

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

**HỆ THỐNG XEM PHIM TRỰC TUYẾN
KẾT HỢP AI GỢI Ý**

Người hướng dẫn: ThS. MAI VĂN HÀ
Sinh viên thực hiện: HOÀNG VĂN QUÝ
Số thẻ sinh viên: 102210182
Lớp: 21TCLC_DT2

Đà Nẵng, 06/2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

HỆ THỐNG XEM PHIM TRỰC TUYẾN KẾT HỢP AI GỢI Ý

Người hướng dẫn: ThS. MAI VĂN HÀ
Sinh viên thực hiện: HOÀNG VĂN QUÝ
Số thẻ sinh viên: 102210182
Lớp: 21TCLC_DT2

Đà Nẵng, 06/2025

TÓM TẮT

Tên đề tài: Hệ thống xem phim trực tuyến kết hợp AI gợi ý

Sinh viên thực hiện: Hoàng Văn Quý

Số thẻ SV: 102210182 Lớp: 21TCLC_DT2

Công nghệ thông tin ngày càng phát triển và đóng vai trò quan trọng trong nhiều lĩnh vực của đời sống xã hội, đặc biệt là trong ngành giải trí số. Những năm gần đây, nhu cầu xem phim trực tuyến tăng cao, kéo theo yêu cầu ngày càng khắt khe về trải nghiệm người dùng. Người dùng không chỉ muốn xem phim mọi lúc, mọi nơi mà còn mong muốn được gợi ý những nội dung phù hợp với sở thích cá nhân, giúp tiết kiệm thời gian tìm kiếm và nâng cao trải nghiệm giải trí.

Tuy nhiên, nhiều nền tảng xem phim hiện nay vẫn còn hạn chế trong khả năng cá nhân hóa. Người dùng thường phải tự tìm kiếm hoặc chọn ngẫu nhiên, dẫn đến trải nghiệm chưa tối ưu. Từ thực tế đó, em chọn đề tài “Hệ thống xem phim trực tuyến kết hợp với AI gợi ý” làm đề án tốt nghiệp, với mục tiêu xây dựng một hệ thống đơn giản nhưng hiệu quả, có khả năng đề xuất phim dựa trên sở thích và hành vi người dùng.

Đề tài áp dụng hai kỹ thuật cơ bản trong hệ thống gợi ý: Content-Based Filtering (CBF) – gợi ý phim dựa trên các đặc điểm nội dung của phim như thể loại, quốc gia, diễn viên,...; và Neural Matrix Factorization (NeuMF) – một mô hình học sâu đơn giản, học từ lịch sử đánh giá và danh sách phim yêu thích của người dùng. Sự kết hợp này giúp hệ thống cá nhân hóa gợi ý tốt hơn so với việc chỉ dùng một phương pháp đơn lẻ.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Hoàng Văn Quý

Số thẻ sinh viên: 102210182

Lớp: 21TCLC_DT2

Khoa: Công nghệ thông tin

Ngành: Công nghệ thông tin (Đặc thù – Hợp tác doanh nghiệp)

- Tên đề tài đồ án:* Hệ thống xem phim trực tuyến kết hợp AI gợi ý
- Đề tài thuộc diện:* Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
- Các số liệu và dữ liệu ban đầu:* Không có
- Nội dung các phần thuyết minh và tính toán*
 - Mở đầu:** Giới thiệu chung về đề tài, mục đích thực hiện, mục tiêu hướng đến, phạm vi, đối tượng tiếp cận, phương pháp nghiên cứu và cấu trúc đồ án.
 - Chương 1:** Cơ sở lý thuyết
 - Chương 2:** Phân tích và thiết kế hệ thống
 - Chương 3:** Triển khai hệ thống
 - Kết luận:** Tổng kết kết quả đạt được, những hạn chế của hệ thống và hướng triển khai trong tương lai
- Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ): Không có*
- Họ tên người hướng dẫn:* ThS. Mai Văn Hà
- Ngày giao nhiệm vụ đồ án:*/...../2025
- Ngày hoàn thành đồ án:*/...../2025

Đà Nẵng, ngày tháng năm 2025

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Để hoàn thành được đề án tốt nghiệp này, em không thể không nhắc đến sự hỗ trợ và đồng hành của rất nhiều người – những người đã tạo điều kiện, truyền cảm hứng và tiếp thêm động lực cho em trong suốt hành trình học tập và thực hiện đề tài.

Trước hết, em xin gửi lời cảm ơn chân thành đến tất cả quý Thầy Cô khoa Công nghệ Thông tin – Trường Đại học Bách Khoa, Đại học Đà Nẵng. Không chỉ là người truyền đạt kiến thức, các Thầy Cô còn là người định hướng, khơi gợi tư duy và giúp em rèn luyện tinh thần trách nhiệm với chính sản phẩm mình làm ra.

Đặc biệt, em muốn gửi lời cảm ơn sâu sắc đến thầy ThS. Mai Văn Hà – người đã đồng hành cùng em từ lúc còn mơ hồ về đề tài cho đến khi đề án hoàn chỉnh. Thầy luôn kiên nhẫn lắng nghe, đưa ra những góp ý rất cụ thể và thực tế, giúp em dần hiểu rõ hơn mình cần làm gì và làm như thế nào. Nhờ có sự hỗ trợ tận tình từ thầy, em mới có thể định hướng được nội dung, xử lý các vấn đề và hoàn thiện đề án một cách trọn vẹn hơn.

Cũng không thể thiếu lời cảm ơn đến gia đình – những người luôn âm thầm ủng hộ em cả về tinh thần lẫn điều kiện học tập. Cảm ơn bạn bè đã giúp đỡ, động viên, chia sẻ kiến thức và cùng nhau vượt qua những giai đoạn áp lực nhất.

Dù đã cố gắng hết sức, nhưng em hiểu rằng đề án này vẫn còn nhiều điểm chưa hoàn hảo. Rất mong nhận được những góp ý, phản hồi chân thành từ quý Thầy Cô và các bạn để em có thể cải thiện và học hỏi thêm cho chặng đường phía trước.

Chân thành cảm ơn tất cả mọi người!

CAM ĐOAN

Tôi xin cam đoan:

1. Đồ án tốt nghiệp: Hệ thống xem phim trực tuyến kết hợp AI gợi ý là chính tôi thực hiện dưới sự hướng dẫn trực tiếp của giảng viên ThS. Mai Văn Hà
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, kèm với đường dẫn hợp lệ.
3. Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Hoàng Văn Quý

MỤC LỤC

| | |
|--|-------------|
| TÓM TẮT | iv |
| NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP | v |
| LỜI NÓI ĐẦU..... | ii |
| CAM ĐOAN..... | iii |
| MỤC LỤC | iv |
| DANH SÁCH CÁC BẢNG..... | vi |
| DANH SÁCH CÁC HÌNH VẼ | viii |
| DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT | xi |
| MỞ ĐẦU..... | 1 |
| 1 CHƯƠNG 1: CƠ SỞ LÝ THUYẾT..... | 4 |
| 1.1 Tổng quát về hệ thống gợi ý | 4 |
| 1.1.1 Hệ thống gợi ý | 4 |
| 1.1.2 Các loại hệ thống gợi ý phổ biến..... | 4 |
| 1.1.3 Các mô hình sử dụng trong đồ án..... | 7 |
| 1.2 Công nghệ và công cụ phát triển..... | 8 |
| 1.2.1 Backend | 8 |
| 1.2.2 Frontend..... | 11 |
| 1.2.3 Cơ sở dữ liệu & lưu trữ..... | 13 |
| 1.2.4 Tích hợp thanh toán | 14 |
| 1.2.5 Git và Github | 14 |
| 1.2.6 Deployment | 15 |
| 1.2.7 Xử lý ngôn ngữ tự nhiên..... | 18 |
| 1.2.8 Recbole | 19 |
| 1.3 Kết luận chương..... | 20 |
| 2 CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG..... | 22 |
| 2.1 Phân tích yêu cầu đề tài..... | 22 |
| 2.2 Yêu cầu chức năng..... | 22 |
| 2.2.1 Khách..... | 22 |
| 2.2.2 Người dùng | 22 |
| 2.2.3 Quản trị viên | 23 |
| 2.3 Các yêu cầu phi chức năng | 23 |

| | | |
|----------|---|------------|
| 2.4 | Thiết kế hệ thống..... | 24 |
| 2.4.1 | Sơ đồ use case..... | 24 |
| 2.4.2 | Đặc tả use case..... | 31 |
| 2.4.3 | Sơ đồ nguyên lý hoạt động..... | 42 |
| 2.4.4 | Sơ đồ tuần tự..... | 48 |
| 2.4.5 | Sơ đồ quan hệ cơ sở dữ liệu..... | 54 |
| 2.5 | Thiết kế tổng quan về kiến trúc hệ thống..... | 64 |
| 2.6 | Kết luận chương..... | 65 |
| 3 | CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG | 66 |
| 3.1 | Quy trình triển khai hệ thống gợi ý..... | 66 |
| 3.1.1 | Mô hình gợi ý lọc theo nội dung (CBF)..... | 66 |
| 3.1.2 | Mô hình NeuMF:..... | 69 |
| 3.1.3 | Kết hợp CBF và NeuMF..... | 72 |
| 3.2 | Công cụ phát triển..... | 74 |
| 3.3 | Kết quả..... | 76 |
| 3.3.1 | Trang đăng ký tài khoản..... | 76 |
| 3.3.2 | Màn hình nhập OTP để tiến hành đăng ký..... | 76 |
| 3.3.3 | Email chứa mã OTP khi đăng ký..... | 77 |
| 3.3.4 | Trang đăng nhập..... | 77 |
| 3.3.5 | Trang chủ của người dùng..... | 78 |
| 3.3.6 | Màn hình tìm kiếm phim..... | 79 |
| 3.3.7 | Kết quả tìm kiếm..... | 79 |
| 3.3.8 | Các màn hình phim..... | 79 |
| 3.3.9 | Gói dịch vụ..... | 84 |
| 3.3.10 | Màn hình quản lý tài khoản..... | 87 |
| 3.3.11 | Các màn hình diễn viên..... | 89 |
| 3.3.12 | Các màn hình của quản trị viên..... | 90 |
| 3.4 | Kết luận chương..... | 98 |
| 4 | KẾT LUẬN | 99 |
| 5 | TÀI LIỆU THAM KHẢO..... | 102 |

DANH SÁCH CÁC BẢNG

| | | |
|-----------|---|----|
| Bảng 2.1 | Đặc tả use case đăng ký tài khoản | 31 |
| Bảng 2.2 | Đặc tả use case đăng nhập | 31 |
| Bảng 2.3 | Đặc tả use case quên mật khẩu | 32 |
| Bảng 2.4 | Đặc tả use case thêm thể loại | 32 |
| Bảng 2.5 | Đặc tả use case cập nhập thể loại | 33 |
| Bảng 2.6 | Đặc tả use case xóa thể loại | 34 |
| Bảng 2.7 | Đặc tả tìm kiếm thể loại | 34 |
| Bảng 2.8 | Đặc tả use case thêm diễn viên | 35 |
| Bảng 2.9 | Đặc tả use case cập nhập diễn viên | 35 |
| Bảng 2.10 | Đặc tả use case thêm phim | 36 |
| Bảng 2.11 | Đặc tả use case cập nhập phim | 36 |
| Bảng 2.12 | Đặc tả use case xóa phim | 37 |
| Bảng 2.13 | Đặc tả use case tìm kiếm phim | 37 |
| Bảng 2.14 | Đặc tả use case thêm gói dịch vụ | 38 |
| Bảng 2.15 | Đặc tả use case cập nhập gói dịch vụ | 38 |
| Bảng 2.16 | Đặc tả use case xóa gói dịch vụ | 38 |
| Bảng 2.17 | Đặc tả use case thống kê doanh thu | 39 |
| Bảng 2.18 | Đặc tả use case mua gói dịch vụ | 40 |
| Bảng 2.19 | Đặc tả use case xem danh sách phim yêu thích | 40 |
| Bảng 2.20 | Đặc tả use case thêm phim vào danh sách yêu thích | 41 |
| Bảng 2.21 | Đặc tả xóa phim khỏi danh sách yêu thích | 41 |
| Bảng 2.22 | Cấu trúc bảng User | 55 |
| Bảng 2.23 | Cấu trúc bảng UserDetails | 55 |
| Bảng 2.24 | Cấu trúc bảng Role | 55 |
| Bảng 2.25 | Cấu trúc bảng Notification | 56 |
| Bảng 2.26 | Cấu trúc bảng UserNotification | 56 |
| Bảng 2.27 | Cấu trúc bảng Movie | 56 |
| Bảng 2.28 | Cấu trúc bảng Actor | 57 |
| Bảng 2.29 | Cấu trúc bảng MovieActor | 58 |
| Bảng 2.30 | Cấu trúc bảng Country | 58 |
| Bảng 2.31 | Cấu trúc bảng MovieCountry | 58 |
| Bảng 2.32 | Cấu trúc bảng Genre | 58 |
| Bảng 2.33 | Cấu trúc bảng MovieGenre | 59 |
| Bảng 2.34 | Cấu trúc bảng StandaloneMovie | 59 |
| Bảng 2.35 | Cấu trúc bảng SeriesMovie | 59 |
| Bảng 2.36 | Cấu trúc bảng VideoVersion | 59 |
| Bảng 2.37 | Cấu trúc bảng Episode | 60 |
| Bảng 2.38 | Cấu trúc bảng SubscriptionPlan | 60 |

| | |
|---|----|
| Bảng 2.39 Cấu trúc bảng ScriptionMovie | 61 |
| Bảng 2.40 Cấu trúc bảng SubscriptionPlanHierarchy | 61 |
| Bảng 2.41 Cấu trúc bảng PlanDuration | 61 |
| Bảng 2.42 Cấu trúc bảng SubscriptionOrder | 62 |
| Bảng 2.43 Cấu trúc bảng FavoriteMovie | 62 |
| Bảng 2.44 Cấu trúc bảng Comment | 63 |
| Bảng 2.45 Cấu trúc bảng UserInteraction | 63 |
| Bảng 2.46 Bảng nhiệm vụ các thành phần trong sơ đồ kiến trúc tổng quát hệ thống ... | 64 |

DANH SÁCH CÁC HÌNH VẼ

| | | |
|-----------|--|----|
| Hình 1.1 | Mối quan hệ giữa người dùng và các đề xuất..... | 4 |
| Hình 1.2 | Lọc cộng tác | 5 |
| Hình 1.3 | Lọc dự trên nội dung | 6 |
| Hình 1.4 | Cấu trúc của mô hình NeuMF | 8 |
| Hình 1.5 | Cơ chế hoạt động của RESTFul API..... | 9 |
| Hình 1.6 | Sơ đồ hoạt động của RESTFul API..... | 9 |
| Hình 1.7 | Cấu trúc của một JWT | 10 |
| Hình 1.8 | Cách hoạt động của websocket..... | 11 |
| Hình 2.1 | Sơ đồ use case tổng quát..... | 24 |
| Hình 2.2 | Sơ đồ use case quản lý thẻ loại..... | 25 |
| Hình 2.3 | Sơ đồ use case quản lý diễn viên..... | 26 |
| Hình 2.4 | Sơ đồ use case quản lý phim | 27 |
| Hình 2.5 | Sơ đồ use case quản lý gói dịch vụ..... | 28 |
| Hình 2.6 | Sơ đồ use case quản lý doanh thu..... | 28 |
| Hình 2.7 | Sơ đồ use case quản lý thông tin tài khoản | 29 |
| Hình 2.8 | Sơ đồ use case đánh giá, bình luận phim..... | 29 |
| Hình 2.9 | Sơ đồ use case quản lý phim yêu thích..... | 30 |
| Hình 2.10 | Sơ đồ use case mua gói dịch vụ..... | 30 |
| Hình 2.11 | Sơ đồ hoạt động đăng nhập | 43 |
| Hình 2.12 | Sơ đồ hoạt động đăng ký | 44 |
| Hình 2.13 | Sơ đồ hoạt động quên mật khẩu | 45 |
| Hình 2.14 | Sơ đồ hoạt động thêm phim..... | 46 |
| Hình 2.15 | Sơ đồ hoạt động mua gói dịch vụ | 47 |
| Hình 2.16 | Sơ đồ tuần tự đăng nhập tài khoản | 48 |
| Hình 2.17 | Sơ đồ tuần tự đăng ký tài khoản | 49 |
| Hình 2.18 | Sơ đồ tuần tự cập nhật thông tin cá nhân..... | 50 |
| Hình 2.19 | Sơ đồ tuần tự thay đổi mật khẩu..... | 51 |
| Hình 2.20 | Sơ đồ tuần tự chức năng thanh toán | 52 |
| Hình 2.21 | Sơ đồ tuần tự thống kê..... | 53 |
| Hình 2.22 | Sơ đồ quan hệ cơ sở dữ liệu | 54 |
| Hình 2.23 | Sơ đồ kiến trúc tổng quát hệ thống..... | 64 |
| Hình 3.1 | Các phim được đề cử bởi CBF | 68 |
| Hình 3.2 | Phản hồi API từ CBF Engine | 68 |
| Hình 3.3 | Kết quả đánh giá mô hình trên tập kiểm tra | 71 |
| Hình 3.4 | Phản hồi API từ NeuMF Engine | 72 |
| Hình 3.5 | Phản hồi API từ Hybrid Engine..... | 73 |
| Hình 3.6 | Sơ đồ triển khai hệ thống CI/CD cho backend..... | 75 |

| | |
|---|----|
| Hình 3.7 Trang đăng ký tài khoản | 76 |
| Hình 3.8 Màn hình nhập OTP | 76 |
| Hình 3.9 Email chứa OTP khi đăng ký | 77 |
| Hình 3.10 Trang đăng nhập | 77 |
| Hình 3.11 Trang chủ của người dùng | 78 |
| Hình 3.12 Màn hình tìm kiếm phim | 79 |
| Hình 3.13 Kết quả tìm kiếm | 79 |
| Hình 3.14 Danh sách phim lẻ | 80 |
| Hình 3.15 Phim lẻ | 80 |
| Hình 3.16 Phiên bản chiếu của phim lẻ | 81 |
| Hình 3.17 Diễn viên và vai diễn trong phim | 81 |
| Hình 3.18 Danh sách phim được đề xuất bởi mô hình | 82 |
| Hình 3.19 Màn hình chiếu phim | 82 |
| Hình 3.20 Danh sách phim bộ | 83 |
| Hình 3.21 Phiên bản và các tập phim của phim bộ | 83 |
| Hình 3.22 Gói dịch vụ của phim | 84 |
| Hình 3.23 Danh sách các gói dịch vụ | 84 |
| Hình 3.24 Các thời hạn gói của dịch vụ | 85 |
| Hình 3.25 Màn hình thanh toán | 85 |
| Hình 3.26 Màn hình thanh toán thành công | 86 |
| Hình 3.27 Màn hình thanh toán thất bại | 86 |
| Hình 3.28 Thông báo khi mua gói thành công | 87 |
| Hình 3.29 Thông tin cá nhân của người dùng | 87 |
| Hình 3.30 Danh sách phim yêu thích | 88 |
| Hình 3.31 Các giao dịch của người dùng | 88 |
| Hình 3.32 Lịch sử xem phim của người dùng | 89 |
| Hình 3.33 Danh sách diễn viên trong hệ thống | 89 |
| Hình 3.34 Thông tin chi tiết của diễn viên | 90 |
| Hình 3.35 Màn hình thống kê hệ thống | 90 |
| Hình 3.36 Danh sách phim | 91 |
| Hình 3.37 Màn hình thêm phim | 92 |
| Hình 3.38 Màn hình thêm diễn viên – vai diễn | 92 |
| Hình 3.39 Danh sách diễn viên | 93 |
| Hình 3.40 Màn hình thêm diễn viên | 93 |
| Hình 3.41 Danh sách thẻ loại | 94 |
| Hình 3.42 Màn hình thêm thẻ loại | 94 |
| Hình 3.43 Danh sách gói dịch vụ | 95 |
| Hình 3.44 Màn hình thêm gói dịch vụ | 96 |
| Hình 3.45 Danh sách lịch sử giao dịch | 97 |
| Hình 3.46 Chi tiết giao dịch | 97 |

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

| Ký hiệu | Mô tả |
|----------------|--|
| CBF | Content-Based Filtering - Lọc theo nội dung |
| NCF | Neural Collaborative Filtering – mô hình học sâu cho việc lọc cộng tác bằng cách sử dụng mạng nơ-ron |
| NeuMF | Neural Matrix Factorization - Mô hình lọc cộng tác học sâu (kết hợp MF và MLP). |
| MF | Matrix Factorization – Phân tích ma trận. |
| GMF | Generalized Matrix Factorization – Thành phần của NeuMF, mô phỏng MF bằng nơ-ron. |
| MLP | Multi-Layer Perceptron – Mạng nơ-ron đa lớp. |
| API | Application Programming Interface – Giao diện lập trình ứng dụng |
| JWT | JSON Web Token – Chuẩn mã hóa dữ liệu đăng nhập |
| NLP | Natural Language Processing - Xử lý ngôn ngữ tự nhiên của máy với văn bản hoặc lời nói. |
| TF-IDF | Term Frequency - Inverse Document Frequency – Phép đo lường mức độ quan trọng của một từ trong văn bản |
| CI/CD | Continuous Integration / Continuous Deployment – Tích hợp và triển khai liên tục |
| VNPay | Cổng thanh toán điện tử phổ biến tại Việt Nam |

MỞ ĐẦU

1. Mục đích thực hiện

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ vào lĩnh vực giải trí nói chung và dịch vụ xem phim trực tuyến nói riêng ngày càng trở nên phổ biến. Người dùng hiện nay không chỉ dừng lại ở nhu cầu xem phim đơn thuần mà còn mong muốn có một trải nghiệm giải trí tiện lợi, nhanh chóng và phù hợp với sở thích cá nhân. Tuy nhiên, trên thực tế, nhiều nền tảng xem phim hiện nay vẫn còn khá hạn chế trong việc cá nhân hóa nội dung hiển thị. Việc người dùng phải tự tìm kiếm hoặc chọn ngẫu nhiên khiến cho trải nghiệm trở nên thiếu hiệu quả, tốn thời gian và dễ gây nhàm chán.

Chính vì vậy, đề tài “**Hệ thống xem phim trực tuyến kết hợp với AI gợi ý**” được thực hiện với mục đích xây dựng một hệ thống không chỉ đáp ứng được chức năng xem phim cơ bản, mà còn tích hợp khả năng gợi ý phim thông minh dựa trên hành vi và sở thích của từng người dùng. Mục đích chính là tối ưu hóa trải nghiệm người dùng thông qua việc đề xuất các bộ phim phù hợp, giảm thiểu thời gian tìm kiếm và tăng mức độ hài lòng khi sử dụng hệ thống.

2. Mục tiêu của đề tài

Mục tiêu tổng quát của đề tài là xây dựng thành công một hệ thống xem phim trực tuyến kết hợp với trí tuệ nhân tạo (AI), nhằm nâng cao trải nghiệm giải trí cho người dùng thông qua khả năng gợi ý phim cá nhân hóa. Hệ thống không chỉ cung cấp các chức năng xem phim cơ bản mà còn tự động đưa ra các đề xuất phù hợp với sở thích và hành vi người dùng.

Để đạt được mục tiêu tổng quát trên, đề tài đặt ra các mục tiêu cụ thể như sau:

- ❖ Xây dựng một nền tảng web xem phim trực tuyến có giao diện thân thiện, dễ sử dụng, hoạt động ổn định.
- ❖ Phát triển các chức năng chính như đăng ký, đăng nhập, xem phim, đánh giá phim, thêm vào danh sách yêu thích, bình luận,...
- ❖ Tích hợp hệ thống gợi ý phim dựa trên trí tuệ nhân tạo với hai kỹ thuật chính:
 - **Content-Based Filtering (CBF):** Gợi ý dựa trên nội dung và đặc điểm của phim.
 - **Neural Matrix Factorization (NeuMF):** Gợi ý dựa trên lịch sử tương tác và đánh giá của người dùng.

- ❖ Tối ưu hóa trải nghiệm người dùng bằng cách đề xuất các bộ phim phù hợp với từng cá nhân, giúp tiết kiệm thời gian tìm kiếm.
- ❖ Đảm bảo hệ thống có khả năng mở rộng, dễ nâng cấp trong tương lai và có mức độ bảo mật phù hợp cho người dùng cuối.

3. Phạm vi nghiên cứu

Phạm vi thực hiện của đề tài tập trung vào việc phát triển một ứng dụng web hoạt động như một nền tảng xem phim trực tuyến tích hợp hệ thống đề xuất dựa trên AI. Đề tài hướng đến việc xây dựng và triển khai các chức năng cốt lõi của một hệ thống xem phim hiện đại, bao gồm cả giao diện người dùng và backend xử lý dữ liệu.

4. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài gồm có 3 đối tượng chính

- ❖ **Quản trị viên:** Là người chịu trách nhiệm quản lý toàn bộ hệ thống. Quản trị viên có quyền tạo và quản lý thông tin phim, thể loại, diễn viên, quốc gia, cũng như thiết lập và cập nhật các gói dịch vụ đăng ký, theo dõi số liệu thống kê
- ❖ **Khách:** là những người chưa đăng nhập vào hệ thống, nhưng có thể tìm kiếm, xem thông tin phim và xem được các bộ phim miễn phí.
- ❖ **Người dùng:** Là những cá nhân có nhu cầu xem phim trực tuyến. Người dùng có thể đăng ký tài khoản, lựa chọn gói dịch vụ phù hợp, xem phim theo sở thích, đánh giá phim, bình luận, thêm phim vào danh sách yêu thích và nhận được các gợi ý cá nhân hóa từ hệ thống.

5. Phương pháp nghiên cứu

Để thực hiện đề tài "Hệ thống xem phim trực tuyến kết hợp với AI gợi ý", em đã áp dụng các phương pháp nghiên cứu chính như sau:

- ❖ **Phân tích và tìm hiểu hệ thống:** Tìm hiểu các yêu cầu thực tế của người dùng và nghiệp vụ cốt lõi của một hệ thống xem phim trực tuyến. Nghiên cứu các chức năng phổ biến như đăng ký tài khoản, xem phim, đánh giá, bình luận, gợi ý phim, quản lý nội dung và quản lý gói dịch vụ. Đồng thời, khảo sát các nền tảng xem phim hiện có để học hỏi và phân tích điểm mạnh – điểm yếu.
- ❖ **Thiết kế hệ thống:** Thiết kế kiến trúc hệ thống tổng thể, mô hình cơ sở dữ liệu, luồng hoạt động chính và giao diện người dùng (UI). Việc thiết kế đảm bảo hệ thống có cấu trúc rõ ràng, dễ mở rộng, dễ bảo trì và đáp ứng tốt các yêu cầu đặt ra.

- ❖ **Xây dựng và triển khai hệ thống:** Tiến hành lập trình và hiện thực hóa các chức năng đã thiết kế bằng công nghệ phù hợp (Spring Boot, ReactJS, v.v.). Đồng thời, tích hợp hai kỹ thuật gợi ý là Content-Based Filtering và Neural Matrix Factorization vào hệ thống để cá nhân hóa trải nghiệm người dùng.
- ❖ **Kiểm thử và đánh giá hệ thống:** Thực hiện kiểm thử chức năng và hiệu năng hệ thống trên nhiều kịch bản sử dụng khác nhau. Đánh giá mức độ đáp ứng yêu cầu ban đầu, mức độ chính xác của hệ thống gợi ý và trải nghiệm tổng thể của người dùng khi tương tác với nền tảng.

6. Cấu trúc của đề án

- ❖ Mở đầu
- ❖ Chương 1: Cơ sở lý thuyết
- ❖ Chương 2: Phân tích và thiết kế hệ thống
- ❖ Chương 3: Triển khai hệ thống
- ❖ Kết luận

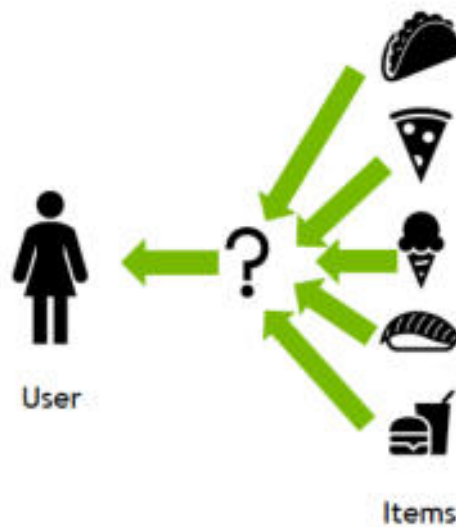
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Tổng quát về hệ thống gợi ý

1.1.1 Hệ thống gợi ý

Hệ thống gợi ý (Recommender System) là một ứng dụng của trí tuệ nhân tạo (AI), thường kết hợp với các thuật toán học máy (Machine Learning), nhằm phân tích dữ liệu lớn (Big Data) để đưa ra các đề xuất sản phẩm hoặc dịch vụ phù hợp với từng người dùng. Hệ thống này có thể sử dụng nhiều yếu tố khác nhau như: lịch sử mua hàng, lịch sử tìm kiếm, thông tin nhân khẩu học, hành vi tương tác,... để tạo nên trải nghiệm cá nhân hóa cho từng người dùng. [1]

Hệ thống gợi ý học từ các hành vi và đặc điểm của người dùng thông qua các dữ liệu như lượt xem, lượt click, lượt thích hoặc lượt mua. Nhờ khả năng dự đoán sở thích và nhu cầu ở mức độ cá nhân, hệ thống gợi ý ngày càng được sử dụng rộng rãi trên các nền tảng cung cấp nội dung và sản phẩm, từ sách, phim ảnh, khóa học đến thời trang,... [1]



Hình 1.1 Mối quan hệ giữa người dùng và các đề xuất

1.1.2 Các loại hệ thống gợi ý phổ biến

❖ Lọc cộng tác (Collaborative Filtering) [1]

Là phương pháp gợi ý dựa trên hành vi tương tác của nhiều người dùng khác nhau. Ý tưởng cốt lõi là: nếu người dùng A và người dùng B từng có sở thích tương đồng (ví dụ Sinh viên: Hoàng Văn Quý Hướng dẫn: ThS. Mai Văn Hà

cùng thích các bộ phim giống nhau), thì những mục mà B yêu thích nhưng A chưa biết đều có thể sẽ phù hợp với A. Phương pháp này thường dựa trên **ma trận tương tác người dùng – sản phẩm**, sử dụng các kỹ thuật như phân rã ma trận (Matrix Factorization) để học đặc trưng tiềm ẩn (latent factors) của người dùng và sản phẩm.

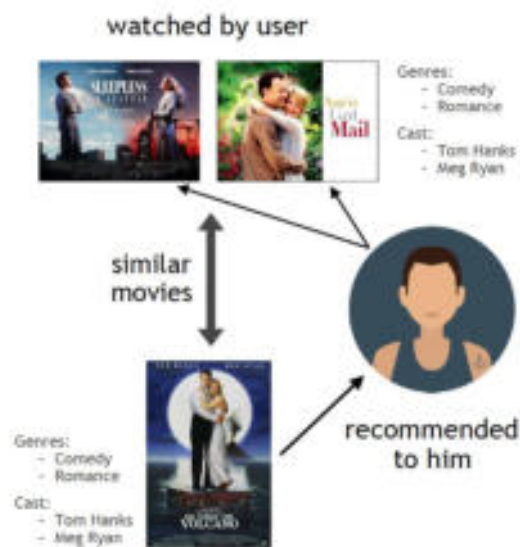


Hình 1.2 Lọc cộng tác

❖ Lọc dựa trên nội dung (Content-Based Filtering) [1]

Gợi ý sản phẩm dựa trên thông tin mô tả sản phẩm và hồ sơ sở thích của người dùng. Ví dụ: nếu người dùng thường xem phim thể loại hành động có sự tham gia của diễn viên X, hệ thống có thể gợi ý thêm các phim tương tự về thể loại hoặc cùng diễn viên đó. Phương pháp này thường xây dựng **vector đặc trưng** có thể cho cả người dùng và sản phẩm, sau đó đo độ tương đồng (ví dụ: cosine similarity) giữa chúng để đưa ra gợi ý.

Content-based Filtering



Hình 1.3 Lọc dựa trên nội dung

❖ Gợi ý dựa trên học sâu (Deep Learning-based Recommendation) [1]

Gần đây, các mô hình dựa trên học sâu (Deep Learning) đang trở thành xu hướng nhờ khả năng mô hình hóa phi tuyến và học biểu diễn tốt hơn. Các mô hình này thường sử dụng **mạng nơ-ron sâu** để học embedding cho người dùng và sản phẩm, sau đó mô hình hóa tương tác giữa chúng để đưa ra dự đoán.

Một số mô hình học sâu tiêu biểu bao gồm:

- Wide & Deep Learning (W&D): Kết hợp giữa mô hình tuyến tính (wide) và mạng nơ-ron sâu (deep).
- DLRM (Deep Learning Recommendation Model): Do Facebook phát triển, tối ưu cho bài toán gợi ý với dữ liệu quy mô rất lớn.
- AutoEncoder & VAE: Học biểu diễn dữ liệu đầu vào thông qua quá trình nén và giải nén.
- BERT4Rec: Ứng dụng kiến trúc Transformer (BERT) để học chuỗi hành vi của người dùng theo thời gian.
- Neural Collaborative Filtering (NCF): Tổng quát hóa mô hình phân rã ma trận truyền thống bằng cách sử dụng mạng nơ-ron để học tương tác phi tuyến giữa người dùng và sản phẩm.

1.1.3 Các mô hình sử dụng trong đề án

Có nhiều mô hình được phát triển cho hệ thống gợi ý, mỗi mô hình có ưu và nhược điểm riêng tùy theo đặc điểm dữ liệu và mục tiêu ứng dụng. Dưới đây là hai mô hình tiêu biểu được sử dụng trong hệ thống của đề tài này.

Content-Based Filtering (CBF) [1]

Content-Based Filtering (CBF) là một trong những phương pháp gợi ý truyền thống phổ biến, dựa trên thông tin mô tả của sản phẩm để cá nhân hóa đề xuất cho từng người dùng. Trong phương pháp này, cả sản phẩm và người dùng đều được biểu diễn dưới dạng vector đặc trưng (feature vector), từ đó hệ thống tính toán độ tương đồng (thường là cosine similarity) để xác định mức độ phù hợp giữa sản phẩm và sở thích người dùng.

CBF đặc biệt hiệu quả trong các hệ thống có thông tin mô tả sản phẩm phong phú nhưng lại thiếu dữ liệu tương tác (ví dụ: người dùng mới, sản phẩm mới). Ngoài ra, mô hình này còn có ưu điểm là dễ triển khai và có khả năng giải thích cao (interpretable), nhờ vào việc dựa trên các đặc trưng có thể quan sát được.

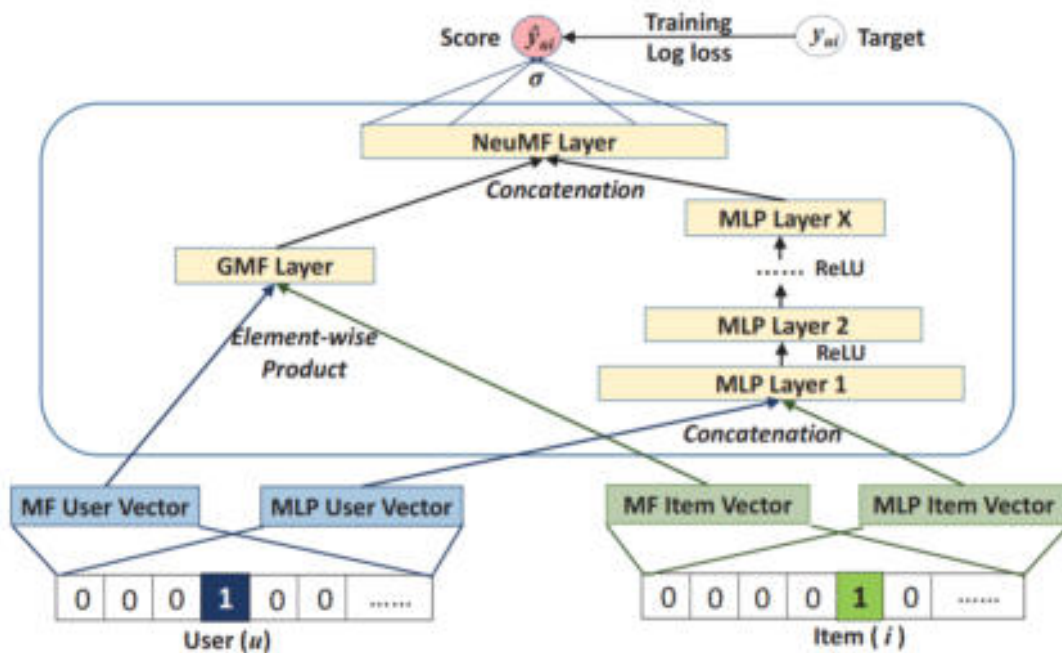
Neural Matrix Factorization (NeuMF) [2]

Neural Matrix Factorization (NeuMF) là một mô hình học sâu thuộc nhóm Neural Collaborative Filtering (NCF), được thiết kế nhằm khắc phục giới hạn tuyến tính của các mô hình phân rã ma trận truyền thống như Matrix Factorization (MF).

Thay vì sử dụng phép nhân vô hướng giữa hai vector đặc trưng tiềm ẩn (latent vectors) của người dùng và sản phẩm như MF, NeuMF khai thác khả năng học phi tuyến bằng cách sử dụng mạng nơ-ron nhiều tầng (Multilayer Perceptron – MLP). Điều này giúp mô hình học được các mối quan hệ phức tạp hơn giữa người dùng và sản phẩm.

Kiến trúc của NeuMF bao gồm hai thành phần chính:

- ❖ **Generalized Matrix Factorization (GMF):** kế thừa ý tưởng của MF để học biểu diễn đặc trưng tiềm ẩn.
- ❖ **Multilayer Perceptron (MLP):** học hàm tương tác phi tuyến giữa các embedding của người dùng và sản phẩm.



Hình 1.4 Cấu trúc của mô hình NeuMF

1.2 Công nghệ và công cụ phát triển

1.2.1 Backend

Spring Boot [\[7\]](#)

Spring Boot là một framework mã nguồn mở được phát triển trên nền tảng Spring Framework. Mục tiêu chính của Spring Boot là giúp lập trình viên xây dựng các ứng dụng web hoặc dịch vụ backend một cách nhanh chóng, dễ dàng, đồng thời giảm thiểu cấu hình phức tạp. Spring Boot cung cấp các cấu hình mặc định, tích hợp sẵn máy chủ web (như Tomcat), hỗ trợ tạo RESTful API, kết nối cơ sở dữ liệu, và dễ dàng mở rộng khi phát triển hệ thống theo kiến trúc microservices hoặc monolithic.

Trong đồ án, Spring Boot được tận dụng để xây dựng RESTful API, đồng thời tích hợp chặt chẽ Spring Security cho các yêu cầu về xác thực và phân quyền, cùng với Spring Data JPA để quản lý tương tác với cơ sở dữ liệu một cách hiệu quả.

Flask

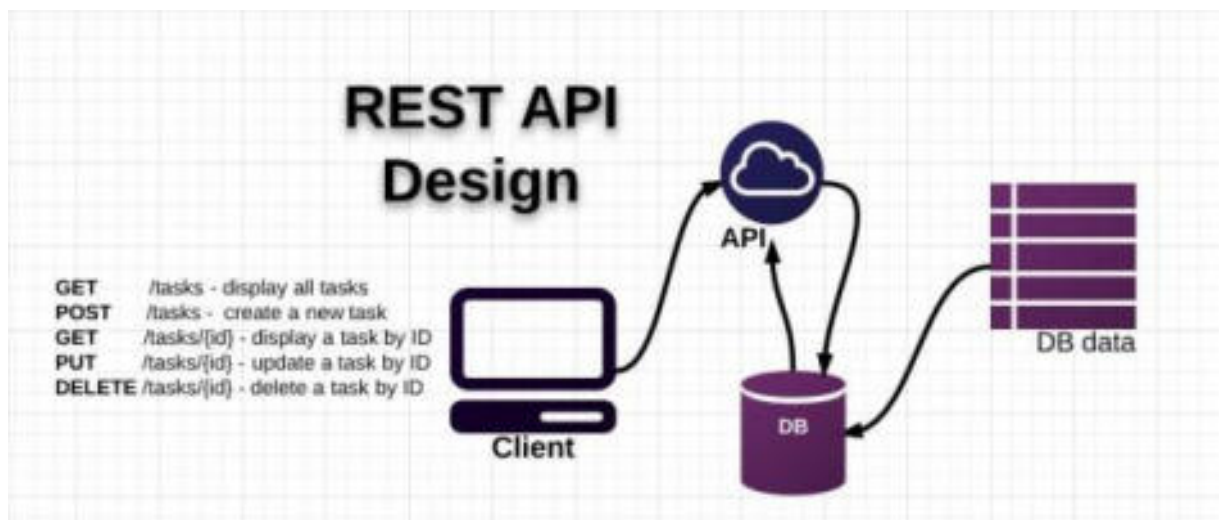
Flask là một microframework mã nguồn mở của Python, được thiết kế để phát triển các ứng dụng web nhỏ gọn hoặc API chuyên biệt một cách nhanh chóng và linh hoạt. Với trọng tâm là sự tối giản và khả năng mở rộng, Flask cung cấp một nền tảng vững chắc cho các tác vụ cần hiệu suất cao và khả năng tùy chỉnh.

Trong dự án, Flask được lựa chọn để phát triển API cho hệ thống gợi ý phim. API này chịu trách nhiệm nhận các yêu cầu và trả về danh sách các Movie ID được đề xuất.

Việc sử dụng Flask cho phép tận dụng triệt để các thư viện mạnh mẽ của Python trong lĩnh vực Machine Learning để xây dựng và triển khai các thuật toán gợi ý hiệu quả.

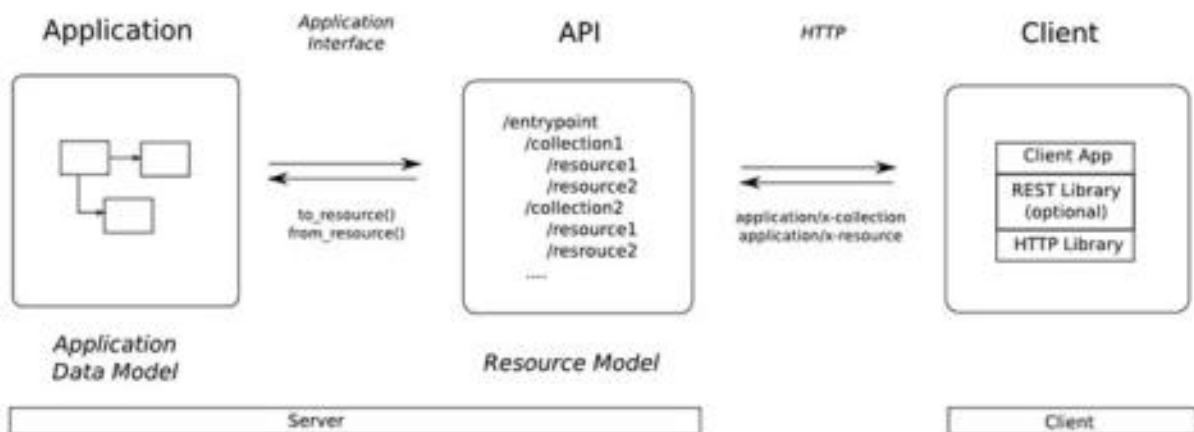
RESTful API [4]

Một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 1.5 Cơ chế hoạt động của RESTful API

Cách hoạt động



Hình 1.6 Sơ đồ hoạt động của RESTful API

REST hoạt động dựa trên các phương thức HTTP cơ bản, trong đó mỗi phương thức tương ứng với một thao tác cụ thể trên tài nguyên:

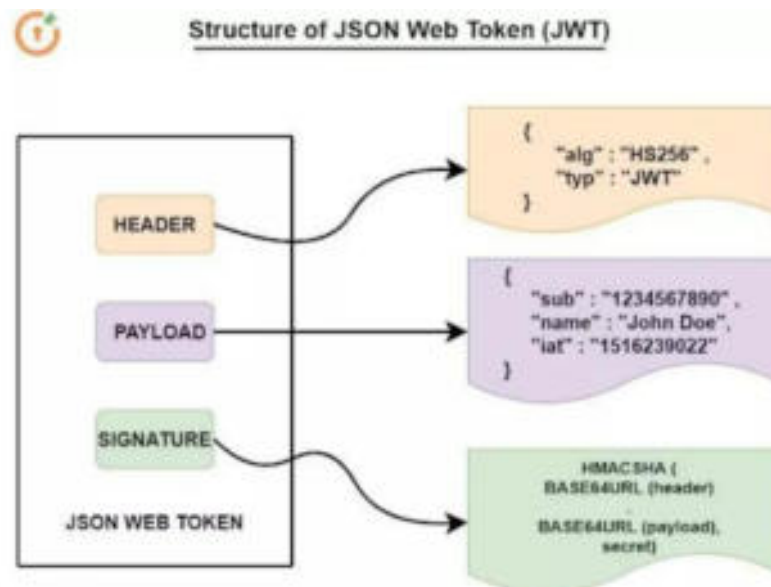
- ❖ **GET:** Dùng để truy xuất thông tin của một tài nguyên hoặc danh sách tài nguyên mà không làm thay đổi trạng thái của tài nguyên đó.
- ❖ **POST:** Dùng để tạo mới một tài nguyên trên máy chủ, thường đi kèm với dữ liệu gửi lên để khởi tạo tài nguyên đó.
- ❖ **PUT:** Dùng để cập nhật hoặc thay thế hoàn toàn một tài nguyên đã tồn tại với dữ liệu mới.
- ❖ **DELETE:** Dùng để xóa bỏ một tài nguyên cụ thể khỏi hệ thống.

JWT

JWT là một chuẩn mã hóa dùng để truyền tải thông tin người dùng một cách an toàn dưới dạng chuỗi ký tự. Token này thường được sử dụng để xác thực và phân quyền trong các hệ thống web hiện đại, thay thế cho cơ chế session truyền thống.

JWT gồm 3 phần: Header, Payload (chứa thông tin người dùng), và Signature (giúp xác minh tính toàn vẹn). Ưu điểm của JWT là không cần lưu trạng thái trên server (stateless), dễ tích hợp với API RESTful và giúp hệ thống mở rộng tốt hơn.

Trong đề tài này, JWT được dùng kết hợp với Spring Security để xác thực người dùng và kiểm soát truy cập đến các tài nguyên của hệ thống.

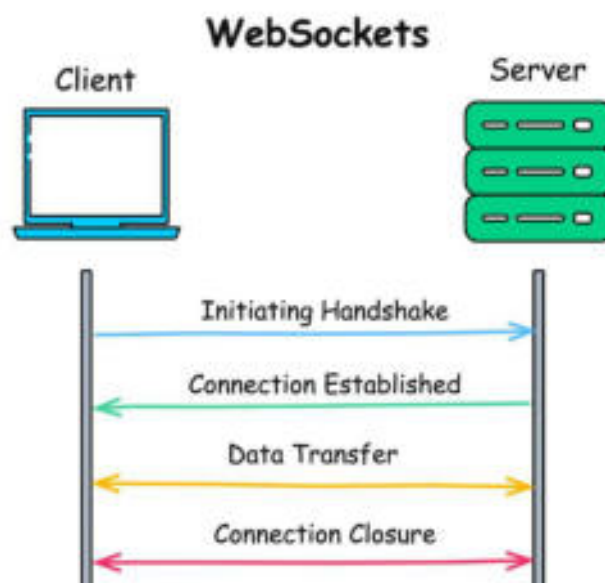


Hình 1.7 Cấu trúc của một JWT

WebSocket [19]

WebSocket là một giao thức truyền thông hai chiều (full-duplex) hoạt động trên nền TCP, cho phép kết nối liên tục giữa client và server. Không giống như HTTP – vốn chỉ truyền dữ liệu theo hướng client gửi – server phản hồi, WebSocket cho phép cả hai phía chủ động gửi và nhận dữ liệu bất kỳ lúc nào sau khi kết nối được thiết lập.

Với độ trễ thấp và khả năng truyền dữ liệu theo thời gian thực, WebSocket thường được sử dụng trong các ứng dụng yêu cầu cập nhật liên tục như: chat trực tuyến, thông báo thời gian thực, game online, hoặc theo dõi dữ liệu liên tục (live tracking).



Hình 1.8 Cách hoạt động của websocket

1.2.2 Frontend

ReactJS [5]

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. React tập trung vào việc phát triển các thành phần giao diện (components) có khả năng tái sử dụng và dễ quản lý trạng thái (state), giúp việc xây dựng giao diện trở nên linh hoạt và hiệu quả hơn.

Điểm nổi bật của React là Virtual DOM – cơ chế giúp tối ưu hiệu suất bằng cách hạn chế việc thao tác trực tiếp lên DOM thật, từ đó mang lại trải nghiệm người dùng mượt mà hơn. React hỗ trợ mô hình luồng dữ liệu một chiều (one-way data flow), giúp quản lý dữ liệu dễ hơn và đơn giản hóa quá trình gỡ lỗi ứng dụng

RTK Query [\[9\]](#)

RTK Query là một thư viện tích hợp sẵn trong Redux Toolkit, giúp đơn giản hóa việc gọi API và quản lý trạng thái dữ liệu từ server. Thay vì phải tự viết các logic xử lý loading, error, success, cache thủ công hoặc tạo reducer, RTK Query giúp tự động hóa hầu hết quy trình này.

Một số đặc điểm nổi bật:

- ❖ Caching thông minh: Tự động lưu và làm mới dữ liệu khi cần (re-fetch).
- ❖ Hỗ trợ debug hiệu quả: Tích hợp sẵn với Redux DevTools.
- ❖ Tối ưu hiệu suất: Giảm thiểu việc re-render không cần thiết.
- ❖ Đơn giản hóa mã nguồn: So với việc sử dụng useEffect kết hợp axios hoặc fetch, RTK Query giúp mã nguồn gọn gàng, dễ kiểm soát hơn và giảm khả năng sai sót trong xử lý bất đồng bộ.

Ant Design [\[8\]](#)

Ant Design (gọi tắt là AntD) là một thư viện giao diện người dùng (UI) được phát triển bởi Alibaba. Thư viện cung cấp sẵn nhiều thành phần giao diện hiện đại như bảng dữ liệu, biểu mẫu, nút bấm, hộp thoại,... giúp đẩy nhanh tiến độ xây dựng giao diện với phong cách nhất quán và chuyên nghiệp.

Một số đặc điểm nổi bật:

- ❖ Bộ component phong phú, dễ sử dụng và có thể tùy biến.
- ❖ Thiết kế hiện đại, đặc biệt phù hợp cho các hệ thống quản trị (admin dashboard).
- ❖ Hỗ trợ tốt TypeScript và tích hợp mượt mà với React.

Tailwind CSS

Tailwind CSS là một framework CSS theo phương pháp utility-first, nghĩa là cung cấp sẵn các lớp tiện ích (utility classes) để thiết kế giao diện trực tiếp trong mã HTML mà không cần viết CSS thủ công.

Tailwind CSS hướng đến việc tăng hiệu quả phát triển giao diện, giảm sự phụ thuộc vào file CSS riêng và tăng tính tái sử dụng. Ngoài ra, Tailwind cho phép tùy chỉnh theme, màu sắc, breakpoint,... thông qua file cấu hình, giúp hệ thống giao diện trở nên nhất quán và dễ mở rộng.

Một số đặc điểm nổi bật:

- ❖ Hỗ trợ thiết kế nhanh chóng mà không cần rời khỏi file HTML/JSX.
- ❖ Giảm thiểu việc trùng lặp và khó quản lý trong CSS truyền thống.

- ❖ Dễ dàng tùy chỉnh và mở rộng giao diện theo hệ thống thiết kế riêng.
- ❖ Tích hợp tốt với các framework như React, Vue, hoặc các thư viện UI như Ant Design.

1.2.3 Cơ sở dữ liệu & lưu trữ

MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất thế giới, được phát triển và duy trì bởi Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để quản lý và thao tác dữ liệu trong các bảng có cấu trúc rõ ràng với các hàng và cột.

MySQL nổi bật với khả năng xử lý nhanh, độ tin cậy cao và khả năng lưu trữ lượng lớn dữ liệu, rất phù hợp cho các ứng dụng web và doanh nghiệp. Bên cạnh đó, MySQL còn hỗ trợ các tính năng tiên tiến như giao dịch (transactions), khóa (locking), sao lưu và phục hồi dữ liệu, cùng với khả năng mở rộng linh hoạt để đáp ứng nhu cầu phát triển của hệ thống

Redis [\[6\]](#)

Redis là một hệ quản trị cơ sở dữ liệu NoSQL dạng key-value mã nguồn mở, nổi bật với khả năng xử lý dữ liệu cực nhanh nhờ lưu trữ toàn bộ dữ liệu trên bộ nhớ RAM thay vì ổ cứng như các hệ quản trị truyền thống. Redis hỗ trợ nhiều kiểu dữ liệu đa dạng như chuỗi (strings), danh sách (lists), tập hợp (sets), bảng băm (hashes), và cả các cấu trúc dữ liệu phức tạp như sorted sets.

Với hiệu suất cao và độ trễ thấp, Redis thường được sử dụng làm bộ nhớ đệm (cache), hàng đợi (message broker), hoặc lưu trữ phiên làm việc (session store) trong các ứng dụng web hiện đại. Ngoài ra, Redis còn cung cấp các tính năng như sao chép dữ liệu, bền vững dữ liệu (persistence), và hỗ trợ clustering để mở rộng quy mô hệ thống.

Azure Blob Storage [\[10\]](#)

Azure Blob Storage là dịch vụ lưu trữ đối tượng (object storage) trong nền tảng đám mây Microsoft Azure, cho phép lưu trữ dữ liệu phi cấu trúc với khả năng mở rộng linh hoạt, độ bền cao và tính sẵn sàng vượt trội. Dữ liệu được tổ chức dưới dạng các **blob** trong các **container**, thích hợp để lưu trữ hình ảnh, video, tài liệu, sao lưu, phân phối nội dung số và các tệp có kích thước lớn.

Azure Blob hỗ trợ nhiều cấp độ truy cập (hot, cool, archive) để tối ưu chi phí lưu trữ, đồng thời cung cấp các tính năng như mã hóa dữ liệu, kiểm soát truy cập chi tiết (Access Policies), tích hợp với các dịch vụ khác trong hệ sinh thái Azure và hỗ trợ quản lý vòng đời dữ liệu tự động.

Trong hệ thống, Azure Blob Storage được sử dụng để lưu trữ hình ảnh người dùng và tài nguyên tĩnh. Các tệp này được thiết lập ở chế độ **public**, cho phép truy cập trực tiếp thông qua URL mà không cần xác thực. Cách làm này giúp frontend hiển thị hình ảnh nhanh chóng, giảm tải cho backend và đơn giản hóa quá trình xử lý ảnh.

1.2.4 Tích hợp thanh toán

VNPAY là một cổng thanh toán điện tử tại Việt Nam, cung cấp giải pháp thanh toán trực tuyến an toàn và tiện lợi cho website và ứng dụng di động. VNPAY hỗ trợ nhiều phương thức thanh toán như thẻ nội địa (ATM), thẻ quốc tế (Visa, MasterCard), mã QR, và ví điện tử. Trong hệ thống, VNPAY được tích hợp để xử lý giao dịch thanh toán của người dùng một cách nhanh chóng và bảo mật, giúp nâng cao trải nghiệm và độ tin cậy khi sử dụng dịch vụ.

Trong đề án, em sử dụng tài khoản sandbox do VNPAY cung cấp để tích hợp và kiểm thử cổng thanh toán mà không phát sinh giao dịch thật. Môi trường sandbox mô phỏng quá trình thanh toán thực tế, cho phép gửi yêu cầu, nhận phản hồi và xử lý kết quả như khi sử dụng hệ thống thật. Việc tích hợp này phục vụ cho chức năng người dùng mua gói dịch vụ để xem các phim tính phí.

1.2.5 Git và Github

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) mã nguồn mở, giúp theo dõi và quản lý sự thay đổi trong mã nguồn của dự án. Git cho phép nhiều người cùng làm việc trên một dự án một cách hiệu quả, hỗ trợ các thao tác như commit, branch, merge, revert, giúp kiểm soát lịch sử phát triển phần mềm và dễ dàng quay lại các phiên bản trước đó.

GitHub là một nền tảng lưu trữ mã nguồn trực tuyến sử dụng Git. Ngoài việc hỗ trợ quản lý phiên bản, GitHub còn cung cấp các tính năng cộng tác mạnh mẽ như pull request, issue tracking, review code, CI/CD, và GitHub Actions, giúp nhóm phát triển làm việc nhóm hiệu quả hơn.

Một số đặc điểm nổi bật:

- ❖ Phân nhánh (branching) linh hoạt, dễ thử nghiệm tính năng mới mà không ảnh hưởng đến nhánh chính.
- ❖ Hợp nhất (merging) mã dễ dàng giữa các nhánh.
- ❖ Theo dõi lịch sử thay đổi chi tiết, giúp dễ dàng gỡ lỗi hoặc xem ai thay đổi gì.
- ❖ Làm việc offline, do Git lưu toàn bộ lịch sử ở máy cục bộ.
- ❖ Tích hợp với GitHub để chia sẻ, cộng tác và quản lý dự án công khai hoặc riêng tư.\

1.2.6 Deployment

Docker [\[11\]](#)

Docker là một nền tảng mã nguồn mở giúp xây dựng, triển khai và chạy ứng dụng bên trong các container – một dạng môi trường ảo hóa nhẹ. Không giống như máy ảo (virtual machine) truyền thống cần hệ điều hành riêng, container chỉ chứa những thành phần cần thiết để ứng dụng chạy, giúp tiết kiệm tài nguyên và khởi động nhanh.

- ❖ Đặc điểm của Docker
 - Tính di động cao: Container có thể chạy trên bất kỳ máy nào có Docker Engine, bất kể hệ điều hành là gì.
 - Cách ly môi trường: Mỗi container hoạt động độc lập, không gây xung đột với ứng dụng khác.
 - Dễ đóng gói và triển khai: Dockerfile giúp mô tả cách xây dựng ứng dụng và môi trường của nó, từ đó tạo ra Docker image có thể tái sử dụng.
 - Tối ưu tài nguyên: So với máy ảo, container khởi chạy nhanh và sử dụng ít bộ nhớ hơn.
 - Hỗ trợ chuẩn hóa môi trường phát triển và production: Tránh lỗi "chạy được trên máy em nhưng không chạy trên máy server".
- ❖ Thành phần chính
 - Docker Engine: công cụ chạy và quản lý container.
 - Dockerfile: file mô tả cách build một Docker image cho ứng dụng.
 - Docker Image: bản mẫu của ứng dụng được tạo ra từ Dockerfile
 - Docker Container: là thể hiện từ một Docker image hoạt động như một đơn vị ứng dụng độc lập

Docker Compose [\[13\]](#)

Docker Compose là công cụ hỗ trợ định nghĩa và quản lý nhiều container một cách dễ dàng thông qua file cấu hình docker-compose.yml.

Nó cho phép khởi tạo và liên kết nhiều dịch vụ trong hệ thống (như backend, frontend, database, cache...) chỉ bằng một lệnh (docker compose up –build), đồng thời đảm bảo các thành phần có thể giao tiếp với nhau nội bộ.

Các đặc điểm của Docker Compose:

- Quản lý nhiều service đồng bộ: Tất cả dịch vụ được định nghĩa và khởi chạy cùng nhau Cấu hình mạng nội bộ giữa các container: Các container có thể giao tiếp với nhau thông qua tên service.
- Hỗ trợ biến môi trường, volume, port,...: Dễ dàng cấu hình cho từng dịch vụ.
- Phù hợp với môi trường phát triển và production: Đặc biệt hữu ích trong CI/CD và cloud server như Azure.

DockerHub [\[12\]](#)

DockerHub là một dịch vụ lưu trữ image do Docker cung cấp, cho phép lập trình viên lưu trữ, quản lý và phân phối các Docker image thông qua internet. Đây là một phần quan trọng trong quy trình triển khai hiện đại, giúp kết nối giữa quá trình build và deploy ứng dụng.

Đặc điểm của DockerHub:

- Lưu trữ tập trung: DockerHub là nơi lưu trữ các image ứng dụng, giúp dễ dàng quản lý và chia sẻ giữa các thành viên trong nhóm hoặc giữa các hệ thống khác nhau.
- Hỗ trợ tích hợp CI/CD: Có thể tích hợp với các hệ thống như GitHub Actions để tự động build và đẩy image sau mỗi lần cập nhật mã nguồn.
- Quản lý phiên bản: Hỗ trợ gắn thẻ (tag) cho từng phiên bản image, giúp kiểm soát và triển khai ứng dụng theo từng giai đoạn cụ thể.
- Public và Private repository: Cho phép lựa chọn giữa việc công khai hoặc giữ riêng tư các image ứng dụng tùy theo nhu cầu bảo mật.
- Khả năng truy cập linh hoạt: Các máy chủ triển khai (server, máy ảo...) có thể dễ dàng pull image từ DockerHub để chạy ứng dụng chỉ với một lệnh đơn giản.

Github Action [\[14\]](#)

GitHub Actions là một nền tảng tích hợp sẵn trong GitHub, cho phép tự động hóa các quy trình phát triển phần mềm, đặc biệt là quy trình CI/CD (Continuous Integration / Continuous Deployment).

Thay vì thực hiện các bước build, test, deploy thủ công, GitHub Actions cho phép lập trình viên mô tả toàn bộ quy trình này trong các tệp cấu hình .yml, đặt tại thư mục .github/workflows trong repository.

Đặc điểm của GitHub Actions:

- Tự động hóa toàn bộ quy trình phát triển: từ kiểm tra mã nguồn, chạy test, build ứng dụng đến triển khai (deploy) lên máy chủ hoặc nền tảng cloud.
- Hỗ trợ nhiều sự kiện kích hoạt: như push code, pull request, tạo release, hoặc chạy định kỳ (schedule).
- Chạy trên môi trường tùy chỉnh: hỗ trợ các hệ điều hành phổ biến và Docker container riêng biệt.
- Tích hợp dễ dàng với nền tảng bên ngoài: như Docker Hub, Azure, AWS, GCP,... để triển khai hoặc phân phối ứng dụng.
- Hệ sinh thái mở, dễ mở rộng: có thể sử dụng action sẵn có từ Marketplace hoặc tự viết action riêng theo nhu cầu.

Azure Virtual Machine [\[16\]](#)

Azure Virtual Machine (VM) là dịch vụ máy ảo do Microsoft Azure cung cấp. Đây là môi trường hạ tầng ảo (IaaS – Infrastructure as a Service) cho phép người dùng cài đặt hệ điều hành, phần mềm và triển khai các dịch vụ như trên một máy chủ vật lý.

Đặc điểm của Azure Virtual Machine

- Tự do cấu hình: Có thể chọn hệ điều hành (Ubuntu, Windows,...), RAM, CPU, SSD tùy theo nhu cầu.
- Khả năng mở rộng linh hoạt: Dễ dàng nâng cấp cấu hình để phù hợp với tải hệ thống.
- Tương thích cao với các công cụ DevOps: như Docker, Docker Compose, GitHub Actions.
- Tích hợp bảo mật và giám sát từ Azure: như firewall, logging, snapshot...
- Khả năng truy cập SSH từ xa: giúp kết nối và thao tác như máy chủ vật lý.

Trong hệ thống xem phim trực tuyến, Azure VM đóng vai trò là máy chủ chính nơi chứa toàn bộ hệ thống đã được container hoá và triển khai.

1.2.7 Xử lý ngôn ngữ tự nhiên

❖ TF-IDF [17]

Term Frequency – Inverse Document Frequency (TF-IDF) là một kỹ thuật phổ biến trong lĩnh vực Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP), được sử dụng để biến đổi dữ liệu văn bản thành vector số học, phục vụ cho việc tính toán độ tương đồng giữa các văn bản.

Mục tiêu chính của TF-IDF là đánh giá mức độ quan trọng của một từ đối với một tài liệu cụ thể trong một tập hợp nhiều tài liệu. Từ đó, những từ ngữ có giá trị phân biệt cao sẽ được giữ lại, còn những từ quá phổ biến sẽ bị giảm trọng số.

TF-IDF gồm hai thành phần:

- TF – Term Frequency (Tần suất của từ)

$$TF(t, d) = \frac{\text{số lần từ } t \text{ xuất hiện trong tài liệu } d}{\text{tổng số từ trong } d}$$

Đo lường số lần một từ xuất hiện trong một tài liệu, nhằm thể hiện mức độ phổ biến của từ đó trong chính tài liệu.

- IDF – Inverse Document Frequency (Tần suất nghịch đảo tài liệu)

Đo lường mức độ quan trọng của một từ trong toàn bộ tập dữ liệu. Nếu một từ xuất hiện ở nhiều tài liệu thì nó kém quan trọng.

$$IDF(t) = \log\left(\frac{N}{1 + DF(t)}\right)$$

Trong đó:

- N: tổng số tài liệu trong tập dữ liệu
- DF(t): số tài liệu chứa từ t
- Cộng thêm 1 để tránh chia cho 0

Công thức tổng quát:

$$TF-IDF(t, d) = TF(t, d) \times IDF(t)$$

Có thể thấy rằng, mỗi từ t trong tài liệu d sẽ được gán một trọng số TF-IDF. Trọng số này phản ánh mức độ quan trọng của từ đó đối với tài liệu hiện tại so với toàn bộ tập dữ liệu. Cụ thể, những từ có tần suất xuất hiện cao trong tài liệu (TF lớn), đồng

thời ít xuất hiện trong các tài liệu khác (IDF lớn), sẽ có giá trị TF-IDF cao. Điều này giúp các từ mang tính đặc trưng cao được làm nổi bật, qua đó hỗ trợ hiệu quả cho các bài toán như tìm kiếm thông tin hoặc hệ thống gợi ý.

❖ Cosine Similarity [\[18\]](#)

Cosine Similarity là phương pháp tính độ tương đồng giữa hai văn bản được biểu diễn bằng vector. Nó dựa trên góc giữa hai vector, với giá trị nằm trong khoảng từ 0 (không liên quan) đến 1 (tương đồng tuyệt đối).

Công thức tính độ tương đồng cosin giữa hai vector văn bản A và B:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Trong đó:

- $A \cdot B$: tích vô hướng giữa 2 vector
- $\|A\|, \|B\|$: độ dài (chuẩn) của từng vector

Giá trị của cosine similarity nằm trong khoảng từ 0 đến 1 (đối với vector không âm), với ý nghĩa:

Gần 1: hai vector (hai mô tả phim) gần như giống nhau → phim tương tự

Gần 0: hai vector không có điểm chung → phim không liên quan

Nhờ hai phương Trong hệ thống gợi ý nội dung, Cosine Similarity được sử dụng để xác định các bộ phim có nội dung tương đồng dựa trên mô tả hoặc đặc điểm nội dung được chuyển thành vector qua TF-IDF.

1.2.8 Recbole

RecBole (Recommendation Baselines Library) là một thư viện mã nguồn mở được phát triển bởi nhóm nghiên cứu tại Đại học Khoa học và Công nghệ Hồng Kông (HKUST), nhằm hỗ trợ việc xây dựng, huấn luyện và đánh giá các mô hình hệ thống gợi ý một cách hiệu quả và linh hoạt.

Thư viện này được thiết kế với mục tiêu chuẩn hóa quy trình thực nghiệm trong lĩnh vực gợi ý, đồng thời cung cấp một tập hợp lớn các mô hình phổ biến, từ các mô hình truyền thống như **Matrix Factorization**, **ItemKNN**, đến các mô hình hiện đại sử dụng **Deep Learning** như **NeuMF**, **BERT4Rec**, **LightGCN**,...

Các đặc điểm nổi bật của RecBole:

- ❖ **Hỗ trợ đa dạng mô hình:** Hơn 70 mô hình gợi ý được tích hợp sẵn, chia thành nhiều nhóm như General Recommendation, Sequential Recommendation, Context-aware Recommendation,...
- ❖ **Cấu hình linh hoạt:** Người dùng có thể tùy chỉnh tham số mô hình, bộ dữ liệu, chiến lược đánh giá... thông qua file cấu hình hoặc dòng lệnh.
- ❖ **Tiền xử lý dữ liệu tự động:** RecBole hỗ trợ chuẩn hóa dữ liệu, chuyển đổi định dạng và sinh các đặc trưng cần thiết cho mô hình.
- ❖ **Hệ thống đánh giá chuẩn:** Cung cấp các chỉ số đánh giá phổ biến như Hit Ratio, NDCG, MAP, Precision, Recall,...

RecBole được xây dựng trên nền tảng **PyTorch** và có khả năng mở rộng cao, giúp các nhà nghiên cứu dễ dàng thử nghiệm mô hình mới hoặc cải tiến mô hình hiện có. Trong phạm vi đề án này, thư viện RecBole được sử dụng để huấn luyện và đánh giá mô hình **NeuMF**, phục vụ cho việc so sánh hiệu quả giữa các phương pháp gợi ý.

1.3 Kết luận chương

Chương này đã trình bày các cơ sở lý thuyết liên qua đến việc triển khai đề tài “Hệ thống xem phim trực tuyến kết hợp AI gợi ý”. Nội dung tập trung vào các khía cạnh sau:

- Tổng quan về hệ thống đề xuất (recommendation system).
- Các phương pháp triển khai trên thực tế, bao gồm lọc theo nội dung (content-based filtering), lọc cộng tác (collaborative filtering) và các phương pháp áp dụng mô hình học sâu (deep learning).
- Đề xuất phương pháp triển khai chính được áp dụng trong đề tài là sự kết hợp giữa
 - lọc theo nội dung và các thuộc tính đặc trưng của phim ,
 - Cùng với mô hình NeuMF(Neural Matrix Factorization) nhằm nâng cao hiệu quả gợi ý.
- Ngoài ra chương cũng trình bày tổng quản các công nghệ được sử dụng trong đề án: Spring Boot, Flask, ReactJS, kiến trúc RESTful API, Websocket, ...
- Bên cạnh đó là các công cụ và nền tảng hỗ trợ triển khai hệ thống như: Docker, Docker Compose, GitHub Actions và Azure Virtual Machine.

- Thư viện RecBole được sử dụng để huấn luyện các mô hình học sâu phục vụ cho hệ thống gợi ý.

Những kiến thức lý thuyết và công nghệ được trình bày ở chương này là nền tảng quan trọng cho quá trình phân tích và thiết kế hệ thống sẽ được trình bày ở chương tiếp theo.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Phân tích yêu cầu đề tài

Đề tài hướng đến việc xây dựng một website xem phim trực tuyến có tích hợp tính năng gợi ý phim dựa trên các đặc điểm của phim và hành vi người dùng. Bên cạnh các chức năng cơ bản như tìm kiếm, xem phim, đánh giá và bình luận, hệ thống bổ sung khả năng đề xuất phim có đặc điểm tương tự hoặc được nhiều người dùng quan tâm. Mục tiêu là cải thiện trải nghiệm người dùng bằng cách hỗ trợ họ dễ dàng tìm thấy các bộ phim phù hợp với sở thích cá nhân hơn, thay vì phải tự tìm kiếm như các nền tảng thông thường

2.2 Yêu cầu chức năng

2.2.1 Khách

- ❖ Xem thông tin phim
- ❖ Xem các phim miễn phí
- ❖ Tìm kiếm phim theo tên hoặc bộ lọc
- ❖ Xem thông tin các gói dịch vụ
- ❖ Xem bình luận và đánh giá phim
- ❖ Nhận gợi ý từ các phim tương tự nhau

2.2.2 Người dùng

- ❖ Xác thực người dùng:
 - Đăng nhập tài khoản bằng email và mật khẩu hoặc bằng tài khoản google
 - Đăng ký tài khoản và xác thực tài khoản đăng ký qua OTP
 - Quên mật khẩu: Gửi link đặt lại mật khẩu thông qua email
- ❖ Quản lý tài khoản người dùng:
 - Cập nhập thông tin cá nhân
 - Thay đổi mật khẩu
- ❖ Quản lý phim:
 - Xem thông tin phim và nội dung phim
 - Tìm kiếm phim theo tên hoặc theo bộ lọc
- ❖ Quản lý danh sách phim yêu thích:
 - Thêm hoặc xóa phim khỏi danh sách yêu thích

- ❖ Quản lý gói dịch vụ:
 - Mua gói dịch vụ
 - Xem lịch sử mua dịch vụ
- ❖ Tương tác phim
 - Bình luận phim
 - Đánh giá phim
- ❖ Nhận gợi ý phim tương tự, và theo lịch sử đánh giá của người dùng

2.2.3 Quản trị viên

- ❖ Quản lý diễn viên
- ❖ Quản lý thể loại phim
- ❖ Quản lý các gói dịch vụ
- ❖ Quản lý phim
- ❖ Thống kê:
 - Thống kê về số lượng người dùng, số lượng các loại phim, các thể loại phim trong hệ thống
 - Thống kê doanh thu theo tháng, theo năm

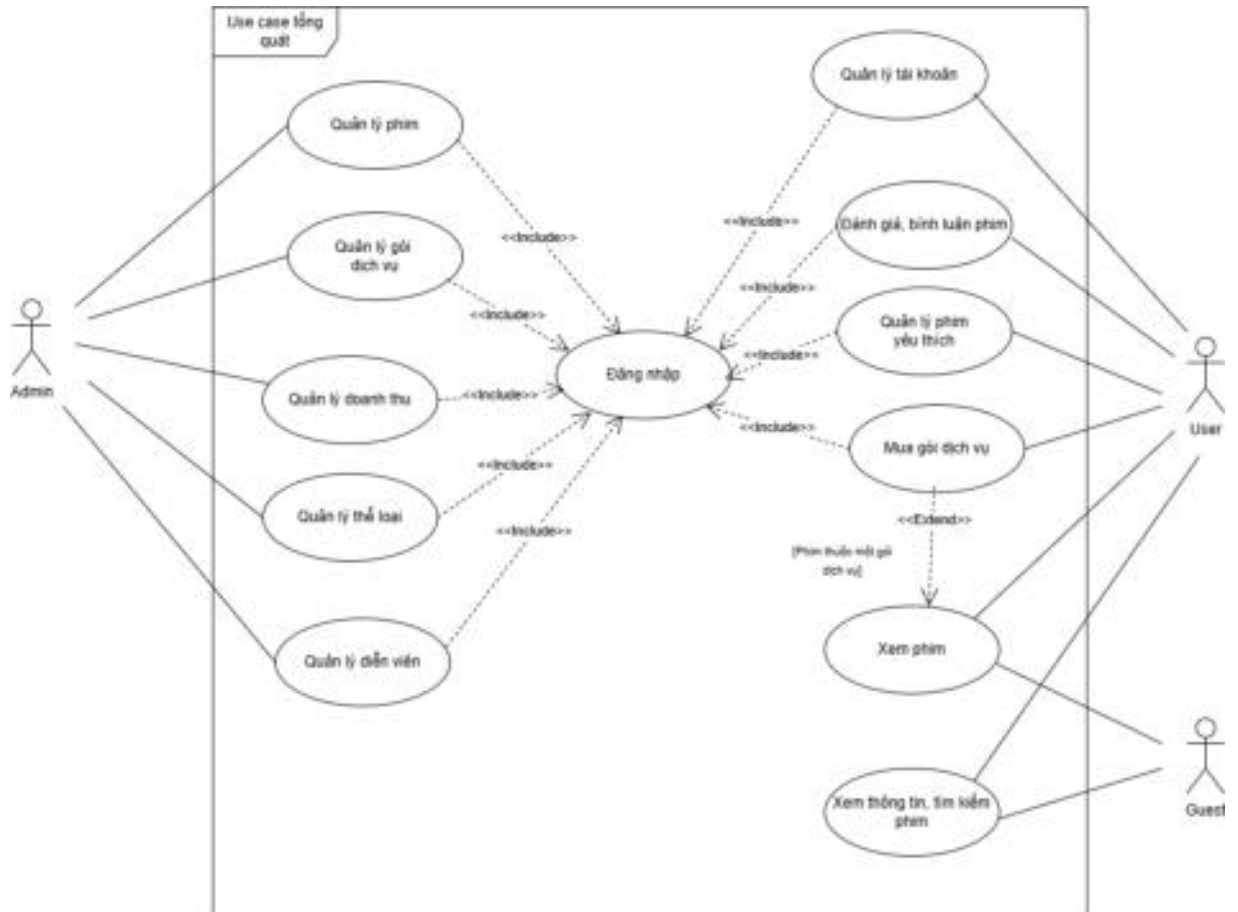
2.3 Các yêu cầu phi chức năng

- ❖ Tính bảo mật
 - Hệ thống hạn chế các chức năng dành cho người dùng chưa đăng nhập
 - Mật khẩu được mã hóa và lưu trữ an toàn trong cơ sở dữ liệu
 - Phân quyền người dùng giữa quản trị viên và các người dùng thông thường
 - Sử dụng JWT để xác thực và phân quyền người dùng với token có thời gian hết hạn và cơ chế làm mới
- ❖ Tính dễ sử dụng
 - Hệ thống có giao diện thân thiện, cho phép người dùng dễ dàng tìm kiếm và xem phim chỉ với vài thao tác
- ❖ Hiệu năng:
 - Hệ thống đảm bảo thời gian phản hồi tương đối nhanh, tối ưu hóa truy vấn và giao tiếp giữa client và server nhằm mang lại trải nghiệm mượt mà cho người dùng.
- ❖ Tính mở rộng:
 - Tạo phòng chiếu phim chung (watch party)
 - Chức năng chat trực tuyến giữa người dùng và quản trị viên

2.4 Thiết kế hệ thống

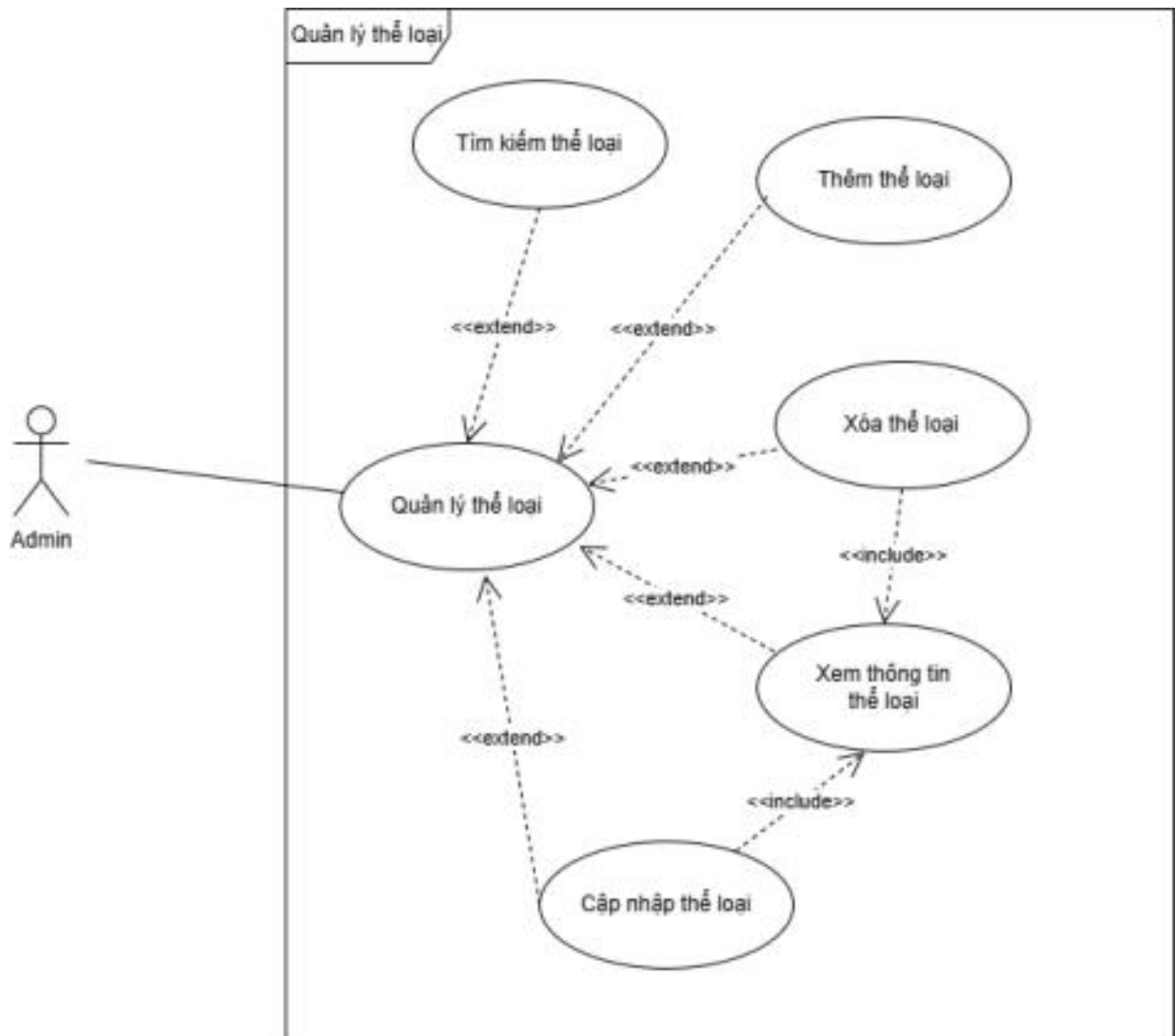
2.4.1 Sơ đồ use case

- Sơ đồ use case tổng quát



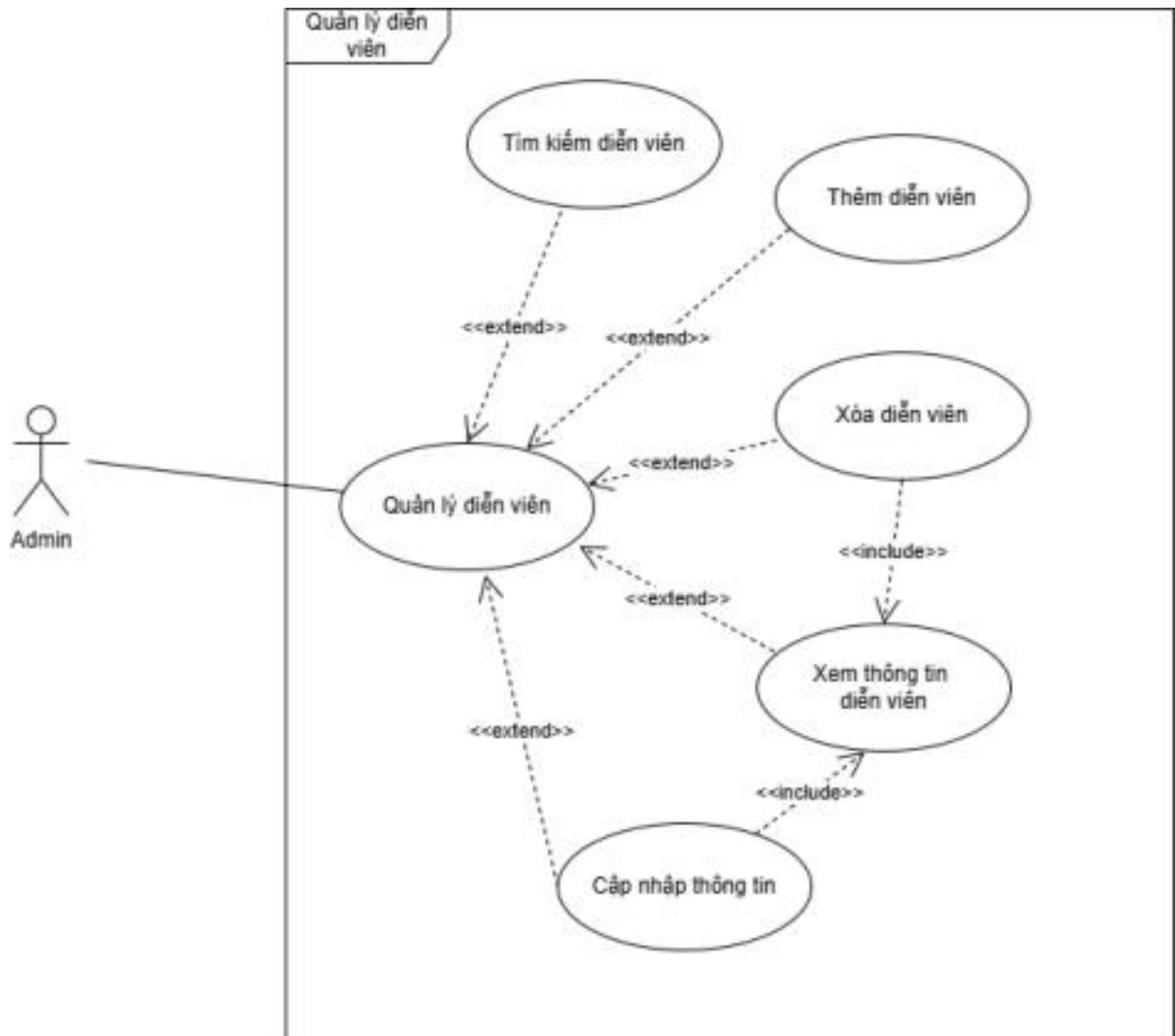
Hình 2.1 Sơ đồ use case tổng quát

- Sơ đồ use case quản lý thể loại



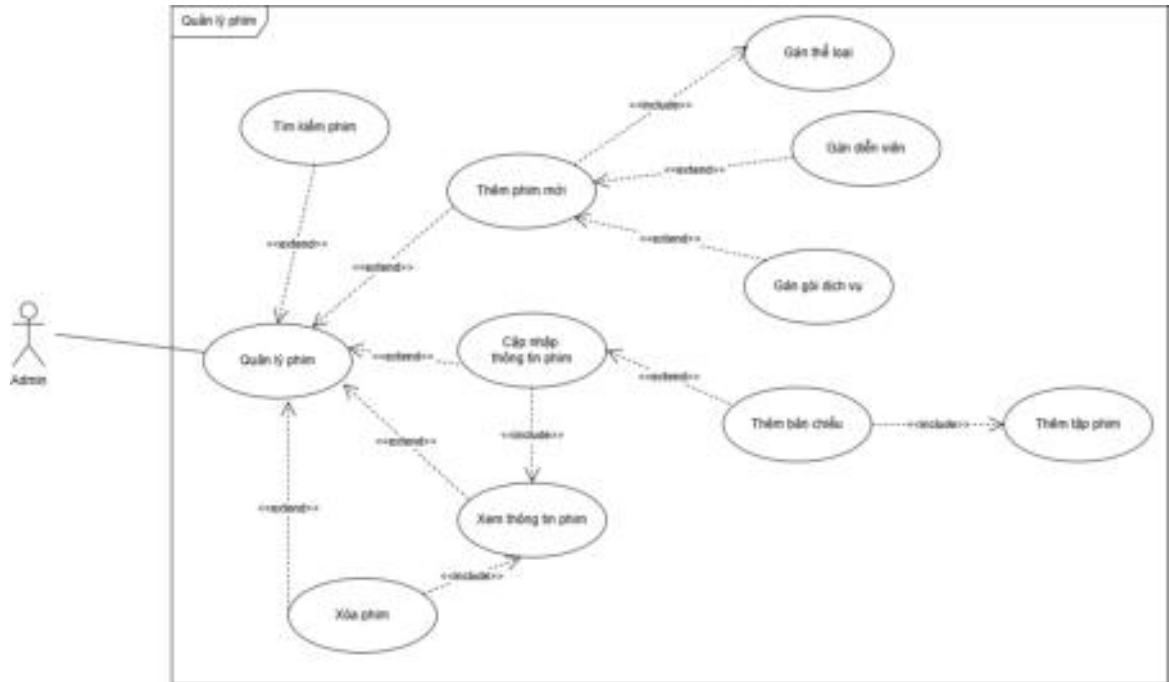
Hình 2.2 Sơ đồ use case quản lý thể loại

- Sơ đồ use case quản lý diễn viên



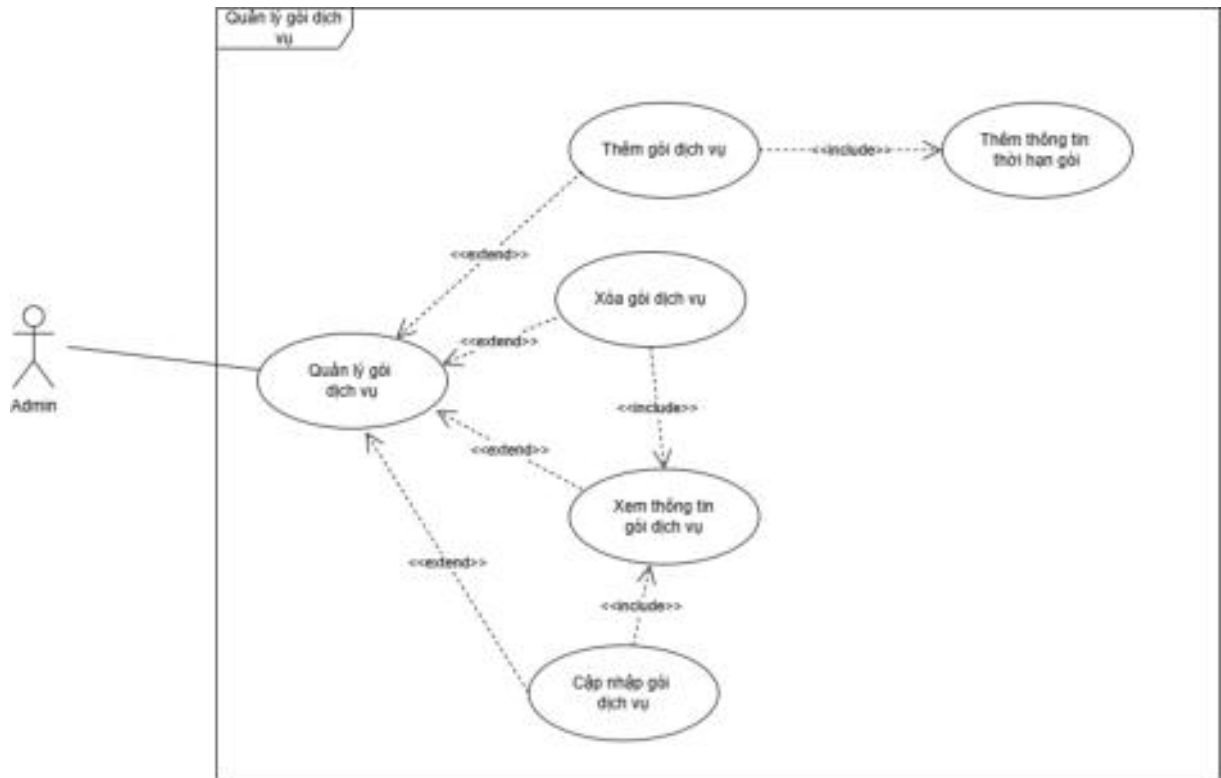
Hình 2.3 Sơ đồ use case quản lý diễn viên

- Sơ đồ use case quản lý phim



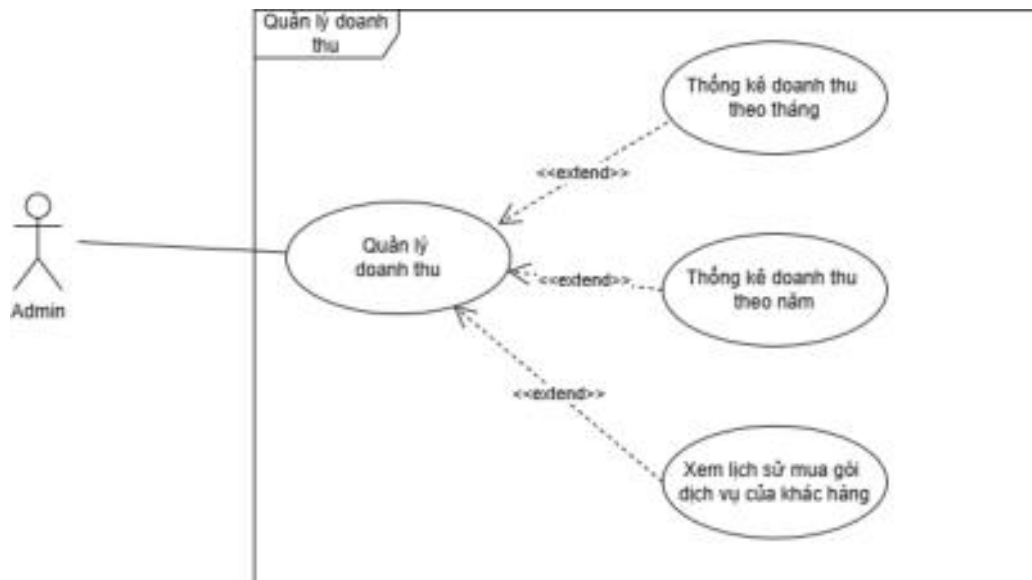
Hình 2.4 Sơ đồ use case quản lý phim

- Sơ đồ use case quản lý gói dịch vụ



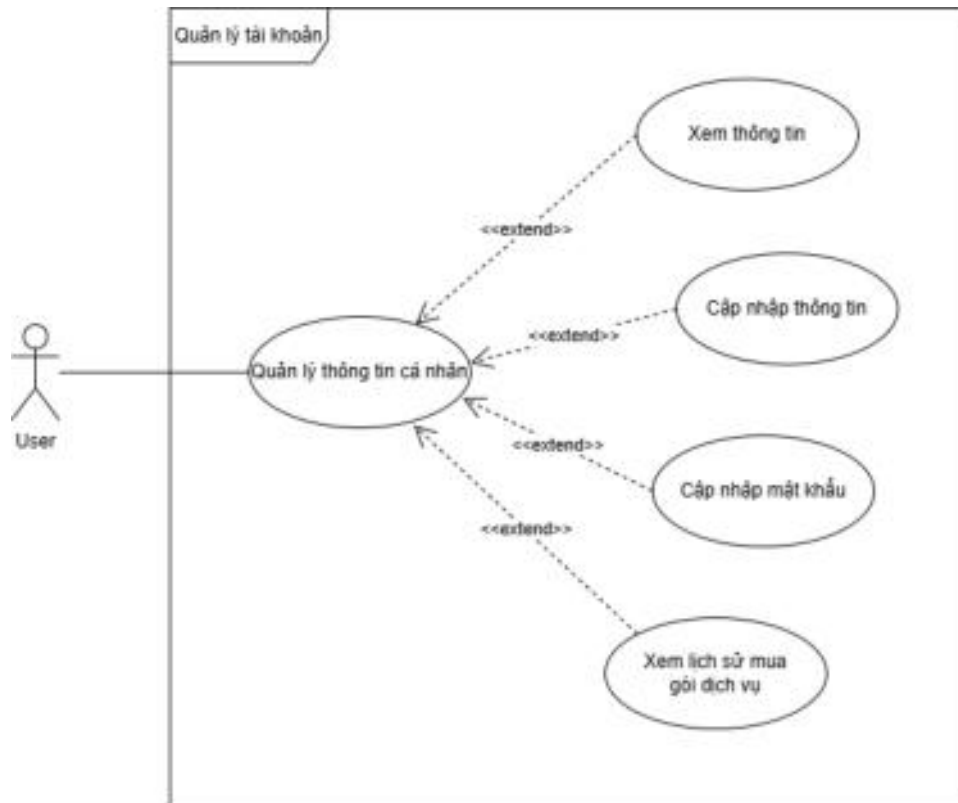
Hình 2.5 Sơ đồ use case quản lý gói dịch vụ

- Sơ đồ use case quản lý doanh thu



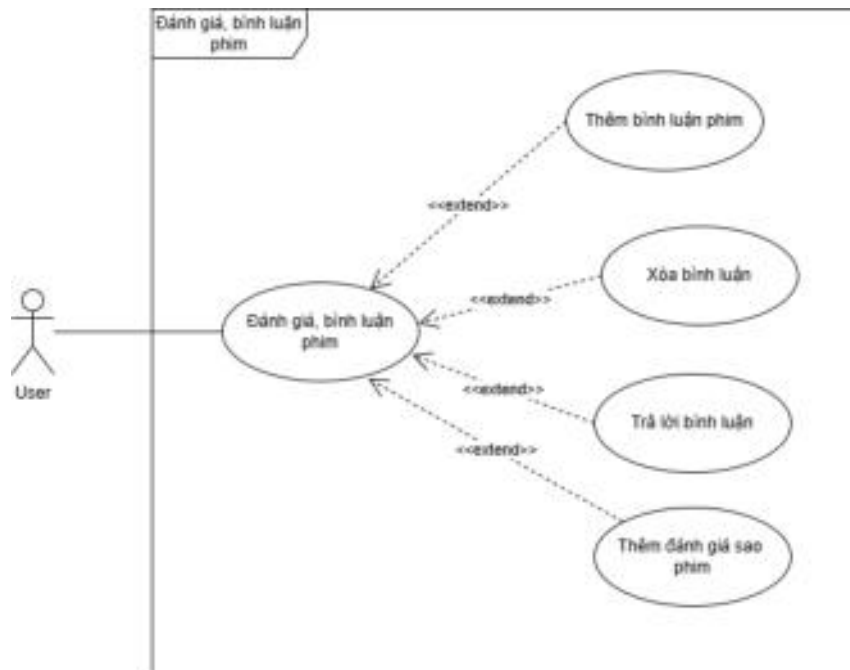
Hình 2.6 Sơ đồ use case quản lý doanh thu

- Sơ đồ use case quản lý thông tin tài khoản



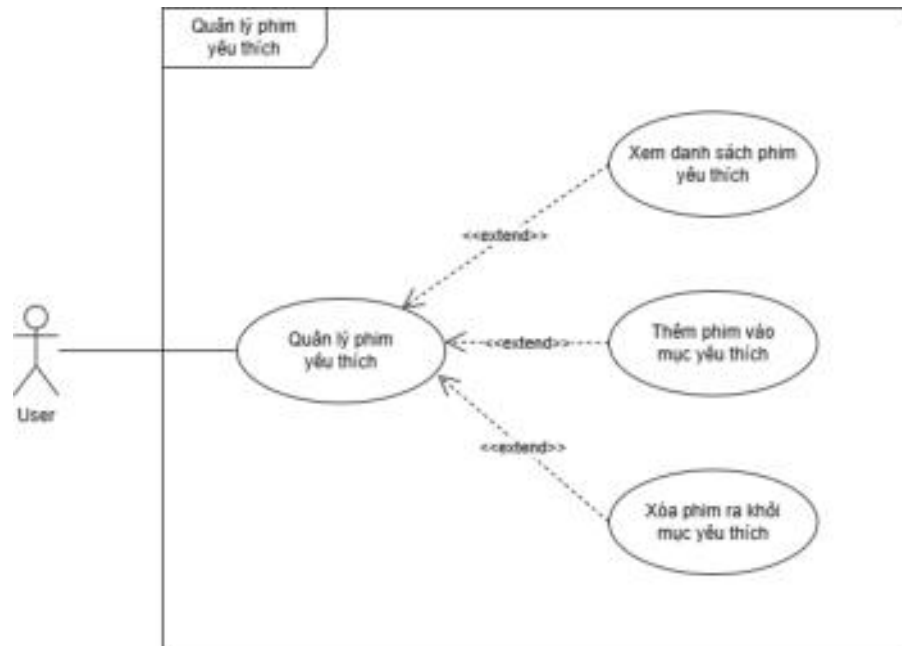
Hình 2.7 Sơ đồ use case quản lý thông tin tài khoản

- Sơ đồ use case đánh giá, bình luận phim



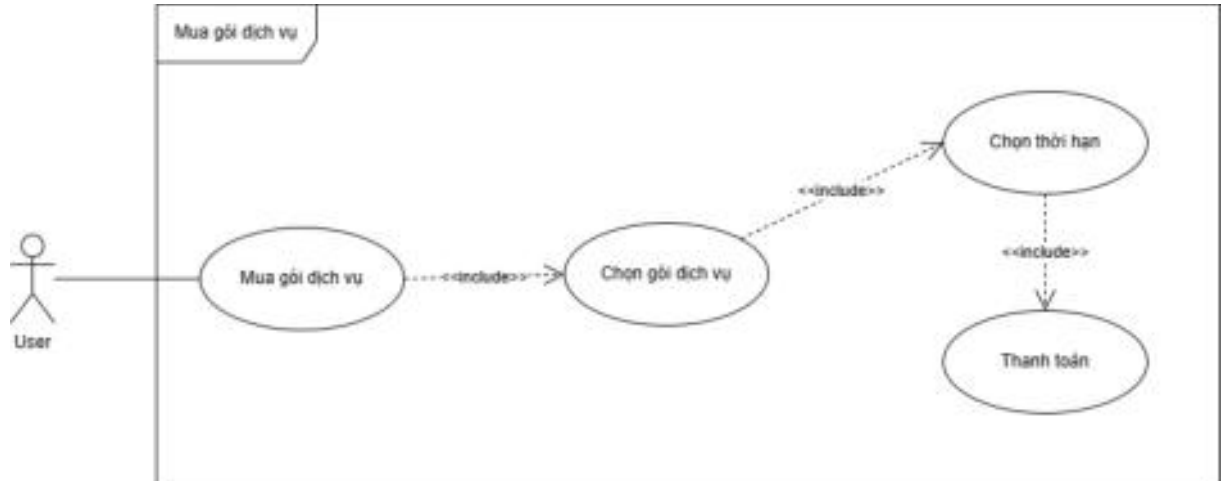
Hình 2.8 Sơ đồ use case đánh giá, bình luận phim

- Sơ đồ use case quản lý phim yêu thích



Hình 2.9 Sơ đồ use case quản lý phim yêu thích

- Sơ đồ use case mua gói dịch vụ



Hình 2.10 Sơ đồ use case mua gói dịch vụ

2.4.2 Đặc tả use case

Quản lý tài khoản

- Đặc tả use case đăng ký tài khoản

Bảng 2.1 Đặc tả use case đăng ký tài khoản

| | |
|----------------------------|--|
| Tên use case | Đăng ký tài khoản |
| Tác nhân | User |
| Mô tả | Cho phép người dùng đăng ký tài khoản vào hệ thống |
| Điều kiện kích hoạt | Người dùng truy cập vào hệ thống nhưng chưa có tài khoản |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Người dùng nhập các thông tin cần thiết cho việc đăng ký: email, tên, mật khẩu, và mật khẩu xác nhận 2. Nhấn vào button đăng ký 3. Hệ thống gửi mã OTP thông qua email mà người dùng đã đăng ký 4. Người dùng điền mã OTP để tiến hành xác nhận, hoặc yêu cầu tạo OTP mới nếu OTP đã hết hạn 5. Hệ thống xác nhận thông tin |
| Kết quả | Nếu OTP hợp lệ, thông báo đăng ký thành công |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Người dùng không điền đủ thông tin, hoặc nhập thông tin không hợp lệ 2. Tài khoản email đã tồn tại trong hệ thống 3. Mã OTP đã hết hạn hoặc không chính xác |

- Đặc tả use case đăng nhập

Bảng 2.2 Đặc tả use case đăng nhập

| | |
|----------------------------|---|
| Tên use case | Đăng nhập |
| Tác nhân | User |
| Mô tả | Cho phép user xác thực thông tin và truy cập vào hệ thống để thực hiện các chức năng phù hợp |
| Điều kiện kích hoạt | Người dùng truy cập vào hệ thống nhưng chưa đăng nhập. Nhấn vào button “Đăng nhập” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhập thông tin cho quá trình đăng nhập: email đã đăng ký, mật khẩu |

| | |
|-----------------------|--|
| | <ol style="list-style-type: none"> Nhấn vào button “Đăng nhập” Hệ thống kiểm tra dữ liệu Thông báo kết quả |
| Kết quả | Đưa đến trang chủ của hệ thống |
| Trường hợp lỗi | <ol style="list-style-type: none"> Nhập định dạng thông tin không chính xác Thông tin email hoặc mật khẩu không đúng |

- Đặc tả use case quên mật khẩu

Bảng 2.3 Đặc tả use case quên mật khẩu

| | |
|----------------------------|--|
| Tên use case | Quên mật khẩu |
| Tác nhân | User |
| Mô tả | Khi người dùng không nhớ mật khẩu đăng nhập |
| Điều kiện kích hoạt | Từ trang “Đăng nhập” chọn “Quên mật khẩu” |
| Các bước thực hiện | <ol style="list-style-type: none"> Nhập email đã đăng ký tài khoản trên hệ thống Hệ thống kiểm tra email và gửi link reset mật khẩu qua email đó Người dùng vào email tìm thư được gửi về và nhấn vào link được gửi Nhập mật khẩu mới và xác nhận mật khẩu |
| Kết quả | Thông báo kết quả và chuyển người dùng đến trang đăng nhập |
| Trường hợp lỗi | <ol style="list-style-type: none"> Người dùng nhập định dạng email không chính xác Email chưa được đăng ký trong hệ thống Đường link reset mật khẩu bị hết hạn Mật khẩu mới và xác nhận mật khẩu không trùng khớp Thông báo màn hình lỗi |

Quản lý thẻ loại

- Đặc tả use case thêm thẻ loại

Bảng 2.4 Đặc tả use case thêm thẻ loại

| | |
|---------------------|--------------------------------------|
| Tên use case | Thêm thẻ loại |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn thêm mới thẻ loại |

| | |
|----------------------------|--|
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Thẻ loại” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào button “Tạo thẻ loại” 2. Hệ thống hiển thị form để điền các thông tin 3. Quản trị viên nhập thông tin cần thiết 4. Nhấn vào nút “Tạo” để tiến hành thêm |
| Kết quả | Hệ thống hiển thị thông báo và quay lại màn hình quản lý thẻ loại |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Điền thiếu thông tin cho quá trình tạo 2. Tên thẻ loại đã tồn tại trên hệ thống 3. Hiển thị thông báo lỗi |

- Đặc tả use case cập nhập thẻ loại

Bảng 2.5 Đặc tả use case cập nhập thẻ loại

| | |
|----------------------------|--|
| Tên use case | Cập nhập thẻ loại |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn cập nhập thông tin thẻ loại |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Thẻ loại” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào nút “Sửa” của 1 thẻ loại 2. Hệ thống hiển thị thông tin của thẻ loại 3. Quản trị viên sửa đổi thông tin 4. Nhấn vào nút “Cập nhập” để tiến hành quá trình |
| Kết quả | Hệ thống hiển thị thông báo và quay lại màn hình quản lý thẻ loại |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Sửa tên thẻ loại thành 1 tên thẻ loại đã tồn tại trên hệ thống 2. Để trống thông tin của thẻ loại 3. Hiển thị thông báo lỗi |

- Đặc tả use case xóa thẻ loại

Bảng 2.6 Đặc tả use case xóa thẻ loại

| | |
|----------------------------|--|
| Tên use case | Xóa thẻ loại |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn xóa một thẻ loại nào đó |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Thẻ loại” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào nút “Xóa” của một thẻ loại 2. Hệ thống hiển thị thông báo xác nhận 3. Quản trị viên nhấn vào nút “Xóa” để xác nhận quá trình hoặc “Thoát” để hủy bỏ 4. Hệ thống tiến hành xóa thẻ loại và nếu thẻ loại đã được gán cho phim nào đó thì xóa thẻ loại ra khỏi phim đó |
| Kết quả | Hệ thống hiển thị thông báo và quay lại màn hình quản lý thẻ loại |
| Trường hợp lỗi | |

- Đặc tả tìm kiếm thẻ loại

Bảng 2.7 Đặc tả tìm kiếm thẻ loại

| | |
|----------------------------|--|
| Tên use case | Tìm kiếm thẻ loại |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn tìm kiếm thẻ loại |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Thẻ loại” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Quản trị viên nhập tên thẻ loại ở input “Tìm kiếm thẻ loại” 2. Hệ thống hiển thị các thẻ loại tìm kiếm |
| Kết quả | Hiển thị danh sách thẻ loại với kết quả đã nhập |
| Trường hợp lỗi | Không có |

Quản lý diễn viên

- Đặc tả use case thêm diễn viên

Bảng 2.8 Đặc tả use case thêm diễn viên

| | |
|----------------------------|---|
| Tên use case | Thêm diễn viên |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn thêm diễn viên mới vào hệ thống |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Diễn viên” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Quản trị viên nhấn vào nút “Thêm diễn viên” 2. Hệ thống hiển thị form thêm diễn viên 3. Quản trị nhập các thông tin về diễn viên <p>Nhấn vào nút “Thêm” để tiến hành quá trình</p> |
| Kết quả | Hệ thống hiển thị thông báo và quay lại màn hình quản lý diễn viên |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Nhập thiếu hoặc không đúng định dạng của các trường thông tin 2. Hiển thị thông báo lỗi |

- Đặc tả use case cập nhập diễn viên

Bảng 2.9 Đặc tả use case cập nhập diễn viên

| | |
|----------------------------|---|
| Tên use case | Cập nhập diễn viên |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn cập nhập thông tin diễn viên |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Diễn viên” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào nút “Sửa” của diễn viên muốn cập nhập 2. Quản trị viên điền các thông tin mới cho diễn viên 3. Nhấn vào nút “Cập nhập” để tiến hành quá trình 4. Hệ thống tiến hành quá trình cập nhập |
| Kết quả | Hệ thống hiển thị thông báo và quay lại trang quản lý diễn viên |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Quản trị viên nhập thiếu hoặc để trống các thông tin 2. Hiển thị thông báo lỗi |

Quản lý phim

- Đặc tả use case thêm phim

Bảng 2.10 Đặc tả use case thêm phim

| | |
|----------------------------|---|
| Tên use case | Thêm phim |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn thêm phim mới vào hệ thống |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Danh sách phim” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào button “Thêm phim” 2. Hệ thống hiển thị 2 loại phim là “Phim lẻ” và “Phim bộ” 3. Quản trị viên chọn loại phim để thêm 4. Hệ thống hiển thị form thêm phim đúng với loại phim 5. Điền các thông tin cần thiết cho quá trình thêm phim 6. Nhấn vào nút “Thêm” để tiến hành quá trình |
| Kết quả | Thông báo kết quả và đưa đến màn hình quản lý phim |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Nhập thiếu thông tin yêu cầu 2. Định dạng của các thông tin không chính xác 3. Thông báo màn hình lỗi |

- Đặc tả use case cập nhập phim

Bảng 2.11 Đặc tả use case cập nhập phim

| | |
|----------------------------|--|
| Tên use case | Cập nhập phim |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn cập nhập thông tin hoặc thêm các bản chiếu, tập phim cho phim |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Danh sách phim” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào button “Sửa” vào phim muốn cập nhập 2. Hệ thống hiển thị thông tin chi tiết của phim Quản trị viên tiến hành cập nhập thông tin phim, phiên bản chiếu, tập phim |
| Kết quả | Thông báo kết quả và đưa đến màn hình quản lý phim |

| | |
|-----------------------|--|
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Nhập thiếu thông tin yêu cầu 2. Định dạng của các thông tin không chính xác 3. Thông báo màn hình lỗi |
|-----------------------|--|

- Đặc tả use case xóa phim

Bảng 2.12 Đặc tả use case xóa phim

| | |
|----------------------------|--|
| Tên use case | Xóa phim |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn xóa phim |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Danh sách phim” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào “Xóa” của phim muốn đi 2. Hệ thống hiển thị thông báo xác nhận 3. Nhấn vào “Xóa” để thực hiện hoặc “Thoát” để hủy bỏ |
| Kết quả | Hiển thị thông báo lên màn hình |
| Trường hợp lỗi | Không có |

- Đặc tả use case tìm kiếm phim

Bảng 2.13 Đặc tả use case tìm kiếm phim

| | |
|----------------------------|--|
| Tên use case | Tìm kiếm phim |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn tìm kiếm phim theo tên hoặc các filters khác |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Quản lý phim” và chọn mục “Danh sách phim” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nếu tìm phim theo tên thì nhập vào tên phim vào input “Tìm kiếm phim” 2. Nếu tìm kiếm phim theo các đặc điểm của phim, nhấn vào “Bộ lọc” và chọn các mô tả về phim 3. Hệ thống tìm kiếm dựa vào kết quả lọc |
| Kết quả | Hiển thị các phim phù hợp với bộ lọc |
| Trường hợp lỗi | Không có |

Quản lý gói dịch vụ

- Đặc tả use case thêm gói dịch vụ

Bảng 2.14 Đặc tả use case thêm gói dịch vụ

| | |
|----------------------------|--|
| Tên use case | Thêm gói dịch vụ |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên thêm gói dịch vụ để có thể thêm các phim đặc biệt vào dịch vụ |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Gói đăng ký” và nhấn vào button “Thêm gói mới” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Hệ thống hiển thị form “Thêm gói đăng ký” 2. Quản trị viên nhập các thông tin cần thiết, thêm thông tin thời hạn của gói 3. Nhấn vào button “Thêm” để tiến hành quá trình |
| Kết quả | Hiển thị thông báo và quay lại tranh danh sách gói dịch vụ |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Nhập thiếu hoặc không đúng định dạng của các trường trong form thêm 2. Hiển thị thông báo lỗi |

- Đặc tả use case cập nhập gói dịch vụ

Bảng 2.15 Đặc tả use case cập nhập gói dịch vụ

| | |
|----------------------------|---|
| Tên use case | Cập nhập gói dịch vụ |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn cập nhập lại thông tin của gói dịch vụ |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Gói đăng ký” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Chọn “Sửa” của gói dịch vụ muốn cập thông tin 2. Hệ thống hiển thị chi tiết thông tin gói dịch vụ 3. Quản trị viên sửa các thông tin để tiến hành cập nhập 4. Nhấn vào “Cập nhập” để tiến hành cập nhập |
| Kết quả | Hiển thị thông báo về trang quản lý gói dịch vụ |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Nhập thiếu thông tin hoặc sai định dạng 2. Thống báo lỗi |

- Đặc tả use case xóa gói dịch vụ

Bảng 2.16 Đặc tả use case xóa gói dịch vụ

| | |
|----------------------------|---|
| Tên use case | Xóa gói dịch vụ |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn xóa gói dịch vụ nào đó |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Gói đăng ký” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Nhấn vào “Xóa” của gói dịch vụ muốn xóa 2. Hệ thống hiển thị thông báo xác nhận 3. Chọn “Xóa” để tiến hành xóa, hoặc “Thoát” để hủy bỏ quá trình |
| Kết quả | Hiện thị thông báo lên màn hình và quay lại trang quản lý gói dịch vụ |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Xóa gói dịch vụ mà gói đó hiện đang là cha của một gói khác 2. Hiện thị thông báo lỗi |

Quản lý doanh thu

- Đặc tả use case thống kê doanh thu

Bảng 2.17 Đặc tả use case thống kê doanh thu

| | |
|----------------------------|---|
| Tên use case | Thống kê doanh thu |
| Tác nhân | Quản trị viên |
| Mô tả | Quản trị viên muốn thống kê doanh thu |
| Điều kiện kích hoạt | Quản trị viên đã đăng nhập vào thống, trên sidebar chọn “Trang chủ” |
| Các bước thực hiện | <ol style="list-style-type: none"> 4. Chọn thống kê doanh thu theo năm hoặc theo tháng 5. Nếu chọn tháng thì nhập vào năm để thống kê doanh thu theo tháng 6. Nếu chọn năm thì nhập năm bắt đầu và năm kết thúc để thống kê doanh thu theo năm |
| Kết quả | Hiện thị biểu đồ hình cột thống kê doanh thu theo thời gian đã chọn |
| Trường hợp lỗi | |

Mua gói dịch vụ

- Đặc tả use case mua gói dịch vụ

Bảng 2.18 Đặc tả use case mua gói dịch vụ

| | |
|----------------------------|---|
| Tên use case | Mua gói dịch vụ |
| Tác nhân | Người dùng |
| Mô tả | Người dùng muốn mua gói dịch vụ để coi các bộ phim của gói dịch đó |
| Điều kiện kích hoạt | Người dùng đăng nhập vào hệ thống, trên navbar chọn “Đăng ký gói” |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Chọn gói dịch vụ muốn mua 2. Hệ thống sẽ hiển thị các thời hạn khác nhau của gói dịch vụ 3. Người dùng chọn gói thời hạn phù hợp và nhấn nút “Thanh toán” 4. Hệ thống sẽ chuyển hướng đến cổng thanh toán (VNPay) 5. Người dùng chọn phương thức thanh toán và thực hiện thanh toán 6. Hệ thống chuyển người dùng về trang xác nhận thông báo và hiển thị kết quả |
| Kết quả | Hiển thị kết quả giao dịch |
| Trường hợp lỗi | Thời gian thanh toán cho phép bị quá hạn |

Quản lý danh sách phim yêu thích

- Đặc tả use case xem danh sách phim yêu thích

Bảng 2.19 Đặc tả use case xem danh sách phim yêu thích

| | |
|----------------------------|---|
| Tên use case | Xem danh sách yêu thích |
| Tác nhân | Người dùng |
| Mô tả | Cho phép người dùng xem danh sách các phim đã thêm vào yêu thích để dễ dàng truy cập lại |
| Điều kiện kích hoạt | Người dùng đăng nhập vào hệ thống |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Người dùng chọn biểu tượng avatar trên thanh điều hướng (navbar). 2. Chọn mục “Yêu thích” trong menu thả xuống. |

| | |
|-----------------------|--|
| | 3. Hệ thống hiển thị danh sách phim yêu thích của người dùng. |
| Kết quả | Hệ thống hiển thị danh sách phim yêu thích tương ứng với tài khoản người dùng. |
| Trường hợp lỗi | |

- Đặc tả use case thêm phim vào danh sách yêu thích

Bảng 2.20 Đặc tả use case thêm phim vào danh sách yêu thích

| | |
|----------------------------|--|
| Tên use case | Thêm phim vào danh sách yêu thích |
| Tác nhân | Người dùng |
| Mô tả | Người dùng muốn thêm phim vào danh sách yêu thích để dễ dàng truy cập lại sau này. |
| Điều kiện kích hoạt | Người dùng đã đăng nhập vào hệ thống và đang ở giao diện chứa phim (trang danh sách, chi tiết phim hoặc trang xem phim). |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Người dùng di chuột (hover) vào phim trong danh sách, hệ thống hiển thị nút "Yêu thích" hoặc biểu tượng tương ứng. 2. Người dùng nhấn vào nút "Yêu thích" ở một trong các vị trí: trang danh sách phim, trang chi tiết phim, hoặc trang xem phim. 3. Hệ thống lưu phim vào danh sách yêu thích của người dùng. 4. Hệ thống hiển thị thông báo thêm thành công. |
| Kết quả | Phim được thêm vào danh sách yêu thích và hiển thị thông báo thành công. |
| Trường hợp lỗi | Người dùng chưa đăng nhập => Hiển thị thông báo lỗi |

- Đặc tả xóa phim khỏi danh sách yêu thích

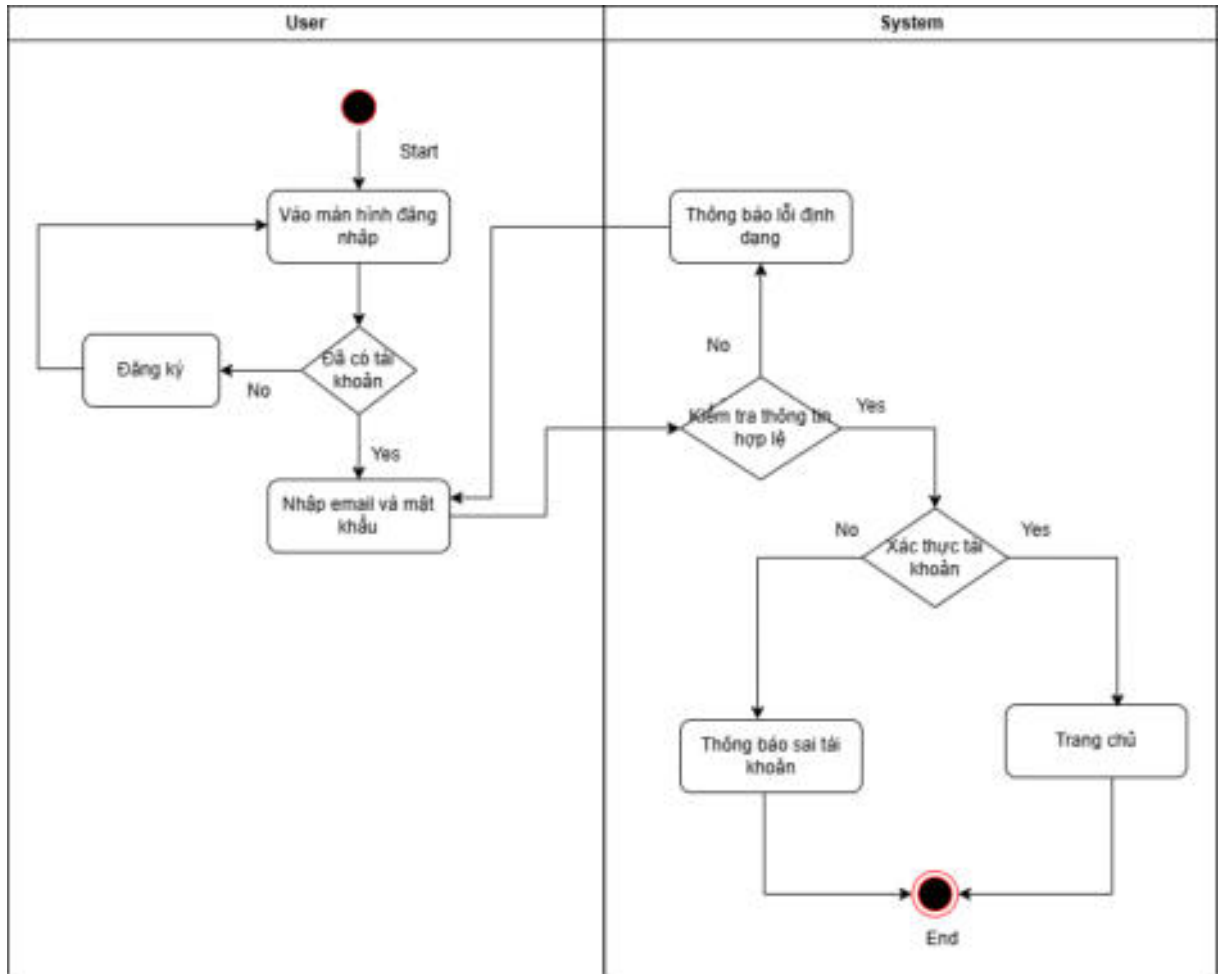
Bảng 2.21 Đặc tả xóa phim khỏi danh sách yêu thích

| | |
|---------------------|-----------------------------------|
| Tên use case | Xóa phim khỏi danh sách yêu thích |
| Tác nhân | Người dùng |

| | |
|----------------------------|---|
| Mô tả | Cho phép người dùng loại bỏ một phim khỏi danh sách yêu thích cá nhân. |
| Điều kiện kích hoạt | Người dùng đã đăng nhập và có ít nhất một phim trong danh sách yêu thích |
| Các bước thực hiện | <ol style="list-style-type: none">1. Người dùng chọn biểu tượng avatar trên thanh điều hướng (navbar).2. Chọn mục “Yêu thích” trong menu thả xuống.3. Hệ thống hiển thị danh sách các phim yêu thích.4. Người dùng nhấn vào biểu tượng “X” tại phim muốn xóa.5. Hệ thống cập nhật danh sách yêu thích và hiển thị thông báo thành công. |
| Kết quả | Hệ thống hiển thị danh sách đã cập nhật. |
| Trường hợp lỗi | |

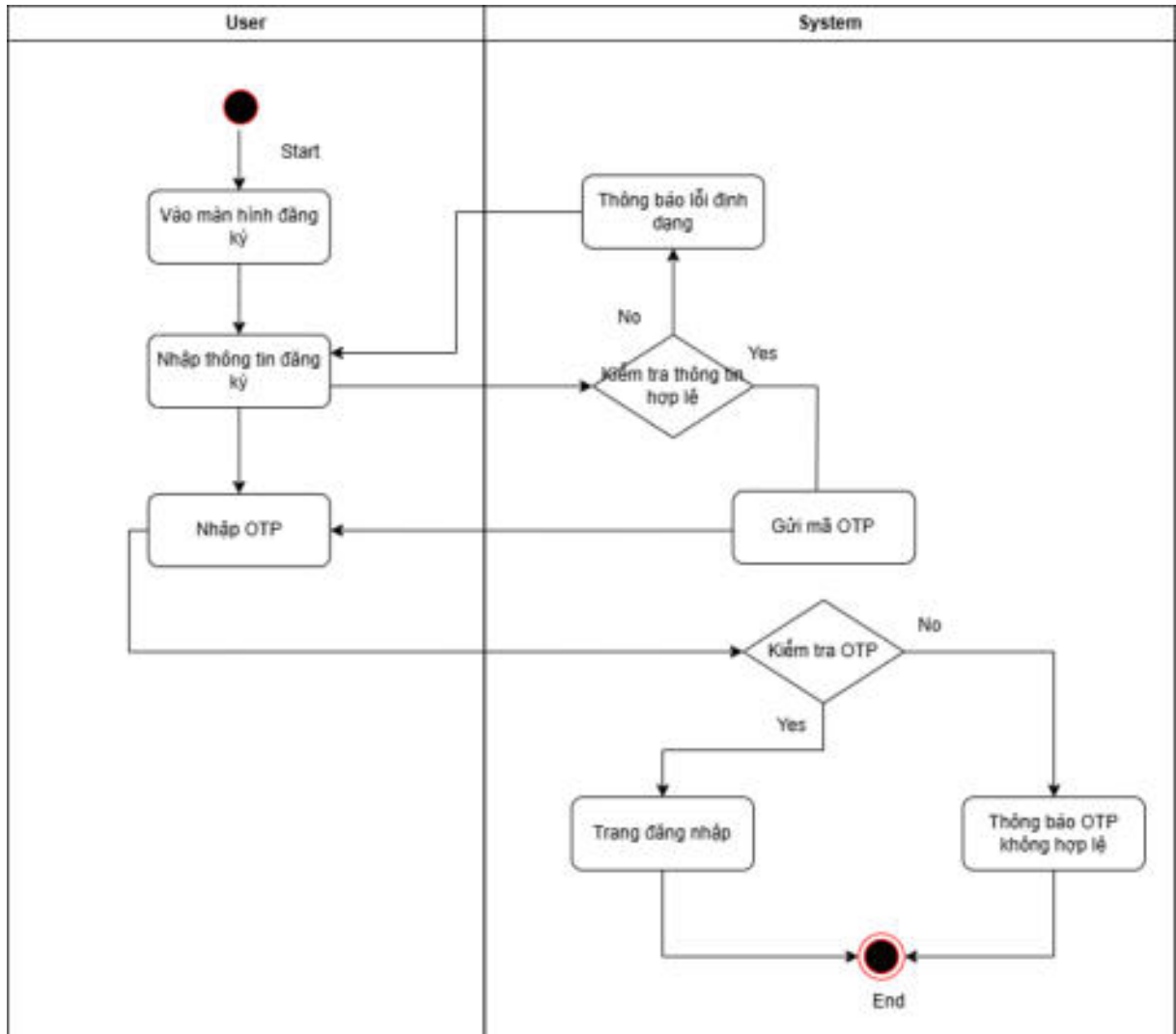
2.4.3 Sơ đồ nguyên lý hoạt động

- Sơ đồ hoạt động đăng nhập



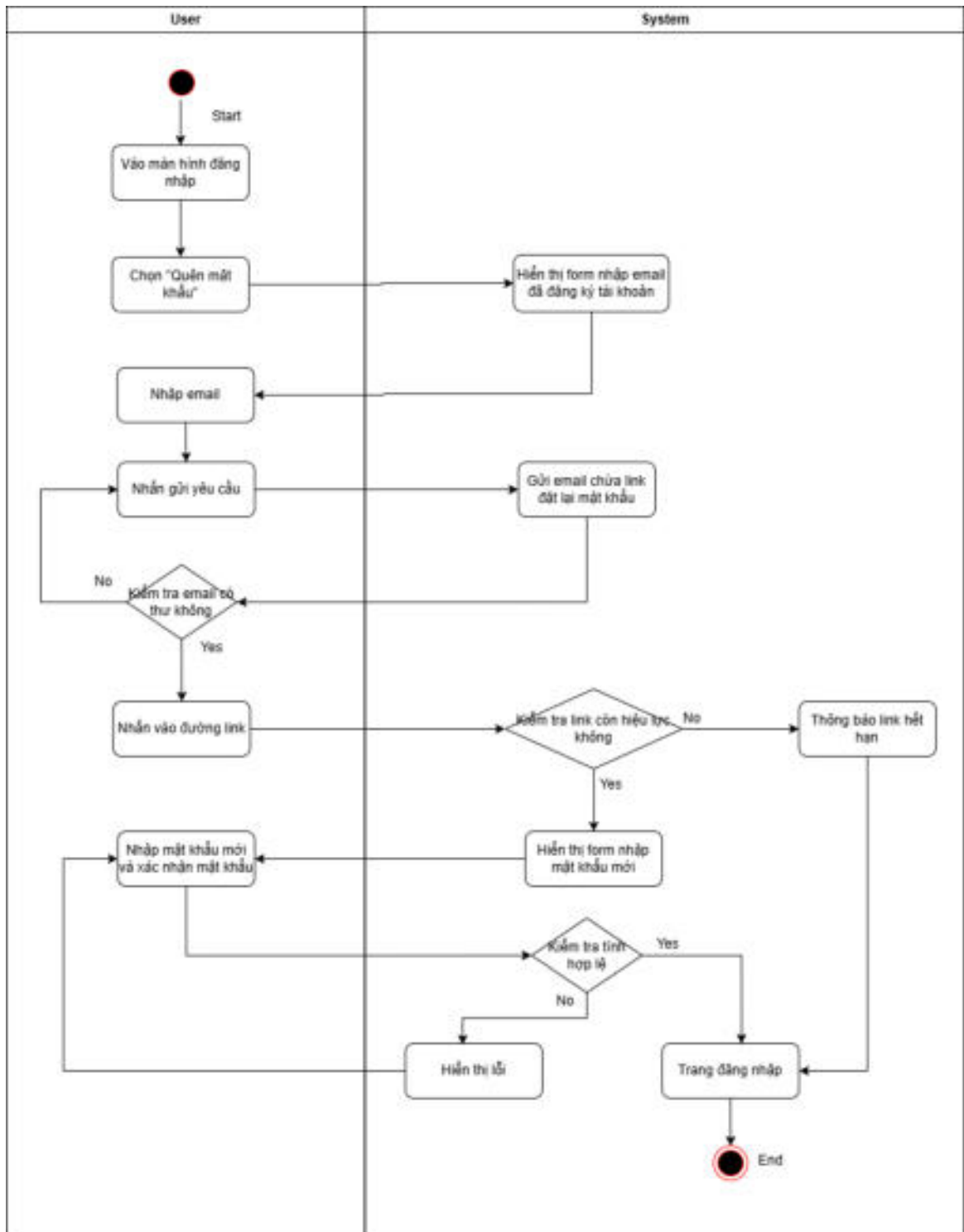
Hình 2.11 Sơ đồ hoạt động đăng nhập

- Sơ đồ hoạt động đăng ký



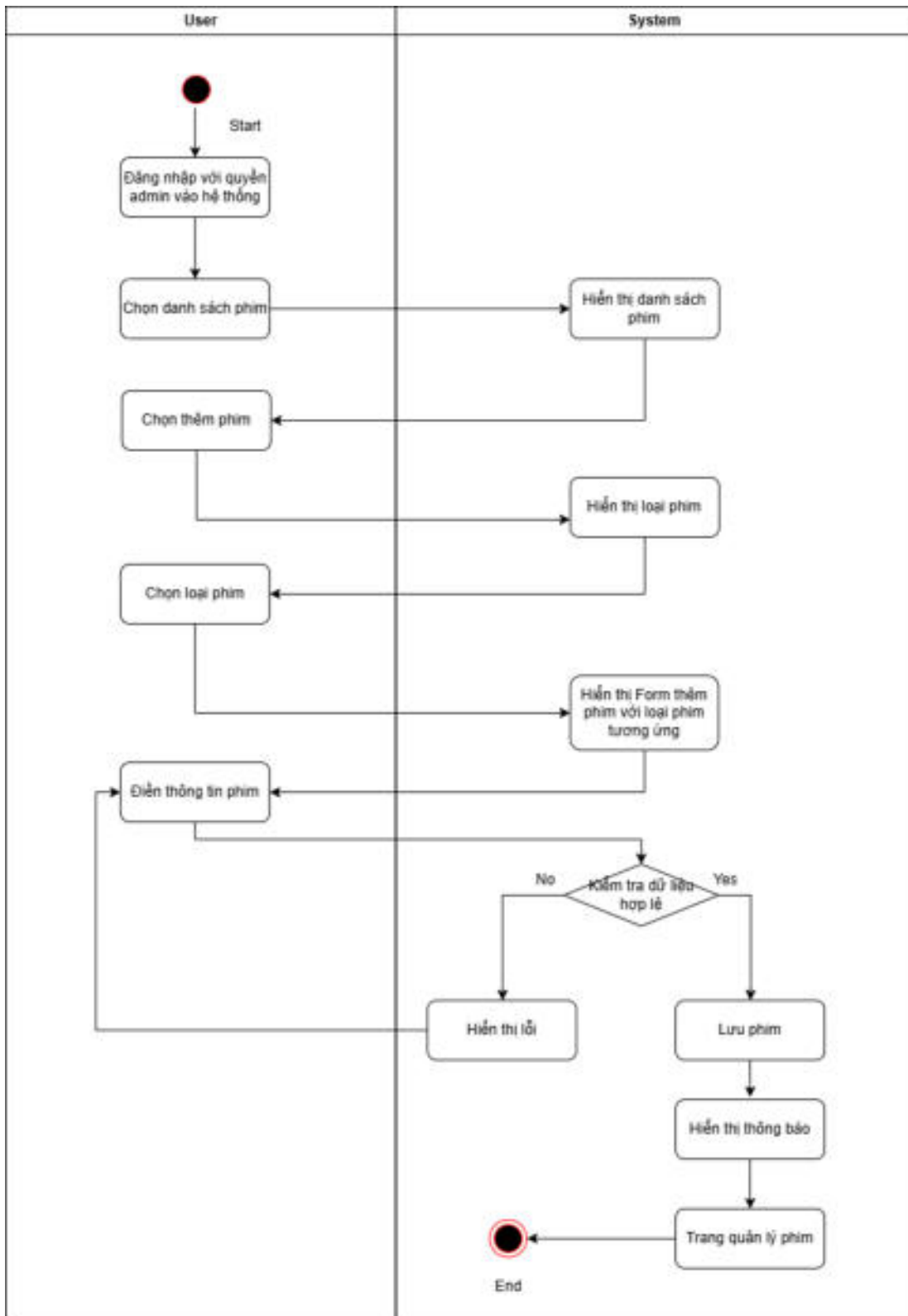
Hình 2.12 Sơ đồ hoạt động đăng ký

- Sơ đồ hoạt động quên mật khẩu



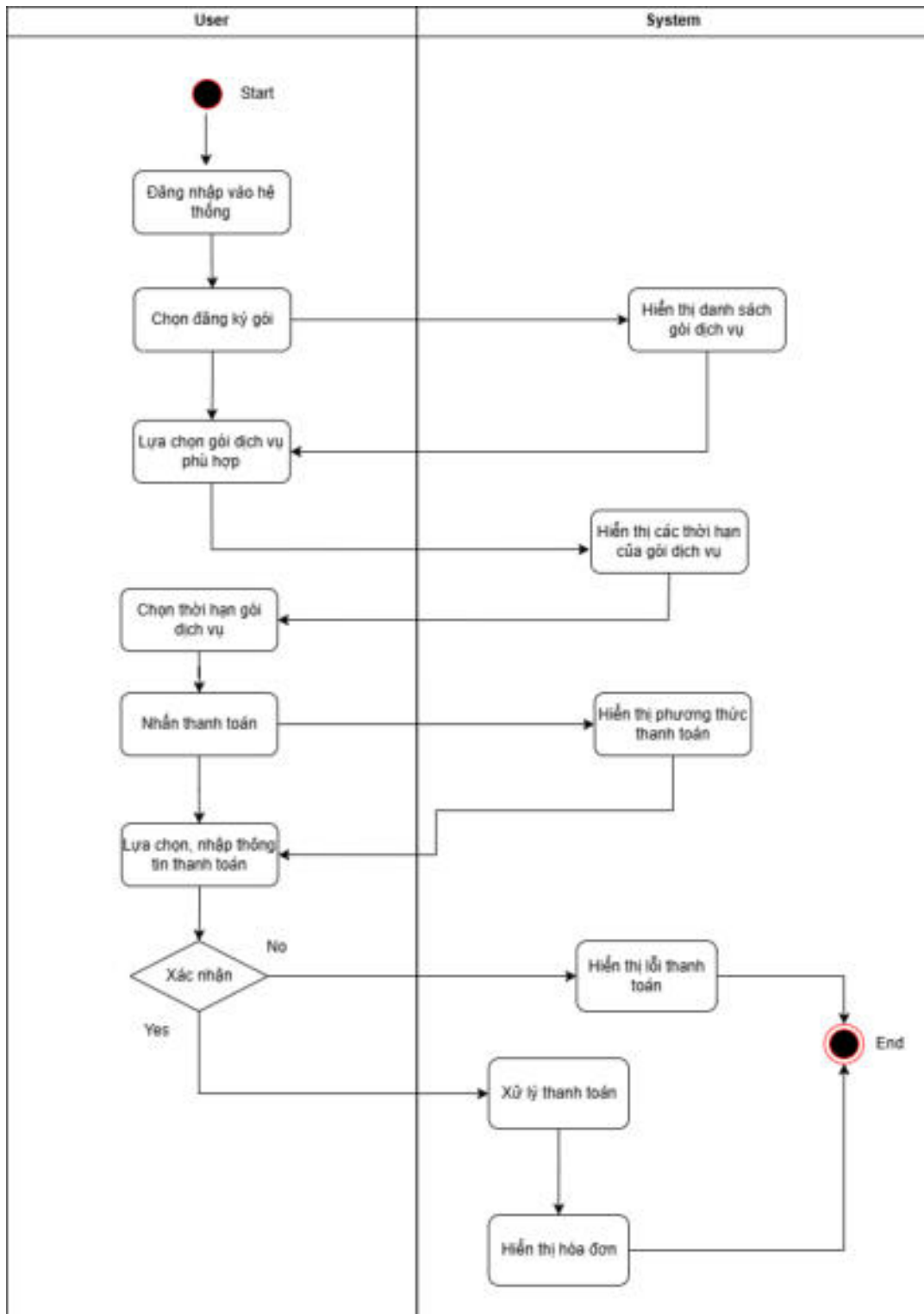
Hình 2.13 Sơ đồ hoạt động quên mật khẩu

- Sơ đồ hoạt động thêm phim



Hình 2.14 Sơ đồ hoạt động thêm phim

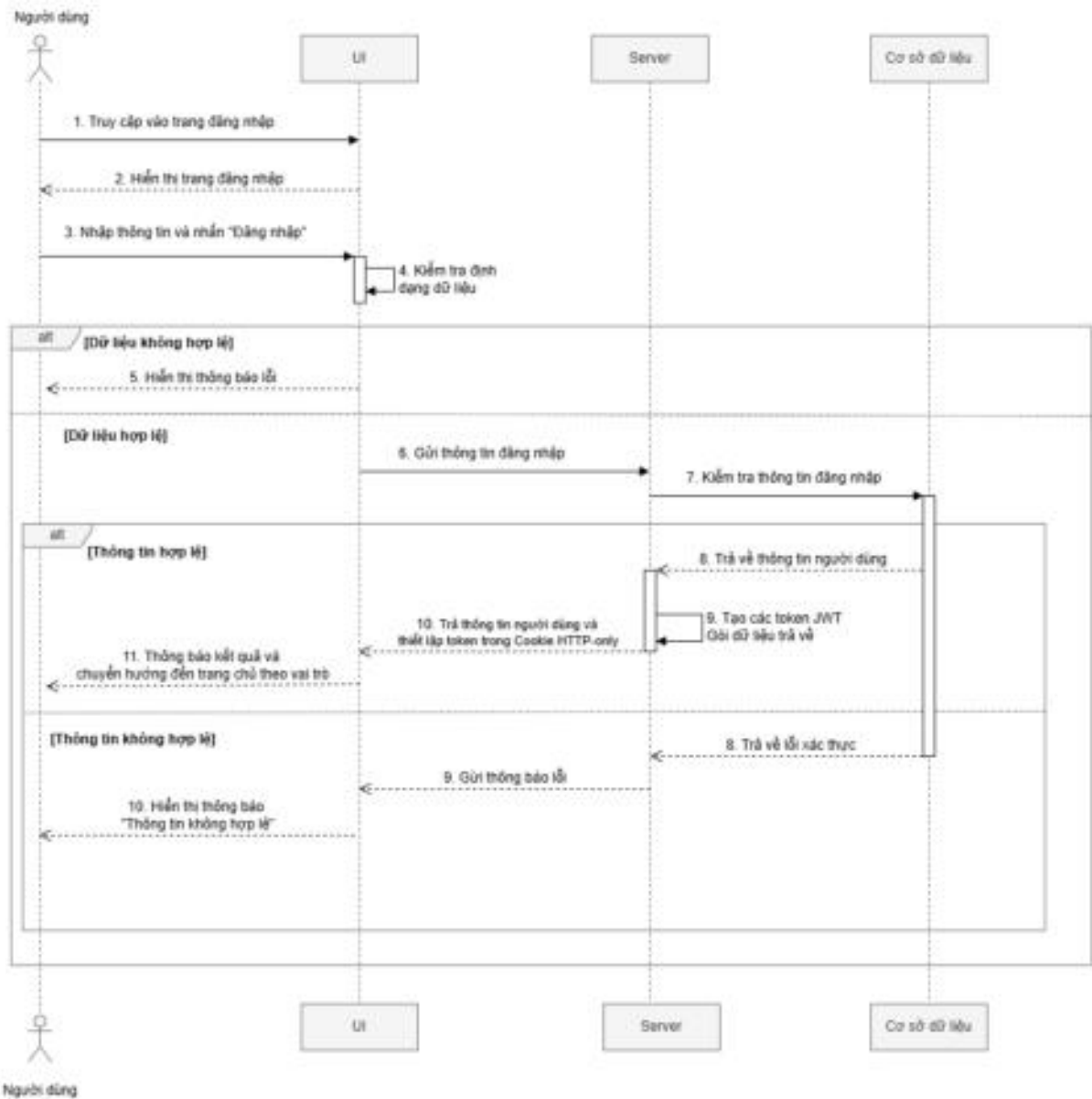
- Sơ đồ hoạt động mua gói dịch vụ



Hình 2.15 Sơ đồ hoạt động mua gói dịch vụ

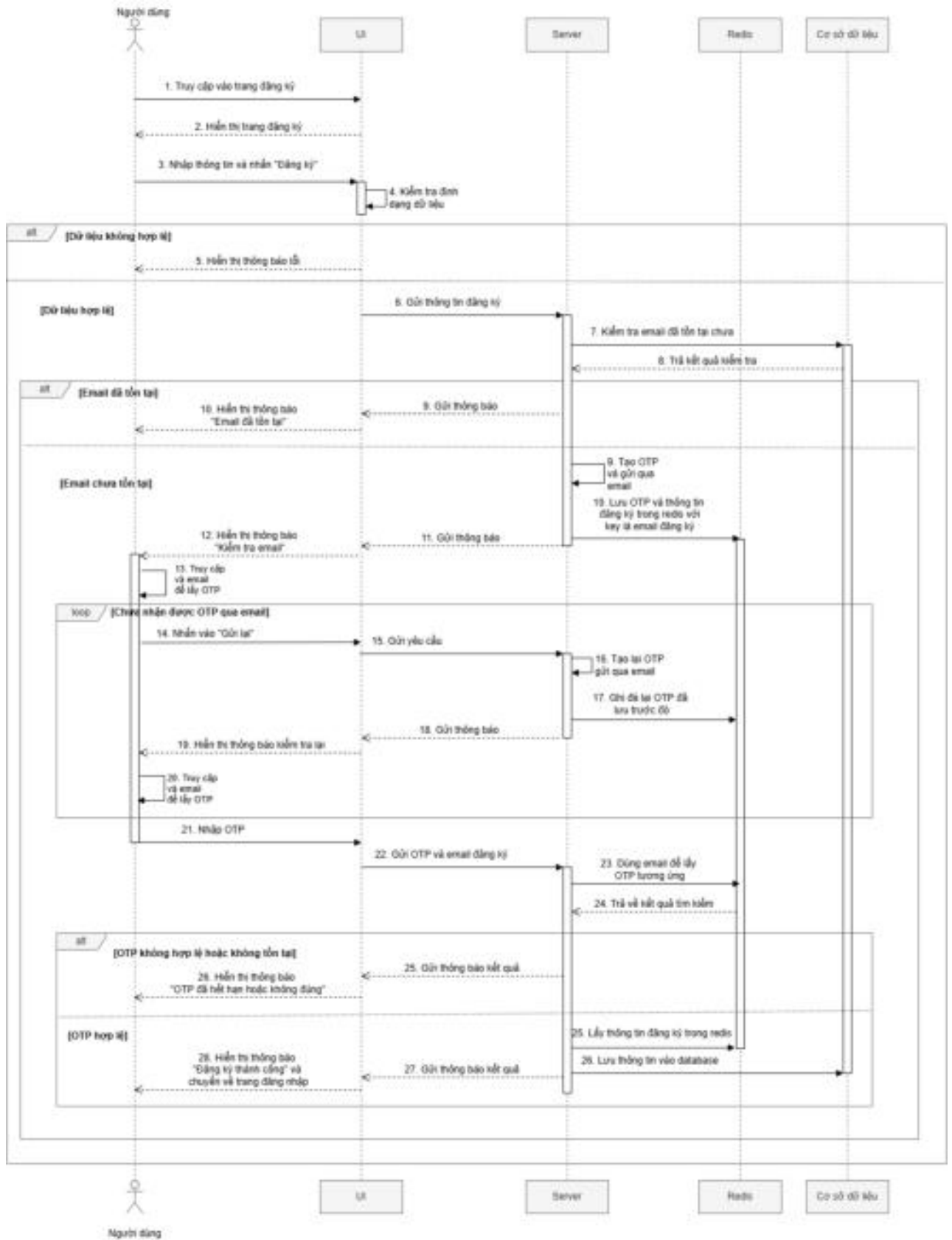
2.4.4 Sơ đồ tuần tự

- Sơ đồ tuần tự đăng nhập tài khoản



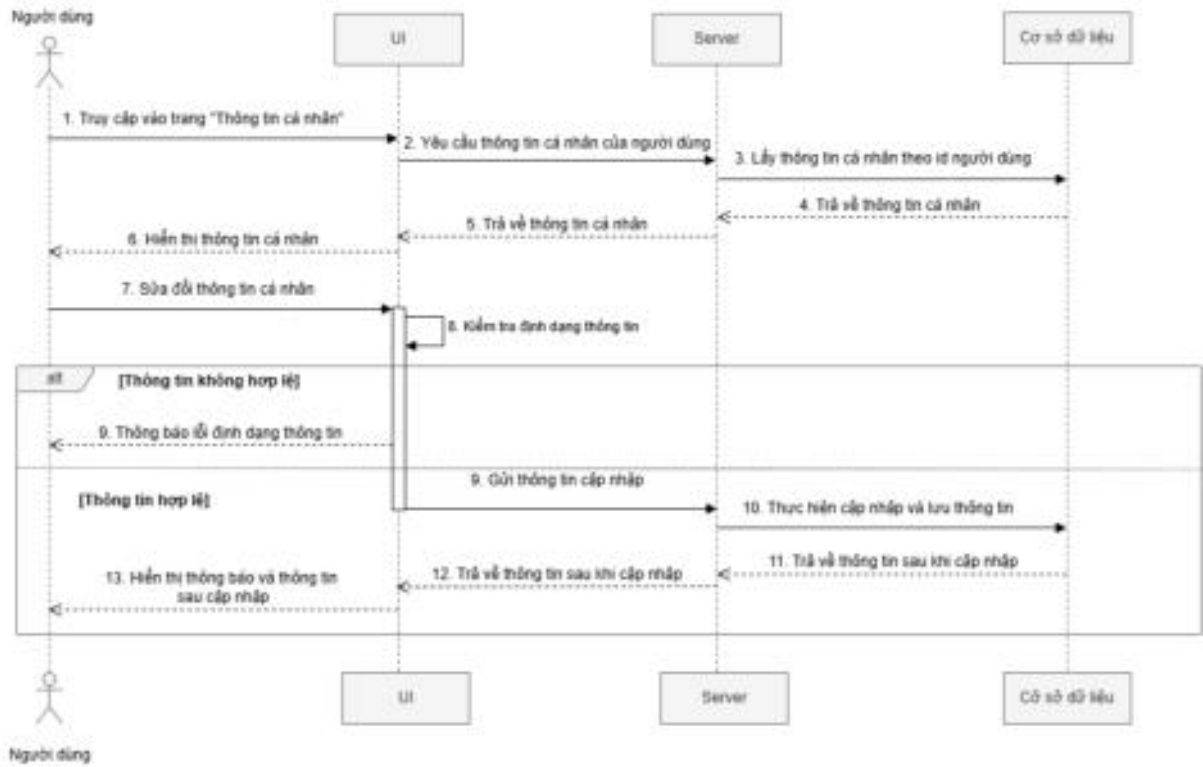
Hình 2.16 Sơ đồ tuần tự đăng nhập tài khoản

- Sơ đồ tuần tự đăng ký tài khoản



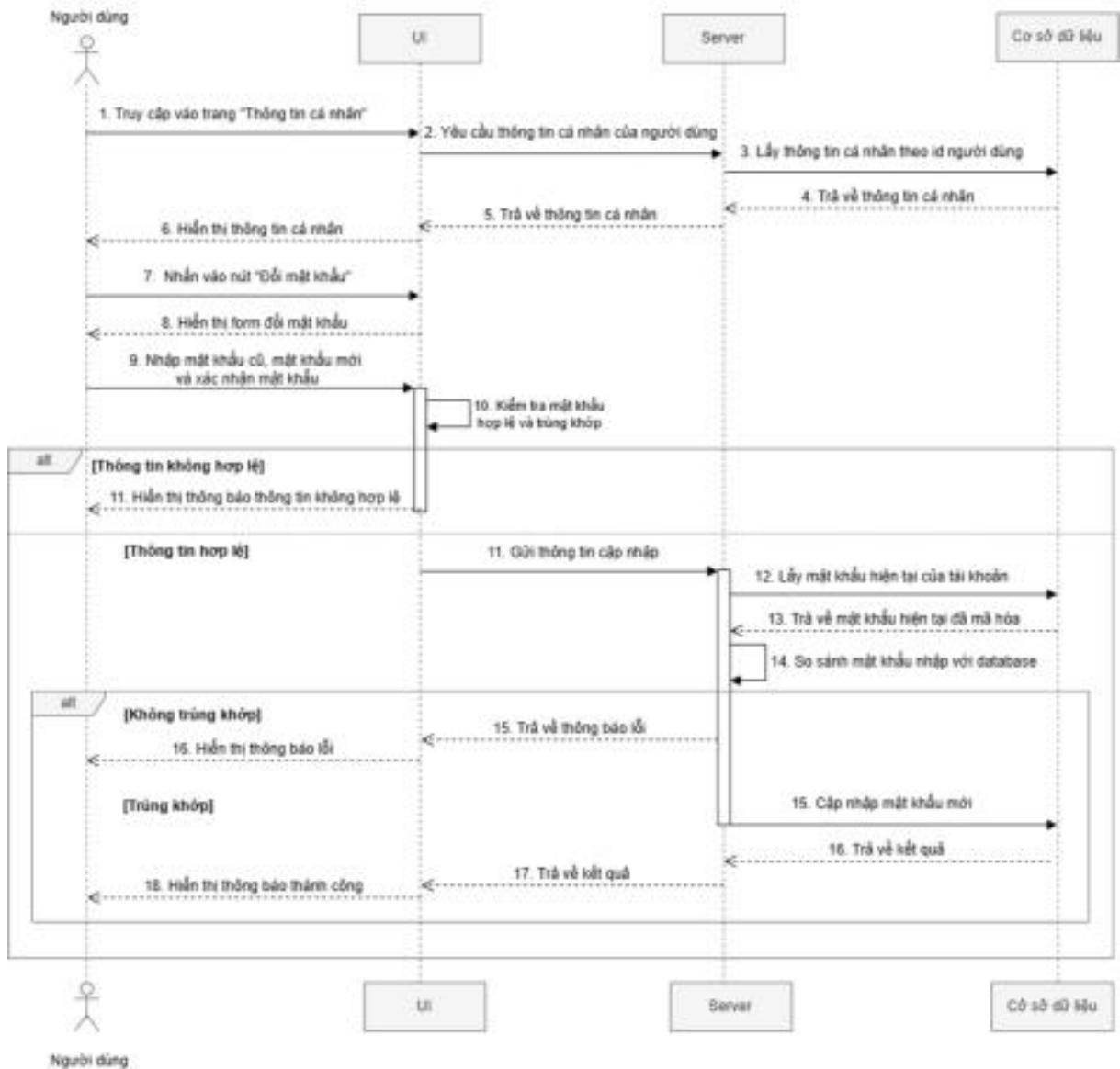
Hình 2.17 Sơ đồ tuần tự đăng ký tài khoản

- Sơ đồ tuần tự cập nhật thông tin cá nhân



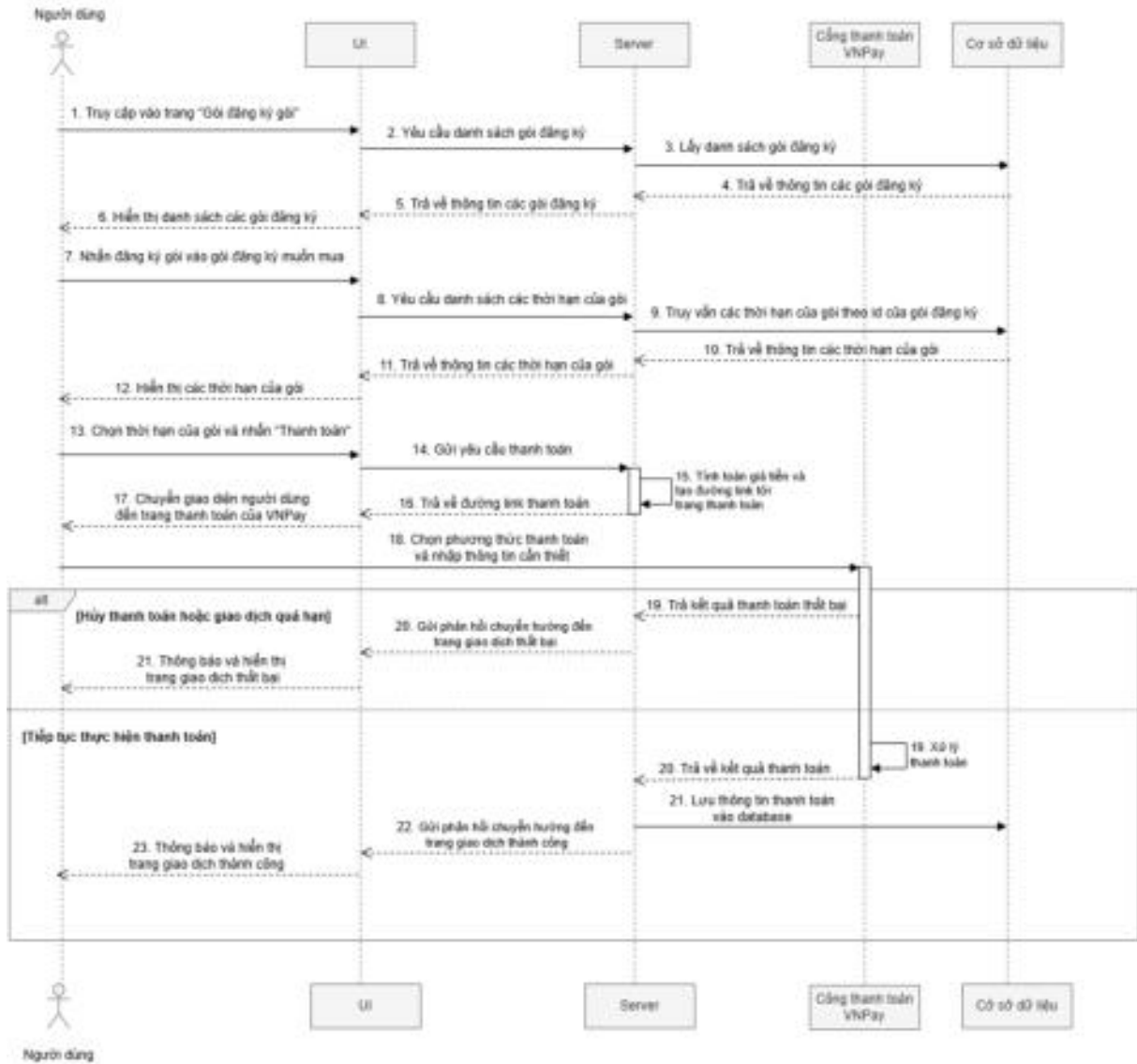
Hình 2.18 Sơ đồ tuần tự cập nhật thông tin cá nhân

- Sơ đồ tuần tự thay đổi mật khẩu



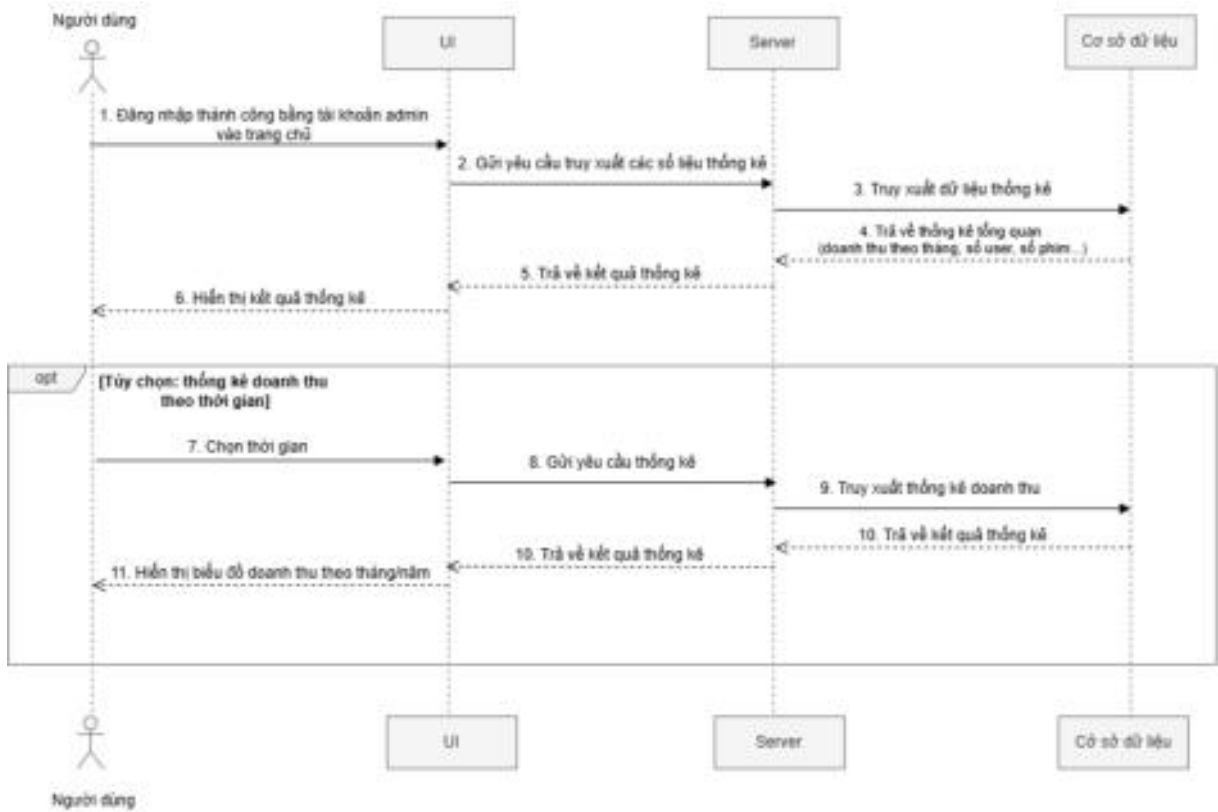
Hình 2.19 Sơ đồ tuần tự thay đổi mật khẩu

- Sơ đồ tuần tự chức năng thanh toán



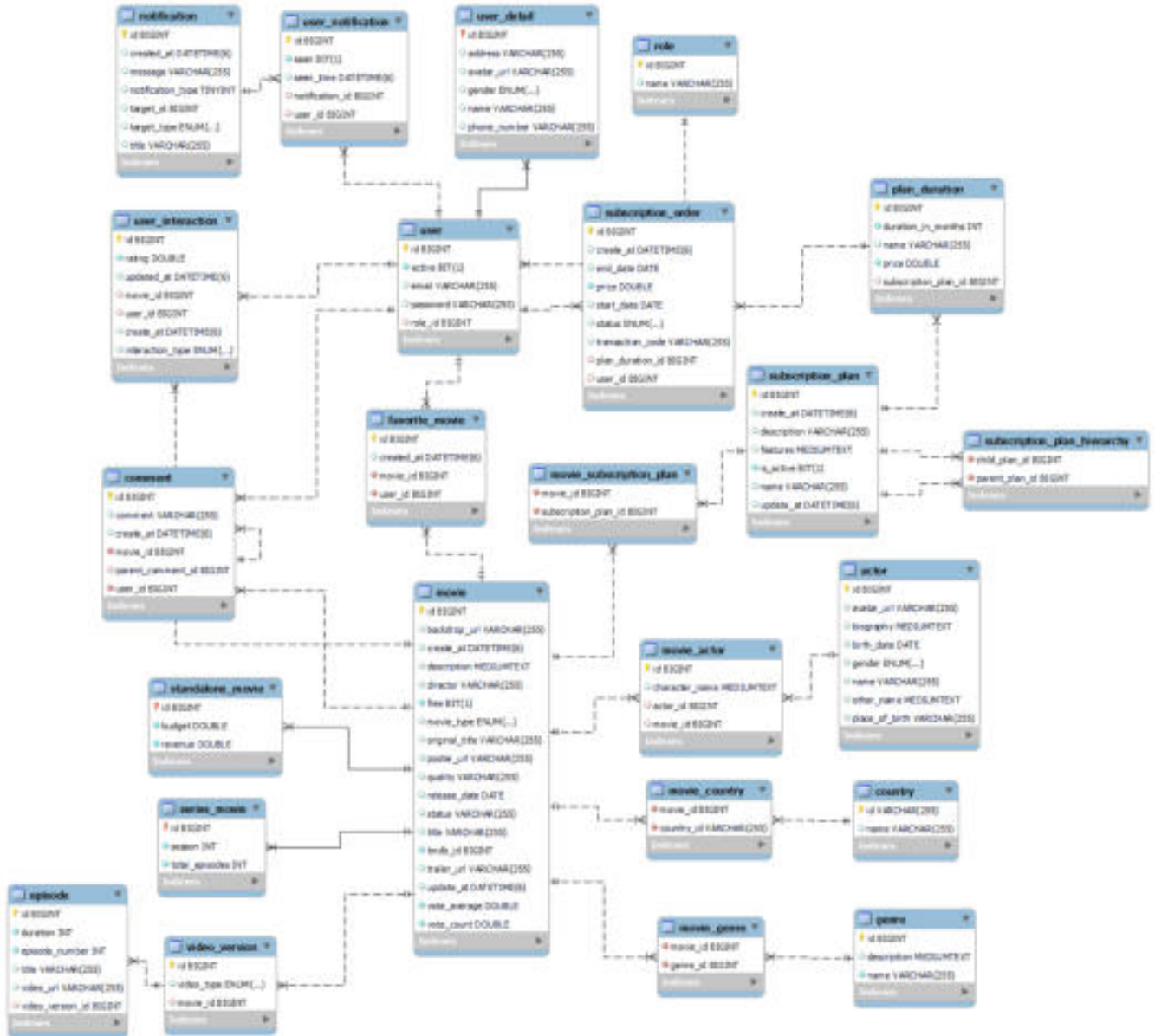
Hình 2.20 Sơ đồ tuần tự chức năng thanh toán

- Sơ đồ tuần tự thống kê



Hình 2.21 Sơ đồ tuần tự thống kê

2.4.5 Sơ đồ quan hệ cơ sở dữ liệu



Hình 2.22 Sơ đồ quan hệ cơ sở dữ liệu

Bảng 2.22 Cấu trúc bảng User

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|--------------------------|-------------|
| id | bigint | Mã người dùng | Primary key |
| email | varchar | Email dùng để đăng nhập | Unique key |
| password | varchar | Mật khẩu đã được mã hóa | |
| active | boolean | Trạng thái của tài khoản | |

Bảng 2.23 Cấu trúc bảng UserDetail

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-------------|--------------|----------------|-----------------------------|
| id | bigint | Mã người dùng | Primary key, Foreign key |
| name | varchar | Tên người dùng | |
| phoneNumber | varchar | Số điện thoại | |
| address | varchar | Địa chỉ | |

Bảng 2.24 Cấu trúc bảng Role

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-------------|--------------|---------------------------|-------------|
| id | bigint | Mã vai trò trong hệ thống | Primary key |
| roleName | varchar | Tên vai trò | Unique key |
| description | varchar | Mô tả về vai trò | |
| createAt | datetime | Thời gian tạo vai trò | |

Bảng 2.25 Cấu trúc bảng Notification

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|-----------------------------------|-------------|
| id | bigint | Mã thông báo | Primary key |
| title | varchar | Tiêu đề của thông báo | |
| message | varchar | Nội dung của thông báo | |
| type | enum | Loại thông báo | |
| targetType | enum | Loại đối tượng được nhắm đến | |
| targetId | bigint | ID của đối tượng cụ thể liên quan | Foreign key |

Bảng 2.26 Cấu trúc bảng UserNotification

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|----------------|--------------|---|-------------|
| id | bigint | Mã của bản ghi | Primary key |
| seen | boolean | Trạng thái đã xem hoặc chưa xem thông báo | |
| seenTime | datetime | Thời gian người dùng xem thông báo | |
| userId | bigint | ID của người dùng nhận thông báo | Foreign key |
| notificationId | bigint | ID của thông báo được gửi tới người dùng | Foreign key |

Bảng 2.27 Cấu trúc bảng Movie

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|---------------|--------------|-------------------|-------------|
| id | bigint | Mã định danh phim | Primary key |
| title | varchar | Tên phim | Not null |
| originalTitle | varchar | Tên gốc của phim | |

| | | | |
|-------------|----------|---|----------|
| description | text | Mô tả nội dung phim | |
| director | varchar | Tên đạo diễn phim | |
| posterUrl | varchar | URL hình ảnh poster phim | Not null |
| backdropUrl | varchar | URL hình ảnh nền (backdrop) | Not null |
| releaseDate | date | Ngày phát hành | |
| free | boolean | Trạng thái miễn phí hay không | |
| trailerUrl | varchar | URL trailer phim | |
| movieType | enum | Loại phim | |
| status | varchar | Trạng thái (đang chiếu, sắp chiếu, dừng chiếu, ...) | |
| voteAverage | double | Điểm đánh giá trung bình | |
| voteCount | double | Số lượt đánh giá | |
| quality | varchar | Chất lượng phim (HD, FullHD, 4K, ...) | |
| createAt | datetime | Thời gian tạo | |
| updateAt | datetime | Thời gian cập nhật | |

Bảng 2.28 Cấu trúc bảng Actor

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|----------------------------|-------------|
| id | bigint | Mã diễn viên | Primary key |
| name | varchar | Tên diễn viên | Not null |
| avatarUrl | varchar | URL ảnh đại diện diễn viên | |
| biography | text | Tiểu sử diễn viên | |
| birthDate | date | Ngày sinh của diễn viên | |

| | | | |
|--------------|---------|--------------------------------|--|
| otherName | text | Tên khác của diễn viên | |
| placeOfBirth | varchar | Nơi sinh | |
| gender | enum | Giới tính (Nam, Nữ, Khác, ...) | |

Bảng 2.29 Cấu trúc bảng MovieActor

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|---------------|--------------|---|-------------|
| id | bigint | Mã định danh của quan hệ diễn viên – phim | Primary key |
| characterName | text | Tên nhân vật mà diễn viên đảm nhận | |
| movieId | bigint | Mã phim | Foreign key |
| actorId | bigint | Mã diễn viên | Foreign key |

Bảng 2.30 Cấu trúc bảng Country

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|--------------|-------------|
| id | varchar | Mã quốc gia | Primary key |
| name | varchar | Tên quốc gia | Not null |

Bảng 2.31 Cấu trúc bảng MovieCountry

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|-----------------|-------------|
| movieId | Bigint | ID của phim | Foreign key |
| countryId | String | ID của quốc gia | Foreign key |

Bảng 2.32 Cấu trúc bảng Genre

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-------------|--------------|----------------|-------------|
| id | bigint | Mã thể loại | Primary key |
| name | text | Tên thể loại | Not null |
| description | text | Mô tả thể loại | |

Bảng 2.33 Cấu trúc bảng MovieGenre

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|---------------------------|-----------------------|
| movieId | bigint | Khóa ngoại đến bảng Movie | Foreign key, Not null |
| genreId | bigint | Khóa ngoại đến bảng Genre | Foreign key, Not null |

Bảng 2.34 Cấu trúc bảng StandaloneMovie

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|--------------------------------|--------------------------|
| id | bigint | Mã phim (cũng là mã của movie) | Primary key, Foreign key |
| budget | double | Kinh phí sản xuất phim | |
| revenue | double | Doanh thu của phim | |

Bảng 2.35 Cấu trúc bảng SeriesMovie

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|---------------|--------------|-----------------------------------|--------------------------|
| id | bigint | Mã phim (cũng là mã của movie) | Primary key, Foreign key |
| season | integer | Số mùa (season) của phim | |
| totalEpisodes | integer | Tổng số tập (episodes) của series | |

Bảng 2.36 Cấu trúc bảng VideoVersion

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|--|-------------|
| id | bigint | Mã phiên bản video | Primary key |
| videoType | enum | Loại phiên bản video (vietsub, voice-over, dubbed) | Not null |
| movieId | bigint | Mã phim liên kết | Foreign key |

Bảng 2.37 Cấu trúc bảng Episode

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|----------------|--------------|--|-------------|
| id | bigint | Mã tập phim | Primary key |
| title | varchar | Tiêu đề tập phim | Not null |
| episodeNumber | integer | Số thứ tự của tập trong phiên bản video | |
| duration | integer | Thời lượng của tập phim (tính bằng phút hoặc giây) | |
| videoUrl | varchar | Đường dẫn video | |
| videoVersionId | bigint | Mã phiên bản video | Foreign key |

Bảng 2.38 Cấu trúc bảng SubscriptionPlan

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-------------|--------------|--------------------------|-------------|
| id | bigint | Mã định danh gói đăng ký | Primary key |
| name | varchar | Tên gói đăng ký | Unique |
| description | varchar | Mô tả ngắn gói đăng ký | |
| features | text | Các tính năng chi tiết | |
| isActive | boolean | Trạng thái hoạt động | |
| createAt | datetime | Thời gian tạo | |

| | | | |
|----------|----------|--------------------|--|
| updateAt | datetime | Thời gian cập nhật | |
|----------|----------|--------------------|--|

Bảng 2.39 Cấu trúc bảng ScriptionMovie

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|--------------------|--------------|--------------------|-------------|
| movieId | bigint | Mã của phim | Foreign key |
| subscriptionPlanId | bigint | Mã của gói đăng ký | Foreign key |

Bảng 2.40 Cấu trúc bảng SubscriptionPlanHierarchy

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|--------------|--------------|-----------------|-------------|
| childPlanId | bigint | Gói đăng ký con | Foreign key |
| parentPlanId | bigint | Gói đăng ký cha | Foreign key |

Bảng 2.41 Cấu trúc bảng PlanDuration

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|--------------------|--------------|----------------------------------|-------------|
| id | bigint | Mã định danh thời lượng gói | Primary key |
| name | varchar | Tên của thời lượng | |
| price | double | Giá tiền của gói theo thời lượng | |
| durationInMonths | interger | Thời lượng tính theo tháng | |
| subscriptionPlanId | bigint | ID của gói đăng ký liên quan | Foreign key |

Bảng 2.42 Cấu trúc bảng SubscriptionOrder

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-----------------|--------------|---|-------------|
| id | bigint | Mã định danh đơn đăng ký | Primary key |
| startDate | date | Ngày bắt đầu gói | |
| endDate | date | Ngày kết thúc gói | |
| price | double | Giá tiền thanh toán | |
| transactionCode | varchar | Mã giao dịch (ví dụ: mã ví, ngân hàng...) | |
| status | enum | Trạng thái đơn đăng ký | |
| createAt | datetime | Thời điểm tạo | |
| userId | bigint | ID người dùng | Foreign key |
| planDurationId | bigint | ID gói thời lượng đã chọn | Foreign key |

Bảng 2.43 Cấu trúc bảng FavoriteMovie

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|------------|--------------|----------------------------------|-------------|
| id | bigint | Mã định danh yêu thích | Primary key |
| createdAt | datetime | Thời điểm đánh dấu yêu thích | |
| userId | bigint | ID người dùng đánh dấu yêu thích | Foreign key |
| movieId | bigint | ID phim được đánh dấu yêu thích | Foreign key |

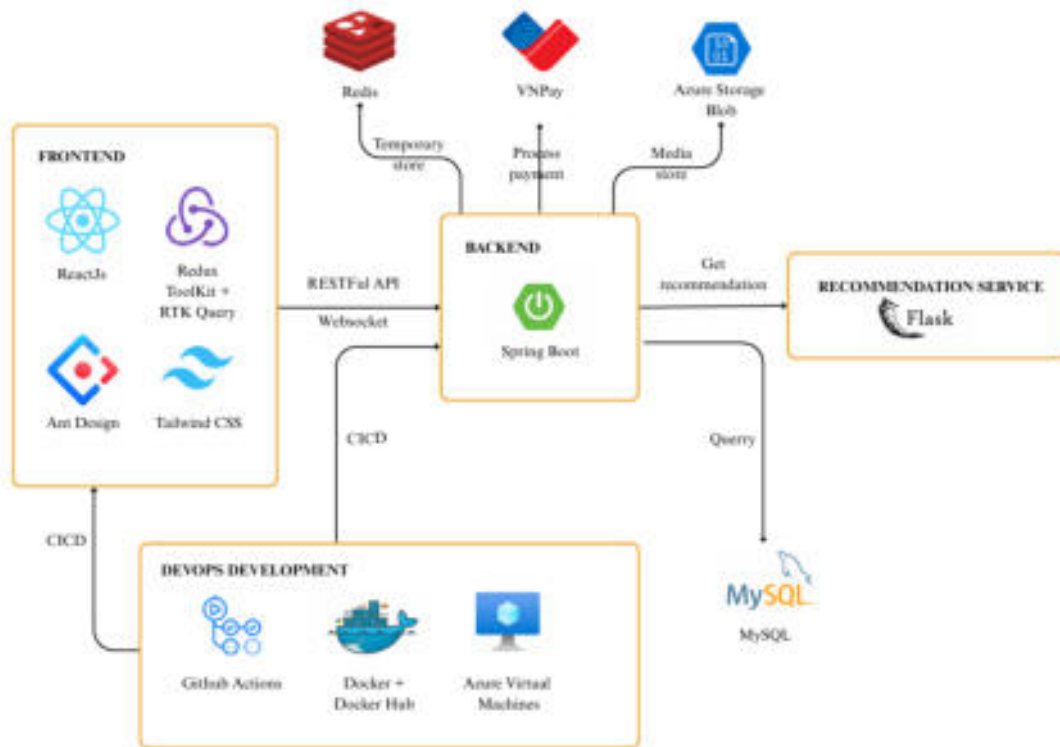
Bảng 2.44 Cấu trúc bảng Comment

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-----------------|--------------|------------------------|-------------|
| id | bigint | Mã bình luận | Primary key |
| comment | text | Nội dung bình luận | |
| createAt | datetime | Thời gian tạo | |
| userId | bigint | Người bình luận | Foreign key |
| movieId | bigint | Phim được bình luận | Foreign key |
| parentCommentId | bigint | Bình luận cha (nếu có) | Foreign key |

Bảng 2.45 Cấu trúc bảng UserInteraction

| Tên trường | Kiểu dữ liệu | Mô tả | Ràng buộc |
|-----------------|--------------|--|-------------|
| id | bigint | Mã định danh đánh giá | Primary key |
| rating | double | Số điểm đánh giá (ví dụ: 4.5) | |
| interactionType | enum | Loại tương tác với phim (Rating, View, Like) | |
| createdAt | datetime | Thời điểm tạo tương tác | |
| updatedAt | datetime | Thời điểm đánh giá được cập nhật | |
| userId | bigint | Người dùng đánh giá | Foreign key |
| movieId | bigint | Phim được đánh giá | Foreign key |

2.5 Thiết kế tổng quan về kiến trúc hệ thống



Hình 2.23 Sơ đồ kiến trúc tổng quát hệ thống

Bảng 2.46 Bảng nhiệm vụ các thành phần trong sơ đồ kiến trúc tổng quát hệ thống

| Thành phần | Nhiệm vụ |
|--------------------|--|
| Frontend | Xây dựng giao diện người dùng bằng ReactJS, sử dụng Ant Design, TailwindCSS để thiết kế UI; Redux Toolkit và RTK Query để quản lý trạng thái và gọi API. |
| Backend | Xử lý RESTful API, giao tiếp với MySQL, Redis, Recommendation Service; xử lý upload hình ảnh lên Azure Storage Blob và logic thanh toán với VNPay. |
| MySQL | Cơ sở dữ liệu chính để lưu trữ thông tin phim, người dùng, các giao dịch, dịch vụ, ... |
| Redis: | Lưu trữ tạm thời các thông tin nhạy cảm như mã OTP, refresh token, token đặt lại mật khẩu và dữ liệu đăng ký chưa xác thực. |
| Azure Storage Blob | Lưu trữ các tài nguyên phương tiện như hình ảnh poster, ảnh nền và video phim. |

| | |
|------------------------|---|
| Recommendation Service | Gợi ý phim dựa trên metadata (thể loại, diễn viên, đạo diễn, từ khóa...) hoặc hành vi, đánh giá của người dùng. |
| VNPay | Công thanh toán trực tuyến, xử lý giao dịch mua các gói dịch vụ từ người dùng. |
| DevOps Development | Sử dụng Docker, Docker Hub, GitHub Actions và Azure Virtual Machine để thiết lập quy trình CI/CD và triển khai hệ thống frontend/backend. |

2.6 Kết luận chương

Chương này đã trình bày chi tiết quá trình phân tích và thiết kế hệ thống, đóng vai trò quan trọng trong việc hiện thực hóa các mục tiêu của đề tài. Trước hết, chương đã xác định rõ các yêu cầu chức năng và phi chức năng của hệ thống, đảm bảo hệ thống đáp ứng đúng nhu cầu thực tế của người dùng và có thể mở rộng, bảo trì dễ dàng trong tương lai.

Phần thiết kế chức năng được mô hình hóa thông qua các công cụ trực quan như biểu đồ use-case, giúp hình dung rõ các tương tác giữa người dùng và hệ thống. Mỗi chức năng chính đều được mô tả chi tiết bằng sơ đồ đặc tả use-case, kèm theo sơ đồ hoạt động và sơ đồ tuần tự, từ đó thể hiện rõ luồng xử lý và quy trình tương tác giữa các thành phần trong hệ thống.

Bên cạnh đó, chương cũng đã trình bày thiết kế cơ sở dữ liệu, thông qua sơ đồ cơ sở dữ liệu (database diagram), thể hiện mối quan hệ giữa các bảng dữ liệu như người dùng, phim, đánh giá, giao dịch,... Thiết kế cơ sở dữ liệu không chỉ đảm bảo tính nhất quán và tối ưu hóa truy vấn, mà còn hỗ trợ tốt cho việc triển khai các chức năng nâng cao như hệ thống gợi ý cá nhân hóa.

Cuối cùng, chương đã giới thiệu kiến trúc tổng quan của hệ thống, mô tả rõ cách các thành phần như frontend, backend, recommendation service, cơ sở dữ liệu và các dịch vụ phụ trợ (Redis, VNPay, Azure Blob Storage, CI/CD DevOps) phối hợp với nhau để tạo thành một hệ thống thống nhất. Kiến trúc này đảm bảo sự phân tách rõ ràng giữa các tầng, giúp hệ thống dễ mở rộng và dễ bảo trì về sau.

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG

3.1 Quy trình triển khai hệ thống gợi ý

3.1.1 Mô hình gợi ý lọc theo nội dung (CBF)

❖ Thu thập và tiền xử lý dữ liệu:

- Thu thập dữ liệu về phim từ nhiều nguồn (TMDB, KKPhim, Kaggle, ...), bao gồm tên phim, mô tả, thể loại, thông tin đoàn làm phim (đạo diễn, diễn viên) và từ khóa, videoUrl, trailer, ...
- Thông tin dataset:
 - Số lượng phim lẻ: 2512
 - Số lượng phim bộ: 1184
 - Số lượng đánh giá của phim lẻ: 597165
(Không có dữ liệu đánh giá cho phim bộ)
- Dữ liệu được làm sạch, chuẩn hóa và hợp nhất để đảm bảo tính nhất quán và sẵn sàng cho các bước xử lý tiếp theo.

❖ Trích xuất và xử lý đặc trưng

Đây là bước quan trọng nhằm tạo ra các “hồ sơ đặc trưng” cho từng bộ phim, giúp hệ thống hiểu được nội dung và tính chất của từng phim dựa trên dữ liệu có sẵn. Các đặc trưng được trích xuất bao gồm:

- Mô tả phim: được xây dựng từ phân tóm tắt nội dung (overview) và khẩu hiệu (tagline) của bộ phim.
- Đạo diễn và diễn viên: tên của đạo diễn và các diễn viên chính được trích xuất, chuẩn hóa và đưa về dạng thống nhất.
- Từ khóa nội dung: các từ khóa quan trọng mô tả nội dung phim được chọn lọc. Sau đó, các từ này được xử lý bằng kỹ thuật Stemming (ví dụ: "running" → "run", "fishes" → "fish") nhằm đưa về dạng gốc, giúp nhận diện chính xác hơn khi so sánh.

Tất cả các đặc trưng trên sau khi được xử lý sẽ được kết hợp lại thành một chuỗi văn bản duy nhất, đại diện cho nội dung của mỗi bộ phim.

❖ Xây dựng ma trận tương đồng

Sau khi đã kết hợp và chuẩn hóa các thông tin đặc trưng như mô tả, thể loại, đạo diễn, diễn viên và từ khóa, mỗi bộ phim sẽ được biểu diễn dưới dạng một chuỗi văn bản đại diện cho nội dung của nó. Hệ thống sử dụng TfidfVectorizer để chuyển đổi các văn bản này thành các vector số học. Mỗi vector có nhiều chiều, tương ứng với các đặc trưng nội dung có trọng số khác nhau.

Sau đó, hệ thống tính toán độ tương đồng giữa các bộ phim bằng cách áp dụng Cosine Similarity cho từng cặp vector. Từ đó, tạo ra một ma trận tương đồng thể hiện mức độ giống nhau về mặt nội dung giữa các phim. Ma trận này là cơ sở để hệ thống xác định và đề xuất những bộ phim có nội dung tương tự cho người dùng.

❖ Cơ chế gợi ý và xếp hạng:

- Khi người dùng chọn một bộ phim, hệ thống sẽ sử dụng ma trận tương đồng để tìm ra các bộ phim khác có nội dung **tương tự nhất**.
- Để đảm bảo chất lượng gợi ý, áp dụng thêm một **hệ số xếp hạng có trọng số** (Weighted Rating). Hệ số này giúp ưu tiên những bộ phim không chỉ tương đồng về nội dung mà còn có **đánh giá cao và nhiều lượt bình chọn**, phản ánh mức độ phổ biến và đáng tin cậy của phim.
- Công thức tính hệ số

$$WR = (v/(v+m)) \times R + (m/(v+m)) \times C$$

Trong đó:

v: số lượt vote của phim

m: ngưỡng tối thiểu số vote

R: rating trung bình của phim

C: rating trung bình của toàn bộ dataset

- Kết quả cuối cùng là danh sách các bộ phim được gợi ý, sắp xếp theo mức độ phù hợp nhất.

❖ Kết quả

Với phim có tên là “Harry Potter and the Chamber of Secrets” thì hệ thống đề xuất các phim tương tự như sau:

| | title | vote_count | vote_average | year | similarity | weighted_rating |
|------|--|------------|--------------|------|------------|-----------------|
| 90 | Harry Potter and the Philosopher's Stone | 7188 | 7 | 2001 | 0.795919 | 6.759751 |
| 1941 | Harry Potter and the Deathly Hallows: Part 2 | 6141 | 7 | 2011 | 0.252447 | 6.729855 |
| 471 | Harry Potter and the Prisoner of Azkaban | 6037 | 7 | 2004 | 0.588602 | 6.726474 |
| 756 | Harry Potter and the Goblet of Fire | 5758 | 7 | 2005 | 0.463415 | 6.716972 |
| 1801 | Harry Potter and the Deathly Hallows: Part 1 | 5708 | 7 | 2010 | 0.252447 | 6.715199 |
| 1069 | Harry Potter and the Order of the Phoenix | 5633 | 7 | 2007 | 0.552638 | 6.712497 |
| 1486 | Harry Potter and the Half-Blood Prince | 5435 | 7 | 2009 | 0.401754 | 6.705112 |
| 2243 | Hotel Transylvania | 2622 | 6 | 2012 | 0.156174 | 6.000000 |
| 3114 | Pixels | 2564 | 5 | 2015 | 0.398167 | 5.469919 |
| 2355 | Oz: The Great and Powerful | 3576 | 5 | 2013 | 0.171080 | 5.388613 |

Hình 3.1 Các phim được đề cử bởi CBF

Kết quả trả về từ server flask khi sử dụng CBF engine

```
{
  "cbf_score": 0.7959190572427999,
  "genres": "Adventure|Fantasy|Family",
  "method": "CBF_Only",
  "title": "Harry Potter and the Philosopher's Stone",
  "tmdb_id": 671,
  "vote_average": 7
}
```

Hình 3.2 Phản hồi API từ CBF Engine

Trong đó:

- cbf_score: điểm tương đồng giữa phim với phim được đề xuất
- genres: thể loại của phim được đề xuất
- method: tên engine đề xuất phim
- title: tên của phim
- tmdb_id: mã phim trên TMDB
- vote_avarage: điểm trung bình của movie

3.1.2 Mô hình NeuMF:

Chuẩn bị dữ liệu đầu vào

Chuyển đổi dữ liệu: Dữ liệu tương tác người dùng-phim từ các file CSV gốc được chuyển đổi sang định dạng của RecBole (file .inter, .item). File .inter là cốt lõi, chứa các tương tác với các trường bắt buộc như user_id, item_id, rating và datetime.

Cấu hình tham số mô hình NeuMF trên RecBole

❖ Kiến trúc mô hình (Model Configuration):

- Kích thước Embedding: mf_embedding_size: 128 và mlp_embedding_size: 128 cho các vector biểu diễn người dùng và phim
- Cấu trúc MLP: mlp_hidden_size: [256, 128, 64] với 3 lớp ẩn
- Chống Overfitting: dropout_prob: 0.2

❖ Cấu hình Huấn luyện (Training Configuration):

- Số Epoch: 100 vòng lặp huấn luyện toàn bộ dữ liệu.
- Kích thước Batch: train_batch_size: 1024 mẫu tương tác mỗi lần cập nhật.
- Tốc độ học: learning_rate: 0.0005, kiểm soát tốc độ điều chỉnh trọng số.
- Tối ưu hóa: Sử dụng thuật toán Adam và weight_decay: 0.00001 (L2 regularization) để tăng cường hiệu quả.
- Đánh giá định kỳ: eval_step: 5 để kiểm tra hiệu suất trên tập validation mỗi 5 epoch.

❖ Chia dữ liệu (Random Split):

- Dữ liệu được chia ngẫu nhiên theo tỷ lệ 80% huấn luyện, 10% validation, 10% kiểm tra, đảm bảo đánh giá công bằng.
- Giữ nguyên thứ tự tương tác

❖ Lấy mẫu tiêu cực (Negative Sampling):

- Sử dụng phân phối uniform để lấy 4 mẫu tiêu cực (sample_num: 4) cho mỗi tương tác tích cực, giúp mô hình học cách phân biệt rõ hơn giữa tương tác tích cực và

tiêu cực. Các mẫu này được giữ cố định (dynamic: false) trong suốt quá trình huấn luyện.

❖ Trường dữ liệu và Tải cột (Data Fields & Load Columns):

- Các cột chính được sử dụng từ file .inter bao gồm: user_id, item_id, rating, và datetime.

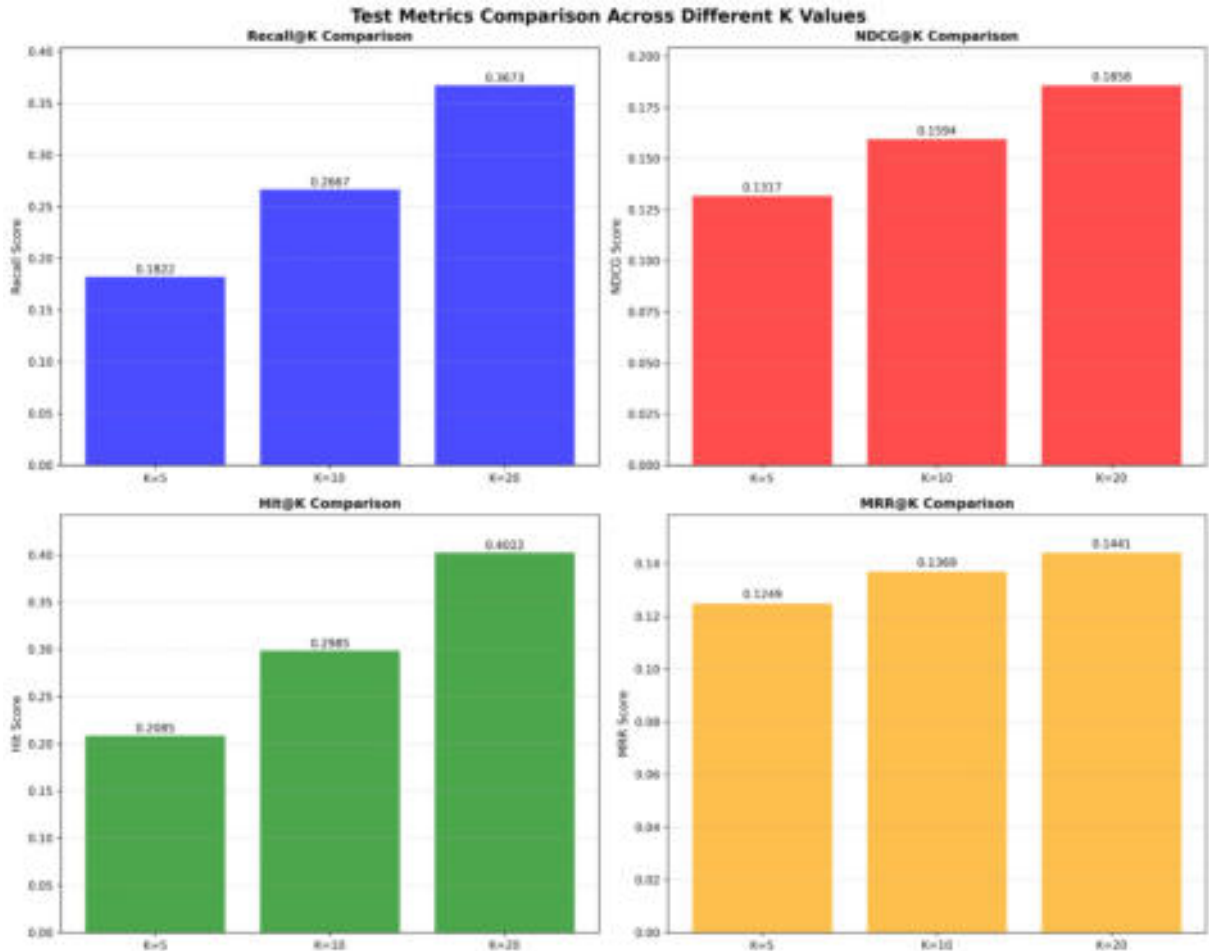
Đánh giá mô hình

Việc đánh giá mô hình dùng các metric sau:

- Recall@K: Đo lường tỷ lệ các mục liên quan thực sự được mô hình tìm thấy và đề xuất trong danh sách top K gợi ý.
- NDCG@K: Đánh giá không chỉ việc các mục liên quan có được gợi ý hay không, mà còn vị trí của chúng trong danh sách. Các mục liên quan xuất hiện ở vị trí càng cao (gần đầu danh sách) thì điểm càng cao.
- Hit@K: Cho biết tỷ lệ người dùng nhận được ít nhất một mục liên quan trong danh sách top K gợi ý của họ.
- MRR@K: Đo lường vị trí trung bình của mục liên quan đầu tiên trong danh sách gợi ý. Nếu mục đầu tiên đúng nằm ở vị trí r, điểm là $1/r$.

Kết quả

Kết quả đánh giá dựa trên các metric của tập kiểm tra



Hình 3.3 Kết quả đánh giá mô hình trên tập kiểm tra

Nhận xét

Từ các biểu đồ kết quả đánh giá trên tập kiểm thử, có thể thấy rằng tất cả các chỉ số hiệu suất (Recall@K, NDCG@K, Hit@K, MRR@K) đều có xu hướng tăng lên đáng kể khi giá trị K tăng dần. Điều này khẳng định hiệu quả của mô hình trong việc gợi ý phim.

Cụ thể:

- Recall@K và Hit@K tăng cho thấy mô hình có khả năng bao phủ tốt, tức là các bộ phim mà người dùng quan tâm có xác suất cao xuất hiện trong danh sách đề xuất.
- NDCG@K và MRR@K cũng tăng, minh chứng rằng hệ thống không chỉ gợi ý được các phim liên quan mà còn xếp hạng chúng một cách hiệu quả, ưu tiên đặt các phim có độ liên quan cao nhất ở những vị trí hàng đầu trong danh sách. Điều này đặc biệt quan trọng để tối ưu trải nghiệm người dùng, giúp họ nhanh chóng tìm thấy nội dung phù hợp.

```
{
  "genres": "Action|Science_Fiction|Adventure",
  "method": "NeuMF_User",
  "neumf_score": 0.874910912032235,
  "predicted_rating": 4.49964364812894,
  "title": "Guardians of the Galaxy",
  "tmdb_id": 118340,
  "vote_average": 7.9
},
```

Hình 3.4 Phản hồi API từ NeuMF Engine

Kết quả trả về từ server flask khi sử dụng NeuMF engine

Trong đó:

- genres: thể loại của phim được đề xuất
- method: tên engine đề xuất phim
- neumf_score: điểm đánh giá phù hợp giữa phim và người dùng
- predicted_rating: điểm đánh giá dự đoán của người dùng
- title: tên của phim
- tmdb_id: mã phim trên TMDB
- vote_avarage: điểm trung bình của movie

3.1.3 Kết hợp CBF và NeuMF

Bước 1: Thu thập kết quả từ cả hai engine

- CBF Engine: Lấy danh sách phim tương tự dựa trên nội dung của phim đầu vào
- NeuMF Engine: Lấy danh sách phim được cá nhân hóa dựa trên user_id và movie_id hiện tại
- Mở rộng pool ứng viên bằng cách lấy số lượng gấp 3 lần yêu cầu từ mỗi engine

Bước 2: Cross-Calculation (Tính Toán Chéo)

- Cho phim từ CBF: Tính thêm NeuMF score bằng batch prediction
- Cho phim từ NeuMF: Tính thêm CBF similarity score bằng cosine similarity matrix
- Đảm bảo mọi phim ứng viên đều có cả CBF score và NeuMF score

Bước 3: Áp dụng công thức Hybrid

$$\text{hybrid_score} = \text{cbf_weight} * \text{cbf_score} + \text{neumf_weight} * \text{neumf_score} + \text{wr_weight} * \text{normalized_wr_score}$$

Trong đó:

- `cbf_weight`: Trọng số cho Content-Based Filtering
- `neumf_weight`: Trọng số cho Neural Matrix Factorization
- `wr_weight`: Trọng số cho Weighted Rating (WR).
- `cbf_score`: điểm tương đồng từ CBF engine
- `neumf_score`: điểm tương đồng từ NeuMF engine
- `normalized_wr_score`: Điểm phổ biến của phim (được tính bằng **Weighted Rating**) sau khi được chuẩn hóa.

Bước 4: Ranking và trả kết quả

- Sắp xếp theo `hybrid_score` giảm dần
- Trả về metadata chi tiết

```
{
  "cbf_score": 0.7959190572427999,
  "genres": "Adventure|Fantasy|Family",
  "hybrid_score": 0.7538487819197613,
  "method": "Hybrid",
  "neumf_score": 0.6681075013642529,
  "predicted_rating": 3.6724300054570116,
  "title": "Harry Potter and the Philosopher's Stone",
  "tmdb_id": 671,
  "vote_average": 7
}
```

Hình 3.5 Phản hồi API từ Hybrid Engine

Kết quả trả về từ server flask khi sử dụng Hybrid engine

Trong đó:

- `cbf_score`: điểm tương đồng giữa phim với phim được đề xuất bởi CBF engine
- `genres`: thể loại của phim được đề xuất
- `hybrid_score`: điểm từ Hybrid engine

- method: tên engine đề xuất phim
- neumf_score: điểm từ NeuMF engine
- predicted_rating: điểm đánh giá dự đoán của người dùng
- title: tên của phim
- tmdb_id: mã phim trên TMDB
- vote_avarage: điểm trung bình của movie

3.2 Công cụ phát triển

❖ Ngôn ngữ, framework:

- Backend:
 - SpringBoot (Java): Phát triển các dịch vụ chính phía Server
 - Flask (Python): Xây dựng các module AI và hệ thống gợi ý phim
- Frontend
 - ReactJS (JavaScript): Thư viện xây dựng giao diện người dùng
 - Redux Toolkit + RTK Query: Quản lý trạng thái và gọi API
 - Thư viện UI: Ant Design, TailwindCSS

❖ Cơ sở dữ liệu:

- MySQL: Lưu trữ dữ liệu có cấu trúc của hệ thống như thông tin người dùng, phim, gói dịch vụ, đánh giá,...

❖ Quản lý mã nguồn:

- Git, Github: Lưu trữ và quản lý mã nguồn, kết hợp triển khai CI/CD

❖ Lưu trữ hình ảnh:

- Azure Blob Storage: lưu trữ các file data phương tiện như poster phim, ảnh nền phim

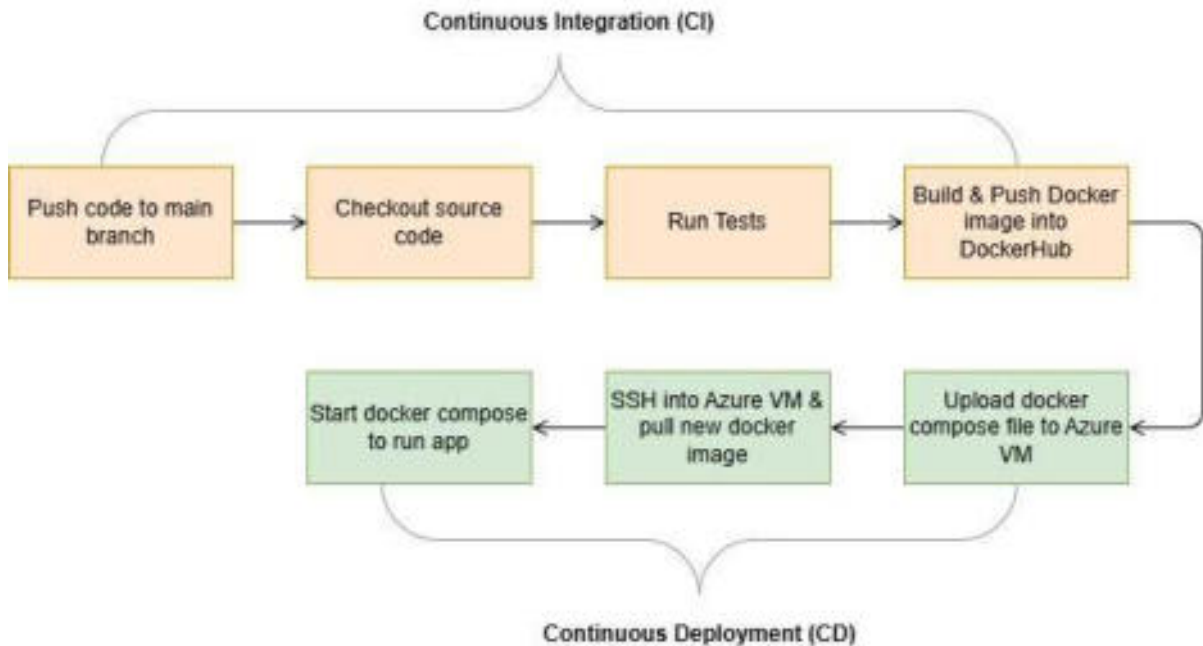
❖ Bộ nhớ tạm thời:

- Redis: sử dụng để lưu trữ tạm thời các dữ liệu nhạy cảm như OTP, refresh token và token khôi phục mật khẩu

❖ Triển khai hệ thống:

- Docker, Docker Compose: Đóng gói, quản lý các container của từng dịch vụ
- Docker Hub: Kho lưu trữ image của các service
- Github Actions: Tự động hóa quá trình CI/CD
- Azure Virtual Machine : Máy chủ triển khai hệ thống thực tế

3.3 Quy trình triển khai hệ thống CI/CD



Hình 3.6 Sơ đồ triển khai hệ thống CI/CD cho backend

Hình trên minh họa quy trình triển khai CI/CD cho hệ thống backend sử dụng GitHub Actions.

Continuous Integration (CI) bắt đầu khi code lên nhánh main. Hệ thống tự động:

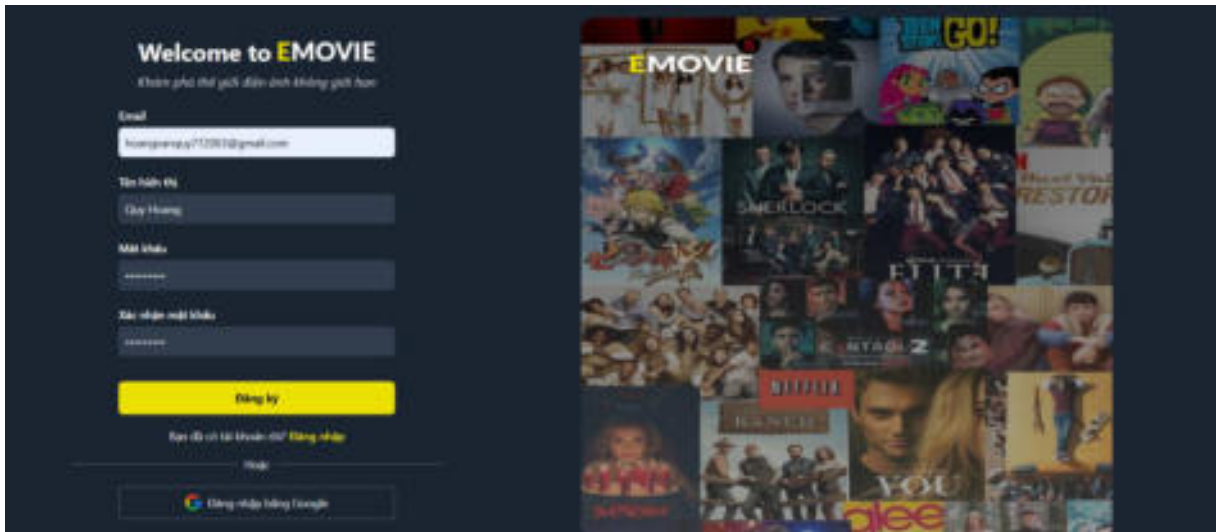
- Lấy mã nguồn (checkout),
- Cài đặt môi trường build,
- Thực hiện kiểm thử (Unit Test, Integration Test),
- Sau đó build ứng dụng và đóng gói thành Docker image,
- Cuối cùng đẩy image lên DockerHub.

Continuous Deployment (CD) tiếp tục với việc:

- Tự động upload file docker-compose.yaml lên máy ảo (Azure VM),
- SSH vào máy ảo để pull image mới nhất,
- Và sử dụng Docker Compose để khởi chạy lại các container của ứng dụng.

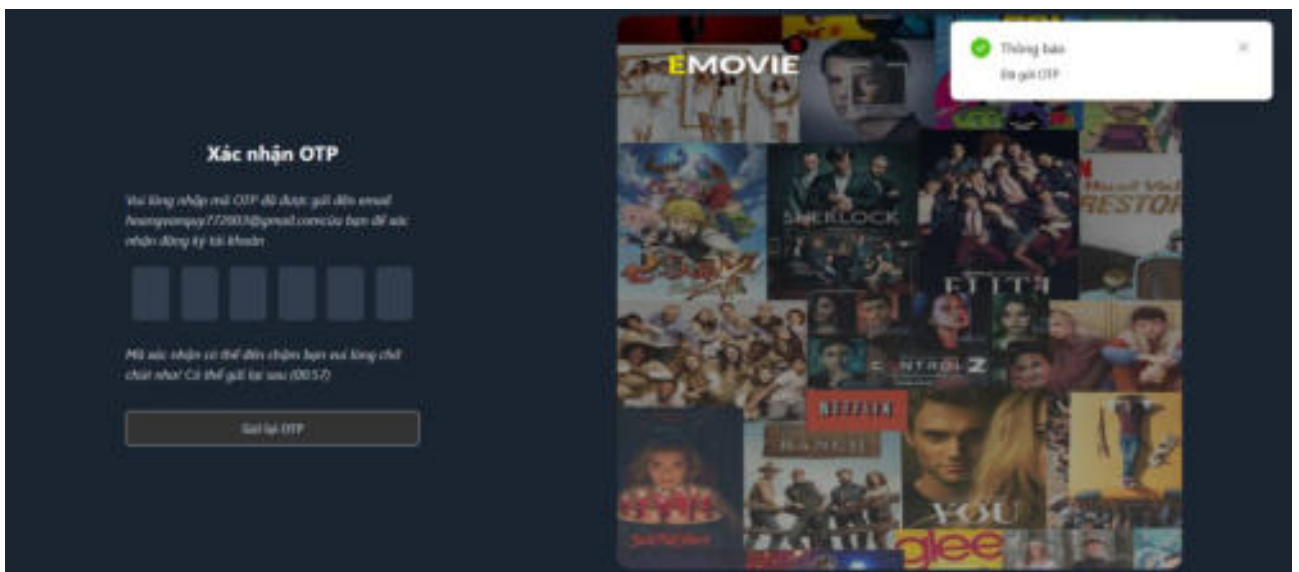
3.4 Kết quả

3.4.1 Trang đăng ký tài khoản



Hình 3.7 Trang đăng ký tài khoản

3.4.2 Màn hình nhập OTP để tiến hành đăng ký



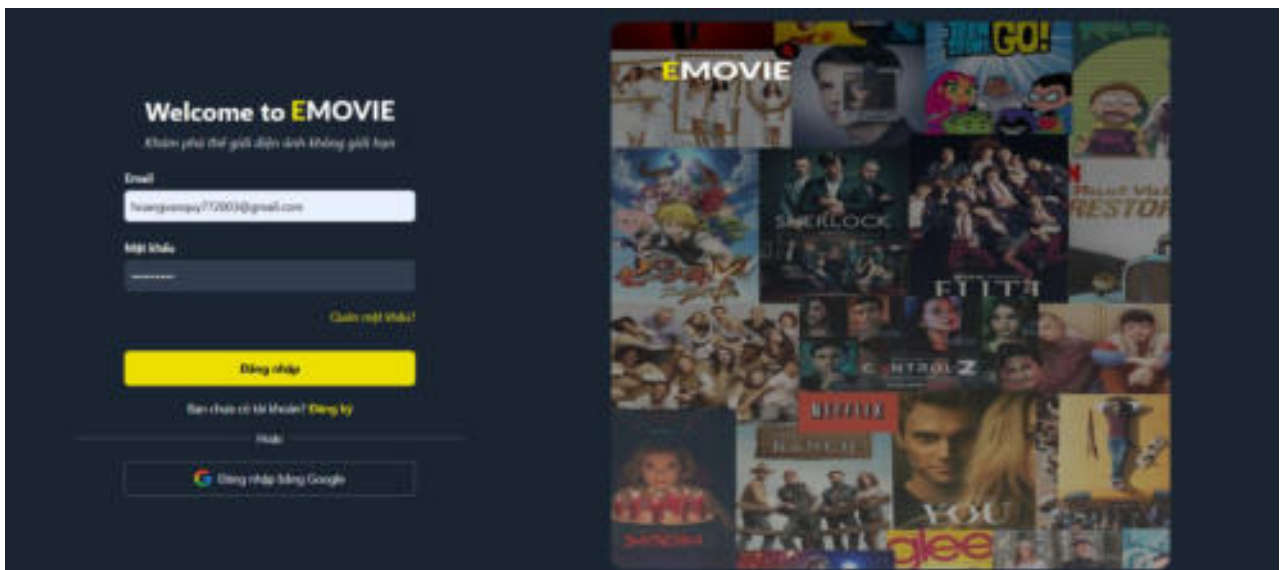
Hình 3.8 Màn hình nhập OTP

3.4.3 Email chứa mã OTP khi đăng ký



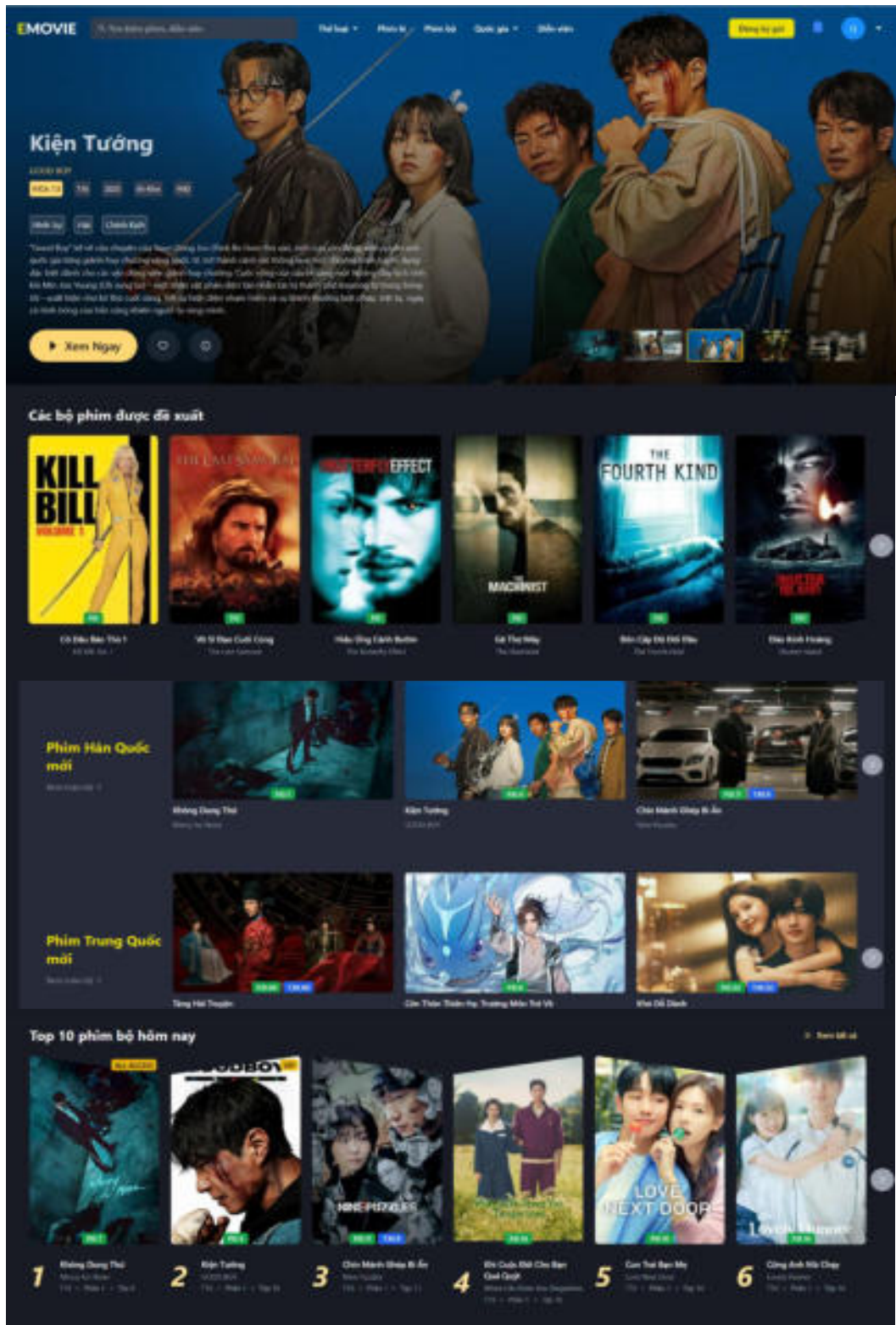
Hình 3.9 Email chứa OTP khi đăng ký

3.4.4 Trang đăng nhập



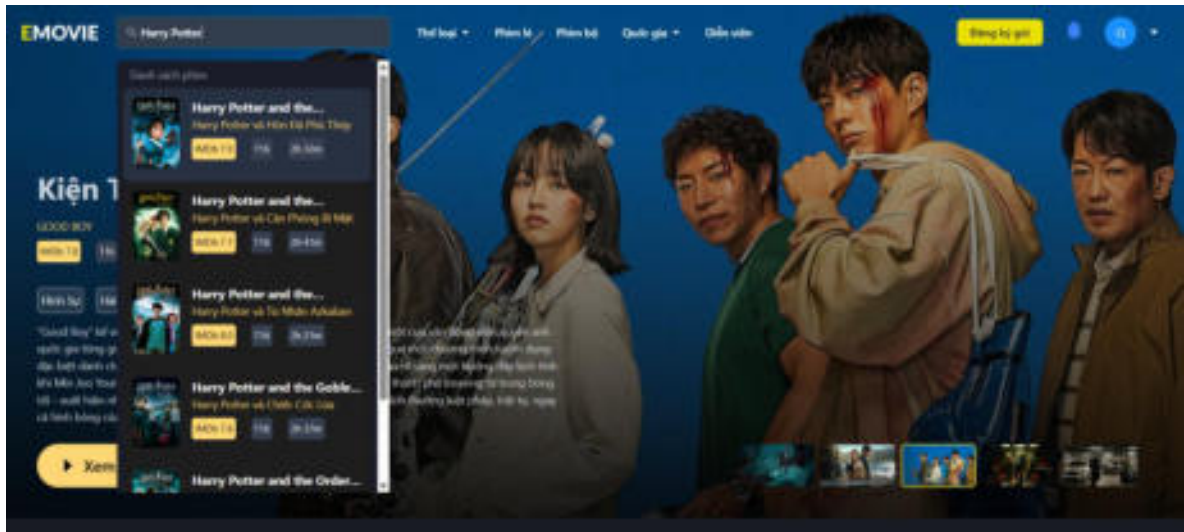
Hình 3.10 Trang đăng nhập

3.4.5 Trang chủ của người dùng



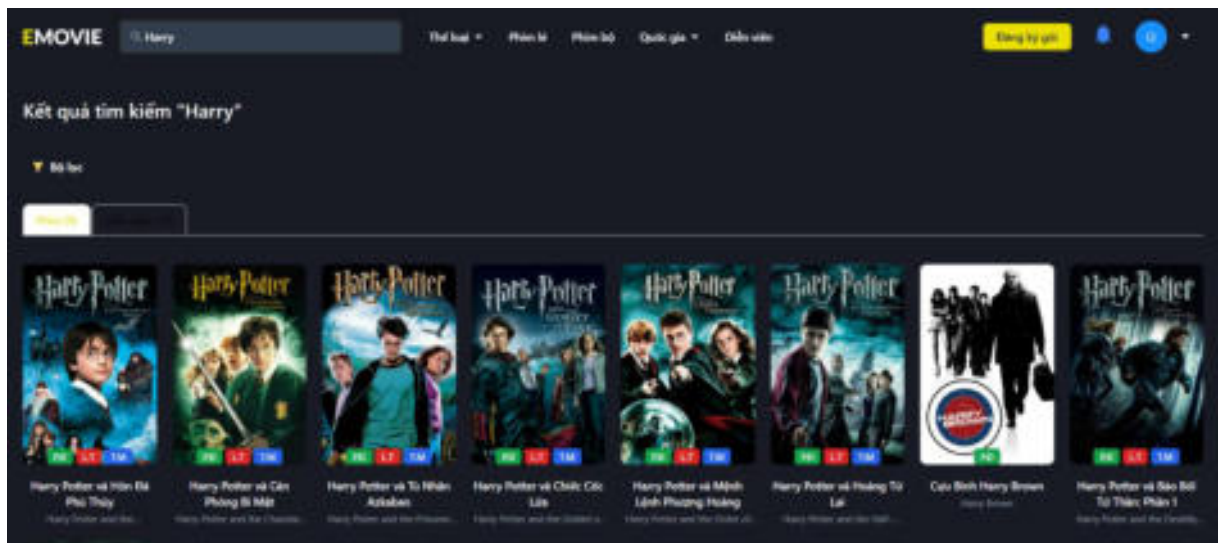
Hình 3.11 Trang chủ của người dùng

3.4.6 Màn hình tìm kiếm phim



Hình 3.12 Màn hình tìm kiếm phim

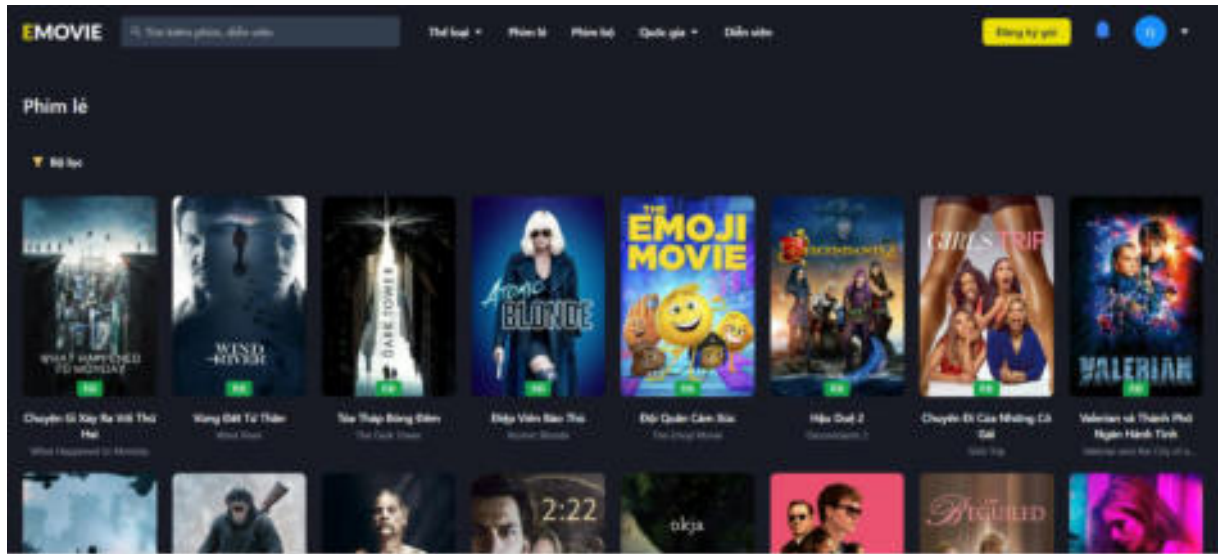
3.4.7 Kết quả tìm kiếm



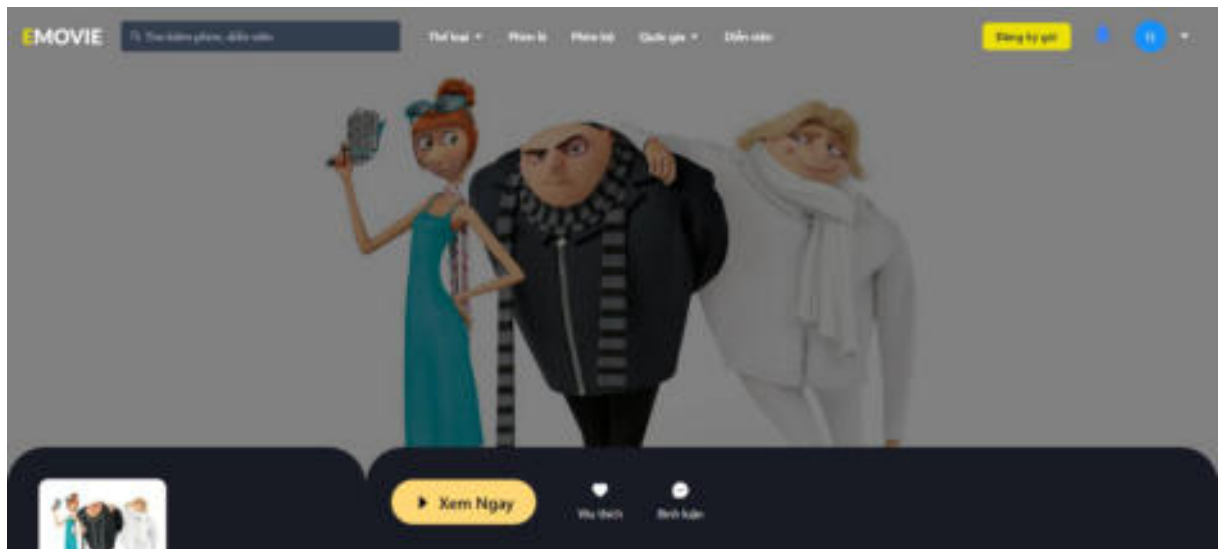
Hình 3.13 Kết quả tìm kiếm

3.4.8 Các màn hình phim

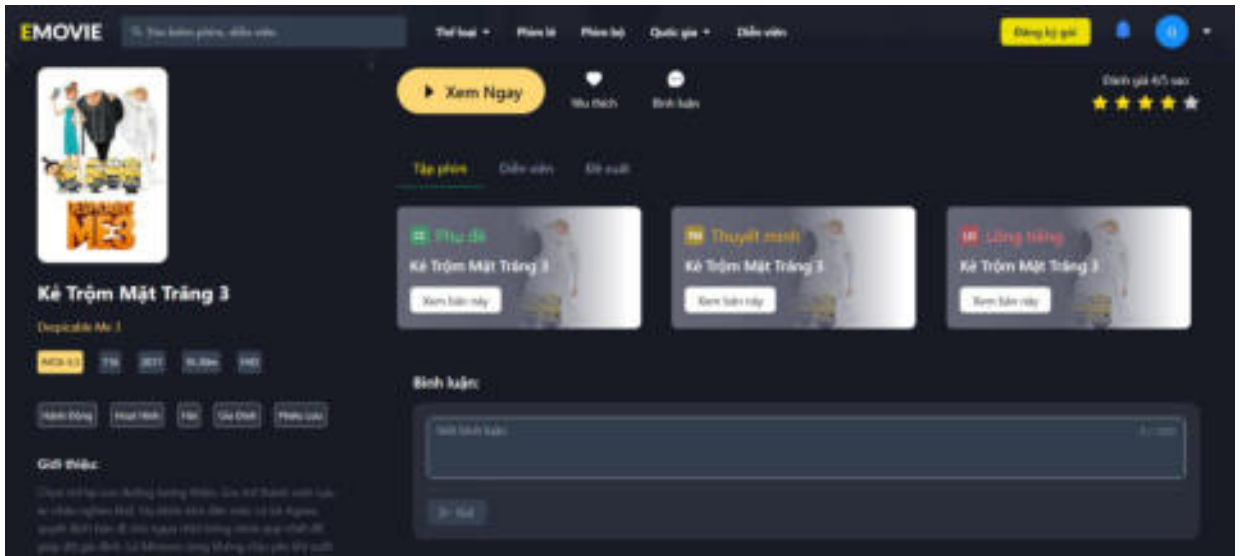
- **Phim lẻ**



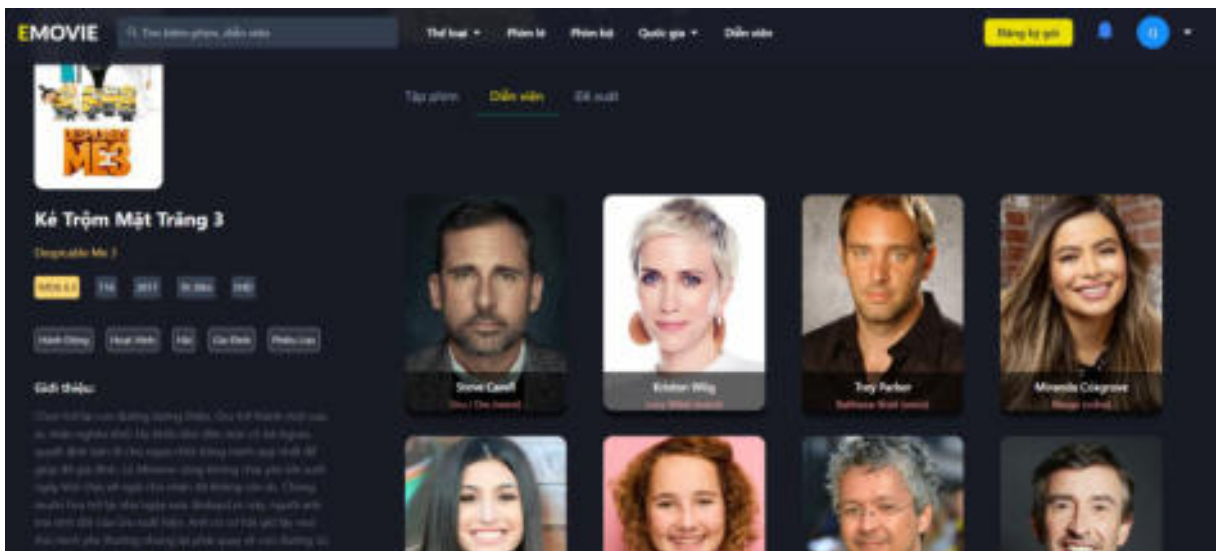
Hình 3.14 Danh sách phim lẻ



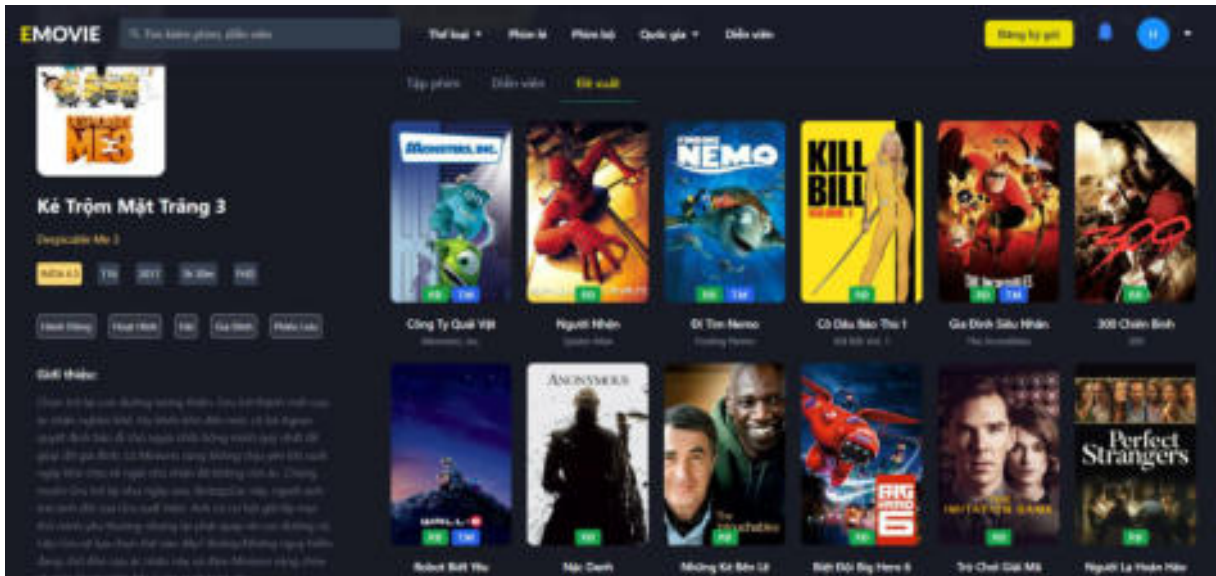
Hình 3.15 Phim lẻ



Hình 3.16 Phiên bản chiếu của phim lẻ



Hình 3.17 Diễn viên và vai diễn trong phim

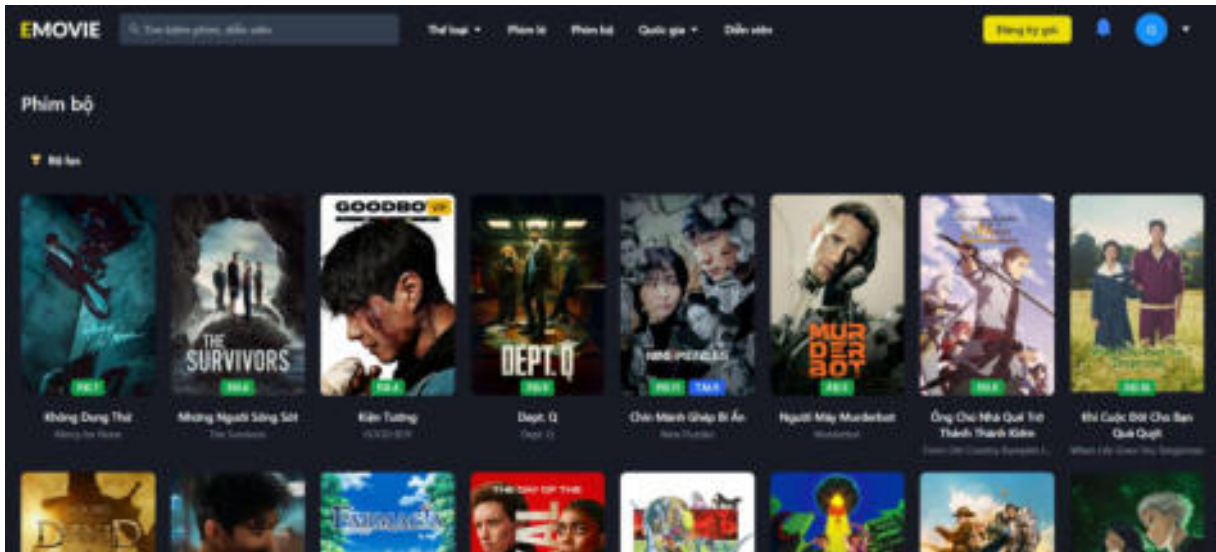


Hình 3.18 Danh sách phim được đề xuất bởi mô hình

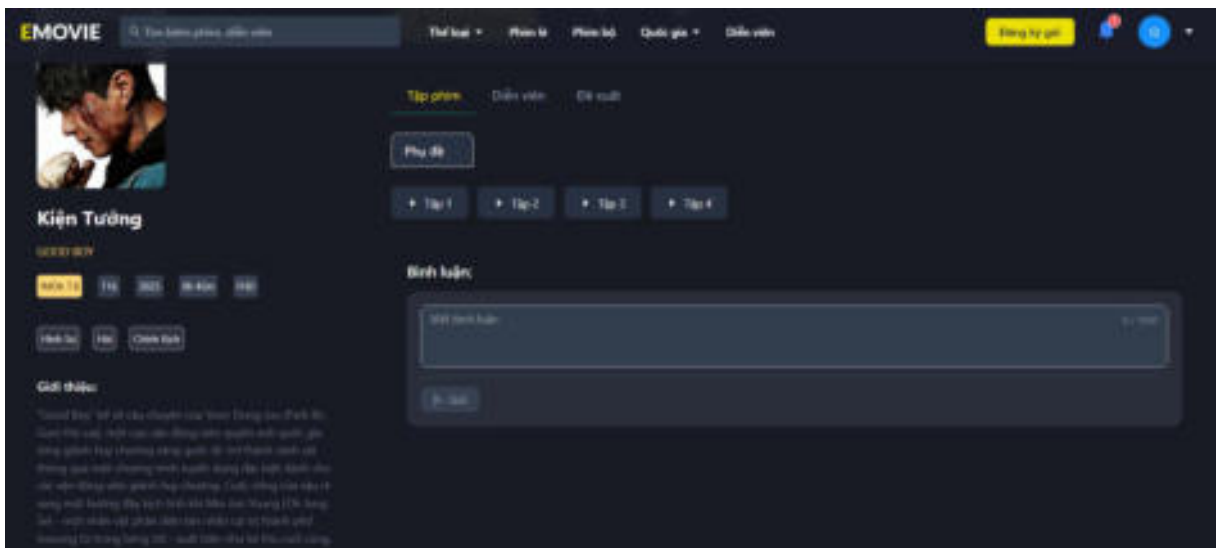


Hình 3.19 Màn hình chiếu phim

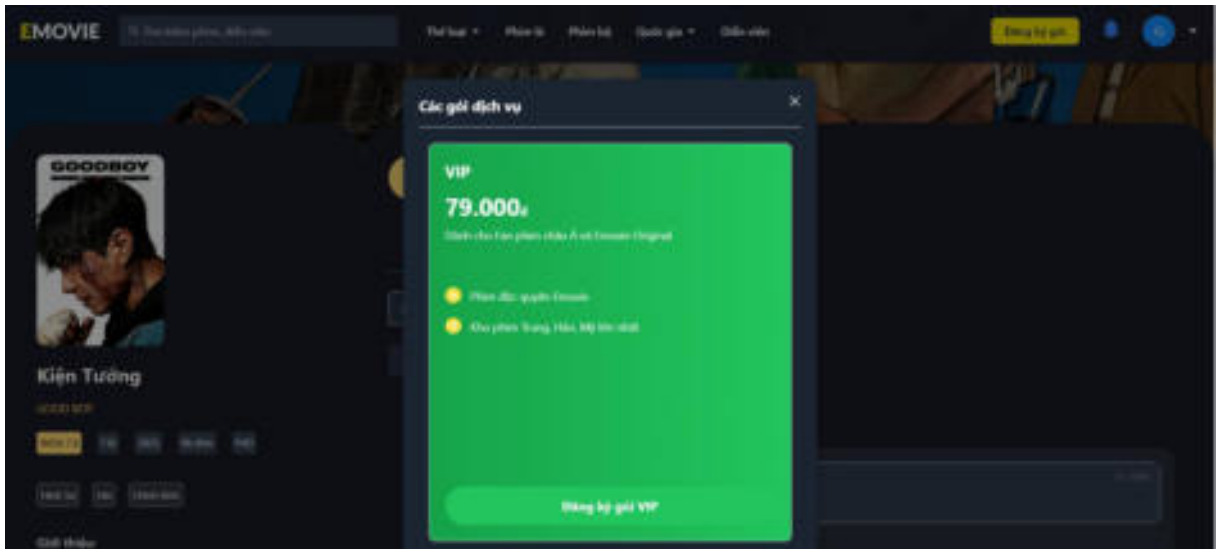
- Phim bộ



Hình 3.20 Danh sách phim bộ

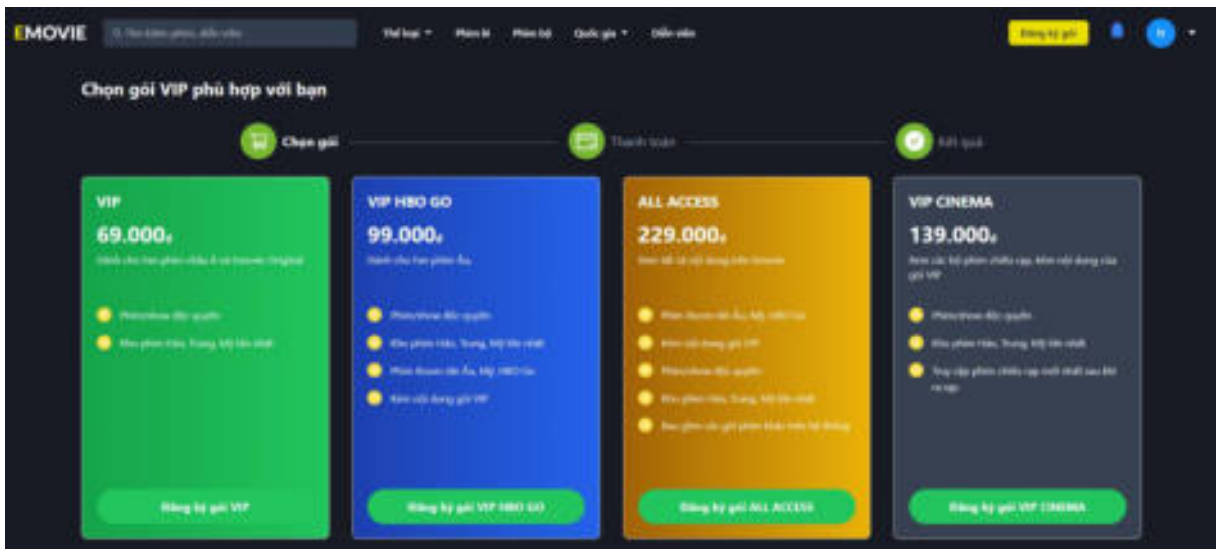


Hình 3.21 Phiên bản và các tập phim của phim bộ



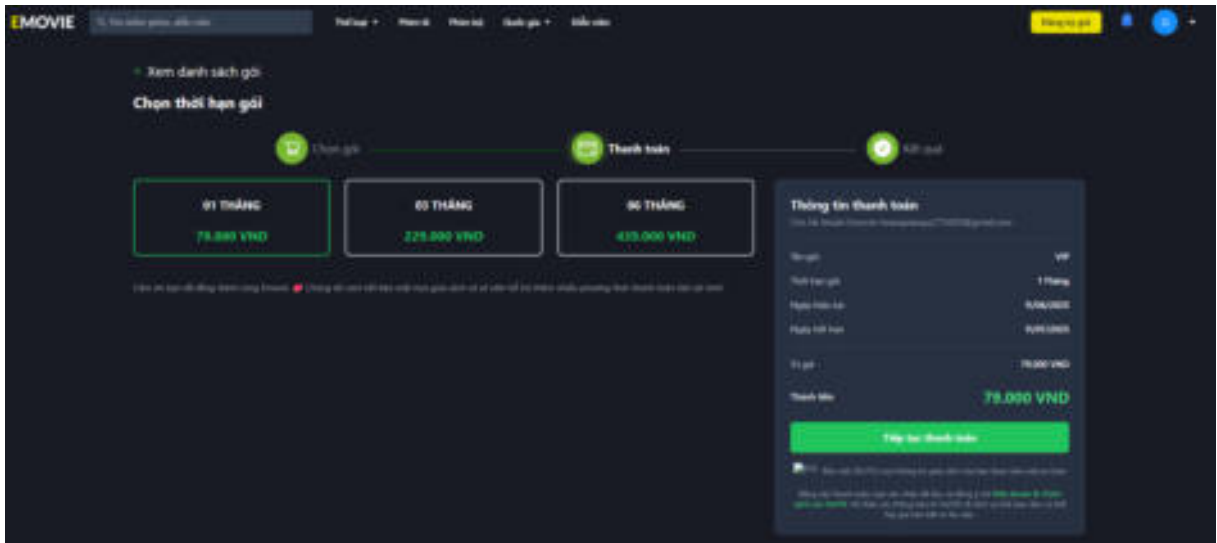
Hình 3.22 Gói dịch vụ của phim

3.4.9 Gói dịch vụ

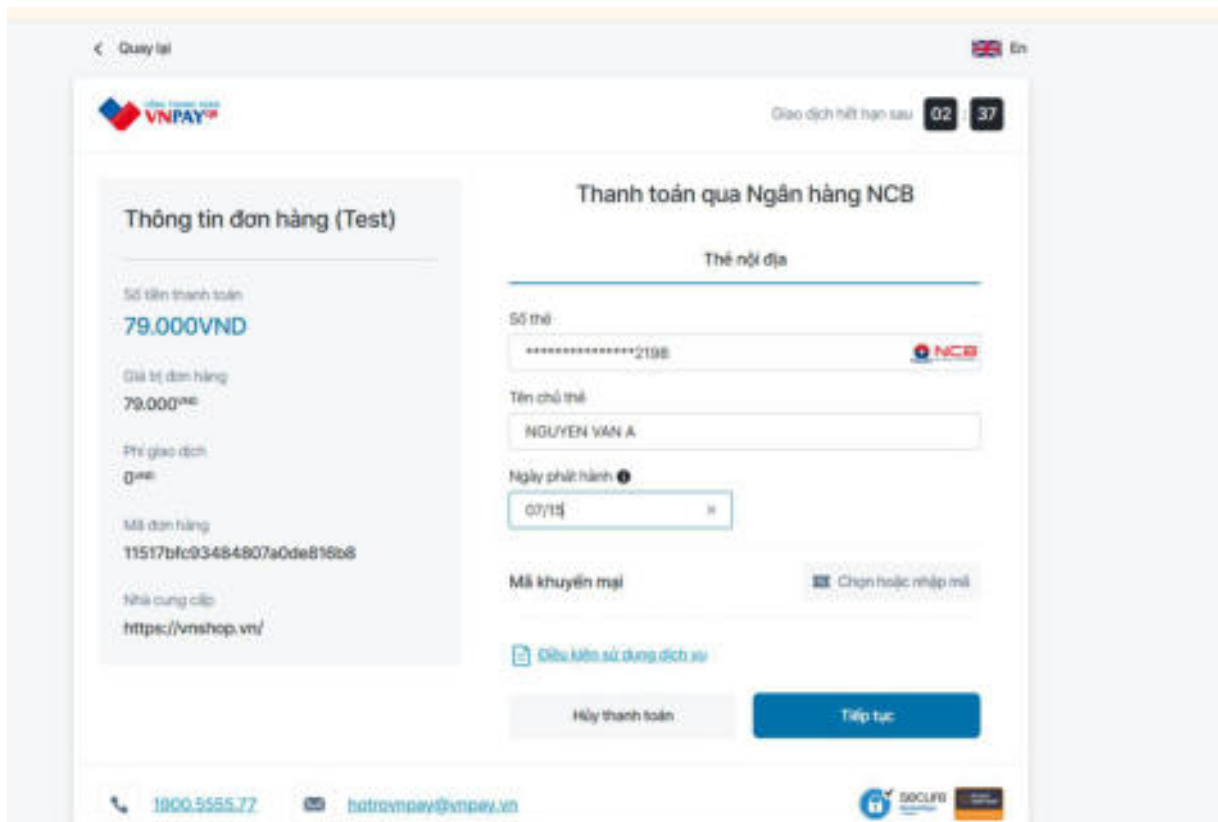


Hình 3.23 Danh sách các gói dịch vụ

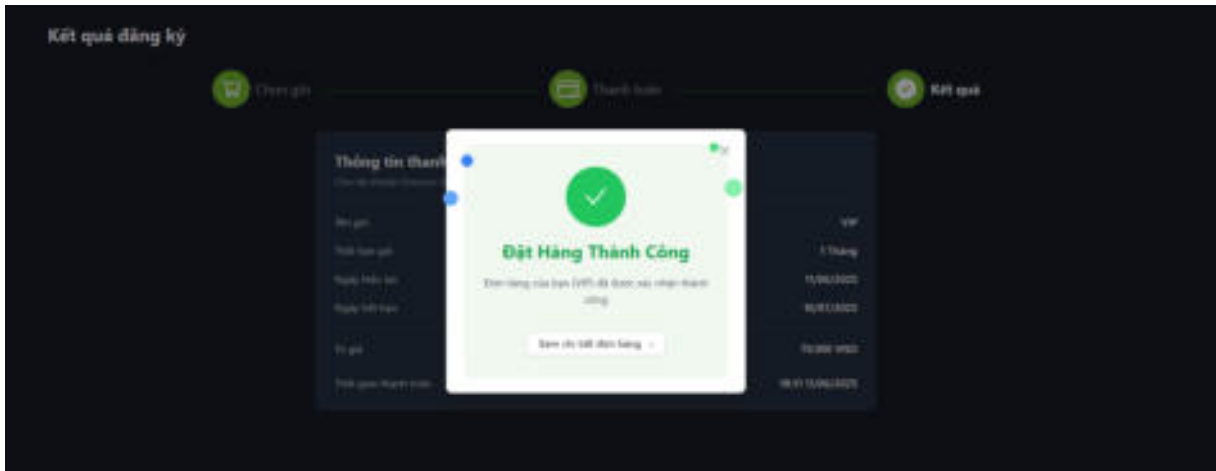
Hệ thống xem phim trực tuyến kết hợp AI gợi ý



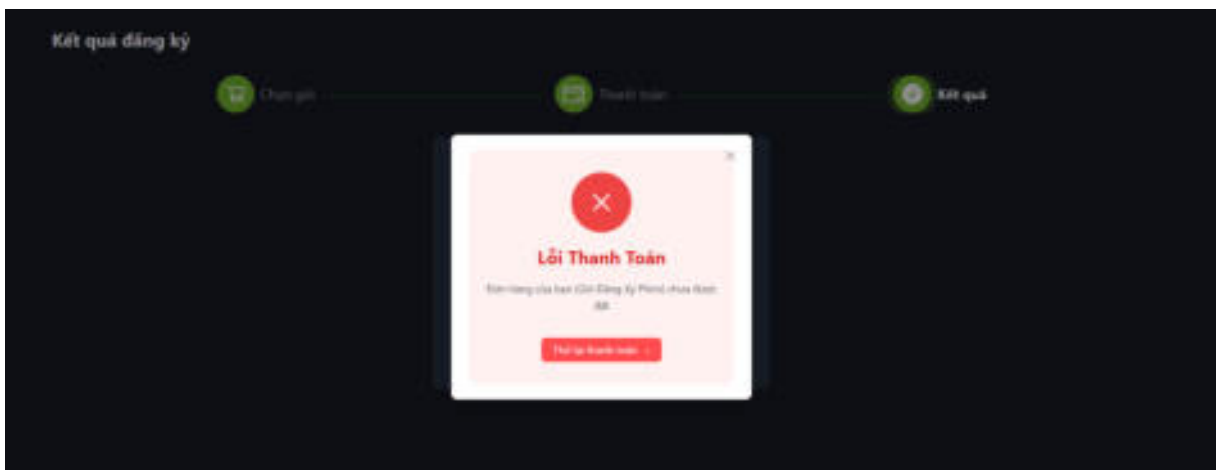
Hình 3.24 Các thời hạn gói của dịch vụ



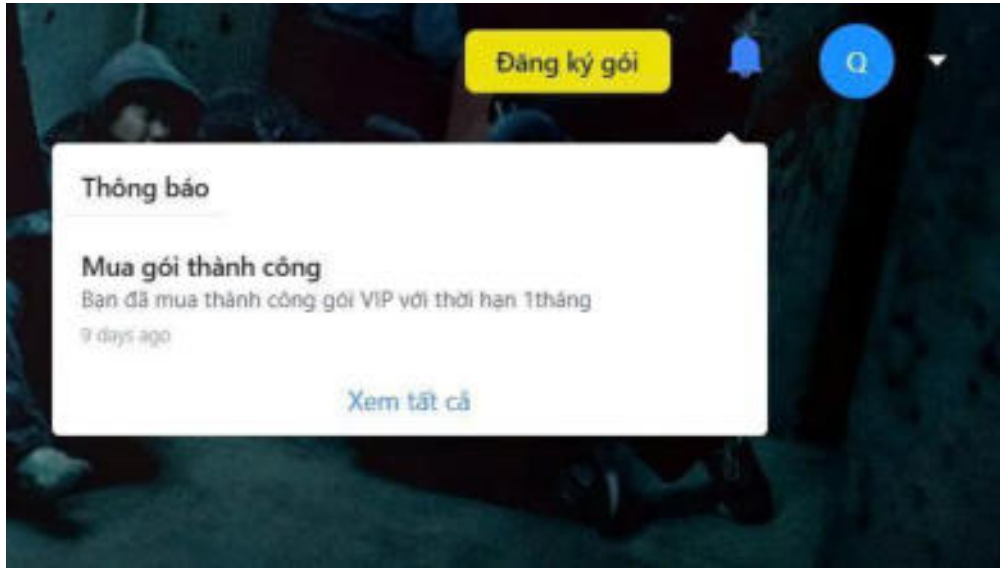
Hình 3.25 Màn hình thanh toán



Hình 3.26 Màn hình thanh toán thành công

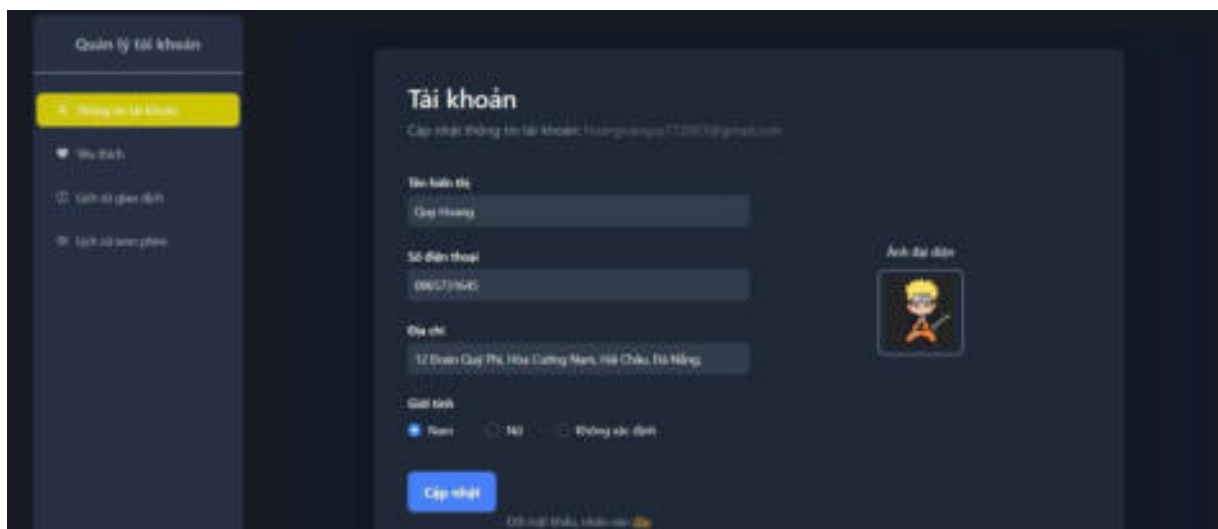


Hình 3.27 Màn hình thanh toán thất bại

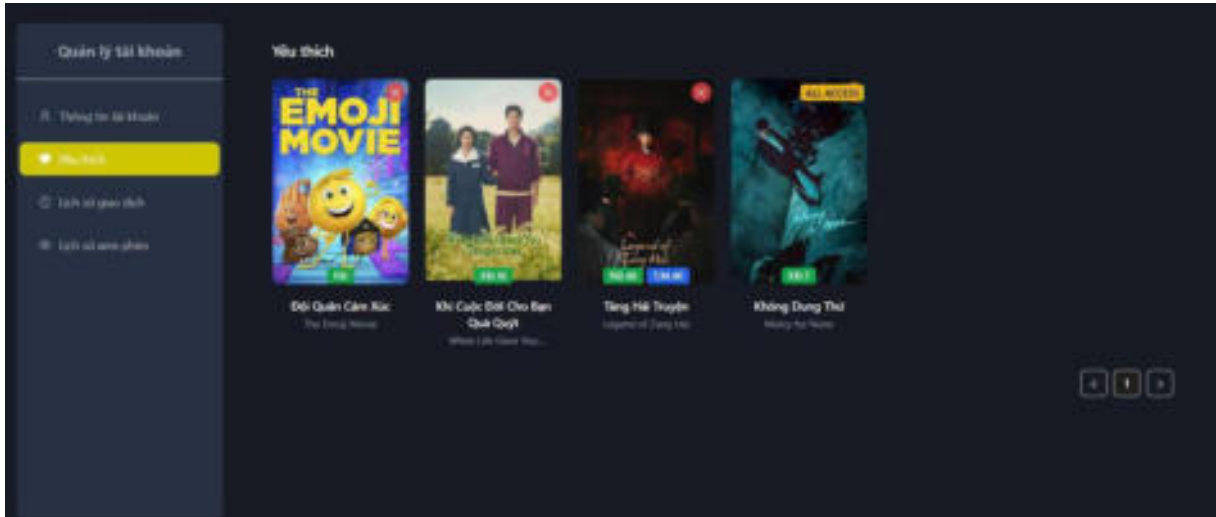


Hình 3.28 Thông báo khi mua gói thành công

3.4.10 Màn hình quản lý tài khoản



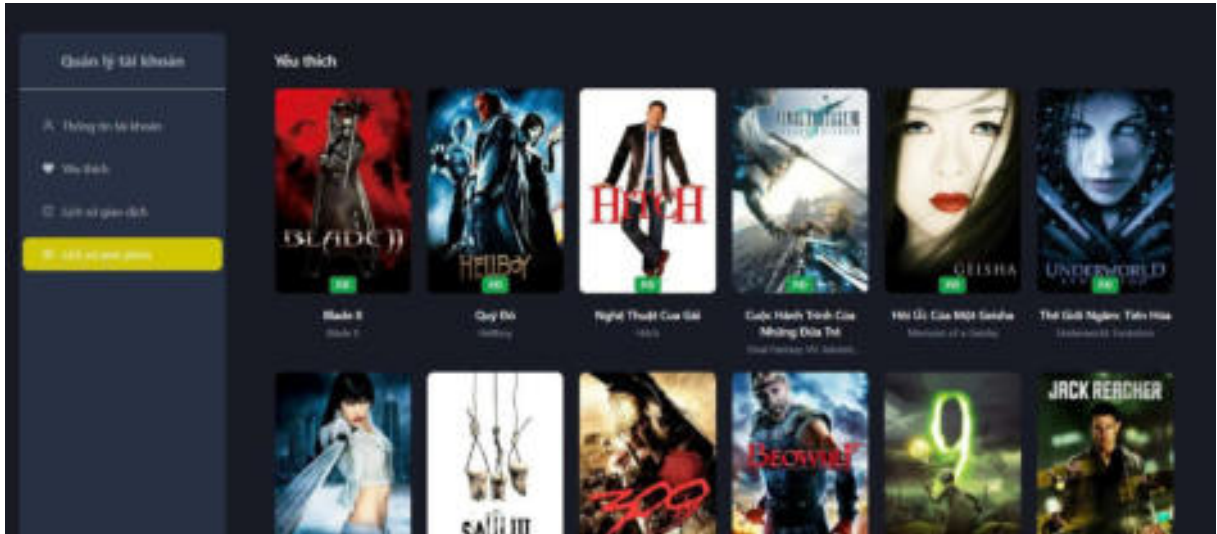
Hình 3.29 Thông tin cá nhân của người dùng



Hình 3.30 Danh sách phim yêu thích

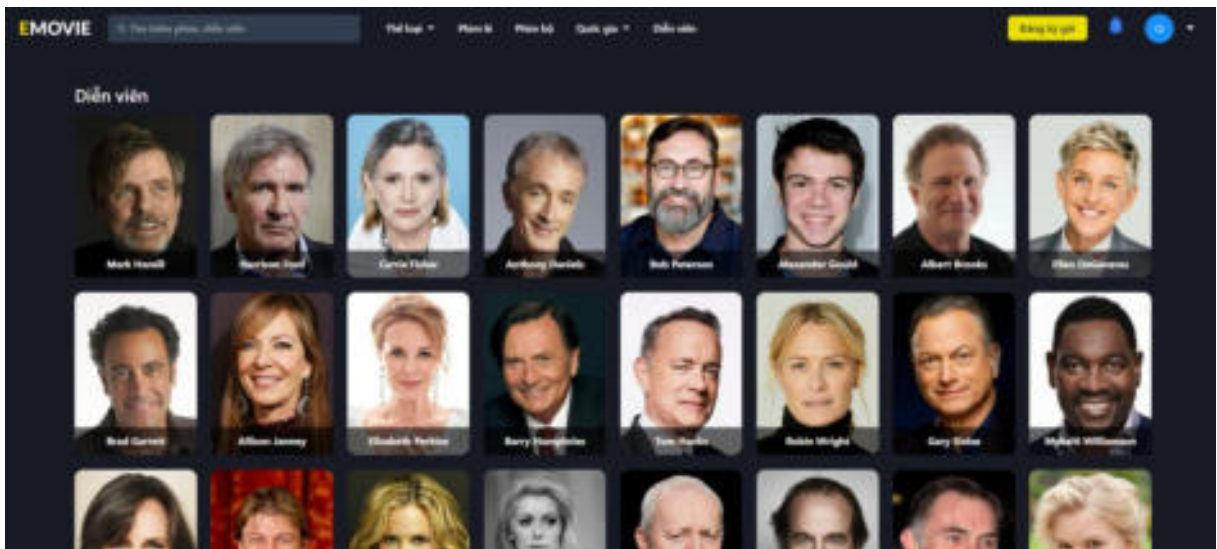


Hình 3.31 Các giao dịch của người dùng

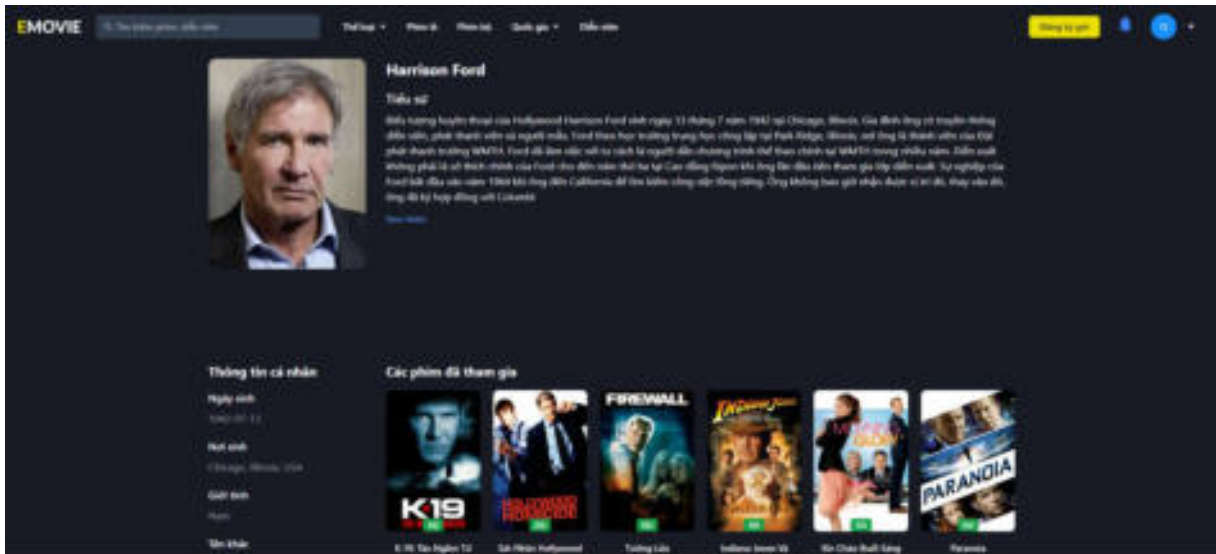


Hình 3.32 Lịch sử xem phim của người dùng

3.4.11 Các màn hình diễn viên



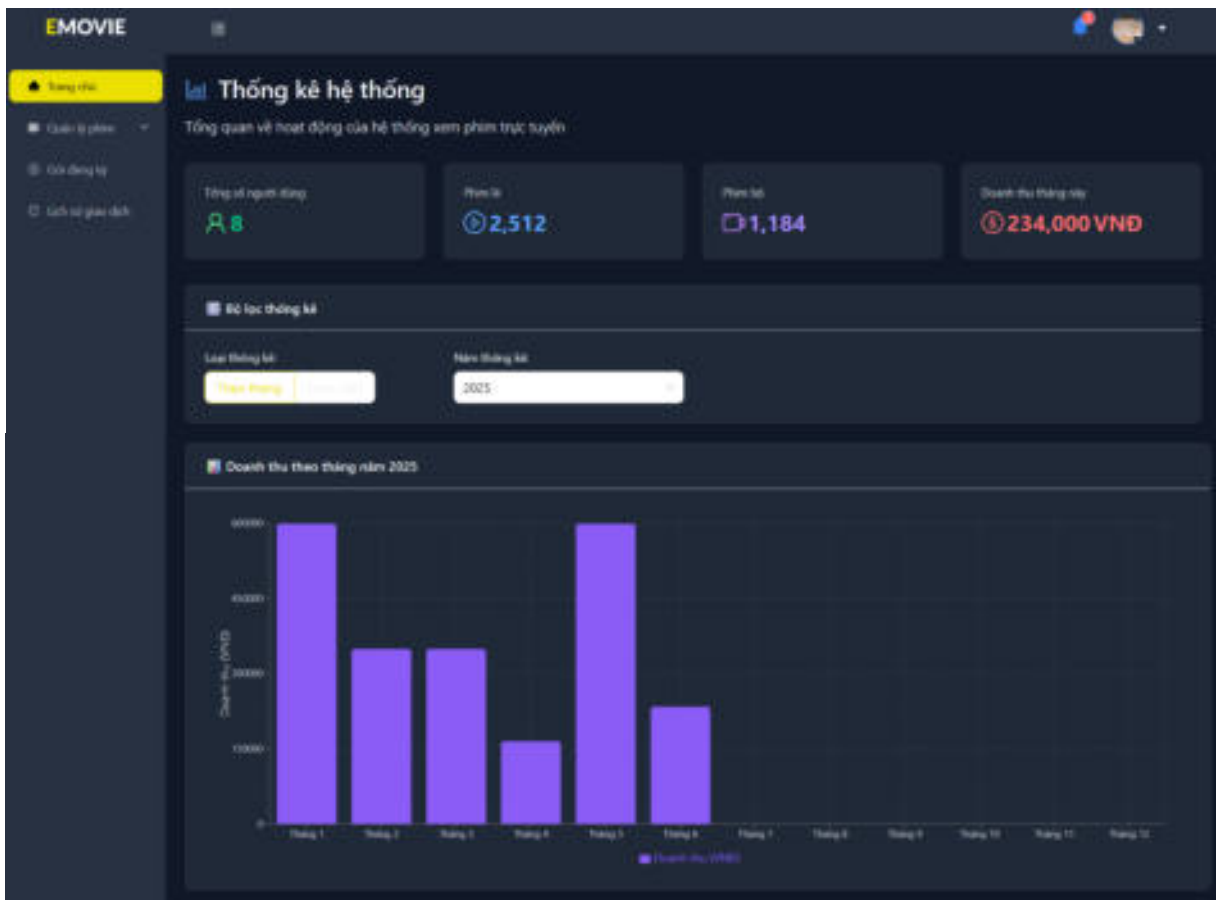
Hình 3.33 Danh sách diễn viên trong hệ thống



Hình 3.34 Thông tin chi tiết của diễn viên

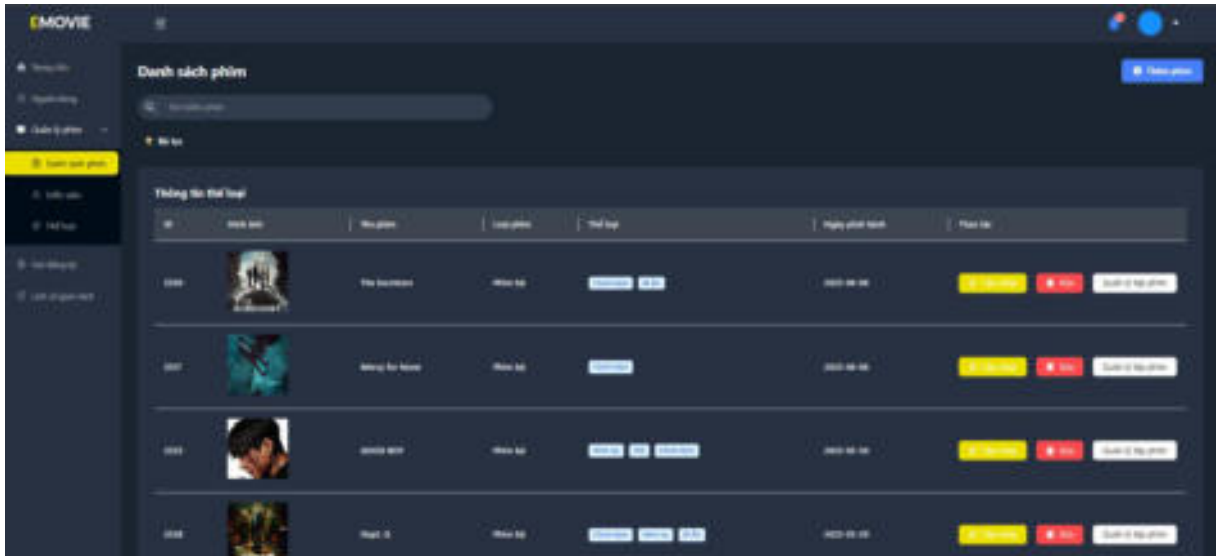
3.4.12 Các màn hình của quản trị viên

- Quản lý thống kê



Hình 3.35 Màn hình thống kê hệ thống

- Quản lý phim

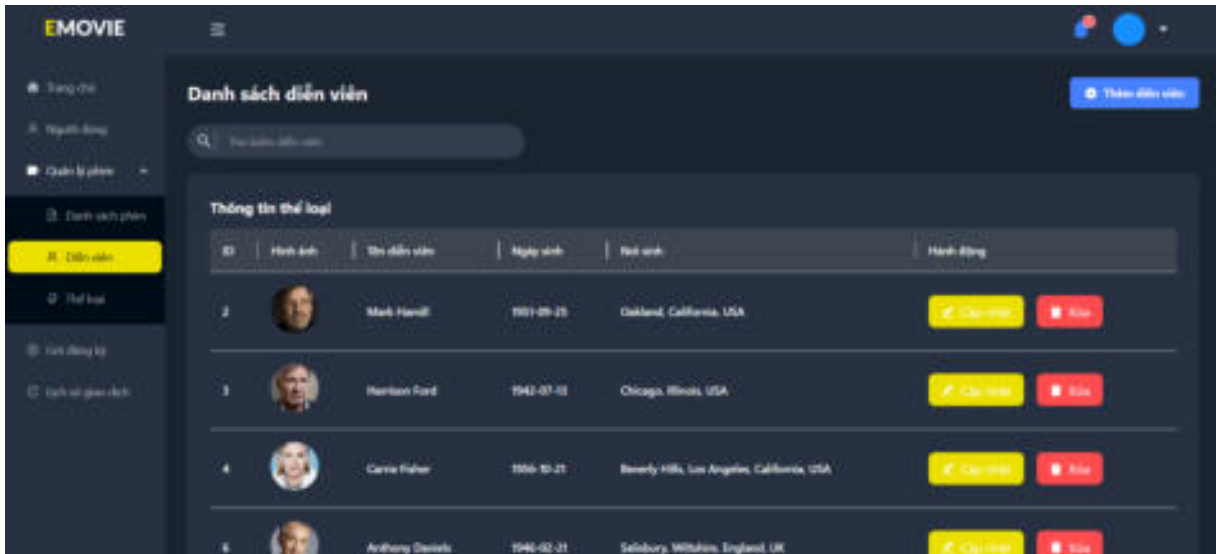


Hình 3.36 Danh sách phim

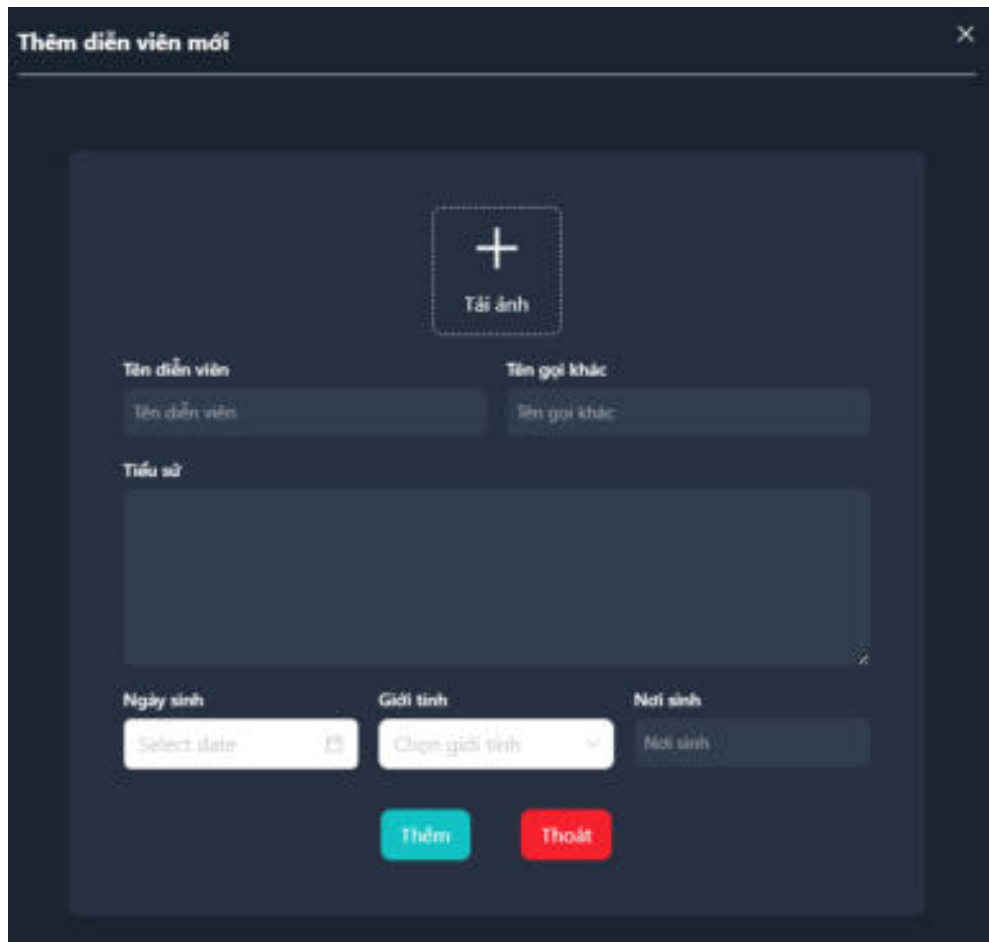
Hình 3.37 Màn hình thêm phim

Hình 3.38 Màn hình thêm diễn viên – vai diễn

- Quản lý diễn viên

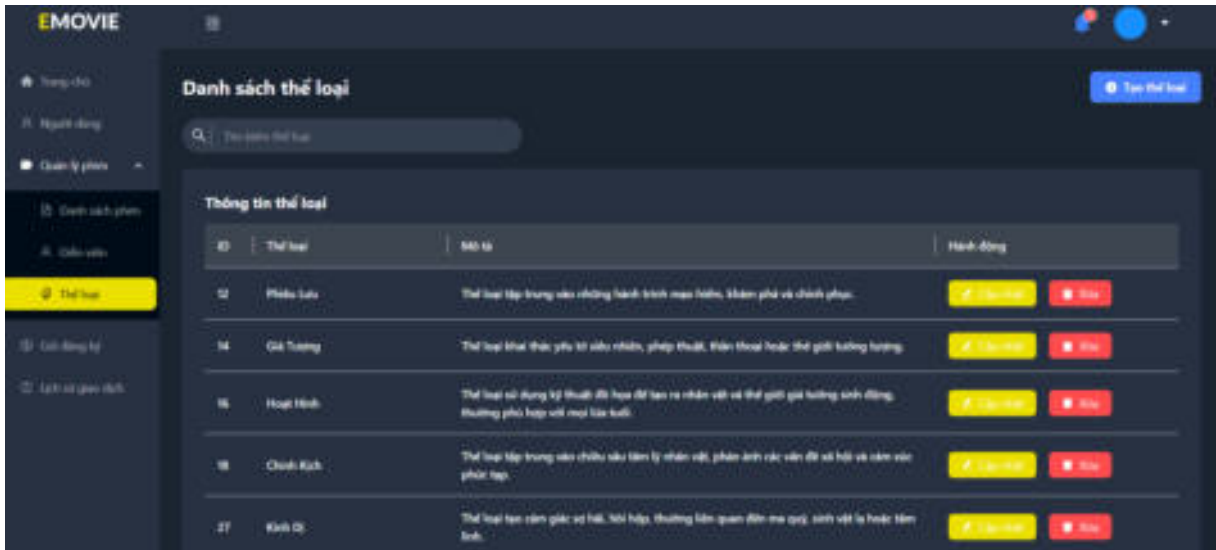


Hình 3.39 Danh sách diễn viên

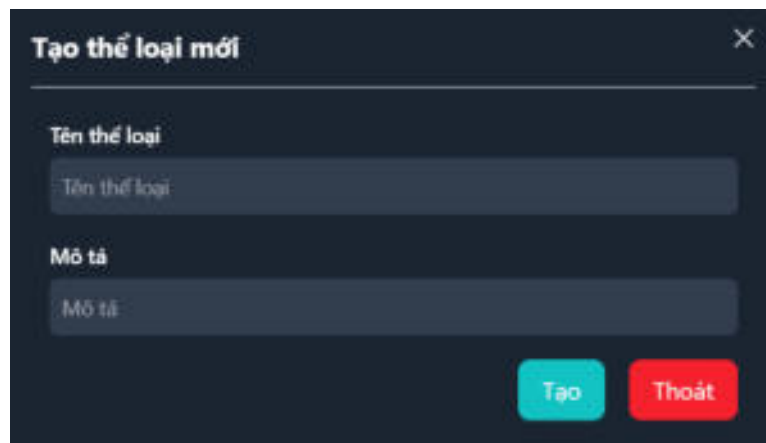


Hình 3.40 Màn hình thêm diễn viên

- Quản lý thể loại



Hình 3.41 Danh sách thể loại



Hình 3.42 Màn hình thêm thể loại

- Quản lý gói dịch vụ

| Tên gói | Mô tả | Thời hạn & Giá | Trạng thái | Thao tác |
|------------|---|--|----------------|--------------|
| VIP | Dành cho fan phim châu Á và Emovie Original | Giá 1 tháng: 19.000đ Giá 3 tháng: 23.000đ Giá 6 tháng: 43.000đ | Đang hoạt động | Thêm gói mới |
| ALL ACCESS | Xem tất cả nội dung trên Emovie | Giá 1 tháng: 19.000đ Giá 3 tháng: 49.000đ | Đang hoạt động | Thêm gói mới |
| VIP HBO GO | Dành cho fan phim Âu, Mỹ, kèm nội dung gói VIP | Giá 1 tháng: 58.000đ Giá 3 tháng: 23.000đ Giá 6 tháng: 49.000đ | Đang hoạt động | Thêm gói mới |
| VIP CINEMA | Xem các bộ phim chiếu rạp, kèm nội dung của gói VIP | Giá 1 tháng: 12.000đ Giá 3 tháng: 34.000đ | Đang hoạt động | Thêm gói mới |

Hình 3.43 Danh sách gói dịch vụ

Thêm gói dịch vụ mới

Tên gói
VIP HBO GO

Mô tả
Dành cho Fan phim Âu, Mỹ, kèm nội dung gói VIP

Gói cho
ALL ACCESS

Thời hạn & Giá Thêm thời hạn

| Tên gói thời hạn | Số tháng | Giá | |
|------------------|----------|---------|--------------------------|
| Gói 1 tháng | 1 | 109.000 | <input type="checkbox"/> |
| Gói 3 tháng | 3 | 297.000 | <input type="checkbox"/> |
| Gói 6 tháng | 6 | 490.000 | <input type="checkbox"/> |

Đặc trưng
Phim boom tấn HBO | Phim độc quyền trên Emovie

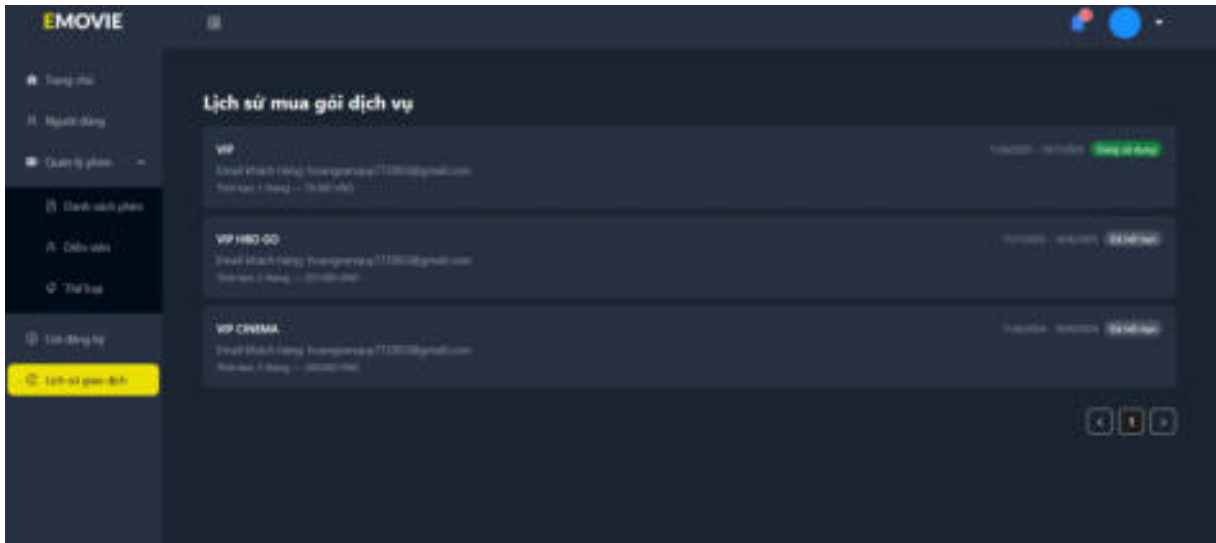
Nhập đặc trưng và nhãn Time. Có thể nhập nhiều đặc trưng một lúc, mỗi đặc trưng là một tag.

Trạng thái
Đang hoạt động

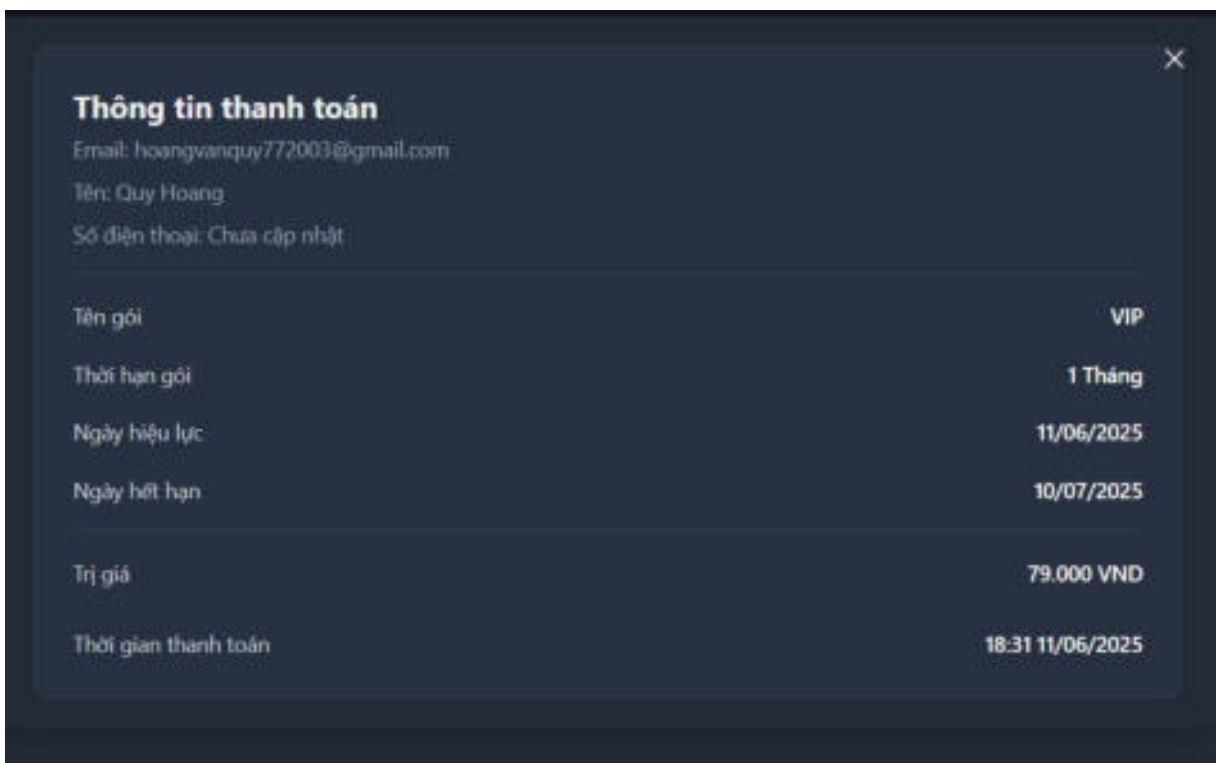
Thêm Thoát

Hình 3.44 Màn hình thêm gói dịch vụ

- Quản lý lịch sử giao dịch



Hình 3.45 Danh sách lịch sử giao dịch



Hình 3.46 Chi tiết giao dịch

3.5 Kết luận chương

Chương này đã trình bày chi tiết quá trình hiện thực hóa các yêu cầu được phân tích ở chương trước, bao gồm việc xây dựng hệ thống gợi ý phim dựa trên đặc điểm của phim bằng phương pháp Content-Based Filtering, triển khai mô hình học sâu NeuMF thông qua thư viện Recbole, và sự kết hợp giữa hai phương pháp nhằm nâng cao hiệu quả gợi ý. Ngoài ra, chương cũng mô tả các công nghệ được sử dụng trong quá trình xây dựng và triển khai hệ thống, đồng thời trình bày kết quả hoàn thiện của dự án

KẾT LUẬN

1. Kết quả đạt được

Sau quá trình nghiên cứu và triển khai, đề án đã xây dựng thành công một hệ thống xem phim trực tuyến tích hợp chức năng gợi ý phim. Hệ thống đã hoàn thiện các chức năng cơ bản cần thiết, đồng thời chú trọng đến tính bảo mật và khả năng mở rộng trong tương lai.

Các nhóm chức năng chính đã được triển khai trong hệ thống bao gồm:

- Quản lý nội dung và nghiệp vụ: Bao gồm các tính năng quản lý thông tin về phim, thể loại, diễn viên, quản lý các gói dịch vụ, cũng như theo dõi doanh thu và thống kê cơ bản, hỗ trợ việc vận hành hệ thống.
- Hỗ trợ thanh toán trực tuyến: Cho phép người dùng mua các gói dịch vụ thông qua tích hợp cổng thanh toán VNPAY, đảm bảo quy trình thanh toán thuận tiện và an toàn.
- Chức năng gợi ý phim: Hệ thống đã triển khai hai phương pháp gợi ý chính để hỗ trợ người dùng khám phá phim:
- Gợi ý dựa trên nội dung (Content-based filtering - CBF): Đề xuất các bộ phim dựa trên sự tương đồng về đặc điểm nội dung (ví dụ: thể loại, diễn viên, đạo diễn, ...) với những phim mà người dùng đang quan tâm hoặc đã xem.
- Mô hình học sâu NeuMF (Neural Matrix Factorization): Ứng dụng mô hình này để phân tích và dự đoán mối quan hệ giữa người dùng và phim, từ đó đưa ra các gợi ý có tính cá nhân hóa cao hơn.

Để hỗ trợ hệ thống hoạt động hiệu quả, một số công nghệ quan trọng đã được tích hợp:

- Redis: Được sử dụng như bộ nhớ tạm cho các dữ liệu nhạy cảm như mã OTP, token đăng nhập, token đặt lại mật khẩu, giúp tăng tốc xử lý và đảm bảo bảo mật.
- Azure Storage Blob: Áp dụng để lưu trữ các tài nguyên đa phương tiện (phim, hình ảnh...) một cách hiệu quả, đảm bảo tính sẵn sàng và khả năng mở rộng cho nội dung số.

- Tự động hóa triển khai: Hệ thống được đóng gói và triển khai thông qua quy trình CI/CD sử dụng Docker, Docker Compose, Docker Hub, GitHub Actions và Azure Virtual Machine, giúp tự động hóa toàn bộ quá trình build và deploy lên môi trường thực tế.

2. Những vấn đề chưa giải quyết

Dù hệ thống đã đạt được các mục tiêu đề ra, vẫn có những điểm có thể cải tiến trong tương lai:

- Hệ thống gợi ý: Hiện tại cần huấn luyện định kỳ thủ công, chưa hỗ trợ cập nhật dữ liệu mới theo thời gian thực, chưa tự động hóa quá trình huấn luyện các mô hình.
- Quy trình tải lên video: Việc tải tệp phim còn khá đơn giản, chưa tích hợp các công cụ hỗ trợ xử lý và tối ưu hóa video một cách chuyên sâu.
- Giao diện chưa hoàn toàn responsive: Một số thành phần giao diện chưa hiển thị tối ưu trên các thiết bị có kích thước màn hình khác nhau.
- Chưa tận dụng được tính năng caching từ redis

3. Hướng phát triển

- Cải tiến hệ thống gợi ý: Nghiên cứu và tích hợp các mô hình học sâu như GRU4Rec, SASRec để khai thác hành vi tuần tự của người dùng; đồng thời ứng dụng mô hình ngôn ngữ BERT nhằm nâng cao chất lượng gợi ý theo nội dung thông qua việc hiểu sâu hơn mô tả phim và sở thích người dùng.
- Tự động hóa pipeline huấn luyện: Xây dựng quy trình tự động train lại mô hình gợi ý theo thời gian định kỳ hoặc khi có dữ liệu mới.
- Phát triển ứng dụng di động: Xây dựng phiên bản mobile nhằm mở rộng khả năng tiếp cận người dùng và nâng cao trải nghiệm sử dụng.
- Tích hợp AI đàm thoại: Áp dụng các mô hình ngôn ngữ lớn (LLM) kết hợp kỹ thuật RAG (Retrieval-Augmented Generation) để người dùng có thể tìm kiếm phim hoặc tương tác với hệ thống bằng ngôn ngữ tự nhiên.
- Tối ưu hóa tải video: Áp dụng các công cụ chuyên sâu trong việc xử lý, mã hóa và tối ưu hóa dung lượng video khi tải lên hệ thống.

- Sử dụng caching của redis: áp dụng cho việc tìm kiếm phim theo bộ lọc, các phim được đề xuất từ hệ thống nhằm tăng hiệu suất, tăng trải nghiệm người dùng.

TÀI LIỆU THAM KHẢO

- [1] "Recommendation System: nvidia," [Online]. Available: <https://www.nvidia.com/en-us/glossary/recommendation-system/>. [Accessed 6 2025].
- [2] "NeuMF: Recbole," [Online]. Available: https://recbole.io/docs/user_guide/model/general/neumf.html. [Accessed 6 2025].
- [3] R. Banik, "Movie Recommender Systems," [Online]. Available: <https://www.kaggle.com/code/rounakbanik/movie-recommender-systems#Hybrid-Recommender>. [Accessed 6 2025].
- [4] "RESTful API là gì? Cách thiết kế RESTful API," [Online]. Available: <https://topdev.vn/blog/restful-api-la-gi/>. [Accessed 6 2025].
- [5] "Quick Start," [Online]. Available: <https://react.dev/learn>. [Accessed 4 2025].
- [6] "Getting Started," [Online]. Available: <https://redis.io/learn/howtos/quick-start>. [Accessed 4 2025].
- [7] "Guides," 4 2025. [Online]. Available: <https://spring.io/guides>.
- [8] "Components Overview," 4 2025. [Online]. Available: <https://ant.design/components/overview/>.
- [9] "RTK Query Overview," [Online]. Available: <https://redux-toolkit.js.org/rtk-query/overview>. [Accessed 4 2025].
- [10] "Quickstart: Upload, download, and list blobs with the Azure portal," [Online]. Available: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal>. [Accessed 4 2025].
- [11] "Docker guides," [Online]. Available: <https://docs.docker.com/guides/>. [Accessed 6 2025].
- [12] "Docker Hub quickstart," [Online]. Available: <https://docs.docker.com/docker-hub/quickstart/>. [Accessed 6 2025].

- [13] "Docker Compose Quickstart," [Online]. Available: <https://docs.docker.com/compose/gettingstarted/>. [Accessed 6 2025].
- [14] "Quickstart for GitHub Actions," [Online]. Available: <https://docs.github.com/en/actions/writing-workflows/quickstart>. [Accessed 6 2025].
- [15] "How to build a CI/CD pipeline with GitHub Actions in four simple steps," [Online]. Available: <https://github.blog/enterprise-software/ci-cd/build-ci-cd-pipeline-github-actions-four-steps/>. [Accessed 6 2025].
- [16] "Quickstart: Create a Windows virtual machine in the Azure portal," [Online]. Available: <https://learn.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal>. [Accessed 6 2025].
- [17] "Understanding TF-IDF (Term Frequency-Inverse Document Frequency)," [Online]. Available: <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>. [Accessed 5 2025].
- [18] "Cosine Similarity," [Online]. Available: <https://www.geeksforgeeks.org/dbms/cosine-similarity/>. [Accessed 5 2025].
- [19] "WebSockets," [Online]. Available: <https://docs.spring.io/spring-framework/reference/web/websocket.html>. [Accessed 5 2025].