**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**---------------------------------------**



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG ỨNG DỤNG WEB MWATCH ĐỂ XEM PHIM TRỰC TUYẾN SỬ DỤNG CÔNG NGHỆ ASP.NET MVC**

**GVHD: TS. Hà Mạnh Đào**

**Sinh viên: Hoàng Tiến Mạnh– K15**

**Mã số sinh viên: 2020608054**

Hà Nội – Năm 2024

**LỜI CẢM ƠN**

Đầu tiên, xin bày tỏ lòng biết ơn sâu sắc đến Trường Đại học Công Nghiệp Hà Nội và tất cả các thầy cô đã dành thời gian và kiến thức để giảng dạy trong suốt những năm học qua. Những kiến thức và kỹ năng mà các thầy cô truyền đạt không chỉ là cơ sở cho dự án này mà còn là nền tảng quan trọng cho sự nghiệp sau này. Đặc biệt, xin gửi lời cảm ơn sâu sắc đến thầy Hà Mạnh Đào, người đã hướng dẫn và chia sẻ những kiến thức quý báu trong suốt quá trình hoàn thành dự án.

Quá trình nghiên cứu đã được thực hiện một cách cẩn thận và chi tiết nhằm hoàn thành bài nghiên cứu này. Tuy nhiên, do hạn chế về thời gian và kinh nghiệm trong nghiên cứu khoa học, dự án vẫn còn một số thiếu sót. Rất mong nhận được sự đóng góp ý kiến từ quý thầy cô để hoàn thiện hơn.

Chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC CÁC BẢNG THUẬT NGỮ, KÍ HIỆU, VIẾT TẮT 5](#_Toc185877445)

[DANH MỤC HÌNH ẢNH 6](#_Toc185877446)

[DANH MỤC BẢNG BIỂU 8](#_Toc185877447)

[MỞ ĐẦU 9](#_Toc185877448)

[CHƯƠNG 1. TỔNG QUAN VỀ ASP.NET FRAMEWORK VÀ MÔ HÌNH MVC 12](#_Toc185877449)

[1.1. Công nghệ ASP.NET Framework. 12](#_Toc185877450)

[1.1.1. Framework ASP.NET 12](#_Toc185877451)

[1.1.2. Các tính năng của ASP.NET Framework. 14](#_Toc185877452)

[1.1.3. Lí do nên sử dụng ASP.NET Framework 15](#_Toc185877453)

[1.2. Mô hình MVC. 16](#_Toc185877454)

[1.2.1. Mô hình MVC 16](#_Toc185877455)

[1.2.2. Luồng đi trong mô hình MVC. 17](#_Toc185877456)

[1.2.3. Ưu nhược điểm của mô hình MVC. 18](#_Toc185877457)

[1.2.4. Quy trình để phát triển mô hình Web MVC 19](#_Toc185877458)

[1.3. Bootstrap Framework 21](#_Toc185877459)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 23](#_Toc185877460)

[2.1. Giới thiệu ứng dụng Web xem phim trực tuyến MWatch 23](#_Toc185877461)

[2.1.1. Mô tả ứng dụng web Mwatch 23](#_Toc185877462)

[2.1.2 Yêu cầu hệ thống 23](#_Toc185877463)

[2.1.3 Thiết kế giao diện Ứng dụng Web MWatch 24](#_Toc185877464)

[2.2 Mô hình hóa chức năng 26](#_Toc185877465)

[2.2.1 Biểu đồ use case 26](#_Toc185877466)

[2.2.2 Biểu đồ use case phía người dùng 27](#_Toc185877467)

[2.2.3 Biểu đồ use cae phía quản trị viên 27](#_Toc185877468)

[2.2.4 Phân rã biểu đồ use case quản lý phim 28](#_Toc185877469)

[2.2.5 Phân rã biểu đồ use case quản lý tài khoản 28](#_Toc185877470)

[2.3 Đặc tả các use case 29](#_Toc185877471)

[2.3.1 Use case xem danh sách phim 29](#_Toc185877472)

[2.3.2 Use case xem chi tiết phim 30](#_Toc185877473)

[2.3.3 Use case xem phim 30](#_Toc185877474)

[2.3.4 Use case đăng kí tài khoản 31](#_Toc185877475)

[2.3.5 Use case đăng nhập 32](#_Toc185877476)

[2.3.6 Use case tìm kiếm phim 33](#_Toc185877477)

[2.3.7 Use case bình luận phim 34](#_Toc185877478)

[2.3.8 Use case quản lý phim 35](#_Toc185877479)

[2.3.9 Use case quản lý tài khoản 37](#_Toc185877480)

[2.3.10 Use case cập nhật thông tin phim 38](#_Toc185877481)

[2.3.11 Use case quản lý bình luận. 39](#_Toc185877482)

[2.3.12 Use case Thanh toán dịch vụ. 40](#_Toc185877483)

[2.3.13 Use case quản lý phim yêu thích. 41](#_Toc185877484)

[2.4 Biểu đồ trình tự của các use case 43](#_Toc185877485)

[2.4.1 Biểu đồ trình tự cho quá trình đăng nhập 43](#_Toc185877486)

[2.4.2 Biểu đồ trình tự cho quá trình tìm kiếm 43](#_Toc185877487)

[2.4.3 Biểu đồ trình tự cho quá trình thêm phim 44](#_Toc185877488)

[2.4.4 Biểu đồ trình tự cho quá trình sửa phim 44](#_Toc185877489)

[2.4.5 Biểu đồ trình tự cho quá trình xoá phim 45](#_Toc185877490)

[2.4.6 Biểu đồ trình tự cho chức năng thêm user 45](#_Toc185877491)

[2.4.7 Biểu đồ trình tự cho chức năng sửa user 46](#_Toc185877492)

[2.4.8 Biểu đồ trình tự chức năng xoá user 46](#_Toc185877493)

[2.4.9 Biểu đồ trình tự chức năng thêm bình luận 47](#_Toc185877494)

[2.5 Các bảng dữ liệu 47](#_Toc185877495)

[2.5.1 Các bảng dữ liệu 47](#_Toc185877496)

[2.5.2 Biểu đồ cơ sở dữ liệu 51](#_Toc185877497)

[CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ 52](#_Toc185877498)

[3.1 Mô hình thử nghiệm 52](#_Toc185877502)

[3.2 Cài đặt hệ thống 52](#_Toc185877503)

[3.2.1 Cấu hình máy chủ 52](#_Toc185877504)

[3.2.1 Cấu hình máy client 53](#_Toc185877505)

[3.3 Báo cáo kiểm thử 54](#_Toc185877506)

[3.4 Kịch bản kiểm thử 57](#_Toc185877507)

[3.4.1 Đăng ký tài khoản (KT07) 57](#_Toc185877508)

[3.4.2 Xem chi tiết phim (KT03) 58](#_Toc185877509)

[3.4.3 Tìm kiếm phim (KT05) 59](#_Toc185877510)

[3.5 Đánh giá kiểm thử 61](#_Toc185877511)

[KẾT LUẬN 62](#_Toc185877512)

[TÀI LIỆU THAM KHẢO 63](#_Toc185877513)

# DANH MỤC CÁC BẢNG THUẬT NGỮ, KÍ HIỆU, VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **TIẾNG VIỆT** | **TIẾNG ANH** |
| **1** | CSDL | Cơ sở dữ liệu | Database |
| **2** | HTML | Ngôn ngữ đánh dấu siêu văn bản | HyperText Markup Language |
| **3** | CSS | Tập tin định kiểu theo tầng | Cascading Style Sheet |
| **4** | IE |  | Internet Explorer |
| **5** | HTTP | Giao thức truyền tải siêu văn bản | HyperText Transfer Protocol |
| **6** | XML | Ngôn ngữ đánh dấu mở rộng | Extensible Markup Language |
| **7** | API | Giao diện lập trình ứng dụng | Application Programming Interface |
| **8** | MVC |  | Model – View – Controller |

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 Kiến trúc Entity Framework 12](#_Toc185876526)

[Hình 1. 2 Các thành phần bên trong của mô hình MVC 15](#_Toc185876527)

[Hình 1. 3 Luồng xử lý trong MVC 16](#_Toc185876528)

[Hình 2. 1 Giao diện trang chủ……………………………………………….23](#_Toc185876529)

[Hình 2. 2 Giao diện chi tiết phim 23](#_Toc185876530)

[Hình 2. 3 Giao diện đăng ký người dùng 24](#_Toc185876531)

[Hình 2. 4 Biểu đồ Use case 24](#_Toc185876532)

[Hình 2. 5 Biểu đồ Use case người dùng 25](#_Toc185876533)

[Hình 2. 6 Biểu đồ Use case phía quản trị viên 25](#_Toc185876534)

[Hình 2. 7 Phân rã biểu đồ Use case quản lý phim 26](#_Toc185876535)

[Hình 2. 8 Phân rã biểu đồ Use case quản lý tài khoản 26](#_Toc185876536)

[Hình 2. 9 Biểu đồ trình tự use case đăng nhập 41](#_Toc185876537)

[Hình 2. 10 Biểu đồ trình tự use case tìm kiếm 41](#_Toc185876538)

[Hình 2. 11 Biểu đồ trình tự cho quá trình thêm phim 42](#_Toc185876539)

[Hình 2. 12 Biểu đồ trình tự cho quá trình sửa phim 42](#_Toc185876540)

[Hình 2. 13 Biểu đồ trình tự cho quá trình xoá phim 43](#_Toc185876541)

[Hình 2. 14 Biểu đồ trình tự cho chức năng thêm user 43](#_Toc185876542)

[Hình 2. 15 Biểu đồ trình tự cho chức năng sửa user 44](#_Toc185876543)

[Hình 2. 16 Biểu đồ trình tự chức năng xoá user 44](#_Toc185876544)

[Hình 2. 17 Biểu đồ trình tự chức năng thêm bình luận 45](#_Toc185876545)

[Hình 2. 18 Biểu đồ cơ sở dữ liệu 49](#_Toc185876546)

[Hình 3. 1 Mô hình client-server ……………………………………………..50](#_Toc185876547)

[Hình 3. 2 Màn hình kiểm thử nhập thông tin đăng ký tài khoản 55](#_Toc185876548)

[Hình 3. 3 Màn hình kiểm thử đăng ký thành công 56](#_Toc185876549)

[Hình 3. 4 Màn hình kiểm thử đăng ký tài khoản thất bại 56](#_Toc185876550)

[Hình 3. 5 Màn hình kiểm thử xem chi tiết phim 57](#_Toc185876551)

[Hình 3. 6 Màn hình kiểm thử xem chi tiết phim thành công 57](#_Toc185876552)

[Hình 3. 7 Màn hình kiểm thử tìm kiếm phim 58](#_Toc185876553)

[Hình 3. 8 Màn hình kiểm thử tìm kiếm phim thành công 58](#_Toc185876554)

[Hình 3. 9 Màn hình kiểm thử tìm kiếm phim thất bại 58](#_Toc185876555)

# DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Bảng user 46](#_Toc178548127)

[Bảng 2. 2 Bảng movie 47](#_Toc178548128)

[Bảng 2. 3 Bảng favoriteMovies 47](#_Toc178548129)

[Bảng 2. 4 Bảng comments 48](#_Toc178548130)

[Bảng 2. 5 Bảng MovieGenres 48](#_Toc178548131)

[Bảng 2.6 Bảng Genres 48](#_Toc178548132)

[Bảng 2.7 Bảng payment 49](#_Toc178548133)

[Bảng 2.8 Bảng watchHistory 49](#_Toc178548134)

# MỞ ĐẦU

**1. Lý do chọn đề tài**

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu giải trí trực tuyến, đặc biệt là xem phim, ngày càng tăng cao. Người dùng hiện nay mong muốn có những trải nghiệm xem phim dễ dàng, nhanh chóng, với chất lượng hình ảnh và âm thanh cao. Tuy nhiên, không phải ai cũng có thể tiếp cận được các nền tảng xem phim chất lượng, hợp pháp và phù hợp với nhu cầu cá nhân. Do đó, việc xây dựng một website xem phim trực tuyến nhằm cung cấp cho người dùng một nền tảng tiện ích, tập hợp các bộ phim đa dạng, được cập nhật liên tục là điều cần thiết. Trang web không chỉ mang đến trải nghiệm giải trí mà còn giúp kết nối cộng đồng người yêu phim, đồng thời mở ra cơ hội phát triển kinh doanh từ các dịch vụ kèm theo như quảng cáo, đăng ký thành viên. Qua dự án này, mục tiêu là tạo ra một sản phẩm công nghệ hữu ích, có tính ứng dụng cao và đáp ứng nhu cầu của thị trường giải trí trực tuyến hiện nay.

**2. Mục tiêu của đề tài**

Xây dựng một sản phẩm hoàn chỉnh, từ khâu phân tích, thiết kế đến phát triển ứng dụng webs xem phim trực tuyến. Ứng dụng sẽ đáp ứng các yêu cầu cơ bản của một trang web giải trí trực tuyến, bao gồm quản lý nội dung phim, hỗ trợ người dùng tìm kiếm và xem phim với chất lượng cao, tích hợp các tính năng như đăng ký thành viên, đánh giá phim, và quản lý nội dung. Đồng thời, xây dựng báo cáo đồ án đúng quy định, thể hiện rõ quy trình và kết quả đạt được trong việc triển khai dự án.

**3. Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là mô hình phát triển website xem phim trực tuyến sử dụng framework ASP.NET MVC, phương pháp thiết kế giao diện người dùng thân thiện và hiệu quả. Ngoài ra, đề tài còn nghiên cứu công cụ quản lý nội dung phim và thuật toán xử lý, truyền tải video. Những đối tượng này được vận dụng để xây dựng một ứng dụng thực tế trong lĩnh vực giải trí trực tuyến.

**4. Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào xây dựng một website xem phim trực tuyến sử dụng mô hình MVC trên nền tảng ASP.NET, với các nội dung chính như sau: nghiên cứu và triển khai mô hình MVC, phân tách rõ ràng giữa Model (quản lý dữ liệu với Entity Framework), View (xây dựng giao diện bằng Razor Pages) và Controller (xử lý logic nghiệp vụ). Thiết kế cơ sở dữ liệu quan hệ trên SQL Server bao gồm các bảng phim, người dùng, đánh giá, và lịch sử xem, đồng thời tích hợp Entity Framework để quản lý dữ liệu hiệu quả. Phát triển các chức năng cơ bản như đăng ký, đăng nhập bằng ASP.NET Identity, xem phim trực tuyến, tìm kiếm, lọc phim, danh sách yêu thích, lịch sử xem và giao diện quản trị để quản lý nội dung và tài khoản người dùng. Giao diện được thiết kế hiện đại, tương thích đa thiết bị, tích hợp hệ thống phát video với các tùy chọn chất lượng và phụ đề. Đảm bảo bảo mật qua mã hóa dữ liệu, xác thực, phân quyền, và ngăn chặn các tấn. Đồng thời, tối ưu hiệu năng và cấu trúc hệ thống để tăng tốc độ phản hồi, đảm bảo khả năng mở rộng và tích hợp thêm các tính năng nâng cao trong tương lai.

**5. Ý nghĩa khoa học và thực tiễn của đề tài**

* Ứng dụng công nghệ tiên tiến như streaming, nén video, và bảo mật, thúc đẩy phát triển công nghệ thông tin và truyền thông.
* Nghiên cứu hành vi người dùng, cải thiện trải nghiệm và tối ưu hóa giao diện, tốc độ tải trang, chất lượng video.
* Cung cấp sự tiện lợi, dễ dàng truy cập phim mọi lúc, mọi nơi, đáp ứng nhu cầu giải trí hiện đại.
* Mở rộng thị trường giải trí, tạo kênh phân phối mới, tăng cơ hội kinh doanh cho ngành công nghiệp điện ảnh.

**6. Kết quả đạt được**

* Một ứng dụng web xem phim trực tuyến hoàn chỉnh, đáp ứng các yêu cầu của người dùng.
* Tài liệu hướng dẫn sử dụng chi tiết.

**7. Cấu trúc đồ án**

* **Chương 1**: Tổng quan về ASP.NET FrameWork và mô hình MVC.

Chương này cung cấp nền tảng kiến thức về ba công nghệ quan trọng trong phát triển web: ASP.NET Framework, mô hình MVC, và Bootstrap. Sự kết hợp giữa chúng không chỉ mang lại khả năng xây dựng ứng dụng mạnh mẽ, mà còn nâng cao hiệu suất và trải nghiệm người dùng.

* **Chương 2**: Phân tích và thiết kế hệ thống xem phim trực tuyến dựa trên nền tảng ASP.NET mô hình MVC: trình bày mô tả hệ thống, các use case và các biểu đồ phân tích.

Trong chương này tập trung vào việc mô tả chi tiết các yêu cầu hệ thống, mô hình hóa chức năng qua các biểu đồ Use Case, và đặc tả cụ thể từng Use Case để chuẩn bị cho việc thiết kế và phát triển hệ thống.

* **Chương 3**: Thử nghiệm và đánh giá: Đưa ra mô hình hoạt động và tiến hành cài đặt, thử nghiệm và đánh giá.

Trong chương này ghi nhận việc hoàn thiện cài đặt hệ thống và thực hiện kiểm thử toàn diện để đảm bảo chất lượng sản phẩm. Hệ thống đã sẵn sàng chuyển sang giai đoạn triển khai và vận hành thực tế.

# CHƯƠNG 1. TỔNG QUAN VỀ ASP.NET FRAMEWORK VÀ MÔ HÌNH MVC

## 1.1. Công nghệ ASP.NET Framework.

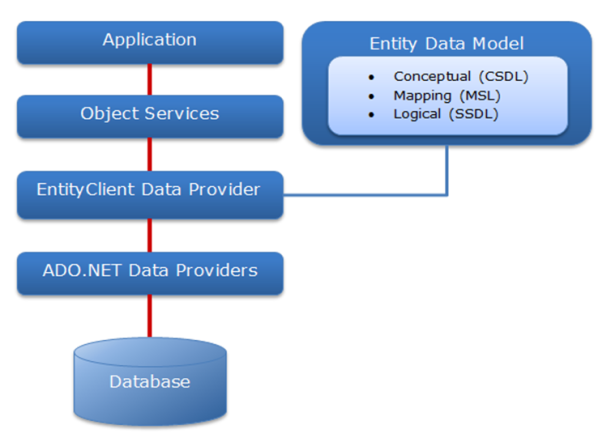
### 1.1.1. Framework ASP.NET

ASP.NET Framework là một nền tảng phát triển ứng dụng web do Microsoft phát triển, được thiết kế để giúp các lập trình viên dễ dàng tạo ra các ứng dụng web và dịch vụ web mạnh mẽ, linh hoạt. Nền tảng này hỗ trợ việc sử dụng nhiều ngôn ngữ lập trình khác nhau như C# và VB.NET, giúp cho việc phát triển trở nên tiện lợi và phù hợp với nhiều đối tượng lập trình viên. ASP.NET cung cấp một mô hình lập trình dựa trên sự kiện, nghĩa là các thao tác xử lý sẽ diễn ra khi có các sự kiện cụ thể xảy ra, chẳng hạn như người dùng nhấn nút hoặc gửi biểu mẫu.

Một trong những ưu điểm nổi bật của ASP.NET Framework là sự đa dạng trong các tính năng được tích hợp sẵn. Ví dụ, nền tảng này hỗ trợ mạnh mẽ việc quản lý trạng thái, cho phép giữ nguyên trạng thái của ứng dụng và dữ liệu người dùng trong quá trình tương tác với hệ thống. Ngoài ra, ASP.NET còn cung cấp các cơ chế xác thực (authentication) và ủy quyền (authorization), giúp đảm bảo tính bảo mật cho ứng dụng, chỉ cho phép những người dùng được cấp quyền truy cập vào các chức năng cụ thể.

Bên cạnh đó, ASP.NET hỗ trợ rất tốt việc quản lý và thao tác dữ liệu, đặc biệt là với các cơ sở dữ liệu thông qua các công cụ như Entity Framework hoặc ADO.NET. Điều này giúp các lập trình viên dễ dàng kết nối, truy vấn, và cập nhật dữ liệu từ các nguồn cơ sở dữ liệu khác nhau.

Với ASP.NET, việc xây dựng giao diện người dùng (UI) cũng trở nên đơn giản hơn nhờ vào các điều khiển web (Web Controls) có sẵn. Các điều khiển này giúp tạo ra các thành phần giao diện như biểu mẫu, nút nhấn, bảng dữ liệu một cách dễ dàng mà không cần phải viết quá nhiều mã. Ngoài ra, ASP.NET Framework cũng hỗ trợ mô hình MVC (Model-View-Controller), giúp phân tách các thành phần trong ứng dụng web một cách rõ ràng và dễ quản lý hơn.



Hình 1. 1 Kiến trúc Entity Framework

* Application (ứng dụng) là tầng chứa giao diện trang Web (HTML, CSS, Javascript, hình ảnh, …) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.
* Object Services là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.
* EntityClient Data Provider là tầng cung cấp các kết nối, diễn dịch các truy vấn thực thể thành truy vấn nguồn dữ liệu (chuyển L2E – LINQ to Entity hay các truy vấn thực thể SQL thành truy vấn SQL), trả về data reader để EF dùng chuyển dữ liệu thực thể thành các đối tượng. Phần này kết nối ADO.NET Data Providers để gửi hoặc lấy dữ liệu từ database. Tầng này hoàn toàn khác với EDM (Entity Data Model) khi thực thi các truy vấn tương tự như cách thực hiện ở ADO.NET Provider. EntityClient Data Provider sử dụng ESQL (Entity SQL), một ngôn ngữ truy vấn độc lập dạng văn bản, tương tự như SQL.
* ADO.NET Data Provider là tầng thấp nhất để dịch các truy vấn L2E (LINQ to Entity) thông qua câu lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.

### 1.1.2. Các tính năng của ASP.NET Framework.

* ASP.NET hỗ trợ cả hai mô hình lập trình là Web Forms và MVC (Model-View-Controller), cho phép lựa chọn theo sở thích và yêu cầu cụ thể của ứng dụng.
* ASP.NET cung cấp các cơ chế quản lý trạng thái như ViewState và Session để theo dõi trạng thái của ứng dụng trên nhiều yêu cầu từ người dùng.
* Xác thực và ủy quyền**:** Bảo vệ ứng dụng bằng cách xác thực người dùng và quản lý quyền truy cập thông qua ủy quyền.
* ASP.NETsử dụng mô hình lập trình dựa trên sự kiện, giúp xử lý sự kiện người dùng và tương tác với các yếu tố trên trang web.
* Hỗ trợ các công nghệ kết nối cơ sở dữ liệu như ADO.NET để tương tác với cơ sở dữ liệu như SQL Server, MySQL, Oracle, và nhiều hệ quản trị cơ sở dữ liệu khác.
* ASP.NET SignalR**:** Cung cấp mô hình lập trình RESTful, giúp tương tác với các ứng dụng và thiết bị khác nhau thời gian thực (real-time) thông qua kết nối hai chiều giữa máy chủ và trình duyệt.
* ASP.NET Core**:** Phiên bản tiếp theo của ASP.NET Framework, ASP.NET Core, mang lại tính di động và đa nền tảng, hỗ trợ Linux và macOS cũng như Windows.
* Hỗ trợ Ngôn ngữ: Hỗ trợ nhiều ngôn ngữ lập trình như C#, VB.NET, và F#.

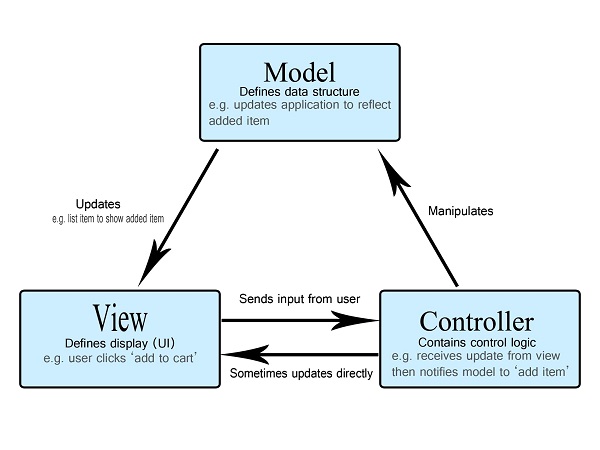
### 1.1.3. Lí do nên sử dụng ASP.NET Framework

* ASP.NET Framework được phát triển và hỗ trợ chính thức bởi Microsoft, điều này mang lại sự tích hợp mạnh mẽ với các sản phẩm và dịch vụ khác của Microsoft như Windows Server, SQL Server, Azure, Visual Studio, và nhiều hơn nữa.
* ASP.NET Framework hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C#, VB.NET, và F#, giúp nhà phát triển lựa chọn ngôn ngữ mà họ thoải mái và hiểu biết.
* ASP.NET Framework sử dụng mô hình lập trình dựa trên sự kiện, giúp quản lý tốt các sự kiện người dùng trên trang web và cung cấp một quy trình làm việc rõ ràng.
* ASP.NET Frameworkcung cấp các cơ chế quản lý trạng thái như ViewState và Session, giúp theo dõi trạng thái ứng dụng trên nhiều yêu cầu từ người dùng.
* ASP.NET Framework tích hợp nhiều tính năng bảo mật như xác thực và ủy quyền, kiểm soát truy cập, SSL, và bảo vệ chống tấn công.
* Phát triển nhanh chóng**:** Sử dụng Visual Studio IDE giúp nhà phát triển tận dụng được nhiều tính năng như gỡ lỗi, tự động hoá, và kiểm thử, giúp tiết kiệm thời gian và tăng hiệu suất phát triển.
* Đa nền tảng (ASP.NET Core)**:** Phiên bản ASP.NET Core mang lại tính di động và đa nền tảng, hỗ trợ Linux và macOS cũng như Windows, giúp ứng dụng có khả năng chạy trên nhiều môi trường.
* Hỗ trợ cộng đồng lớn**:** ASP.NET có một cộng đồng lớn và tích cực, với nhiều tài nguyên, tài liệu, và diễn đàn hỗ trợ trực tuyến.
* Dễ dàng tích hợp với cơ sở dữ liệu**:** Hỗ trợ tốt cho kết nối và tương tác với các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, và Oracle.
* ASP.NET Framework được tối ưu hóa để đạt được hiệu suất cao và xử lý một lượng lớn người dùng đồng thời.

## 1.2. Mô hình MVC.

### 1.2.1. Mô hình MVC

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình

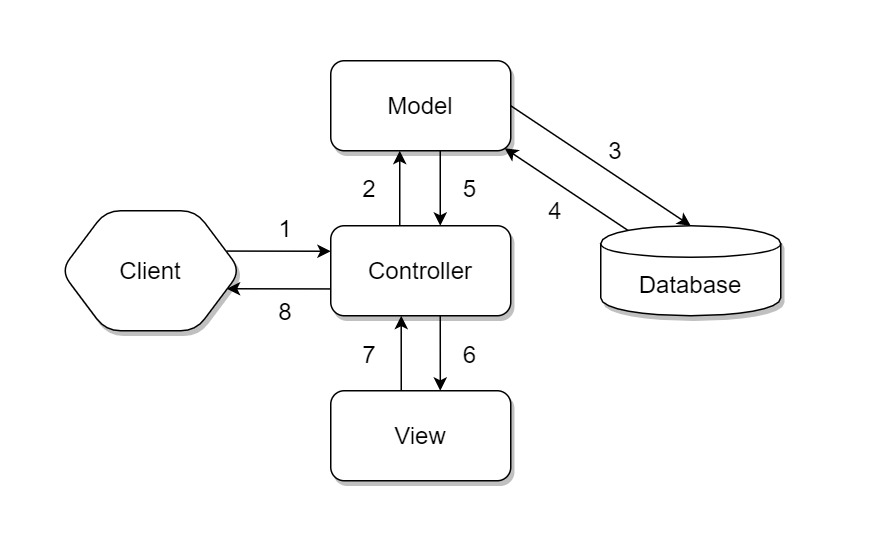


Hình 1. 2 Các thành phần bên trong của mô hình MVC

Các thành phần bên trong MVC:

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

### 1.2.2. Luồng đi trong mô hình MVC.



Hình 1. 3 Luồng xử lý trong MVC

1. Đầu tiên, client sẽ gửi một request tới server thông qua Controller.
2. Controller tiếp nhận request, và xử lý luồng đi tiếp theo của yêu cầu. Trong trường hợp chỉ chuyển từ trang này sang trang khác mà không yêu cầu gửi và nhận dữ liệu thì lúc này Controller sẽ redirect (chuyển hướng) lại cho browser và kết thúc luồng. Lúc này luồng xử lý sẽ đi theo thứ tự 1-6-7-8.
3. Nếu request của người dùng yêu cầu phải xử lý dữ liệu thì Controller gọi xuống Model để lấy dữ liệu. Lúc này Model sẽ tương tác với database để lấy dữ liệu, dữ liệu được Model gửi về Controller, Controller gọi đến View phù hợp với request kèm theo dữ liệu cho View, View sẽ lắp dữ liệu tương ứng vào HTML và gửi lại một HTML cho Controller sau khi thực hiện xong nhiệm vụ của mình. Cuối cùng Controller sẽ trả kết quả về Browser. Lúc này luồng xử lý sẽ đi từ 1-2-3-4-5-6-7-8.
4. Nếu request của người dùng chỉ yêu cầu dữ liệu nhưng không chuyển trang thì thay vì Controller sẽ trả dữ liệu về cho View thì Controller sẽ respone lại cho trình duyệt thông qua API, dữ liệu trả về thường sẽ là dạng JSON. Luồng xử lý lúc này sẽ là 1-2-3-4-5-8.

### 1.2.3. Ưu nhược điểm của mô hình MVC.

Ưu điểm của mô hình MVC

* + Nhẹ, tiết kiệm băng thông: việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn
  + Kiểm tra dễ dàng: Với MVC, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.
  + Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, Javascript có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.
  + View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
  + Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.
  + Tính kết hợp: Việc tích hợp ở mô hình MVC cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.
  + Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

Nhược điểm của mô hình MVC

* MVC thường được sử dụng vào những dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

### 1.2.4. Quy trình để phát triển mô hình Web MVC

Phát triển một ứng dụng web theo mô hình MVC (Model-View-Controller) là một quy trình phân tách rõ ràng giữa các thành phần để đảm bảo tính tổ chức, dễ bảo trì và mở rộng. Dưới đây là các bước cần thực hiện:

* 1. **Phân tích yêu cầu và thiết kế kiến trúc:**

Thu thập thông tin từ khách hàng hoặc tài liệu để hiểu rõ các chức năng cần thiết, phân tích các thực thể (entities) và mối quan hệ giữa chúng để xây dựng cấu trúc dữ liệu.

* 1. **Xây dựng thành phần Model:**
* Thiết kế và tạo cơ sở dữ liệu: Dựa trên mô hình dữ liệu, xây dựng cơ sở dữ liệu với các bảng, khóa chính, và ràng buộc.
* Tạo lớp Model: Viết các lớp đại diện cho dữ liệu và logic nghiệp vụ. Các lớp này sẽ bao gồm các phương thức truy vấn, thêm, sửa, xóa dữ liệu.
* Kết nối cơ sở dữ liệu: Cài đặt các lớp DAO (Data Access Object) hoặc sử dụng ORM (như Entity Framework, Hibernate) để giao tiếp với cơ sở dữ liệu.
  1. **Xây dựng thành phần Controller**
* Tạo các lớp Controller: Đây là nơi xử lý yêu cầu từ người dùng và điều hướng logic nghiệp vụ.
* Triển khai các phương thức: Mỗi phương thức trong Controller sẽ đảm nhận một hành động cụ thể, như hiển thị danh sách, xử lý form, hoặc điều hướng luồng ứng dụng.
* Xử lý dữ liệu: Lấy dữ liệu từ Model, xử lý, và chuẩn bị dữ liệu cho View.
  1. **Xây dựng thành phần View**
* Thiết kế giao diện: Tạo các tệp HTML, CSS để xây dựng giao diện người dùng.
* Tích hợp dữ liệu động: Sử dụng các công cụ hoặc framework (như Razor, JSP, hoặc React) để hiển thị dữ liệu từ Controller ra giao diện.
* Xử lý tương tác: Thêm các tính năng tương tác như form nhập liệu, nút bấm, hoặc AJAX để tăng tính năng động cho giao diện.
  1. **Tích hợp và kiểm tra:**
* Liên kết các thành phần: Đảm bảo các thành phần Model, View, Controller hoạt động nhịp nhàng với nhau.
* Kiểm tra đơn vị (Unit Test): Viết các bài kiểm tra cho từng thành phần để đảm bảo chúng hoạt động đúng.
* Kiểm tra tích hợp: Đảm bảo các thành phần hoạt động mượt mà khi kết hợp.

## 1.3. Bootstrap Framework

* Bootstrap là một framework front-end mã nguồn mở, được giới thiệu bởi Twitter, giúp các nhà phát triển xây dựng giao diện web hiện đại một cách nhanh chóng và dễ dàng.
* Framework này được xây dựng dựa trên ba thành phần chính: HTML, CSS, và JavaScript, cung cấp một bộ công cụ phong phú để tạo các thành phần giao diện tiêu chuẩn như nút bấm (buttons), biểu mẫu (forms), bảng (tables), menu điều hướng (navbar), và cửa sổ bật lên (modals).
* Một trong những đặc điểm nổi bật nhất của Bootstrap là hệ thống lưới (grid system) linh hoạt, cho phép chia bố cục trang web thành các cột và hàng một cách dễ dàng. Nhờ hệ thống lưới này, lập trình viên có thể tạo ra các giao diện tương thích với nhiều kích thước màn hình khác nhau, từ máy tính để bàn lớn đến các thiết bị di động nhỏ.
* Bootstrap tích hợp sẵn khả năng thiết kế đáp ứng (responsive design), giúp giao diện của trang web tự động điều chỉnh để hiển thị tối ưu trên mọi thiết bị mà không cần viết lại mã. Điều này làm giảm đáng kể thời gian và công sức để phát triển các trang web đa nền tảng.
* Framework này đi kèm với một tập hợp các tiện ích CSS và JavaScript mạnh mẽ, giúp xử lý các tác vụ phổ biến như hiển thị slider, dropdown menu, hoặc tooltip một cách dễ dàng mà không cần viết thêm mã phức tạp.
* Bootstrap hỗ trợ khả năng tùy chỉnh cao. Người dùng có thể thay đổi các biến CSS gốc hoặc sử dụng các công cụ như SASS/LESS để điều chỉnh giao diện theo phong cách thiết kế riêng. Ngoài ra, phiên bản biên dịch sẵn của Bootstrap cũng cho phép sử dụng nhanh mà không cần cấu hình nhiều.
* Một ưu điểm khác của Bootstrap là tài liệu chi tiết, dễ hiểu, và được tổ chức tốt, giúp cả người mới bắt đầu lẫn lập trình viên có kinh nghiệm dễ dàng học tập và áp dụng.
* Cộng đồng người dùng và nhà phát triển của Bootstrap rất lớn, với nhiều nguồn tài nguyên mở rộng như thư viện themes, plugin, và ví dụ mẫu, hỗ trợ việc phát triển giao diện nhanh chóng hơn.
* Nhờ tính phổ biến và tiện dụng, Bootstrap đã trở thành một lựa chọn hàng đầu cho các dự án web từ cá nhân đến doanh nghiệp, giúp tiết kiệm thời gian, giảm chi phí phát triển, và đảm bảo giao diện nhất quán trên các trình duyệt và thiết bị khác nhau.

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Giới thiệu ứng dụng Web xem phim trực tuyến MWatch

### 2.1.1. Mô tả ứng dụng web Mwatch

Ứng dụng web MWatch được thiết kế nhằm mang đến một nền tảng xem phim trực tuyến hiện đại, tập trung vào việc cung cấp trải nghiệm giải trí chất lượng cao và thuận tiện cho người dùng. Với giao diện thân thiện và dễ sử dụng, MWatch cho phép người dùng nhanh chóng tìm kiếm, chọn lọc và thưởng thức các bộ phim phù hợp với sở thích cá nhân.

Để nâng cao tính cá nhân hóa, hệ thống hỗ trợ đăng ký tài khoản, theo dõi lịch sử xem phim, và lưu trữ danh sách phim yêu thích. Ngoài ra, người dùng có thể tham gia đánh giá và bình luận, góp phần xây dựng một cộng đồng tương tác sôi động và gắn kết.

Quản trị viên sẽ có bộ công cụ chuyên biệt để quản lý nội dung phim, giám sát hoạt động người dùng, và kiểm duyệt đánh giá, đảm bảo chất lượng nội dung luôn được duy trì ở mức cao.

MWatch được thiết kế với khả năng mở rộng linh hoạt, đáp ứng nhu cầu ngày càng gia tăng của người dùng, đồng thời khắc phục những hạn chế của các nền tảng xem phim hiện có. Hệ thống hướng tới mục tiêu trở thành một nền tảng giải trí trực tuyến toàn diện, chuyên nghiệp, và tiện lợi cho cả người dùng lẫn nhà quản lý.

### 2.1.2 Yêu cầu hệ thống

**Các yêu cầu về chức năng với khách hàng**

* Đăng ký: Người dùng phải nhập đầy đủ thông tin để có thể đăng ký thành công tài khoản .
* Đăng nhập: Khi người dùng nhập tên tài khoản và mật khẩu đúng với tài khoản đã đăng ký thì có thể đăng nhập vào hệ thống.
* Xem trang chủ home: người dùng có thể xem thông tin về các phim, tin tức.
* Tìm kiếm: Người dùng có thể tìm kiếm phim bằng cách nhập từ khóa vào ô tìm kiếm và click chọn nút Tìm kiếm.
* Bình luận : Khi người dùng click vào bình buận trong trang chi tiết thì có thể bình luận

**Các yêu cầu về chức năng với người quản trị:**

* Người quản trị có thể quản lý thêm, sửa,xóa phim.
* Người quản trị có thể quản lý phim, cập nhật phim.
* Người quản trị có thể quản lý thêm, sửa, xóa thông tin bài đăng
* Người quản trị có thể quản lý tài khoản người dùng.

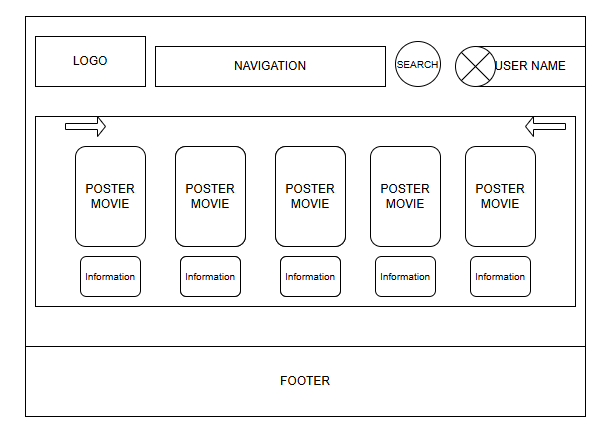
**Các yêu cầu phi chức năng**

* Giao diện chuyên nghiệp, đẹp mắt thân thiện với người dùng.
* Tốc độ tải trang nhanh.
* Độ khả dụng, tin cậy cao.
* Khả năng phục hồi tốt.
* Có khả năng bảo trì dễ dàng.
* Khả năng quản lý tốt.
* Toàn vẹn dữ liệu.
* Có khả năng sử dụng và tương tác tốt.

### 2.1.3 Thiết kế giao diện Ứng dụng Web MWatch

**Giao diện trang chủ**

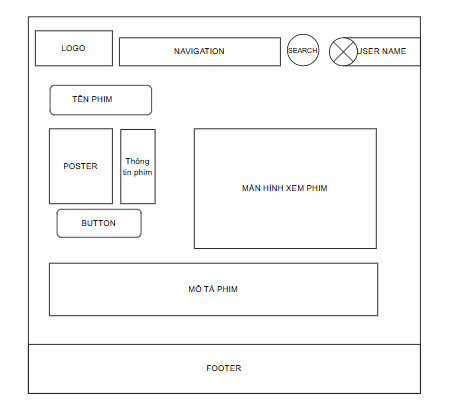
Ở giao diện này cung cấp các thông tin về logo trang web, thanh menu, hiển thị thanh công cụ tìm kiếm, thông tin người dùng, danh sách các phim hiển thị Poster phim, tên phim, tên thể loại và số sao đánh giá.



Hình 2. 1 Giao diện trang chủ

**Giao diện chi tiết phim**

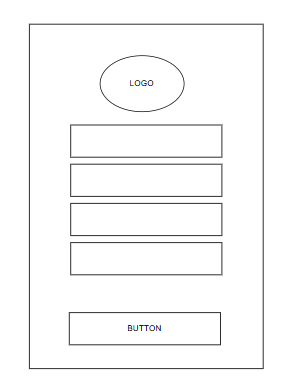
Giao diện này cung cấp thông tin chi tiết của phim và cho phép người dùng có thể xem phim, bình luận, đánh giá phim, thêm phim vào danh sách yêu thích

****

Hình 2. 2 Giao diện chi tiết phim

**Giao diện đăng ký người dùng**

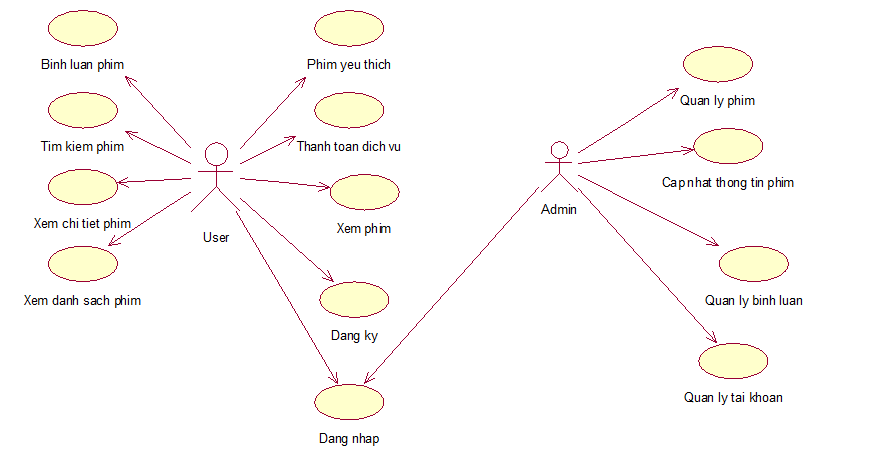
Giao diện này cho phép người dùng đăng ký tài khoản mới, khi điền đầy đủ thông tin, người dùng có thể bấm chọn BUTTON để đăng ký tài khoản mới

****

Hình 2. 3 Giao diện đăng ký người dùng

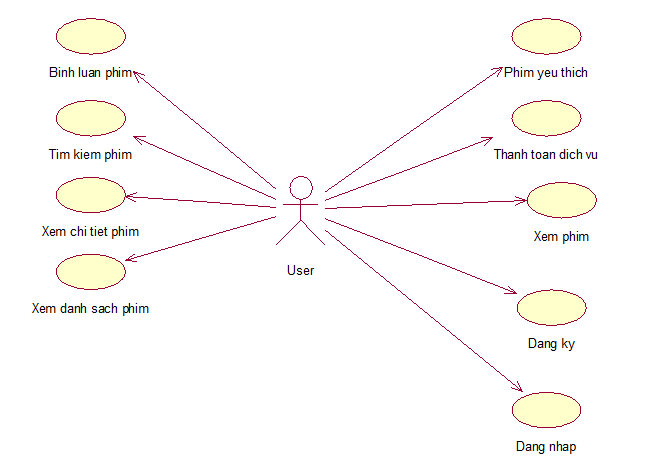
## 2.2 Mô hình hóa chức năng

### 2.2.1 Biểu đồ use case



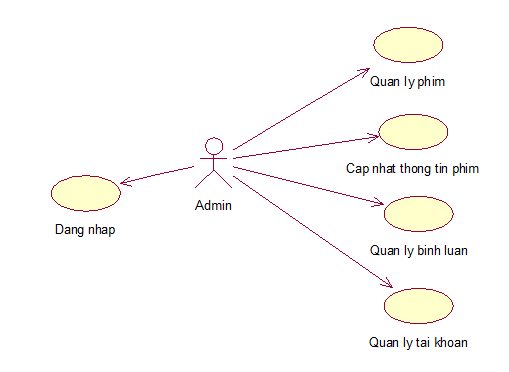
Hình 2. Biểu đồ Use case

### 2.2.2 Biểu đồ use case phía người dùng



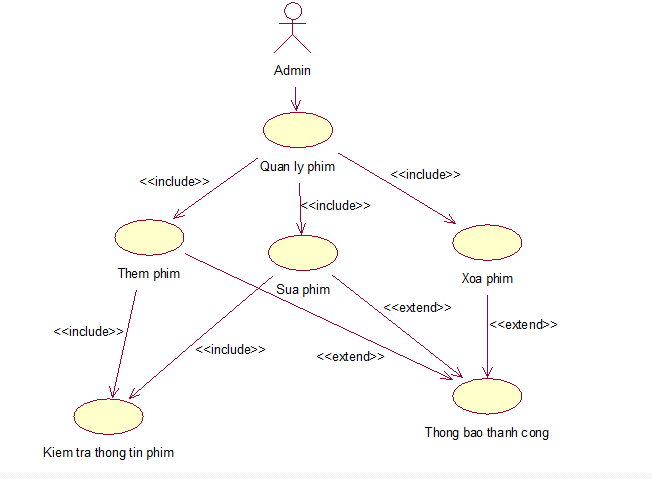
Hình 2. Biểu đồ Use case người dùng

### 2.2.3 Biểu đồ use cae phía quản trị viên

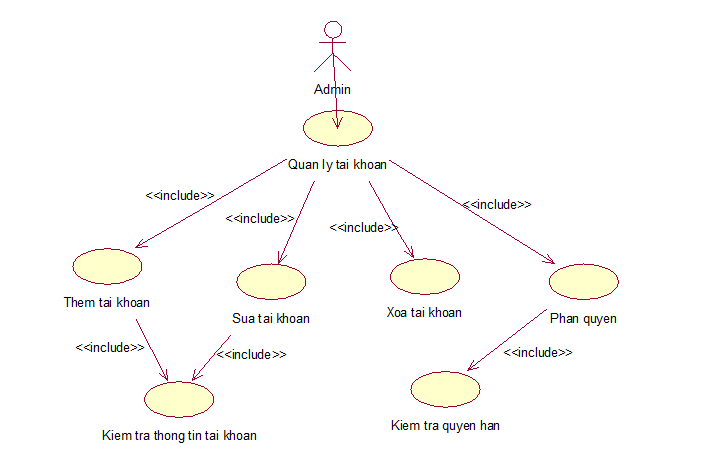


Hình 2. Biểu đồ Use case phía quản trị viên

### 2.2.4 Phân rã biểu đồ use case quản lý phim



Hình 2. Phân rã biểu đồ Use case quản lý phim

2.2.5 Phân rã biểu đồ use case quản lý tài khoản   


Hình 2. Phân rã biểu đồ Use case quản lý tài khoản

## 2.3 Đặc tả các use case

### 2.3.1 Use case xem danh sách phim

Use case này cho phép người dùng xem danh sách các bộ phim hiện có trên trang web.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn “MWatch” trên thanh menu.
2. Hệ thống sẽ lấy thông tin về các bộ phim gồm tên phim, hình ảnh, hiển thị lên màn hình trang chủ.
3. Người dùng có thể chọn xem chi tiết hoặc bắt đầu xem một bộ phim từ danh sách.
4. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Tại bước 2 trong luồng cơ bản, nếu chưa đường dẫn sai hoặc không có internet thì hệ thống sẽ thông báo: “không có dữ liệu” và use case kết thúc.

**Các yêu cầu đặc biệt:**

Không có.

**Tiền điều kiện:**

Không có.

**Hậu điều kiện:**

Không có.

**Điểm mở rộng:**

Không có.

### 2.3.2 Use case xem chi tiết phim

Use case này cho phép người dùng xem chi tiết thông tin của một bộ phim cụ thể.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn hình ảnh của một bộ phim trong danh sách phim.
2. Hệ thống sẽ lấy thông tin về bộ phim bao gồm tên phim, hình ảnh, mô tả chi tiết, danh sách diễn viên và hiển thị lên màn hình.
3. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Tại bước 1 trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc.

**Các yêu cầu đặc biệt:**

Không có

**Tiền điều kiện:**

Không có.

**Hậu điều kiện:**

Không có.

**Điểm mở rộng:**

Không có.

### 2.3.3 Use case xem phim

Use case này cho phép người dùng xem một bộ phim trực tuyến.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn biểu tượng xem phim trên trang chi tiết của một bộ phim.
2. Hệ thống sẽ tải và phát bộ phim, hiển thị trên màn hình với các tùy chọn điều khiển như tua nhanh, dừng.
3. Người dùng có thể tạm dừng, tiếp tục hoặc thoát khỏi chế độ xem phim bất cứ lúc nào.
4. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Tại 2 trong luồng cơ bản, nếu không kết nối được với máy chủ hoặc đường truyền internet kém, hệ thống sẽ thông báo lỗi và không thể phát phim. Use case kết thúc.

**Các yêu cầu đặc biệt:**

Không có.

**Tiền điều kiện:**

Không có.

**Hậu điều kiện:**

Không có.

**Điểm mở rộng:**

Không có.

### 2.3.4 Use case đăng kí tài khoản

Use case này cho phép người dùng tạo một tài khoản mới trên hệ thống.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn "Đăng ký" trên trang chủ.
2. Hệ thống hiển thị biểu mẫu yêu cầu người dùng nhập thông tin cá nhân bao gồm tên người dùng, mật khẩu và các thông tin khác nếu cần.
3. Người dùng điền đầy đủ thông tin và nhấn nút "Đăng ký".
4. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Tại bước 3 trong luồng cơ bản, nếu thông tin không hợp lệ (ví dụ mật khẩu đã tồn tại), hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại thông tin. Use case quay lại bước 2.
2. Nếu có sự cố khi lưu thông tin vào cơ sở dữ liệu, hệ thống sẽ thông báo lỗi và use case kết thúc.thúc.

**Các yêu cầu đặc biệt:**

Không có.

**Tiền điều kiện:**

Không có.

**Hậu điều kiện:**

Không có.

**Điểm mở rộng:**

Không có.

### 2.3.5 Use case đăng nhập

Use case này cho phép người dùng và quản trị viên đăng nhập vào hệ thống bằng tài khoản đã đăng ký.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn "Login" trên trang chủ.
2. Hệ thống hiển thị biểu mẫu yêu cầu người dùng nhập tên đăng nhập và mật khẩu.
3. Người dùng nhập thông tin và nhấn nút "Login".
4. Hệ thống kiểm tra thông tin đăng nhập.
5. Nếu thông tin đúng, hệ thống cho phép người dùng truy cập tài khoản của họ.
6. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Tại bước 4 trong luồng cơ bản, nếu thông tin đăng nhập sai, hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại. Use case quay lại bước 3.
2. Nếu có sự cố với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:**

Không có.

**Tiền điều kiện:**

Tài khoản đã tồn tại trong hệ thống

**Hậu điều kiện:**

Người dùng đăng nhập thành công và có thể sử dụng các tính năng dành riêng cho thành viên.

**Điểm mở rộng:**

Không có.

### 2.3.6 Use case tìm kiếm phim

Use case này cho người dùng tìm kiếm các bộ phim có sẵn trên hệ thống.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng nhập từ khóa vào thanh tìm kiếm trên trang chủ.
2. Hệ thống tìm kiếm các bộ phim khớp với từ khóa dựa trên tên phim, diễn viên, đạo diễn.
3. Hệ thống hiển thị danh sách các phim khớp với từ khóa.
4. Người dùng chọn một phim từ danh sách để xem chi tiết.
5. Use case kết thúc

**Luồng rẽ nhánh:**

1. Tại bước 2 trong luồng cơ bản, nếu không tìm thấy phim nào khớp với từ khóa, hệ thống sẽ thông báo "Không tìm thấy kết quả" và use case kết thúc.

**Các yêu cầu đặc biệt:**

Không có.

**Tiền điều kiện:**

Không

**Hậu điều kiện:**

Danh sách phim khớp với từ khóa được hiển thị cho người dùng.

**Điểm mở rộng:**

Không có.

### 2.3.7 Use case bình luận phim

Use case này cho phép người dùng bình luận về các bộ phim đã xem.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn bình luận trên trang chi tiết phim.
2. Hệ thống hiển thị biểu mẫu yêu cầu người dùng nhập nội dung đánh giá hoặc bình luận.
3. Người dùng điền thông tin và nhấn nút "Gửi".
4. Hệ thống lưu bình luận vào cơ sở dữ liệu và hiển thị nó trên trang chi tiết phim.
5. Use case kết thúc

**Luồng rẽ nhánh:**

1. Tại bước 4 trong luồng cơ bản, nếu có sự cố khi lưu thông tin vào cơ sở dữ liệu, hệ thống sẽ thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:**

Chỉ thành viên đã đăng nhập mới có thể thực hiện đánh giá hoặc bình luận.

**Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống.

**Hậu điều kiện:**

Đánh giá hoặc bình luận được hiển thị công khai trên trang chi tiết phim.

**Điểm mở rộng:**

Không có.

### 2.3.8 Use case quản lý phim

Use case này cho phép quản trị viên quản lý danh sách phim trên hệ thống, bao gồm thêm, sửa, và xóa phim.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi quản trị viên chọn "Quản lý phim" từ trang quản trị.

1. Hệ thống hiển thị danh sách các phim hiện có và các tùy chọn để thêm mới, chỉnh sửa hoặc xóa phim.
2. Quản trị viên chọn một trong các tùy chọn: thêm mới, sửa hoặc xóa phim:

* Thêm mới: Hệ thống hiển thị biểu mẫu để nhập thông tin phim mới. Quản trị viên nhập thông tin và nhấn "Lưu". Hệ thống lưu phim mới vào cơ sở dữ liệu.
* Sửa: Quản trị viên chọn một phim trong danh sách để chỉnh sửa thông tin. Hệ thống hiển thị biểu mẫu chỉnh sửa. Quản trị viên cập nhật thông tin và nhấn "Lưu". Hệ thống cập nhật thông tin phim trong cơ sở dữ liệu.
* Xóa: Quản trị viên chọn một phim trong danh sách và nhấn "Xóa". Hệ thống xác nhận hành động và xóa phim khỏi cơ sở dữ liệu.

1. Hệ thống cập nhật danh sách phim sau khi thực hiện các thay đổi.
2. Use case kết thúc

**Luồng rẽ nhánh:**

1. Tại bước 1 trong luồng cơ bản, nếu trong bảng khóa học chưa có bài viết nào thì hệ thống sẽ đưa ra một thông báo: “Chưa có dữ liệu!” và use case kết thúc.
2. Tại bước 2 nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:**

Chỉ quản trị viên mới có quyền truy cập và quản lý danh sách phim.

**Tiền điều kiện:**

Quản trị viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:**

Danh sách phim được cập nhật và lưu trữ trong cơ sở dữ liệu.

**Điểm mở rộng:**

Không có.

### 2.3.9 Use case quản lý tài khoản

Use case này cho phép quản trị viên quản lý danh sách thành viên, bao gồm xem thông tin, xóa tài khoản hoặc phân quyền tài khoản thành viên.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi quản trị viên chọn "Quản lý thành viên" từ trang quản trị.
2. Hệ thống hiển thị danh sách các thành viên hiện có và các tùy chọn để xem thông tin chi tiết, phân quyền hoặc xóa tài khoản.
3. Quản trị viên chọn một trong các tùy chọn:
4. Xem thông tin chi tiết: Quản trị viên chọn một thành viên từ danh sách để xem thông tin chi tiết. Hệ thống hiển thị các thông tin chi tiết về tài khoản.
5. Phân quyền tài khoản: Quản trị viên chọn một thành viên và nhấn phân quyền. Hệ thống xác nhận hành động và phân quyền tài khoản của thành viên đó.
6. Xóa tài khoản: Quản trị viên chọn một thành viên và nhấn "Xóa". Hệ thống xác nhận hành động và xóa tài khoản khỏi cơ sở dữ liệu.
7. Hệ thống cập nhật danh sách thành viên sau khi thực hiện các thay đổi.
8. Use case kết thúc

**Luồng rẽ nhánh:**

1. Tại bước 3 trong luồng cơ bản, nếu có sự cố khi cập nhật thông tin vào cơ sở dữ liệu, hệ thống sẽ thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:**

Chỉ quản trị viên mới có quyền truy cập và quản lý danh sách thành viên.

**Tiền điều kiện:**

Quản trị viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:**

Danh sách thành viên được cập nhật và lưu trữ trong cơ sở dữ liệu.

**Điểm mở rộng:**

Không có.

### 2.3.10 Use case cập nhật thông tin phim

Use case này cho phép quản trị viên cập nhật thông tin chi tiết của một phim đã có trong hệ thống.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi quản trị viên chọn "Quản lý phim" và sau đó chọn một phim để chỉnh sửa.
2. Hệ thống hiển thị biểu mẫu chỉnh sửa thông tin của phim đã chọn.
3. Quản trị viên cập nhật các thông tin cần thiết (ví dụ: tên phim, mô tả, thể loại, poster, thời gian chiếu, ...) và nhấn "Lưu".
4. Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập.
5. Nếu dữ liệu hợp lệ, hệ thống cập nhật thông tin phim vào cơ sở dữ liệu và hiển thị thông báo thành công.
6. Hệ thống quay lại danh sách phim đã cập nhật.
7. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu thông tin nhập không hợp lệ (ví dụ: tên phim trống hoặc dữ liệu không đúng định dạng), hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
2. Nếu không kết nối được với cơ sở dữ liệu, hệ thống hiển thị thông báo lỗi và use case kết thúc.

**Yêu cầu đặc biệt:**

* Chỉ quản trị viên mới có quyền cập nhật thông tin phim.

**Tiền điều kiện:**

* Quản trị viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:**

* Thông tin phim được cập nhật và lưu trữ trong cơ sở dữ liệu.

**Điểm mở rộng:**

* Hệ thống có thể lưu trữ lịch sử cập nhật của phim để quản lý thay đổi.

### 2.3.11 Use case quản lý bình luận.

Use case này cho phép quản trị viên quản lý các bình luận của người dùng trên các phim, bao gồm xóa bình luận.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi quản trị viên chọn "Quản lý bình luận" từ trang quản trị.
2. Hệ thống hiển thị danh sách các bình luận trên các phim, cùng với các tùy chọn để xóa bình luận.
3. Quản trị viên chọn một bình luận trong danh sách và chọn hành động:
   * **Xóa:** Quản trị viên chọn "Xóa", hệ thống xác nhận hành động và xóa bình luận khỏi cơ sở dữ liệu.
4. Hệ thống cập nhật danh sách bình luận sau khi thực hiện các thay đổi.
5. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Nếu không có bình luận nào trong hệ thống, hệ thống hiển thị thông báo "Chưa có bình luận nào!" và use case kết thúc.
2. Nếu không kết nối được với cơ sở dữ liệu, hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

**Yêu cầu đặc biệt:**

* Chỉ quản trị viên mới có quyền truy cập và quản lý bình luận.

**Tiền điều kiện:**

* Quản trị viên đã đăng nhập vào hệ thống.

**Hậu điều kiện:**

* Danh sách bình luận được cập nhật và lưu trữ trong cơ sở dữ liệu.

**Điểm mở rộng:**

* Có thể thêm tính năng lọc bình luận theo trạng thái hoặc người dùng.

### 2.3.12 Use case Thanh toán dịch vụ.

Người dùng thanh toán cho dịch vụ (gói phim) trên hệ thống.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Người dùng chọn dịch vụ muốn thanh toán.
2. Hệ thống hiển thị chi tiết dịch vụ và số tiền cần thanh toán.
3. Người dùng chọn phương thức và nhập thông tin thanh toán.
4. Hệ thống xử lý thanh toán qua cổng thanh toán.
5. Nếu thành công, hệ thống cập nhật trạng thái dịch vụ và hiển thị xác nhận.
6. Use case kết thúc.

**Luồng rẽ nhánh:**

1. Thông tin thanh toán sai → Hiển thị lỗi, yêu cầu nhập lại.
2. Thanh toán thất bại → Hiển thị lỗi, yêu cầu thử lại.
3. Người dùng hủy giao dịch → Quay lại trang chi tiết dịch vụ.

**Yêu cầu đặc biệt:**

* Chỉ người dùng đã đăng nhập mới được thanh toán.

**Tiền điều kiện:**

* Người dùng đã đăng nhập.

**Hậu điều kiện:**

* Thanh toán được ghi nhận, trạng thái dịch vụ được cập nhật.

### 2.3.13 Use case quản lý phim yêu thích.

Use case này cho phép người dùng thêm phim vào mục phim yêu thích.

**Luồng sự kiện:**

**Luồng cơ bản:**

1. Use case bắt đầu khi người dùng chọn "Phim yêu thích" từ trang cá nhân.
2. Hệ thống hiển thị danh sách các phim yêu thích của người dùng và các tùy chọn để xem thông tin chi tiết hoặc xóa phim khỏi danh sách yêu thích.

* Người dùng chọn một trong các tùy chọn:
  + Xem thông tin chi tiết: Người dùng chọn một phim từ danh sách để xem thông tin chi tiết. Hệ thống hiển thị các thông tin chi tiết về phim.
  + Xóa khỏi danh sách yêu thích: Người dùng chọn một phim và nhấn "Xóa". Hệ thống xác nhận hành động và xóa phim đó khỏi danh sách yêu thích của người dùng.
  + Hệ thống cập nhật danh sách phim yêu thích sau khi thực hiện các thay đổi.

3.Use case kết thúc.

**Luồng rẽ nhánh:**

Tại bước 2 trong luồng cơ bản, nếu có sự cố khi cập nhật thông tin vào cơ sở dữ liệu, hệ thống sẽ thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:**

Chỉ người dùng đã đăng nhập mới có quyền truy cập và quản lý danh sách phim yêu thích của họ.

**Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống

**Hậu điều kiện:**

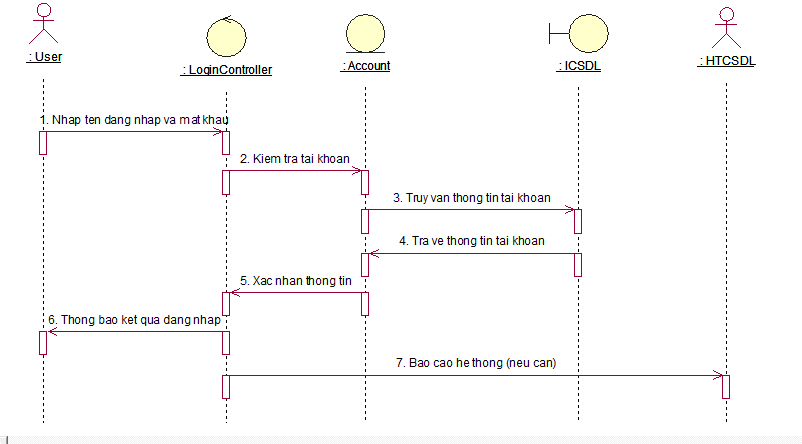
Danh sách phim yêu thích được cập nhật và lưu trữ trong cơ sở dữ liệu.

**Điểm mở rộng:**

Không có.

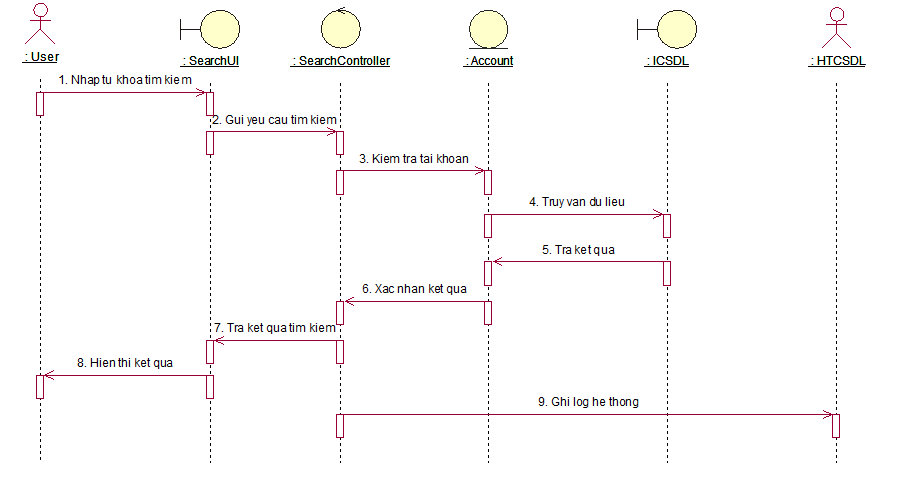
## 2.4 Biểu đồ trình tự của các use case

### 2.4.1 Biểu đồ trình tự cho quá trình đăng nhập



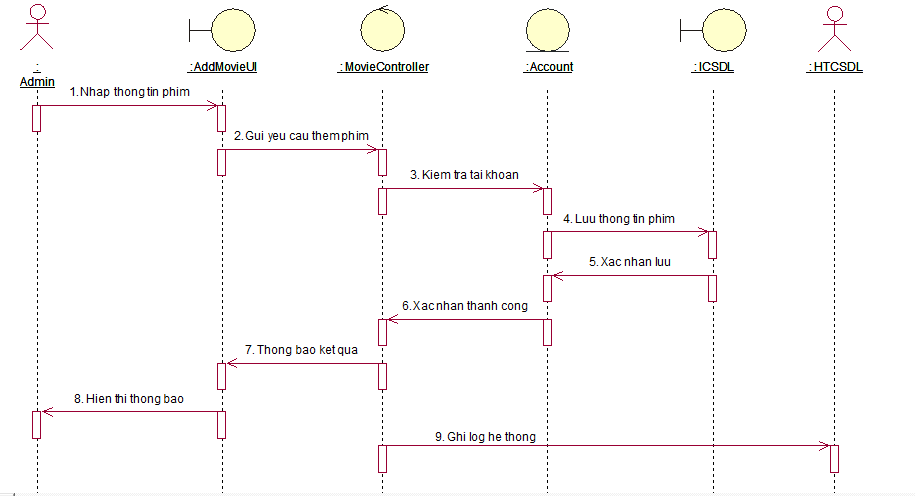
Hình 2. Biểu đồ trình tự use case đăng nhập

### 2.4.2 Biểu đồ trình tự cho quá trình tìm kiếm



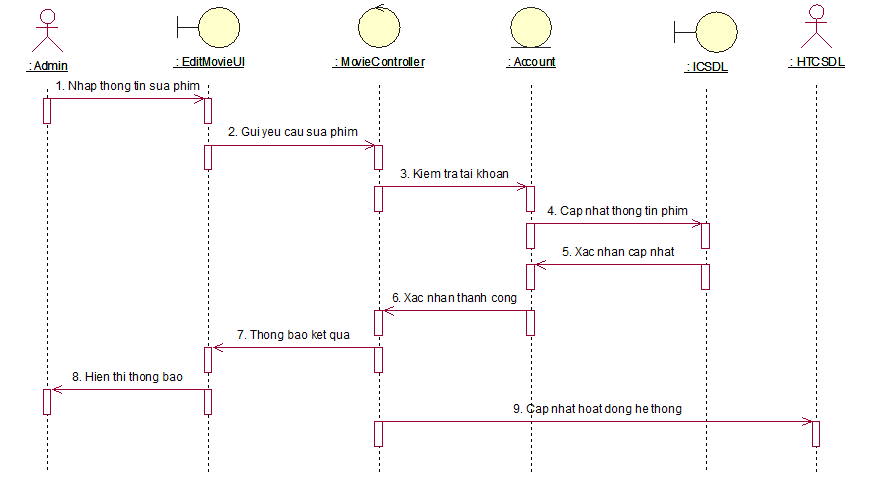
Hình 2. Biểu đồ trình tự use case tìm kiếm

### 2.4.3 Biểu đồ trình tự cho quá trình thêm phim



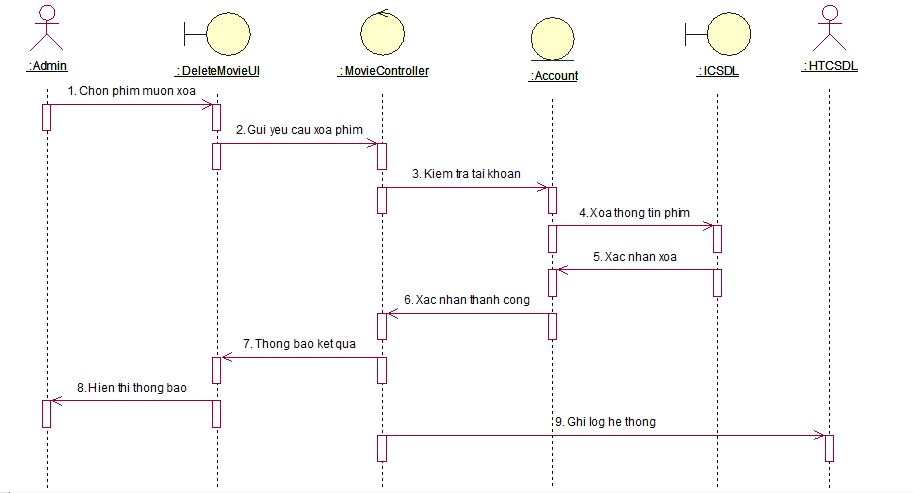
Hình 2. 11 Biểu đồ trình tự cho quá trình thêm phim

### 2.4.4 Biểu đồ trình tự cho quá trình sửa phim



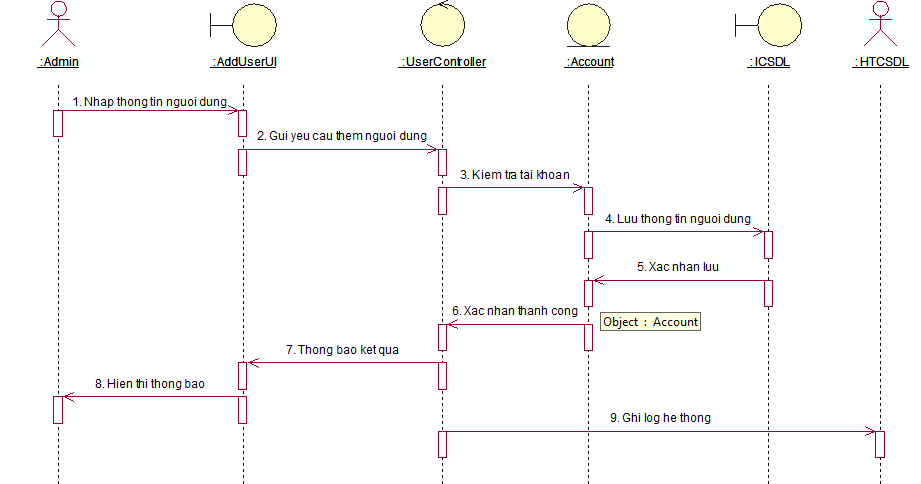
Hình 2. 12 Biểu đồ trình tự cho quá trình sửa phim

### 2.4.5 Biểu đồ trình tự cho quá trình xoá phim



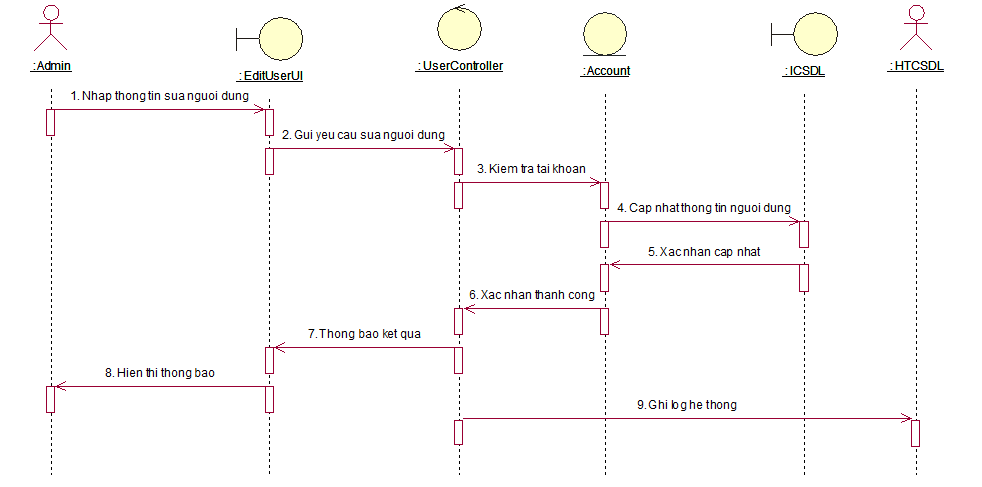
Hình 2. 13 Biểu đồ trình tự cho quá trình xoá phim

### 2.4.6 Biểu đồ trình tự cho chức năng thêm user



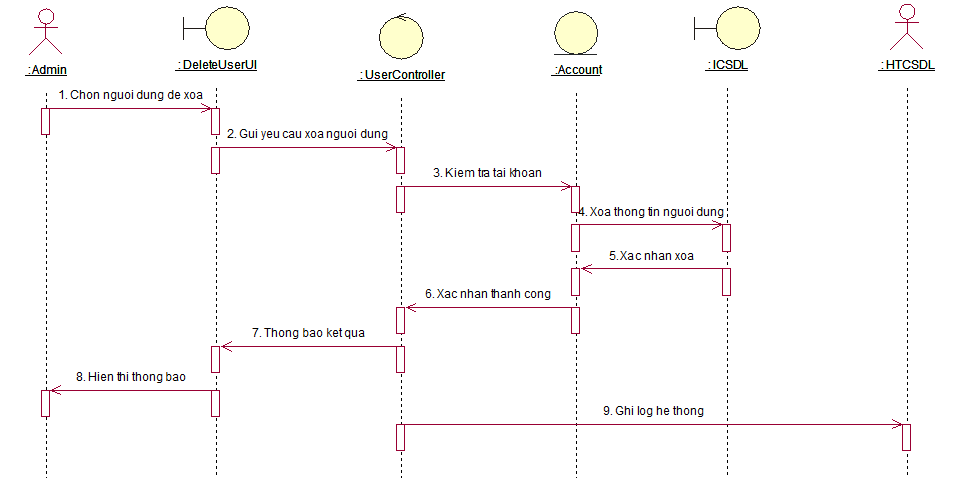
Hình 2. Biểu đồ trình tự cho chức năng thêm user

### 2.4.7 Biểu đồ trình tự cho chức năng sửa user



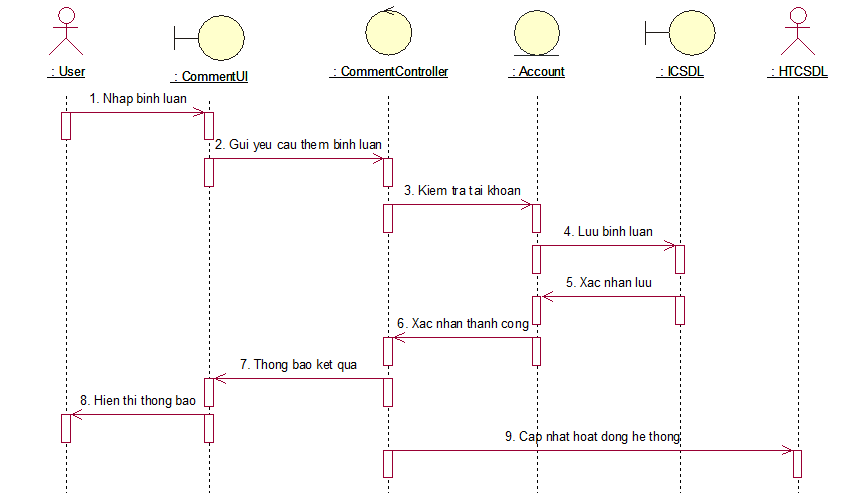
Hình 2. 15 Biểu đồ trình tự cho chức năng sửa user

### 2.4.8 Biểu đồ trình tự chức năng xoá user



Hình 2. Biểu đồ trình tự chức năng xoá user

### 2.4.9 Biểu đồ trình tự chức năng thêm bình luận



Hình 2. Biểu đồ trình tự chức năng thêm bình luận

## 2.5 Các bảng dữ liệu

### 2.5.1 Các bảng dữ liệu

**a) Bảng user**

Bảng 2. Bảng user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | UserID | int | Not null | Khóa chính |
| 2 | Name | Varchar (191) | Not null | Tên user |
| 3 | UserName | Varchar (100) | Not null | Tên đăng nhập |
| 4 | Password | Varchar (50) | Not null | Mật khẩu |
| 5 | CreatedDate | datetime | Null | Ngày tạo |

**b) Bảng Movie**

Bảng 2. Bảng movie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1. | MovieID | int | Not null | Khóa chính |
| 2. | Name | Nvarchar (100) | Not null | Tên phim |
| 3. | Image | Nvarchar (max) | Not null | Ảnh phim |
| 4. | Actor | Nvarchar (100) | Not null | Diễn viên |
| 5. | Description | Nvarchar (100) | Not null | Nội dung |
| 6. | Directors | Nvarchar (100) | Not null | Đạo diễn |
| 7. | Time | Nvarchar (100) | Not null | Thời gian |
| 8. | Year | Nvarchar (100) | Not null | Năm sản xuất |
| 9. | Country | Nvarchar (100) | Not null | Quốc gia |
| 10. | MovieLink | Nvarchar (max) | Not null | Liên kết phim |
| 11. | TrailerLink | Nvarchar (max) | Not null | Liên kết Trailer |
| 12. | CategoryID | int | Not null | Mã thể loại |
| 13. | Rate | int | Not null | Điểm đánh giá |
| 14. | Viewed | int | Not null | Lượt xem |
| 15. | Status | bit | Not null | Trạng thái |
| 16. | CreatedDate | datetime | Not null | Ngày tạo |

**c) Bảng favoriteMovies**

Bảng 2. Bảng favoriteMovies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
|  | FavoriteID | int | Not null | Khóa chính |
|  | UserID | int | Not null | Khoá ngoại |
|  | MovieID | int | Not null | Khoá ngoại |
|  | CreatesAt | datetime | null | Ngày thêm |

**d) Bảng comments**

Bảng 2. Bảng comments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
|  | CommentID | int | Not null | Khóa chính |
|  | MovieID | int | Not null | Khóa ngoại tham chiếu Movies |
|  | UserID | int | Not null | Khóa ngoại tham chiếu Users |
|  | Content | text | Not null | Nội dung bình luận |
|  | CreatedAt | datetime | Null | Thời gian bình luận |

**e) Bảng MovieGenres**

Bảng 2. Bảng MovieGenres

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MovieID | int | Not null | Khóa ngoại tham chiếu Movies |
| 2 | GenreID | int | Not null | Khóa ngoại tham chiếu Genres |
| 3 | PRIMARY KEY | (MovieID, GenreID) |  | Khóa chính kết hợp |

**f) Bảng Genres**

Bảng 2.6 Bảng Genres

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | GenreID | int | Not null | Khóa chính |
| 2 | GenreName | Varchar(100) | Not null | Tên thể loại phim |

**g) Bảng payment**

*Bảng 2.7 Bảng payment*

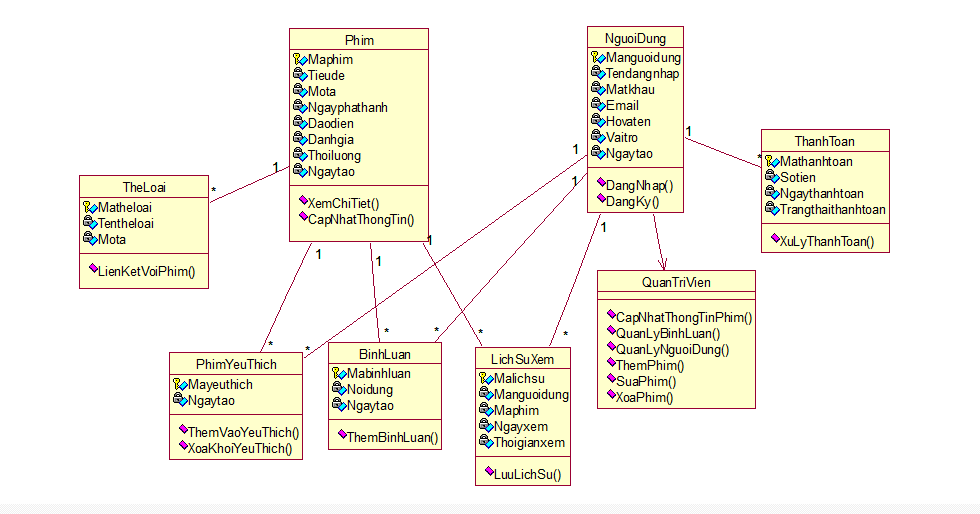
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | PaymentID | int | Not null | Khóa chính |
| 2 | Amount | Decimal (10, 2) | Not null | Số tiền thanh toán |
| 3 | PaymentDate | datetime | Not null | Ngày thanh toán |
| 4 | Status | Varchar (50) | Not null | Trạng thái thanh toán |

**h) Bảng WatchHistory**

*Bảng 2.8 Bảng watchHistory*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | WatchHistoryID | int | Not null | Khóa chính |
| 2 | UserID | int | Not null | Khóa ngoại đến người dùng |
| 3 | MovieID | int | Not null | Khóa ngoại đến phim |
| 4 | WatchDate | datetime | Not null | Ngày xem |
| 5 | WatchTime | int | A. Null | Thời gian đã xem |

### 2.5.2 Biểu đồ cơ sở dữ liệu



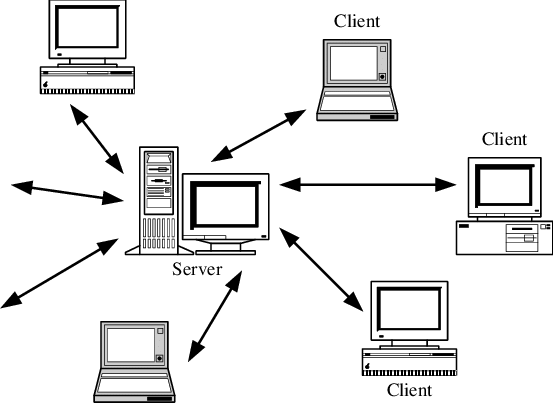
Hình 2. Biểu đồ cơ sở dữ liệu

# CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ



## Mô hình thử nghiệm

Mô hình thử nghiệm là mô hình client-server thông qua mạng LAN



Hình 3. 1 Mô hình client-server

Mô hình Client-Server là một kiến trúc mạng phổ biến, trong đó máy khách (client) gửi yêu cầu và máy chủ (server) xử lý rồi trả về kết quả. Máy khách thường đảm nhận giao diện người dùng, trong khi máy chủ tập trung lưu trữ dữ liệu và thực thi logic. Hai thành phần này giao tiếp với nhau thông qua các giao thức mạng như HTTP hoặc TCP/IP. Ưu điểm của mô hình là dữ liệu được quản lý tập trung, dễ mở rộng và bảo mật cao, nhưng nhược điểm là phụ thuộc nhiều vào máy chủ và chi phí vận hành cao. Mô hình này được ứng dụng rộng rãi trong các hệ thống web, email và dịch vụ trực tuyến.

## 3.2 Cài đặt hệ thống

### 3.2.1 Cấu hình máy chủ

**a. Cấu hình phần cứng**

* Bộ xử lý (CPU): Intel Xeon Silver 4310, 12 nhân, 24 luồng, tốc độ 2.10 GHz (Turbo Boost lên đến 3.30 GHz).
* Bộ nhớ (RAM): 64GB DDR4 ECC (8 x 8GB), hỗ trợ mở rộng lên đến 1TB.
* Ổ cứng (Storage): 2 x 1TB SSD NVMe cho hệ điều hành và ứng dụng, 4 x 2TB HDD SATA 7200RPM cho dữ liệu.
* Card mạng (NIC): Dual-Port 10GbE, hỗ trợ kết nối tốc độ cao.
* Nguồn điện: 750W, dự phòng, hiệu suất 80+ Platinum.

**b. Cấu hình phần mềm**

* Hệ điều hành (OS): Ubuntu Server 22.04 LTS
* Phần mềm quản lý: VMware vSphere 7 hoặc Proxmox VE 7.
* Ứng dụng: Cấu hình cho lưu trữ web (Apache/Nginx), cơ sở dữ liệu (MySQL/PostgreSQL), và quản lý tập tin (Nextcloud).

### 3.2.1 Cấu hình máy client

**a. Cấu hình phần cứng**

* Bộ xử lý (CPU): Intel Core i5-12400 6 nhân, 12 luồng.
* Bộ nhớ (RAM): 8GB DDR4 3200MHz
* Ổ cứng (Storage): 256GB SSD NVMe
* Card đồ họa (GPU): Intel UHD 730
* Màn hình: 21.5 inches Full HD (1920x1080), tấm nền AMOLED.
* Kết nối mạng (NIC): Card mạng Gigabit Ethernet, hỗ trợ Wi-Fi 5

**b. Cấu hình phần mềm**

* Hệ điều hành (OS): Windows 10 Home
* Phần mềm văn phòng: Microsoft Office 365
* Trình duyệt web: Google Chrome, Mozilla Firefox, Microsoft Edge.

## 3.3 Báo cáo kiểm thử

**- Kiểm thử các hoạt động bên phía khách hàng**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mã kiểm thử** | **Tên hoạt động** | **Các bước kiểm thử** | **Trường hợp** | **Kết quả mong muốn** | **Kết quả** |
| 1 | **KT01** | **Xem danh sách phim** | 1. Thiết lập request HTTP là GET.  2. Nhấn nút Send. | Có danh sách phim. | Hiển thị danh sách phim. | Pass |
| 2 | **KT02** |  |  | Không có phim. | Hiển thị thông báo không có phim. | Pass |
| 3 | **KT03** | **Xem chi tiết phim** | 1. Chọn một phim trong danh sách.  2. Nhấn vào xem chi tiết | Có thông tin chi tiết. | Hiển thị thông tin chi tiết của phim. | Pass |
| 4 | **KT04** |  |  | Phim không có thông tin chi tiết. | Hiển thị thông báo không có thông tin. | Pass |
| 5 | **KT05** | **Tìm kiếm phim** | 1. Nhập tên phim vào ô tìm kiếm.  2. Nhấn nút "Tìm kiếm". | Có phim tương ứng. | Hiển thị kết quả tìm kiếm. | Pass |
| 6 | **KT06** |  |  | Không có phim tương ứng. | Hiển thị thông báo không tìm thấy phim. | Pass |
| 7 | **KT07** | **Đăng ký tài khoản** | 1. Nhấn vào "Đăng ký". 2. Nhập thông tin vào form. 3. Nhấn "Đăng ký". | Nhập đủ thông tin. | Tạo tài khoản và chuyển đến trang đăng nhập. | Pass |
| 8 | **KT08** |  |  | Nhập thiếu thông tin. | Hiển thị thông báo yêu cầu nhập đủ thông tin. | Pass |
| 9 | **KT09** | **Đăng nhập** | 1. Nhấn vào "Login". 2. Nhập thông tin vào form. 3. Nhấn "Login". | Nhập đủ thông tin. | Chuyển đến trang chính đã đăng nhập. | Pass |
| 10 | **KT010** |  |  | Nhập sai thông tin. | Hiển thị thông báo lỗi và yêu cầu đăng nhập lại. | Pass |
| 11 | **KT011** | **Xem phim** | 1. Nhấn vào phim cần xem. 2. Chờ phim tải và phát. | Có phim. | Phát phim thành công. | Pass |
| 12 | **KT012** |  |  | Không có phim. | Hiển thị thông báo không có phim để xem. | Pass |
| 13 | **KT013** | **Bình luận phim** | 1. Nhấn vào phim cần xem. 2. Nhậpbình luận.  3. Nhấn "Gửi". | Đã đăng nhập. | Hiển thị bình luận mới dưới phim. | Pass |
| 14 | **KT014** |  |  | Chưa đăng nhập. | Yêu cầu đăng nhập trước khi bình luận. | Pass |

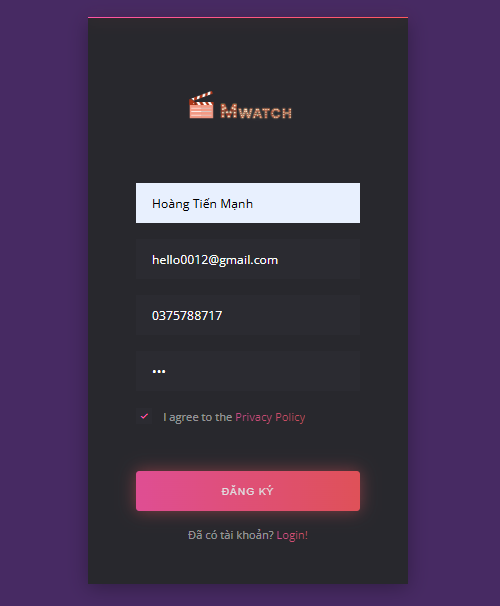
**- Kiểm thử các hoạt động bên phía quản trị viên**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mã kiểm thử** | **Tên hoạt động** | **Các bước kiểm thử** | **Trường hợp** | **Kết quả mong muốn** | **Kết quả** |
| **1** | **KT1** | **Quản lý phim** | **Thêm**  1. Nhấn "Thêm mới".  2. Nhập thông tin.  3. Nhấn "Thêm". | Nhập đủ thông tin. | Phim được thêm vào cơ sở dữ liệu. | Pass |
| **2** | **KT2** |  |  | Nhập thiếu thông tin. | Hiển thị thông báo lỗi. | Pass |
| **3** | **KT3** |  | **Sửa**  1. Nhấn "Sửa".  2. Nhập thông tin cần sửa.  3. Nhấn "Lưu". | Nhập đủ thông tin. | Cập nhật phim thành công. | Pass |
| **4** | **KT4** |  |  | Nhập thiếu thông tin. | Hiển thị thông báo lỗi. | Pass |
| **5** | **KT5** |  | **Xóa**  1. Nhấn "Xóa".  2. Xác nhận xóa. | Xác nhận xóa. | Phim bị xóa khỏi cơ sở dữ liệu. | Pass |
| **6** | **KT6** |  |  | Từ chối xóa. | Phim không bị xóa. | Pass |
| **7** | **KT7** | **Quản lý tài khoản** | **Thêm**  1. Nhấn "Thêm mới".  2. Nhập thông tin tài khoản.  3. Nhấn "Thêm". | Nhập đủ thông tin. | Tài khoản được thêm thành công. | Pass |
| **8** | **KT8** |  |  | Nhập thiếu thông tin. | Hiển thị thông báo yêu cầu nhập đủ thông tin. | Pass |
| **9** | **KT9** |  | **Sửa**  1. Nhấn "Sửa".  2. Nhập thông tin cần sửa.  3. Nhấn "Lưu". | Nhập đủ thông tin. | Cập nhật tài khoản thành công. | Pass |
| **10** | **KT10** |  |  | Nhập thiếu thông tin. | Hiển thị thông báo lỗi. | Pass |
| **11** | **KT11** | **Quản lý bình luận** | **Xóa**  1. Nhấn "Xóa" trên bình luận.  2. Xác nhận xóa. | Xác nhận xóa. | Bình luận bị xóa thành công. | Pass |

## 3.4 Kịch bản kiểm thử

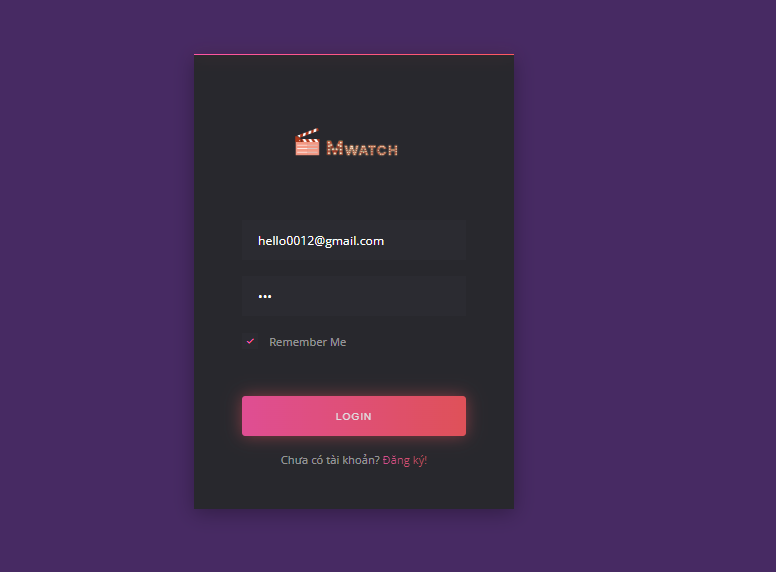
### 3.4.1 Đăng ký tài khoản (KT07)

Sau khi vào tới màn hình chức năng Đăng ký tài khoản, nhập đầy đủ thông tin Họ và tên, gmail, số điện thoại và mật khẩu đăng nhập



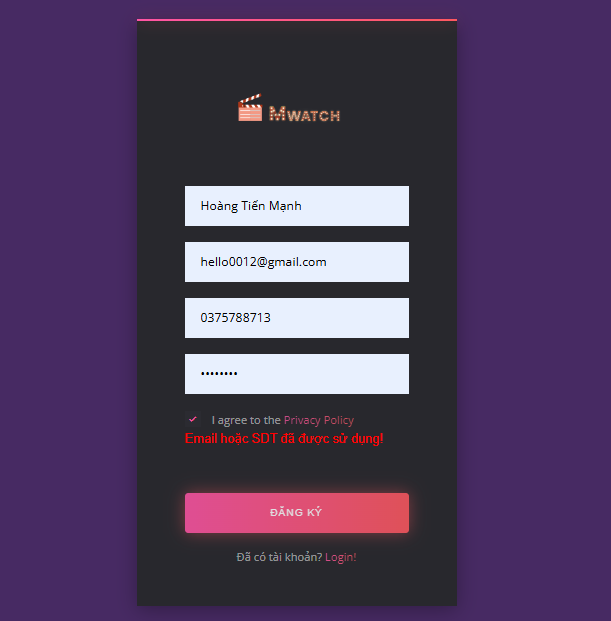
Hình 3. 2 Màn hình kiểm thử nhập thông tin đăng ký tài khoản

Nhấn vào nút đăng ký thì sẽ chuyển qua màn hình đăng nhập, chứng tỏ rằng quá trình đăng ký tài khoản đã hoàn tất và không gặp vấn đề gì.



Hình 3. 3 Màn hình kiểm thử đăng ký thành công

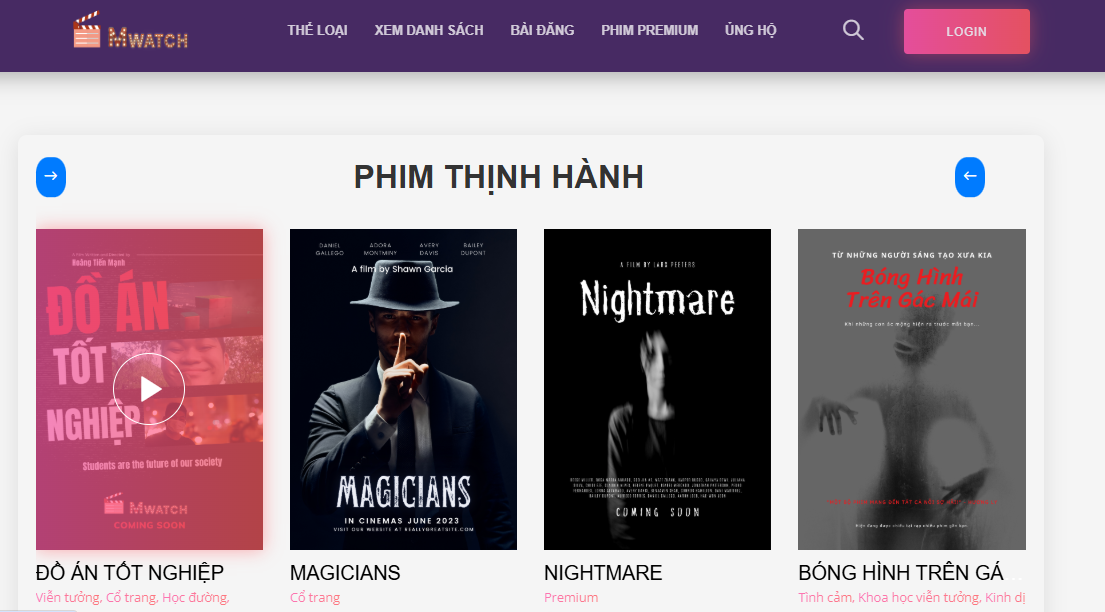
Còn nếu như thông tin số điện thoại hoặc Email đã được sử dụng thì sẽ hiện lên thông báo cho người dùng



Hình 3. 4 Màn hình kiểm thử đăng ký tài khoản thất bại

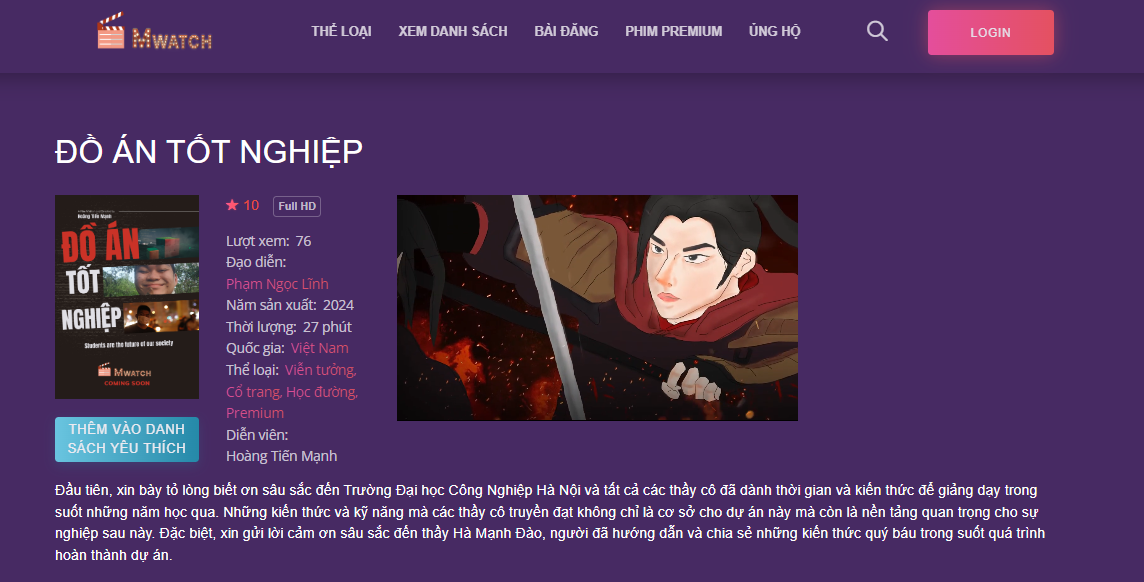
### 3.4.2 Xem chi tiết phim (KT03)

Khi người dùng chọn một phim bất kỳ trên trang chủ, nhấn vào xem chi tiết để xem thông tin chi tiết của bộ phim đó



Hình 3. 5 Màn hình kiểm thử xem chi tiết phim

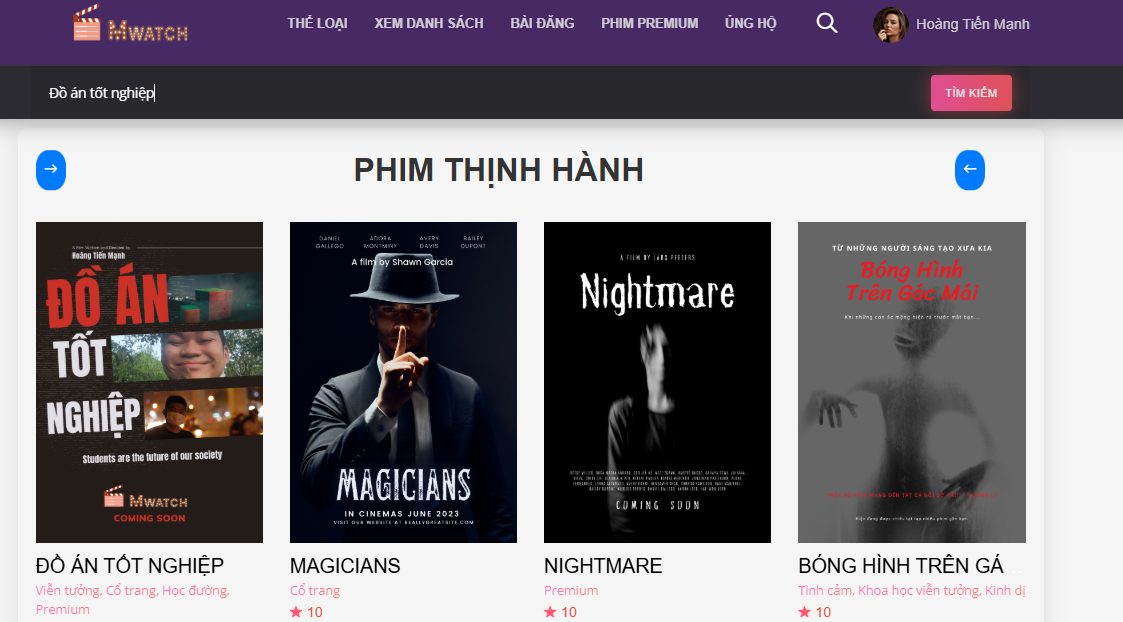
Chương trình sẽ hiện ra thông tin chi tiết của phim đó



Hình 3. 6 Màn hình kiểm thử xem chi tiết phim thành công

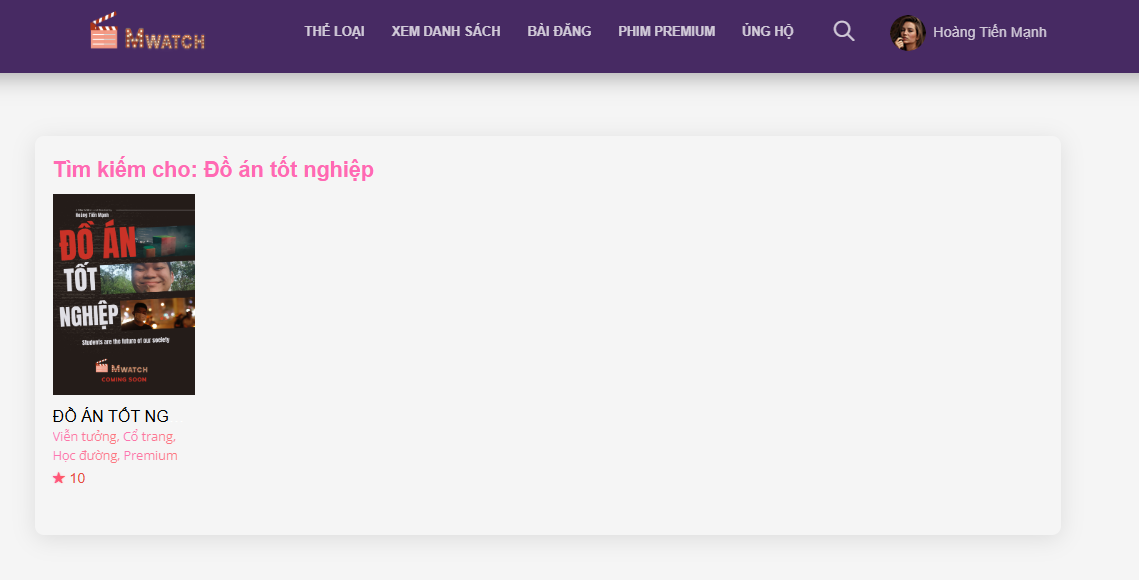
### 3.4.3 Tìm kiếm phim (KT05)

Người dùng chọn thanh công cụ tìm kiếm, tìm kiếm tên phim và chọn vào button “Tìm kiếm”



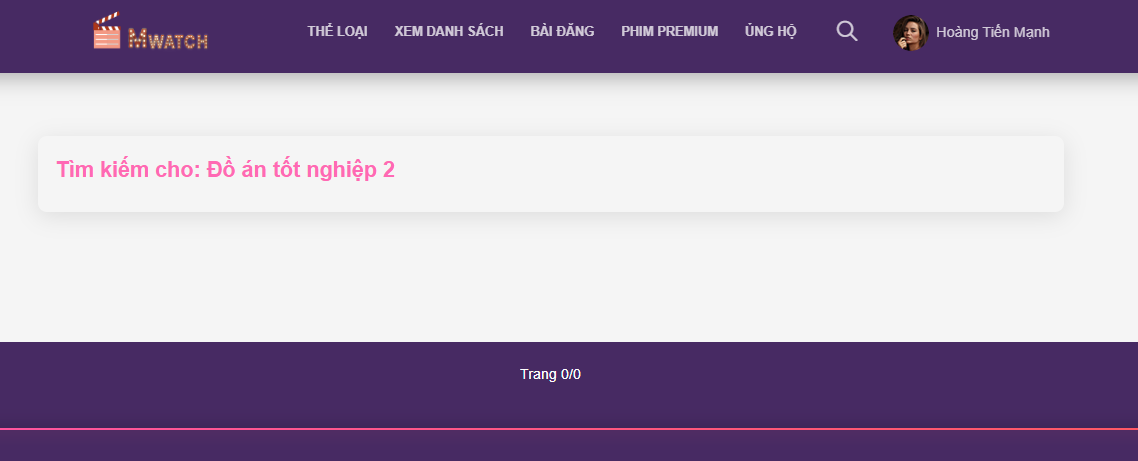
Hình 3. 7 Màn hình kiểm thử tìm kiếm phim

Chương trình hiển thị thông tin về bộ phim đó nếu bộ phim đó có trong cơ sở dữ liệu



Hình 3. 8 Màn hình kiểm thử tìm kiếm phim thành công

Nếu không có tên phim người dùng vừa nhập, sẽ không hiện thông tin phim



Hình 3. 9 Màn hình kiểm thử tìm kiếm phim thất bại

## 3.5 Đánh giá kiểm thử

Sau khi tiến hành kiểm thử toàn diện, kết quả cho thấy chương trình hoạt động đúng với các yêu cầu đã đặt ra. Tất cả các trường hợp kiểm thử (test cases) được thiết kế đều cho kết quả như mong đợi, không phát hiện lỗi hoặc vấn đề nào ảnh hưởng đến chức năng và hiệu suất. Các tính năng chính như đăng nhập, tìm kiếm, cập nhật dữ liệu, và xử lý giao dịch đều hoạt động ổn định và chính xác.

Hiệu suất hệ thống đạt mức tốt, với thời gian phản hồi nhanh và khả năng xử lý đồng thời nhiều yêu cầu người dùng mà không xảy ra hiện tượng quá tải. Đồng thời, giao diện người dùng thân thiện và dễ sử dụng, đáp ứng tiêu chuẩn thiết kế UI/UX.

# KẾT LUẬN

**Kết luận**

Đồ án tốt nghiệp được thực hiện với sự hỗ trợ từ nguồn tài liệu, mã nguồn mở và các tài nguyên liên quan, tập trung vào đề tài "Xây dựng ứng dụng web MWatch để xem phim trực tuyến sử dụng công nghệ ASP.NET MVC " Quá trình nghiên cứu, phát triển và thử nghiệm đã đạt được những kết quả quan trọng như sau:

* Hiểu rõ nhu cầu và tiềm năng phát triển của nền tảng xem phim trực tuyến trong bối cảnh hiện nay.
* Hoàn thiện việc thiết kế và triển khai giao diện người dùng thân thiện, đảm bảo khả năng tìm kiếm và xem phim theo sở thích.
* Xây dựng thành công các tính năng cốt lõi như hệ thống quản lý tài khoản người dùng, lưu trữ danh sách yêu thích, theo dõi lịch sử xem phim, và đánh giá phim.
* Nghiên cứu và tích hợp các công nghệ để đảm bảo hệ thống hoạt động ổn định, hiệu quả và có khả năng mở rộng khi nhu cầu sử dụng tăng cao.

**Kiến nghị**

Trong quá trình thực hiện, tài liệu nghiên cứu về việc xây dựng ứng dụng xem phim trực tuyến còn hạn chế, đặc biệt là trong việc cá nhân hóa trải nghiệm người dùng và tối ưu hóa hệ thống cho lưu lượng truy cập lớn. Do đó, việc xây dựng một cộng đồng chia sẻ kinh nghiệm và tài nguyên liên quan đến phát triển các nền tảng giải trí trực tuyến tại Việt Nam sẽ góp phần thúc đẩy ngành công nghệ giải trí phát triển mạnh mẽ hơn.

Ứng dụng web MWatch đã chứng minh tiềm năng ứng dụng thực tế, đáp ứng nhu cầu giải trí của người dùng. Đề tài này cần được tiếp tục hoàn thiện, mở rộng tính năng và triển khai rộng rãi hơn để trở thành một nền tảng giải trí trực tuyến uy tín và chuyên nghiệp.

## TÀI LIỆU THAM KHẢO

[1] Nguyễn Minh Đạo, Lập trình web với ASP.NET, 2014, Nhà xuất bản: Đại học Sư phạm Kỹ thuật TP.HCM

[2] Bạch Ngọc Toàn, Khóa học lập trình ASP.NET MVC, 2016

[3] Dương Quang Thiện, Lập trình Web dùng ASP.NET và C#, 2005, Nhà xuất bản Tổng hợp TP.HCM, phiên bản 5

[4] Mai Chi, Hướng dẫn tự học lập trình ADO.NET và Entity Framework, 2019