

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**  
**ĐỀ TÀI**

**NGHIÊN CỨU VÀ ỨNG DỤNG HYBRID METHOD KẾT HỢP**  
**COLLABORATIVE FILTERING VÀ CONTENT-BASED**  
**FILTERING VÀO APP XEM PHIM TRỰC TUYẾN.**

Giảng viên hướng dẫn : Ths. Phạm Thị Miên

Sinh viên thực hiện : Nguyễn Gia Ngọc

Lớp : Công nghệ thông tin

Khóa : K60

Mã số sinh viên : 6051071077

Tp. Hồ Chí Minh, năm 2024

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**  
**ĐỀ TÀI:**  
**NGHIÊN CỨU VÀ ỨNG DỤNG HYBRID METHOD KẾT HỢP**  
**COLLABORATIVE FILTERING VÀ CONTENT-BASED**  
**FILTERING VÀO APP XEM PHIM TRỰC TUYẾN.**

Giảng viên hướng dẫn : Ths. Phạm Thị Miên

Sinh viên thực hiện : Nguyễn Gia Ngọc

Lớp : Công nghệ thông tin

Khóa : K60

Mã số sinh viên : 6051071077

Tp. Hồ Chí Minh, năm 2024

\*\*\*

**NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**  
**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

\*\*\*

**Mã sinh viên:** 6051071077..... **Họ tên SV:** Nguyễn Gia Ngọc.....

**Khóa:** 60..... **Lớp:** Công nghệ thông tin.....

**1. Tên đề tài**

- Nghiên Cứu và Ứng Dụng Hybrid Method Kết Hợp Collaborative Filtering và Content-Based Filtering Vào App Xem Phim Trực Tuyến.

**2. Mục đích, yêu cầu**

**a. Mục đích**

- Xây dựng ứng dụng xem phim trực tuyến, để giải trí, xem những bộ phim hay và mới nhất.
- Hỗ trợ người dùng xem phim khi họ có thời gian với nhu cầu giải trí.

**b. Yêu cầu**

- Tìm hiểu về JAVA, Android Studio.
- Tìm hiểu về,MySQL Function
- Tìm hiểu về Nodejs
- Tìm hiểu và áp dụng các API.
- Tìm hiểu thuật toán Content-based Filtering.
- Xây dựng, thiết kế

### **3. Nội dung và phạm vi đề tài**

#### **a. Nội dung đề tài**

- Xây dựng ứng dụng xem phim gồm 2 giao diện người dùng và admin
  - Về giao diện người dùng cho phép người dùng có thể xem, tìm kiếm, đăng ký tài khoản và tiến hành xem những bộ phim yêu thích. Ngoài ra còn có gợi ý những bộ phim tương tự cho người dùng tham khảo.
  - Về giao diện admin cho phép quản lý những bộ phim, quản lý người dùng, thêm sửa xóa các bộ phim, người dùng qua CSDL.

#### **b. Phạm vi đề tài**

- Đối tượng nghiên cứu của đề tài là những người yêu thích xem phim và sử dụng các ứng dụng xem phim trực tuyến.

### **4. Công nghệ, công cụ và ngôn ngữ lập trình**

#### **a. Ngôn ngữ lập trình và công nghệ.**

- Java
- MYSQL, LARAGON
- Nodejs, TypeScripts

#### **b. Công cụ**

- Androids Studio, SDK 32, Máy ảo pixel 6 Pro API 28.
- XAMPP, MYSQL, LARAGON
- NodeJs, TypeScripts

### **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

- Hoàn chỉnh bản báo cáo đề tài.
- Xây dựng đầy đủ các chức năng cần thiết cho ứng dụng xem phim.
- Xây dựng giao diện thuận tiện cho người sử dụng.
- Nắm được quy trình nghiệp vụ của ứng dụng.
- Xây dựng thành công ứng dụng.

## **6. Giáo viên và cán bộ hướng dẫn**

Họ tên: Ths. Phạm Thị Miên

Đơn vị công tác: Bộ môn Công nghệ thông tin – Trường đại học Giao Thông Vận tải phân hiệu tại TP HCM.

Điện thoại: 0961170638

Email: ptmien@utc2.edu.vn

**Trưởng BM Công nghệ Thông tin**

**Giáo viên hướng dẫn**

**ThS. Trần Phong Nhã**

**ThS. Phạm Thị Miên**

Sinh viên: Nguyễn Gia Ngọc

Ký tên:

Điện thoại: 0346325760

Email: 6051071077@st.utc2.edu.vn

## LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi tới Quý thầy cô trong Bộ môn Công Nghệ Thông Tin, cũng như Ban Giám Hiệu Trường Đại học Giao thông Vận tải phân hiệu tại Thành Phố Hồ Chí Minh lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, nay em có thể hoàn thành đề tài tốt nghiệp **“Nghiên Cứu và Ứng Dụng Hybrid Method Kết Hợp Collaborative Filtering và Content-Based Filtering Vào App Xem Phim Trực Tuyến.”**.

Để hoàn thành nhiệm vụ được giao này, ngoài sự nỗ lực học hỏi không ngừng của bản thân còn có sự hướng dẫn tận tình của các giảng viên trong 4 năm vừa qua, đặc biệt hơn hết nhờ có giảng viên **Phạm Thị Miên**, người đã hướng dẫn cho em những hướng đi, truyền đạt cho em những kiến thức, kỹ năng để em có thể hoàn thành đề tài tốt nghiệp này.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài, nhưng chắc chắn rằng sẽ khó tránh khỏi những thiếu sót. Em rất mong nhận được những sự đánh giá, góp ý của Quý thầy cô để em có thể rút ra cho mình những bài học, kinh nghiệm quý báu.

Sau cùng, em cũng không biết nói gì hơn ngoài kính chúc Quý thầy cô trong Bộ môn Công Nghệ Thông Tin và đặc biệt là Cô **Phạm Thị Miên** thật dồi dào sức khỏe và ngày càng gặt hái được nhiều thành công hơn nữa trong cuộc sống cũng như trong sự nghiệp giảng dạy của mình.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày tháng năm 2024

**Sinh viên thực hiện**

**Nguyễn Gia Ngọc**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ..... tháng ..... năm .....

**Giảng viên hướng dẫn**

**Ths. Phạm Thị Miên**

## MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP .....	i
LỜI CẢM ƠN .....	iv
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....	v
MỤC LỤC .....	vi
DANH MỤC CHỮ VIẾT TẮT .....	x
MỤC LỤC HÌNH ẢNH .....	xi
TỔNG QUAN .....	1
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT .....	3
1.1. Giới thiệu về Java trong Android Studio .....	3
1.1.1. Lịch sử phát triển của Java trong Android Studio: .....	3
1.1.2. Đặc điểm của Android .....	3
1.2. Nội dung cơ bản về Java trong Android .....	5
1.2.1. Đặc điểm của Java trong Android Studio: .....	5
1.2.2. Ứng dụng của Java trong Android Studio .....	5
1.2.3. Khái niệm cơ bản về Java .....	6
1.2.4. Lập trình hướng đối tượng trong Java .....	7
1.2.5. Thành phần trong Android Studio sử dụng Java .....	7
1.3. Tổng quan về MySQL .....	9
1.3.1. Ưu điểm .....	10
1.3.2. Nhược điểm .....	10
1.4. Tổng quan về Node.js .....	10
1.4.1. Đặc điểm của Node.js .....	11
1.4.2. Ưu điểm của Node.js .....	11
1.4.3. Nhược điểm của Node.js .....	12



1.5. Tổng quan về API.....	12
1.5.1. Đặc điểm về API .....	12
1.5.2. Ưu điểm của API.....	13
1.5.3. Nhược điểm của API .....	13
CHƯƠNG 2. NGUYÊN CỨU HYBRID METHOD, COLLABORATIVE FILTERING VÀ CONTENT-BASED FILTERING .....	14
2.1. Giới thiệu về Content-Based Filtering .....	14
2.1.1. Khái niệm và đặc trưng của Content-Based Filtering.....	14
2.1.2. Nguyên lý hoạt động của Content-Based Filtering.....	16
2.1.3. Xây dựng mô hình và ví dụ Content Based Filtering .....	16
2.2. Giới thiệu về Collaborative Filtering .....	22
2.2.1. Định nghĩa.....	22
2.2.2. Nguyên lý hoạt động.....	22
2.2.3. Ưu điểm và nhược điểm .....	23
2.2.4. Phân tích thuật toán Collaborative Filtering.....	23
2.2.5. Ví dụ minh họa.....	23
2.3. Giới thiệu về Hybrid Method.....	24
2.3.1. Định nghĩa và mục tiêu.....	24
2.3.2. Các hình thức kết hợp Hybrid Method.....	24
2.3.3. Phân tích chi tiết các phương pháp kết hợp.....	25
2.3.4. Các bước triển khai Hybrid Method.....	25
2.3.5. Tính toán độ tương đồng.....	26
2.3.6. Tạo đề xuất.....	26
2.3.7. Ví dụ minh họa.....	26
2.3.8. Kết luận.....	26
2.4. Tìm hiểu về hệ thống đề xuất phim:.....	27

2.4.1. Giới thiệu hệ thống gợi ý phim: .....	27
2.4.2. Áp dụng vào ứng dụng xem phim.....	27
2.4.3. Lọc cộng tác dựa vào người dùng – người dùng .....	28
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	34
3.1. Khảo sát và đánh giá hệ thống .....	34
3.2. Quy trình thực hiện quản trị hệ thống của nhân viên quản trị.....	34
3.3. Yêu cầu hệ thống .....	35
3.3.1. Yêu cầu chức năng .....	35
3.3.2. Yêu cầu phi chức năng.....	35
3.3.3. Yêu cầu khác: .....	35
3.4. Sơ đồ phân rã chức năng và luồng dữ liệu .....	35
3.4.1. Yêu cầu người dùng.....	36
3.4.2. Sơ đồ phân rã chức năng.....	36
3.4.3. Sơ đồ luồng dữ liệu.....	37
3.5. Biểu đồ UseCase .....	38
3.6. Sơ đồ ER.....	41
3.7. Sơ đồ hoạt động.....	42
3.7.1. Với người dùng .....	42
3.7.2. Với người quản trị .....	49
3.8. Sơ đồ Database Diagram .....	53
3.9. Sơ đồ ClassDiagram .....	54
CHƯƠNG 4. TRIỂN KHAI ỨNG DỤNG .....	55
4.1. Tổng quan ứng dụng.....	55
4.2. Chương trình demo.....	55
4.2.1. Giao diện chức năng đăng nhập và đăng ký .....	56
4.2.2. Giao diện chức năng trang chủ .....	58

4.2.3. Giao diện tìm kiếm .....	59
4.2.4. Giao diện thông tin và chỉnh sửa thông tin người dùng .....	60
4.2.5. Giao diện thêm phim .....	61
4.2.6. Giao diện yêu thích, đánh giá, bình luận và danh sách phim yêu thích, đánh giá, bình luận .....	62
4.2.7. Giao diện đề xuất phim.....	65
KẾT LUẬN VÀ KIẾN NGHỊ .....	67
TÀI LIỆU THAM KHẢO .....	68

## DANH MỤC CHỮ VIẾT TẮT

STT	Từ viết tắt	Thuật ngữ	Ý nghĩa
1	JVM	Java Virtual Machine	Là một phần mềm đọc mã Java và chạy các chương trình Java.
3	NFC	Near Field Communication	Là một công nghệ không dây cho phép truyền tải dữ liệu giữa các thiết bị gần nhau một cách an toàn và nhanh chóng
4	SSL	Secure Sockets Layer	Là một giao thức bảo mật được sử dụng để thiết lập một kết nối mật giữa hai máy tính trên Internet
5	CDN	Content Delivery Network	Là một mạng lưới phân phối nội dung được sử dụng để cung cấp dữ liệu và tài nguyên trên Internet
6	APK	Android Package Kit	Là một định dạng tệp tin được sử dụng trên hệ điều hành Android để cài đặt và phân phối các ứng dụng cho các thiết bị Android.
7	SDK	Software Development Kit	Là một bộ công cụ phần mềm được cung cấp cho nhà phát triển
8	IDE	Integrated Development Environment	Cung cấp cho người dùng một giao diện đồ họa đơn giản và thuận tiện triển phần mềm

## MỤC LỤC HÌNH ẢNH

Hình 1 Ví dụ về utility matrix dựa trên số sao một user rate cho một item..	28
Hình 2 : Mô tả ví dụ Lạc cộng tác người dùng – người dùng..	29
Hình 3 Kết quả chạy thuật toán.	32
Hình 4 Kết quả chạy chương trình	33
Hình 5. Sơ đồ phân rã chức năng ứng dụng xem phim	36
Hình 6. Sơ đồ phân rã chức năng Quản lý ứng dụng	37
Hình 7. Sơ đồ luồng dữ liệu.....	37
Hình 8. Biểu đồ UseCase tổng quát.....	38
Hình 9. UseCase Admin	39
Hình 10. Sơ đồ UseCase người dùng	39
Hình 11. Sơ đồ Usecase đăng nhập	40
Hình 12 Sơ đồ ER.....	41
Hình 14. Sơ đồ hoạt động chức năng đăng nhập	42
Hình 15. Sơ đồ hoạt động chức năng đăng ký.....	43
Hình 16. Sơ đồ hoạt động chức năng tìm kiếm phim.....	44
Hình 17. Sơ đồ hoạt động chức năng xem chi tiết phim	45
Hình 18. Sơ đồ hoạt động chức năng đánh giá phim	46
Hình 19. Sơ đồ hoạt động chức năng bình luận phim	47
Hình 20. Sơ đồ hoạt động chức năng yêu thích phim	48
Hình 21. Sơ đồ hoạt động chức năng tìm kiếm phim.....	49
Hình 22. Sơ đồ hoạt động chức năng thêm phim	50
Hình 23. Sơ đồ hoạt động chức năng xoá phim	51
Hình 24. Sơ đồ hoạt động chức năng sửa phim.....	52
Hình 25. Sơ đồ Diagram.....	53
Hình 26. Sơ đồ ClassDiagram	54
Hình 27. Giao diện chức năng đăng nhập	56

Hình 28. Giao diện chức năng đăng ký .....	57
Hình 29. Giao diện trang chủ.....	58
Hình 30. Giao diện tìm kiếm .....	59
Hình 31. Giao diện thông tin và chỉnh sửa thông tin người dùng .....	60
Hình 32. Giao diện thêm phim .....	61
Hình 33. Giao diện yêu thích và danh sách phim yêu thích .....	62
Hình 34. Giao diện yêu đánh giá và danh sách đánh giá.....	63
Hình 35. Giao diện bình luận và danh sách phim đã bình luận.....	64
Hình 36. Giao diện đề xuất phim.....	65
Hình 37. Giao diện xem phim .....	66

# TỔNG QUAN

## 1. Tổng quan đề tài

Ngày nay với các công nghệ hiện đại đã thay đổi cuộc sống của chúng ta rất nhiều. Giờ đây với sự phổ biến của thiết bị điện tử và bùng nổ công nghệ, bất kì ai cũng có thể tìm kiếm những thông tin và phim cần thiết chỉ bằng vài thao tác tìm kiếm. Trong đó mô hình được nhiều ưa thích vì nó có thể cho phép người dùng ở bất kì đâu, chỉ cần có kết nối Internet là dễ dàng truy cập vào và sử dụng. Có rất nhiều lĩnh vực ứng dụng như là web xem phim, ứng dụng, thương mại điện tử, y tế, giáo dục và đào tạo, vui chơi giải trí,..

Ứng dụng xem phim là một ứng dụng giải trí cho phép người dùng xem các bộ phim trên điện thoại di động của mình. Cung cấp cho người dùng một thư viện phim đa dạng và có thể được sử dụng để xem các bộ phim mới nhất và cũ nhất. Ngoài ra, nhiều ứng dụng xem phim còn cung cấp cho người dùng các tính năng khác như tải xuống phim để xem ngoại tuyến, chia sẻ phim với bạn bè và gia đình và tùy chỉnh chất lượng video. Các ứng dụng xem phim phổ biến hiện nay bao gồm Netflix, Amazon Prime Video, Disney+, HBO Max và Hulu.

Trong đó đề án này, với mục đích xây dựng một hệ thống xem phim trực tuyến, tôi thiết kế để xây dựng một cho phép người dùng có thể truy cập vào để xem phim trực tuyến thông qua mạng Internet. Hệ thống được lập trình bằng ngôn ngữ Java và hệ thống QTCSDL MYSQL với những tính năng cơ bản như đăng ký người dùng, xem phim, tìm phim và các phản hồi ý kiến liên quan.

## 2. Mục tiêu nghiên cứu

- Cung cấp cho người dùng một trải nghiệm xem phim trực tuyến tốt hơn và thuận tiện hơn so với các phương thức truyền thống khác như xem phim trên đĩa DVD hoặc điện thoại di động.
- Tạo ra một nền tảng xem phim trực tuyến chất lượng cao với các tính năng như chất lượng hình ảnh và âm thanh tốt nhất có thể, tính năng tìm kiếm và lọc phim thông minh, và tính năng gợi ý phim dựa trên sở thích của người dùng.
- Phát triển các công nghệ mới để cải thiện trải nghiệm xem phim trực tuyến, bao gồm việc sử dụng trí tuệ nhân tạo để cải thiện khả năng gợi ý phim và việc sử dụng công nghệ đám mây để cải thiện khả năng lưu trữ và phân phối phim.
- Hoàn thành nghiên cứu và xây dựng thành công ứng dụng trên trước thời gian báo cáo.

### **3. Phạm vi nghiên cứu**

- Sử dụng Android Studio Xây dựng giao diện người dùng bằng Layout Editor (Trình chỉnh sửa bố cục).
- Sử dụng ngôn ngữ lập trình Java, Nodejs, TypeScript để hoàn thành ứng dụng.

### **4. Cách tiếp cận, phương pháp nghiên cứu và thực hiện**

Để có thể hiểu rõ, nắm bắt được đề tài và có cái nhìn tổng quan về vấn đề đang nghiên cứu, cần thực hiện:

- Tìm hiểu phân tích và thiết kế hướng đối tượng có cấu trúc.
- Khảo sát tình hình thực tế về nhu cầu của các người dùng khi xem phim trực tuyến và chất lượng các ứng dụng xem phim hiện nay.
- Tìm hiểu thiết kế, quy trình hoạt động của các ứng dụng đó.
- Tổng hợp các tài liệu cần thiết về đề tài.
- Xây dựng và phát triển chương trình theo một quy trình nhất định đã được vạch ra từ trước.
- Tổ chức code dễ đọc, dễ bảo trì.



# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## 1.1. Giới thiệu về Java trong Android Studio

### 1.1.1. Lịch sử phát triển của Java trong Android Studio:

Lịch sử phát triển của Java trong Android Studio được bắt đầu từ khi Google mua lại Android Inc. vào năm 2005. Khi đó, Android Inc. đã sử dụng ngôn ngữ lập trình Java để phát triển hệ điều hành Android. Sau đó, Google phát triển Android Studio là một công cụ phát triển tích hợp (IDE) cho phát triển ứng dụng Android, và Java tiếp tục được sử dụng làm ngôn ngữ lập trình chính cho phát triển ứng dụng trong Android Studio.

Trong những năm đầu, để phát triển ứng dụng Android, người lập trình cần phải sử dụng Eclipse IDE và plugin ADT (Android Development Tools) để lập trình bằng Java. Tuy nhiên, vào năm 2013, Google đã chính thức ra mắt Android Studio, công cụ phát triển tích hợp mới cho phát triển ứng dụng Android, với nhiều tính năng mới và cải tiến so với Eclipse.

Android Studio được xây dựng trên nền tảng IntelliJ IDEA của JetBrains, và hỗ trợ nhiều tính năng tiên tiến cho phát triển ứng dụng Android, bao gồm thiết kế giao diện người dùng, lập trình Java và Kotlin, gỡ lỗi, thử nghiệm và triển khai ứng dụng.

Trong quá trình phát triển, Java trong Android Studio đã trải qua nhiều cải tiến và cập nhật để tối ưu hóa hiệu suất và tính năng cho phát triển ứng dụng Android. Ngoài ra, Kotlin, một ngôn ngữ lập trình khác, cũng được hỗ trợ trong Android Studio và được xem là một sự lựa chọn thay thế cho Java trong phát triển ứng dụng Android. Tuy nhiên, Java vẫn là một ngôn ngữ lập trình chính trong Android Studio và được sử dụng rộng rãi bởi cộng đồng lập trình viên Android.

### 1.1.2. Đặc điểm của Android

Android là hệ điều hành mã nguồn mở được phát triển bởi Google và được sử dụng rộng rãi trên các thiết bị di động như điện thoại thông minh, máy tính bảng, đồng hồ thông minh, TV và các thiết bị IoT. Dưới đây là một số đặc điểm của Android:

- Mã nguồn mở: Android là một hệ điều hành mã nguồn mở, cho phép các nhà phát triển có thể tùy chỉnh và phát triển theo nhu cầu của họ.
- Đa nền tảng: Android là một hệ điều hành đa nền tảng, hỗ trợ cho nhiều loại thiết bị di động khác nhau.
- Tích hợp dịch vụ Google: Android tích hợp các dịch vụ của Google như Google

Maps, Google Drive, Google Play Store, Google Assistant, v.v. giúp người dùng dễ dàng truy cập vào các dịch vụ này.

- Hỗ trợ ứng dụng thứ ba: Android cho phép người dùng cài đặt và sử dụng các ứng dụng thứ ba bên ngoài Google Play Store.
- Tùy chỉnh giao diện người dùng: Android cho phép người dùng tùy chỉnh giao diện theo ý muốn, từ hình nền cho đến bố cục các ứng dụng.
- Hỗ trợ kết nối mạng: Android hỗ trợ nhiều loại kết nối mạng như Wi-Fi, 3G, 4G, Bluetooth, NFC, v.v.
- Hỗ trợ đa nhiệm: Android cho phép người dùng sử dụng nhiều ứng dụng cùng một lúc và chuyển đổi giữa chúng dễ dàng.
- Hỗ trợ đa phương tiện: Android hỗ trợ nhiều định dạng âm thanh, video và hình ảnh khác nhau.
- Bảo mật: Android được thiết kế để đảm bảo tính bảo mật cho người dùng và các dữ liệu của họ.
- Cộng đồng phát triển: Android có cộng đồng phát triển lớn và nhiều tài liệu hướng dẫn sẵn có, giúp các nhà phát triển dễ dàng phát triển ứng dụng trên nền tảng này.

#### ***1.1.2.1. Ưu điểm của Android Studio***

- Tích hợp đầy đủ: Android Studio tích hợp tất cả các công cụ cần thiết để phát triển ứng dụng Android, bao gồm cả bộ SDK, trình biên dịch và máy ảo.
- Tính linh hoạt: Android Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau, cho phép bạn phát triển ứng dụng bằng Java, Kotlin và nhiều ngôn ngữ khác.
- Công cụ tốt cho thiết kế giao diện: Android Studio cung cấp cho người dùng một số công cụ thiết kế giao diện tốt như Layout Editor, giúp thiết kế giao diện trở nên dễ dàng hơn.
- Được hỗ trợ bởi cộng đồng lớn: Android Studio được hỗ trợ bởi cộng đồng lớn của các nhà phát triển Android, điều này có nghĩa là bạn có thể tìm thấy hỗ trợ và tài liệu rất nhiều trên mạng.
- Hỗ trợ kiểm tra và gỡ lỗi chuyên nghiệp: Android Studio cung cấp cho người dùng nhiều công cụ để kiểm tra và gỡ lỗi ứng dụng, giúp bạn tìm ra và sửa lỗi nhanh chóng.

### ***1.1.2.2. Nhược điểm của Android Studio***

- Yêu cầu tài nguyên hệ thống cao: Android Studio là một phần mềm khá nặng và yêu cầu một hệ thống mạnh để chạy mượt mà.
- Học tập khá phức tạp: Android Studio đòi hỏi người sử dụng có kiến thức về lập trình, điều này có thể gây khó khăn cho những người mới bắt đầu phát triển ứng dụng.
- Đôi khi chậm và không ổn định: Mặc dù Android Studio đã được cải tiến rất nhiều, nhưng vẫn có thể gặp phải các vấn đề về tốc độ và ổn định.
- Không hỗ trợ tốt cho các phiên bản Android cũ: Android Studio hỗ trợ tốt nhất cho các phiên bản Android mới nhất, điều này có nghĩa là nếu bạn muốn phát triển ứng dụng cho các phiên bản cũ hơn của Android, bạn sẽ phải tìm kiếm giải pháp khác.

## **1.2. Nội dung cơ bản về Java trong Android**

### **1.2.1. Đặc điểm của Java trong Android Studio:**

- Đa nền tảng: Java là một ngôn ngữ lập trình đa nền tảng, có thể sử dụng trên nhiều hệ điều hành khác nhau bao gồm cả Android, iOS và máy tính để bàn.
- Tính đồng nhất: Java được sử dụng rộng rãi trong cộng đồng lập trình viên Android, do đó có sự đồng nhất trong cách lập trình và chia sẻ mã giữa các dự án khác nhau.
- Tính bảo mật: Java được phát triển với mục đích tăng cường tính bảo mật, đặc biệt là trong việc xử lý dữ liệu đầu vào và tiếp cận cơ sở dữ liệu.
- Tính linh hoạt: Java cung cấp nhiều tính năng linh hoạt cho phát triển ứng dụng Android, bao gồm hỗ trợ các thư viện bên thứ ba, khả năng xử lý dữ liệu theo thời gian thực và tính tương thích với các công nghệ khác.
- Tính hiệu suất: Java được tối ưu hóa để đạt hiệu suất cao, đặc biệt là trong việc xử lý dữ liệu lớn và tương tác với các cơ sở dữ liệu.
- Tính khả dụng cao: Java là một ngôn ngữ lập trình rất phổ biến và có nhiều tài liệu và cộng đồng hỗ trợ, do đó rất dễ dàng để tìm kiếm giải pháp khi gặp vấn đề trong quá trình phát triển ứng dụng Android.

### **1.2.2. Ứng dụng của Java trong Android Studio**

Java là ngôn ngữ lập trình chính được sử dụng trong Android Studio để phát triển ứng dụng Android. Dưới đây là một số ứng dụng của Java trong Android Studio:

- Xây dựng giao diện người dùng: Java được sử dụng để xây dựng các thành phần giao diện người dùng trong Android, bao gồm TextView, EditText, Button, ImageView và các thành phần khác.
- Quản lý dữ liệu: Java được sử dụng để lưu trữ, truy xuất và xử lý dữ liệu trong ứng dụng Android. Nó được sử dụng để kết nối với cơ sở dữ liệu SQLite, truy vấn dữ liệu và xử lý các tác vụ liên quan đến dữ liệu.
- Xử lý sự kiện: Java được sử dụng để xử lý các sự kiện người dùng, bao gồm cả các sự kiện cảm ứng màn hình, như chạm và vuốt trên màn hình, và các sự kiện khác như nhấn nút.
- Xử lý thông báo: Java được sử dụng để xử lý các thông báo trong ứng dụng Android, bao gồm cả các thông báo từ hệ thống và thông báo tùy chỉnh.
- Kết nối mạng: Java được sử dụng để kết nối với các dịch vụ web và truy cập các tài nguyên mạng trong ứng dụng Android.
- Tương tác với các thành phần khác: Java được sử dụng để tương tác với các thành phần khác trong Android, bao gồm Service, BroadcastReceiver .
- Xử lý ngoại lệ: Java được sử dụng để xử lý các ngoại lệ trong ứng dụng Android, bao gồm các lỗi hệ thống và các lỗi phát sinh trong quá trình thực thi ứng dụng.

### 1.2.3. Khái niệm cơ bản về Java

**Ngôn ngữ lập trình:** Java là một ngôn ngữ lập trình, được sử dụng để tạo ra các ứng dụng và phần mềm trên nhiều nền tảng khác nhau.

**Cấu trúc:** Java có cấu trúc chương trình được xây dựng trên các lớp (class) và đối tượng (object), với sự hỗ trợ của các khái niệm như kế thừa (inheritance), đa hình (polymorphism) và trừu tượng hóa (abstraction).

**Độc lập nền tảng:** Java là một ngôn ngữ độc lập nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi mã nguồn.

**Cú pháp:** Java có cú pháp đơn giản và dễ hiểu, với nhiều từ khóa và quy tắc viết code.

**Thư viện:** Java cung cấp nhiều thư viện chuẩn để hỗ trợ phát triển ứng dụng, bao gồm các thư viện đồ họa, xử lý dữ liệu và kết nối mạng.

**JVM:** Java Virtual Machine (JVM) là một máy ảo được sử dụng để chạy các ứng dụng Java trên các hệ điều hành khác nhau.

Xử lý ngoại lệ: Java có cơ chế xử lý ngoại lệ (exception handling) để quản lý các lỗi và ngoại lệ trong quá trình thực thi chương trình.

Bảo mật: Java được thiết kế để đảm bảo tính bảo mật và an toàn trong quá trình thực thi chương trình.

Hỗ trợ đa luồng: Java hỗ trợ đa luồng (multithreading), cho phép chương trình thực thi nhiều tác vụ cùng một lúc.

Phát triển ứng dụng web: Java cũng được sử dụng để phát triển các ứng dụng web, với các khái niệm như Servlet, JSP và Enterprise JavaBeans (EJB).

#### **1.2.4. Lập trình hướng đối tượng trong Java**

Lập trình hướng đối tượng (OOP) là một phương pháp lập trình phổ biến trong Java, nơi các đối tượng được tạo ra để đại diện cho các thực thể trong thế giới thực.

- Lớp (class): Lớp là một mô tả chung về một nhóm đối tượng có các thuộc tính và phương thức chung. Ví dụ, một lớp có thể đại diện cho một đối tượng như con mèo, với các thuộc tính như tên, màu sắc và tuổi.
- Đối tượng (object): Đối tượng là một thể hiện của một lớp. Ví dụ, một đối tượng con mèo có thể có các giá trị cụ thể cho các thuộc tính như tên là "Miu Miu", màu sắc là "đen" và tuổi là "3".
- Kế thừa (inheritance): Kế thừa là một khái niệm trong OOP cho phép một lớp con (subclass) kế thừa các thuộc tính và phương thức từ một lớp cha (superclass).
- Đa hình (polymorphism): Đa hình là một khái niệm trong OOP cho phép một đối tượng của một lớp có thể được sử dụng như một đối tượng của một lớp khác.
- Trừu tượng (abstraction): Trừu tượng là một khái niệm trong OOP cho phép che giấu các chi tiết cài đặt của một đối tượng và tập trung vào các tính năng của đối tượng đó.
- Giao diện (interface): Giao diện là một khái niệm trong OOP cho phép định nghĩa các phương thức mà các lớp khác có thể triển khai.

#### **1.2.5. Thành phần trong Android Studio sử dụng Java**

Activity là thành phần chính của một ứng dụng Android. Nó đại diện cho một màn hình giao diện người dùng, mà người dùng có thể tương tác để thực hiện các hành động nhất định. Activity được viết bằng Java và là một lớp Java.

Service là một thành phần được sử dụng để thực hiện các tác vụ dài hạn mà không cần tương tác với người dùng. Nó được sử dụng để thực hiện các tác vụ nền trong khi người dùng đang sử dụng các ứng dụng khác. Service được viết bằng Java và là một lớp Java.

Broadcast Receiver là một thành phần được sử dụng để nhận các thông báo từ hệ thống hoặc ứng dụng khác. Nó được sử dụng để xử lý các sự kiện hệ thống như kết nối mạng thay đổi, pin yếu và các sự kiện khác. Broadcast Receiver được viết bằng Java và là một lớp Java.

Content Provider là một thành phần được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng khác nhau. Nó cung cấp cơ chế để truy cập và thao tác với dữ liệu được lưu trữ trong cơ sở dữ liệu của ứng dụng. Content Provider được viết bằng Java và là một lớp Java.

Fragment là một thành phần được sử dụng để tạo ra một phần của giao diện người dùng trong một Activity. Nó cho phép bạn tạo ra giao diện người dùng động và linh hoạt hơn. Fragment được viết bằng Java và là một lớp Java.

Layout là một thành phần được sử dụng để định dạng giao diện người dùng. Nó đại diện cho cách các thành phần khác nhau như TextView, EditText, Button... được sắp xếp trên màn hình. Layout được viết bằng XML, tuy nhiên, Java cũng được sử dụng để thao tác với các thành phần của Layout.

View là một thành phần cơ bản của giao diện người dùng. Nó đại diện cho các thành phần như TextView, EditText, Button, CheckBox... được hiển thị trên màn hình. ViewGroup là một lớp cha của các thành phần View và được sử dụng để sắp xếp các thành phần View trên màn hình. Cả View và ViewGroup được viết bằng Java và là các lớp Java.

Style và Theme là hai thành phần được sử dụng để định dạng giao diện người dùng. Style được sử dụng để định dạng một thành phần cụ thể, trong khi Theme được sử dụng để định dạng toàn bộ ứng dụng. Cả Style và Theme được viết bằng XML.

Intent là một thành phần được sử dụng để kết nối các thành phần khác nhau của ứng dụng Android. Nó được sử dụng để chuyển dữ liệu và thực hiện các hành động nhất định. Intent được viết bằng Java và là một lớp Java.

AsyncTask và Thread là hai thành phần được sử dụng để thực hiện các tác vụ nền trong ứng dụng Android. AsyncTask được sử dụng để thực hiện các tác vụ nền nhỏ nhẹ, trong khi Thread được sử dụng để thực hiện các tác vụ nền nặng hơn và cần nhiều thời gian hơn. Cả AsyncTask và Thread đều được viết bằng Java và là các lớp Java.

#### ***1.2.5.1. Ưu điểm***

- Tính đa nền tảng: Java là một ngôn ngữ lập trình đa nền tảng, cho phép các nhà phát triển viết một lần và chạy trên nhiều nền tảng khác nhau.
- Tính bảo mật: Java có các tính năng bảo mật để giúp bảo vệ ứng dụng Android của bạn.
- Tính khả năng mở rộng: Java có tính khả năng mở rộng cao, giúp các nhà phát triển có thể thêm các tính năng mới vào ứng dụng một cách dễ dàng.
- Tính dễ học: Java là một ngôn ngữ lập trình dễ học và dễ sử dụng, đặc biệt là đối với những người mới bắt đầu học lập trình.
- Tính hiệu suất: Java có tính hiệu suất tốt, giúp ứng dụng Android chạy mượt mà và nhanh chóng.

#### ***1.2.5.2. Nhược điểm***

- Tính khả năng mở rộng: Mặc dù Java có tính khả năng mở rộng cao, nhưng cũng có thể gặp phải các vấn đề về quản lý mã nguồn khi ứng dụng phát triển.
- Tính tương thích: Một số vấn đề tương thích có thể xảy ra khi sử dụng Java trong Android, đặc biệt là đối với các phiên bản Android cũ hơn.
- Tính công kênh: Một số người cho rằng Java có tính công kênh và không linh hoạt trong việc phát triển ứng dụng Android.
- Tính tốc độ: Một số người cho rằng Java không có tốc độ cao nhất trong việc phát triển ứng dụng Android.

### **1.3. Tổng quan về MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được phát triển bởi Oracle Corporation. Nó được sử dụng rộng rãi trong các ứng dụng web và là một phần quan trọng của nhiều ứng dụng web lớn.

MySQL có thể được cài đặt trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux. Nó hỗ trợ nhiều ngôn ngữ lập trình, bao gồm PHP, Java, Python và C++, và có thể được sử dụng để lưu trữ và truy xuất các dữ liệu khác nhau, bao gồm các dữ liệu số, văn bản và hình ảnh.

### 1.3.1. Ưu điểm

- Tốc độ xử lý cao: MySQL có thể xử lý các yêu cầu cơ sở dữ liệu lớn một cách nhanh chóng và hiệu quả.
- Tính bảo mật: MySQL cung cấp các tính năng bảo mật như quản lý người dùng, mã hóa dữ liệu và nhật ký giao dịch để đảm bảo tính bảo mật và an toàn cho dữ liệu.
- Độ tin cậy cao: MySQL được thiết kế để đảm bảo tính độ tin cậy cao và có thể đáp ứng các yêu cầu của các ứng dụng web lớn.
- Tính linh hoạt: MySQL có thể được sử dụng với nhiều ngôn ngữ lập trình khác nhau và có thể được cấu hình để đáp ứng nhu cầu cụ thể của người dùng.
- Dễ sử dụng: MySQL cung cấp cho người dùng một giao diện trực quan để quản lý cơ sở dữ liệu và có thể được sử dụng một cách dễ dàng và hiệu quả.
- Miễn phí và mã nguồn mở: MySQL là miễn phí và mã nguồn mở, cho phép người dùng sửa đổi và phát triển nó theo nhu cầu của mình.

### 1.3.2. Nhược điểm

- Yêu cầu tài nguyên cao: MySQL có thể yêu cầu tài nguyên phần cứng cao để đảm bảo hiệu suất và độ tin cậy.
- Khó khăn trong việc quản lý và cấu hình: MySQL có thể khó khăn trong việc quản lý và cấu hình đối với người dùng không có kinh nghiệm.
- Không phù hợp cho các ứng dụng không quan hệ: MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, do đó nó không phù hợp cho các ứng dụng không quan hệ.
- Khả năng mở rộng hạn chế: MySQL có thể có giới hạn về khả năng mở rộng, đặc biệt là trong các ứng dụng web lớn và phức tạp.

## 1.4. Tổng quan về Node.js

Node.js là một nền tảng phát triển ứng dụng web và backend được xây dựng trên JavaScript. Nó cho phép các nhà phát triển sử dụng JavaScript để viết các ứng dụng web, thay vì phải sử dụng các ngôn ngữ khác như PHP hoặc Ruby. Node.js được phát triển bởi Ryan Dahl vào năm 2009 và nhanh chóng trở thành một trong những công nghệ phát triển ứng dụng backend phổ biến nhất hiện nay.

Node.js sử dụng mô hình xử lý không đồng bộ (asynchronous) để xử lý các yêu cầu và phản hồi (request/response) trong khi giữ cho các tiến trình khác vẫn hoạt động. Điều này làm



cho Node.js rất hiệu quả trong việc xử lý các ứng dụng web với lưu lượng truy cập lớn.

Node.js cũng cung cấp một số module tích hợp sẵn, cho phép các nhà phát triển xây dựng các ứng dụng web hoàn chỉnh với các chức năng như routing, xử lý yêu cầu HTTP, xử lý tệp, xử lý dữ liệu JSON và nhiều tính năng khác.

Node.js cũng có thể được sử dụng để xây dựng các ứng dụng desktop và mobile, nhờ các framework như Electron và React Native.

Node.js là một công nghệ phát triển ứng dụng web và backend phổ biến, được sử dụng bởi hàng ngàn các ứng dụng web và các công ty lớn trên toàn thế giới.

#### **1.4.1. Đặc điểm của Node.js**

- Single-threaded: Node.js là single-threaded, điều này cho phép nó xử lý nhiều yêu cầu đồng thời mà không tốn quá nhiều tài nguyên hệ thống.
- Asynchronous: Node.js sử dụng mô hình lập trình bất đồng bộ để tối đa hóa hiệu suất. Việc sử dụng callback hoặc promise cho phép Node.js xử lý nhiều yêu cầu mà không bị chặn bởi các hoạt động đồng bộ.
- Event-driven: Node.js sử dụng mô hình lập trình dựa trên sự kiện, cho phép các ứng dụng được xây dựng để phản ứng nhanh chóng với các sự kiện như yêu cầu HTTP, các kết nối cơ sở dữ liệu, hoặc các sự kiện khác.
- Cross-platform: Node.js được hỗ trợ trên nhiều nền tảng khác nhau bao gồm Windows, Linux và macOS.
- Lightweight and scalable: Node.js được thiết kế để là một nền tảng nhẹ và có thể mở rộng linh hoạt. Nó có thể xử lý hàng ngàn kết nối đồng thời và có thể được triển khai trên nhiều máy chủ để tạo thành một hệ thống phân tán.
- NPM: Node.js đi kèm với npm - một hệ thống quản lý gói phổ biến, cho phép các nhà phát triển dễ dàng chia sẻ và tái sử dụng mã nguồn mở.

#### **1.4.2. Ưu điểm của Node.js**

- Hiệu suất cao: Node.js được thiết kế để xử lý các yêu cầu và phản hồi trong khi giữ cho các tiến trình khác vẫn hoạt động, điều này giúp tối ưu hóa hiệu suất của ứng dụng web.
- Xử lý không đồng bộ: Node.js sử dụng mô hình xử lý không đồng bộ, cho phép xử lý đồng thời nhiều yêu cầu mà không cần đợi hoàn tất từng yêu cầu trước.

- Dễ dàng mở rộng: Node.js cho phép mở rộng dễ dàng bằng cách sử dụng các module và thư viện và cũng hỗ trợ các kiến trúc mở rộng như microservices.
- Cộng đồng phát triển lớn: Node.js có một cộng đồng phát triển lớn và năng động, với nhiều module và thư viện hữu ích được phát triển liên tục.
- Sử dụng JavaScript: Với Node.js, các nhà phát triển có thể sử dụng JavaScript để phát triển cả các ứng dụng frontend và backend, giúp giảm thời gian học và tăng tính hiệu quả.

#### **1.4.3. Nhược điểm của Node.js**

- Không thích hợp cho các ứng dụng nặng: Node.js không phù hợp cho các ứng dụng có lưu lượng truy cập lớn và yêu cầu xử lý dữ liệu nặng.
- Không hỗ trợ đa luồng: Node.js không hỗ trợ đa luồng, điều này có thể dẫn đến khó khăn trong việc xử lý các tác vụ phức tạp và yêu cầu sử dụng các module và thư viện để giải quyết vấn đề này.
- Khó khăn trong việc quản lý callback: Vì Node.js sử dụng mô hình xử lý không đồng bộ và callback, việc quản lý callback có thể gây khó khăn cho các nhà phát triển.
- Module không đồng nhất: Node.js có nhiều module và thư viện được phát triển bởi cộng đồng, điều này dẫn đến khó khăn trong việc đảm bảo tính đồng nhất và chất lượng của các module và thư viện này.

### **1.5. Tổng quan về API**

API (Application Programming Interface) là một tập hợp các quy tắc, giao thức và công nghệ được sử dụng để tương tác với một ứng dụng hoặc hệ thống khác. API cho phép các ứng dụng khác nhau trao đổi dữ liệu và chức năng với nhau một cách đơn giản và hiệu quả.

#### **1.5.1. Đặc điểm về API**

- Standardization: API thường được thiết kế để tuân thủ các chuẩn định dạng và giao thức nhất định, giúp các ứng dụng tương tác với nhau một cách dễ dàng và hiệu quả.
- Interoperability: API cho phép các ứng dụng khác nhau tương tác với nhau bất kể hệ thống hoặc ngôn ngữ lập trình được sử dụng. Vì vậy, các ứng dụng có thể được phát triển độc lập và sau đó kết hợp lại với nhau thông qua API.
- Scalability: API có thể được sử dụng để kết nối các ứng dụng với số lượng lớn người dùng hoặc đối tác. Vì vậy, nó cũng cung cấp khả năng mở rộng cao để hỗ trợ các kịch

bản tăng trưởng hoặc tải.

- Security: API cung cấp các tính năng bảo mật để đảm bảo rằng dữ liệu được truyền đi và được truy cập chỉ bởi các ứng dụng được ủy quyền và các người dùng được phép.
- Documentation and testing: API cung cấp tài liệu và công cụ kiểm thử để giúp các nhà phát triển dễ dàng sử dụng và kiểm tra tính năng của API.
- Versioning: API có thể được phát triển và cập nhật mà không ảnh hưởng đến các ứng dụng sử dụng nó bằng cách sử dụng các phiên bản khác nhau của API.

### **1.5.2. Ưu điểm của API**

- Tính linh hoạt: API cho phép các ứng dụng khác nhau tương tác với nhau một cách độc lập và linh hoạt. Các ứng dụng có thể được phát triển trên các nền tảng khác nhau và vẫn có thể sử dụng API để tương tác với nhau.
- Tính khả chuyên: API cho phép các ứng dụng khác nhau tương tác với nhau một cách hiệu quả và chính xác, giúp giảm thiểu sự nhầm lẫn và tăng tính khả chuyên của hệ thống.
- Tăng tốc độ phát triển: API cho phép các nhà phát triển đơn giản hóa quá trình tương tác giữa các ứng dụng. Điều này giúp giảm thời gian và chi phí phát triển, đồng thời tăng tốc độ triển khai ứng dụng.

### **1.5.3. Nhược điểm của API**

- Độ phức tạp: Việc phát triển và triển khai API có thể rất phức tạp và đòi hỏi kiến thức chuyên môn cao.
- Bảo mật: API có thể trở thành một điểm yếu trong hệ thống, nếu không được thiết kế và triển khai đúng cách, có thể dẫn đến các vấn đề bảo mật.
- Không ổn định: API có thể thay đổi theo thời gian, do đó các ứng dụng phải được thiết kế để có thể xử lý các thay đổi này một cách linh hoạt.

## CHƯƠNG 2. NGUYÊN CỨU HYBRID METHOD, COLLABORATIVE FILTERING VÀ CONTENT-BASED FILTERING

### 2.1. Giới thiệu về Content-Based Filtering

Content-Based Filtering là một thuật toán lọc cộng tác được sử dụng trong hệ thống khuyến nghị để đưa ra các khuyến nghị dựa trên nội dung của các mục đã được người dùng đánh giá hoặc xem trước đó. Nó dựa trên việc phân tích nội dung của các mục để đưa ra các khuyến nghị dựa trên sự tương đồng của các mục với nhau.

Content-Based Filtering áp dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để phân tích nội dung của các mục và trích xuất các đặc trưng của chúng để xây dựng mô hình dự đoán. Các đặc trưng này có thể bao gồm các thuộc tính như tác giả, thể loại, chủ đề, từ khóa, mô tả, v.v.

Content-Based Filtering thường được sử dụng trong các hệ thống khuyến nghị cho các phim, phim, sách, bài báo, v.v. Nó cho phép các hệ thống khuyến nghị đưa ra các khuyến nghị dựa trên sở thích cá nhân của người dùng và cải thiện trải nghiệm người dùng.

#### 2.1.1. Khái niệm và đặc trưng của Content-Based Filtering

##### 2.1.1.1. Khái niệm của Content-Based Filtering

Content-Based Filtering là một thuật toán lọc cộng tác trong hệ thống khuyến nghị, được sử dụng để đưa ra các khuyến nghị dựa trên nội dung của các mục đã được người dùng đánh giá hoặc xem trước đó. Nó dựa trên việc phân tích nội dung của các mục để đưa ra các khuyến nghị dựa trên sự tương đồng của các mục với nhau.

##### 2.1.1.2. Đặc trưng của Content-Based Filtering

- Phân tích nội dung: Content-Based Filtering sử dụng các kỹ thuật phân tích nội dung để trích xuất các đặc trưng của các mục, bao gồm các thuộc tính như tác giả, thể loại, chủ đề, từ khóa, mô tả, v.v.
- Đánh giá sự tương đồng: Content-Based Filtering sử dụng các phương pháp đo độ tương đồng để so sánh các mục với nhau, bao gồm phương pháp đo độ tương đồng dựa trên cosine, Euclidean distance, Pearson correlation, v.v.
- Xây dựng mô hình dự đoán: Content-Based Filtering sử dụng các đặc trưng của các mục để xây dựng mô hình dự đoán, dựa trên các thuật toán như k-Nearest Neighbors, Naive Bayes, Decision Trees, v.v.

- Tính cá nhân hóa: Content-Based Filtering cho phép tính cá nhân hóa trong khuyến nghị bằng cách đưa ra các khuyến nghị dựa trên sở thích cá nhân của người dùng.
- Độc lập với dữ liệu người dùng: Content-Based Filtering không yêu cầu thông tin về các người dùng khác nhau, chỉ cần dữ liệu về các mục trong hệ thống.
- Hạn chế trong việc đưa ra các khuyến nghị mới: Content-Based Filtering có hạn chế trong việc đưa ra các khuyến nghị mới cho người dùng, do nó chỉ đưa ra các khuyến nghị dựa trên các mục tương tự với các mục đã được người dùng đánh giá hoặc xem trước đó.

#### ***2.1.1.3. Ưu điểm của Content-Based Filtering***

- Độc lập với dữ liệu người dùng: Content-Based Filtering không yêu cầu thông tin về các người dùng khác nhau, chỉ cần dữ liệu về các mục trong hệ thống.
- Tính cá nhân hóa: Content-Based Filtering cho phép tính cá nhân hóa trong khuyến nghị bằng cách đưa ra các khuyến nghị dựa trên sở thích cá nhân của người dùng.
- Hiệu quả với các mục mới: Content-Based Filtering có thể đưa ra các khuyến nghị cho các mục mới trong hệ thống khi có đủ thông tin về các đặc trưng của chúng.
- Dễ triển khai và thực hiện: Content-Based Filtering có tính đơn giản và dễ dàng triển khai trong các hệ thống khuyến nghị.
- Không bị ảnh hưởng bởi các đánh giá sai lệch: Content-Based Filtering không bị ảnh hưởng bởi các đánh giá sai lệch của người dùng.

#### ***2.1.1.4. Nhược điểm của Content-Based Filtering***

- Hạn chế trong việc đưa ra các khuyến nghị mới: Content-Based Filtering có hạn chế trong việc đưa ra các khuyến nghị mới cho người dùng, do nó chỉ đưa ra các khuyến nghị dựa trên các mục tương tự với các mục đã được người dùng đánh giá hoặc xem trước đó.
- Cần nhiều dữ liệu và đặc trưng: Content-Based Filtering cần có nhiều dữ liệu và đặc trưng của các mục để đưa ra các khuyến nghị chính xác.
- Không hiệu quả với các mục phức tạp: Content-Based Filtering không hiệu quả với các mục có nội dung phức tạp và không thể phân tích một cách đầy đủ.
- Không đưa ra khuyến nghị xã hội: Content-Based Filtering không đưa ra khuyến nghị dựa trên các mối quan hệ xã hội giữa các người dùng.

- Khó đo lường độ tương đồng giữa các mục: Content-Based Filtering có thể gặp khó khăn trong việc đo lường độ tương đồng giữa các mục nếu chúng có nội dung phức tạp và cần phân tích chi tiết hơn.

## **2.1.2. Nguyên lý hoạt động của Content-Based Filtering**

### ***2.1.2.1. Các bước thực hiện của Content-Based Filtering***

- Tiền xử lý dữ liệu: Trước khi bắt đầu phân tích nội dung, dữ liệu cần được tiền xử lý để loại bỏ các dữ liệu không cần thiết và chuẩn hóa các dữ liệu còn lại.
- Trích xuất đặc trưng: Sau khi tiền xử lý dữ liệu, các đặc trưng của các mục cần được trích xuất. Các đặc trưng này bao gồm các thuộc tính như tác giả, thể loại, chủ đề, từ khóa, mô tả, v.v.
- Đánh giá độ tương đồng giữa các mục: Sau khi đã trích xuất các đặc trưng, các đặc trưng này cần được sử dụng để đánh giá độ tương đồng giữa các mục. Các phương pháp đánh giá độ tương đồng bao gồm phương pháp đo độ tương đồng dựa trên cosine, Euclidean distance, Pearson correlation, v.v.
- Xây dựng mô hình dự đoán: Sau khi đã đánh giá độ tương đồng giữa các mục, các đặc trưng này có thể được sử dụng để xây dựng mô hình dự đoán. Các thuật toán xây dựng mô hình dự đoán bao gồm k-Nearest Neighbors, Naive Bayes, Decision.
- Đánh giá và kiểm tra mô hình: Cuối cùng, mô hình cần được đánh giá và kiểm tra để đảm bảo rằng nó đưa ra các khuyến nghị chính xác cho người dùng. Các phương pháp đánh giá và kiểm tra bao gồm phân chia dữ liệu thành hai phần (train và test) và sử dụng các phương pháp đánh giá khác nhau như accuracy, precision, recall, F1-score, ROC curve, v.v.

## **2.1.3. Xây dựng mô hình và ví dụ Content Based Filtering**

### ***2.1.3.1. Giải thuật về thuật toán Content-Based Filtering***

Xây dựng mô hình đặc trưng: Đầu tiên, chúng ta cần xây dựng một mô hình để trích xuất các đặc trưng của các mục trong hệ thống khuyến nghị. Điều này có thể được thực hiện bằng cách sử dụng các phương pháp phân tích nội dung như xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), trích xuất thông tin (Information Retrieval - IR) và phân tích dữ liệu.

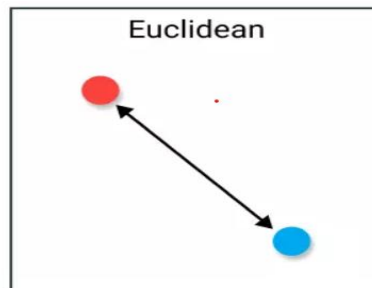
Đánh giá độ tương đồng: Tiếp theo, chúng ta cần đánh giá độ tương đồng giữa các mục trong hệ thống khuyến nghị, dựa trên các đặc trưng được trích xuất từ mô hình đặc trưng.

Tạo ra danh sách khuyến nghị: Sau khi đã đánh giá độ tương đồng giữa các mục, chúng ta có thể tạo ra danh sách các mục tương tự nhất, dựa trên sở thích của người dùng.

Đưa ra khuyến nghị: Cuối cùng, chúng ta có thể đưa ra các khuyến nghị cho người dùng, dựa trên danh sách các mục tương tự nhất được tạo ra từ bước trước đó. Các khuyến nghị này có thể được đưa ra dưới dạng danh sách các mục được sắp xếp theo độ tương đồng giữa chúng hoặc được đưa ra dưới dạng các bản ghi đề xuất.

#### 2.1.3.2. Tính toán độ tương đồng Euclid, Jaccard, Consine

**Khoảng cách Euclid:** là một khái niệm trong toán học, đo lường khoảng cách giữa hai điểm trong không gian Euclid bằng cách tính độ dài của đoạn thẳng nối hai điểm đó.



$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

- Trong không gian hai chiều, khoảng cách Euclid giữa hai điểm  $A(x_1, y_1)$  và  $B(x_2, y_2)$  được tính bằng công thức:

$$D(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.2)$$

- Trong không gian ba chiều, khoảng cách Euclid giữa hai điểm  $A(x_1, y_1, z_1)$  và  $B(x_2, y_2, z_2)$  được tính bằng công thức:

$$D(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (2.3)$$

Ví dụ, giả sử chúng ta có một cơ sở dữ liệu các phim và mỗi phim có các đặc trưng như sau

Phim 1: [1, 0, 1, 4.5]

Phim 2: [0, 1, 1, 3.0]

Phim 3: [1, 1, 0, 4.8]

Phim 4: [0, 1, 0, 2.5]

Trong đó, đặc trưng thứ nhất là thể loại hành động (1 nếu phim thuộc thể loại hành động, 0 nếu không), đặc trưng thứ hai là thể loại tình cảm, đặc trưng thứ ba là đạo diễn nam (1 nếu phim được đạo diễn bởi nam giới, 0 nếu ngược lại), và đặc trưng thứ tư là đánh giá trung bình của người dùng.

Để tính toán sự tương tự giữa các phim, ta có thể sử dụng công thức khoảng cách Euclid như đã mô tả ở trên. Ví dụ, nếu ta muốn tính toán khoảng cách Euclid giữa phim 1 và phim 2, ta sẽ có:

$$\begin{aligned}d(\text{Phim 1}, \text{Phim 2}) &= \sqrt{[(0 - 1)^2 + (1 - 0)^2 + (1 - 1)^2 + (3.0 - 4.5)^2]} \\&= \sqrt{1 + 1 + 2.25 + 2.25} \\&= \sqrt{6.5} \\&\approx 2.55\end{aligned}$$

### **Ưu điểm của Euclid**

- Dễ hiểu và tính toán: Công thức khoảng cách Euclid rất đơn giản và dễ hiểu, có thể áp dụng cho nhiều loại dữ liệu và không yêu cầu nhiều kỹ năng toán học phức tạp.
- Tính toán nhanh: Việc tính toán khoảng cách Euclid rất nhanh chóng và hiệu quả đối với các bộ dữ liệu có kích thước nhỏ đến trung bình.
- Đo đạc khoảng cách chính xác: Khoảng cách Euclid đo đạc khoảng cách chính xác giữa hai điểm trong không gian và có thể được sử dụng để tính toán các phương trình đường thẳng, đường cong và các hình học khác trong không gian hai hoặc ba chiều.

### **Nhược điểm của Euclid**

- Nhạy cảm với nhiễu: Khoảng cách Euclid không phù hợp cho các bộ dữ liệu có nhiễu hoặc giá trị bất thường (outliers), vì một giá trị bất thường có thể ảnh hưởng đến khoảng cách giữa các điểm khác.



- Không phù hợp cho các bộ dữ liệu thưa: Khoảng cách Euclid không phù hợp cho các bộ dữ liệu thưa (sparse data), nghĩa là các điểm trong không gian của dữ liệu được phân bố rải rác.
- Không phù hợp cho các bộ dữ liệu với định dạng khác nhau: Khoảng cách Euclid không phù hợp cho các bộ dữ liệu có định dạng khác nhau. Điều này có nghĩa là nếu các đặc trưng của các điểm không có thang đo chung, các khoảng cách Euclid sẽ không có ý nghĩa.

**Độ tương đồng Jaccard:** là một công thức đo lường sự tương đồng giữa hai tập hợp. Độ tương đồng Jaccard được tính bằng tỉ lệ giữa số phần tử chung của hai tập hợp và tổng số phần tử trong hai tập hợp đó. Công thức tính độ tương đồng Jaccard như sau:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3)$$

Trong đó:

- A và B là hai tập hợp cần tính độ tương đồng.
- $|A \cap B|$  là số phần tử chung của hai tập hợp A và B.
- $|A \cup B|$  là tổng số phần tử trong hai tập hợp A và B.

Ví dụ: giả sử có hai tập hợp A và B như sau:

$$A = \{1, 2, 3, 4\}$$

$$B = \{2, 4, 6, 8\}$$

Để tính độ tương đồng Jaccard giữa hai tập hợp này, ta tính số phần tử chung và tổng số phần tử như sau:

$$|A \cap B| = \{2, 4\} \Rightarrow \text{có 2 phần tử chung giữa hai tập hợp A và B.}$$

$$|A \cup B| = \{1, 2, 3, 4, 6, 8\} \Rightarrow \text{có tổng cộng 6 phần tử trong hai tập hợp A và B.}$$

Áp dụng công thức tính độ tương đồng Jaccard, ta có:

$$J(A, B) = |A \cap B| / |A \cup B| = 2 / 6 = 0.33$$

Vậy, độ tương đồng Jaccard giữa hai tập hợp A và B là 0.33.

### Ưu điểm của độ tương đồng Jaccard

- Độ tương đồng Jaccard đơn giản và dễ dàng tính toán, không yêu cầu nhiều tài nguyên tính toán.
- Thường được sử dụng trong các bài toán phân cụm và phát hiện bất thường trong dữ liệu.
- Giá trị độ tương đồng Jaccard giữa hai tập hợp nằm trong khoảng từ 0 đến 1, dễ dàng hiểu và giải thích.

### Nhược điểm của độ tương đồng Jaccard

- Độ tương đồng Jaccard không lưu lại thông tin về mức độ tương đồng giữa các phần tử của hai tập hợp. Nó chỉ cho biết mức độ tương đồng giữa hai tập hợp như một toàn thể.
- Độ tương đồng Jaccard không phù hợp trong các trường hợp tập hợp cần so sánh có kích thước lớn và không có nhiều phần tử chung. Khi đó, nó không cho kết quả chính xác về độ tương đồng giữa hai tập hợp và cần phải sử dụng các phương pháp khác để tính toán.
- Độ tương đồng Jaccard không phù hợp cho các tập hợp có sự khác biệt về kích thước một cách lớn, vì nó có thể dẫn đến sai sót trong việc so sánh độ tương đồng giữa hai tập hợp.

**Độ tương đồng của Cosine:** là một phương pháp đo độ tương đồng giữa hai vector trong không gian đa chiều. Nó được sử dụng rộng rãi trong các ứng dụng liên quan đến xử lý ngôn ngữ tự nhiên, khai phá dữ liệu và học máy để so sánh các tài liệu văn bản và tìm kiếm các mẫu trong dữ liệu.

Công thức tính độ tương đồng Cosine giữa hai vector  $x$  và  $y$  được tính như sau:

$$\text{Cos}(x, y) = \frac{(x, y)}{(|x| |y|)} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4)$$

Trong đó:

- $x$  và  $y$  là hai vector cần tính độ tương đồng.
- $(x \cdot y)$  là tích vô hướng của hai vector  $x$  và  $y$ .
- $|x|$  và  $|y|$  là độ dài của hai vector  $x$  và  $y$ .

Giá trị độ tương đồng Cosine dao động từ -1 đến 1, với 1 nghĩa là hai vector hoàn toàn giống nhau và -1 nghĩa là hai vector hoàn toàn trái ngược nhau. Giá trị 0 nghĩa là hai vector không có tương đồng với nhau.

### **Ưu điểm của độ tương đồng Cosine**

- Độ tương đồng Cosine tính toán nhanh và hiệu quả với những vector có số chiều lớn.
- Nó không bị ảnh hưởng bởi kích thước của vector.
- Nó không yêu cầu các vector phải có cùng số chiều.

### **Nhược điểm của độ tương đồng Cosine**

- Nó không giải quyết được vấn đề về tần suất xuất hiện của các thuộc tính.
- Nó không xem xét mối quan hệ giữa các chiều trong không gian vector.

### **Ví dụ về thuật toán Content-Based Filtering**

Giả sử chúng ta có một hệ thống khuyến nghị phim sử dụng Content-Based Filtering để đưa ra các khuyến nghị dựa trên đặc trưng của các phim. Các đặc trưng của các phim bao gồm thể loại, diễn viên, đạo diễn, năm sản xuất, mô tả v.v.

Để xây dựng mô hình Content-Based Filtering, chúng ta sử dụng các phương pháp phân tích nội dung để trích xuất các đặc trưng của các phim và đánh giá độ tương đồng giữa các phim. Chẳng hạn, chúng ta có thể sử dụng phương pháp đo độ tương đồng cosine để so sánh đặc trưng của các phim.

Sau khi đã xây dựng mô hình, chúng ta có thể đưa ra các khuyến nghị dựa trên sở thích của người dùng. Ví dụ, nếu người dùng đã đánh giá cao các phim thuộc thể loại hành động, chúng ta có thể đưa ra các khuyến nghị về các phim thuộc thể loại tương tự. Nếu người dùng đã xem các phim của một diễn viên cụ thể, chúng ta có thể đưa ra các khuyến nghị về các phim có sự tham gia của diễn viên đó.

### **Kết luận**

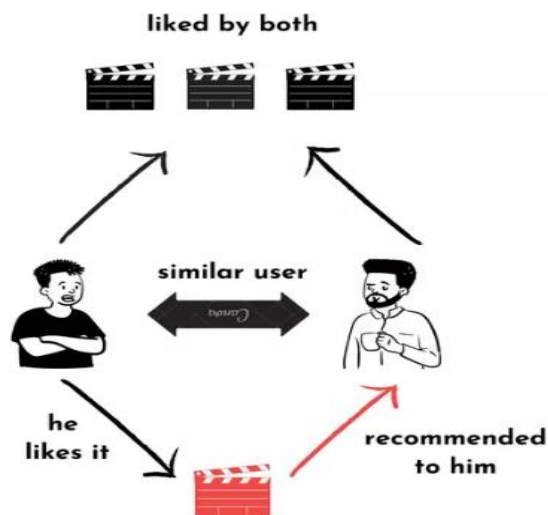
Content-based filtering là một phương pháp lọc dựa trên nội dung của các phim để đưa ra các gợi ý phim tương đồng. Phương pháp này đánh giá độ tương đồng giữa các phim bằng cách so sánh các thuộc tính của chúng, như mô tả phim, thể loại, tác giả, diễn viên, đạo diễn, v.v. từ đó đưa ra các đề xuất cho người dùng.

- Giới hạn của phương pháp này là chỉ đề xuất các phim tương đồng với những phim mà người dùng đã xem hoặc mua trước đó. Nó không đưa ra các đề xuất mới hoàn toàn.
- Nó không xem xét các thông tin liên quan đến người dùng, như lịch sử tìm kiếm, lịch sử mua hàng, sở thích cá nhân, v.v. Do đó, nó có thể không đưa ra các đề xuất phù hợp với sở thích của người dùng.
- Nếu các phim không có các thuộc tính phù hợp hoặc các thuộc tính quá giống nhau, phương pháp này có thể không đưa ra các đề xuất chính xác.

## 2.2. Giới thiệu về Collaborative Filtering

### 2.2.1. Định nghĩa

Collaborative Filtering là phương pháp khuyến nghị dựa trên việc khai thác sở thích và đánh giá của nhiều người dùng khác nhau để đưa ra đề xuất. Phương pháp này giả định rằng những người dùng có sở thích tương tự sẽ có xu hướng thích những mục giống nhau.



### 2.2.2. Nguyên lý hoạt động

Collaborative Filtering hoạt động dựa trên việc thu thập và phân tích các đánh giá hoặc tương tác của người dùng với các mục. Có hai dạng chính:

- User-based Collaborative Filtering: Dựa trên việc tìm kiếm người dùng có sở thích tương tự để đưa ra đề xuất.
- Item-based Collaborative Filtering: Dựa trên việc tìm kiếm các mục tương tự dựa trên đánh giá của người dùng để đưa ra đề xuất.

### **2.2.3. Ưu điểm và nhược điểm**

#### ***2.2.3.1. Ưu điểm***

- Không cần thông tin chi tiết về các mục.
- Khả năng khám phá và đưa ra các đề xuất bất ngờ.
- Tính cá nhân hóa cao khi có đủ dữ liệu người dùng.

#### ***2.2.3.2. Nhược điểm***

- Vấn đề "khởi đầu lạnh" (cold start) khi không có đủ dữ liệu về người dùng hoặc mục.
- Độ phức tạp tính toán cao khi số lượng người dùng và mục tăng.
- Khó khăn trong việc xử lý dữ liệu thưa (sparse data).

### **2.2.4. Phân tích thuật toán Collaborative Filtering**

#### ***2.2.4.1. Các bước chính***

- Thu thập dữ liệu: Thu thập thông tin về đánh giá hoặc tương tác của người dùng với các mục.
- Xử lý dữ liệu: Làm sạch và chuẩn hóa dữ liệu.
- Tính toán độ tương đồng: Sử dụng các phương pháp như cosine similarity hoặc Pearson correlation để đo lường độ tương đồng giữa người dùng hoặc các mục.
- Tạo đề xuất: Dựa trên độ tương đồng, đề xuất các mục phù hợp với người dùng.

#### ***2.2.4.2. Các phương pháp tính toán độ tương đồng:***

- Cosine Similarity: Đo lường độ tương đồng giữa hai vector dựa trên góc giữa chúng.
- Pearson Correlation: Đo lường độ tương đồng dựa trên sự tương quan tuyến tính giữa hai tập dữ liệu.

### **2.2.5. Ví dụ minh họa**

Giả sử một người dùng thích các bộ phim hành động và phiêu lưu. Hệ thống User-based Collaborative Filtering sẽ tìm những người dùng khác có sở thích tương tự và đề xuất các bộ phim mà họ đã thích. Trong khi đó, Item-based Collaborative Filtering sẽ tìm các bộ phim hành động và phiêu lưu mà người dùng khác có xu hướng thích và đề xuất chúng.

## **2.3. Giới thiệu về Hybrid Method**

### **2.3.1. Định nghĩa và mục tiêu**

Hybrid Method là phương pháp khuyến nghị tích hợp các ưu điểm của cả hai thuật toán để cải thiện độ chính xác và chất lượng của các đề xuất. Mục tiêu chính của Hybrid Method là cung cấp các đề xuất cá nhân hóa tốt hơn bằng cách tận dụng thông tin về sở thích của người dùng và đặc điểm của các mục (items).

### **2.3.2. Các hình thức kết hợp Hybrid Method**

#### ***2.3.2.1. Weighted Hybrid***

Weighted Hybrid kết hợp đầu ra của các phương pháp khác nhau với các trọng số cụ thể. Kết quả cuối cùng được tạo ra bằng cách tính toán tổng trọng số của các kết quả từ Collaborative Filtering và Content-Based Filtering. Ví dụ, nếu Collaborative Filtering và Content-Based Filtering được gán trọng số 0.5, hệ thống sẽ kết hợp kết quả của cả hai phương pháp với trọng số này để đưa ra đề xuất cuối cùng.

#### ***2.3.2.2. Switching Hybrid***

Switching Hybrid chuyển đổi giữa các phương pháp dựa trên ngữ cảnh hoặc loại dữ liệu hiện có. Hệ thống có thể sử dụng Content-Based Filtering cho người dùng mới đăng ký để giải quyết vấn đề "khởi đầu lạnh" và sau đó chuyển sang Collaborative Filtering khi có đủ dữ liệu về người dùng.

#### ***2.3.2.3. Mixed Hybrid***

Mixed Hybrid đưa ra đề xuất từ cả hai phương pháp đồng thời và hiển thị tất cả cho người dùng. Điều này cho phép người dùng có nhiều lựa chọn hơn và khả năng khám phá cao hơn. Kết quả của cả Collaborative Filtering và Content-Based Filtering được hiển thị song song.

#### ***2.3.2.4. Feature Combination Hybrid***

Feature Combination Hybrid kết hợp các đặc điểm từ cả hai phương pháp trong một mô hình học máy duy nhất. Hồ sơ người dùng có thể bao gồm các đặc điểm từ cả Content-Based Filtering và Collaborative Filtering để tạo ra đề xuất chính xác hơn.

#### ***2.3.2.5. Cascade Hybrid***

Cascade Hybrid sử dụng một phương pháp để lọc hoặc đánh giá ban đầu, sau đó sử dụng phương pháp khác để tinh chỉnh các đề xuất. Ví dụ, Content-Based Filtering để tạo danh sách sơ bộ các mục, sau đó sử dụng Collaborative Filtering tạo ra các đề xuất cuối cùng.

### 2.3.3. Phân tích chi tiết các phương pháp kết hợp

#### 2.3.3.1. Ưu điểm của Hybrid Method

- **Cải thiện độ chính xác:** Kết hợp các phương pháp giúp tăng cường độ chính xác của các đề xuất bằng cách tận dụng thông tin từ nhiều nguồn khác nhau.
- **Giảm thiểu vấn đề khởi đầu lạnh:** Việc kết hợp Content-Based Filtering có thể giúp hệ thống hoạt động hiệu quả hơn ngay cả khi thiếu dữ liệu người dùng.
- **Khả năng khám phá:** Hybrid Method giúp người dùng khám phá các mục mới ngoài sở thích hiện tại của họ.
- **Tính cá nhân hóa cao:** Sự kết hợp các thuật toán giúp tạo ra các đề xuất được cá nhân hóa tốt hơn.

#### 2.3.3.2. Nhược điểm của Hybrid Method

- **Độ phức tạp cao:** Việc kết hợp các thuật toán đòi hỏi phải xử lý và tối ưu hóa nhiều biến số, điều này làm tăng độ phức tạp của hệ thống.
- **Yêu cầu tài nguyên tính toán lớn:** Việc tính toán và duy trì hai hoặc nhiều mô hình đồng thời yêu cầu tài nguyên tính toán lớn.
- **Khó khăn trong việc tối ưu hóa:** Tối ưu hóa cách kết hợp giữa các thuật toán để đạt được hiệu quả tốt nhất có thể là một thách thức.

### 2.3.4. Các bước triển khai Hybrid Method

#### 2.3.4.1. Thu thập dữ liệu

- **Thu thập thông tin về các mục:** Bao gồm các đặc điểm, thuộc tính của mục.
- **Thu thập dữ liệu người dùng:** Bao gồm lịch sử tương tác, đánh giá của người dùng.

#### 2.3.4.2. Xử lý dữ liệu

- **Làm sạch và chuẩn hóa dữ liệu:** Đảm bảo dữ liệu đủ tin cậy và phù hợp cho việc phân tích.
- **Tích hợp dữ liệu:** Kết hợp dữ liệu từ nhiều nguồn khác nhau để tạo ra một tập dữ liệu duy nhất.

#### 2.3.4.3. Mô hình hóa

- **Tạo hồ sơ người dùng và mục:** Sử dụng các kỹ thuật Content-Based Filtering để tạo hồ sơ.
- **Tính toán độ tương đồng:** Sử dụng các kỹ thuật Collaborative Filtering để tính toán độ tương đồng giữa người dùng hoặc giữa các mục.

#### 2.3.5. Tính toán độ tương đồng

- **Cosine Similarity:** Đo lường độ tương đồng giữa hai vector dựa trên góc giữa chúng.
- **Pearson Correlation:** Đo lường độ tương đồng dựa trên sự tương quan tuyến tính giữa hai tập dữ liệu.

#### 2.3.6. Tạo đề xuất

- **Kết hợp các phương pháp:** Sử dụng một trong các hình thức kết hợp để tạo ra đề xuất cuối cùng.

#### 2.3.7. Ví dụ minh họa

Giả sử một người dùng mới đăng ký vào một dịch vụ streaming phim. Vì không có dữ liệu lịch sử về người dùng này (vấn đề "khởi đầu lạnh"), hệ thống có thể sử dụng Content-Based Filtering để đề xuất các phim dựa trên thông tin cá nhân (như thể loại yêu thích được khai báo khi đăng ký). Sau khi người dùng bắt đầu xem và đánh giá phim, hệ thống sẽ dần dần chuyển sang sử dụng Collaborative Filtering để cải thiện độ chính xác của các đề xuất.

#### 2.3.8. Kết luận

Hybrid Method một cách tiếp cận mạnh mẽ trong việc phát triển hệ thống khuyến nghị, tận dụng được ưu điểm của cả Collaborative Filtering và Content-Based Filtering. Phương pháp này không chỉ cải thiện độ chính xác và chất lượng của các đề xuất mà còn giúp hệ thống khuyến nghị thích ứng tốt hơn với các tình huống và dữ liệu khác nhau. Việc hiểu rõ và áp dụng đúng cách các kỹ thuật kết hợp này là yếu tố quan trọng để xây dựng các hệ thống khuyến nghị hiệu quả và đáp ứng tốt nhu cầu của người dùng.



## 2.4. Tìm hiểu về hệ thống đề xuất phim:

### 2.4.1. Giới thiệu hệ thống gợi ý phim:

Hệ thống gợi ý (Recommendation Systems) được định nghĩa là hệ thống đề xuất, gợi ý các phim phù hợp nhất với người dùng cụ thể bằng cách dự đoán mối quan tâm của người dùng về một mặt hàng dựa trên thông tin liên quan đến phim, người dùng và sự tương tác giữa chúng. Có hai đối tượng chính trong hệ thống gợi ý: users – người dùng và items – phim. Mục đích chính của hệ thống gợi ý là dự đoán mức độ quan tâm của một người dùng tới một item nào đó dựa trên hành vi của người dùng này và các người dùng khác qua đó để xây dựng hệ khuyến nghị cá nhân.

Các hệ thống gợi ý thường được chia thành hai nhóm lớn:

- **Lọc theo nội dung (Content-based systems):** Đánh giá đặc tính của phim được gợi ý. Ví dụ: một người dùng xem rất nhiều các sách văn học, vậy thì gợi ý một quyển sách trong cơ sở dữ liệu có chung đặc tính văn học tới người dùng này, ví dụ sách Mất Biếc. Cách tiếp cận này yêu cầu việc sắp xếp các phim vào từng nhóm hoặc đi tìm các đặc trưng của từng phim. Tuy nhiên, có những phim không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng phim đôi khi là bất khả thi.
- **Lọc cộng tác (Collaborative filtering):** Hệ thống gợi ý phim dựa trên sự tương quan (similarity) giữa các người dùng và hoặc các phim. Có thể hiểu rằng ở nhóm này một phim được gợi ý tới một người dùng dựa trên những người dùng có hành vi tương tự. Ví dụ: người dùng A, B, C đều thích các sách của Nguyễn Ngọc Ánh. Ngoài ra, hệ thống biết rằng người dùng B, C cũng thích các sách của Tony Buổi sáng nhưng chưa có thông tin về việc liệu người dùng A có thích sách của Tony Buổi sáng hay không. Dựa trên thông tin của những người dùng tương tự là B và C, hệ thống có thể dự đoán rằng A cũng thích sách của Tony Buổi sáng và gợi ý các quyển sách của tác giả này tới A.

### 2.4.2. Áp dụng vào ứng dụng xem phim

Trong đề tài này, nhóm tác giả sử dụng Neighborhood-based Collaborative Filtering (NBCF) và cụ thể hơn là User-user Collaborative Filtering dùng để xây dựng hệ thống gợi ý sách cho app. Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một người dùng tới một bộ phim dựa trên các sự tương quan với các người dùng khác. Việc gần giống nhau giữa các người dùng có thể được xác định thông qua mức độ quan tâm của các người dùng này tới các phim khác mà hệ thống đã biết. Ví dụ, A, B được đánh giá là tương quan với nhau.

Ta đã biết A cũng thích phim ” MAI”, vậy nhiều khả năng B cũng thích phim này.

### 2.4.3. Lọc cộng tác dựa vào người dùng – người dùng

#### 2.4.3.1. Mức độ tương đồng (Similarity functions)

Để giải quyết vấn đề đầu tiên, tức là xác định được độ giống nhau giữa các người dùng. Như đã đề cập ở trên, có hai thực thể chính ở trong hệ thống gợi ý là người dùng và phim, mỗi người dùng sẽ có một mức độ quan tâm, tạm gọi là điểm đánh giá (rating), tới từng bộ phim khác nhau. Tập hợp các đánh giá bao gồm các giá trị chưa biết cần được dự đoán, tạo nên một ma trận gọi là Utility Matrix. Trong đề tài này, ta gọi nó là Utility matrix Y, vậy nên sự giống nhau này phải được xác định dựa trên các cột tương ứng với hai người dùng trong ma trận này. Xét ví dụ trong Hình1. Ở ví dụ dưới đây  $u_0$  đến  $u_6$  tượng trưng cho người dùng,  $i_0$  đến  $i_4$  tượng trưng cho các sách.

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	5	5	2	0	1	?	?
$i_1$	3	?	?	0	?	?	?
$i_2$	?	4	1	?	?	1	2
$i_3$	2	2	3	4	4	?	4
$i_4$	2	0	4	?	?	?	5

Hình 1 Ví dụ về utility matrix dựa trên số sao một user rate cho một item. Một cách trực quan, hành vi của  $u_0$  giống với  $u_1$  hơn là,  $u_2, u_3, u_4, u_5$ . Từ đó có thể dự đoán rằng  $u_0$  sẽ quan tâm với  $i_2$  vì  $u_1$  cũng quan tâm đến với item này.

Giả sử có các người dùng từ  $u_0$  đến  $u_6$  và các phim từ  $i_0$  đến  $i_4$  trong đó các số trong mỗi ô vuông thể hiện số điểm mà mỗi người dùng đã đánh giá cho phim với giá trị càng cao thì thể hiện người dùng càng thích phim. Các dấu chấm hỏi là các giá trị chưa biết, tức là những giá trị mà hệ thống gợi ý phải đi tìm. Gọi độ giống nhau giữa hai người dùng  $u_i, u_j$  là  $\text{sim}(u_i, u_j)$ . Xây dựng app xem phim và gợi ý phim cho người dùng.

Đầu tiên chúng ta có thể nhận thấy là các  $u_0, u_1$  có vẻ thích phim  $i_0, i_1, i_2$  và không thích phim  $i_3, i_4$ , qua đánh giá ban đầu thì ta có thể cho rằng  $u_0$  và  $u_1$  có vẻ giống nhau và khác với các người dùng còn lại. Điều ngược lại xảy ra ở các người dùng còn lại. Vì vậy, để bài toán có mức chính xác cao nhất, thì ta cần lựa chọn sim cao nhất. Ở đây người dùng giống với  $u_0$  nhất là  $u_1$ :

$$\text{sim}(u_0, u_1) > \text{sim}(u_0, u_i), \forall i > 1 \quad (2.1)$$

Từ đó, để xác định mức độ quan tâm của  $u_0$  lên  $i_2$ , chúng ta nên dựa trên hành vi của  $u_1$  lên bộ phim này. Vì rằng  $u_1$  đã thích  $i_2$  nên hệ thống cần gợi ý  $i_2$  cho  $u_0$ .

Để đo độ tương tự giữa hai người dùng, cách thường làm là xây dựng vector cho mỗi người dùng rồi áp dụng một hàm có khả năng đo độ tương tự giữa hai vectors đó. Với mỗi người dùng, thông tin duy nhất chúng ta biết là các điểm đánh giá mà người dùng đó đã thực hiện, tức cột tương ứng với người dùng đó trong Utility matrix. Tuy nhiên, khó khăn là các cột này thường có rất nhiều giá trị trống vì mỗi người dùng thường chỉ đánh giá một số lượng rất nhỏ các bộ phim. Cách khắc phục là bằng cách nào đó, ta giúp hệ thống điền các giá trị này sao cho việc điền không làm ảnh hưởng nhiều tới sự giống nhau giữa hai vector. Việc điền này chỉ phục vụ cho việc tính độ tương tự chứ không phải là suy luận ra giá trị cuối cùng. Để giải quyết được vấn đề này, ta cần chuẩn hóa dữ liệu ma trận, nội dung được trình bày ở phần tiếp theo.

### 2.4.3.2. Chuẩn hóa dữ liệu

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	5	5	2	0	1	?	?
$i_1$	4	?	?	0	?	2	?
$i_2$	?	4	1	?	?	1	1
$i_3$	2	2	3	4	4	?	4
$i_4$	2	0	4	?	?	?	5
	↓	↓	↓	↓	↓	↓	↓
$\bar{u}_j$	3.25	2.75	2.5	1.33	2.5	1.5	3.33

a) Original utility matrix  $\mathbf{Y}$  and mean user ratings.

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	1.75	2.25	-0.5	-1.33	-1.5	0	0
$i_1$	0.75	0	0	-1.33	0	0.5	0
$i_2$	0	1.25	-1.5	0	0	-0.5	-2.33
$i_3$	-1.25	-0.75	0.5	2.67	1.5	0	0.67
$i_4$	-1.25	-2.75	1.5	0	0	0	1.67

b) Normalized utility matrix  $\bar{\mathbf{Y}}$ .

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$u_0$	1	0.83	-0.58	-0.79	-0.82	0.2	-0.38
$u_1$	0.83	1	-0.87	-0.40	-0.55	-0.23	-0.71
$u_2$	-0.58	-0.87	1	0.27	0.32	0.47	0.96
$u_3$	-0.79	-0.40	0.27	1	0.87	-0.29	0.18
$u_4$	-0.82	-0.55	0.32	0.87	1	0	0.16
$u_5$	0.2	-0.23	0.47	-0.29	0	1	0.56
$u_6$	-0.38	-0.71	0.96	0.18	0.16	0.56	1

c) User similarity matrix  $\mathbf{S}$ .

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	1.75	2.25	-0.5	-1.33	-1.5	0.18	-0.63
$i_1$	0.75	0.48	-0.17	-1.33	-1.33	0.5	0.05
$i_2$	0.91	1.25	-1.5	-1.84	-1.78	-0.5	-2.33
$i_3$	-1.25	-0.75	0.5	2.67	1.5	0.59	0.67
$i_4$	-1.25	-2.75	1.5	1.57	1.56	1.59	1.67

d)  $\hat{\mathbf{Y}}$

Predict normalized rating of  $u_1$  on  $i_1$  with  $k = 2$

Users who rated  $i_1$  :  $\{u_0, u_3, u_5\}$

Corresponding similarities:  $\{0.83, -0.40, -0.23\}$

$\Rightarrow$  most similar users:  $\mathcal{N}(u_1, i_1) = \{u_0, u_5\}$

with **normalized ratings**  $\{0.75, 0.5\}$

$$\Rightarrow \hat{g}_{i_1, u_1} = \frac{0.83 \cdot 0.75 + (-0.23) \cdot 0.5}{0.83 + |-0.23|} \approx 0.48$$

e) Example

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	5	5	2	0	1	1.68	2.70
$i_1$	4	3.23	2.33	0	1.67	2	3.38
$i_2$	4.15	4	1	-0.5	0.71	1	1
$i_3$	2	2	3	4	4	2.10	4
$i_4$	2	0	4	2.9	4.06	3.10	5

f) Full  $\mathbf{Y}$

Hình 2 : Mô tả ví dụ Lọc cộng tác người dùng – người dùng. Ví dụ mô tả User-user Collaborative Filtering. a) Utility Matrix ban đầu. b) Utility Matrix đã được chuẩn hoá. c) User similarity matrix.

d) Dự đoán các (normalized) ratings còn thiếu. e) Ví dụ về cách dự đoán normalized rating của  $u_1$  cho  $i_1$ . f) Dự đoán các (denormalized) ratings còn thiếu.

Để giải quyết vấn đề chuẩn hóa dữ liệu, ta cần tính giá trị trung bình điểm đánh giá của các người dùng. Hàng cuối cùng trong Hình 2 a) là giá trị trung bình của điểm đánh giá cho mỗi người dùng. Giá trị cao tương ứng với các người dùng dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi điểm đánh giá đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được ma trận chuẩn hóa (normalized utility matrix) như trong Hình 2 b). Bước chuẩn hóa này có chức năng quan trọng vì:

- Việc trừ đi trung bình cộng của mỗi cột khiến cho trong mỗi cột có những giá trị dương và âm. Những giá trị dương tương ứng với việc người dùng thích bộ phim, ngược lại với những giá trị âm. Những giá trị bằng 0 tương ứng với việc chưa xác định được liệu người dùng có thích phim hay không.

- Về mặt kỹ thuật, số cột và hàng của utility matrix là rất lớn với hàng triệu người dùng và phim, nếu lưu toàn bộ các giá trị này trong một ma trận thì khả năng cao là sẽ không đủ bộ nhớ. Quan sát thấy rằng vì số lượng đánh giá biết trước thường là một số rất nhỏ so với kích thước của utility matrix, sẽ tốt hơn nếu chúng ta lưu ma trận này dưới dạng ma trận thưa (sparse matrix), tức chỉ lưu các giá trị khác không và vị trí của chúng. Vì vậy, tốt hơn hết, các dấu ‘?’ nên được thay bằng giá trị ‘0’, tức chưa xác định liệu người dùng có thích phim hay không. Việc này không những tối ưu bộ nhớ mà việc tính toán ma trận tương tự (similarity matrix) sau này cũng hiệu quả hơn. Sau khi đã chuẩn hoá dữ liệu như trên, ở đề tài này, nhóm tác giả sử dụng hàm tính độ tương tự là hàm Cosine Similarity.

$$\text{consine\_Similarity}(u_1, u_2) = \cos(u_1, u_2) = \frac{u_1^T \cdot u_2}{\|u_1\|_2 \cdot \|u_2\|_2} \quad (2.2)$$

Trong đó  $u_1, u_2$  là vectors tương ứng với người dùng 1, 2 đã được chuẩn hoá như ở trên. Độ tương tự của hai vector là 1 số trong đoạn  $[-1, 1]$ . Giá trị bằng 1 thể hiện hai vector hoàn toàn giống nhau và ngược lại. Hình 2 c) là ma trận tương tự Similartiy matrix  $S$  sau khi tính toán độ tương tự giữa các người dùng bằng công thức Cosine Similarity. Ma trận  $S$  là một ma trận đối xứng. Ta có thể thấy ở các vector hàng tương ứng với  $u_0, u_1, u_2$

- $u_0$  với  $u_1$  và  $u_5$  có độ tương tự là lớn hơn 0, tức là so với các người dùng còn lại,  $u_1$  và  $u_5$  được cho là tương tự với  $u_0$ . Với việc  $u_0$  và  $u_1$  giống nhau, ta có thể dễ dàng nhận xét được như ở phần trước. Việc  $u_0$  gần với  $u_5$  thoạt đầu có vẻ vô lý vì  $u_5$  đánh giá thấp các phim mà  $u_0$  đánh giá cao (Hình 2 a)); tuy nhiên khi nhìn vào ma trận utility đã chuẩn hoá ở Hình 2b), ta thấy rằng điều này là hợp lý.

Vì phim duy nhất mà cả hai người dùng này đã cung cấp thông tin là  $i1$  với các giá trị tương ứng đều là tích cực.

- $u1$  gần với  $u0$  và xa các người dùng còn lại.
- $u2$  gần với  $u3, u4, u5, u6$  và xa các người dùng còn lại.

Từ ma trận tương tự này, chúng ta có thể phân nhóm các người dùng ra làm hai nhóm ( $u0, u1$ ) và ( $u2, u3, u4, u5, u6$ ). Vì ma trận  $S$  này nhỏ nên chúng ta có thể dễ dàng quan sát thấy điều này. Khi số người dùng lớn hơn, việc xác định bằng mắt thường là không khả thi. Có một chú ý quan trọng ở đây là khi số lượng người dùng lớn, ma trận  $S$  cũng rất lớn và nhiều khả năng là không có đủ bộ nhớ để lưu trữ, ngay cả khi chỉ lưu hơn một nửa số các phần tử của ma trận đối xứng này. Với các trường hợp đó, với mỗi người dùng, chúng ta chỉ cần tính và lưu kết quả của một hàng của ma trận tương tự, tương ứng với việc độ giống nhau giữa người dùng đó và các người dùng còn lại.

#### 2.4.3.3. Dự đoán kết quả (Rating prediction)

Công thức được sử dụng để dự đoán điểm đánh giá (rating) của  $u$  cho  $i$  là:

$$y_{i,u} = \frac{\sum_{u_j \in N(u, i)} \bar{y}_{i, u_j} \text{sim}(u, u_j)}{\sum_{u_j \in N(u, i)} |\text{sim}(u, u_j)|} \quad (2.3)$$

Công thức dự đoán rating của người dùng cho phim [1] Công thức có nghĩa là tổng của các tích giữa giá trị chuẩn hóa của người dùng tại phim cần dự đoán và độ tương tự của người dùng đối với các người dùng khác đánh giá phim cần dự đoán.

Một ví dụ về việc dự đoán điểm đánh giá chuẩn hóa của  $u1$  cho  $i1$  được cho trong Hình 2 e) với số láng giềng gần nhất (nearest neighbors) là  $k=2$ . Các bước thực hiện là:

1. Xác định các người dùng đã đánh giá  $i1$ , đó là  $u0, u3, u5$ .
2. Xác định độ tương tự của  $u1$  với các người dùng này ta nhận được các giá trị  $0.83, -0.40, -0.23$ . Hai ( $k=2$ ) giá trị lớn nhất là  $0.83$  và  $-0.23$  tương ứng với  $u0$  và  $u5$ .
3. Xác định các điểm đánh giá chuẩn hóa của  $u0, u5$  cho  $i1$ , ta thu được hai giá trị lần lượt là  $0.75$  và  $0.5$ .
4. Dự đoán kết quả:

$$y_{i,u} = \frac{0.83 \times 0.75 + (-0.23) \times 0.5}{0.83 + |-0.23|} \approx 0.48$$

Việc quy đổi các giá trị đánh giá đã chuẩn hoá về thang 5 có thể được thực hiện bằng cách cộng các cột của ma trận điểm đánh giá chuẩn hóa (hình 2 d) với giá trị đánh giá trung bình của mỗi người dùng như đã tính trong Hình 2 a).

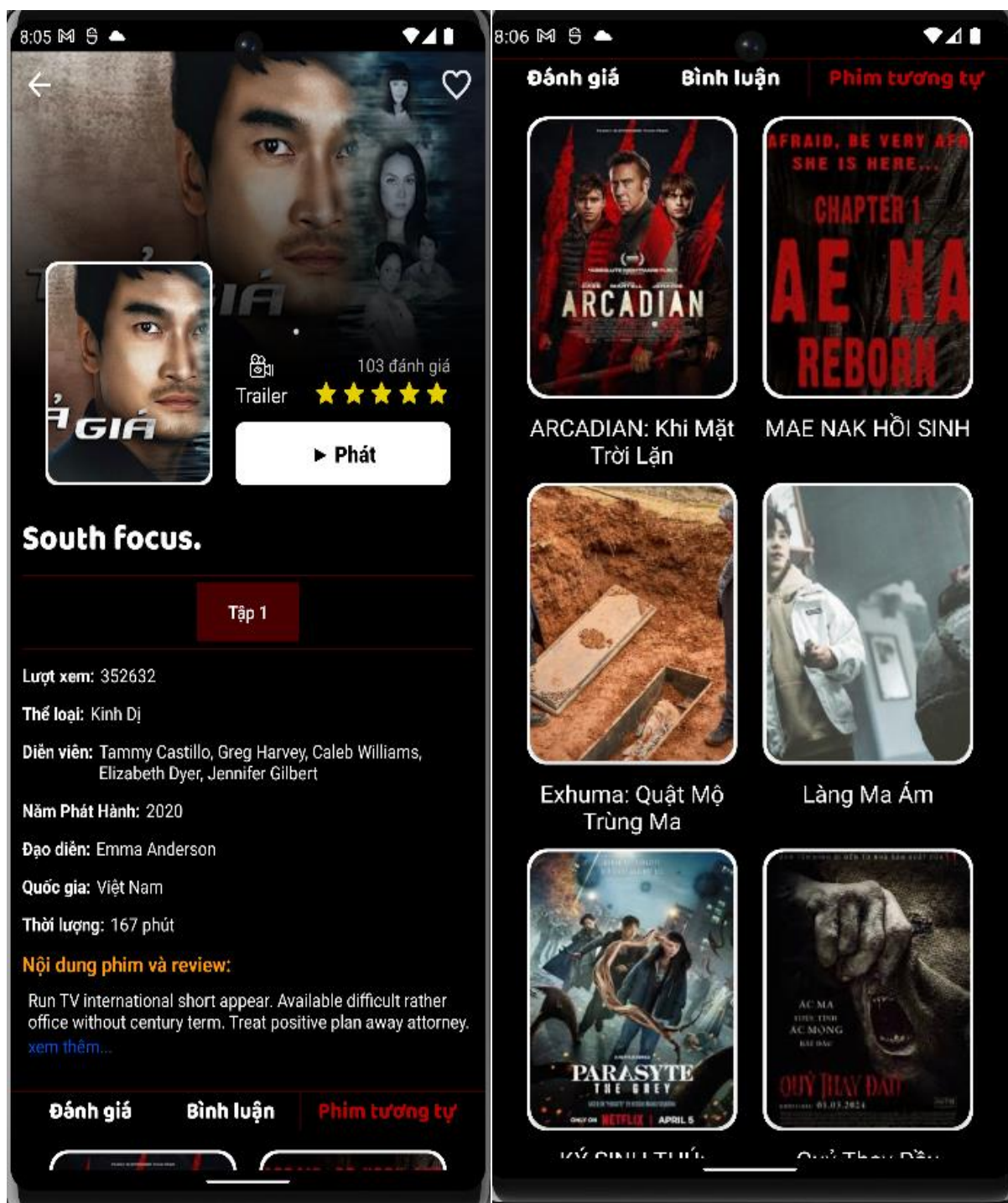
**Kết Luận:** Việc hệ thống quyết định gợi ý phim nào cho mỗi người dùng có thể được xác định bằng nhiều cách khác nhau. Có thể sắp xếp các phim chưa được đánh giá theo thứ tự từ lớn đến bé của các điểm dự đoán (predict rating), hoặc chỉ chọn các phim có điểm dự đoán dương - tương ứng với việc người dùng này có nhiều khả năng thích hơn

**Kết quả:** Sau khi chạy thuật toán, kết quả được in ra và ứng dụng sẽ đề xuất cho người dùng khác.

Độ tương đồng giữa 50 và 1027:	0.372201646265082
Độ tương đồng giữa 50 và 1028:	0.3717229811218043
Độ tương đồng giữa 50 và 1029:	0.3723638152082504
Độ tương đồng giữa 50 và 1030:	0.3947138549436597
Độ tương đồng giữa 50 và 1031:	0.3715303019880305
Độ tương đồng giữa 50 và 1032:	0.3708296235522264
Độ tương đồng giữa 50 và 1033:	0.3738360085528195
Độ tương đồng giữa 50 và 1034:	0.5817160507976249
Độ tương đồng giữa 50 và 1035:	0.3709225215106125
Độ tương đồng giữa 50 và 1036:	0.3868738486877749
Độ tương đồng giữa 50 và 1037:	0.3716251207971464
Độ tương đồng giữa 50 và 1038:	0.3709849090495104
Độ tương đồng giữa 50 và 1039:	0.3708725241353033
Độ tương đồng giữa 50 và 1040:	0.373252706027831
Độ tương đồng giữa 50 và 1041:	0.3708391089135586
Độ tương đồng giữa 50 và 1042:	0.3713393781759506
Độ tương đồng giữa 50 và 1043:	0.371650100878234
Độ tương đồng giữa 50 và 1044:	0.3711851638750373
Độ tương đồng giữa 50 và 1045:	0.3927291569360112
Độ tương đồng giữa 50 và 1046:	0.3707884610321355
Độ tương đồng giữa 50 và 1047:	0.3711793976151814
Độ tương đồng giữa 50 và 1048:	0.3708058476505899
Độ tương đồng giữa 50 và 1049:	0.3715644728963475
Độ tương đồng giữa 50 và 1050:	0.4259786028949024
Độ tương đồng giữa 50 và 1051:	0.3708997506245941
Độ tương đồng giữa 50 và 1052:	0.3723433121958413
Độ tương đồng giữa 50 và 1053:	0.3711565014868265
Độ tương đồng giữa 50 và 1054:	0.3919971183184541
Độ tương đồng giữa 50 và 1055:	0.3724151707751339
Độ tương đồng giữa 50 và 1056:	0.3707987414802687
Độ tương đồng giữa 50 và 1057:	0.3728403361957639
Độ tương đồng giữa 50 và 1058:	0.3709176613960154
Độ tương đồng giữa 50 và 1059:	0.3717567558420519
Độ tương đồng giữa 50 và 1060:	0.3709261658108129
Độ tương đồng giữa 50 và 1061:	0.3714061362811369
Độ tương đồng giữa 50 và 1062:	0.3715451707584258
Độ tương đồng giữa 50 và 1063:	0.5820371528368284
Độ tương đồng giữa 50 và 1064:	0.3710704499929196
Độ tương đồng giữa 50 và 1065:	0.3711828574466954
Độ tương đồng giữa 50 và 1066:	0.3707904918835009
Độ tương đồng giữa 50 và 1067:	0.3712140457015444
Độ tương đồng giữa 50 và 1068:	0.5809368213216157
Độ tương đồng giữa 50 và 1069:	0.3715586165204023
Độ tương đồng giữa 50 và 1070:	0.3717528128841123
Độ tương đồng giữa 50 và 1071:	0.3709494491290324
Độ tương đồng giữa 50 và 1072:	0.3711269030537146
Độ tương đồng giữa 50 và 1073:	0.3720374513605949
Độ tương đồng giữa 50 và 1074:	0.3716545400454748
Độ tương đồng giữa 50 và 1075:	0.4058195703178911
Độ tương đồng giữa 50 và 1076:	0.3709595245078587
Độ tương đồng giữa 50 và 1077:	0.3721229993183367
Độ tương đồng giữa 50 và 1078:	0.3709093627184761
Độ tương đồng giữa 50 và 1079:	0.6080679672750157

Hình 3. Kết quả chạy thuật toán.





Hình 4 Kết quả chạy chương trình

## **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **3.1. Khảo sát và đánh giá hệ thống**

Hiện nay, có rất nhiều ứng dụng xem phim trên internet. Nhưng từ quá trình khảo sát, em nhận thấy ngoài các ứng dụng có số lượng đông đảo người sử dụng như:

- Clip TV
- VTV Go
- Zing TV
- FPT Play
- Netflix

Qua quá trình khảo sát, hệ thống xem phim được chia ra làm hai phần chính:

- Phần thứ nhất là giao dịch với khách hàng trên ứng dụng.
- Phần thứ hai là dành cho nhân viên quản trị hệ thống.

Mỗi phần có chức năng con khác nhau phục vụ cho mục đích quản lý.

Sau khi khảo sát các hệ thống trên, em muốn xây dựng một ứng dụng có các chức năng tương tự để nhằm mục đích nâng cao trình độ lập trình, khả năng phân tích hệ thống và khả năng tự tìm hiểu.

### **3.2. Quy trình thực hiện quản trị hệ thống của nhân viên quản trị**

Để truy cập vào chức năng quản trị hệ thống, quản trị viên cần có một tài khoản để đăng nhập, sau khi đăng nhập thì có thể thực hiện các chức năng sau:

- Quản trị danh mục loại phim: Bao gồm các công việc thêm, xóa, cập nhật thông tin về phim.
- Quản trị phim: Thêm, xóa và cập nhật thông tin về mỗi bộ phim.
- Quản trị người dùng: Thêm, xóa và cập nhật thông tin người dùng và theo dõi người dùng.
- Quản trị bình luận, yêu thích, đánh giá: Xem được những đánh giá, bình luận, lượt yêu thích phim của người dùng.



### **3.3. Yêu cầu hệ thống**

#### **3.3.1. Yêu cầu chức năng**

- Ở giao diện người quản trị ứng dụng
  - Quản lý danh sách phim
  - Lọc, tìm kiếm phim theo nhiều tiêu chí.
  - Thêm sửa xóa phim, người dùng.
- Ở giao diện người dùng
  - Chức năng đăng nhập, đăng ký tài khoản.
  - Chức năng đánh giá, bình luận, yêu thích.
  - Chức năng tìm kiếm, lọc phim, lưu tiến trình, lịch sử xem phim

#### **3.3.2. Yêu cầu phi chức năng**

- Giao diện ứng dụng xem phim trực tuyến phải đơn giản, thân thiện với người dùng, tạo cho khách hàng cảm giác thích thú mà muốn khám phá ngay lần đầu tiên khi truy cập vào ứng dụng.
- Bố cục ứng dụng phải gọn gàng, phù hợp với người dùng.

#### **3.3.3. Yêu cầu khác:**

- Ứng dụng phải đơn giản, dễ sử dụng.
- Tốc độ tải và tốc độ mở ứng không được quá chậm, tránh việc gây khó chịu cho người dùng.
- Khi có thay đổi trong quá trình xử lý hay tạm ngừng ứng dụng để bảo trì, sửa chữa phải có thông báo cho người dùng.
- Chạy ổn định trên android studio.

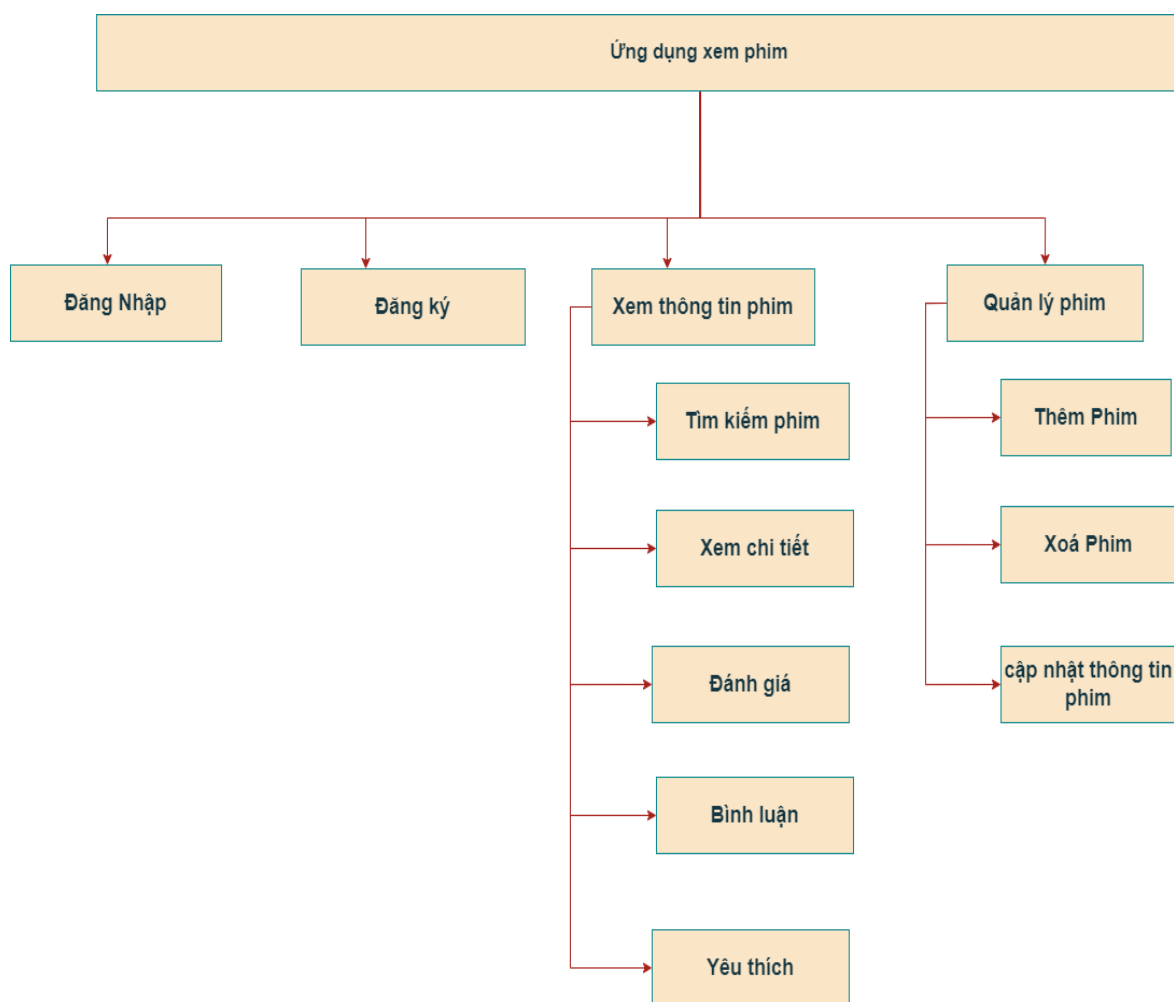
### **3.4. Sơ đồ phân rã chức năng và luồng dữ liệu**

Sau khi khảo sát và đánh giá một số ứng dụng xem phim, em đã phân tích quy trình hoạt động của những ứng dụng đó và đưa ra được sơ đồ phân rã chức năng như sau:

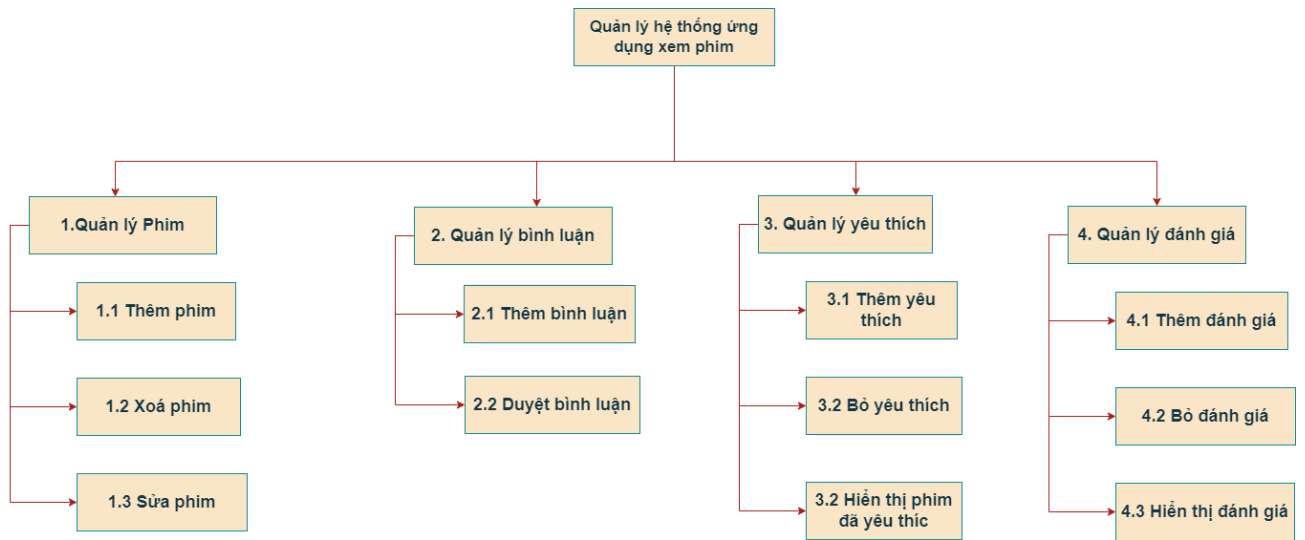
### 3.4.1. Yêu cầu người dùng

- Đối với tất cả người dùng:
  - Cần có thiết bị để kết nối internet để truy cập ứng dụng
  - Người dùng chưa đăng ký có thể tra cứu và xem thông tin phim, cần đăng ký để có thể xem phim.
- Đối với quản trị viên:
  - Cần có tài khoản được cấp quyền admin để truy cập hệ thống.

### 3.4.2. Sơ đồ phân rã chức năng

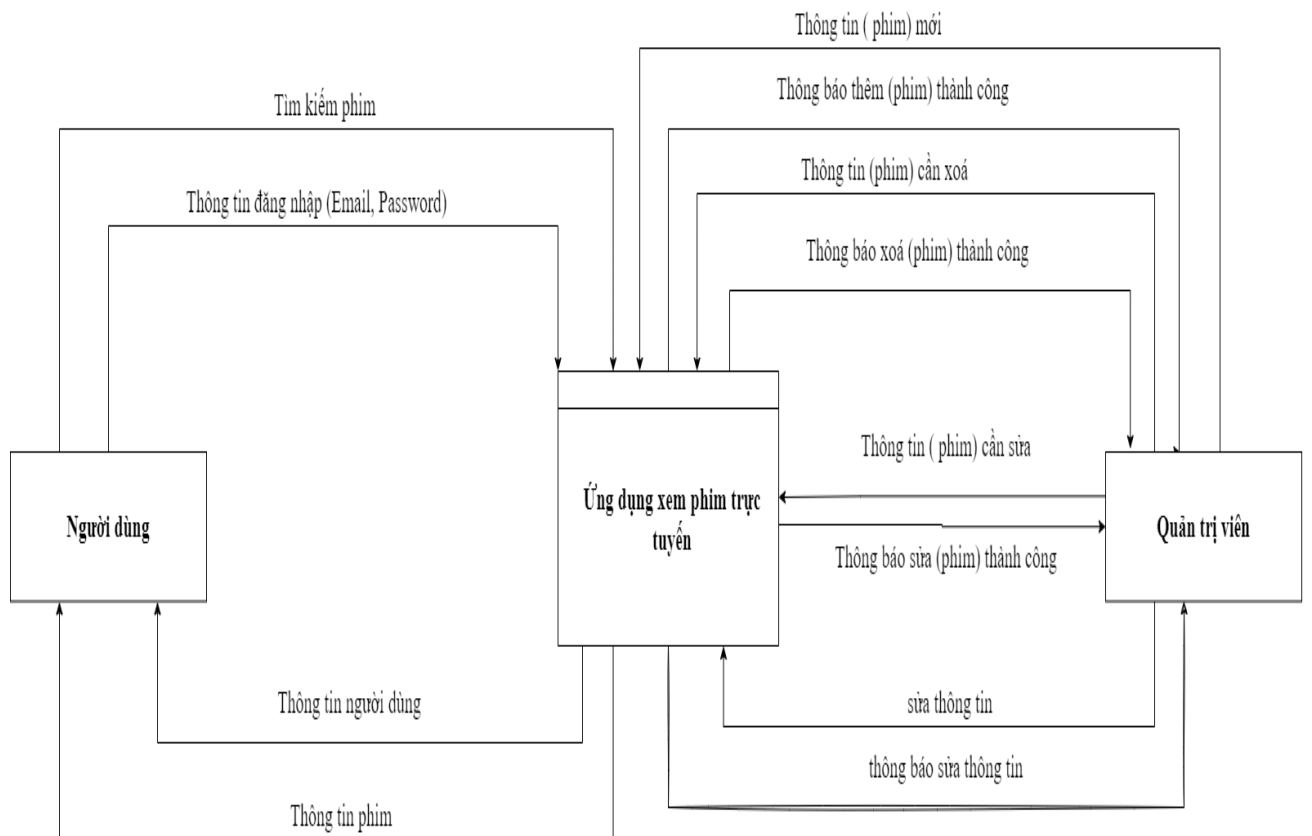


Hình 5. Sơ đồ phân rã chức năng ứng dụng xem phim



Hình 6. Sơ đồ phân rã chức năng Quản lý ứng dụng

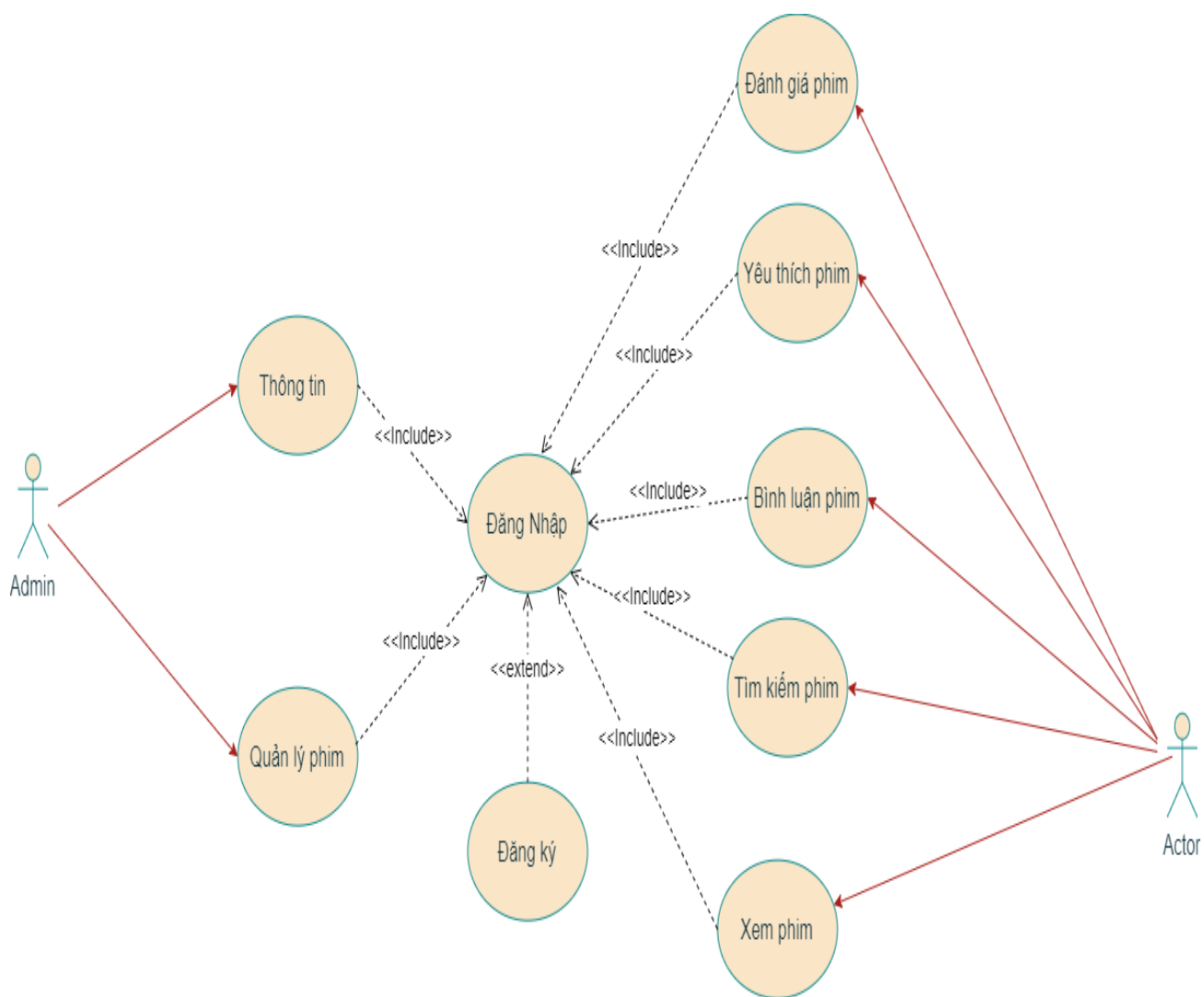
### 3.4.3. Sơ đồ luồng dữ liệu



Hình 7. Sơ đồ luồng dữ liệu

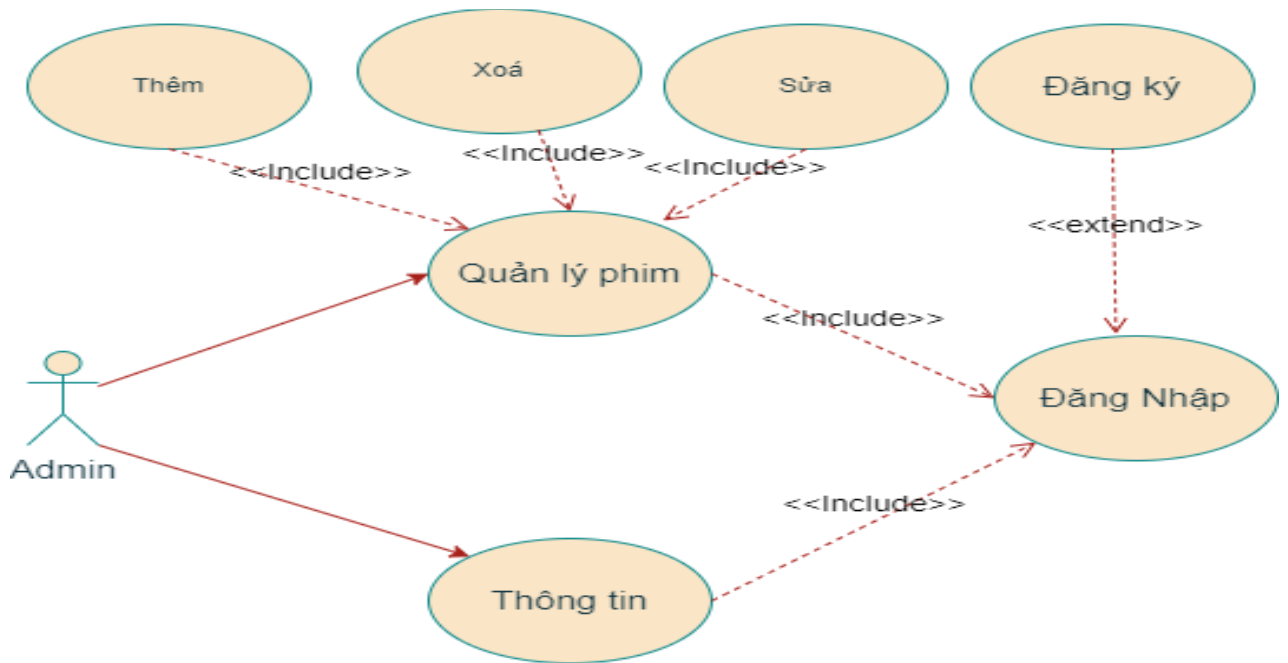
### 3.5. Biểu đồ UseCase

#### 3.5.1.1. Biểu đồ UseCase tổng quát



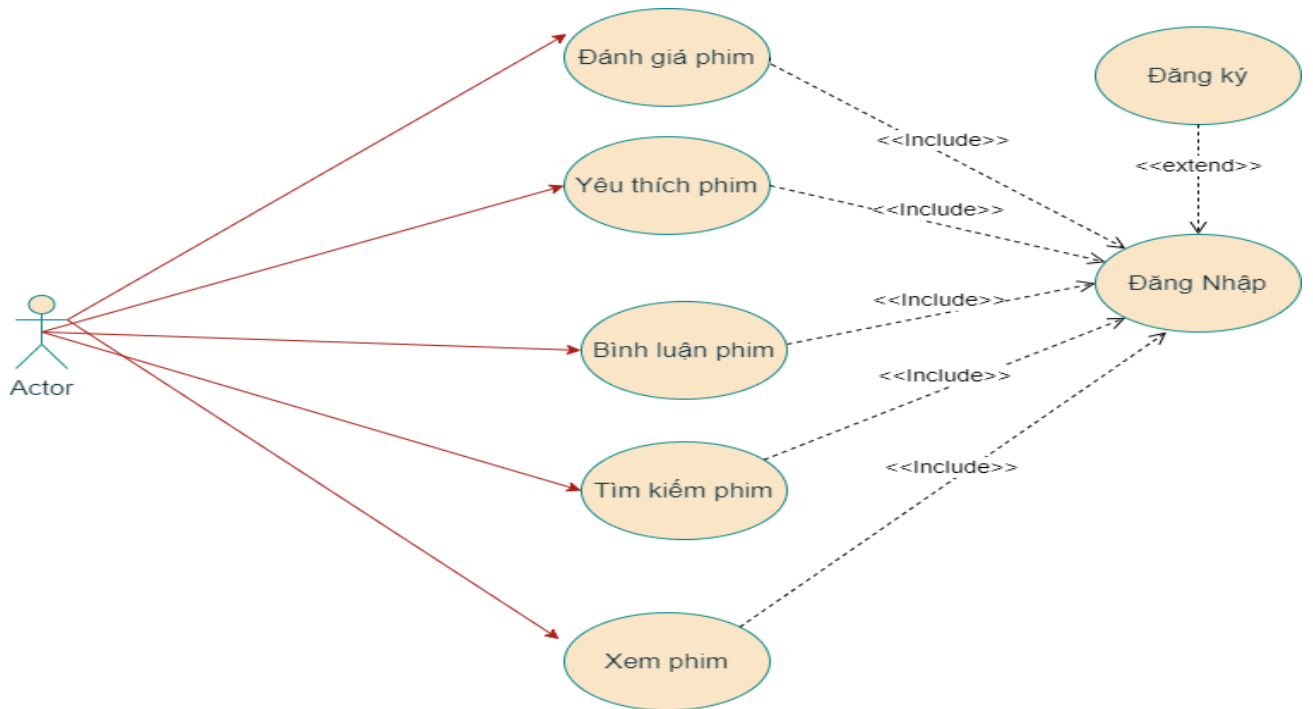
Hình 8. Biểu đồ UseCase tổng quát

### 3.5.1.2. Biểu đồ UseCase Admin



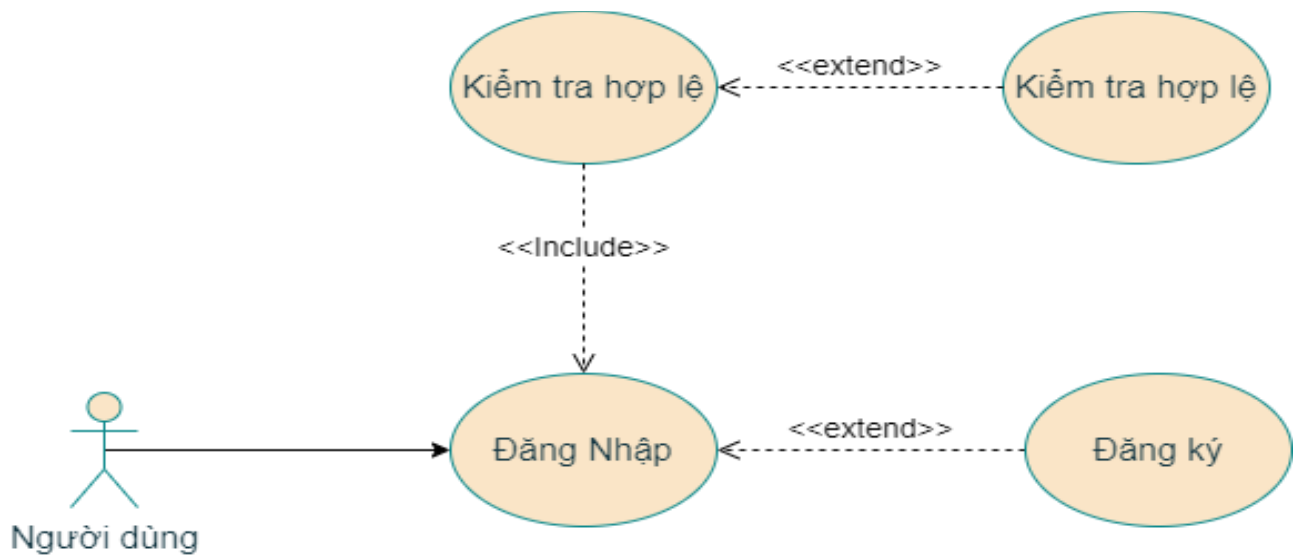
Hình 9. UseCase Admin

### 3.5.1.3. Biểu đồ UseCase người dùng



Hình 10. Sơ đồ UseCase người dùng

#### 3.5.1.4. UseCase đăng nhập



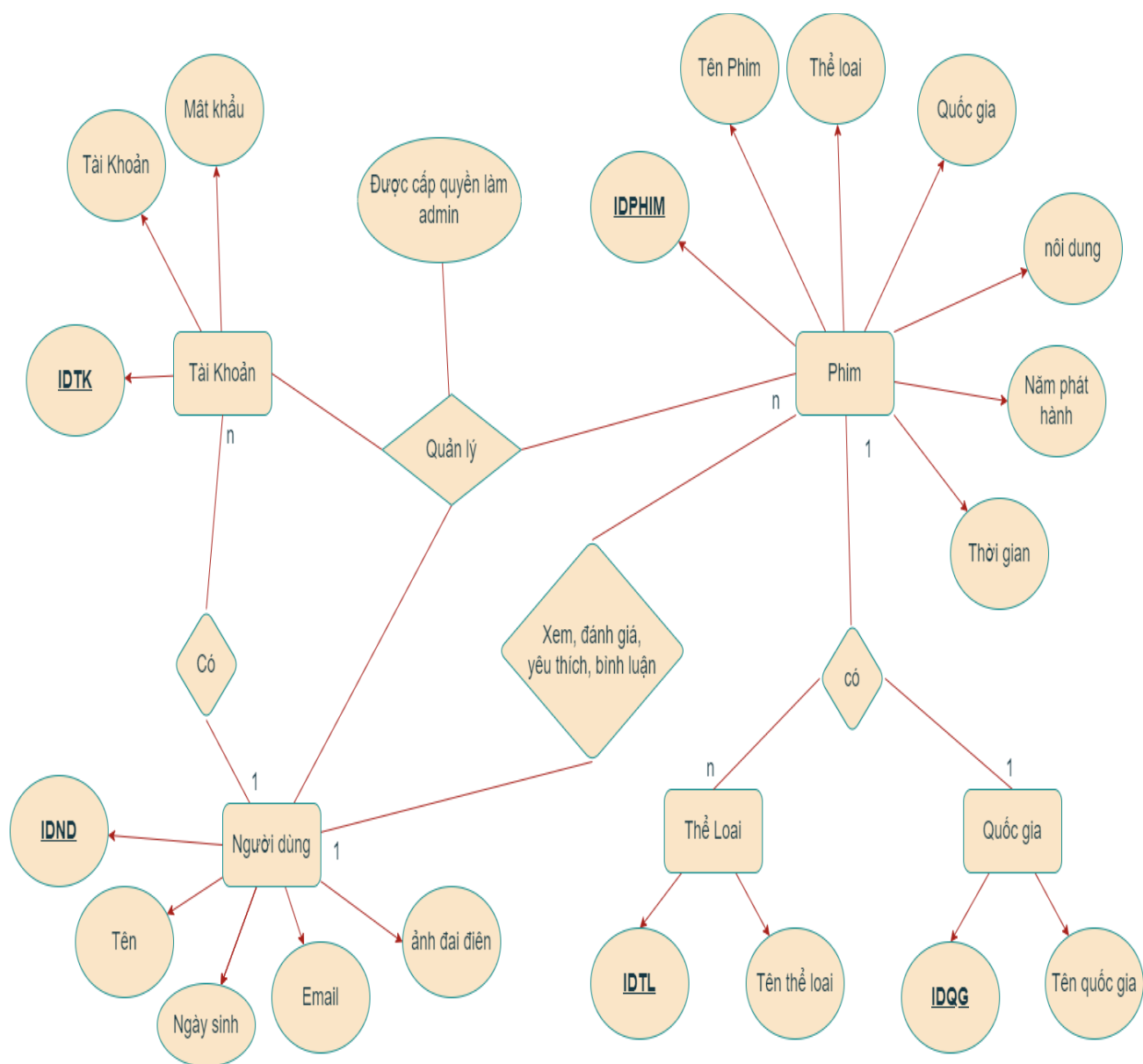
Hình 11. Sơ đồ Usecase đăng nhập

#### Mô Tả

Biểu đồ usecase biểu diễn sơ đồ chức năng của hệ thống. Từ các yêu cầu của hệ thống, biểu đồ usecase chỉ ra hệ thống cần thực hiện những điều gì để đáp ứng nhu cầu của hệ thống sơ đồ luồng dữ liệu. Các chức năng của từng tác nhân được thể hiện như sau:

- Người quản trị ứng cần đăng nhập để thực hiện các chức năng quản lý phim, quản lý người dùng.
- Khách hàng có thể đăng ký tài khoản, xem, tìm kiếm phim.
- Khách hàng sau khi đăng nhập thực hiện đánh giá, bình luận hoặc thêm phim vào mục ưa thích.

### 3.6. Sơ đồ ER

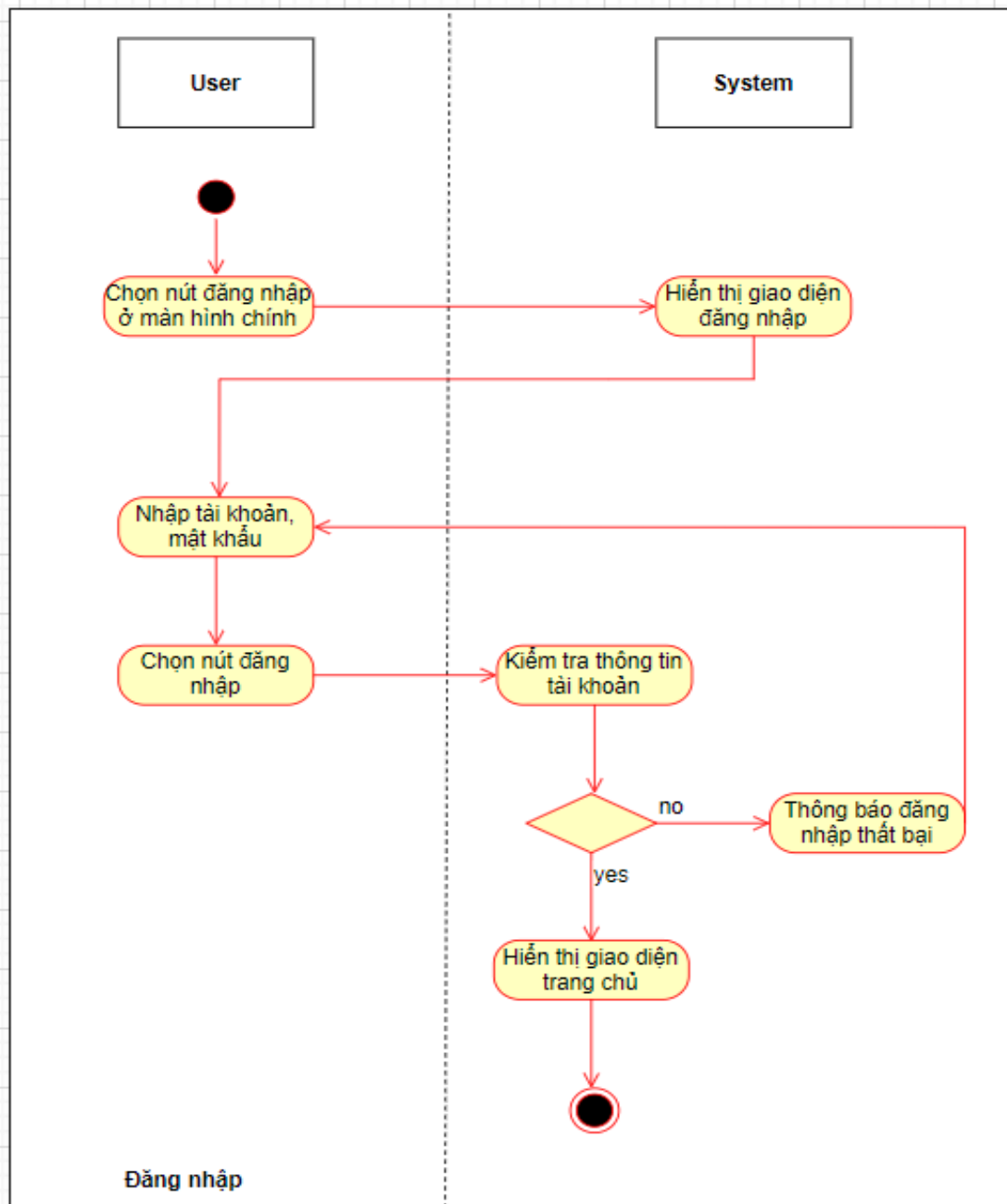


Hình 12 Sơ đồ ER

### 3.7. Sơ đồ hoạt động

#### 3.7.1. Với người dùng

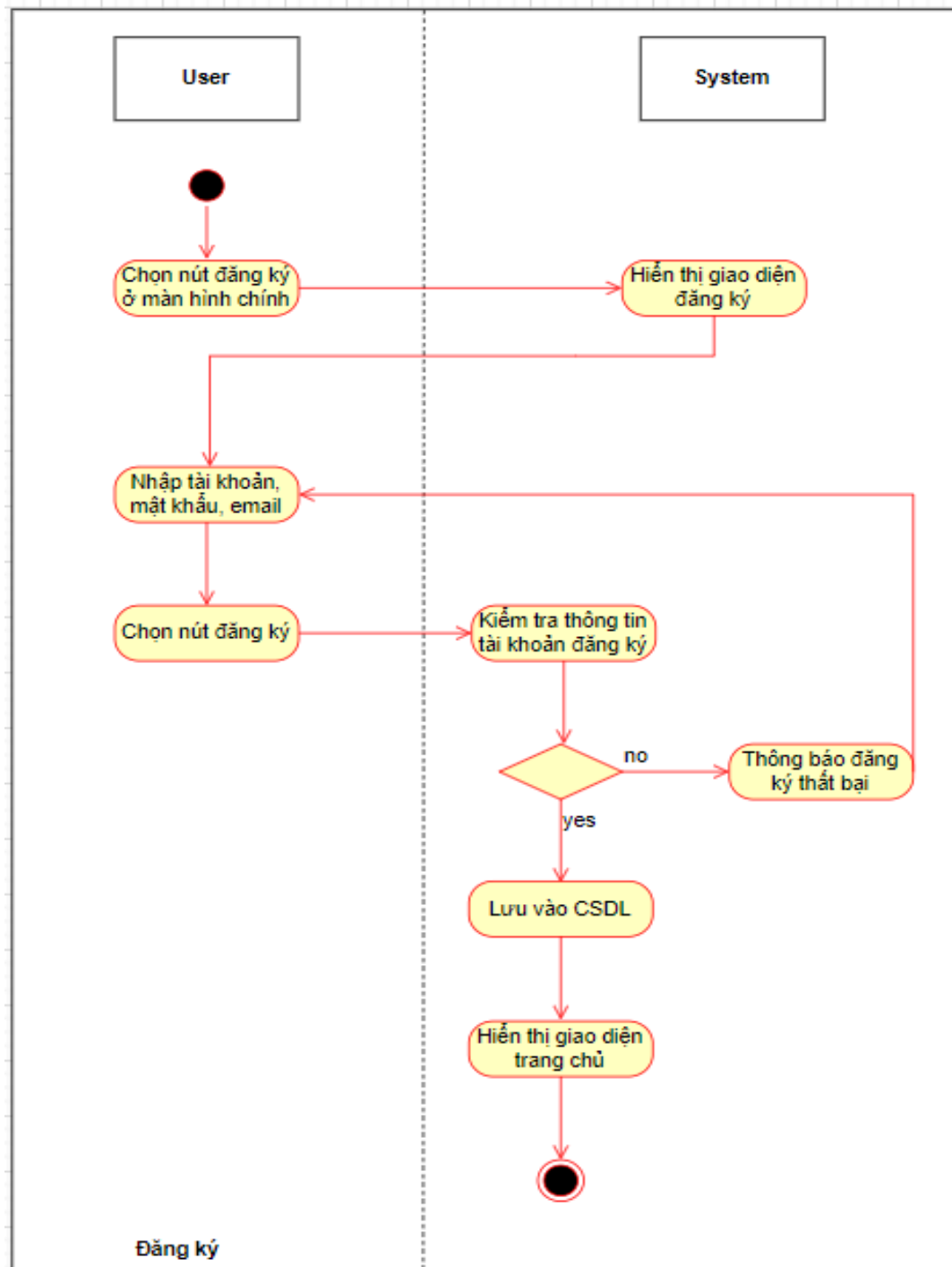
##### 3.7.1.1. Đăng nhập



Hình 13. Sơ đồ hoạt động chức năng đăng nhập

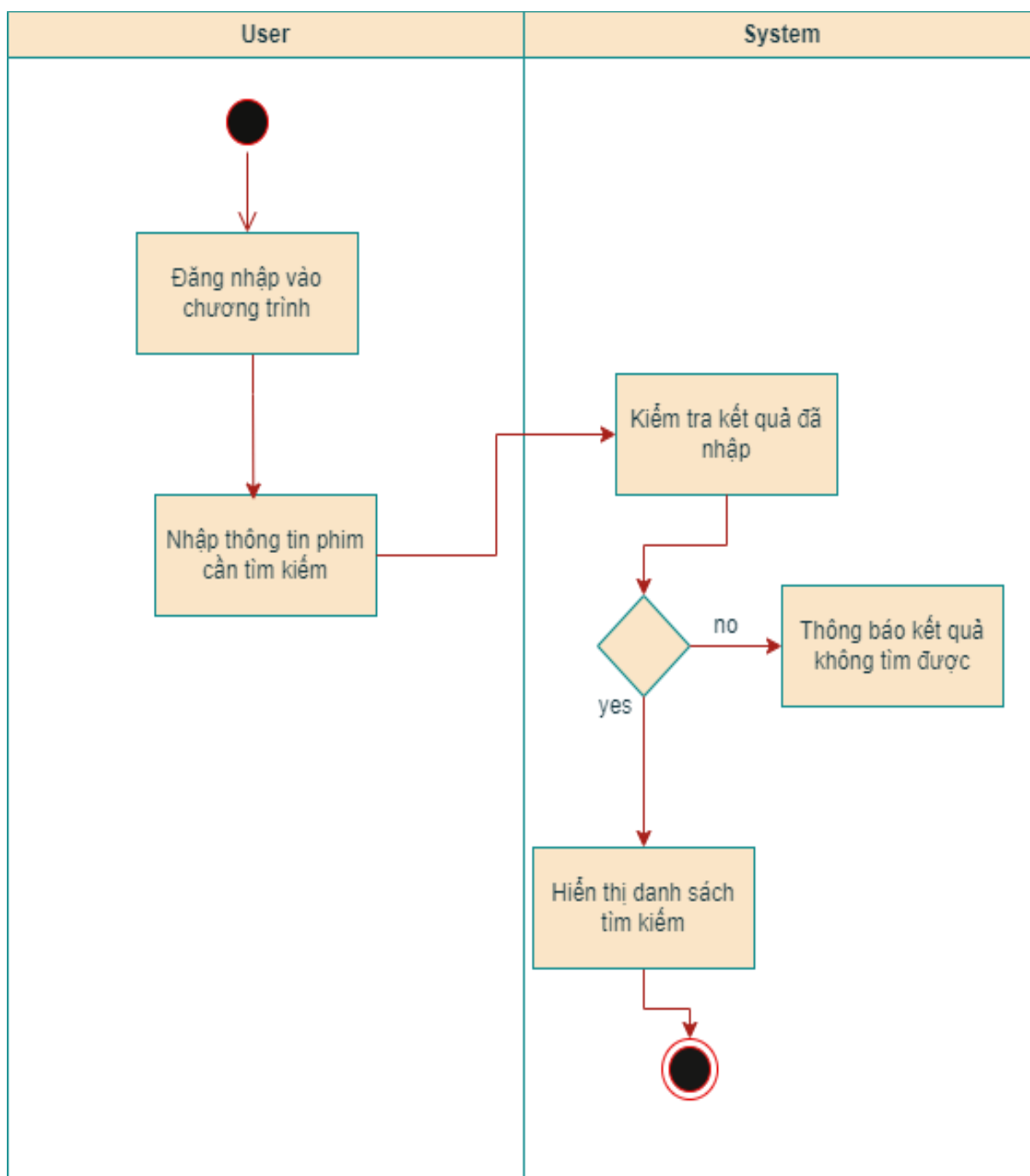


### 3.7.1.2. Đăng ký



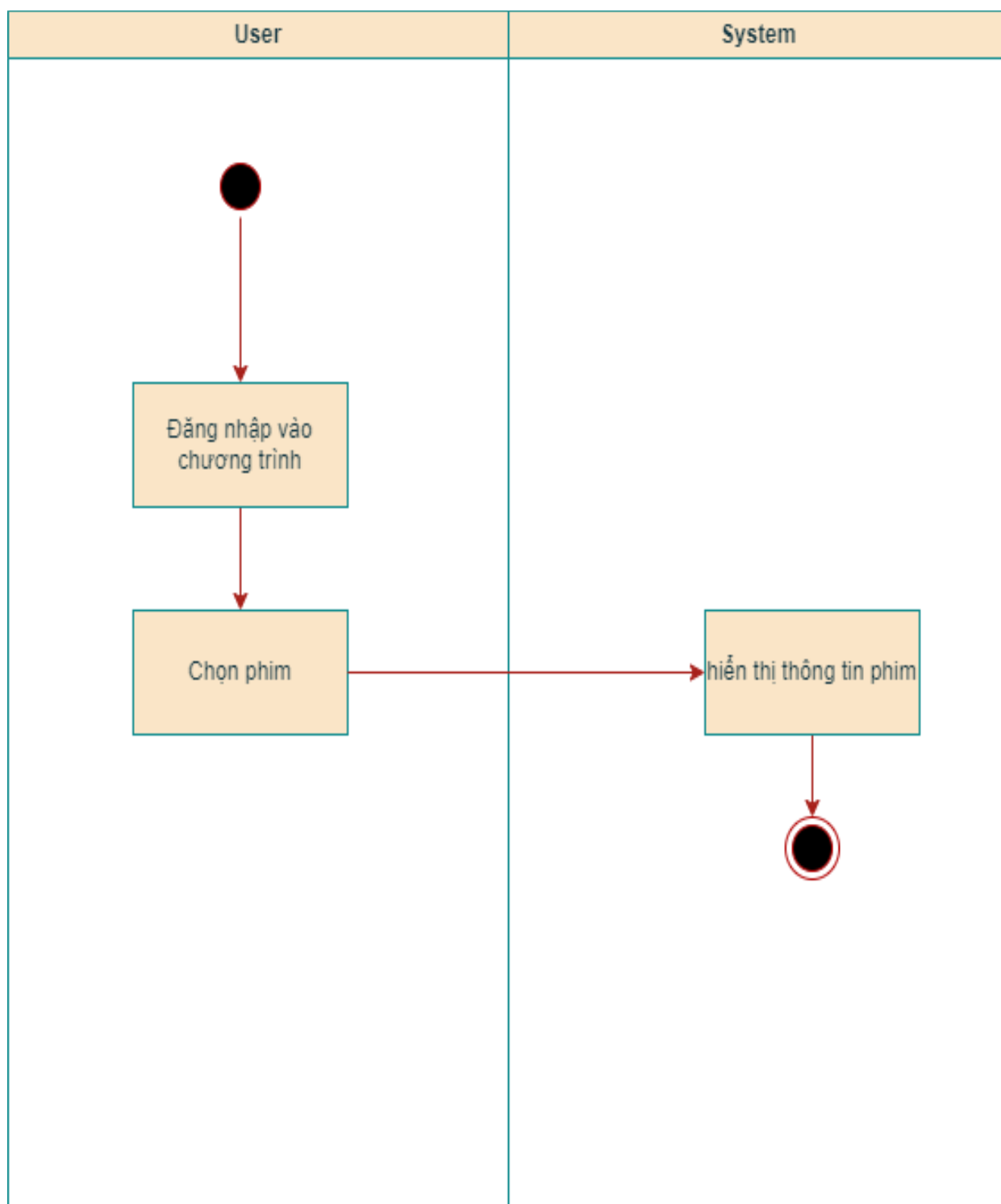
Hình 14. Sơ đồ hoạt động chức năng đăng ký

### 3.7.1.3. Tìm kiếm phim



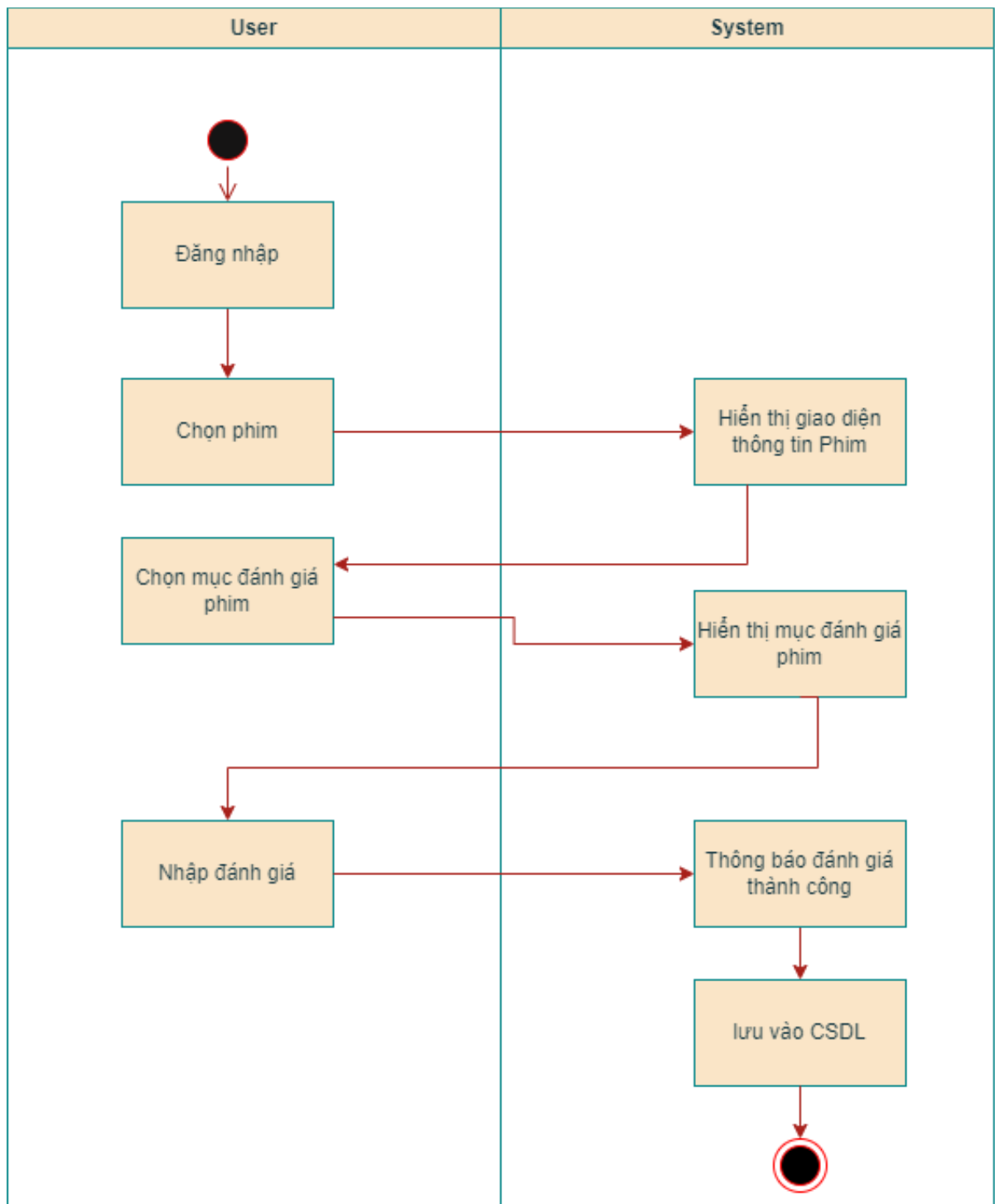
Hình 15. Sơ đồ hoạt động chức năng tìm kiếm phim

#### 3.7.1.4. Xem thông tin phim



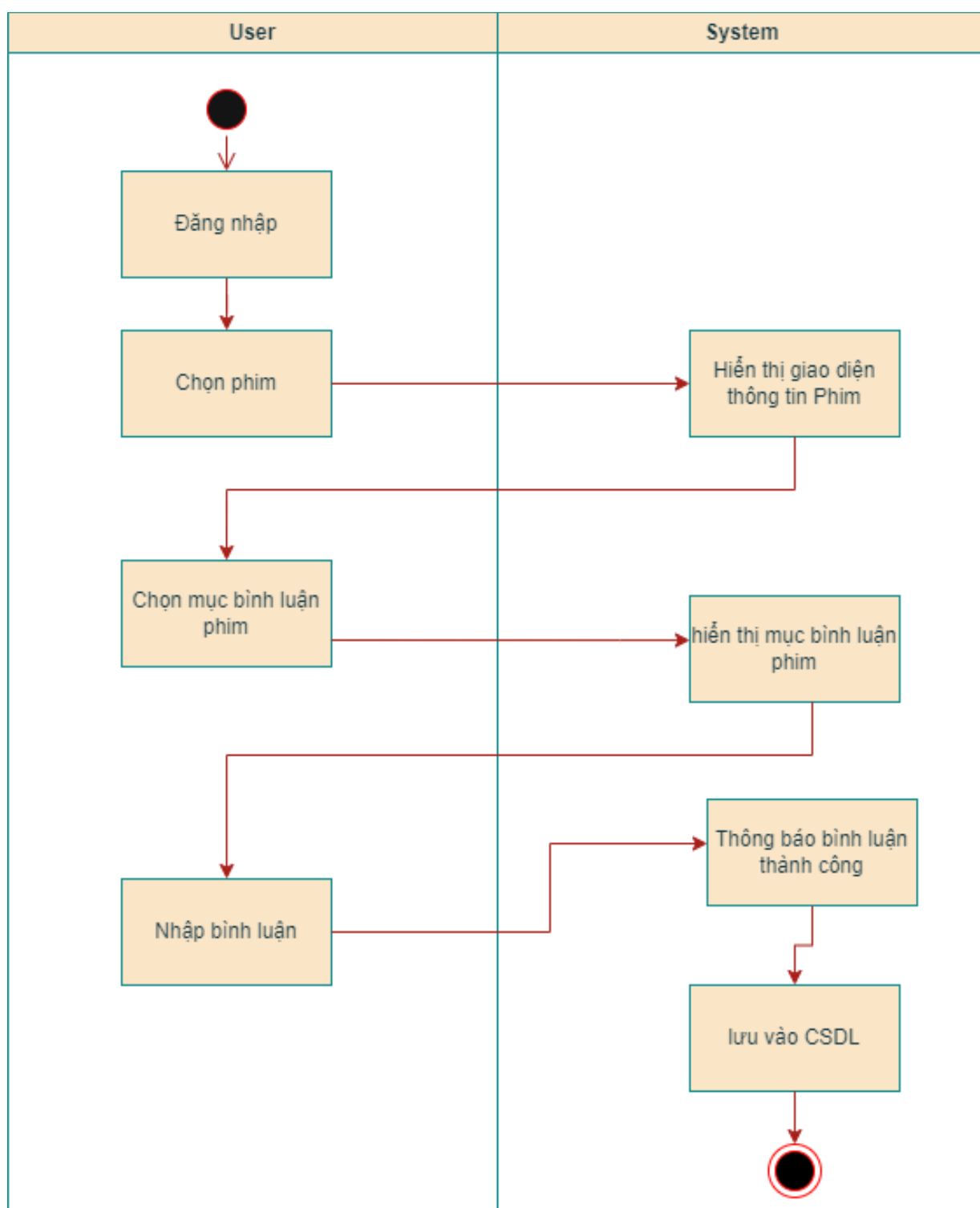
Hình 16. Sơ đồ hoạt động chức năng xem chi tiết phim

### 3.7.1.5. Đánh giá phim



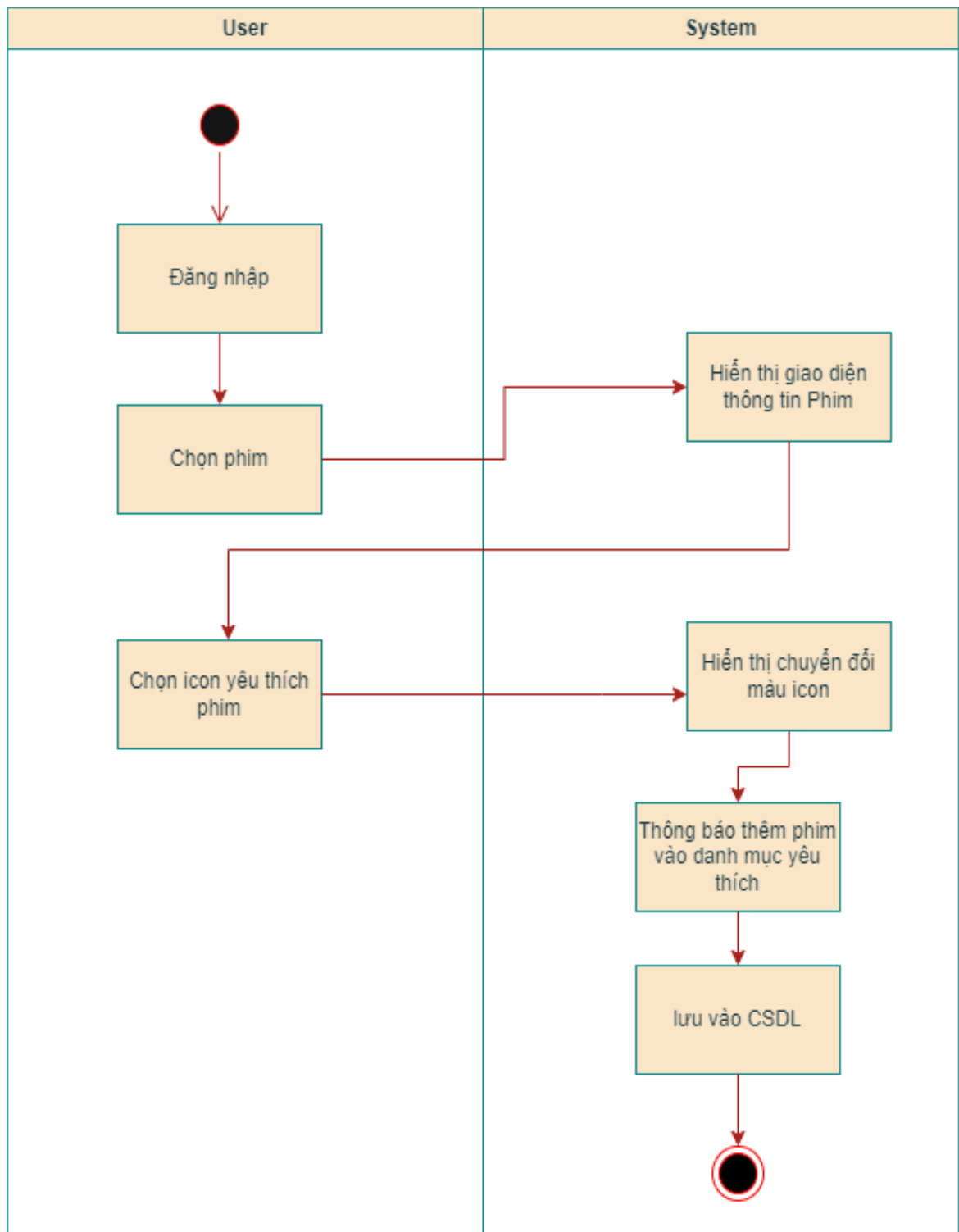
Hình 17. Sơ đồ hoạt động chức năng đánh giá phim

### 3.7.1.6. Bình luận phim



Hình 18. Sơ đồ hoạt động chức năng bình luận phim

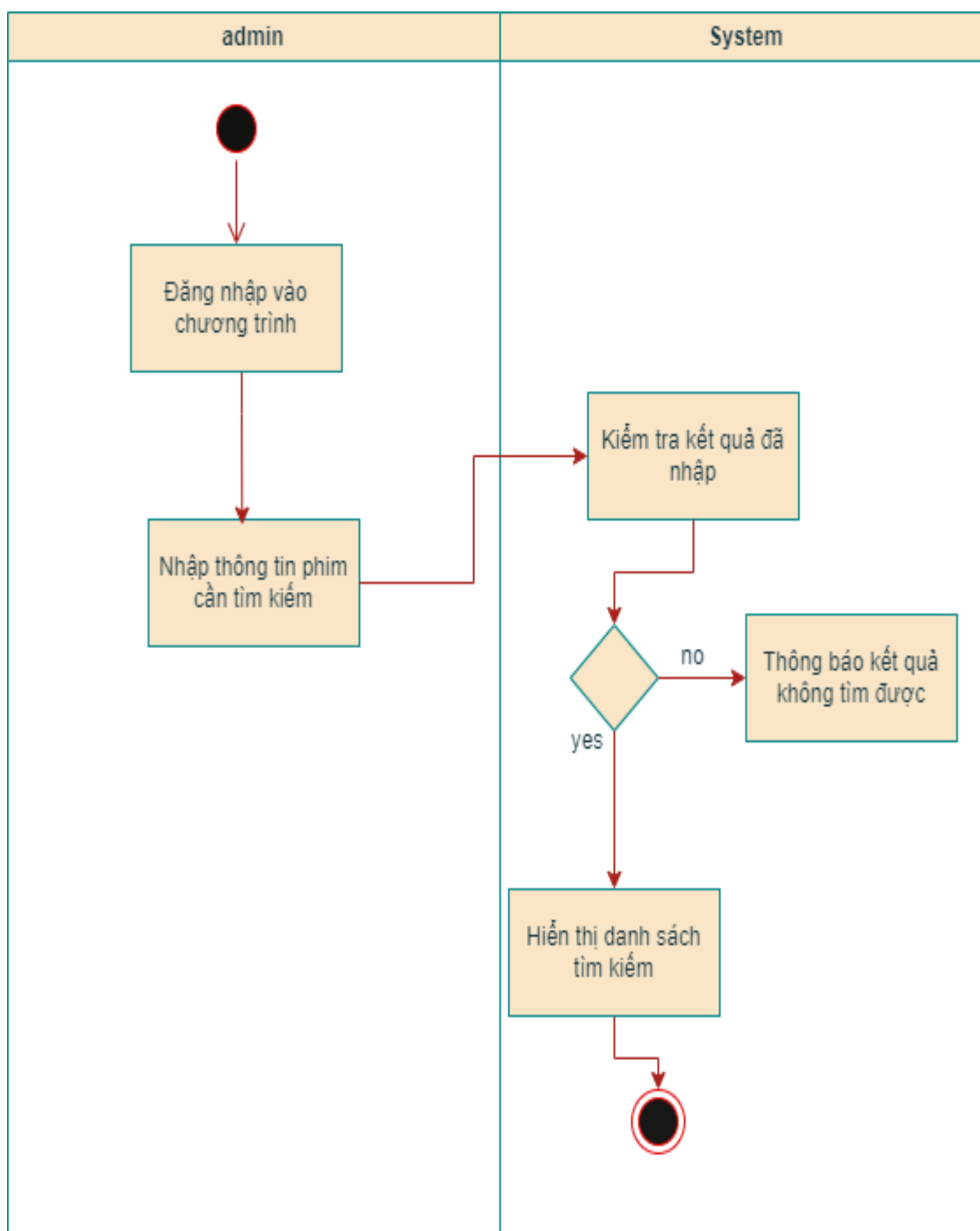
### 3.7.1.7. Yêu thích phim



Hình 19. Sơ đồ hoạt động chức năng yêu thích phim

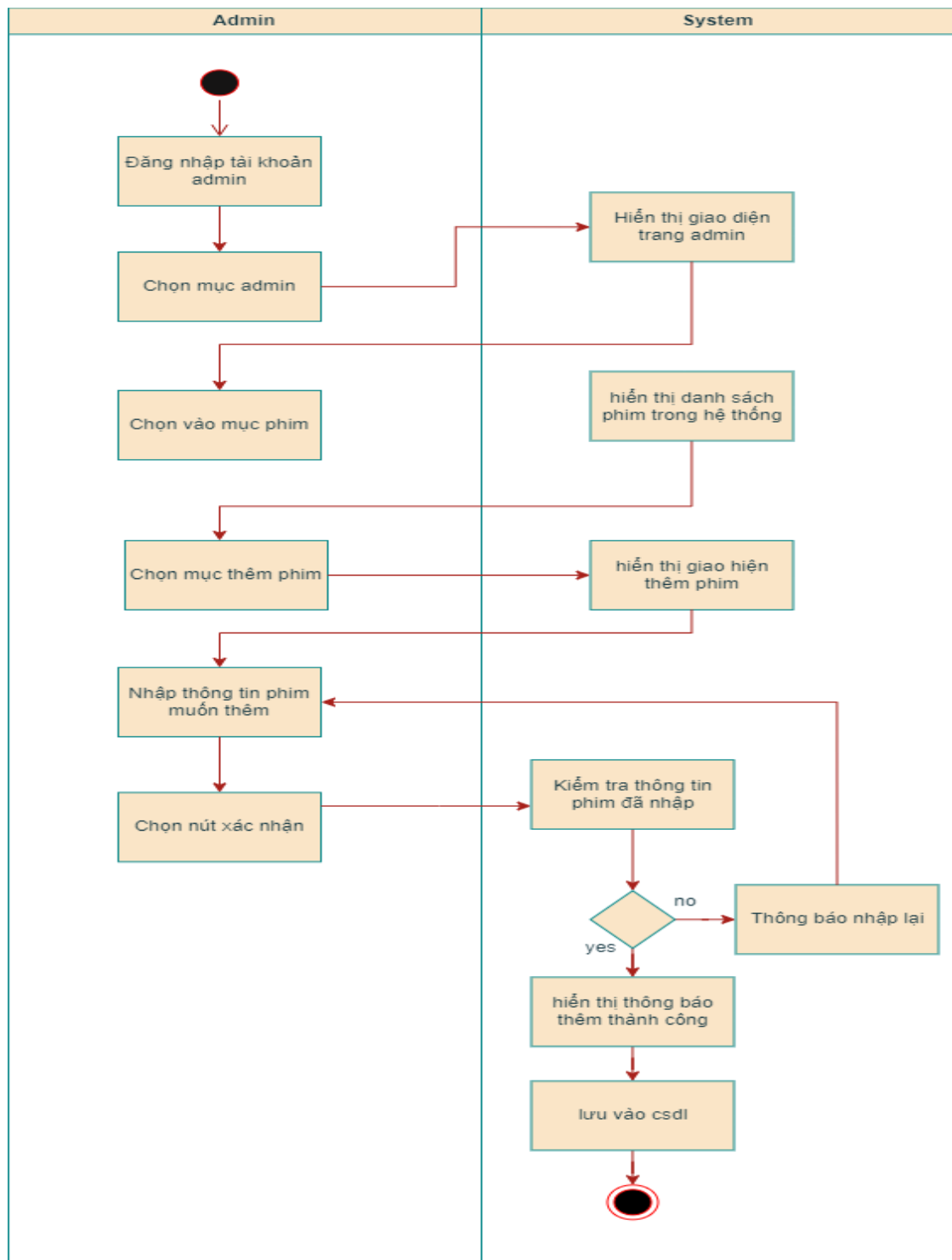
### 3.7.2. Với người quản trị

#### 3.7.2.1. Tìm kiếm phim



Hình 20. Sơ đồ hoạt động chức năng tìm kiếm phim

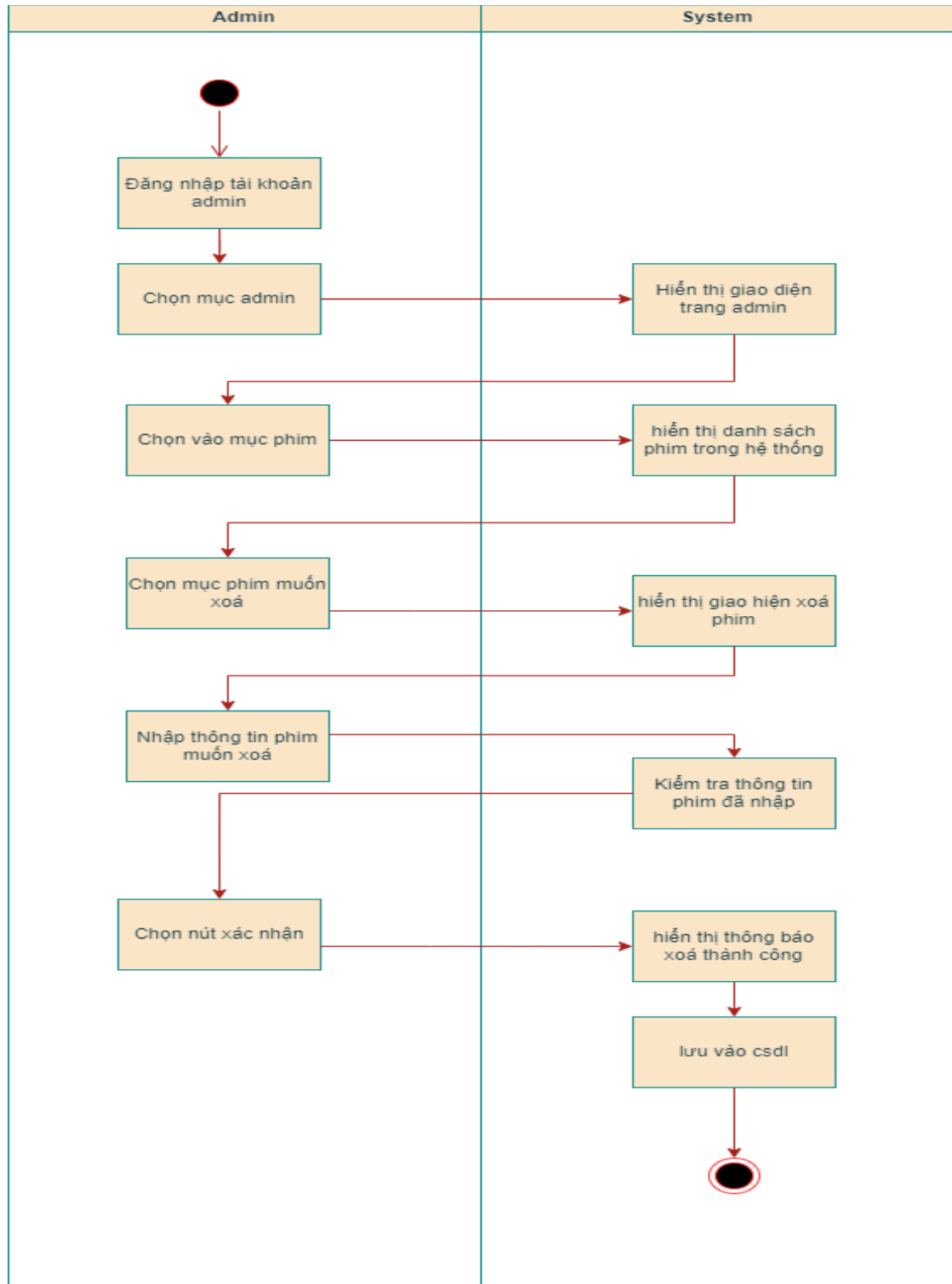
### 3.7.2.2. Thêm phim



Hình 21. Sơ đồ hoạt động chức năng thêm phim

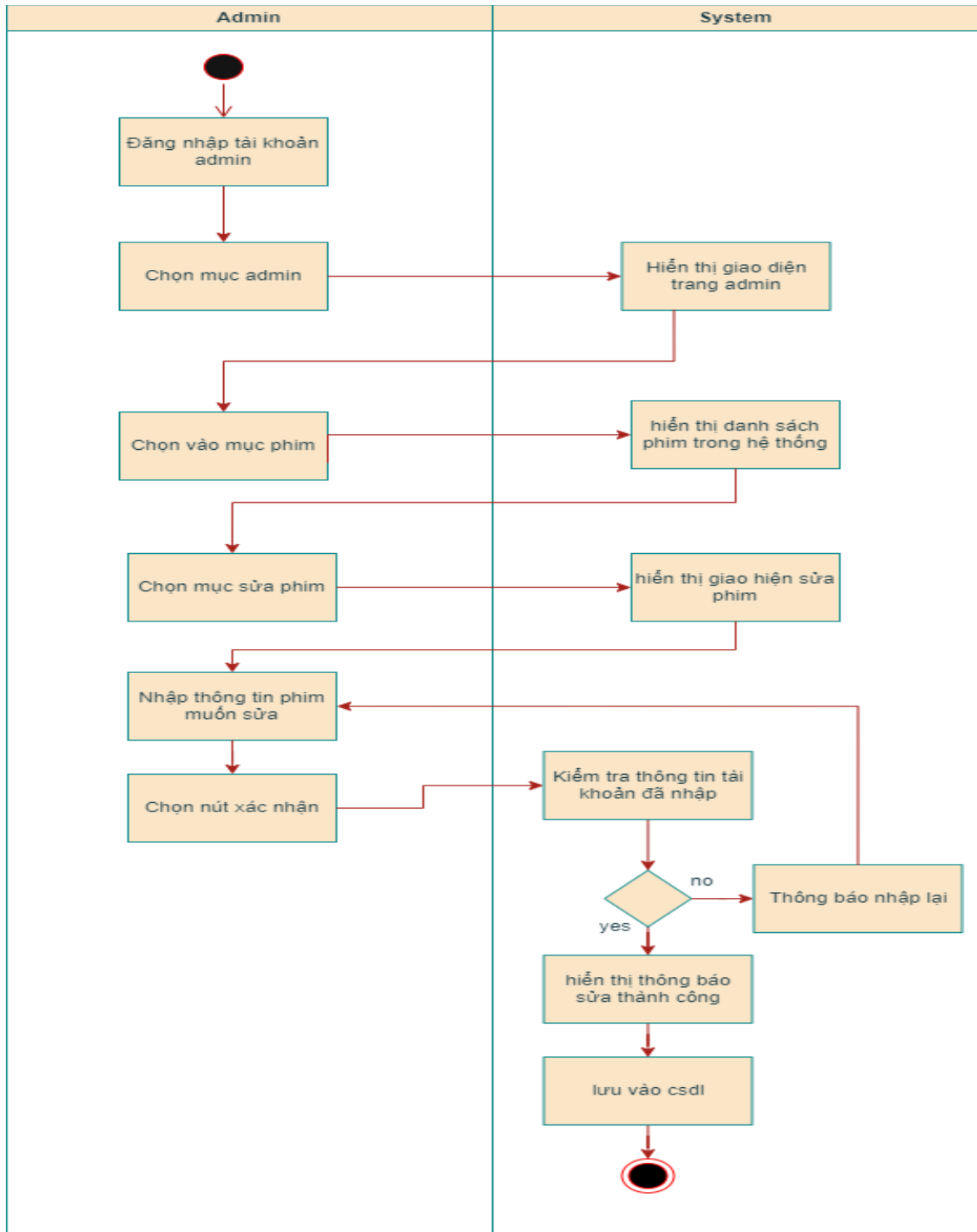


### 3.7.2.3. Xoá phim



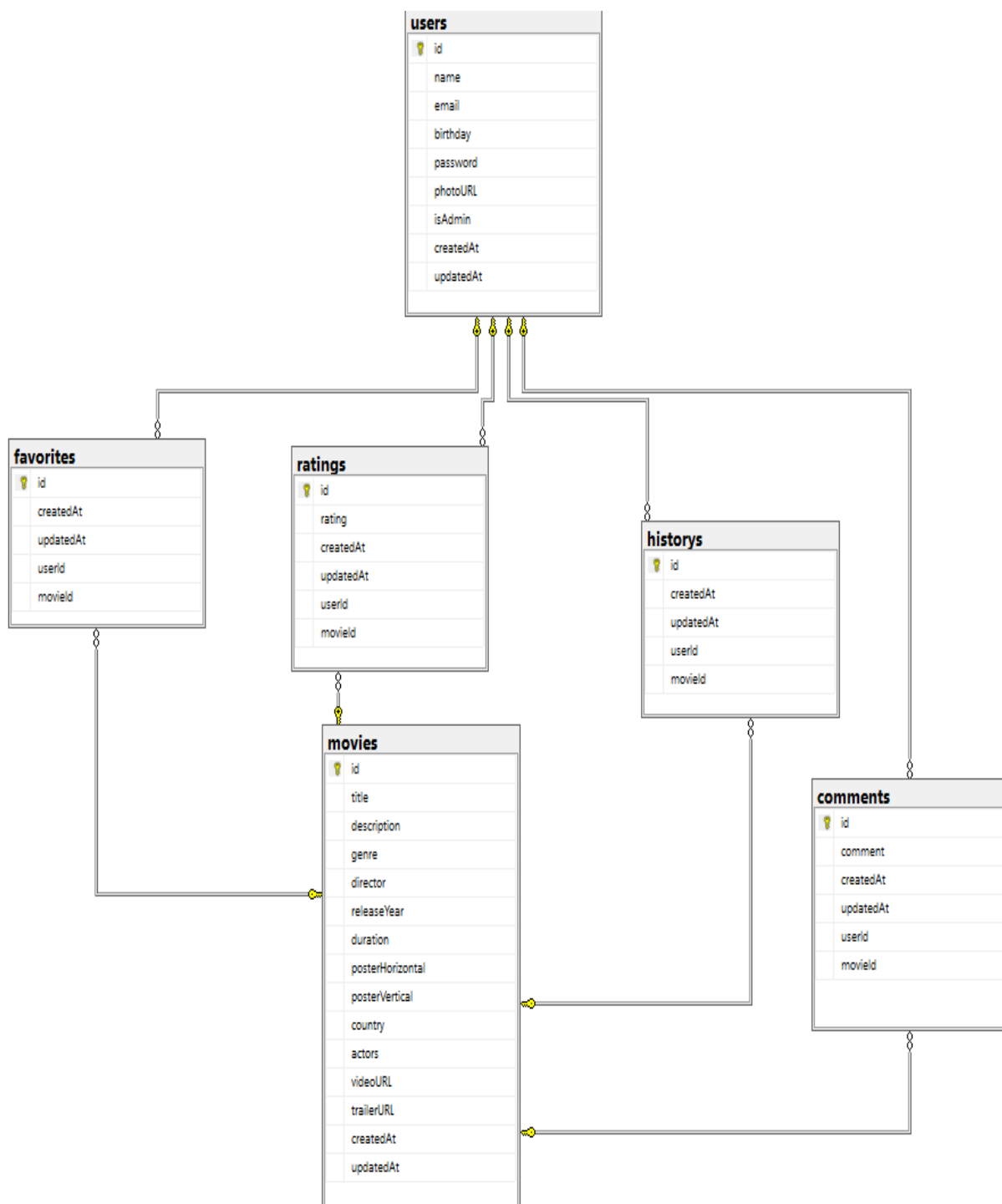
Hình 22. Sơ đồ hoạt động chức năng xoá phim

### 3.7.2.4. Sửa phim



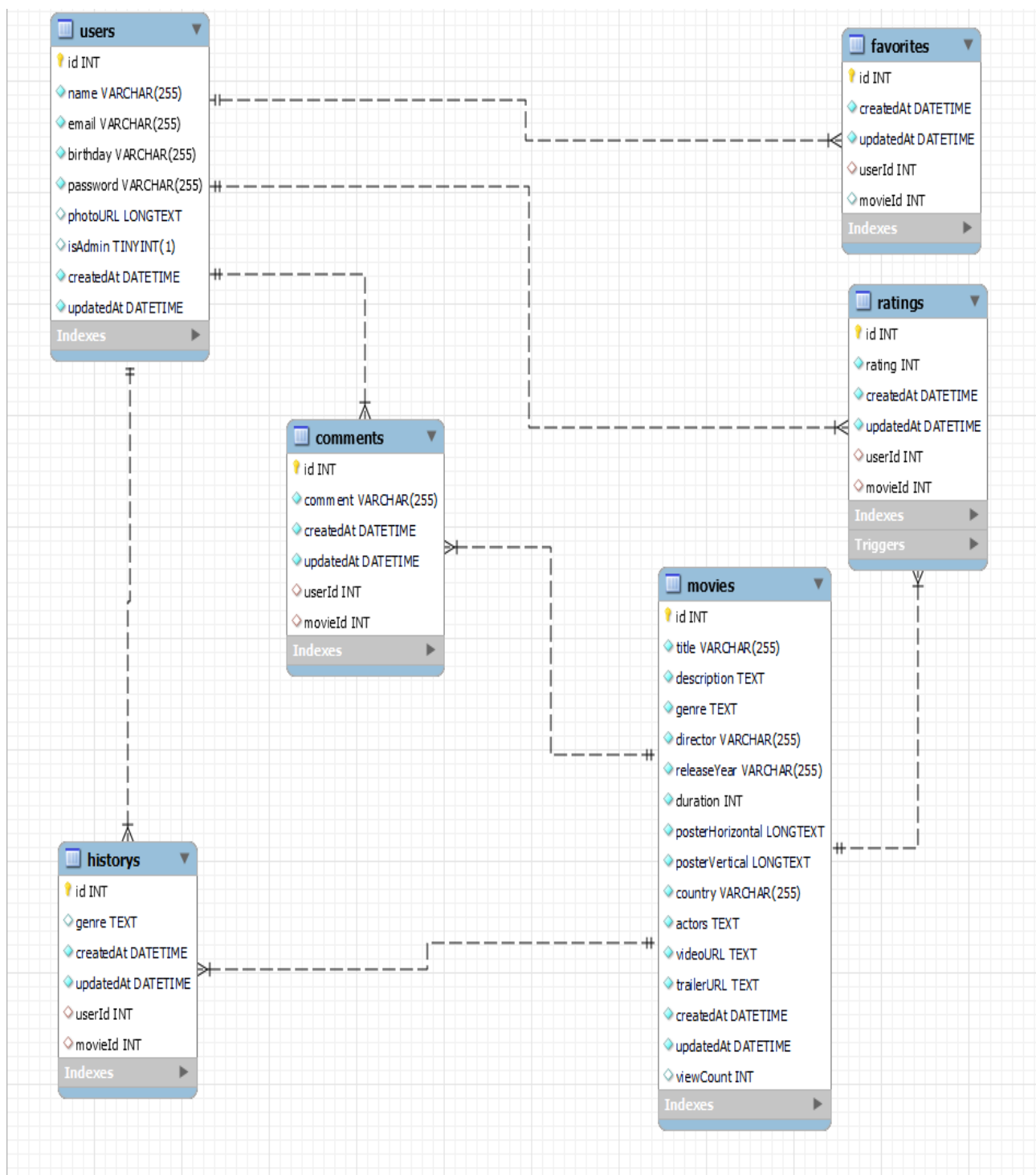
Hình 23. Sơ đồ hoạt động chức năng sửa phim

### 3.8. Sơ đồ Database Diagram



Hình 24. Sơ đồ Diagram

### 3.9. Sơ đồ ClassDiagram



Hình 25. Sơ đồ ClassDiagram

## **CHƯƠNG 4. TRIỂN KHAI ỨNG DỤNG**

### **4.1. Tổng quan ứng dụng**

Ứng dụng xem phim là một loại ứng dụng giải trí phổ biến trên các thiết bị di động, máy tính bảng và smart TV. Ứng dụng này cung cấp cho người dùng các tính năng để tìm kiếm, xem và quản lý các bộ phim, chương trình truyền hình, và nội dung giải trí khác.

Các tính năng chính của ứng dụng xem phim bao gồm:

**Tìm kiếm nội dung:** Người dùng có thể tìm kiếm các bộ phim, chương trình truyền hình, và nội dung giải trí khác bằng cách sử dụng các từ khóa hoặc thể loại.

**Xem phim:** Người dùng có thể xem các bộ phim và chương trình truyền hình với chất lượng cao trên thiết bị của mình. Nhiều ứng dụng cung cấp tính năng xem trực tuyến hoặc tải xuống nội dung để xem ngoại tuyến.

**Đề xuất nội dung:** Các ứng dụng xem phim có thể đề xuất cho người dùng các bộ phim và chương trình truyền hình dựa trên sở thích và lịch sử xem của người dùng.

**Cập nhật thông tin mới nhất:** Người dùng có thể cập nhật các thông tin mới nhất về các bộ phim và chương trình truyền hình, bao gồm thông tin về diễn viên, đạo diễn và nhà sản xuất.

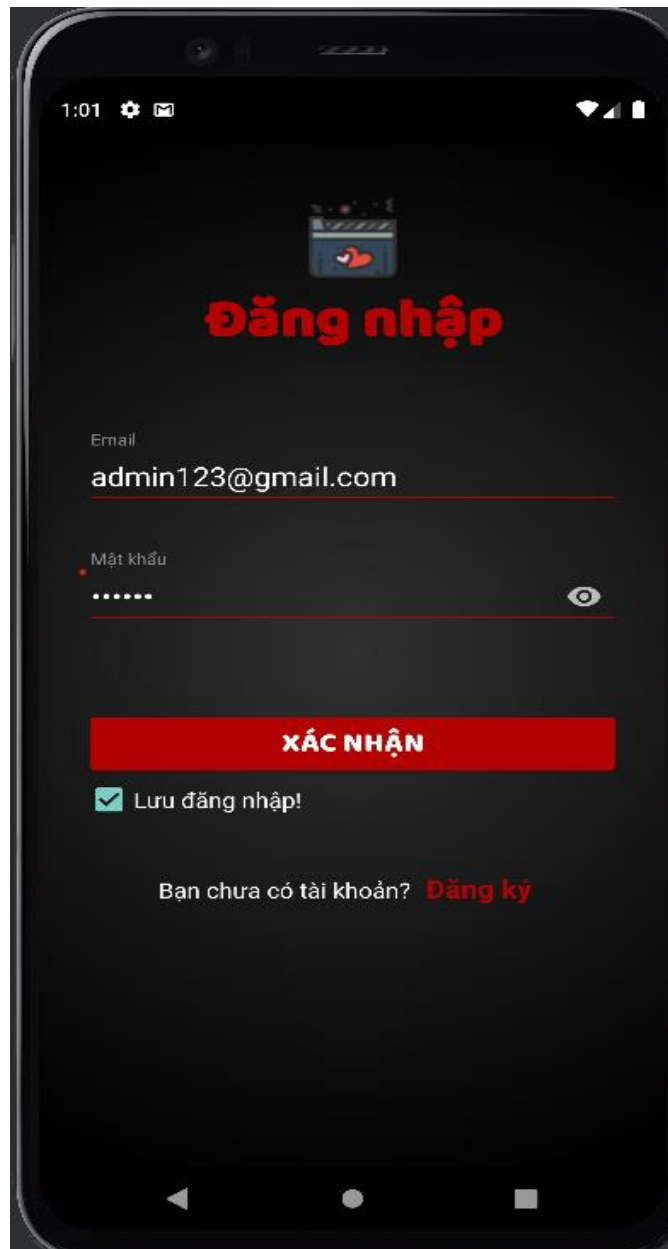
### **4.2. Chương trình demo**

Khi truy cập vào ứng dụng nếu chưa đăng nhập hệ thống sẽ tự điều hướng đến trang đăng nhập.

## 4.2.1. Giao diện chức năng đăng nhập và đăng ký

### 4.2.1.1. Giao diện chức năng đăng nhập

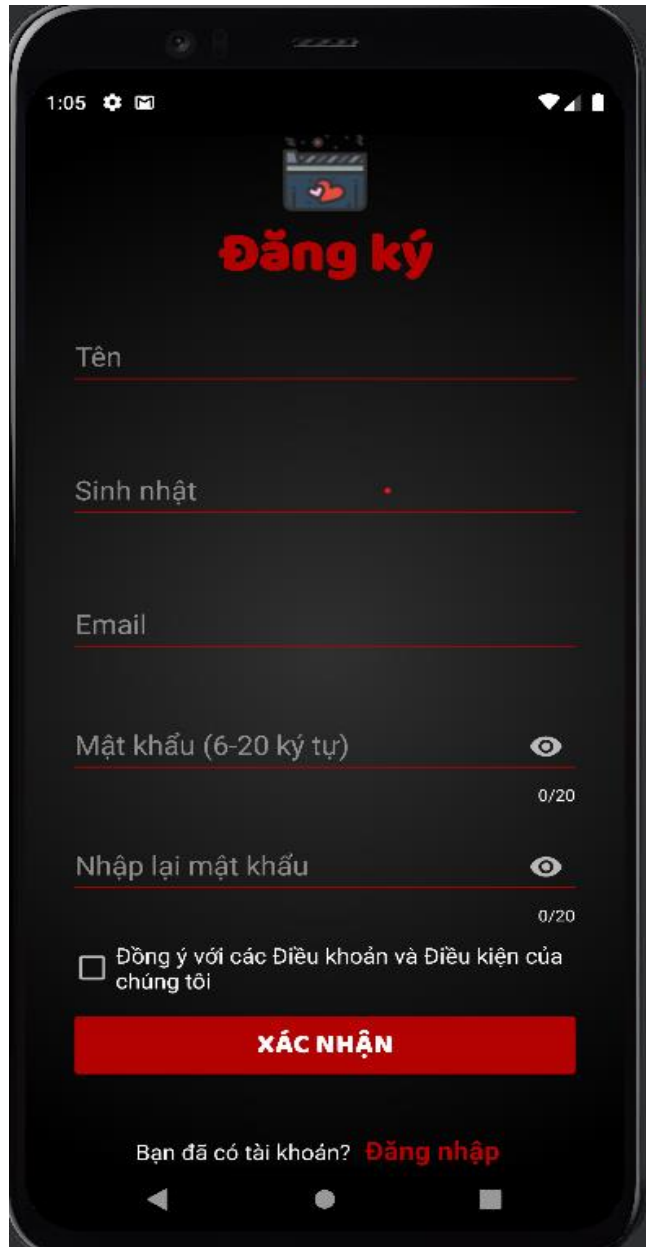
Khi người dùng chạy ứng dụng thì bước đầu tiên để đăng nhập vào ứng dụng thì chúng ta cần phải đăng nhập tài khoản và mật khẩu (nếu đã có tài khoản trước), Nếu chưa có tài khoản thì chúng ta cần phải đăng ký tài khoản để đăng nhập vào ứng dụng.



Hình 26. Giao diện chức năng đăng nhập

#### 4.2.1.2. Giao diện chức năng đăng ký

Với chức năng đăng ký thì người dùng sẽ điền vào những thông tin cần thiết mà ứng dụng đã yêu cầu. Sau khi nhập xong thông tin thì chúng ta tích vào điều khoản và bắt đầu đăng ký tài khoản riêng cho mình. Sau đó ta sẽ tài khoản của mình và bắt đầu đăng nhập vào ứng dụng



1:05

**Đăng ký**

Tên

Sinh nhật

Email

Mật khẩu (6-20 ký tự) 0/20

Nhập lại mật khẩu 0/20

☐ Đồng ý với các Điều khoản và Điều kiện của chúng tôi

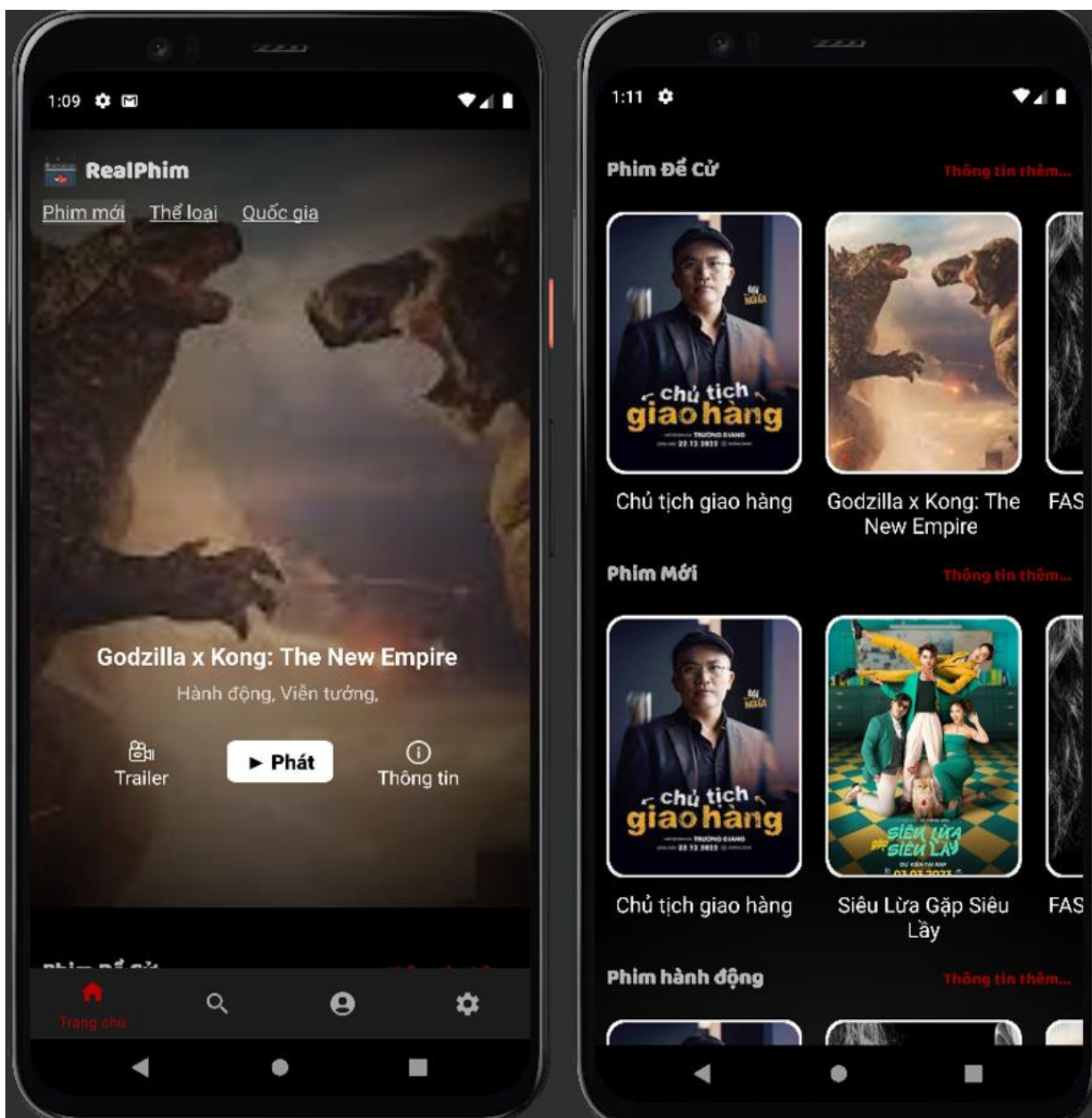
**XÁC NHẬN**

Bạn đã có tài khoản? **Đăng nhập**

Hình 27. Giao diện chức năng đăng ký

#### 4.2.2. Giao diện chức năng trang chủ

Giao diện trang chủ là trang đầu tiên hiển thị cho khách hàng truy cập vào ứng dụng. Tại đây người dùng có thể xem thông tin về các bộ phim, hiển thị danh sách phim mới nhất, phim đề cử và thể loại phim.

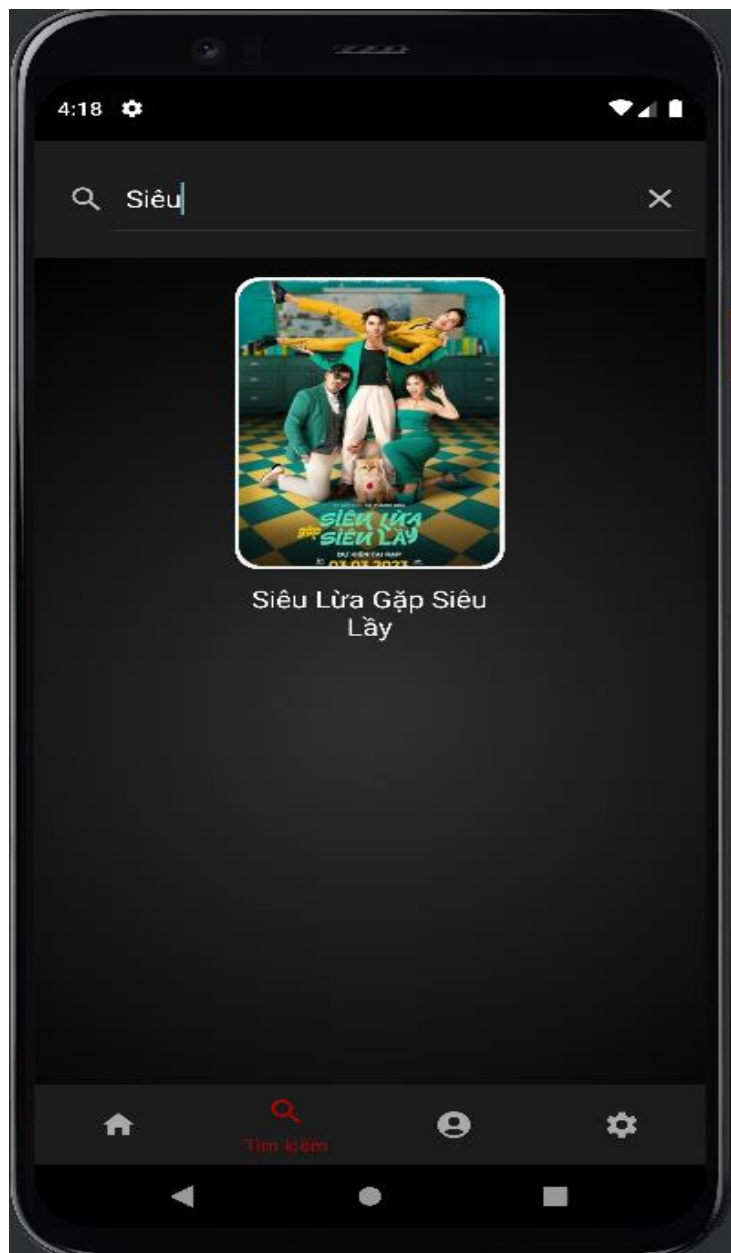


Hình 28. Giao diện trang chủ



### 4.2.3. Giao diện tìm kiếm

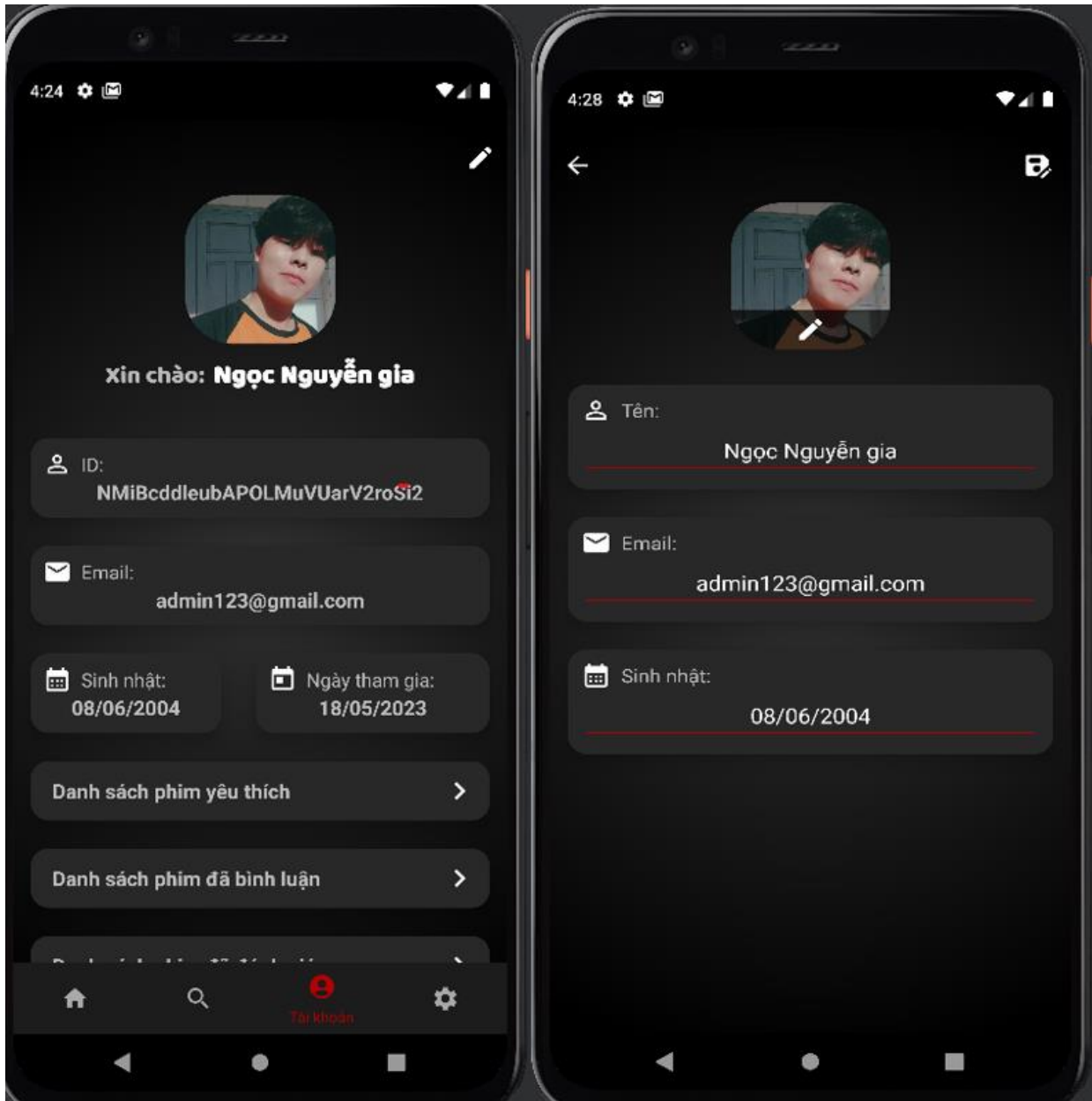
Khi người dùng truy cập vào mục tìm kiếm và bắt đầu tìm kiếm thông tin phim thì hệ thống sẽ hiển thị ra những phim mà người dùng muốn xem



Hình 29. Giao diện tìm kiếm

#### 4.2.4. Giao diện thông tin và chỉnh sửa thông tin người dùng

Với giao diện thông tin người dùng thì người dùng có thể xem lại hồ sơ của mình và chỉnh sửa lại thông tin của mình.



Hình 30. Giao diện thông tin và chỉnh sửa thông tin người dùng

#### 4.2.5. Giao diện thêm phim

Giao diện thêm phim chỉ dùng cho admin thêm một bộ phim mới nhất

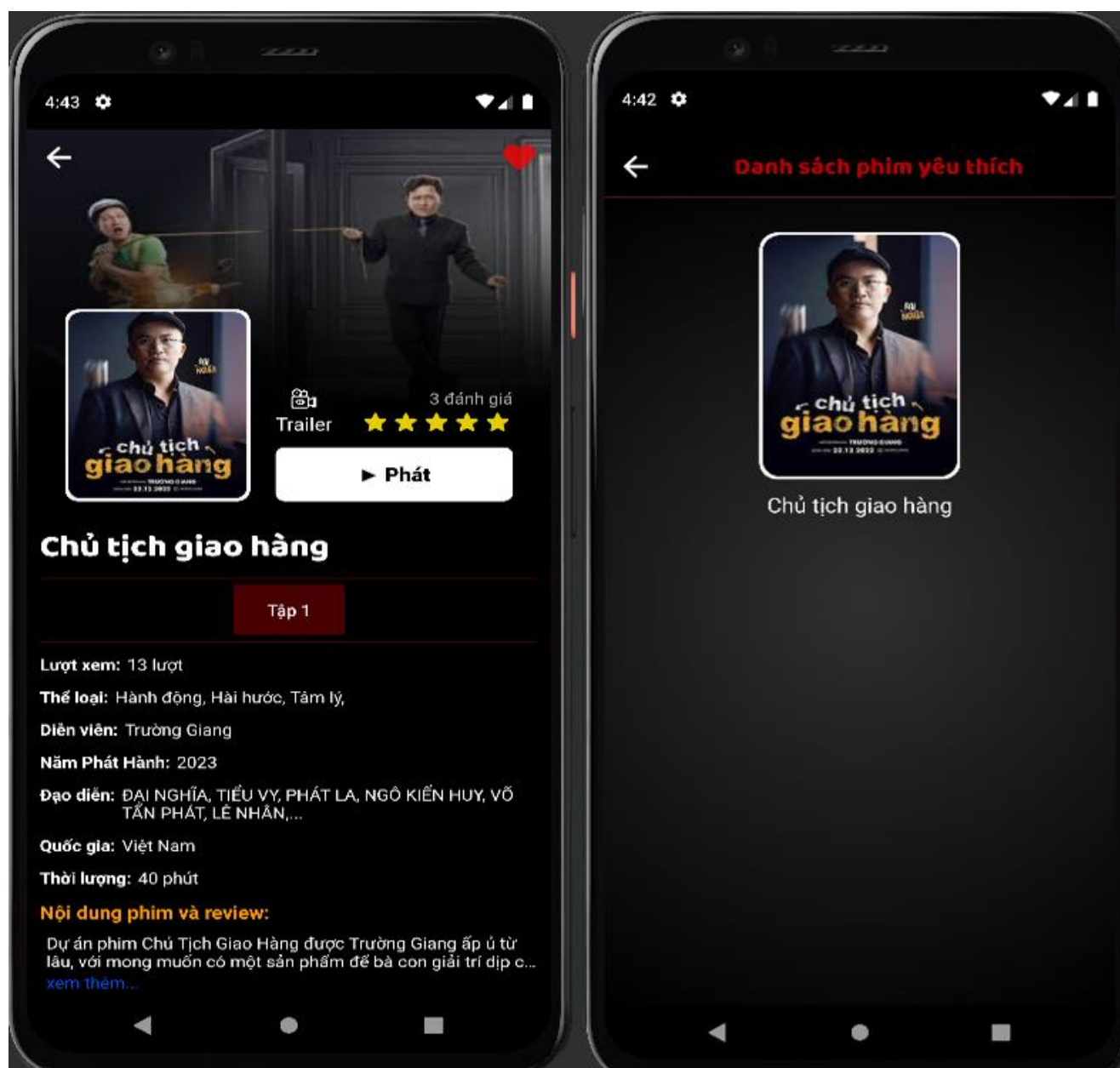
The image displays two side-by-side mobile app screens for adding a new movie. The left screen, timestamped 4:31, features a 'Link phim:' field with a 'Link' input box and '+'/'-' buttons. Below are 'Poster 3x4:' and 'Poster 4x3:' fields with camera icons. A 'Thể loại:' (Genre) section lists various categories with checkboxes: Hành động, Võ thuật, Tình cảm, Hoạt hình, Hải hước, Viễn tưởng, Cổ trang, Phiêu lưu, Tâm lý, Kinh dị, Hình sự, Thần thoại, and Chiến tranh. A 'Tên phim:' field is at the bottom. The right screen, timestamped 4:32, continues the form with fields for 'Diễn viên:', 'Đạo diễn:', 'Thời lượng (Phút):', 'Năm phát hành:', 'Quốc gia:', 'Nội dung phim:', and 'Trailer link:'. A prominent red button labeled 'Thêm' (Add) is at the bottom.

Hình 31. Giao diện thêm phim

## 4.2.6. Giao diện yêu thích, đánh giá, bình luận và danh sách phim yêu thích, đánh giá, bình luận

### 4.2.6.1. Giao diện yêu thích

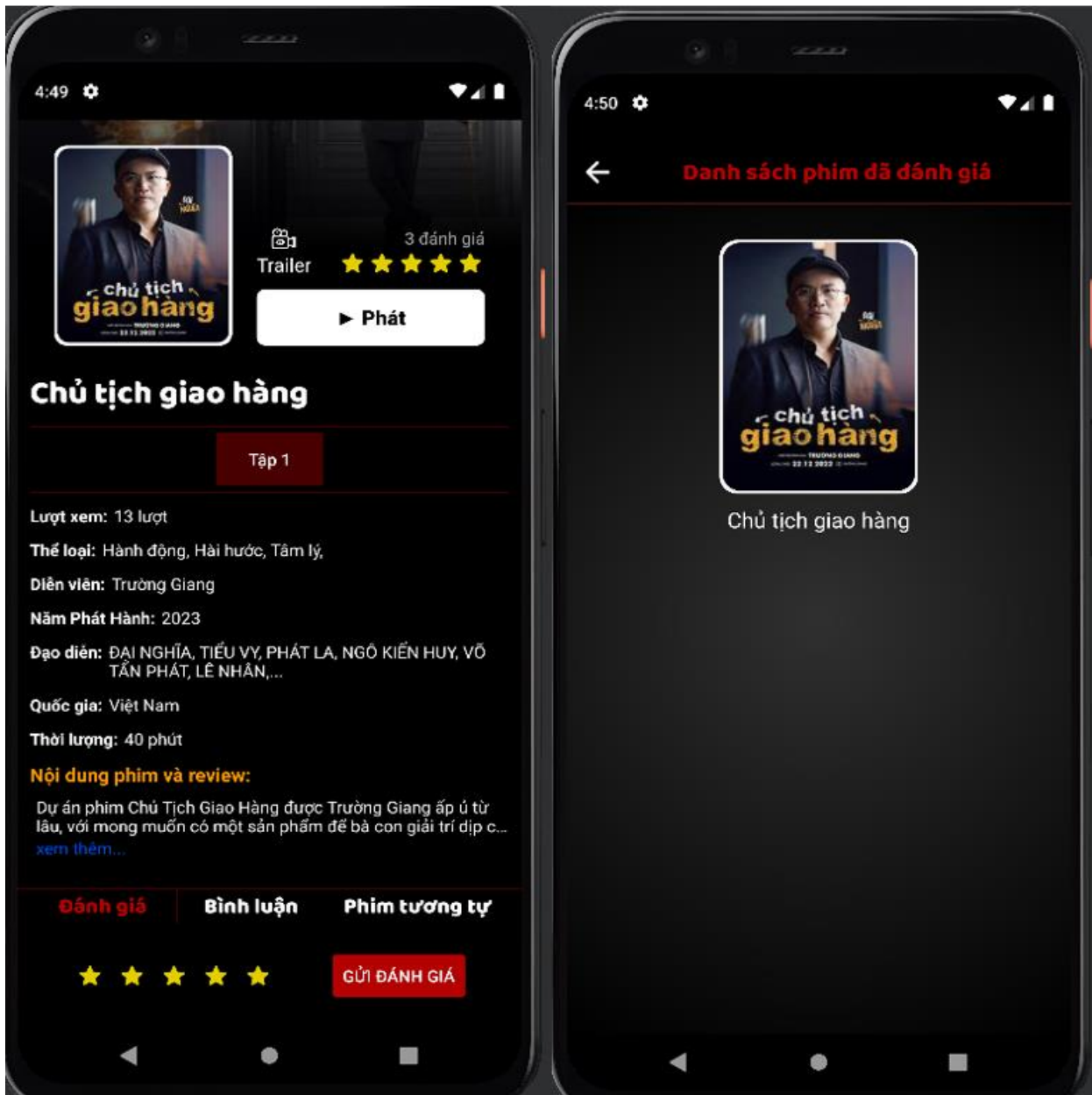
Khi người dùng nhấn vào yêu thích phim thì những bộ phim mà người dùng đã yêu thích sẽ lưu vào danh mục danh sách phim mà mình đã yêu thích



Hình 32. Giao diện yêu thích và danh sách phim yêu thích

#### 4.2.6.2. Giao diện đánh giá và danh sách phim đã đánh giá

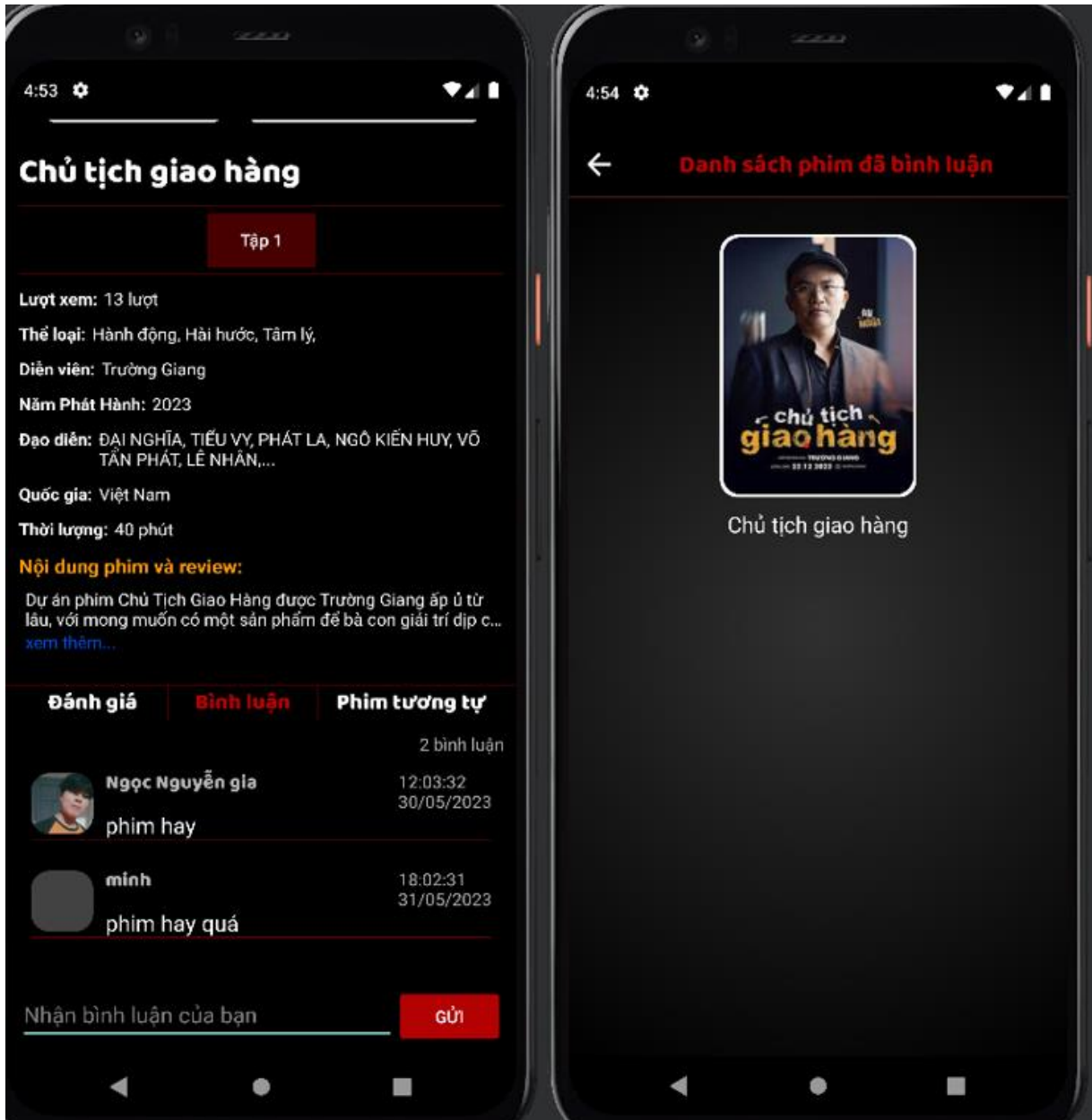
Khi người dùng đánh giá phim thì những bộ phim mà người dùng đã đánh giá sẽ lưu vào danh mục danh sách phim mà mình đã đánh giá



Hình 33. Giao diện yêu đánh giá và danh sách đánh giá

#### 4.2.6.3. Giao diện bình luận và danh sách phim đã bình luận

Khi người dùng bình luận một bộ phim mà mình đã xem thì những bộ phim mà người dùng đã bình luận sẽ lưu vào danh sách các phim mà người dùng đã bình luận

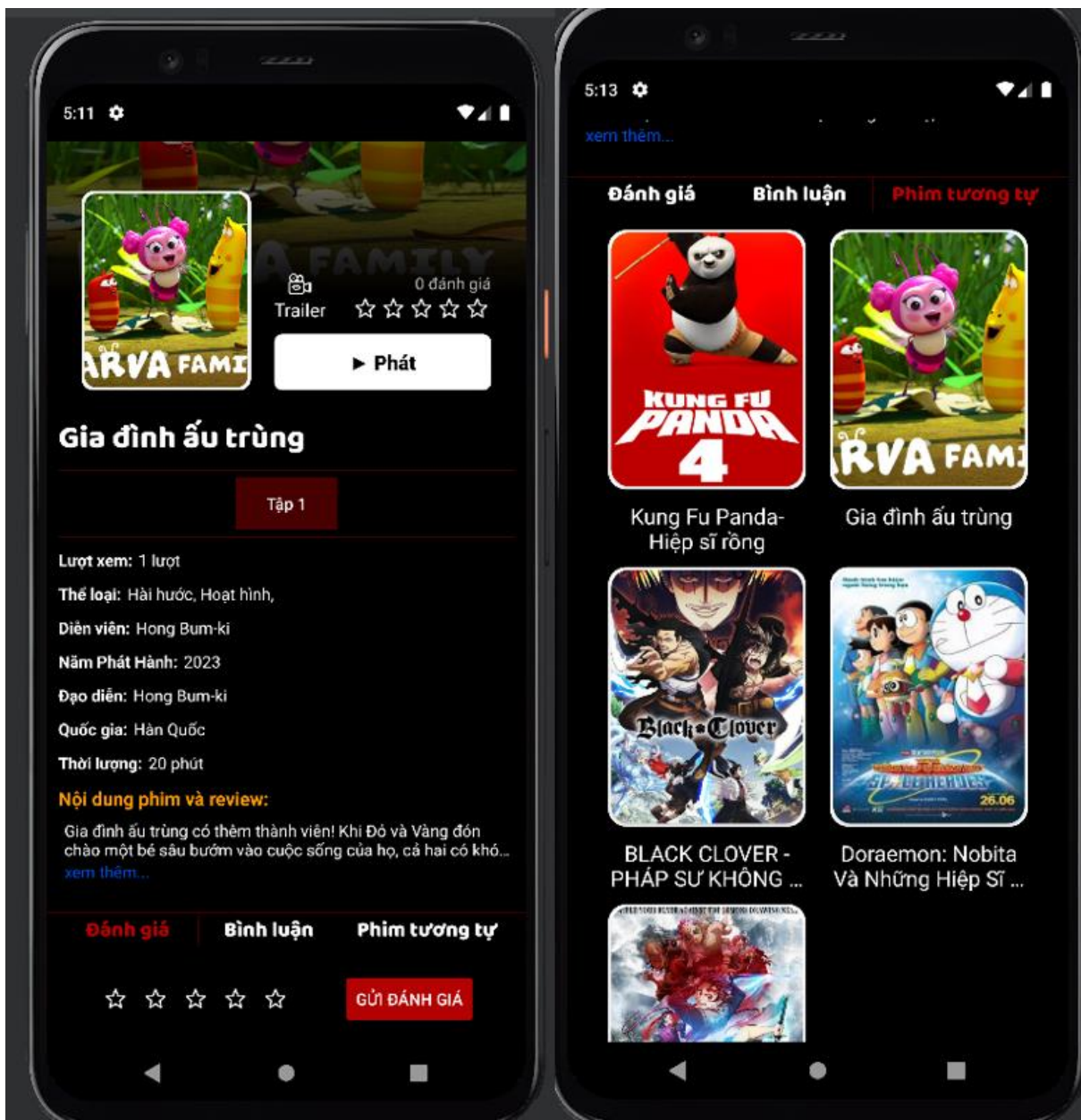


Hình 34. Giao diện bình luận và danh sách phim đã bình luận



#### 4.2.7. Giao diện đề xuất phim

Khi người dùng “click” vào những bộ phim có thể loại như hành động, hoạt hình, tình cảm,... hệ thống sẽ đề xuất cho người dùng những bộ phim tương tự.



Hình 35. Giao diện đề xuất phim

## 5. Giao diện xem phim

Khi người dùng click vào phát phim hoặc click vào các tập phim thì hệ thống sẽ hiển thị ra giao diện như dưới đây và người dùng có thể trải nghiệm những bộ phim mà họ muốn xem



Hình 36. Giao diện xem phim



# KẾT LUẬN VÀ KIẾN NGHỊ

## 1. Kết quả đạt được

Ứng dụng xem phim trực tuyến tuy không phải là đề tài mới nhưng nó sẽ dần trở thành một phần quan trọng trong ứng dụng công nghệ thông tin vào cuộc sống. Tuy nhiên hệ thống vẫn còn đang ở mức độ cơ bản, sơ khai chưa có một chuẩn rõ ràng, hoàn chỉnh. Trong khuôn khổ cho phép và kiến thức kinh nghiệm học trong suốt thời gian qua em đã cố gắng xây dựng hệ thống một cách tổng quan nhất. Qua quá trình khảo sát, đề án “Xây dựng ứng dụng xem phim trực tuyến” đã đạt được những kết quả nhất định.

Tìm hiểu nội dung yêu cầu bài toán và các ứng dụng - Nắm bắt kỹ năng phân tích thiết kế hướng đối tượng với UML, tìm hiểu công nghệ phát triển app.

## 2. Hạn chế

Tuy nhiên, còn nhiều chức năng chương trình còn chưa hoàn thiện, giao diện chưa đẹp, các giao diện còn chưa tường minh. Chưa tối ưu quá trình tìm kiếm, nếu dữ liệu quá lớn thì hệ thống sẽ hoạt động kém hiệu quả và chưa đa dạng các thuộc tính tìm kiếm. Tốc độ chạy của ứng dụng chưa được nhanh, chạy phim vẫn còn thiếu sót ở phần giao diện. Tính xác thực về dữ liệu phim chưa được chính xác.

## 3. Hướng phát triển

- Khắc phục những nhược điểm trên.
- Cải thiện giao diện và trải nghiệm cho người dùng.
- Phát triển thêm ứng dụng có thể đăng kí tài khoản VIP, và thanh toán trực tiếp trên ứng dụng.
- Phát triển ứng dụng chạy trên mọi nền tảng thiết bị di động để nâng cao trải nghiệm của khách hàng, phục vụ nhu cầu sử dụng ứng dụng trên thiết bị di động.

## TÀI LIỆU THAM KHẢO

- [1] "Bill Phillips, Chris Stewart, Kristin Marsicano, và Joe Hummel", "Android Programming: The Big Nerd Ranch Guide (4th Edition)", Big Nerd Ranch Guides, 2019
- [2] "Cay S. Horstmann", "Java Concepts: Late Objects", John Wiley & Sons, 2018
- [3] "Joshua Bloch", "Effective Java", Addison-Wesley Professional, 2018
- [4] "Marko Gargenta và Masumi Nakamura", "Learning Android: Develop Mobile Apps Using Java and Eclipse", O'Reilly Media, 2018
- [5] "Ted Hagos", "Android Programming: The Big Nerd Ranch Guide", Big Nerd Ranch Guides, 2019
- [6] "Dietmar Jannach và Markus Zanker", "Recommender Systems: An Introduction", Cambridge University Press, 2019
- [7] "Robin Burke", "Hybrid Recommender Systems Handbook", Springer, 2019
- [8] "Paul Deitel và Harvey Deitel", "Android 6 for Programmers: An App-Driven Approach", Prentice Hall, 2018
- [9] "Neil Smyth", "Firebase Essentials for Android", Payload Media, 2018
- [10] "Suresh Kumar Gorakala", "Building Recommender Systems: An Introduction", Apress, 2018
- [11] "Charu Aggarwal", "Recommender Systems: The Textbook", Springer, 2018
- [12] Dietmar Jannach, Markus Zanker, Alexander Felfernig, và Gerhard Friedrich, "Recommender Systems: An Introduction," Cambridge University Press, 2020
- [13] Shlomo Berkovsky, Jill Freyne, và Geoffrey Holmes, "Collaborative Filtering Recommender Systems," Springer, 2019.
- [14] Michael D. Ekstrand, John T. Riedl, và Joseph A. Konstan, "Collaborative Filtering Recommender Systems," Now Publishers Inc, 2020
- [15] Julian McAuley và Hrvoje Stojic, "Content-Based and Collaborative Filtering for Personalized Recommendations," CRC Press, 2021.
- [16] Bang Wang, Li Guo, và Tiejian Luo, "Hybrid Recommender Systems: Survey and Experiments," Elsevier, 2020.
- [17] Deepak Agarwal và Bee-Chung Chen, "Statistical Methods for Recommender

Systems," Cambridge University Press, 2018.

- [18] <https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/>
- [19] <https://machinelearningcoban.com/2017/05/24/collaborativefiltering/>
- [20] <https://developer.android.com/studio/write/firebase?hl=vi>
- [21] <https://viblo.asia/p/tim-hieu-so-luoc-ve-firebase-Eb85oeOmZ2G>
- [22] <https://freetuts.net/tai-lieu-hoc-java-android-kotlin-lo-trinh-hoc-android-2243.html>
- [23] <https://www.javatpoint.com/android-tutorial>
- [24] [https://www.tutorialspoint.com/firebase/firebase\\_overview.htm](https://www.tutorialspoint.com/firebase/firebase_overview.htm)
- [25] <https://www.javacodegeeks.com/>