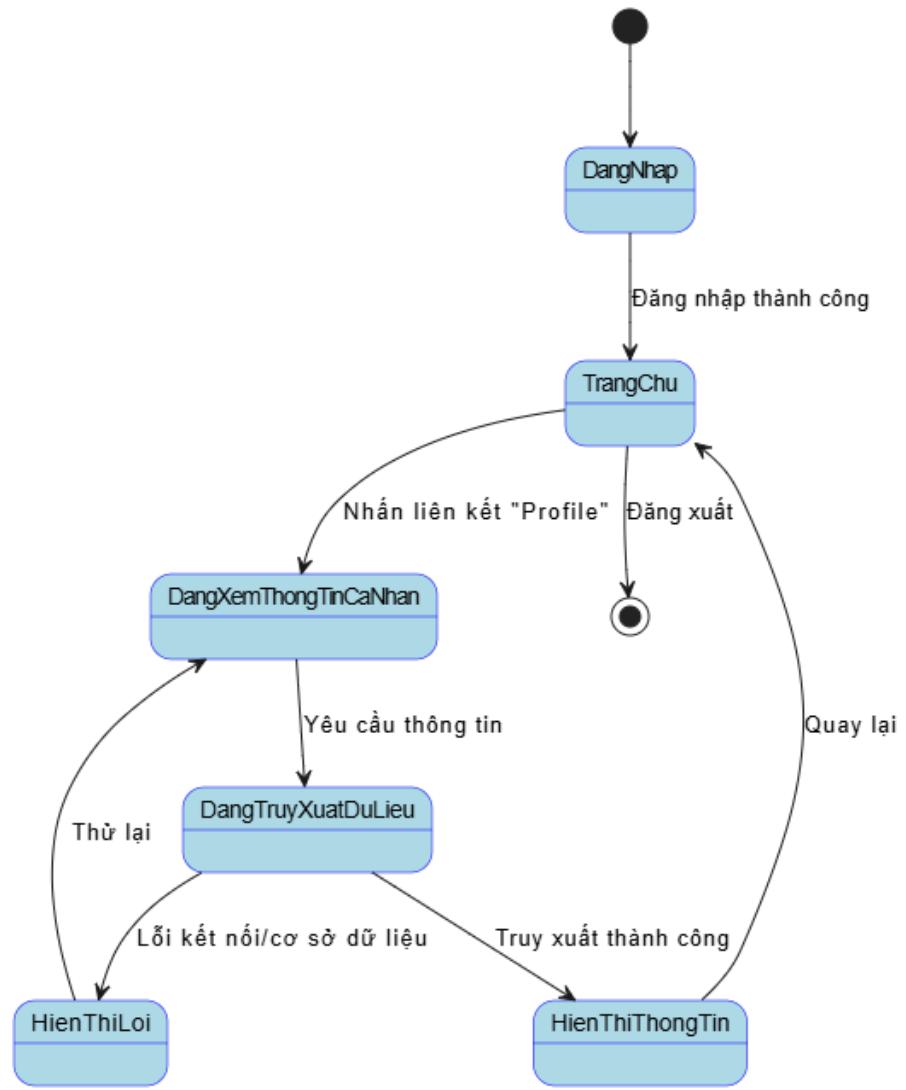


Figure 5.8: Biểu đồ trạng thái cho use case xem hồ sơ người dùng (mức phân tích)

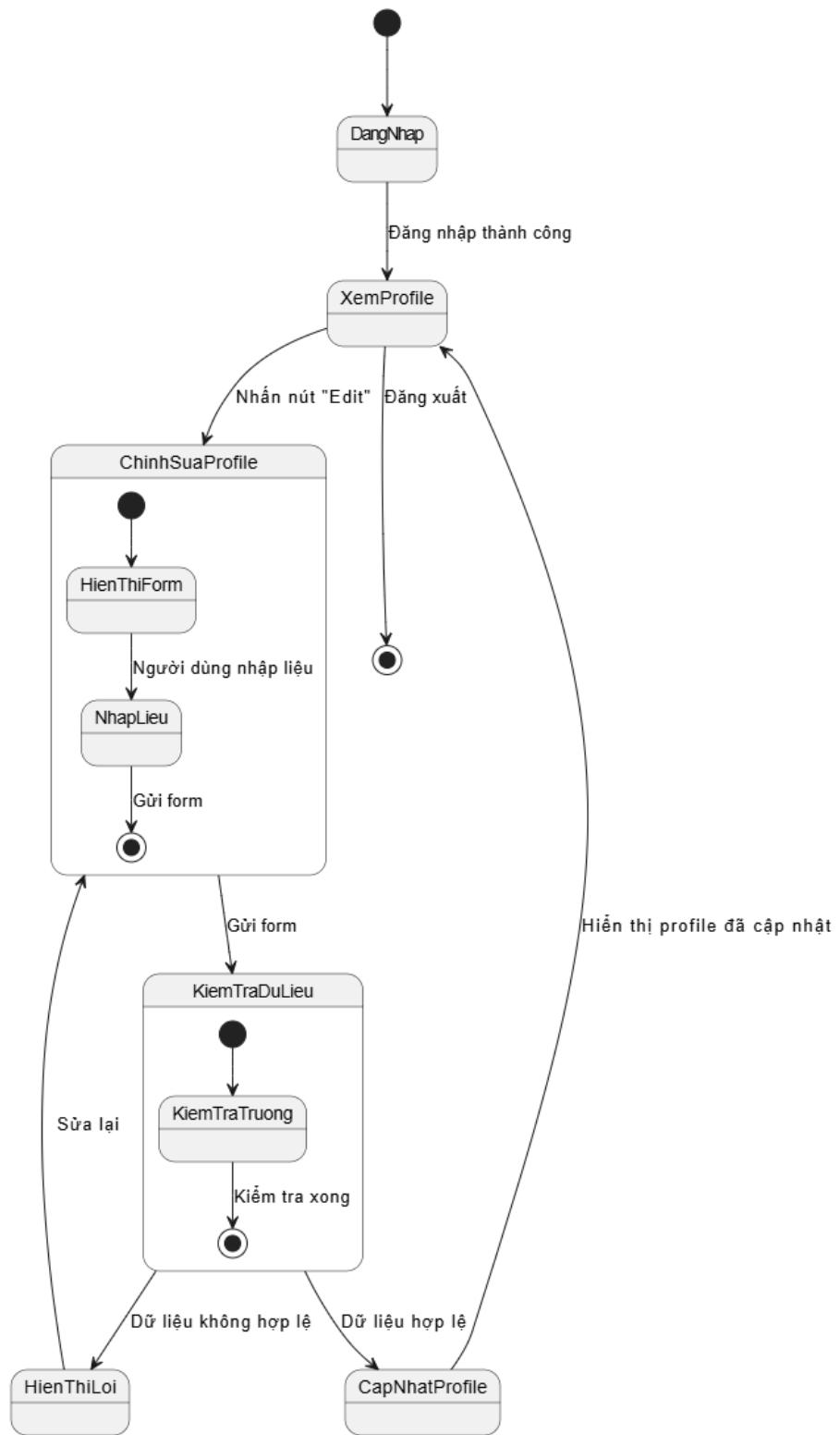


Figure 5.9: Biểu đồ trạng thái cho use case sửa hồ sơ người dùng (mức phân tích)

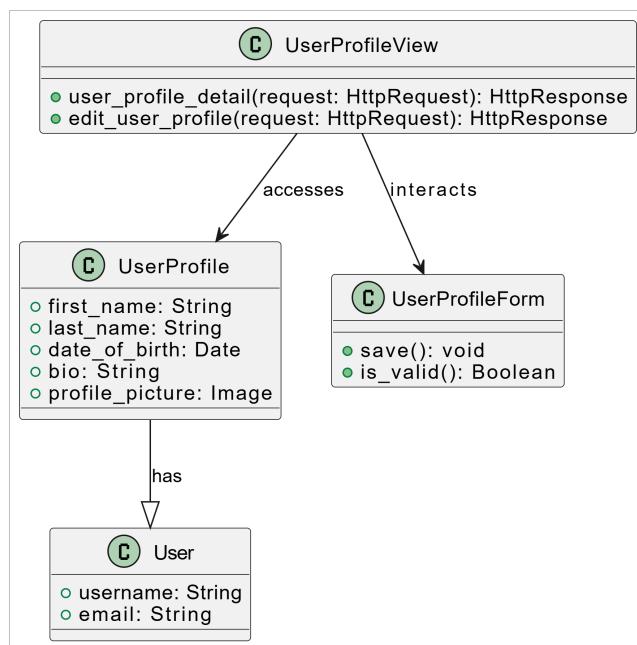


Figure 5.10: Biểu đồ lớp cho use case quản lý hồ sơ người dùng (mức phân tích)

5.5 Quản lý danh sách phim yêu thích (Watchlist Management)

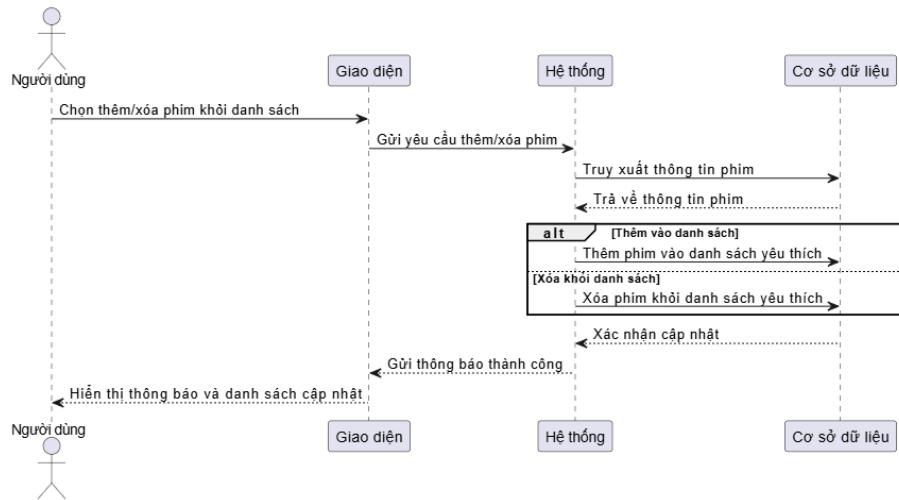


Figure 5.11: Biểu đồ tuần tự cho use case quản lý danh sách phim yêu thích (mức phân tích)

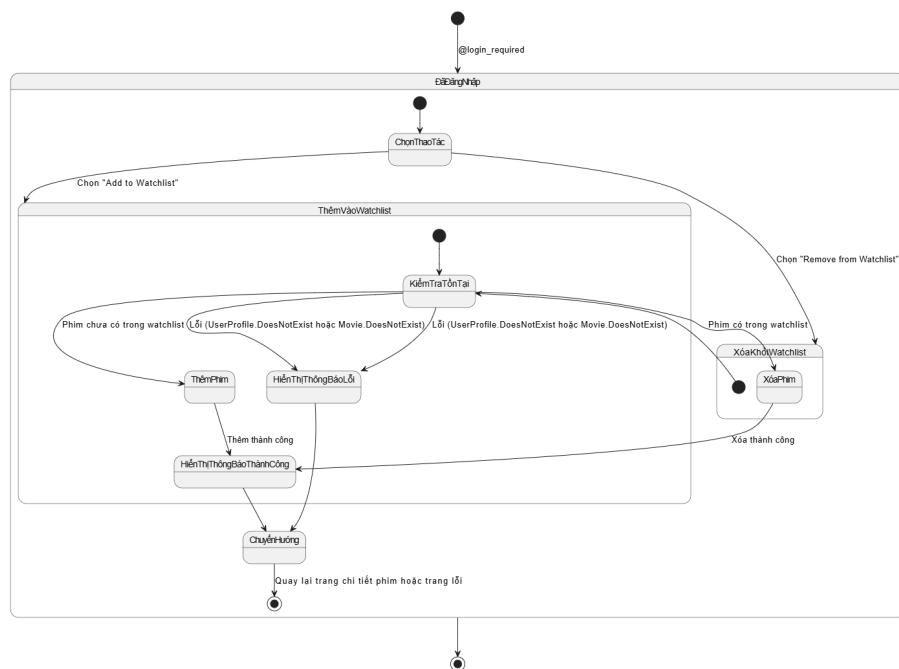


Figure 5.12: Biểu đồ trạng thái cho use case quản lý danh sách phim yêu thích (mức phân tích)

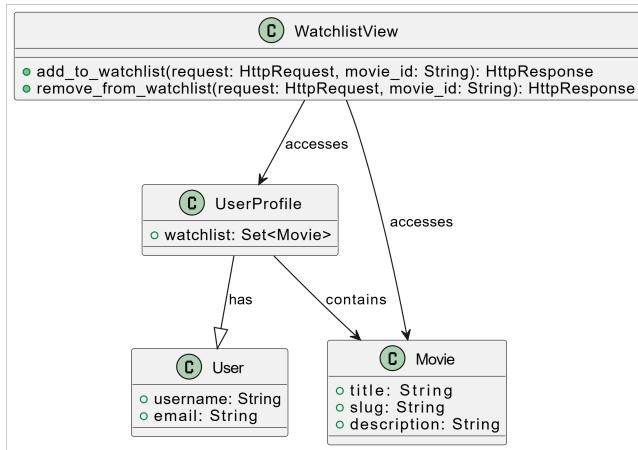


Figure 5.13: Biểu đồ lớp cho use case quản lý danh sách phim yêu thích (mức phân tích)

5.6 Browse Movies (Duyệt Danh Sách Phim)

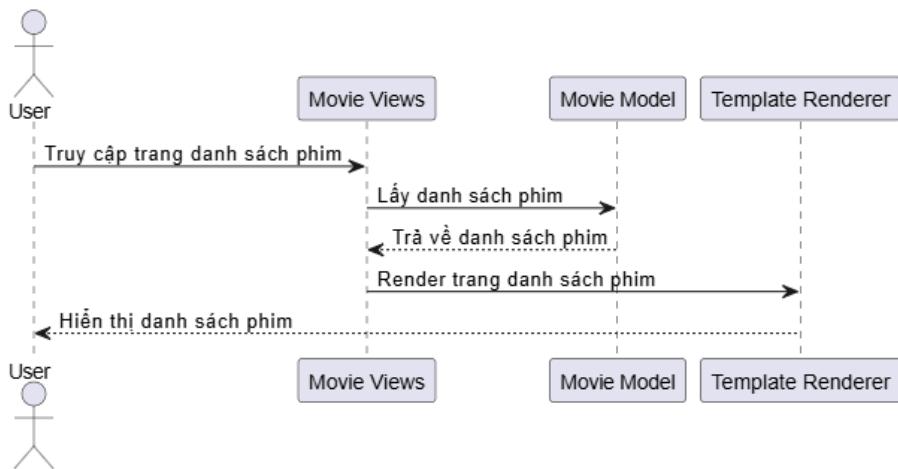


Figure 5.14: Biểu đồ tuần tự cho use case hiển thị danh sách phim (mức phân tích)



Figure 5.15: Biểu đồ trạng thái cho use case quản lý danh sách phim yêu thích (mức phân tích)

5.7 View Movie Details (Xem Chi Tiết Phim)

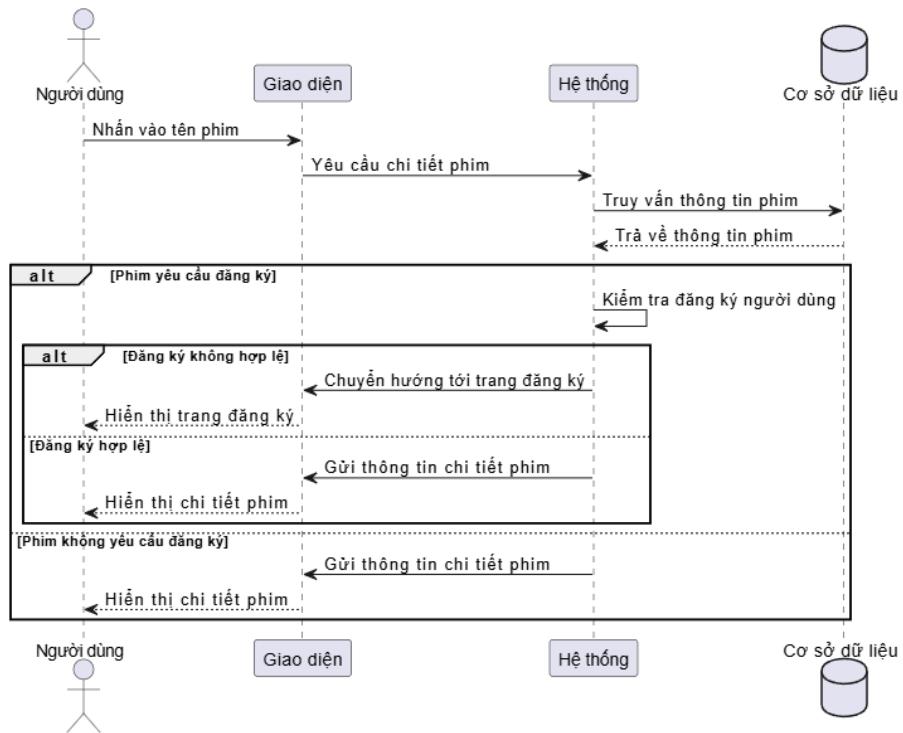


Figure 5.16: Biểu đồ tuần tự cho use case hiển thị chi tiết phim (mức phân tích)



Figure 5.17: Biểu đồ trạng thái cho use case hiển thi chi tiết phim (mức phân tích)

5.8 Movie Management

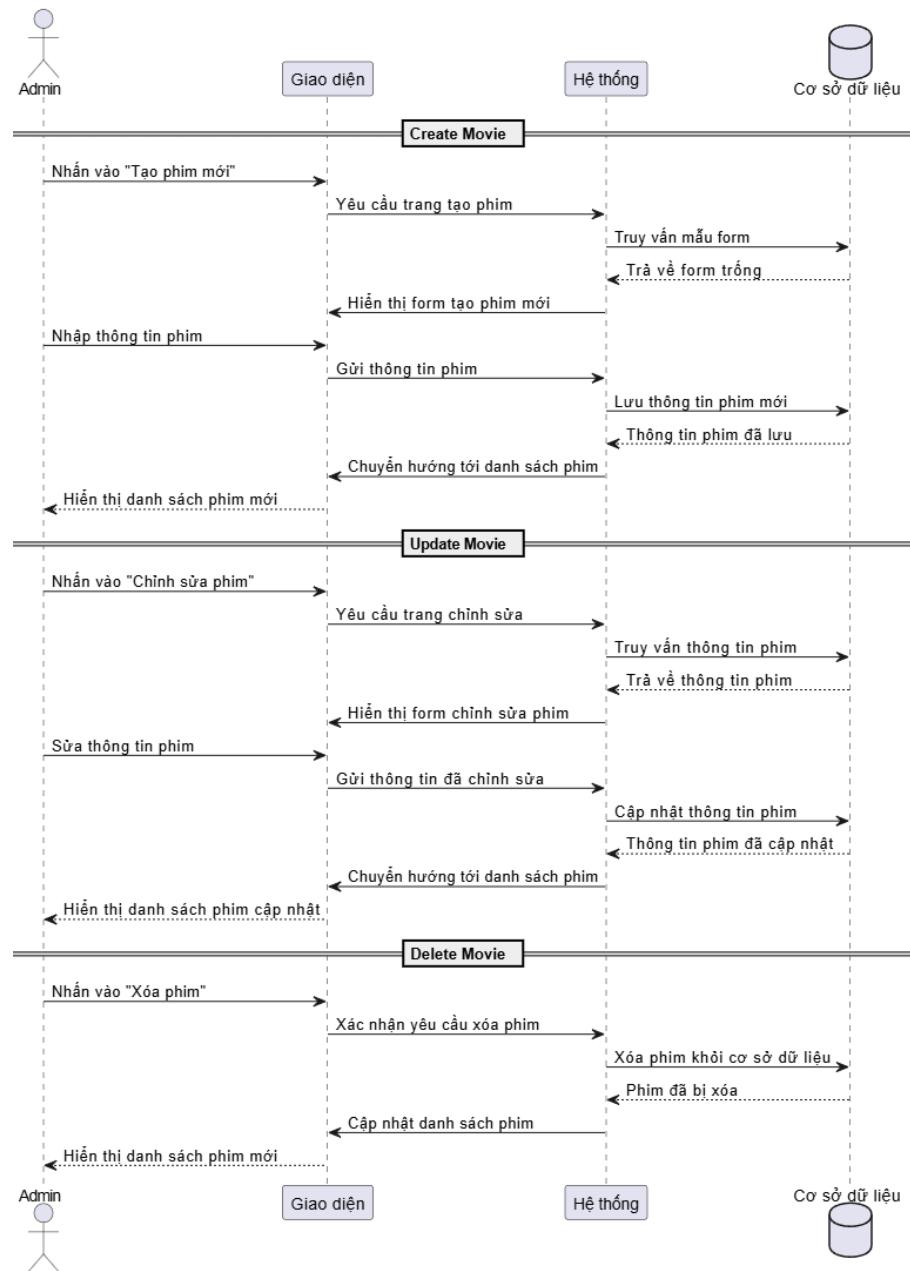


Figure 5.18: Biểu đồ tuần tự cho use case quản lý phim (mức phân tích)

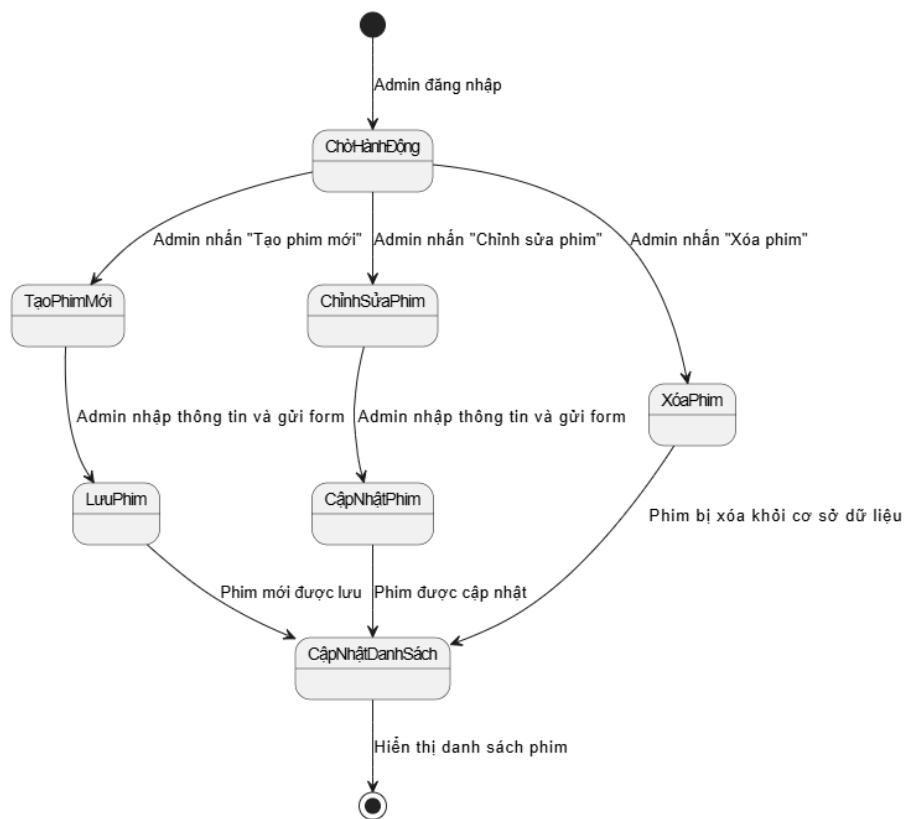


Figure 5.19: Biểu đồ trạng thái cho use case quản lý phim (mức phân tích)

5.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)

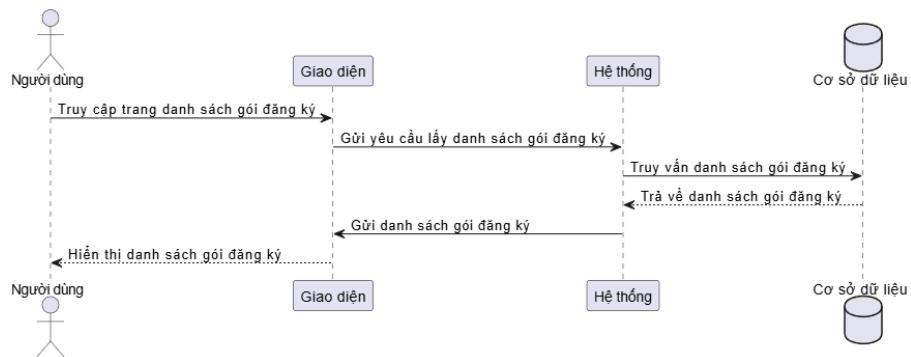


Figure 5.20: Biểu đồ tuần tự cho use case hiển thị danh sách gói đăng kí (mức phân tích)



Figure 5.21: Biểu đồ trạng thái cho use case hiển thị danh sách gói đăng kí (mức phân tích)

5.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)

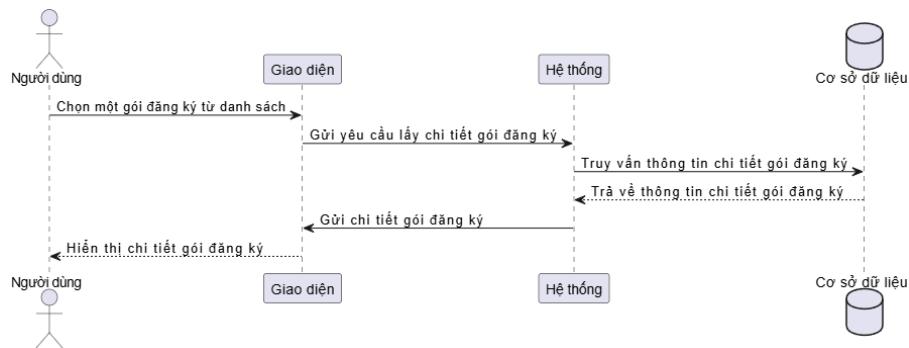


Figure 5.22: Biểu đồ tuần tự cho use case hiển thị chi tiết gói đăng kí (mức phân tích)



Figure 5.23: Biểu đồ trạng thái cho use case hiển thị chi tiết gói đăng kí (mức phân tích)

5.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)

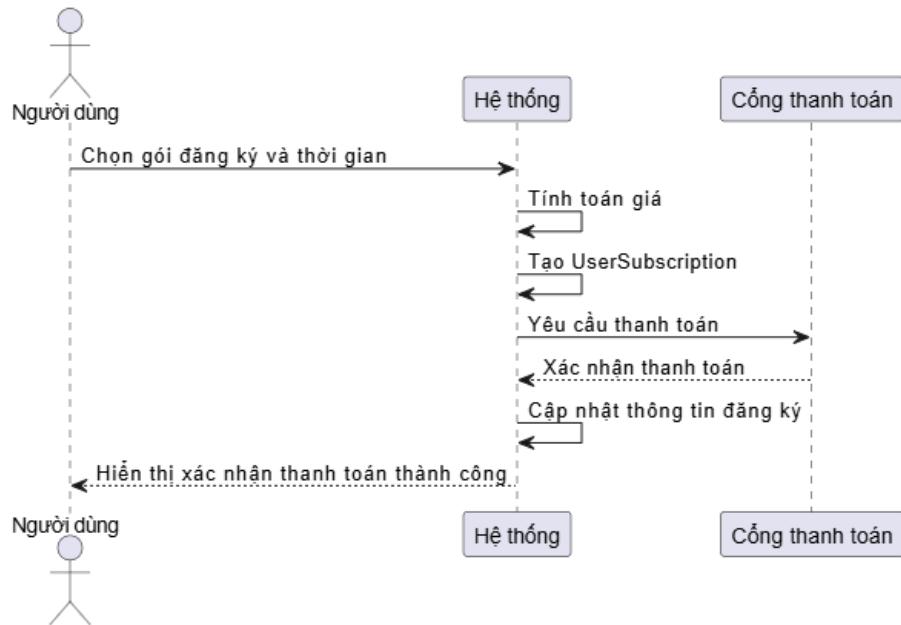


Figure 5.24: Biểu đồ tuần tự cho use case thanh toán gói đăng ký (mức phân tích)

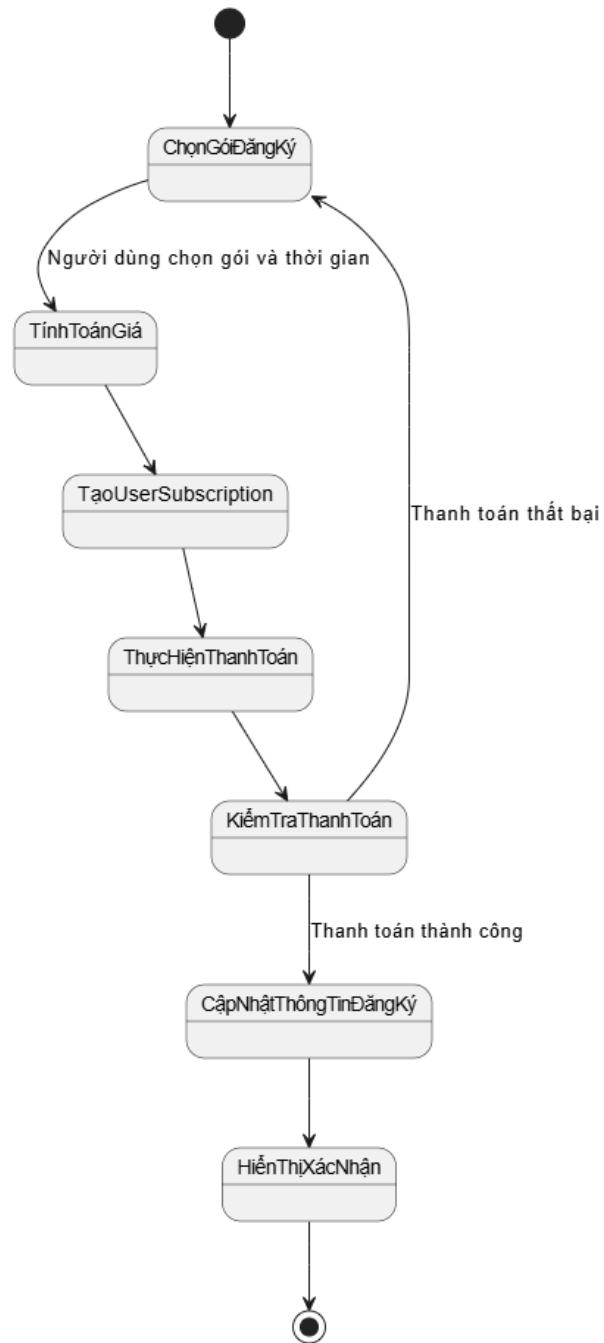


Figure 5.25: Biểu đồ trạng thái cho use case thanh toán gói đăng ký (mức phân tích)

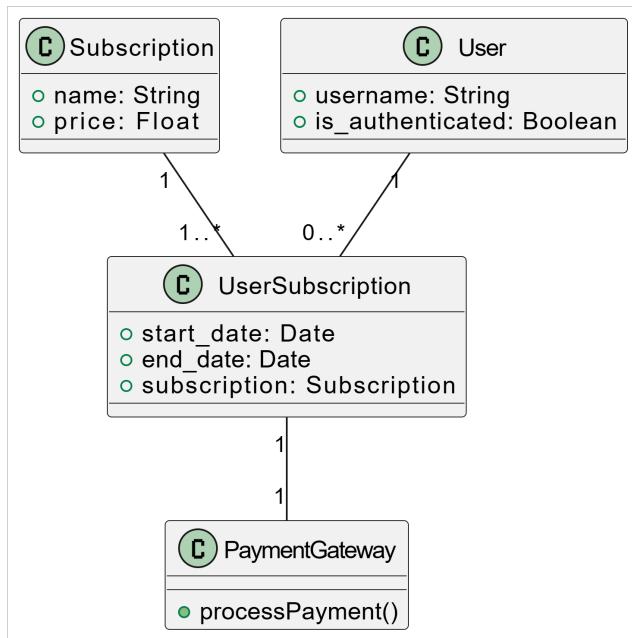


Figure 5.26: Biểu đồ lớp cho các use case `subscriptionsList`, `subscriptionDetail`, `vcheckout`

5.12 Review Movie (Đánh Giá Phim)

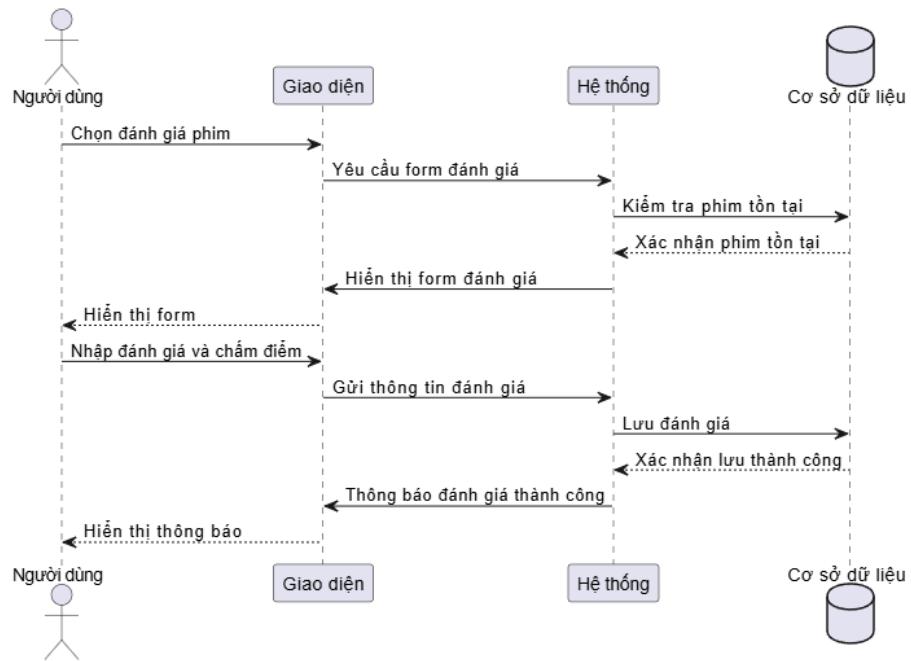


Figure 5.27: Biểu đồ tuần tự cho use case đánh giá phim (mức phân tích)

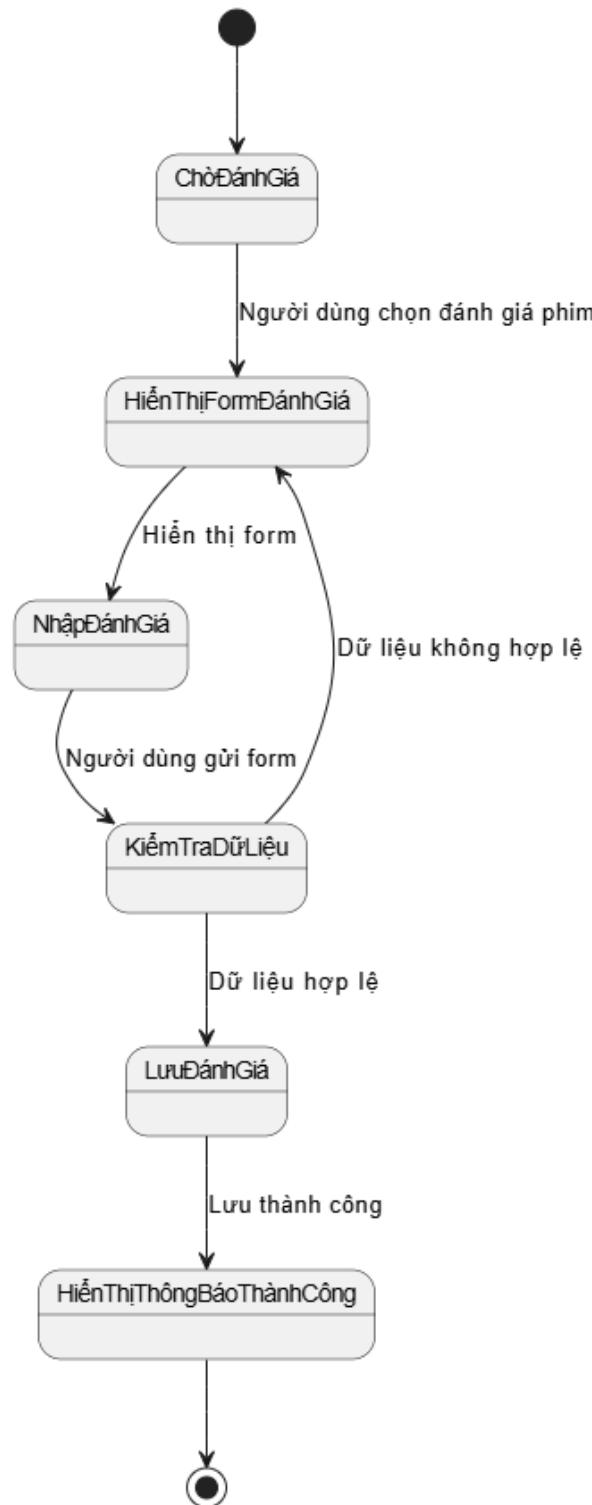


Figure 5.28: Biểu đồ trạng thái cho use case đánh giá phim (mức phân tích)

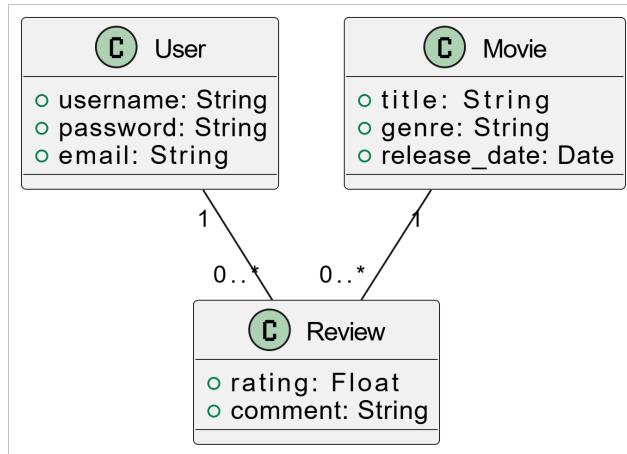


Figure 5.29: Biểu đồ lớp cho use case review

Chapter 6

Biểu đồ mức thiết kế

6.1 Đăng ký tài khoản (User Registration)

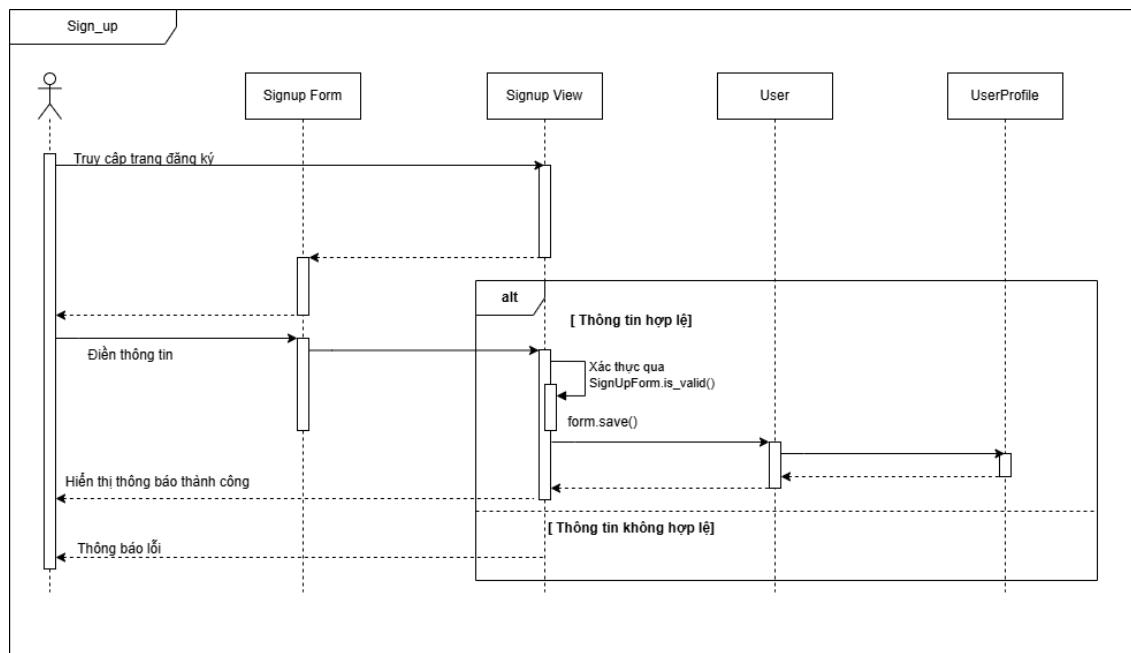


Figure 6.1: Biểu đồ tuần tự cho use case đăng ký (mức phân tích)

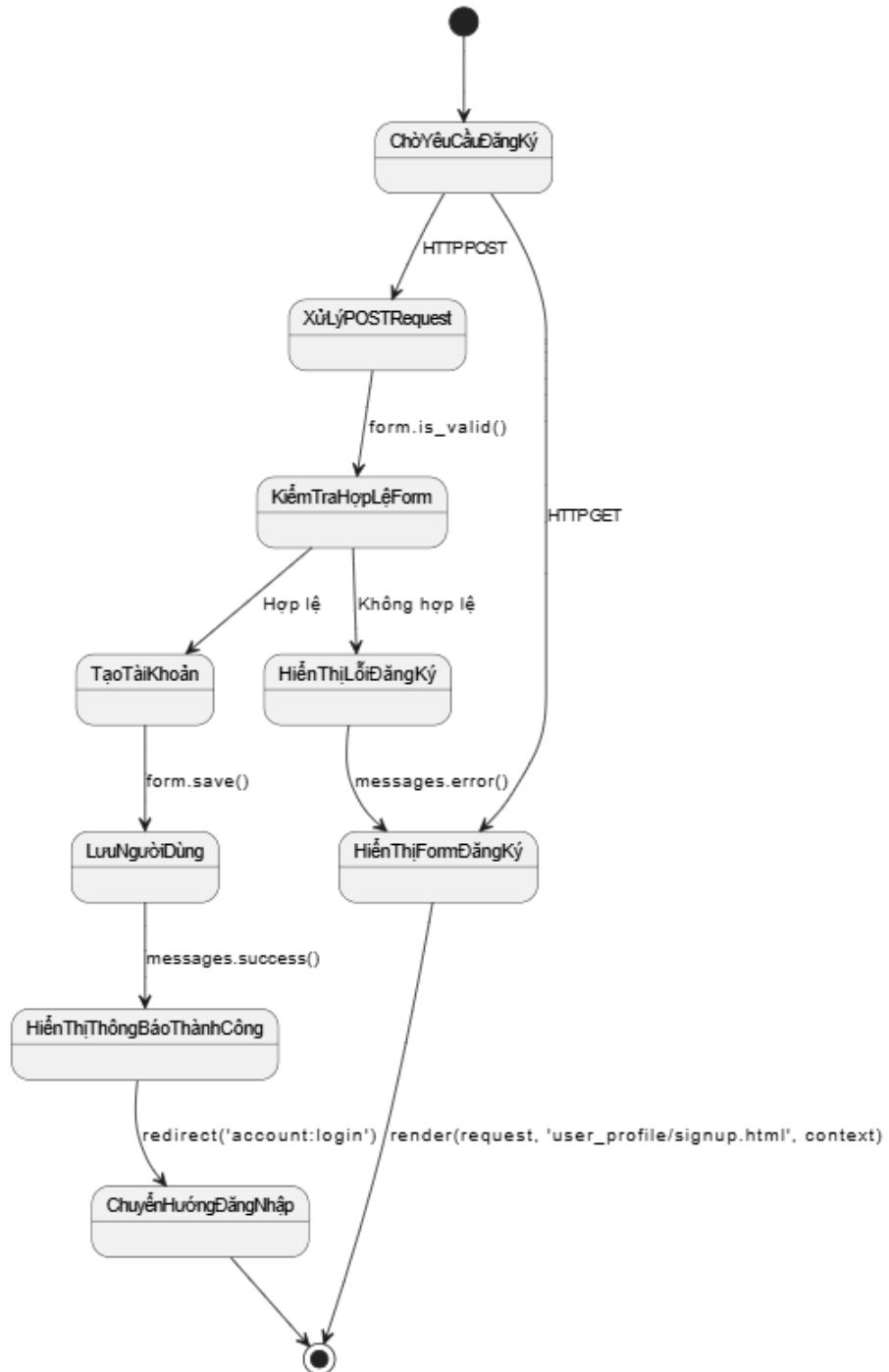


Figure 6.2: Biểu đồ trạng thái cho use case đăng ký (mức thiết kế)

6.2 Đăng nhập (User Login)

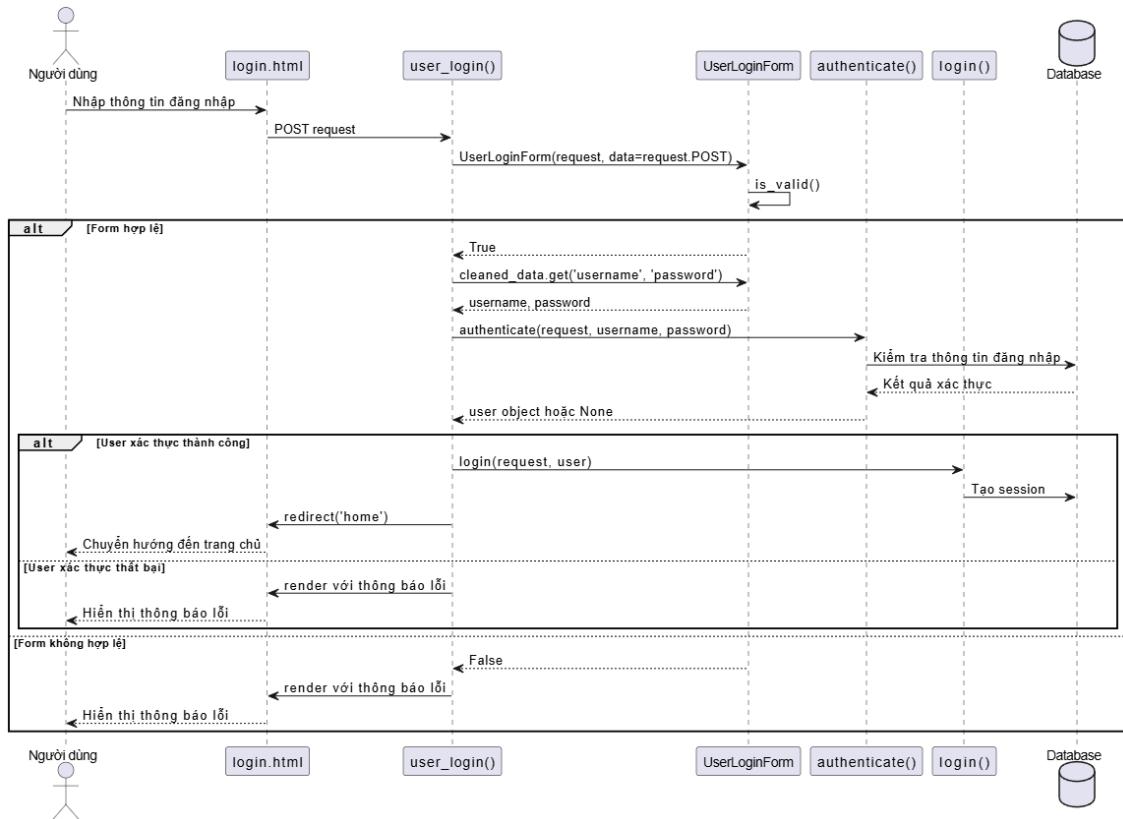


Figure 6.3: Biểu đồ tuần tự cho use case đăng nhập (mức thiết kế)

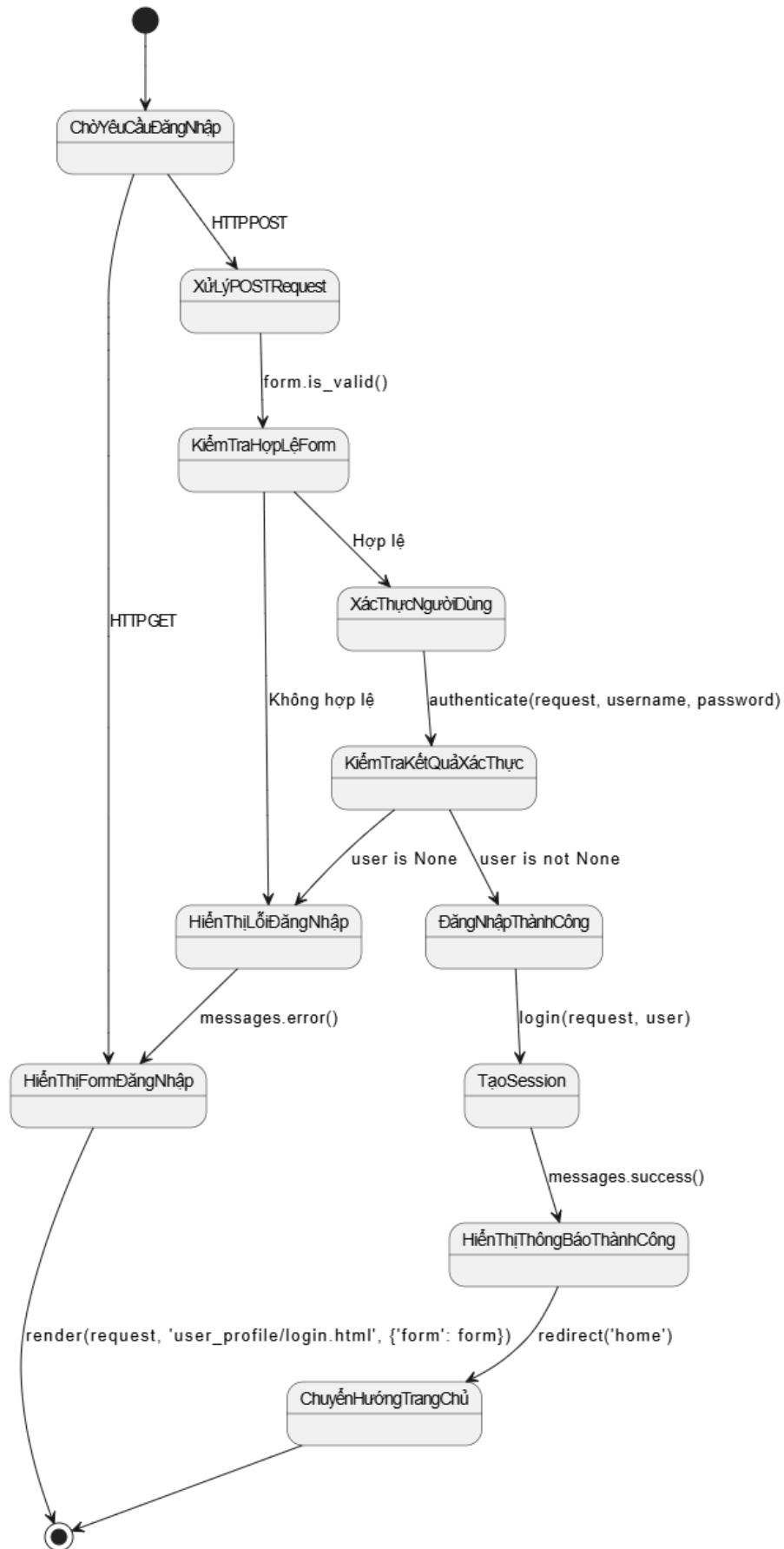


Figure 6.4: Biểu đồ trạng thái cho use case đăng nhập (mức thiết kế)

6.3 Đăng xuất (User Logout)

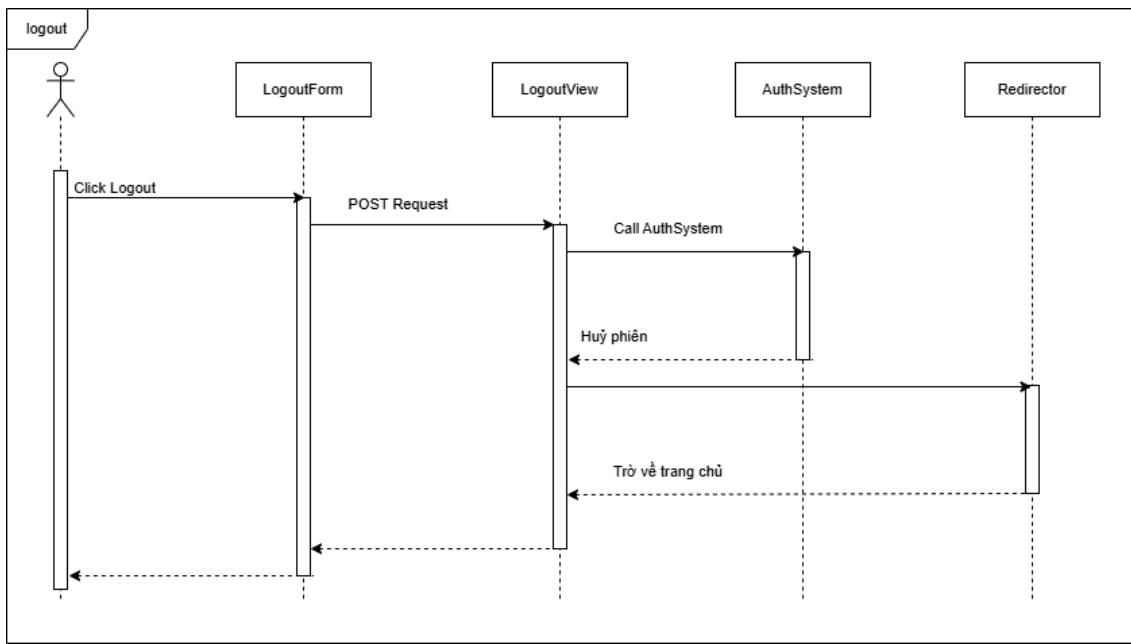


Figure 6.5: Biểu đồ tuần tự cho use case đăng xuất (mức thiết kế)

6.4 Quản lý hồ sơ người dùng (User Profile Management)

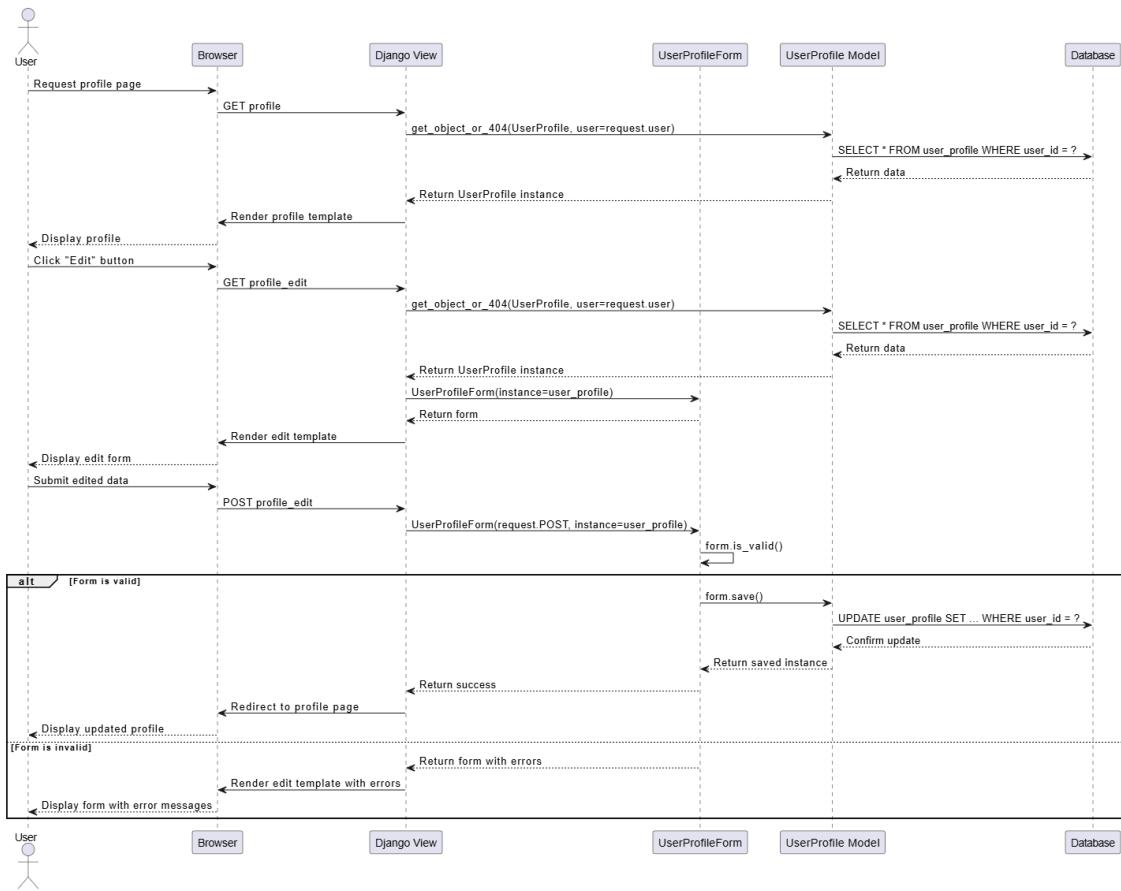


Figure 6.6: Biểu đồ tuần tự cho use case quản lý hồ sơ người dùng(bao gồm xem và sửa) (mức thiết kế)

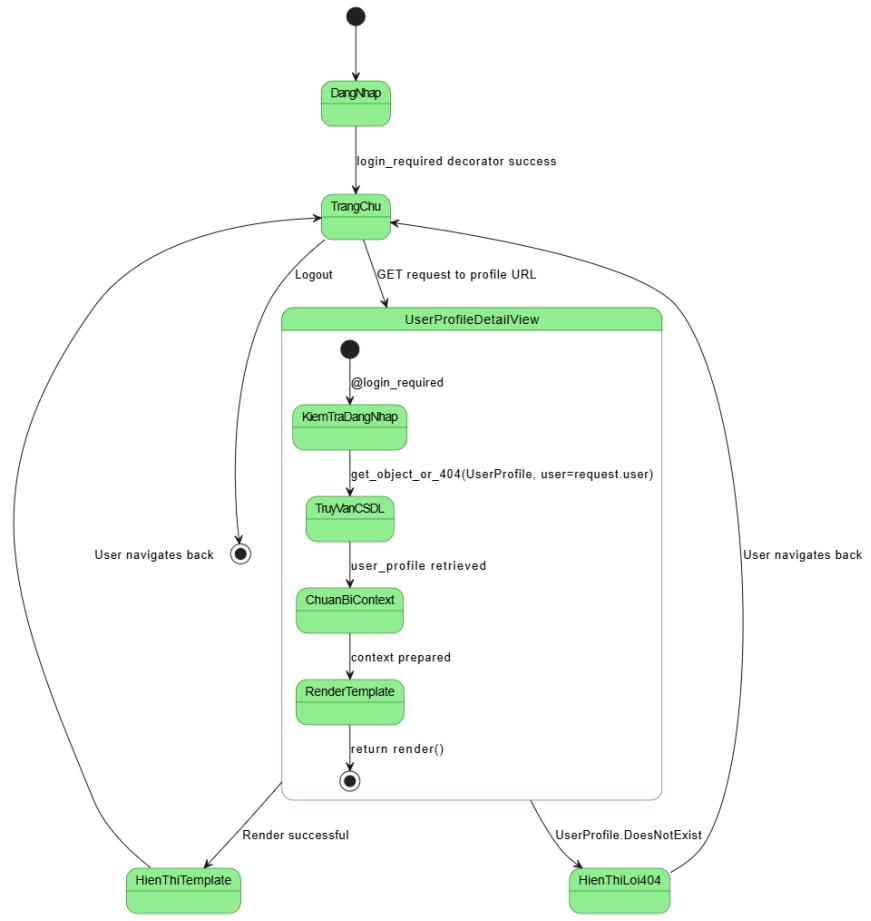


Figure 6.7: Biểu đồ trạng thái cho use case xem hồ sơ người dùng (mức thiết kế)

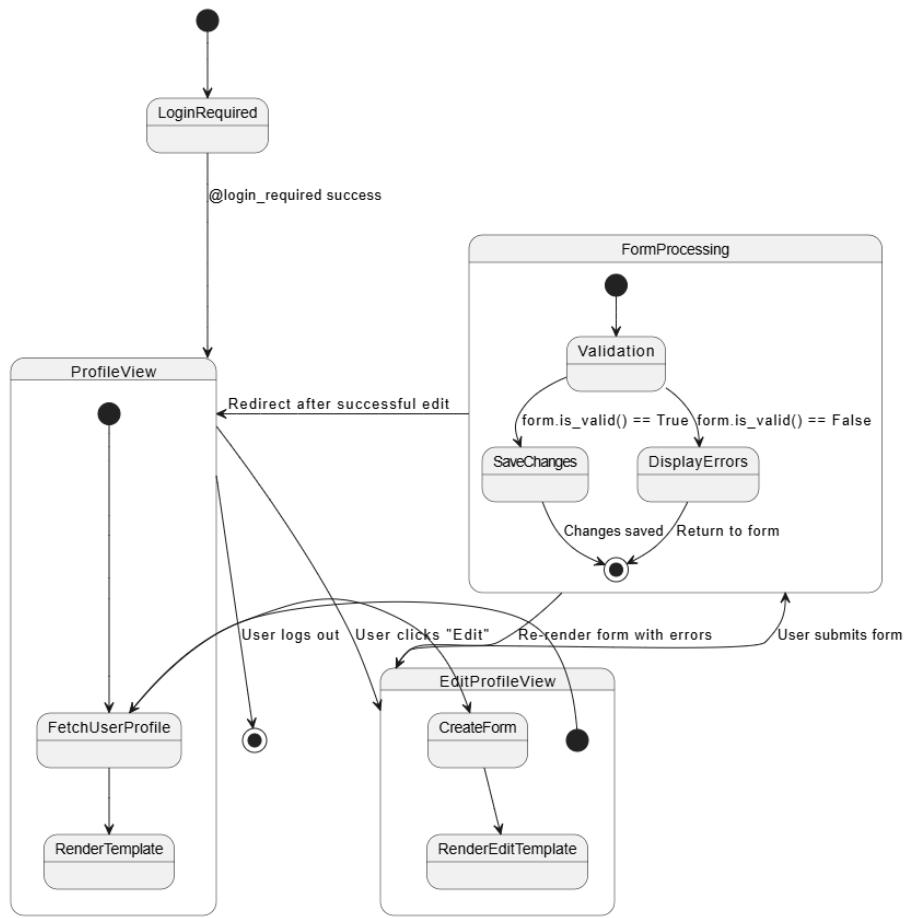


Figure 6.8: Biểu đồ trạng thái cho use case sửa hồ sơ người dùng (mức thiết kế)

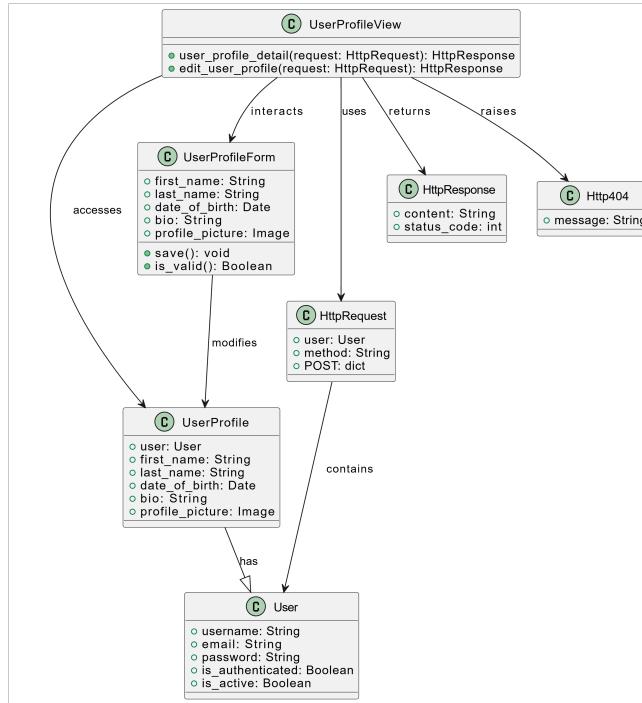


Figure 6.9: Biểu đồ lớp cho use case quản lý hồ sơ người dùng (mức thiết kế)

6.5 Quản lý danh sách phim yêu thích (Watchlist Management)

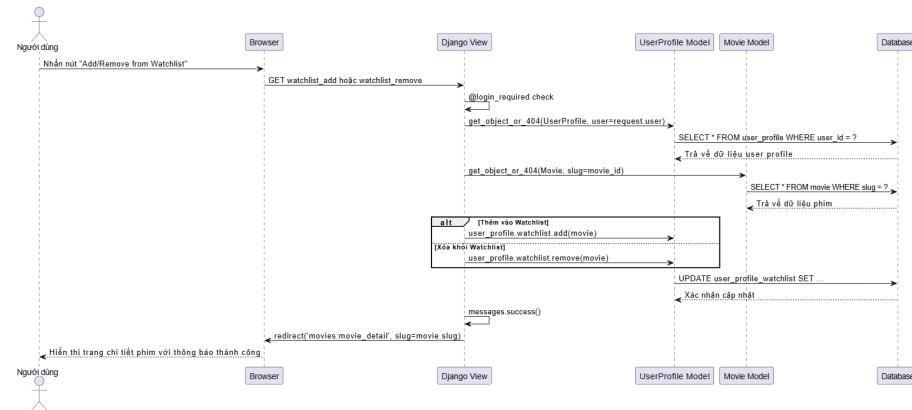


Figure 6.10: Biểu đồ tuần tự cho use case quản lý danh sách phim yêu thích (mức thiết kế)

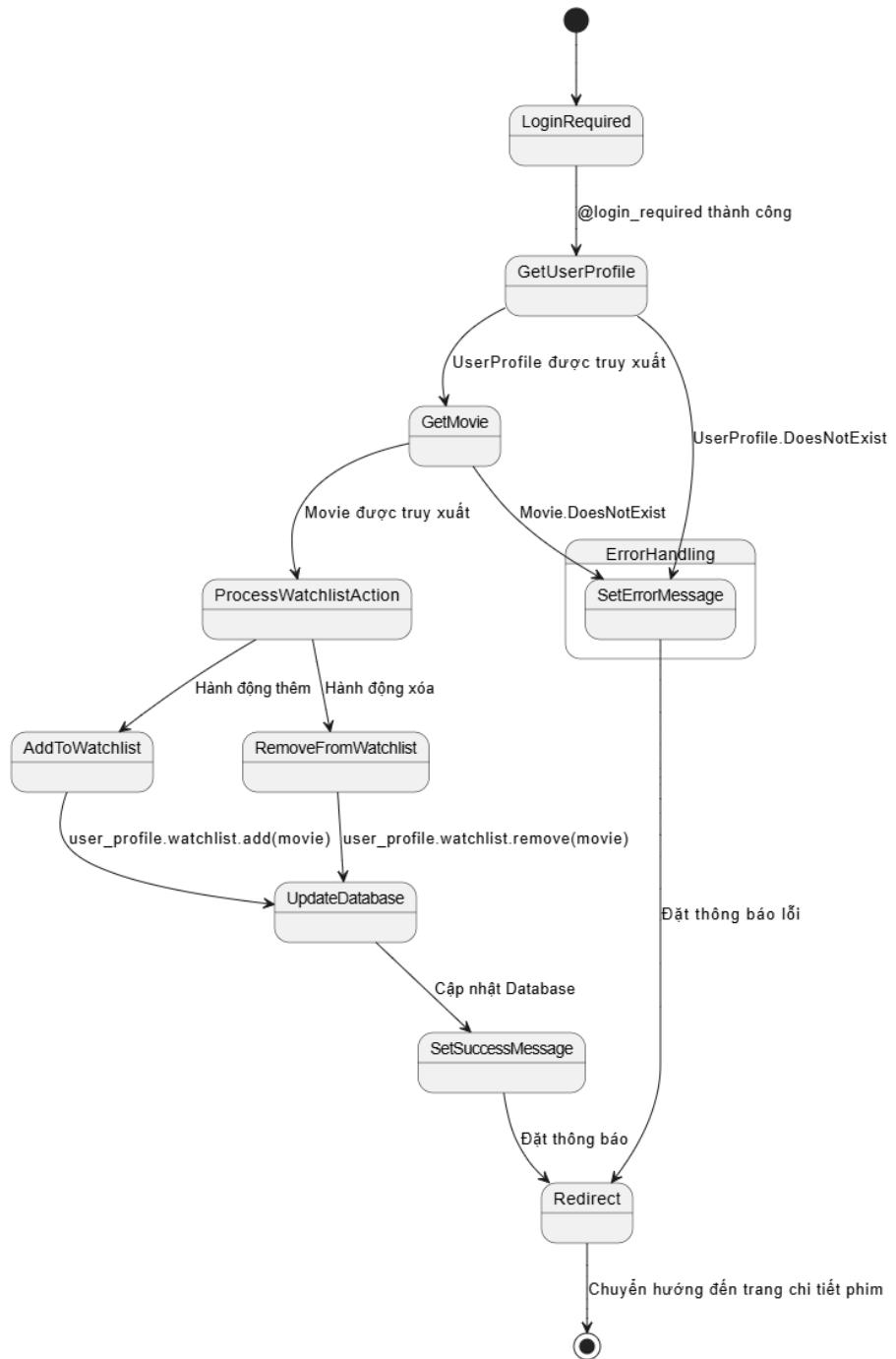


Figure 6.11: Biểu đồ trạng thái cho use case quản lý danh sách phim yêu thích (mức thiết kế)

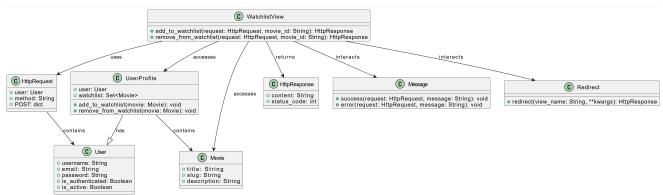


Figure 6.12: Biểu đồ lớp cho use case quản lý danh sách phim yêu thích (mức thiết kế)

6.6 Browse Movies (Duyệt Danh Sách Phim)

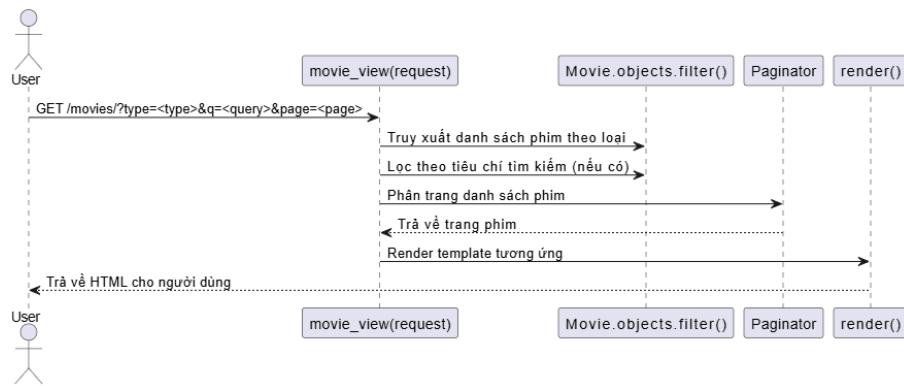


Figure 6.13: Biểu đồ tuần tự cho use case hiển thị danh sách phim (mức thiết kế)

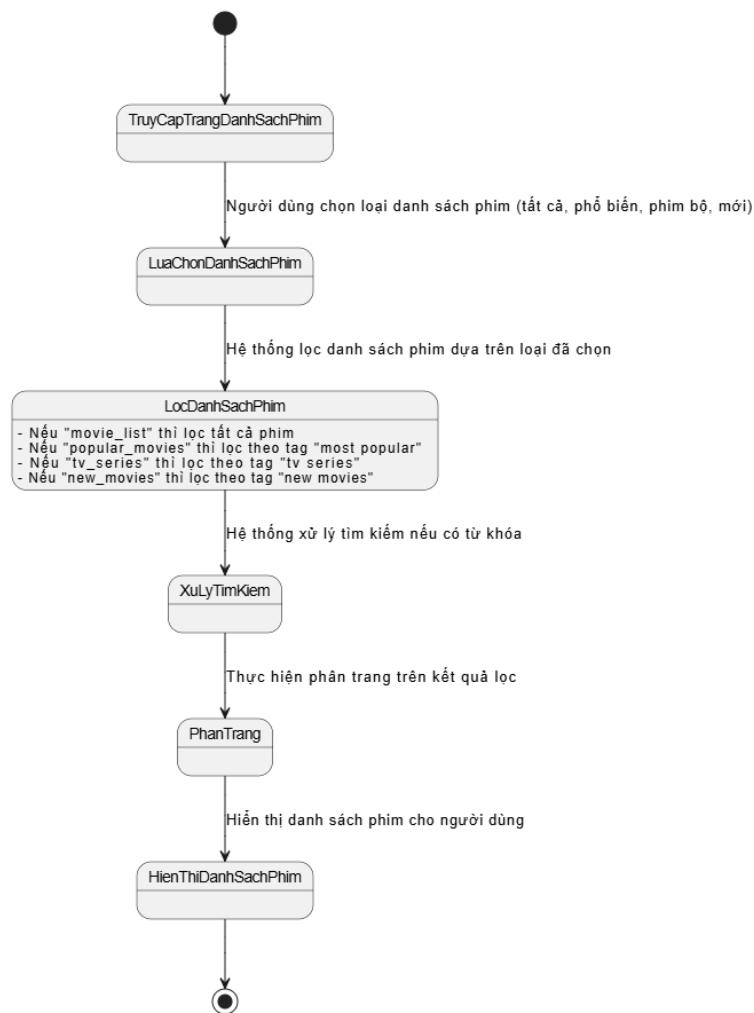


Figure 6.14: Biểu đồ trạng thái cho use case quản lý danh sách phim yêu thích (mức thiết kế)

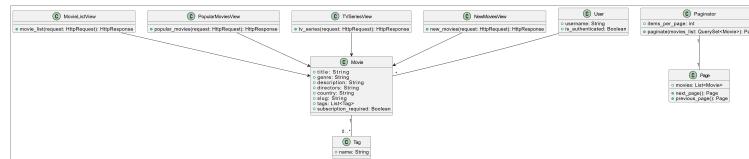


Figure 6.15: Biểu đồ lớp cho use case quản lý danh sách phim yêu thích (mức thiết kế)

6.7 View Movie Details (Xem Chi Tiết Phim)

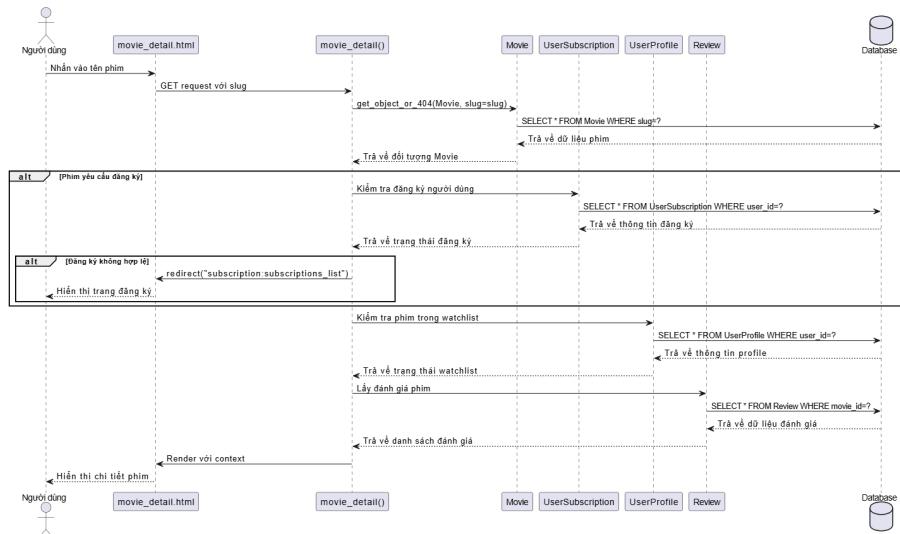


Figure 6.16: Biểu đồ tuần tự cho use case hiển thị chi tiết phim (mức thiết kế)

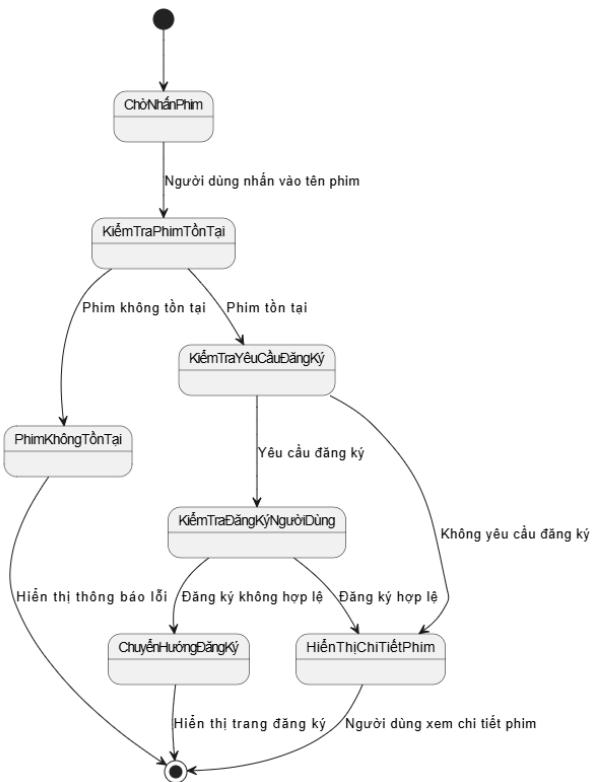


Figure 6.17: Biểu đồ trạng thái cho use case hiển thi chi tiết phim (mức thiết kế)

6.8 Movie Management

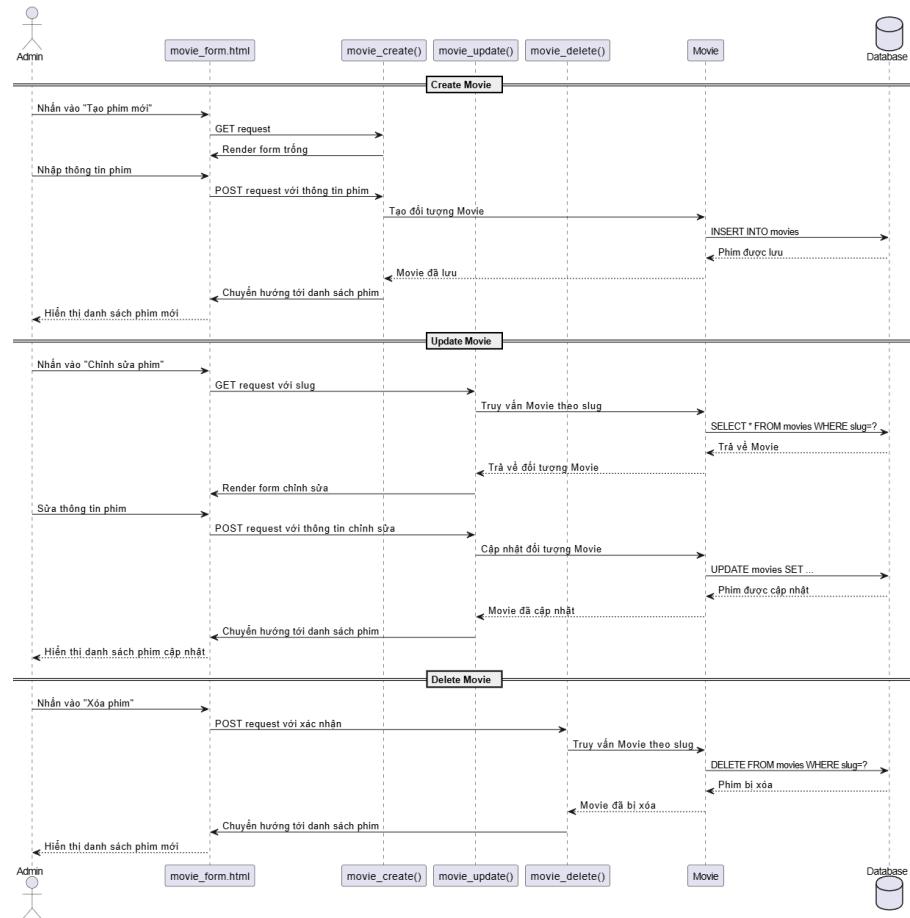


Figure 6.18: Biểu đồ tuần tự cho use case quản lý phim (mức thiết kế)

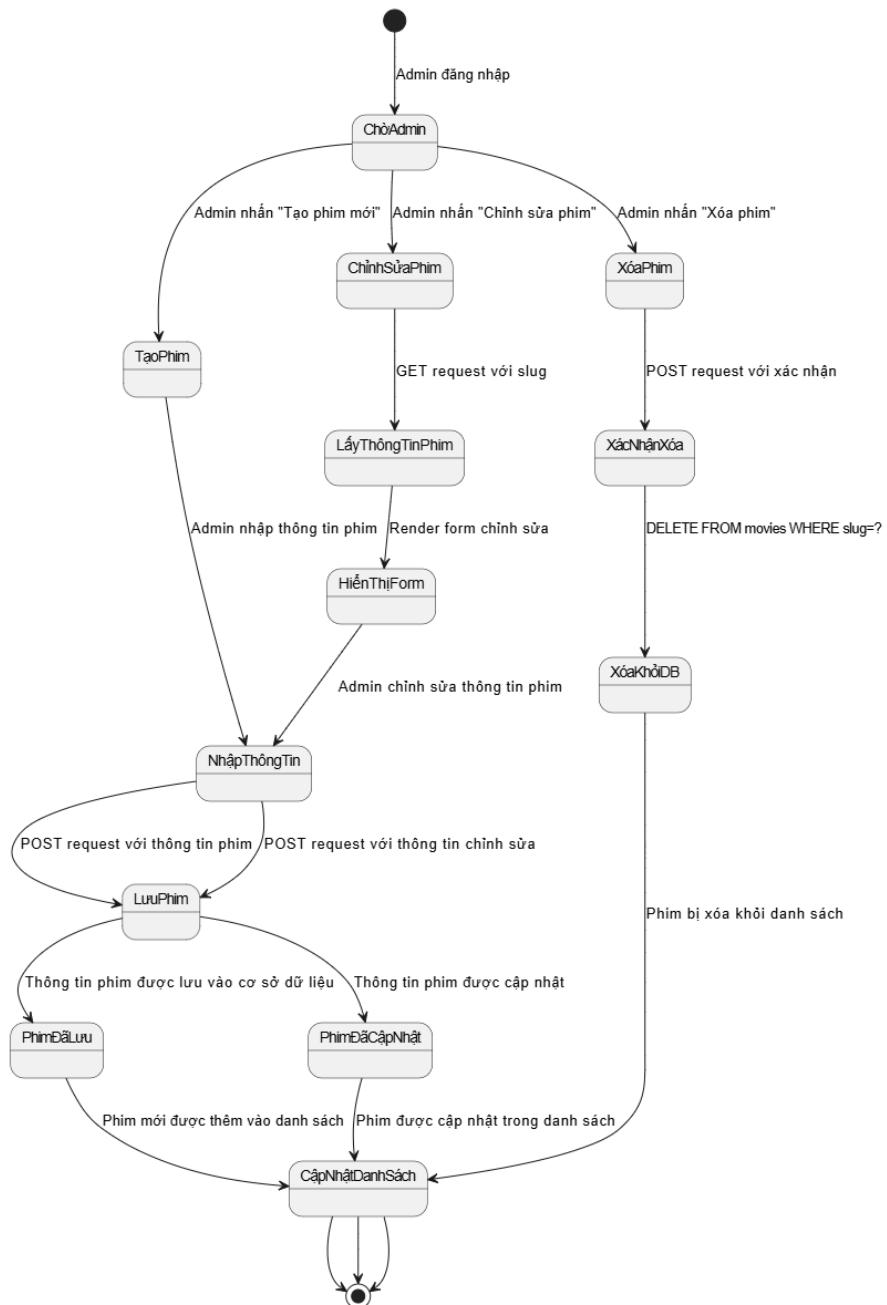


Figure 6.19: Biểu đồ trạng thái cho use case quản lý phim (mức thiết kế)

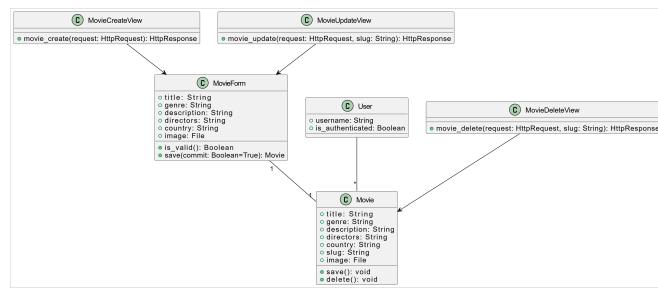


Figure 6.20: Biểu đồ lớp cho use case quản lý phim (mức thiết kế)

6.9 List Subscriptions (Hiển Thị Danh Sách Gói Đăng Ký)

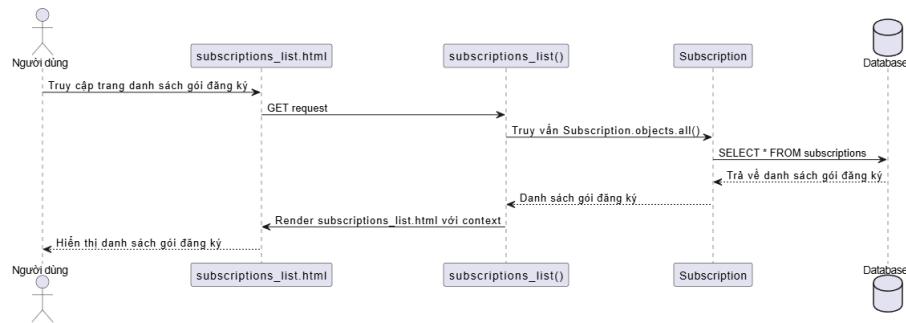


Figure 6.21: Biểu đồ tuần tự cho use case hiển thị danh sách gói đăng kí (mức thiết kế)

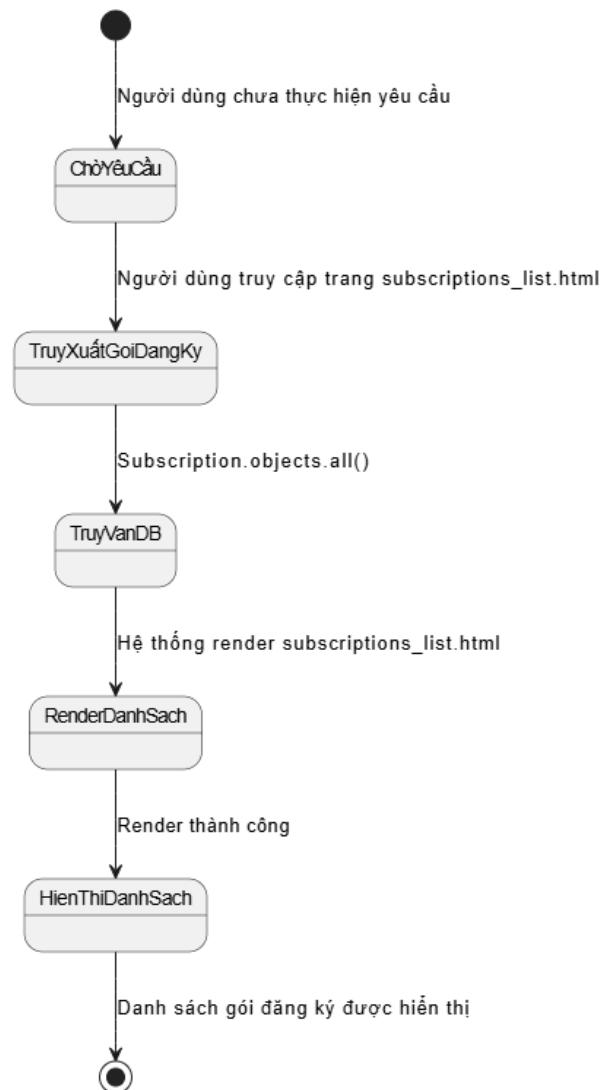


Figure 6.22: Biểu đồ trạng thái cho use case hiển thị danh sách gói đăng kí (mức thiết kế)

6.10 Subscription Detail (Hiển Thị Chi Tiết Gói Đăng Ký)

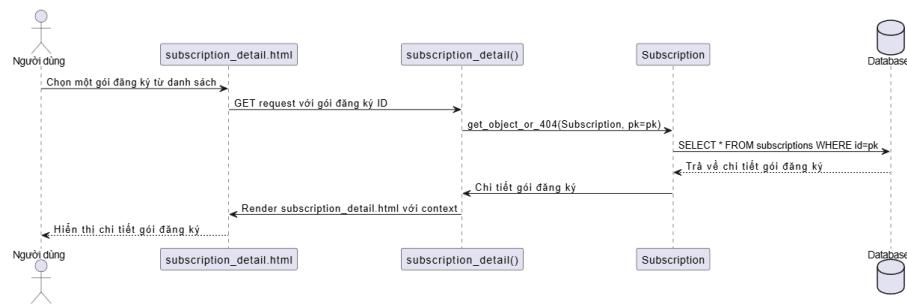


Figure 6.23: Biểu đồ tuần tự cho use case hiển thị chi tiết gói đăng kí (mức thiết kế)

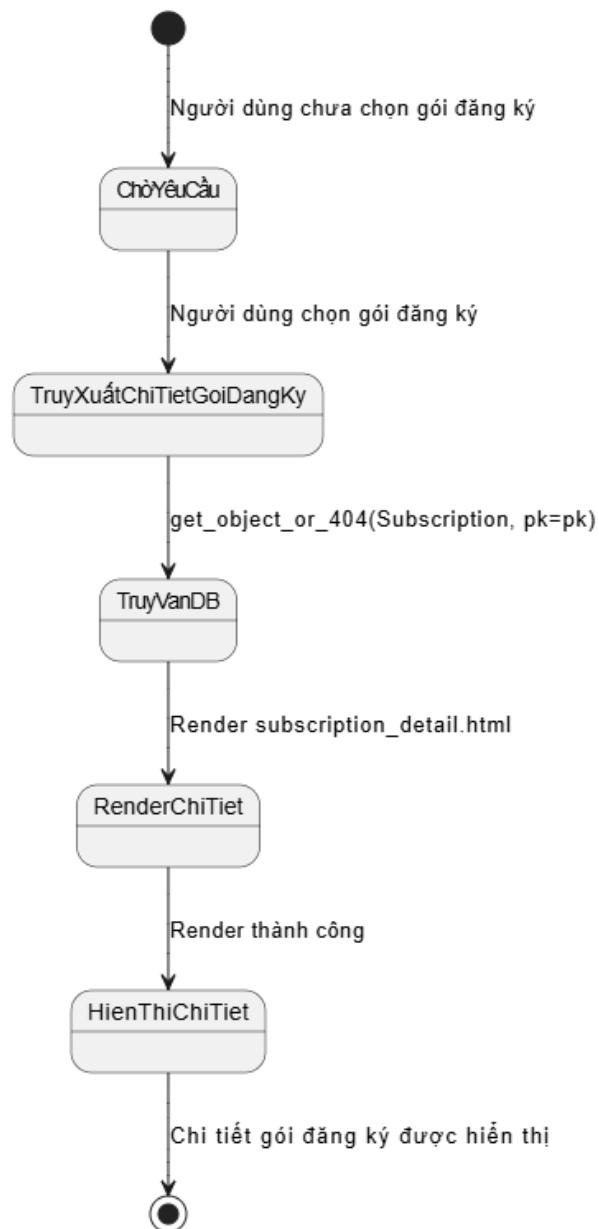


Figure 6.24: Biểu đồ trạng thái cho use case hiển thị chi tiết gói đăng kí (mức thiết kế)

6.11 Checkout Subscription (Thanh Toán Gói Đăng Ký)

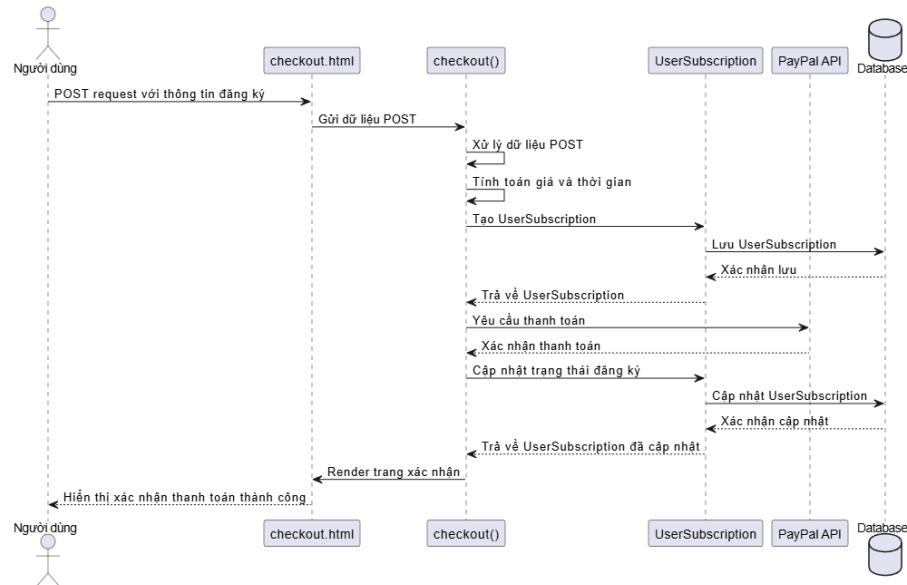


Figure 6.25: Biểu đồ tuần tự cho use case thanh toán gói đăng ký (mức thiết kế)

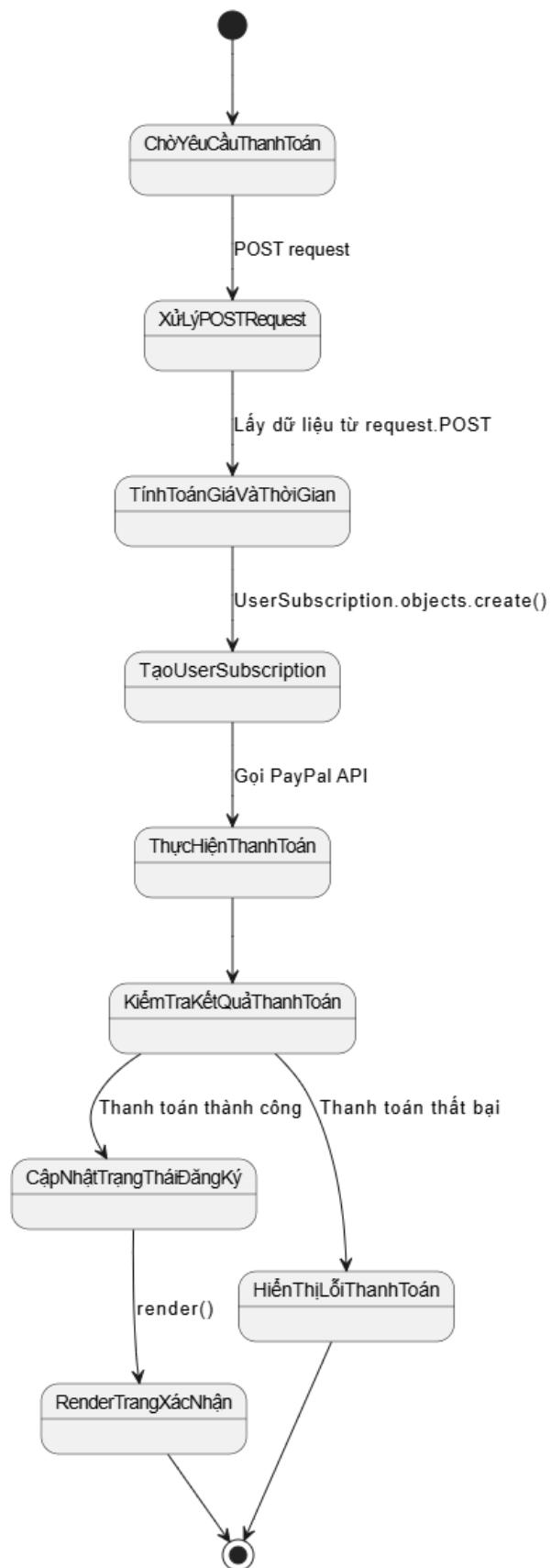


Figure 6.26: Biểu đồ trạng thái cho use case thanh toán gói đăng ký (mức thiết kế)

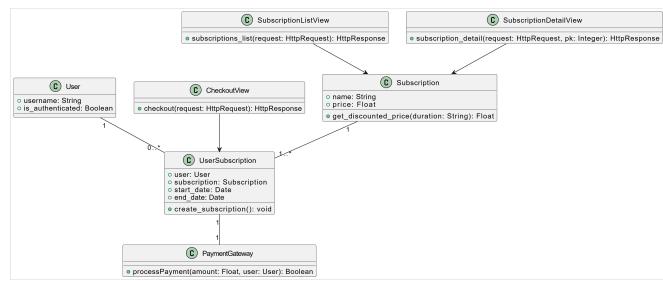


Figure 6.27: Biểu đồ lớp cho các use case `subscriptionsList`, `subscriptionDetail`, `vcheckout`

6.12 Review Movie (Đánh Giá Phim)

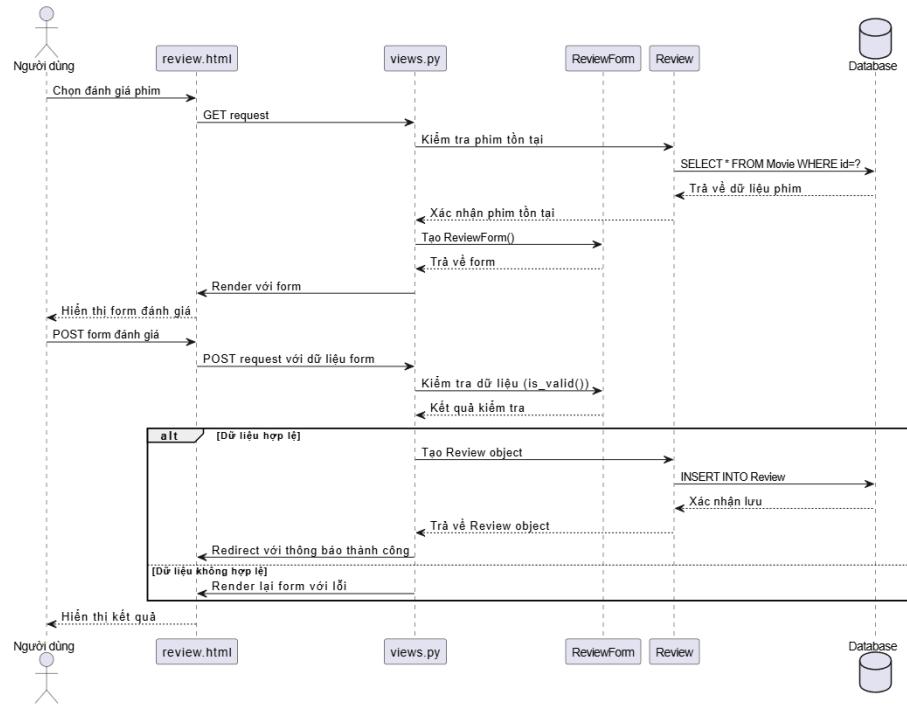


Figure 6.28: Biểu đồ tuần tự cho use case đánh giá phim (mức thiết kế)

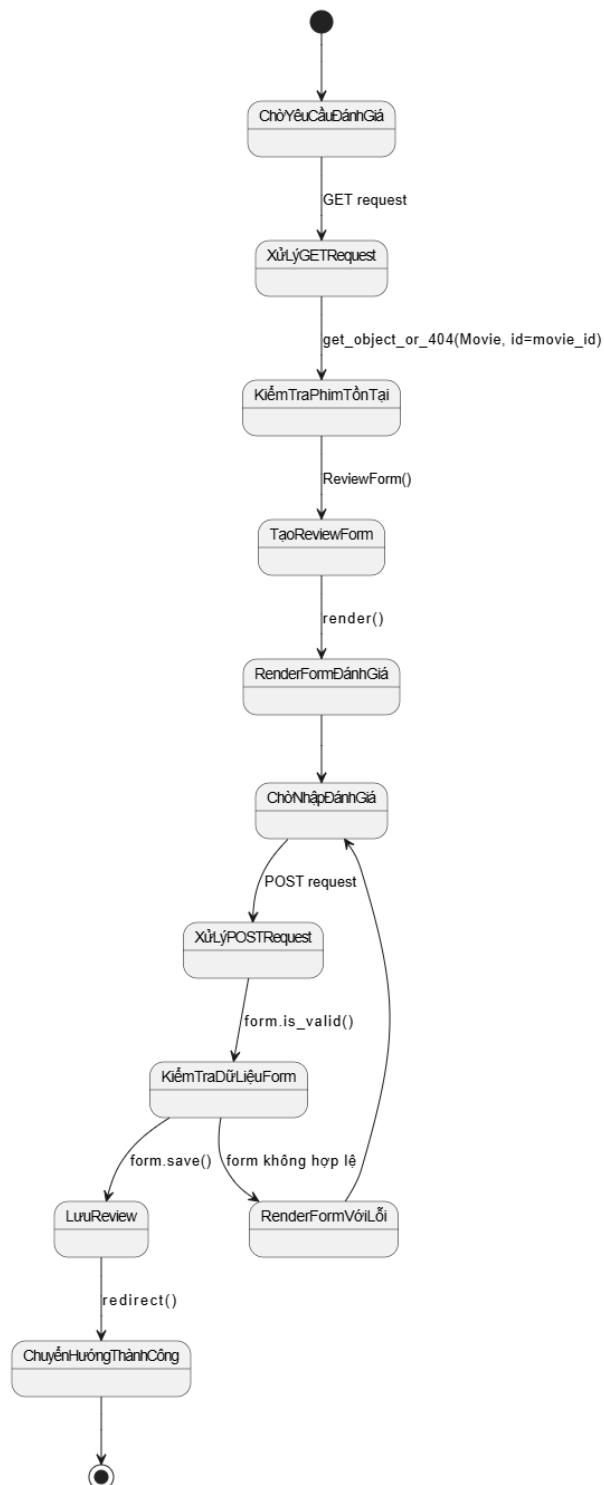


Figure 6.29: Biểu đồ trạng thái cho use case đánh giá phim (mức thiết kế)

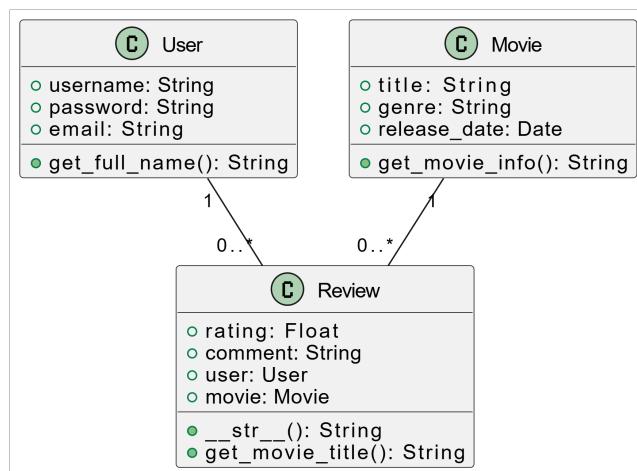


Figure 6.30: Biểu đồ lớp cho use case đánh giá phim (mức thiết kế)

Chapter 7

Databases

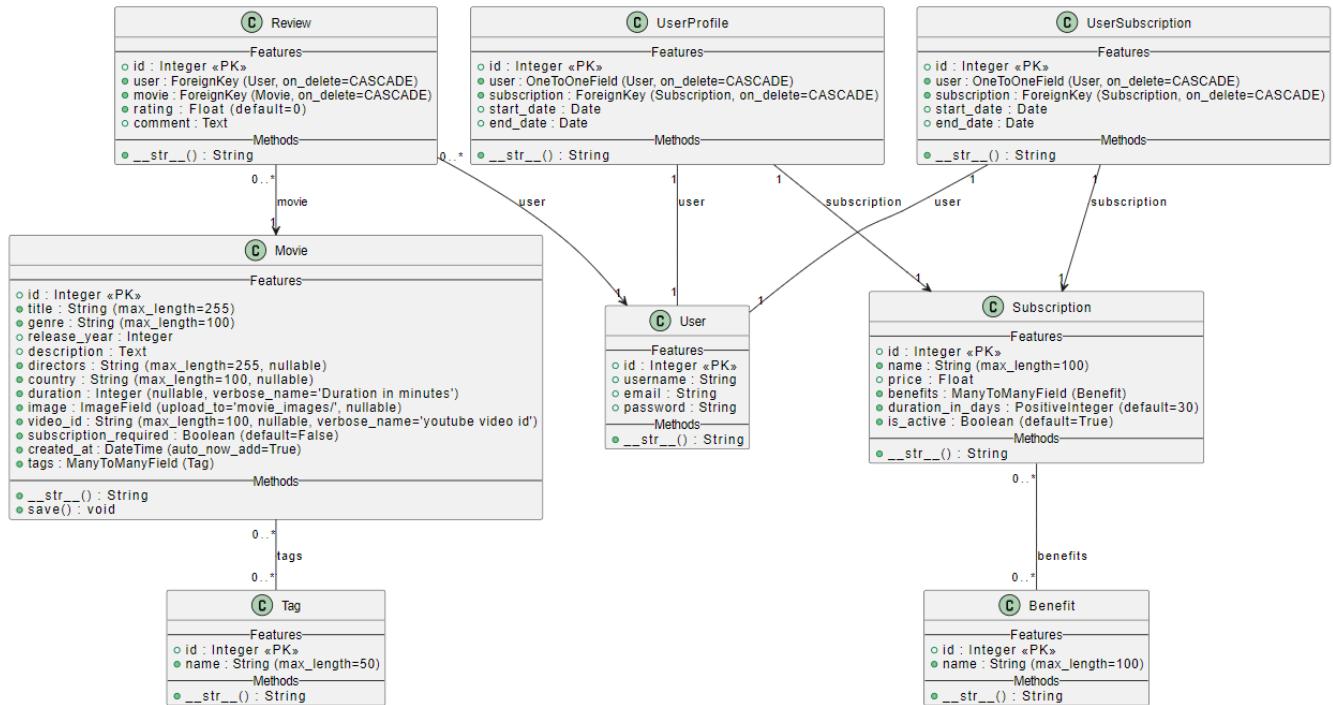


Figure 7.1: Hệ thống MotChill

Chapter 8

Demo

8.1 Video Demo Sản Phẩm

Để xem video demo sản phẩm, vui lòng truy cập đường dẫn sau:

https://drive.google.com/file/d/16k280oaMYDAGU6SnIa_TdCV0uCdmVFPw/view?usp=sharing

8.2 Github

Đường dẫn tới github dự án:

https://github.com/DucDM2202/Movie_Web

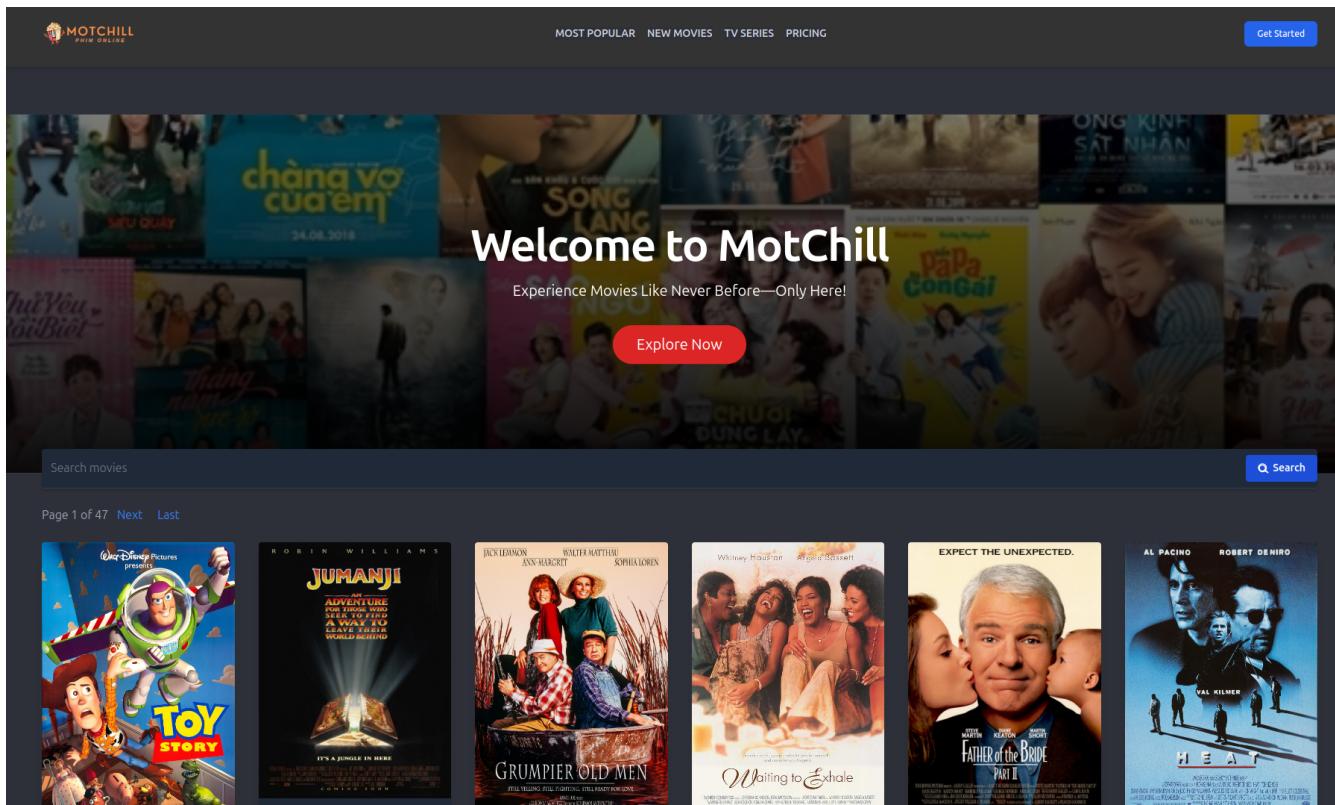


Figure 8.1: Demo giao diện

Chapter 9

Đóng góp

9.1 Bảng tự đánh giá đóng góp của từng thành viên

Thành viên	Công việc	Định lượng đóng góp
Đặng Văn Khải (MSSV: 22022550)	<ul style="list-style-type: none">Trưởng nhómThiết kế cơ sở dữ liệuPhát triển BEHỗ trợ viết báo cáo	25%
Lê Việt Hùng (MSSV: 22022666)	<ul style="list-style-type: none">Phát triển, tích hợp thuật toán học máy vào ứng dụngViết báo cáo nội dung thuật toánContainer hóa ứng dụngHỗ trợ phát triển BE	25%
Đàm Thái Ninh (MSSV: 22022522)	<ul style="list-style-type: none">Thiết kế, phân tích các use caseVẽ biểu đồViết báo cáo chínhHỗ trợ phát triển BE	25%
Dương Minh Đức (MSSV: 22022606)	<ul style="list-style-type: none">Thiết kế giao diện, phát triển FEKiểm tra, sửa lỗi giao diệnXây dựng Unit testHỗ trợ vẽ biểu đồ	25%

Chapter 10

Tổng kết

MotChill là một hệ thống web quản lý nội dung phim (MovieSubscription) cho phép người dùng đăng ký tài khoản, quản lý hồ sơ cá nhân, xem danh sách phim, xem chi tiết phim, quản lý danh sách phim yêu thích, đăng ký gói dịch vụ, và thanh toán qua PayPal. Hệ thống phải hỗ trợ các chức năng cơ bản như xác thực người dùng, quản lý hồ sơ, và xử lý các yêu cầu liên quan đến phim và đánh giá phim.

10.1 Một số vấn đề ứng dụng giải quyết

- **Nền tảng tập trung:** Cung cấp một nền tảng duy nhất với nhiều loại nội dung (phim phô biến, phim truyền hình, phim mới phát hành), giảm sự phân tán.
- **Cải thiện khả năng khám phá nội dung:**
 - Cung cấp danh sách phim được phân loại (phô biến nhất, phim truyền hình, mới phát hành).
 - Tính năng tìm kiếm phim giúp người dùng dễ dàng tìm nội dung mong muốn.
- **Trải nghiệm cá nhân hóa:**
 - Hồ sơ người dùng cho phép theo dõi sở thích và lịch sử xem.
 - Tính năng watchlist giúp người dùng lưu trữ phim yêu thích.
- **Đơn giản hóa quản lý đăng ký:**
 - Cung cấp các gói đăng ký khác nhau trên một nền tảng duy nhất.
 - Quản lý thông tin đăng ký của người dùng và gửi thông báo khi gói sắp hết hạn.
- **Tăng cường tương tác cộng đồng:**
 - Cho phép người dùng viết và đọc đánh giá phim.
 - Hệ thống xếp hạng sao giúp người dùng nhanh chóng đánh giá chất lượng phim.
- **Theo dõi nội dung dễ dàng:**
 - Tính năng watchlist giúp người dùng lưu trữ phim muốn xem.
 - Hiển thị lịch sử xem phim giúp người dùng theo dõi nội dung đã xem.

10.2 Cấu trúc dự án

Thông nhất cấu trúc dự án như sau:

- **MovieSubscription** (thư mục gốc)
 - **movie** (ứng dụng quản lý phim)
 - **user_profile** (ứng dụng quản lý hồ sơ người dùng)
 - **review** (ứng dụng quản lý đánh giá)
 - **subscription** (ứng dụng quản lý gói đăng ký)

10.3 Điểm tích cực

- Ứng dụng chạy ổn định, đáp ứng cơ bản các yêu cầu đặt ra
- Phân chia thành các app nhỏ hơn
- Giao diện dễ sử dụng
- Tích hợp thêm thuật toán gợi ý
- Container hoá nhằm đảm bảo tính dễ dàng mở rộng

10.4 Các điểm có thể cải thiện

- Microservice: Do thời gian ngắn hạn nên nhóm chưa triển khai được ứng dụng trên kiến trúc này, hoàn toàn có thể áp dụng kiến trúc này cho ứng dụng
- Stateless
- Có thể triển khai trên AWS