

**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:**

**XÂY DỰNG ỨNG DỤNG XEM PHIM TRỰC TUYẾN VÀ NGHIÊN**  
**CỨU THUẬT TOÁN CONTENT-BASED FILTERING VÀO HỆ**  
**THỐNG ĐỀ XUẤT NỘI DUNG**

Giảng viên hướng dẫn : Ths.Trần Phong Nhã

Sinh viên thực hiện : Nguyễn Gia Ngọc

Lớp : Công nghệ thông tin

Khóa : K60

Mã số sinh viên : 6051071077

**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:**

**XÂY DỰNG ỨNG DỤNG XEM PHIM TRỰC TUYẾN VÀ NGHIÊN**  
**CỨU THUẬT TOÁN CONTENT-BASED FILTERING VÀO HỆ**  
**THỐNG ĐỀ XUẤT NỘI DUNG**

Giảng viên hướng dẫn : Ths.Trần Phong Nhã

Sinh viên thực hiện : Nguyễn Gia Ngọc

Lớp : Công nghệ thông tin

Khóa : K60

Mã số sinh viên : 6051071077

\*\*\*

**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**

- Xây dựng ứng dụng 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 Firebase
- 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 Firebase.

#### **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
- Firebase

#### **b. Công cụ**

- Androids Studio, SDK 32, Máy ảo pixel 6 Pro API 28.
- Firebase

### **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.Trần Phong Nhã

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

Điện thoại: 0981058040

Email: tpnha@st.utc2.edu.vn

**Trưởng BM Công nghệ Thông tin**

**Giáo viên hướng dẫn**

**ThS. Trần Phong Nhã**

**ThS. Trần Phong Nhã**

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 “Xây dựng ứng dụng 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 **Trần Phong Nhã**, người đã hướng dẫn cho tôi những hướng đi, truyền đạt cho em những kiến thức, kỹ năng để tôi 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 Thầy **Trần Phong Nhã** 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 2023

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. Trần Phong Nhã**

# MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>i</b>
<b>NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....</b>	<b>ii</b>
<b>MỤC LỤC .....</b>	<b>iii</b>
<b>DANH MỤC CHỮ VIẾT TẮT .....</b>	<b>vi</b>
<b>DANH MỤC HÌNH VẼ .....</b>	<b>vii</b>
<b>TỔNG QUAN.....</b>	<b>1</b>
<b>Tổng quan đề tài.....</b>	<b>1</b>
<b>Mục tiêu nghiên cứu .....</b>	<b>1</b>
<b>Phạm vi nghiên cứu .....</b>	<b>2</b>
<b>Cách tiếp cận, phương pháp nghiên cứu và thực hiện.....</b>	<b>2</b>
<b>Cấu trúc báo cáo đồ án tốt nghiệp.....</b>	<b>2</b>
<b>CHƯƠNG 1 CƠ SỞ LÝ THUYẾT.....</b>	<b>4</b>
<b>1.1 Giới thiệu về Java trong Android Studio .....</b>	<b>4</b>
1.1.1 Lịch sử phát triển của Java trong Android Studio:.....	4
1.1.2 Đặc điểm của Android.....	4
1.1.3 Kiến trúc của Android .....	5
1.1.4 Môi trường phát triển android studio .....	6
1.1.5 Các tính năng của Android.....	7
<b>1.2 Nội dung cơ bản về Java trong Android .....</b>	<b>8</b>
1.2.1 Đặc điểm của Java trong Android Studio:.....	8
1.2.2 Các ứng dụng của Java trong Android Studio.....	8
1.2.3 Các khái niệm cơ bản về Java .....	9
1.2.4 Lập trình hướng đối tượng trong Java .....	10
1.2.5 Các thành phần trong Android Studio sử dụng Java .....	10



1.2.6 Các tính năng của Java trong Android Studio .....	11
1.2.7 Kết Luận .....	12
<b>1.3 Giới thiệu về Firebase .....</b>	<b>13</b>
1.3.1 Lịch sử phát triển của Firebase.....	13
1.3.2 Đặc điểm của Firebase.....	13
1.3.3 Tính năng của Firebase.....	14
1.3.4 Tính năng nâng cao của Firebase .....	15
1.3.5 Ứng dụng của Firebase .....	15
1.3.6 Kết luận .....	16
<b>CHƯƠNG 2 PHÂN TÍCH THUẬT TOÁN CONTENT BASED FILTERING ..</b>	<b>17</b>
2.1 Giới thiệu về Content-Based Filtering.....	17
2.2 Khái niệm và đặc trưng của Content-Based Filtering.....	17
2.3 Các bước thực hiện của Content-Based Filtering: .....	18
2.4 Cách giải và ví dụ về thuật toán Content-Based Filtering .....	20
2.5 Kết luận .....	23
<b>CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....</b>	<b>24</b>
3.1 Khảo sát và đánh giá hệ thống .....	24
3.2 Quy trình thực hiện quản trị hệ thống của nhân viên quản trị .....	24
3.3 Yêu cầu hệ thống .....	25
3.3.1 Yêu cầu chức năng.....	25
3.3.2 Yêu cầu phi chức năng .....	25
3.3.3 Yêu cầu khác: .....	25
3.4 Sơ đồ phân rã chức năng và luồng dữ liệu.....	25
3.4.1 Yêu cầu người dùng.....	26
3.4.2 Sơ đồ phân rã chức năng.....	26
3.4.3 Sơ đồ luồng dữ liệu.....	27

<b>3.5 Biểu đồ UseCase .....</b>	<b>28</b>
<b>3.6 Sơ đồ ER.....</b>	<b>31</b>
<b>3.7 Sơ đồ hoạt động .....</b>	<b>32</b>
3.7.1 Với người dùng.....	32
3.7.2 Với người quản trị .....	39
<b>3.8 Sơ đồ Database Diagram .....</b>	<b>46</b>
<b>3.9 Sơ đồ ClassDiagram .....</b>	<b>47</b>
<b>CHƯƠNG 4 TRIỂN KHAI CHƯƠNG TRÌNH ỨNG DỤNG XEM PHIM TRỰC TUYÊN .....</b>	<b>48</b>
4.1 Giao diện chức năng đăng nhập: .....	48
4.2 Giao diện chức năng đăng ký .....	49
4.3 Giao diện chức năng trạng chủ .....	50
4.4 Giao diện tìm kiếm .....	51
4.5 Giao diện thông tin và chỉnh sửa thông tin người dùng .....	52
4.6 Giao diện thêm phim.....	53
4.7 Giao diện yêu thích và danh sách phim yêu thích.....	54
4.8 Giao diện đánh giá và danh sách phim đã đánh giá .....	55
4.9 Giao diện bình luận và danh sách phim đã bình luận .....	56
4.10 Giao diện đề xuất phim.....	57
4.11 Giao diện xem phim .....	58
<b>KẾT LUẬN VÀ KIẾN NGHỊ.....</b>	<b>59</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>60</b>

## 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

## DANH MỤC HÌNH VẼ

Hình 1 Sơ đồ phân rã chức năng ứng dụng xem phim .....	26
Hình 2 Sơ đồ phân rã chức năng Quản lý ứng dụng .....	27
Hình 3 Sơ đồ luồng dữ liệu.....	27
Hình 4 Biểu đồ UseCase tổng quát.....	28
Hình 5 UseCase Admin .....	29
Hình 6 UseCase Người dùng.....	30
Hình 7 Sơ đồ ER.....	31
Hình 8 Sơ đồ hoạt động chức năng đăng nhập .....	32
Hình 9 Sơ đồ hoạt động chức năng đăng ký.....	33
Hình 10 Sơ đồ hoạt động chức năng tìm kiếm phim.....	34
Hình 11 Sơ đồ hoạt động chức năng xem chi tiết phim .....	35
Hình 12 Sơ đồ hoạt động chức năng đánh giá phim .....	36
Hình 13 Sơ đồ hoạt động chức năng bình luận phim .....	37
Hình 14 Sơ đồ hoạt động chức năng yêu thích phim .....	38
Hình 15 Sơ đồ hoạt động chức năng tìm kiếm phim.....	39
Hình 16 Sơ đồ hoạt động chức năng thêm phim .....	40
Hình 17 Sơ đồ hoạt động chức năng xóa phim .....	41
Hình 18 Sơ đồ hoạt động chức năng sửa phim.....	42
Hình 19 : Sơ đồ hoạt động chức năng thêm tài khoản .....	43
Hình 20 Sơ đồ hoạt động chức năng sửa tài khoản .....	44
Hình 21 Sơ đồ hoạt động chức năng xóa tài khoản.....	45
Hình 22 Sơ đồ Diagram.....	46
Hình 23 Sơ đồ ClassDiagram .....	47
Hình 24 Giao diện chức năng đăng nhập .....	48
Hình 25 Giao diện chức năng đăng ký .....	49
Hình 26 Giao diện trạng chủ.....	50
Hình 27 Giao diện tìm kiếm .....	51
Hình 28 Giao diện thông tin và chỉnh sửa thông tin người dùng .....	52
Hình 29 Giao diện thêm phim .....	53
Hình 30 Giao diện yêu thích và danh sách phim yêu thích .....	54
Hình 31 Giao diện yêu đánh giá và danh sách đánh giá.....	55
Hình 32 Giao diện bình luận và danh sách phim đã bình luận.....	56
Hình 33 Giao diện đề xuất phim.....	57
Hình 34 Giao diện xem phim .....	58

# TỔNG QUAN

## 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 kỳ 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 kỳ 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. Các ứng dụng này 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, chúng 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 Firebase với những tính năng cơ bản như đăng ký người dùng, xem phim, tìm phim và các phản hồi ý kiến liên quan.

## 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.

## **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 và Firebase để hoàn thành ứng dụng.

## **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.

## **Cấu trúc báo cáo đồ án tốt nghiệp**

### **Tổng quan**

- Tổng quan về đề tài đồ án tốt nghiệp.
- Nêu ra mục tiêu và phạm vi nghiên cứu.
- Cách tiếp cận đề tài, phương pháp nghiên cứu và thực hiện.

### **Chương 1: Cơ sở lý thuyết**

- Giới thiệu về các ngôn ngữ đã sử dụng.
- Giới thiệu về framework.
- Các kỹ thuật và thư viện đã sử dụng trong quá trình thực hiện đề tài

### **Chương 2: Phân tích thuật toán content based filtering comender**

- Giới thiệu
- Các bước thực hiện thuật toán Content-Based Filtering

- Các phương pháp tiếp cận trong Content-Based Filtering
- Cách giải và ví dụ về thuật toán Content-Based Filtering
- Kết luận

### **Chương 3: Phân tích và thiết kế**

- Yêu cầu hệ thống.
- Sơ đồ rõ chức năng.
- Sơ đồ luồng dữ liệu.
- Sơ đồ usecase.
- Sơ đồ hoạt động.
- Mô hình ERD.
- Cơ sở dữ liệu.

### **Chương 4: Triển khai chương trình**

- Giao diện ứng dụng.

### **Kết quả - Kiến nghị**

- Kết luận.
- Tài liệu tham khảo.

# 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 mạnh mẽ: 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.

### **1.1.3 Kiến trúc của Android**

Kiến trúc của Android là một hệ thống phần mềm được phát triển bởi Google để quản lý và điều khiển các thiết bị di động chạy hệ điều hành Android. Kiến trúc của Android bao gồm các thành phần chính như sau:

- Kernel: Là thành phần cốt lõi của hệ điều hành, quản lý tài nguyên phần cứng như bộ nhớ, CPU, thiết bị ngoại vi và kết nối mạng.
- Thư viện Android: Bao gồm các thư viện cơ bản để xử lý các tác vụ như đồ họa, âm thanh, video, mạng, v.v.
- Runtime: Là một môi trường thực thi ứng dụng, hỗ trợ các ngôn ngữ lập trình như Java hoặc Kotlin.

- Lớp ứng dụng: Là các ứng dụng được phát triển bởi các nhà phát triển, chạy trên nền tảng Android và có thể truy cập vào các tài nguyên hệ thống như cơ sở dữ liệu và thiết bị ngoại vi.
- Framework: Cung cấp các API cho các lớp ứng dụng để tương tác với các tài nguyên hệ thống và các dịch vụ khác nhau.
- Services: Các dịch vụ chạy dưới nền, được sử dụng để thực hiện các tác vụ độc lập với các ứng dụng đang chạy.
- Content Providers: Cung cấp cơ chế để truy cập và chia sẻ dữ liệu giữa các ứng dụng.
- Broadcast Receivers: Nhận các thông báo từ hệ thống hoặc các ứng dụng khác và thực hiện các tác vụ tương ứng.
- Activities: Là các thành phần của ứng dụng, cung cấp giao diện người dùng cho người dùng tương tác với ứng dụng.
- Intents: Là các thông báo giữa các thành phần khác nhau của hệ thống, được sử dụng để kích hoạt các hoạt động hoặc truyền dữ liệu giữa các thành phần khác nhau.

#### **1.1.4 Môi trường phát triển android studio**

Để chuẩn bị môi trường phát triển Android Studio thực hiện các bước theo các bước sau:

- Tải và cài đặt Android Studio: Truy cập vào trang web chính thức của Android Studio và tải xuống phiên bản phù hợp với hệ điều hành của bạn. Sau đó, cài đặt Android Studio trên máy tính của bạn.
- Cài đặt JDK: Android Studio yêu cầu JDK (Java Development Kit) để chạy. Nếu bạn chưa cài đặt JDK trên máy tính của mình, bạn có thể tải xuống phiên bản mới nhất từ trang web của Oracle và cài đặt nó.
- Cập nhật Android SDK: Android SDK là bộ công cụ phát triển ứng dụng Android và nó bao gồm những gì bạn cần để phát triển ứng dụng Android. Bạn có thể cập nhật Android SDK bằng cách mở Android Studio và chọn menu "SDK Manager". Tại đây, bạn có thể cập nhật Android SDK và các công cụ phát triển khác.
- Tạo Android Virtual Device (AVD): AVD là một máy ảo Android được sử dụng để kiểm tra ứng dụng của bạn trên nhiều thiết bị khác nhau. Để tạo AVD, mở

Android Studio và chọn menu "AVD Manager". Tại đây, bạn có thể tạo một AVD mới hoặc sửa đổi một AVD hiện có.

- Cập nhật các plugin cần thiết: Android Studio hỗ trợ nhiều plugin để giúp bạn phát triển ứng dụng Android. Bạn có thể cập nhật các plugin cần thiết bằng cách mở Android Studio và chọn menu "Preferences" (trên macOS) hoặc "Settings" (trên Windows và Linux), sau đó chọn "Plugins".

### **1.1.5 Các tính năng của Android**

Android là một hệ điều hành phổ biến trên các thiết bị di động, được phát triển bởi Google và có nhiều tính năng hữu ích cho người dùng và nhà phát triển. Dưới đây là một số tính năng chính của Android:

- Đa nhiệm: Android cho phép người dùng sử dụng nhiều ứng dụng cùng một lúc và chuyển đổi giữa chúng dễ dàng. Người dùng có thể chạy các ứng dụng ngay cả khi họ đang sử dụng ứng dụng khác.
- Giao diện tùy chỉnh: Android cho phép người dùng tùy chỉnh giao diện theo ý muốn, từ hình nền cho đến bố cục các ứng dụng.
- 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.
- 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 kết nối mạng: Android hỗ trợ nhiều loại kết nối mạng như Wi-Fi, 3G, 4G, Bluetooth, NFC, v.v.
- Đa nền tảng: Android là một hệ điều hành đa nền tảng, hỗ trợ cho nhiều loại thiết bị di động khác nhau.
- Bảo mật: Android được thiết kế để đảm bảo tính bảo mật cho người dùng và các dữ liệu của họ. Nó cung cấp các tính năng bảo mật như mã hóa dữ liệu, chế độ khóa màn hình, quản lý quyền truy cập ứng dụng, v.v.
- Hỗ trợ đa phương tiện: Android hỗ trợ nhiều định dạng âm thanh, video và hình ảnh khác nhau.
- Công nghệ NFC: Android hỗ trợ công nghệ NFC, cho phép người dùng thực hiện

các giao dịch thanh toán và truyền tải dữ liệu đến các thiết bị khác.

- Tối ưu hoá pin: Android cung cấp các tính năng tối ưu hoá pin để giúp tiết kiệm pin và tăng tuổi thọ của pin.

## **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 Các 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 Các 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. Dưới đây là một số khái niệm cơ bản về lập trình hướng đối tượng trong Java:

- 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 Các 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.

### **1.2.6 Các tính năng của Java trong Android Studio**

Java là một ngôn ngữ lập trình quan trọng trong Android Studio vì nó cung cấp nhiều tính năng để phát triển ứng dụng Android, bao gồm:

### **Tính năng đa nền tảng:**

Java là một ngôn ngữ đa nền tảng, tức là nó có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm cả hệ điều hành Android. Điều này cho phép các nhà phát triển viết mã một lần và sử dụng nó trên nhiều nền tảng khác nhau.

### **Tính năng bảo mật:**

Java cung cấp các tính năng bảo mật cao, bao gồm cơ chế kiểm soát truy cập và kiểm soát dữ liệu. Các tính năng này giúp bảo vệ ứng dụng Android khỏi các cuộc tấn công và đảm bảo rằng dữ liệu của người dùng được bảo vệ.

### **Tính năng độc lập với thiết bị:**

Java được thiết kế để chạy độc lập với thiết bị nên các ứng dụng Java có thể chạy trên nhiều loại thiết bị khác nhau mà không cần sửa đổi mã nguồn.

### **Tính năng hỗ trợ đa luồng:**

Java cung cấp tính năng đa luồng, cho phép các nhà phát triển tạo và quản lý các luồng khác nhau trong ứng dụng.

## **1.2.7 Kết Luận**

Android Studio là một công cụ phát triển ứng dụng di động chuyên nghiệp được phát triển bởi Google và sử dụng rộng rãi trong cộng đồng phát triển ứng dụng Android. Nó cung cấp một loạt các tính năng và công cụ để giúp nhà phát triển xây dựng các ứng dụng Android chất lượng cao và hiệu quả.

Một số tính năng nổi bật của Android Studio bao gồm:

- Hỗ trợ lập trình Kotlin và Java.
- Các công cụ tạo giao diện người dùng mạnh mẽ và dễ sử dụng.
- Hỗ trợ để phát triển ứng dụng Android cho nhiều thiết bị khác nhau.
- Các công cụ tối ưu hóa hiệu suất để giúp tăng tốc độ xây dựng và chạy ứng dụng.
- Các công cụ kiểm thử tích hợp và hỗ trợ cho kiểm thử tự động.
- Hỗ trợ tích hợp cho các dịch vụ của Google như Google Maps và Firebase.



## 1.3 Giới thiệu về Firebase

### 1.3.1 Lịch sử phát triển của Firebase

Firebase được thành lập năm 2011 bởi James Tamplin và Andrew Lee. Ban đầu, Firebase là một nền tảng cho phép các nhà phát triển xây dựng ứng dụng di động nhanh chóng và dễ dàng bằng cách cung cấp các công cụ để lưu trữ và đồng bộ dữ liệu.

Năm 2014, Firebase đã được mua lại bởi Google với mục đích cung cấp cho các nhà phát triển các công cụ và dịch vụ để xây dựng các ứng dụng di động và web hiệu quả. Sau đó, Firebase đã trở thành một phần của Google Cloud Platform.

Kể từ khi được mua lại, Firebase đã phát triển và bổ sung thêm nhiều tính năng mới, ví dụ như Cloud Firestore, Authentication, Cloud Functions, và nhiều dịch vụ khác. Firebase cũng đã hướng đến cộng đồng nhà phát triển bằng cách cung cấp các tài liệu, hướng dẫn và hỗ trợ để giúp họ sử dụng Firebase một cách dễ dàng và hiệu quả.

Hiện nay, Firebase được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng di động và web, từ các công ty khởi nghiệp đến các tập đoàn lớn. Nó đã trở thành một trong những nền tảng phát triển ứng dụng di động hàng đầu trên thế giới.

### 1.3.2 Đặc điểm của Firebase

- Dễ sử dụng: Firebase cung cấp các công cụ và dịch vụ để giúp nhà phát triển xây dựng, triển khai và quản lý các ứng dụng di động và web một cách dễ dàng và nhanh chóng.
- Đồng bộ thời gian thực: Firebase hỗ trợ đồng bộ thời gian thực, cho phép các ứng dụng di động và web hiển thị dữ liệu mới nhất và cập nhật tức thì.
- Đám mây linh hoạt: Firebase là một dịch vụ đám mây, cho phép nhà phát triển mở rộng dịch vụ của họ một cách linh hoạt và hiệu quả.
- Tích hợp tốt với các nền tảng khác: Firebase tích hợp tốt với các nền tảng khác, ví dụ như Google Cloud Platform, Android, iOS, Unity, JavaScript và Node.js.
- Tính năng phong phú: Firebase cung cấp nhiều tính năng và dịch vụ như Realtime Database, Cloud Firestore, Authentication, Cloud Storage, Cloud Functions, Firebase Hosting, Firebase Analytics, Firebase Performance Monitoring, Firebase Remote Config và Firebase Test Lab.
- Hỗ trợ đa nền tảng: Firebase hỗ trợ đa nền tảng, cho phép nhà phát triển xây dựng

các ứng dụng di động và web trên nhiều nền tảng khác nhau.

- Hỗ trợ cộng đồng: Firebase hỗ trợ một cộng đồng phát triển rộng lớn, cung cấp các tài liệu, hướng dẫn và hỗ trợ để giúp nhà phát triển sử dụng Firebase một cách dễ dàng và hiệu quả.

### **1.3.3 Tính năng của Firebase**

Firebase cung cấp nhiều tính năng để giúp nhà phát triển xây dựng các ứng dụng di động và web hiệu quả, bao gồm:

- Realtime Database: Cơ sở dữ liệu thời gian thực cho phép các ứng dụng đồng bộ dữ liệu và hiển thị nội dung mới nhất.
- Cloud Firestore: Cơ sở dữ liệu NoSQL linh hoạt, cho phép các nhà phát triển lưu trữ và truy vấn dữ liệu hiệu quả.
- Authentication: Dịch vụ xác thực người dùng cho phép các nhà phát triển quản lý đăng nhập, đăng ký và xác thực người dùng.
- Cloud Storage: Dịch vụ lưu trữ đám mây cho phép các nhà phát triển lưu trữ các tệp và dữ liệu của ứng dụng.
- Cloud Functions: Dịch vụ tính toán đám mây cho phép các nhà phát triển chạy mã tùy chỉnh được kích hoạt bởi các sự kiện Firebase.
- Hosting: Dịch vụ hosting cho phép các nhà phát triển phát triển và triển khai các ứng dụng của họ trên nền tảng Firebase.
- Firebase Analytics: Dịch vụ phân tích để giúp các nhà phát triển theo dõi và phân tích thông tin về người dùng và hoạt động của ứng dụng.
- Firebase Performance Monitoring: Dịch vụ giám sát hiệu suất để giúp các nhà phát triển theo dõi và cải thiện hiệu suất của ứng dụng.
- Firebase Remote Config: Dịch vụ cấu hình từ xa cho phép các nhà phát triển thay đổi cấu hình ứng dụng mà không cần phải phát hành lại ứng dụng.
- Firebase Test Lab: Dịch vụ kiểm tra và thử nghiệm để giúp các nhà phát triển kiểm tra và đảm bảo chất lượng của các ứng dụng của họ trước khi triển khai.

### 1.3.4 Tính năng nâng cao của Firebase

Firebase ML: Dịch vụ học máy để giúp các nhà phát triển phát triển các tính năng học máy trong ứng dụng của họ.

- Firebase A/B Testing: Dịch vụ kiểm tra A/B để giúp các nhà phát triển thử nghiệm và đánh giá các tính năng của ứng dụng với các nhóm người dùng khác nhau.
- Firebase Dynamic Links: Dịch vụ tạo liên kết động cho phép các nhà phát triển tạo liên kết chuyển hướng đến nội dung cụ thể trong ứng dụng của họ.
- Firebase In-App Messaging: Dịch vụ tin nhắn trong ứng dụng cho phép các nhà phát triển gửi tin nhắn tới người dùng trong ứng dụng của họ.
- Firebase Predictions: Dịch vụ dự đoán để giúp các nhà phát triển dự đoán hành vi người dùng và cải thiện trải nghiệm của họ.
- Firebase Crashlytics: Dịch vụ phân tích lỗi để giúp các nhà phát triển phát hiện và sửa lỗi trong ứng dụng của họ.
- Firebase Cloud Messaging: Dịch vụ gửi thông báo đám mây cho phép các nhà phát triển gửi thông báo đến người dùng trong ứng dụng của họ.
- Firebase App Indexing: Dịch vụ đánh chỉ mục ứng dụng cho phép các nhà phát triển đánh chỉ mục các nội dung trong ứng dụng của họ để hiển thị trong kết quả tìm kiếm của Google.
- Firebase Extensions: Các tiện ích mở rộng cho phép các nhà phát triển mở rộng tính năng của Firebase một cách dễ dàng và nhanh chóng.
- Firebase Security Rules: Các quy tắc bảo mật cho phép các nhà phát triển định nghĩa quyền truy cập dữ liệu trong ứng dụng của họ.

### 1.3.5 Ứng dụng của Firebase

- Xây dựng ứng dụng di động và web: Firebase cung cấp các công cụ và dịch vụ để giúp nhà phát triển xây dựng các ứng dụng di động và web hiệu quả, bao gồm Realtime Database, Cloud Firestore, Authentication, Cloud Storage, Cloud Functions, và nhiều dịch vụ khác.
- Quản lý dữ liệu: Firebase cung cấp các dịch vụ để lưu trữ, đồng bộ và quản lý dữ liệu, giúp các nhà phát triển xây dựng các ứng dụng đồng bộ thời gian thực và linh hoạt.

- Xác thực người dùng: Firebase cung cấp dịch vụ xác thực người dùng, cho phép các nhà phát triển xác thực người dùng và quản lý đăng nhập, đăng ký và xác thực người dùng.
- Lưu trữ đám mây: Firebase cung cấp dịch vụ lưu trữ đám mây để lưu trữ các tệp và dữ liệu của ứng dụng.
- Tính toán đám mây: Firebase cung cấp dịch vụ tính toán đám mây cho phép nhà phát triển chạy mã tùy chỉnh được kích hoạt bởi các sự kiện Firebase.
- Phân tích và giám sát hiệu suất: Firebase cung cấp dịch vụ phân tích và giám sát hiệu suất để giúp các nhà phát triển theo dõi và cải thiện hiệu suất của ứng dụng.
- Hosting: Firebase cung cấp dịch vụ hosting để cho phép các nhà phát triển phát triển và triển khai các ứng dụng của họ trên nền tảng Firebase.
- Kiểm tra và thử nghiệm: Firebase cung cấp dịch vụ kiểm tra và thử nghiệm để giúp các nhà phát triển kiểm tra và đảm bảo chất lượng của các ứng dụng của họ trước khi triển khai.

### **1.3.6 Kết luận**

Tổng kết lại, Firebase là một nền tảng đám mây tích hợp cung cấp các dịch vụ và công cụ để giúp nhà phát triển xây dựng, triển khai và quản lý các ứng dụng di động và web một cách dễ dàng và hiệu quả. Firebase cung cấp nhiều tính năng và dịch vụ như Realtime Database, Cloud Firestore, Authentication, Cloud Storage, Cloud Functions, Firebase Hosting, Firebase Analytics, Firebase Performance Monitoring, Firebase Remote Config và Firebase Test Lab. Ngoài ra, Firebase còn có các tính năng nâng cao như Firebase ML, Firebase A/B Testing, Firebase Dynamic Links, Firebase In-App Messaging, Firebase Predictions, Firebase Crashlytics, Firebase Cloud Messaging, Firebase App Indexing, Firebase Extensions và Firebase Security Rules. Sử dụng Firebase sẽ giúp các nhà phát triển tiết kiệm thời gian và nỗ lực trong việc xây dựng, triển khai và quản lý các ứng dụng của họ.

## CHƯƠNG 2 PHÂN TÍCH THUẬT TOÁN 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.

Sau khi đã xác định các đặc trưng của các mục, Content-Based Filtering sử dụng các phương pháp đo độ tương đồng để so sánh các mục với nhau và đưa ra các khuyến nghị dựa trên các mục có nội dung tương đồng nhất với các mục đã được người dùng đánh giá hoặc xem trước đó.

Content-Based Filtering thường được sử dụng trong các hệ thống khuyến nghị cho các sản phẩm, 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.

### 2.2 Khái niệm và đặc trưng của Content-Based Filtering

#### 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.

#### 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.

### **2.3 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 Trees, v.v.
- 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.

### **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.

### **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.

## 2.4 Cách giải và ví dụ về thuật toán Content-Based Filtering

### Cách giải về thuật toán Content-Based Filtering

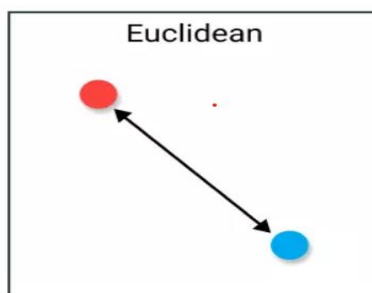
**Xây dựng mô hình đặc trưng:** Đầu tiên, chúng ta cần xây dựng một mô hình để trích xuất các đặc trưng của các mục trong hệ thống khuyến nghị. Điều này có thể được thực hiện bằng cách sử dụng các phương pháp phân tích nội dung như xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), trích xuất 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. Các phương pháp thường được sử dụng để đánh giá độ tương đồng bao gồm đo độ tương đồng cosine, độ tương đồng Jaccard và khoảng cách Euclid

**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. Điều này có thể được thực hiện bằng cách sử dụng các thuật toán phù hợp để lọc và sắp xếp các mục.

**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.

**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.



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



- 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:

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

- 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:

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

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:**

- 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:**
  - **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.

### **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.

## 2.5 Kết luận

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

Phương pháp này có thể được sử dụng để giải quyết các vấn đề như lọc tin tức, gợi ý sản phẩm, chọn bài hát, v.v. Tuy nhiên, phương pháp này cũng có những hạn chế, bao gồm:

- Giới hạn của phương pháp này là chỉ đề xuất các sản phẩm tương đồng với những sản phẩm 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 sản phẩm 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.

## **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 nhân viên.
- Quản trị khách hàng: Theo dõi thông tin khách hàng, xóa khách hàng khỏi danh sách khi cần.
- 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 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

#### **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 gần gũi, 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.

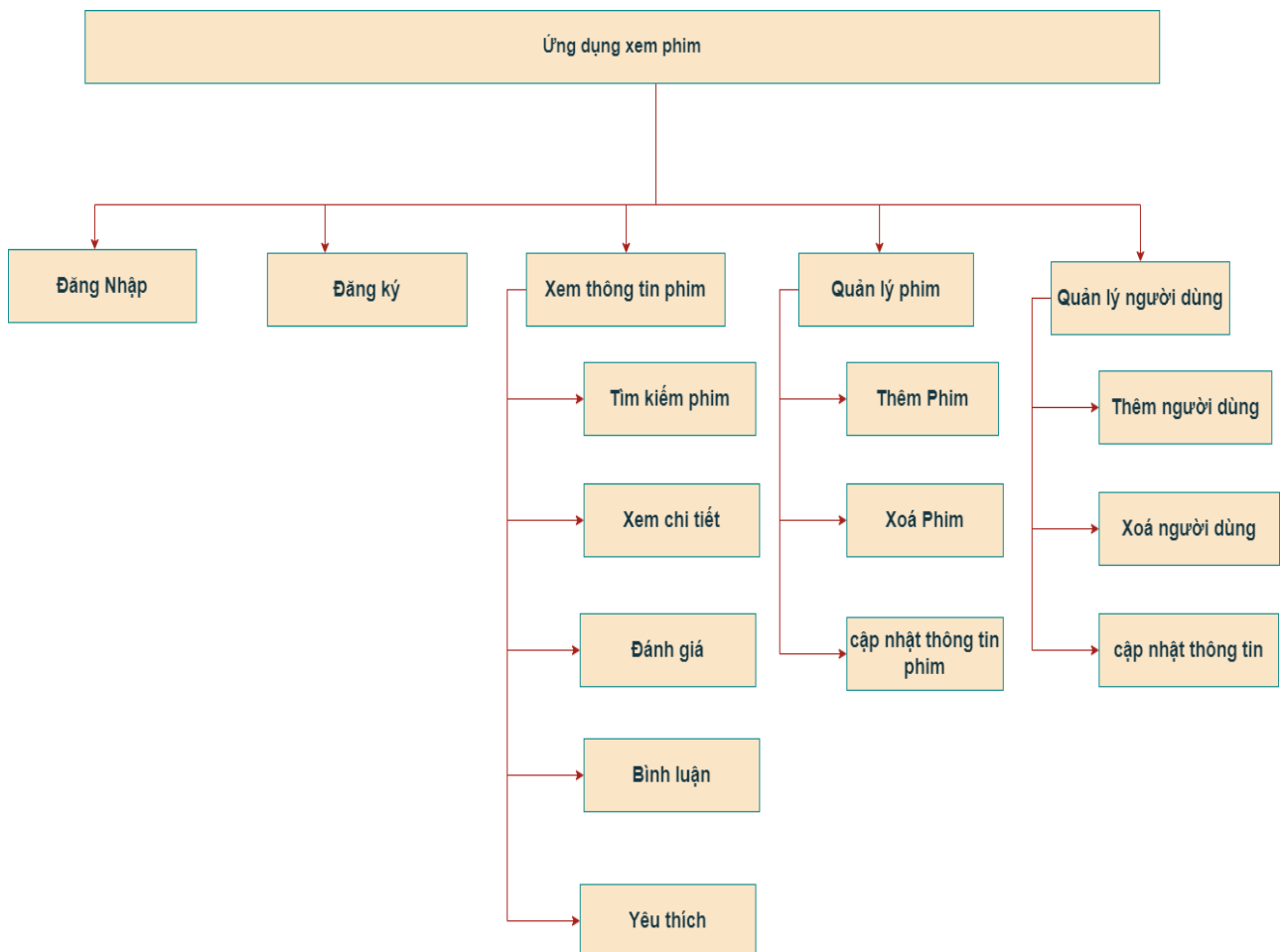
### **3.4 Sơ đồ phân rã chức năng và luồng dữ liệu**

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:

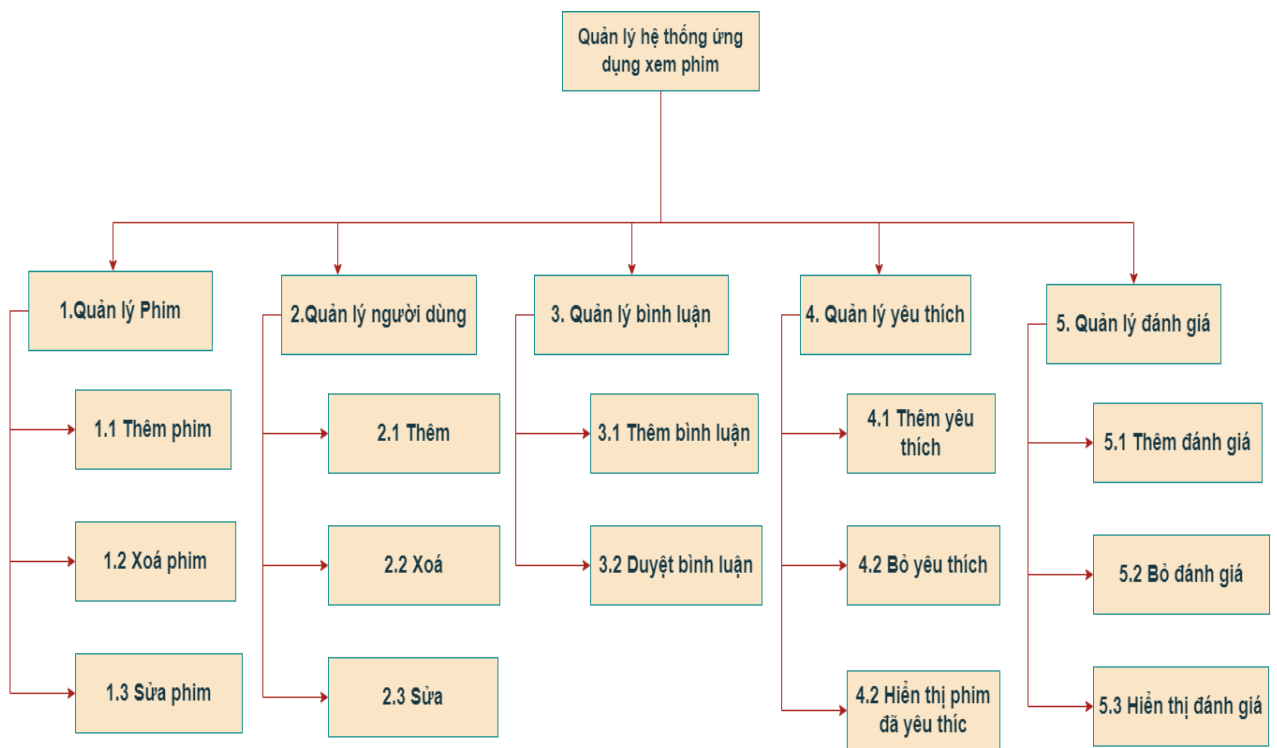
### 3.4.1 Yêu cầu người dùng

- 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

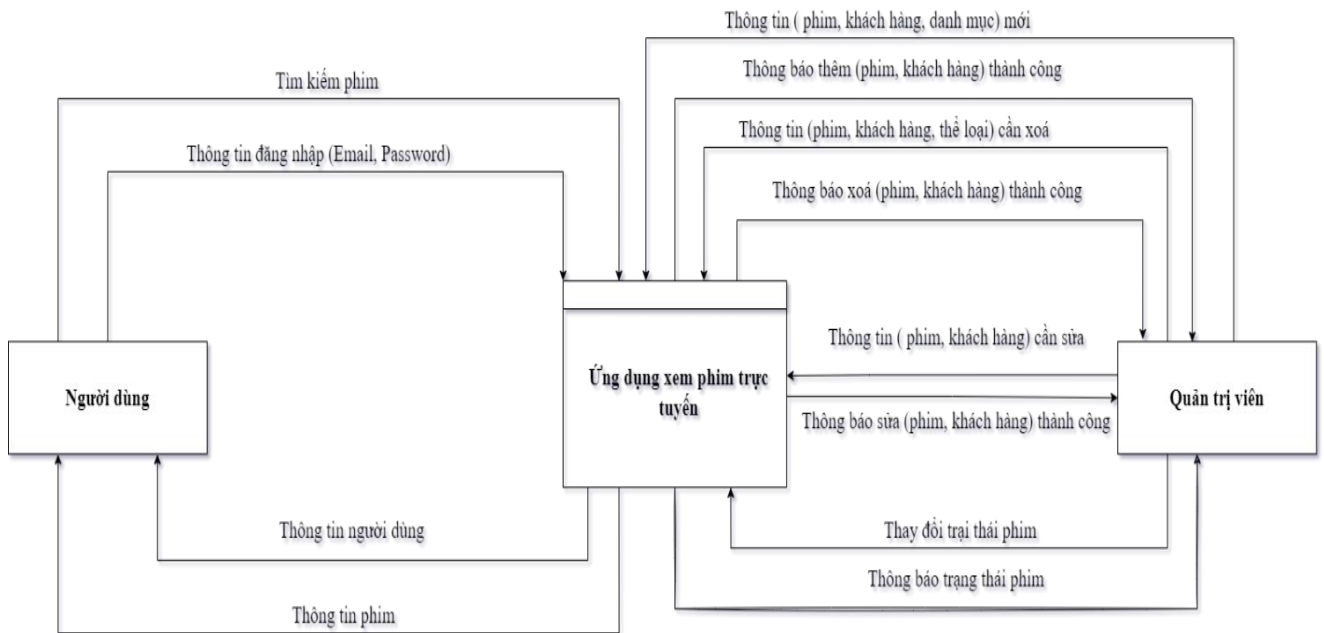


Hình 1 Sơ đồ phân rã chức năng ứng dụng xem phim



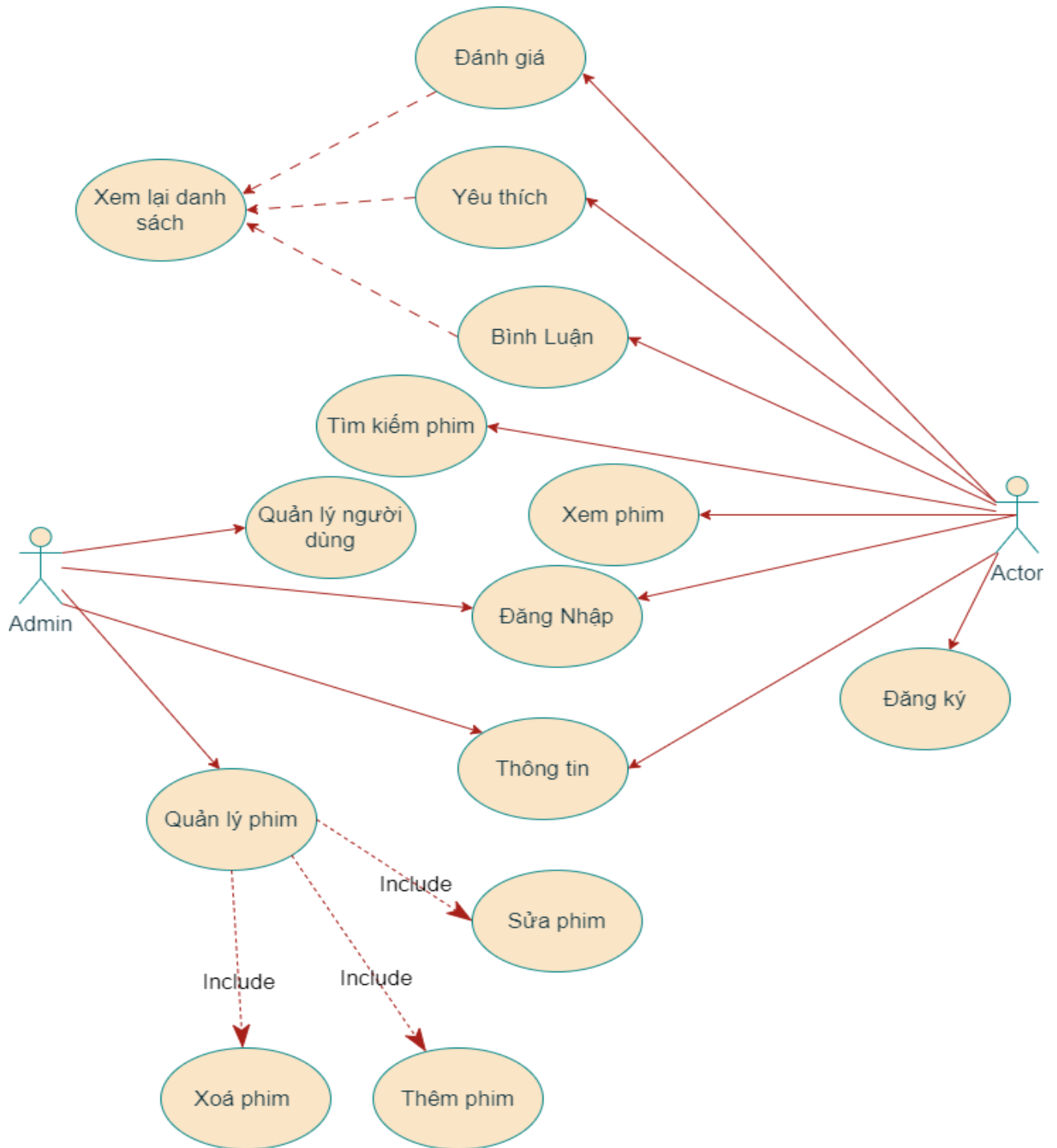
Hình 2 Sơ đồ phân rã chức năng Quản lý ứng dụng

### 3.4.3 Sơ đồ luồng dữ liệu



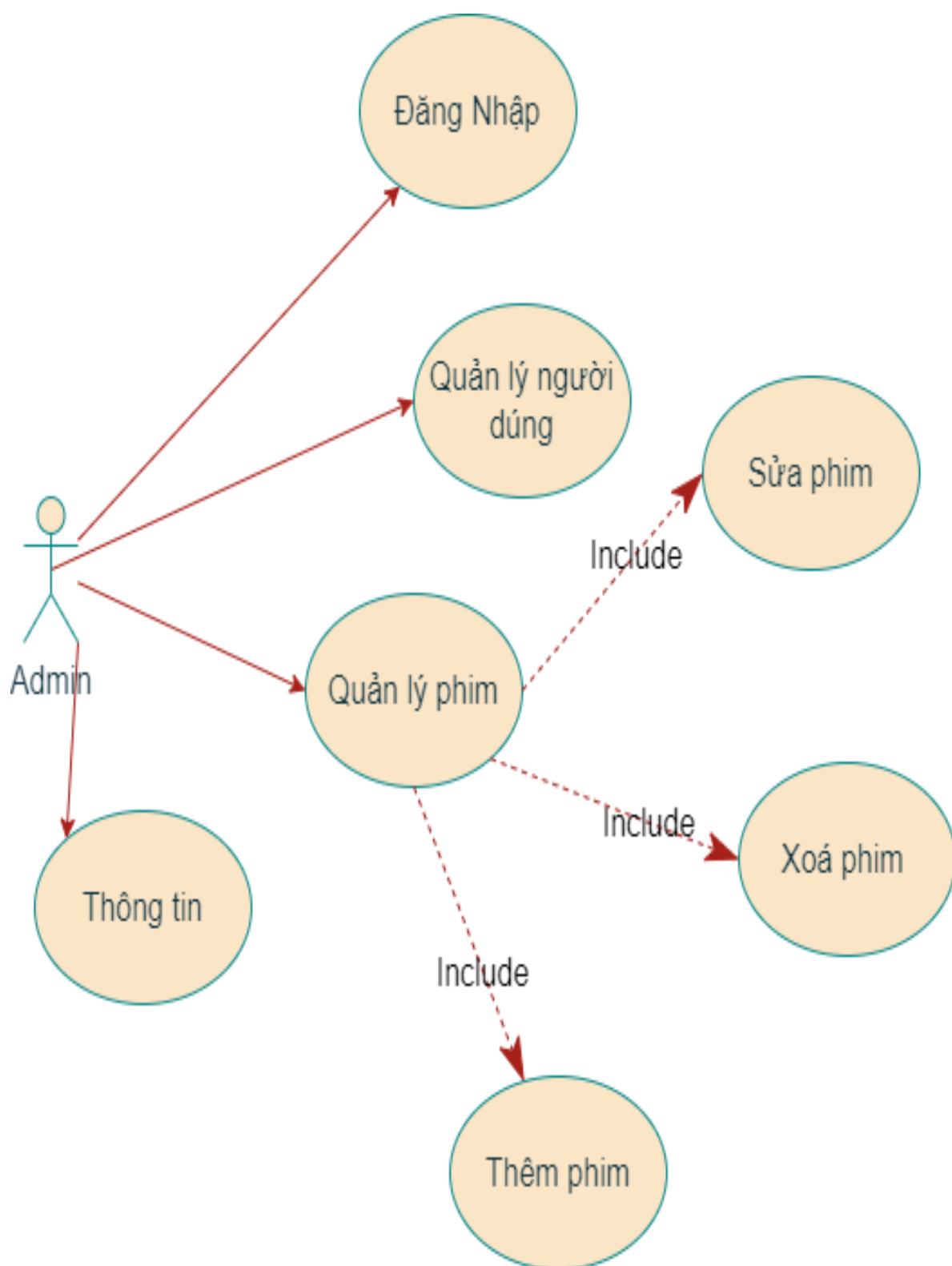
Hình 3 Sơ đồ luồng dữ liệu

### 3.5 Biểu đồ UseCase

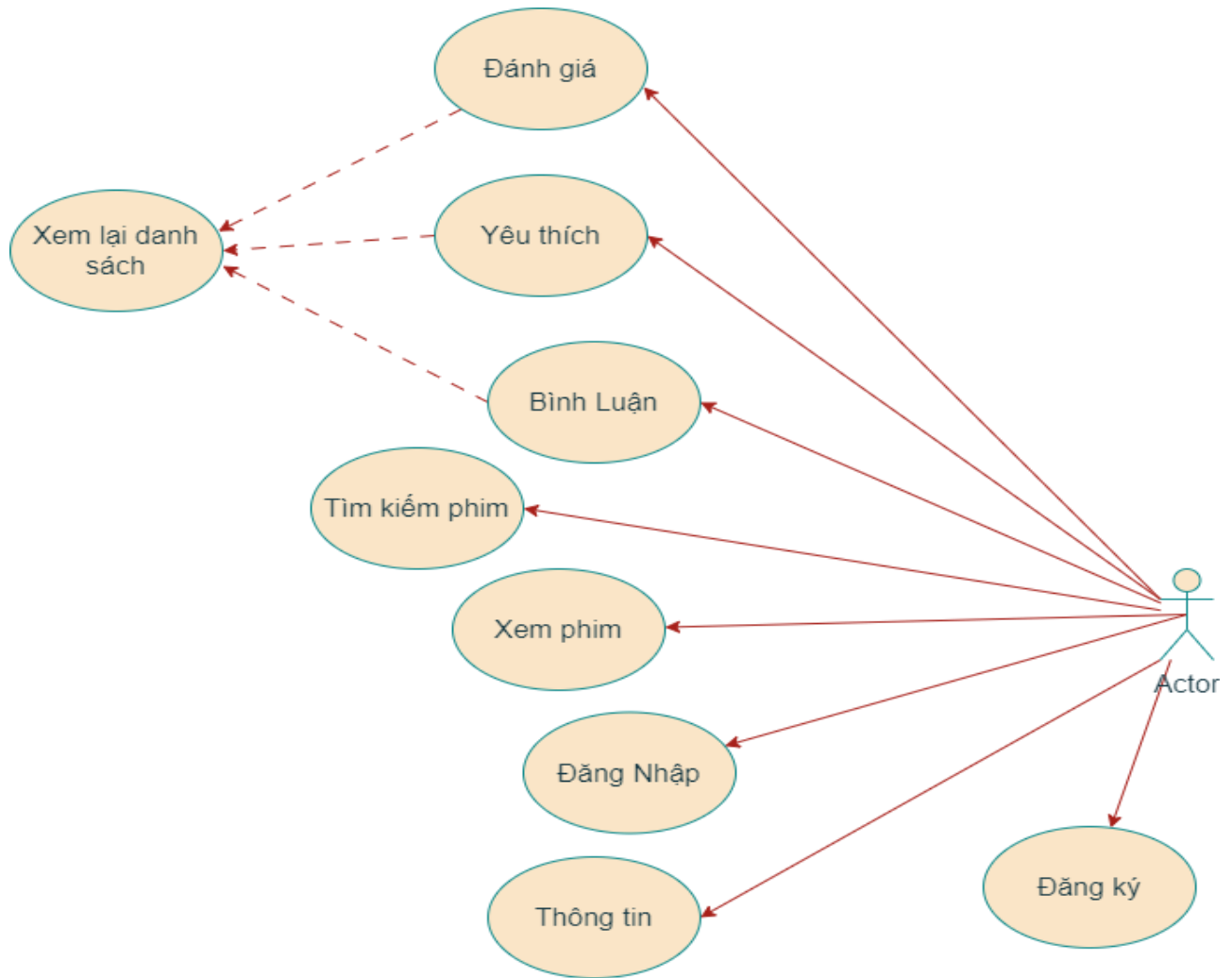


Hình 4 Biểu đồ UseCase tổng quát





*Hình 5 UseCase Admin*



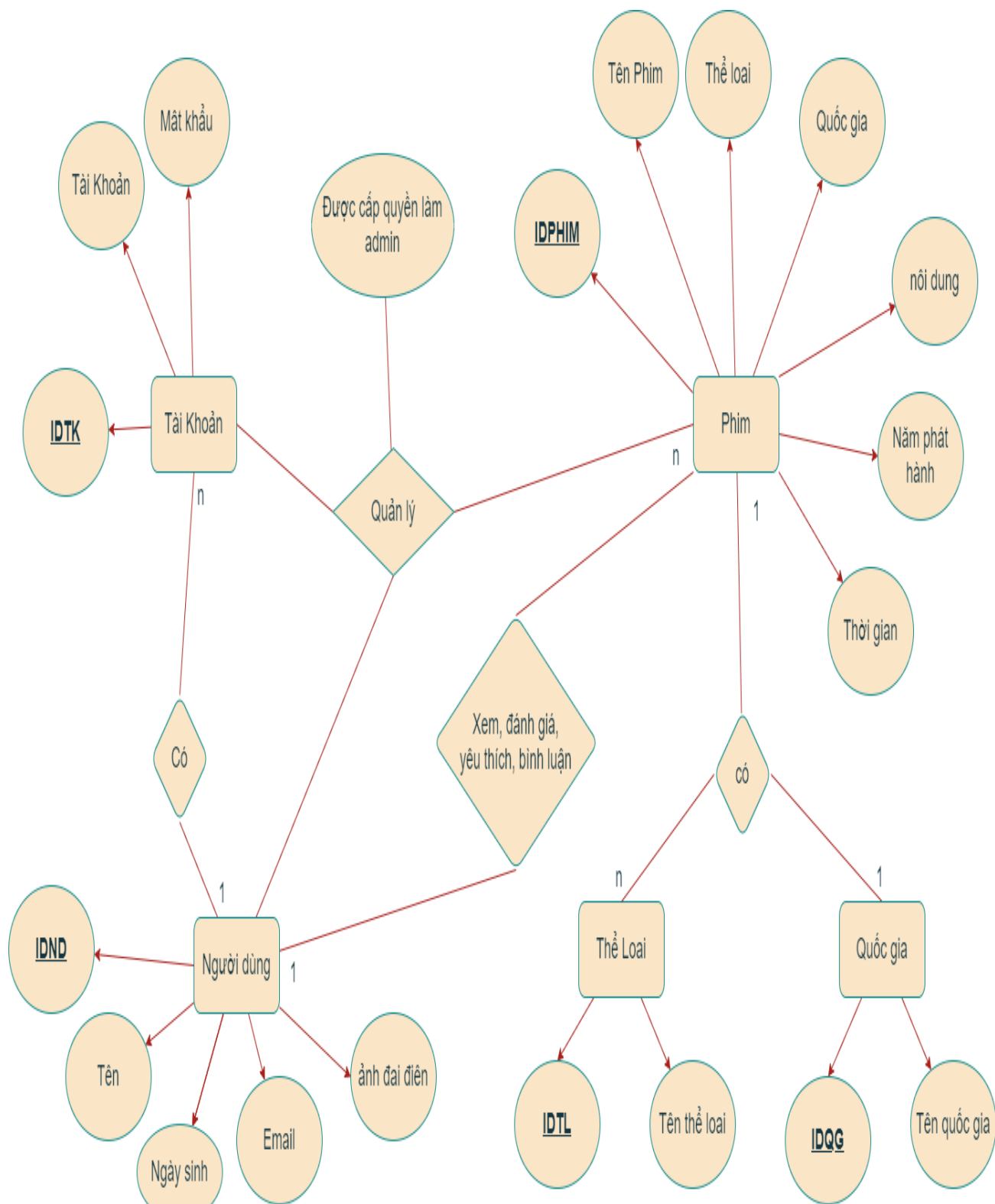
*Hình 6 UseCase Người dùng*

### **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

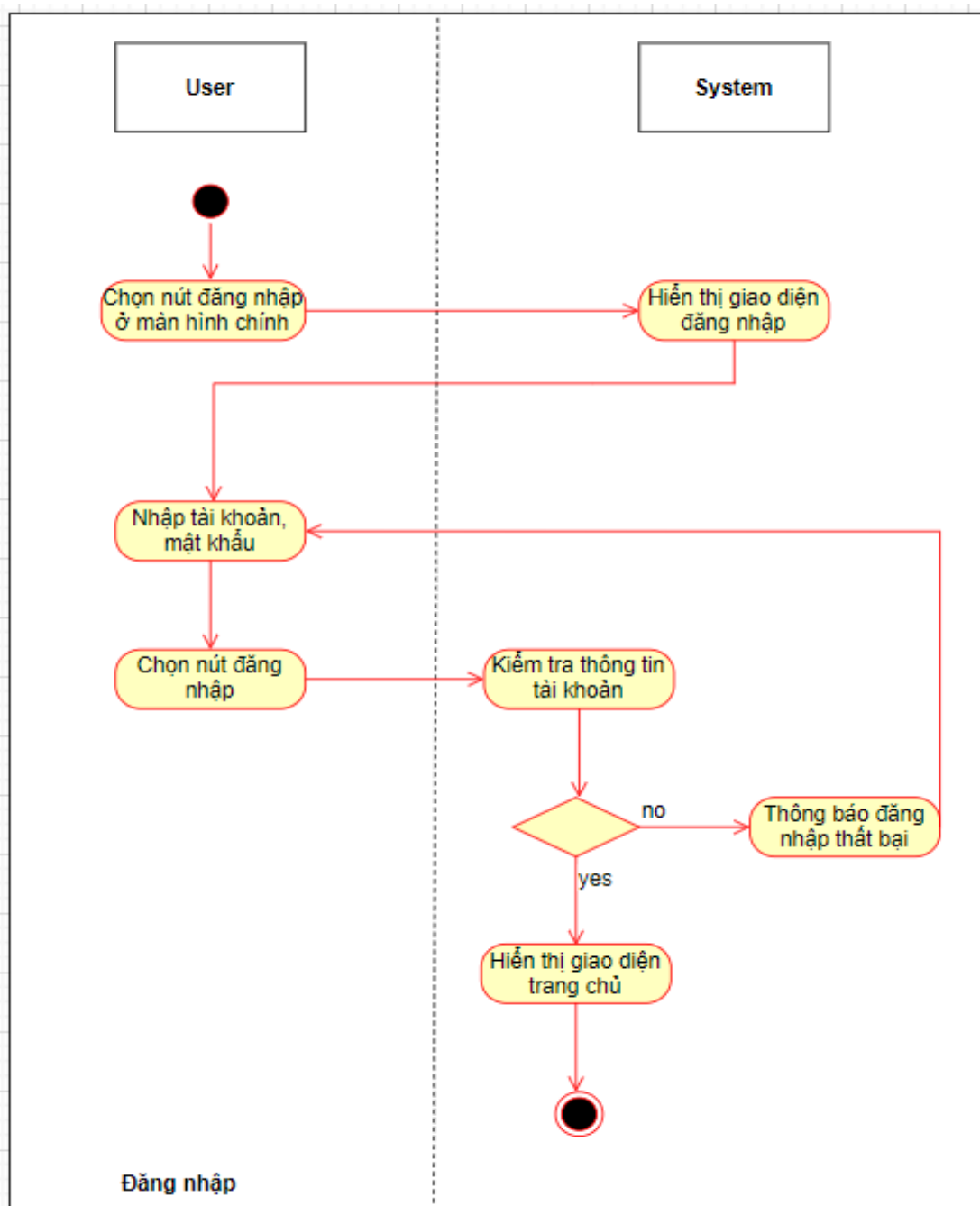


Hình 7 Sơ đồ ER

### 3.7 Sơ đồ hoạt động

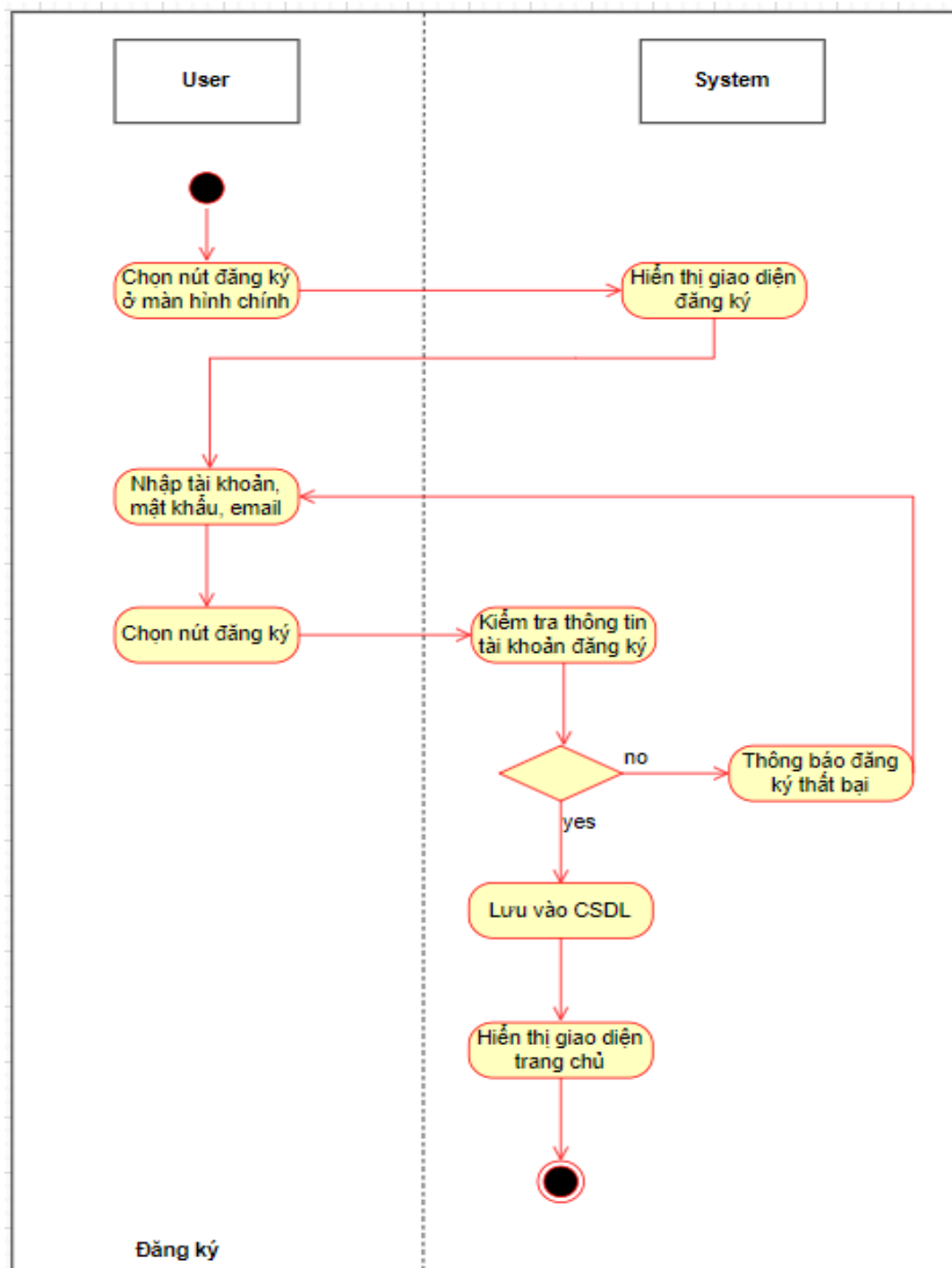
#### 3.7.1 Với người dùng

##### a. Đăng nhập



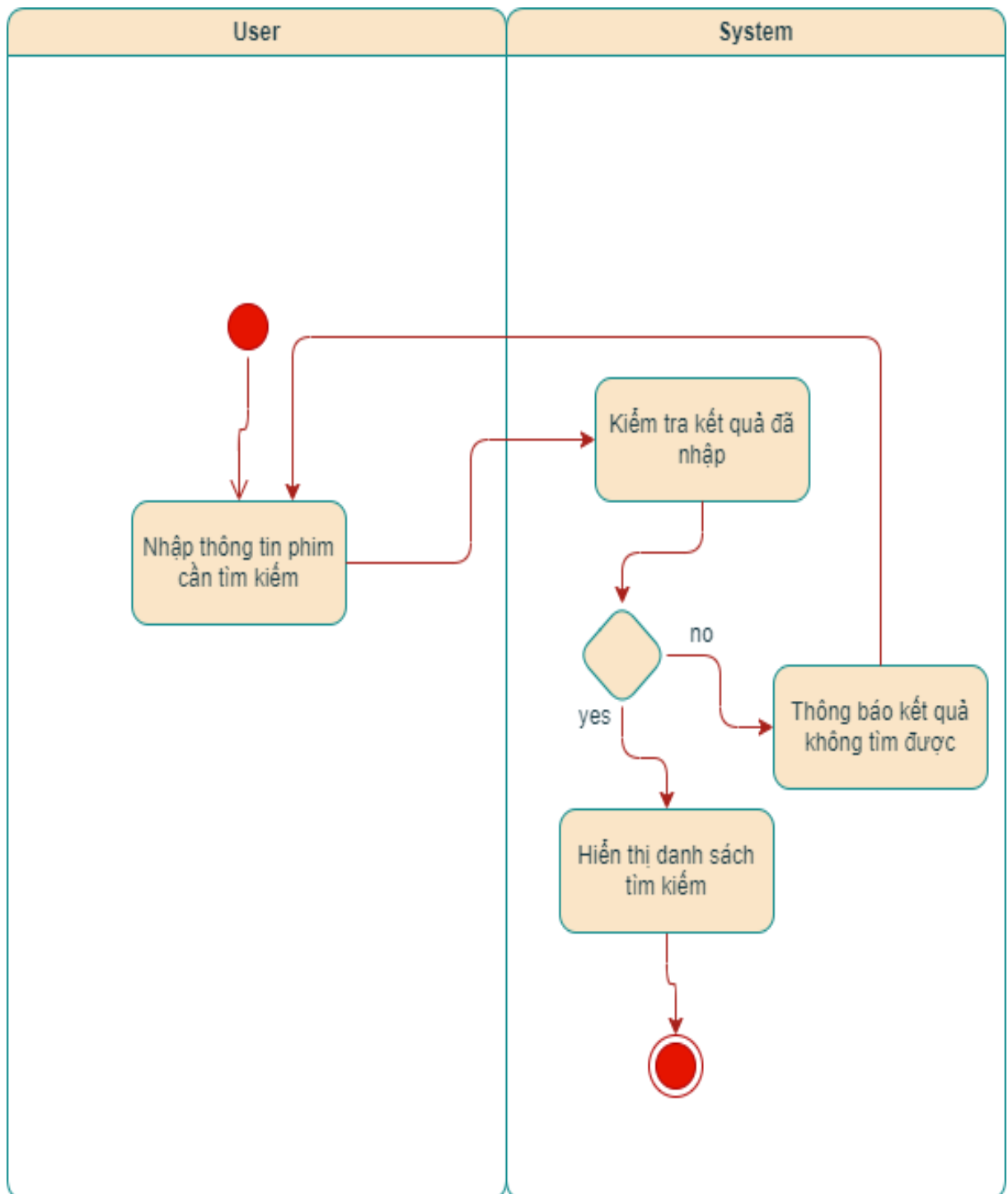
Hình 8 Sơ đồ hoạt động chức năng đăng nhập

b. Đăng ký



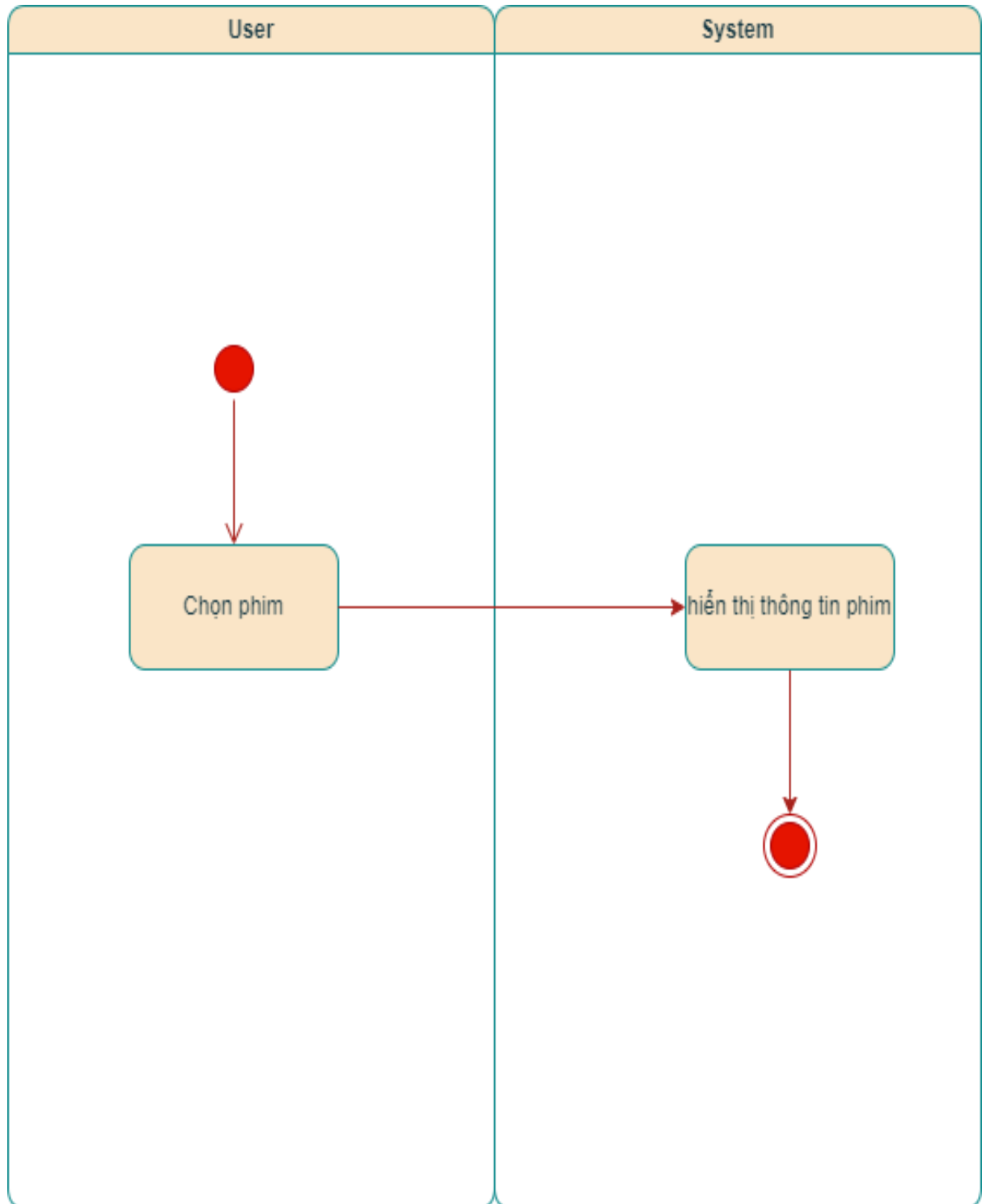
Hình 9 Sơ đồ hoạt động chức năng đăng ký

c. Tìm kiếm phim



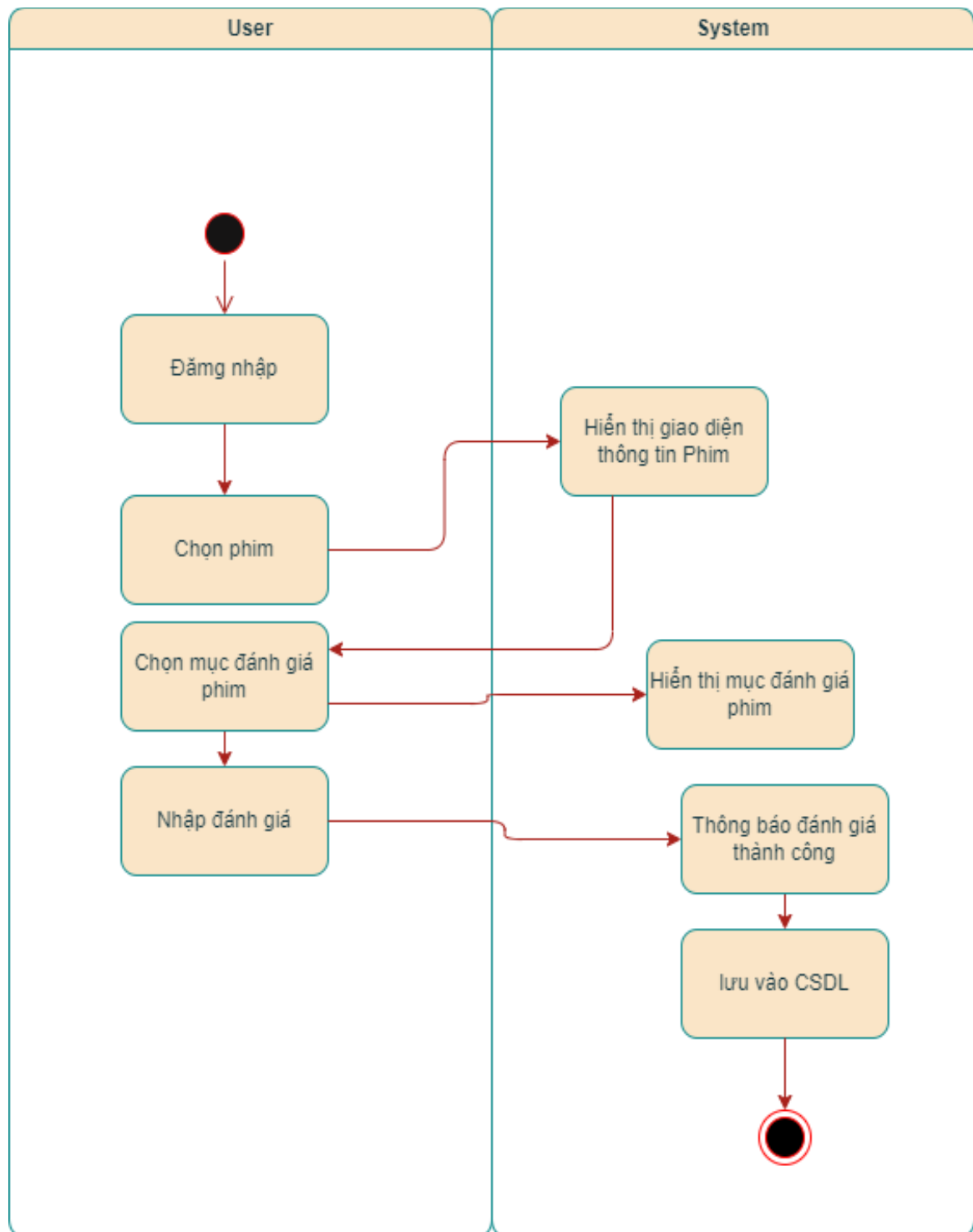
Hình 10 Sơ đồ hoạt động chức năng tìm kiếm phim

*d. Xem thông tin phim*



*Hình 11 Sơ đồ hoạt động chức năng xem chi tiết phim*

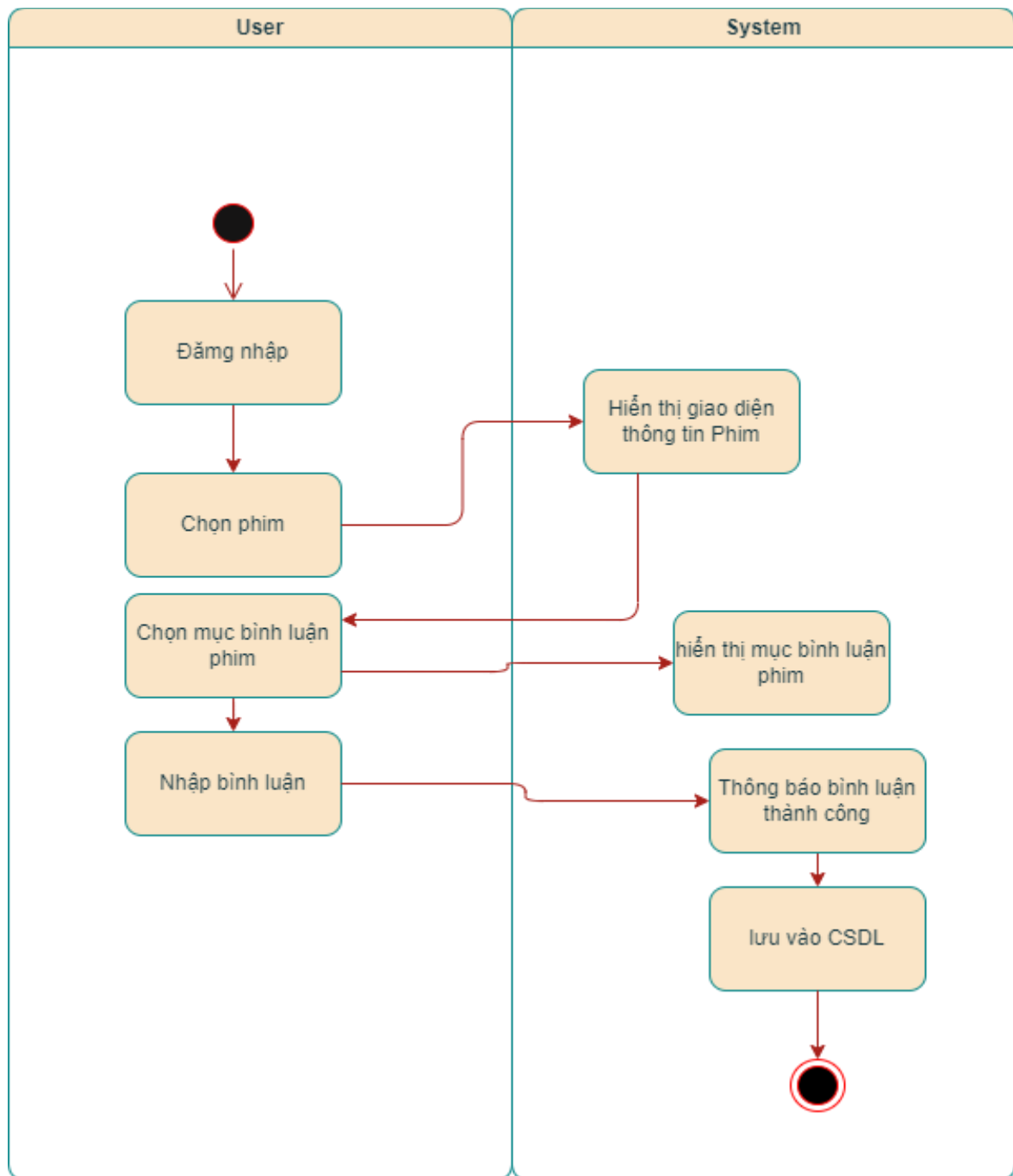
*e. Đánh giá sản phẩm*



*Hình 12 Sơ đồ hoạt động chức năng đánh giá phim*

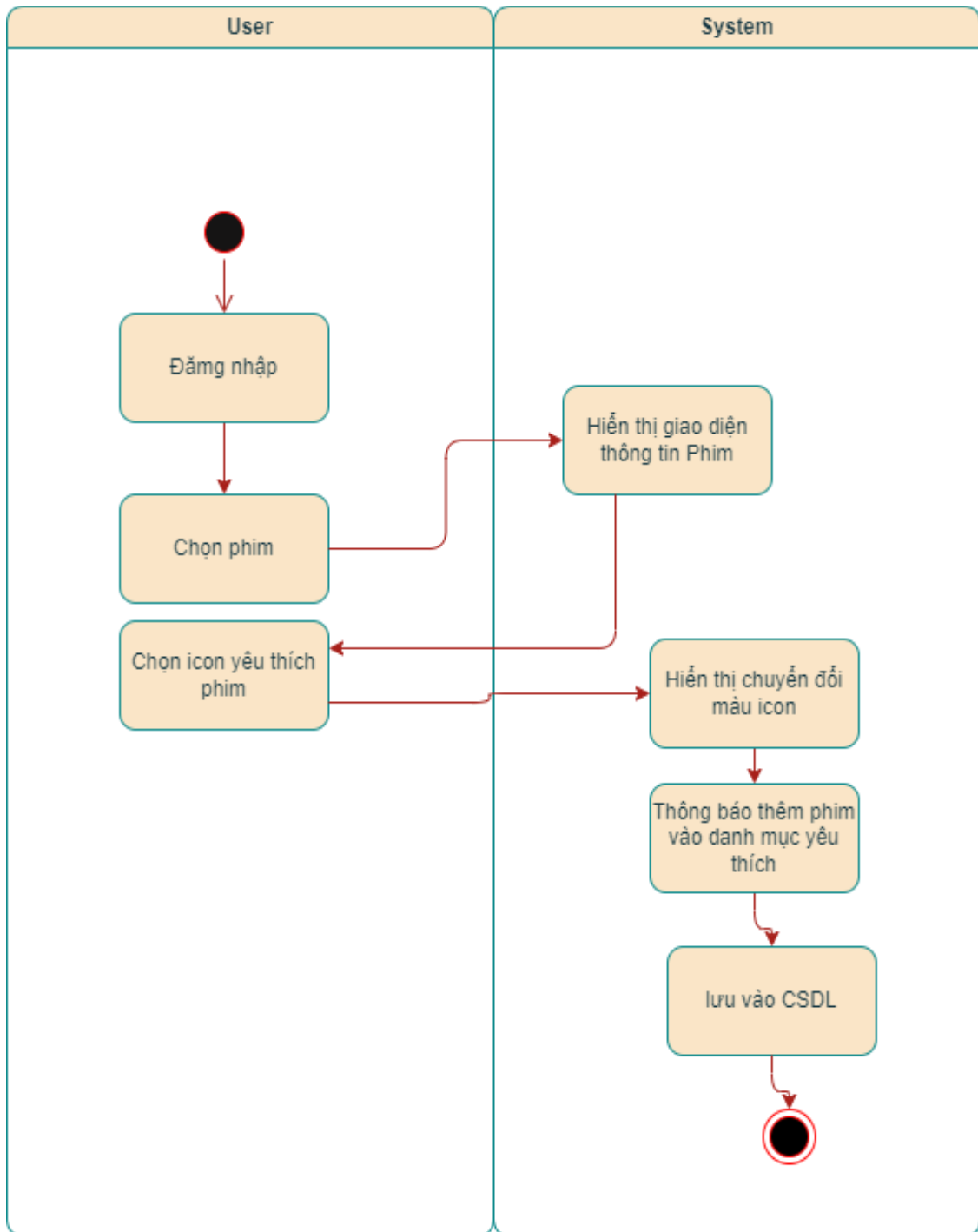


*f. Bình luận phim*



*Hình 13 Sơ đồ hoạt động chức năng bình luận phim*

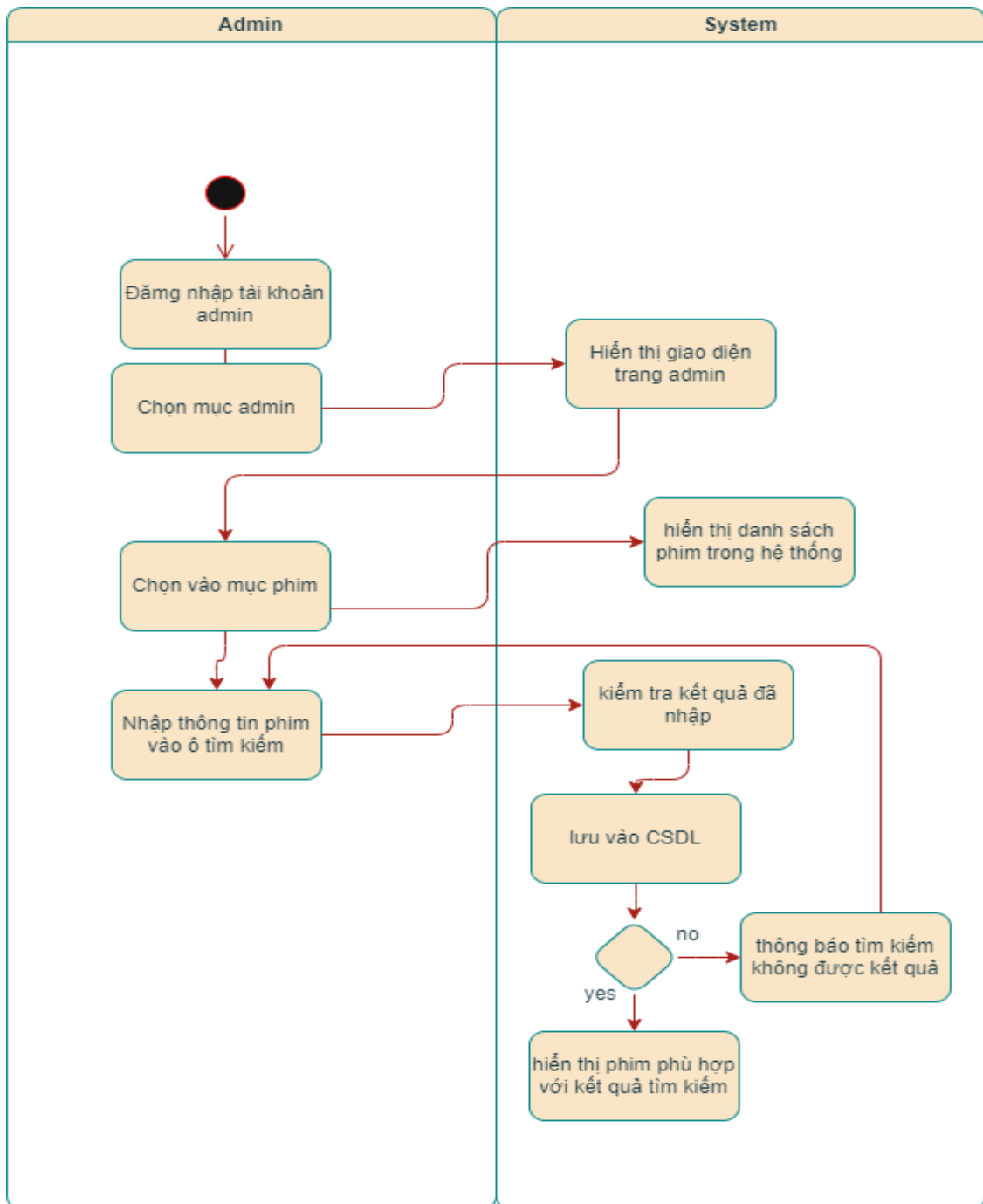
*j. yêu thích phim*



*Hình 14 Sơ đồ hoạt động chức năng yêu thích phim*

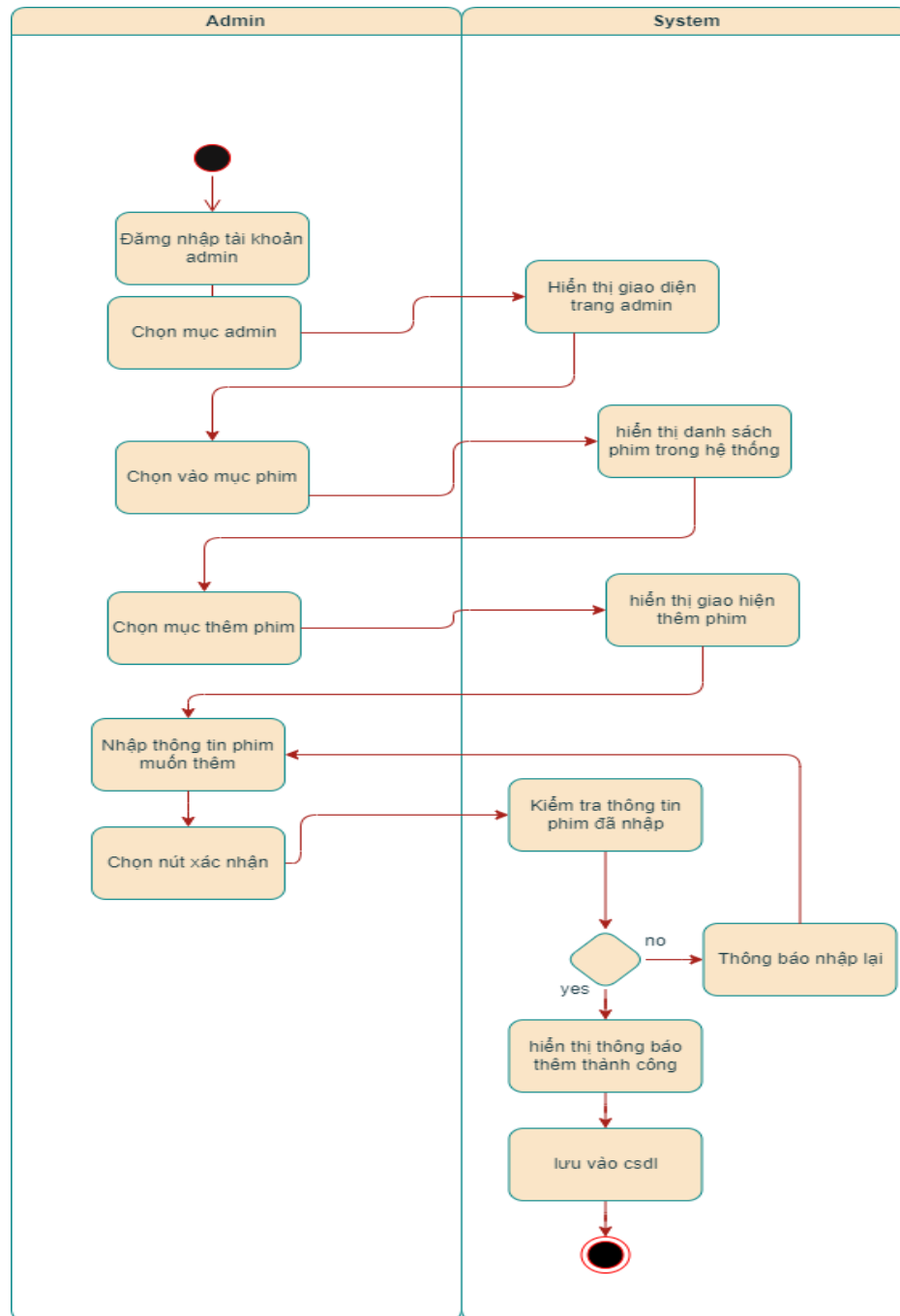
### 3.7.2 Với người quản trị

#### a. Tìm kiếm phim



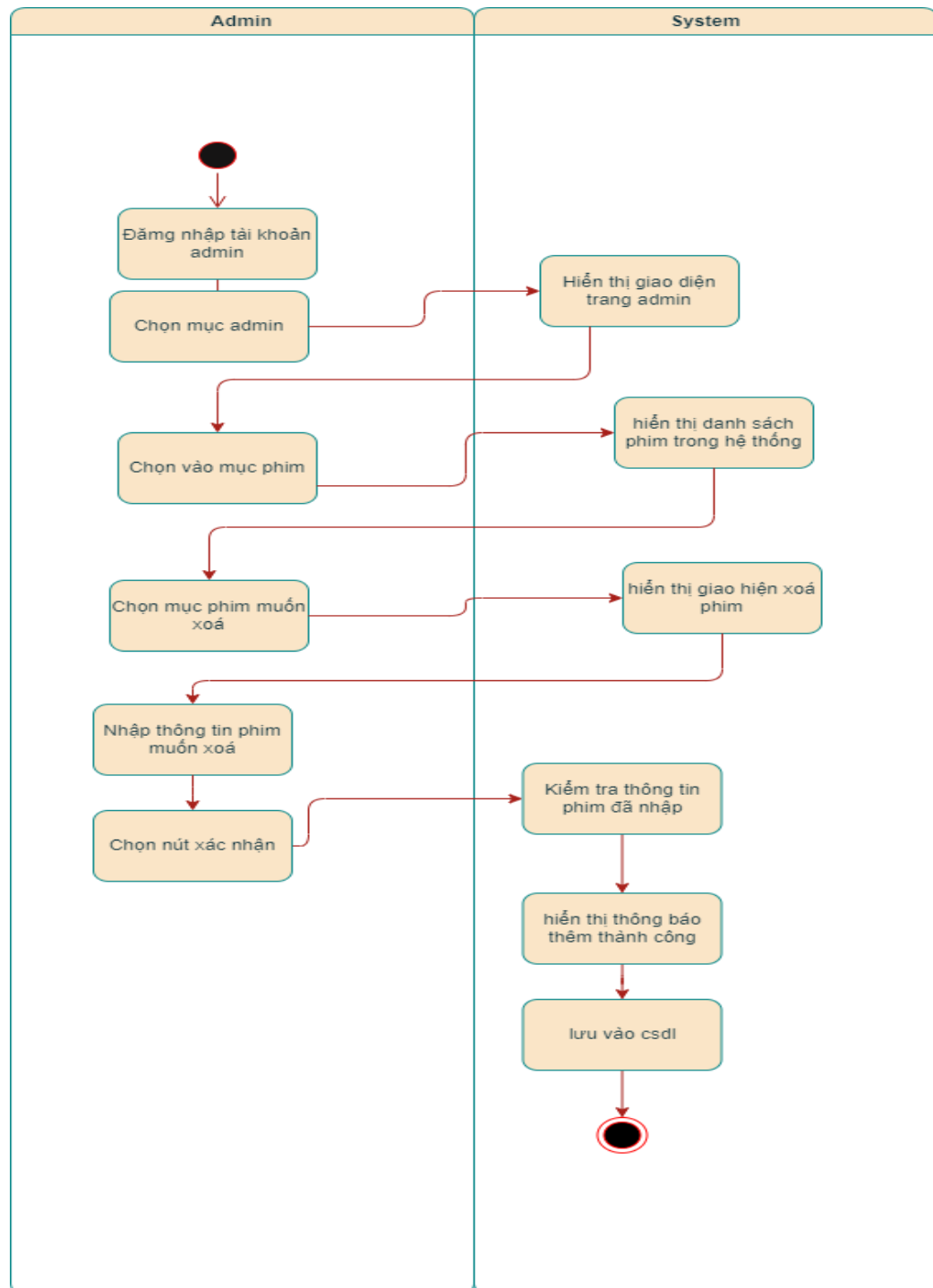
Hình 15 Sơ đồ hoạt động chức năng tìm kiếm phim

*b. Thêm phim*



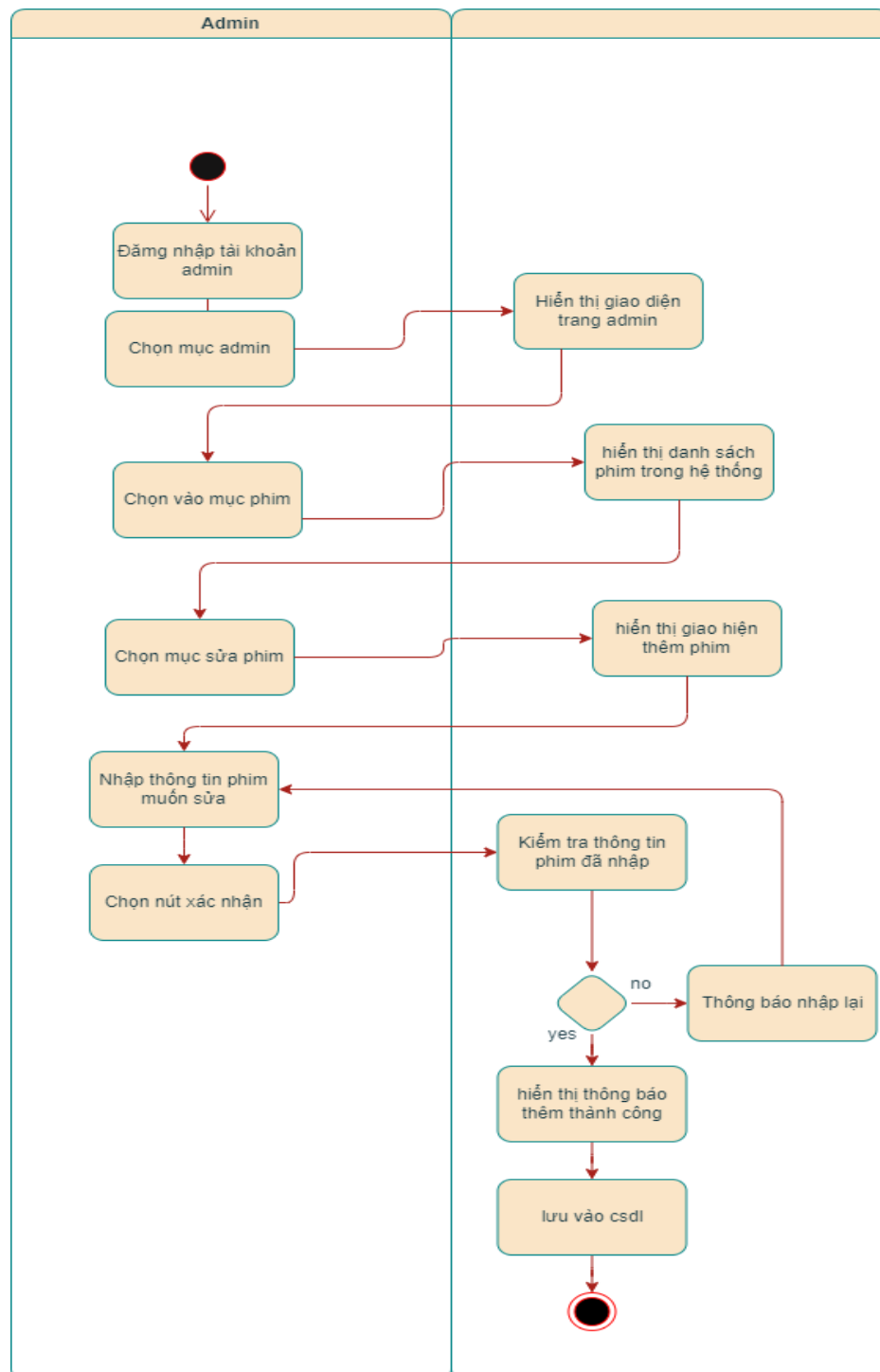
*Hình 16 Sơ đồ hoạt động chức năng thêm phim*

c. Xoá phim



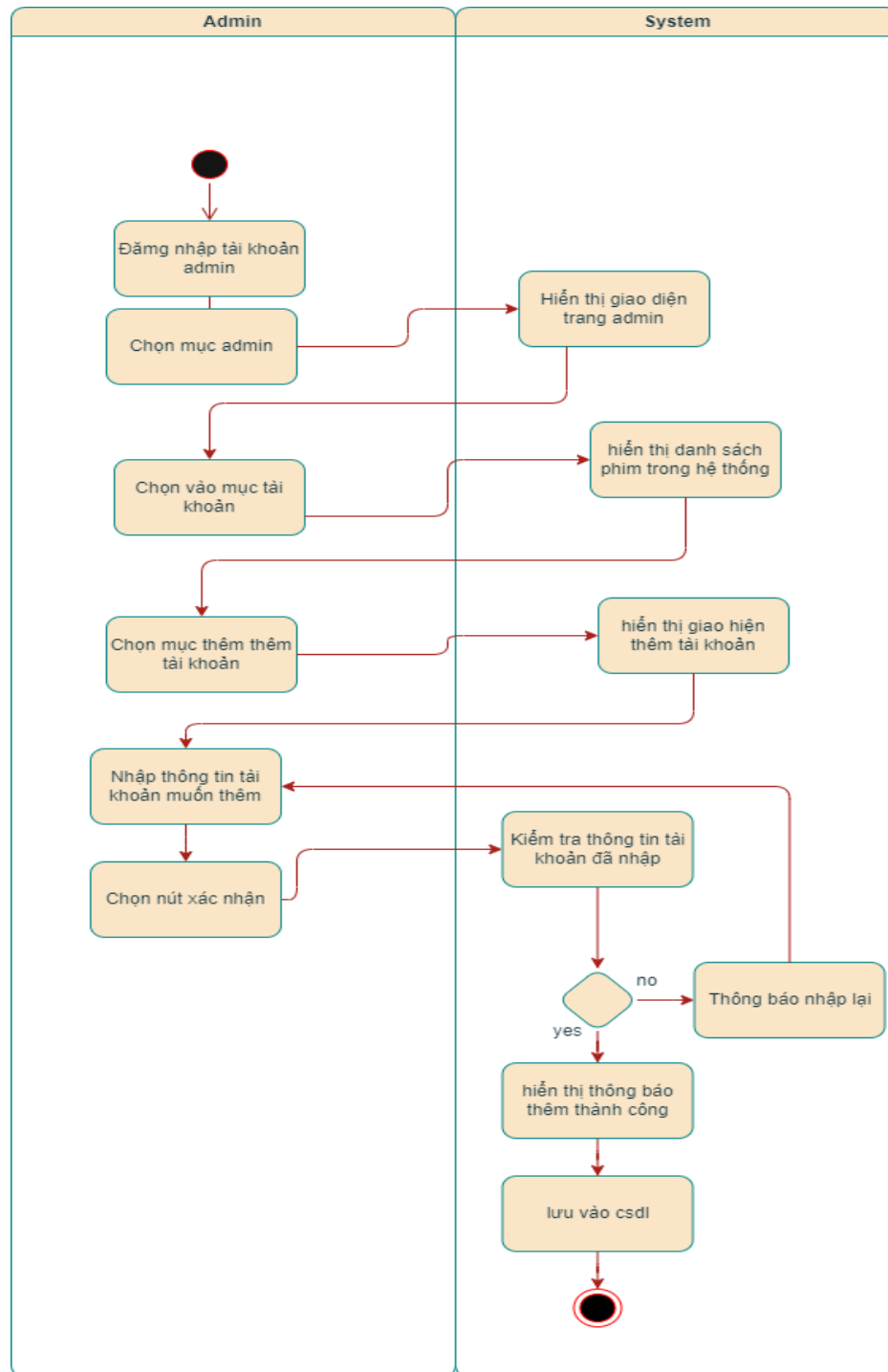
Hình 17 Sơ đồ hoạt động chức năng xoá phim

*d. Sửa phim*



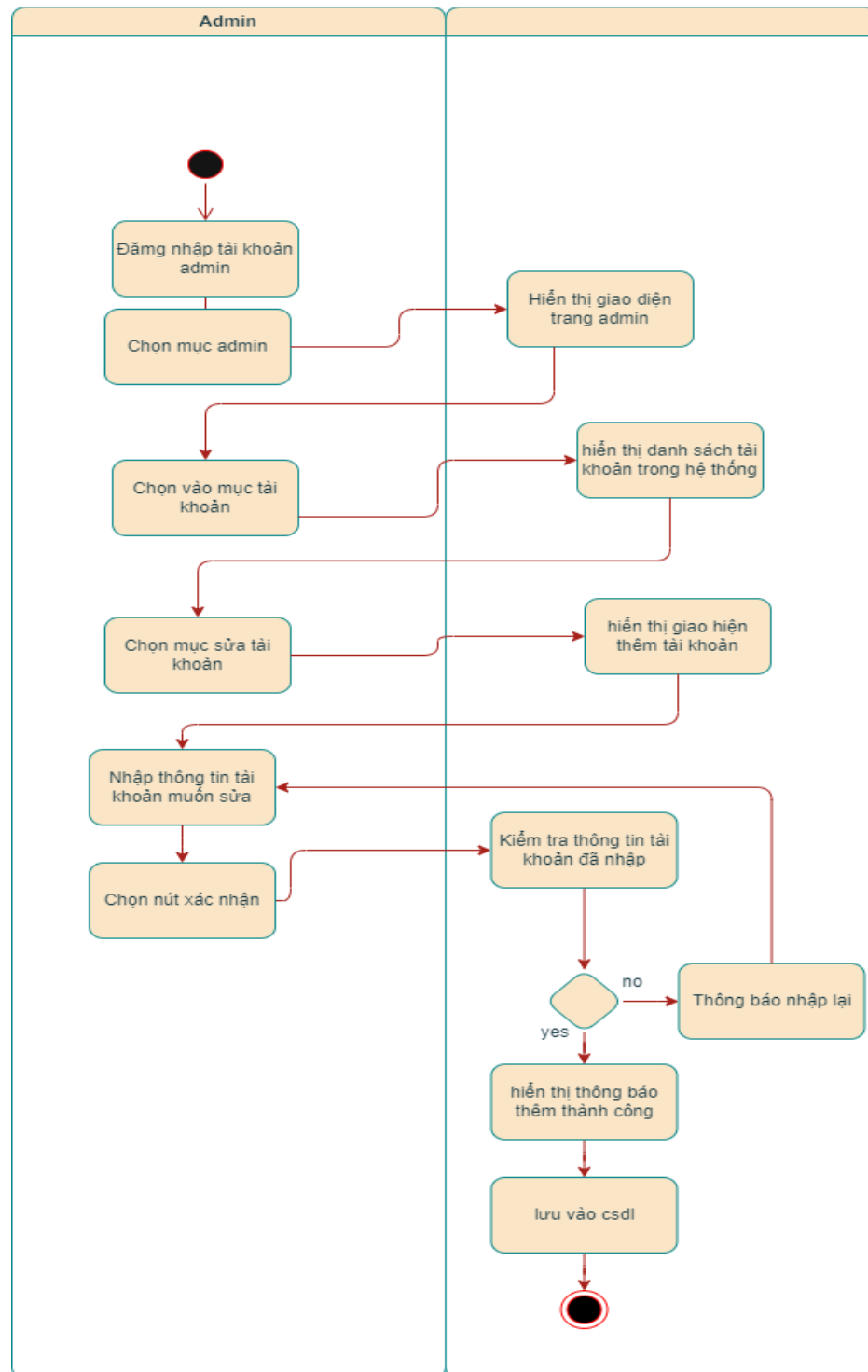
*Hình 18 Sơ đồ hoạt động chức năng sửa phim*

e. Thêm tài khoản



Hình 19 : Sơ đồ hoạt động chức năng thêm tài khoản

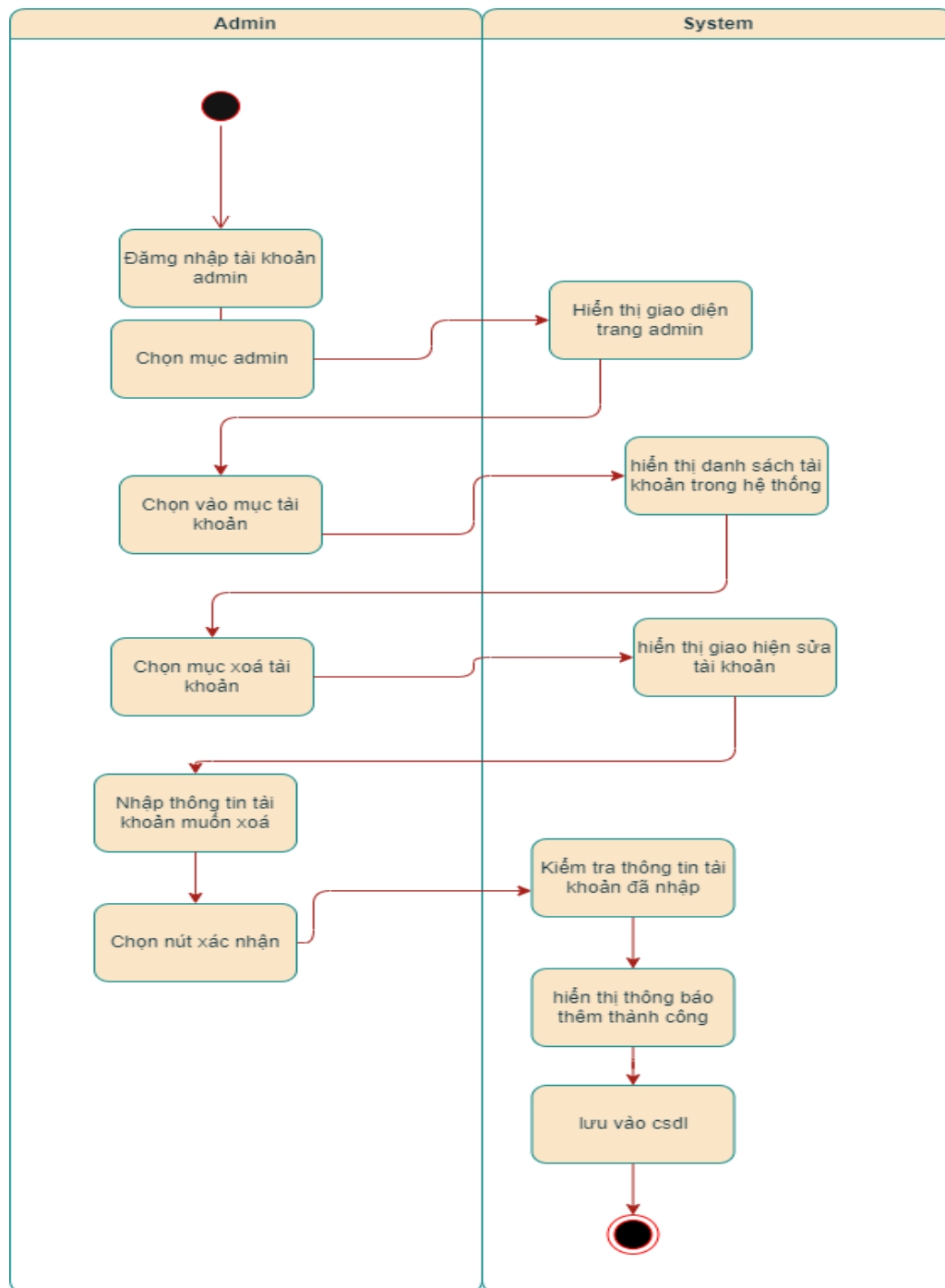
*f. Sửa tài khoản*



*Hình 20 Sơ đồ hoạt động chức năng sửa tài khoản*

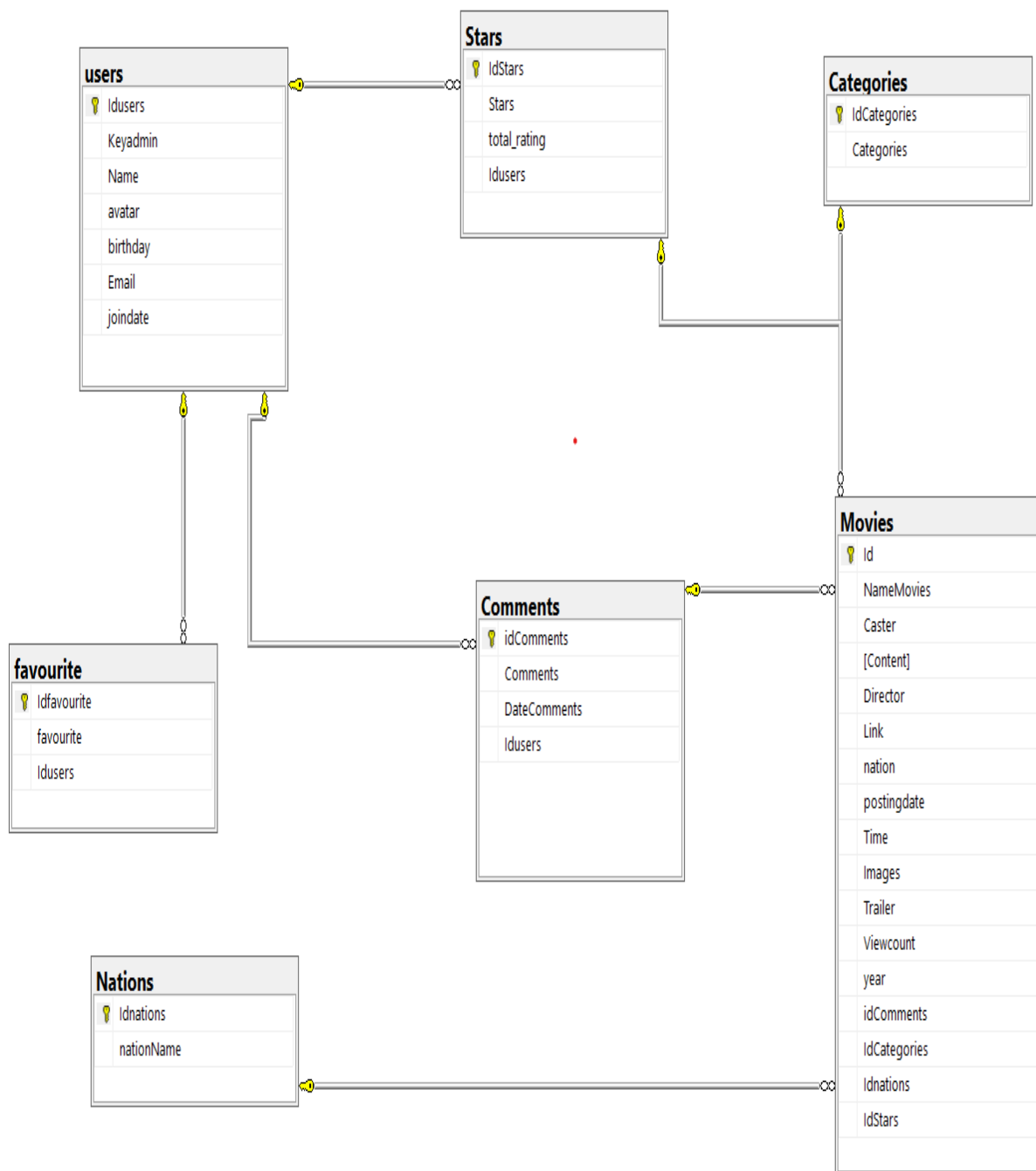


g. Xoá tài khoản



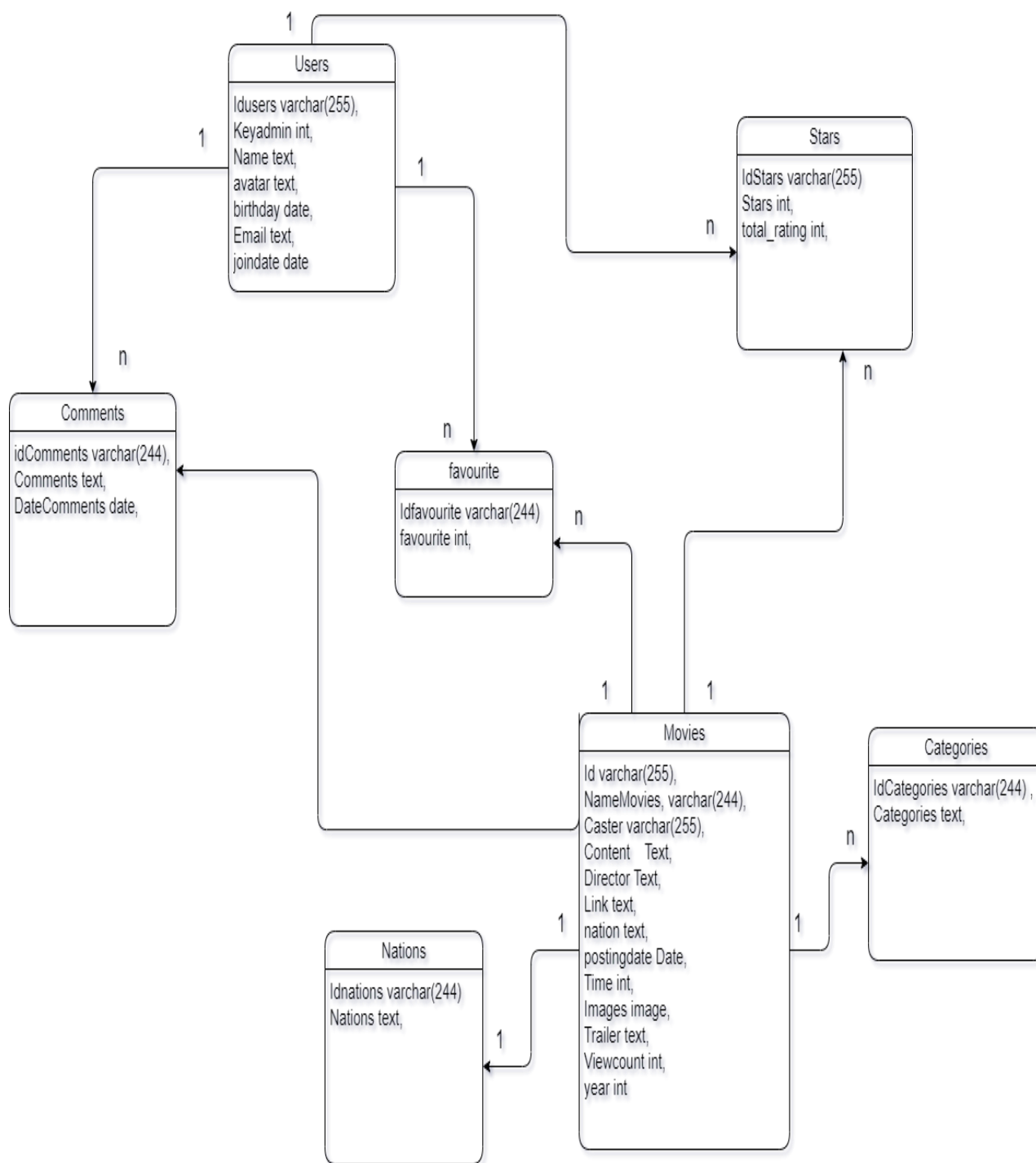
Hình 21 Sơ đồ hoạt động chức năng xoá tài khoản

### 3.8 Sơ đồ Database Diagram



Hình 22 Sơ đồ Diagram

### 3.9 Sơ đồ ClassDiagram



Hình 23 Sơ đồ ClassDiagram

## CHƯƠNG 4 TRIỂN KHAI CHƯƠNG TRÌNH ỨNG DỤNG XEM PHIM TRỰC TUYẾN

### 4.1 Giao diện chức năng đăng nhập:

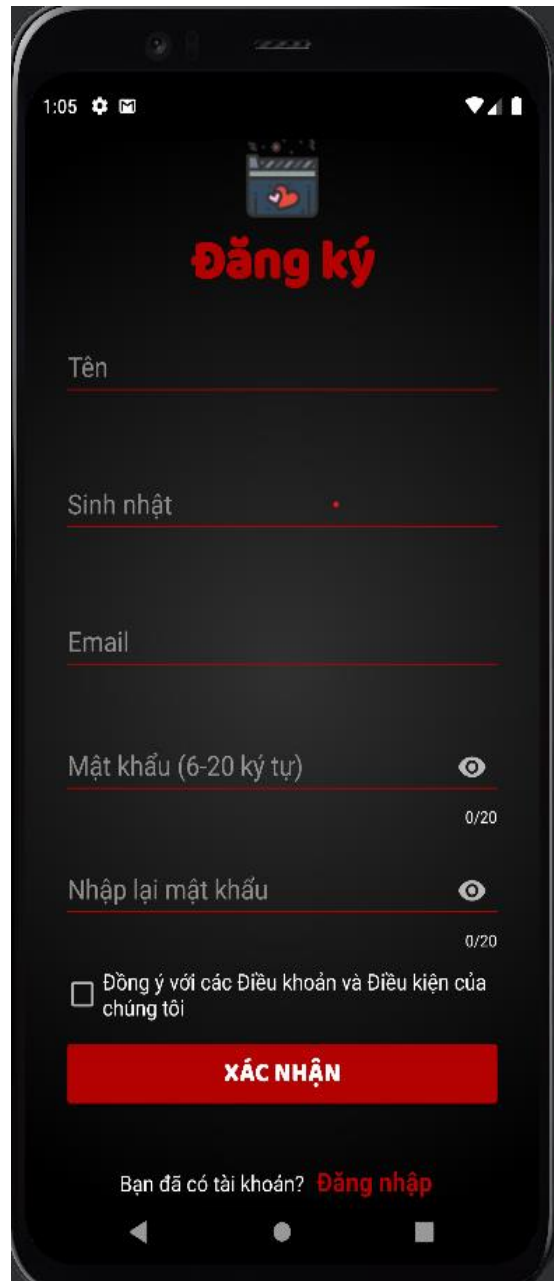
Khi người dùng chạy ứng dụng thì bước đầu tiên để đăng nhập vào ứng dụng thì chúng ta cần phải đăng nhập tài khoản và mật khẩu (nếu đã có tài khoản trước), Nếu chưa có tài khoản thì chúng ta cần phải đăng ký tài khoản để đăng nhập vào ứng dụng.



Hình 24 Giao diện chức năng đăng nhập

## 4.2 Giao diện chức năng đăng ký

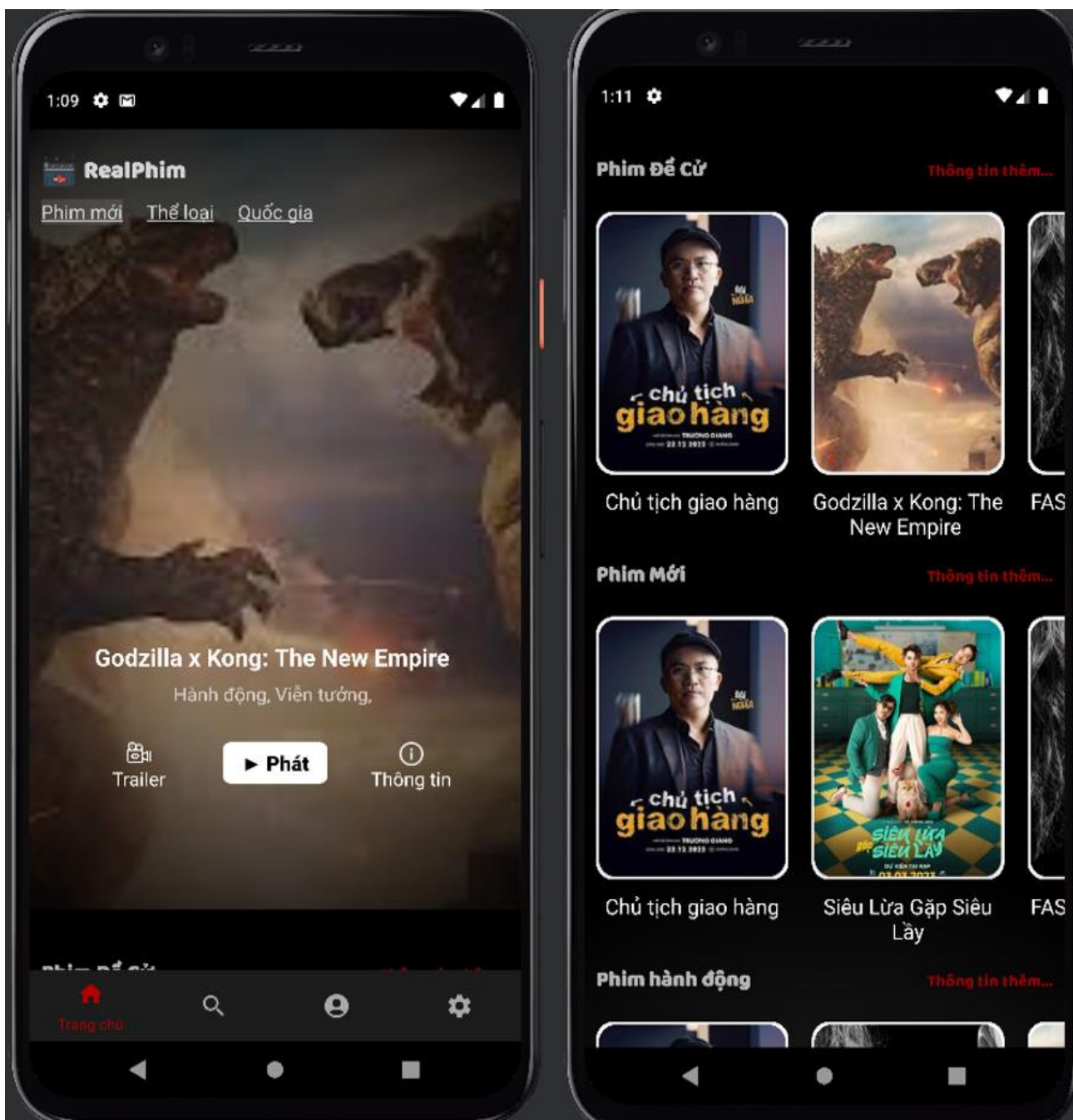
Với chức năng đăng ký thì người dùng sẽ điền vào những thông tin cần thiết mà ứng dụng đã yêu cầu. Sau khi nhập xong thông tin thì chúng ta tích vào điều khoản và bắt đầu đăng ký tài khoản riêng cho mình. Sau đó ta sẽ tài khoản của mình và bắt đầu đăng nhập vào ứng dụng



Hình 25 Giao diện chức năng đăng ký

### 4.3 Giao diện chức năng trạng chủ

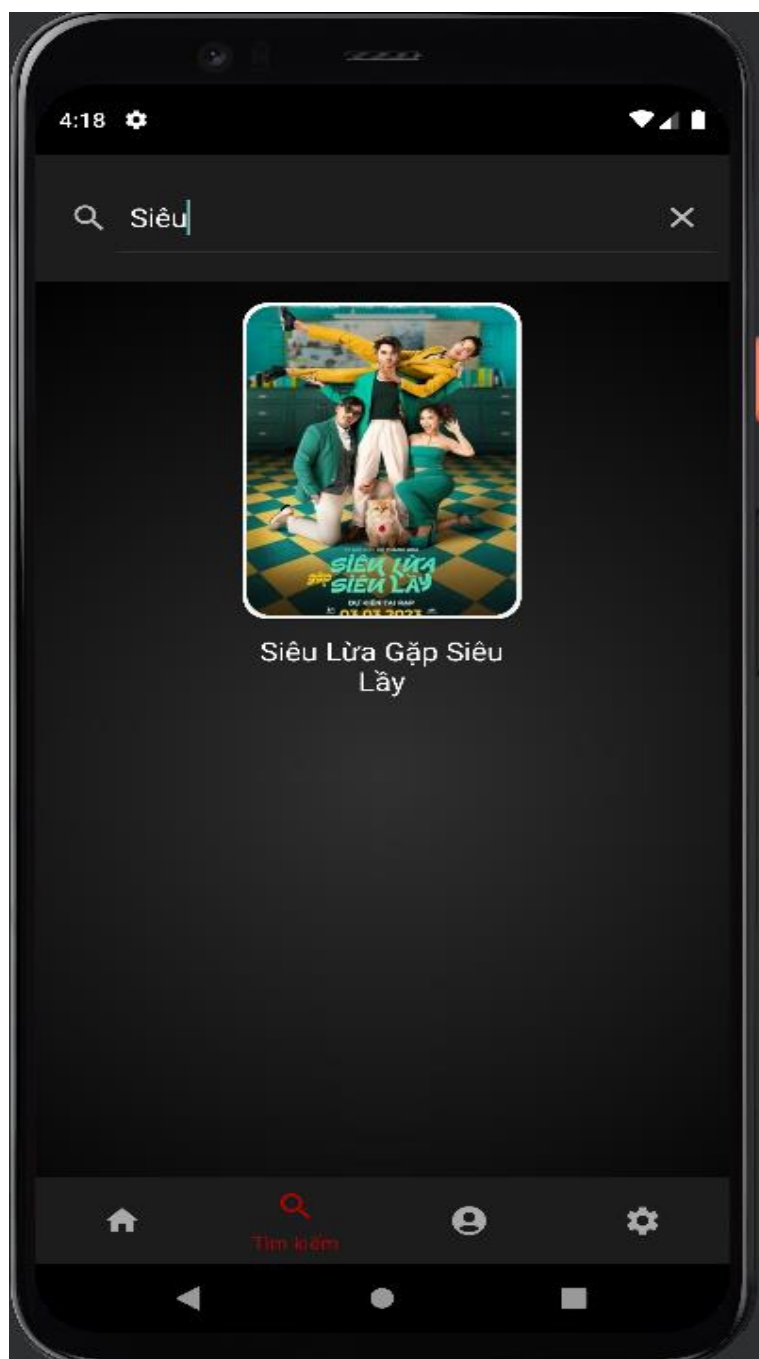
Giao diện trang chủ là trang đầu tiên hiển thị cho khách hàng truy cập vào ứng dụng. Tại đây người dùng có thể xem thông tin về các bộ phim, hiển thị danh sách phim mới nhất, phim đề cử và thể loại phim.



Hình 26 Giao diện trạng chủ

#### 4.4 Giao diện tìm kiếm

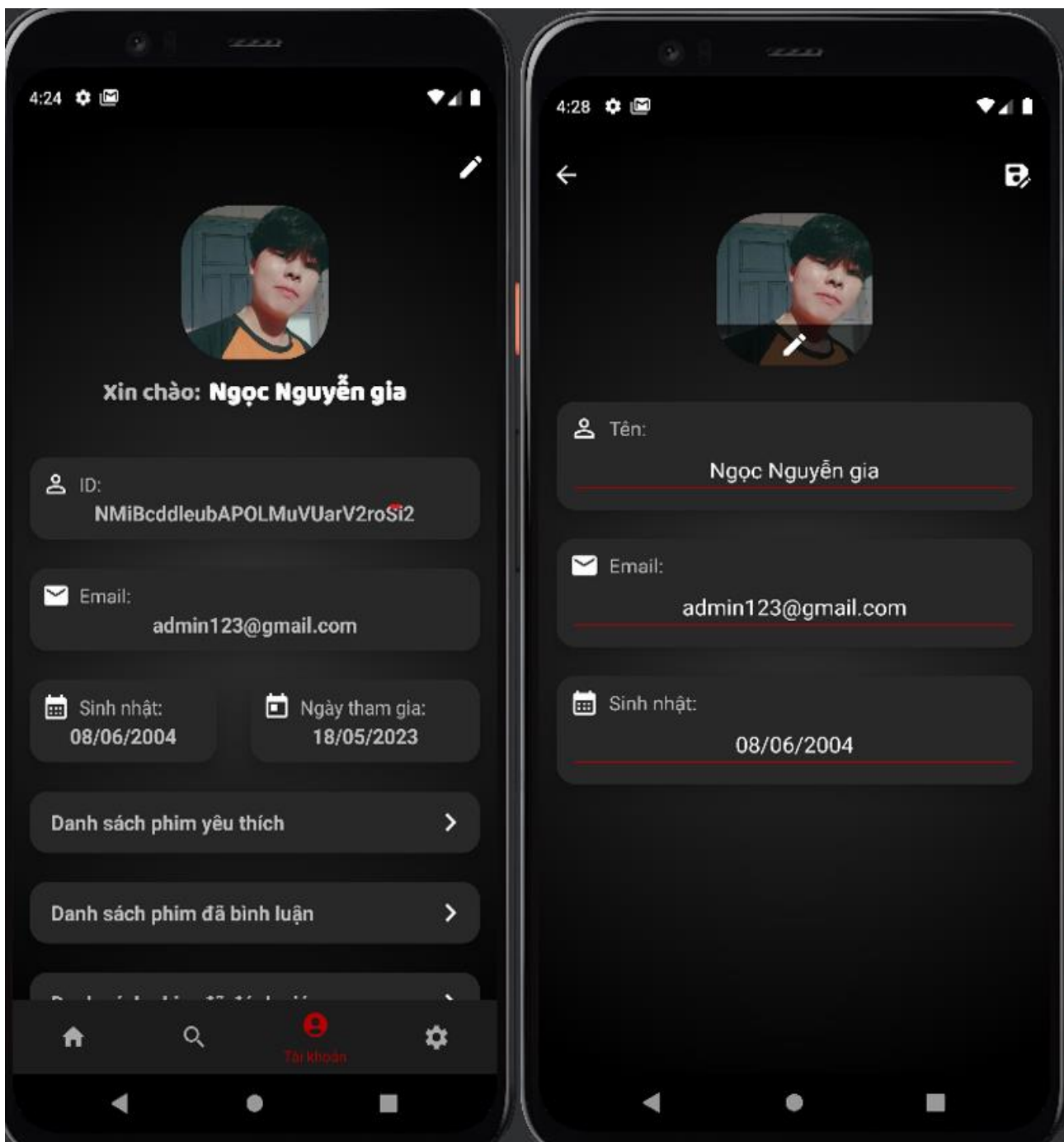
Khi người dùng truy cập vào mục tìm kiếm và bắt đầu tìm kiếm thông tin phim thì hệ thống sẽ hiển thị ra những phim mà người dùng muốn xem



Hình 27 Giao diện tìm kiếm

#### 4.5 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 28 Giao diện thông tin và chỉnh sửa thông tin người dùng



## 4.6 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 showing the 'Add Movie' interface. The left screen shows the top half with fields for movie link, posters, and genre selection. The right screen shows the bottom half with fields for movie title, cast, director, runtime, release year, country, synopsis, and trailer link, followed by a red 'Thêm' (Add) button.

**Left Screen (4:31):**

- Link phim:  
Link
- Poster 3x4: [Camera icon]
- Poster 4x3: [Camera icon]
- Thể loại:
  - ☐ Hành động ☐ Võ thuật
  - ☐ Tình cảm ☐ Hoạt hình
  - ☐ Hải hước ☐ Viễn tưởng
  - ☐ Cổ trang ☐ Phiêu lưu
  - ☐ Tâm lý ☐ Kinh dị
  - ☐ Hình sự ☐ Thần thoại
  - ☐ Chiến tranh
- Tên phim:

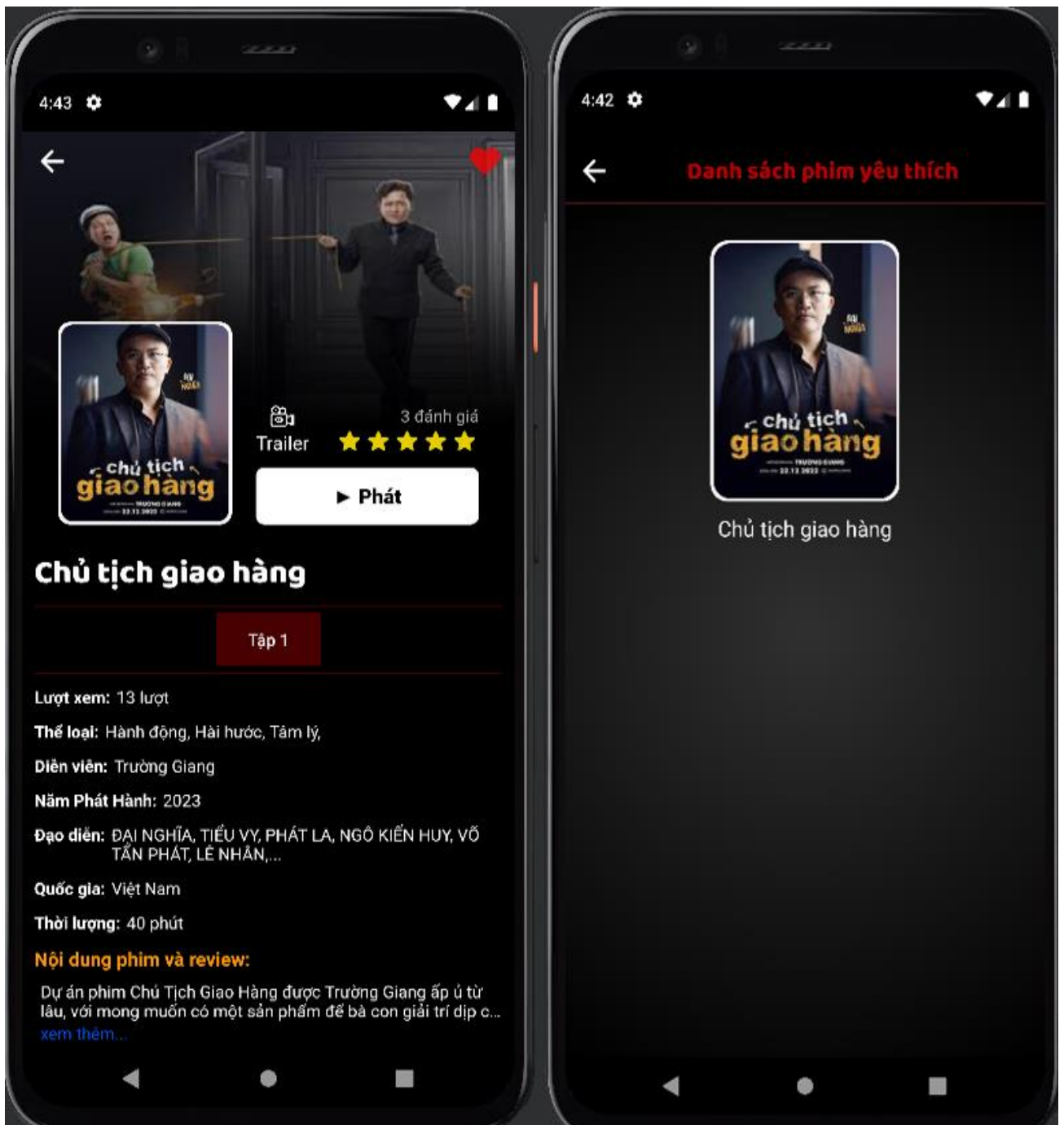
**Right Screen (4:32):**

- Tên phim:
- Diễn viên:
- Đạo diễn:
- Thời lượng (Phút):
- Năm phát hành:
- Quốc gia:
- Nội dung phim:
- Trailer link:
- Thêm**

Hình 29 Giao diện thêm phim

#### 4.7 Giao diện yêu thích và danh sách phim yêu thích

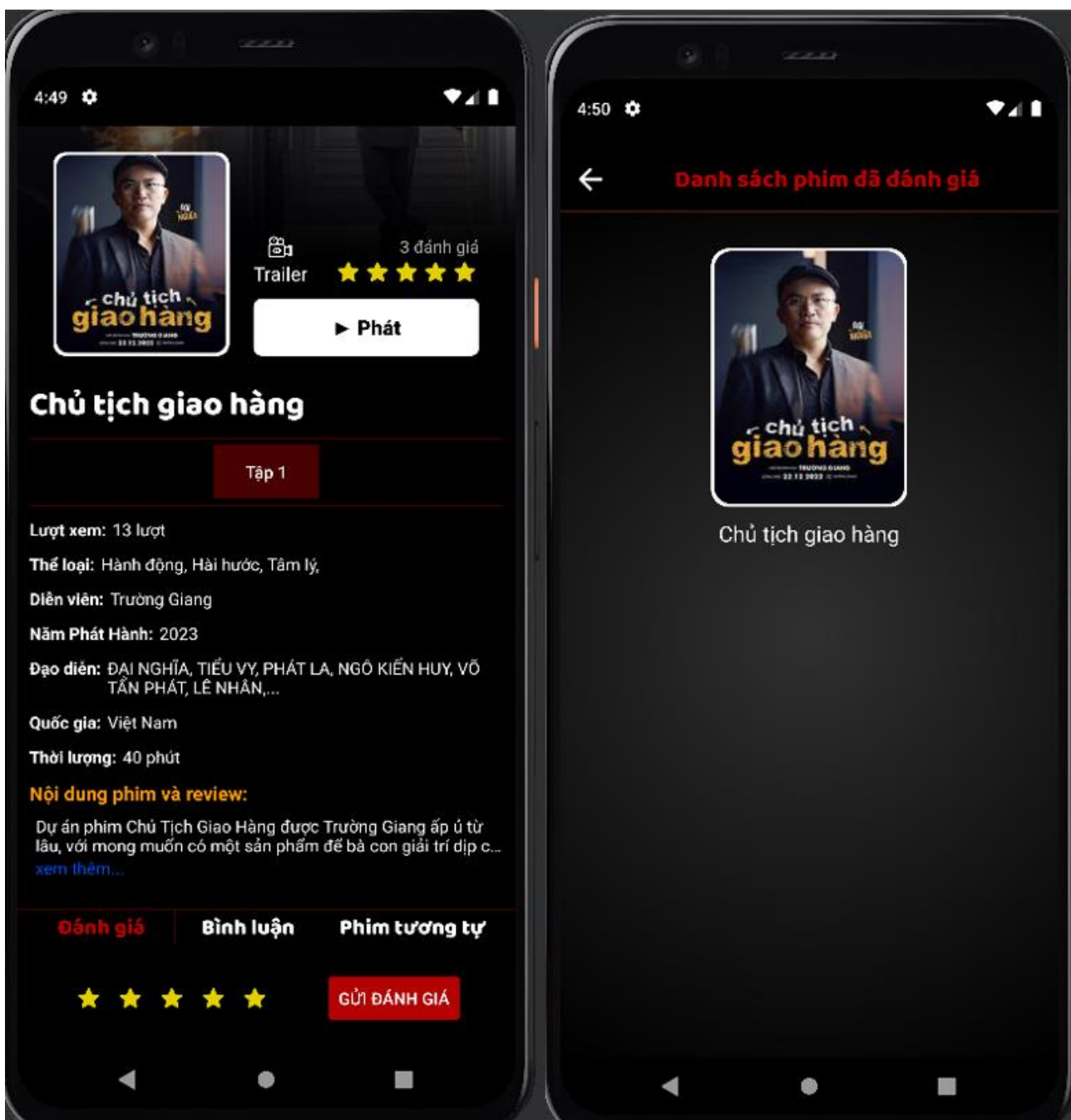
Khi người dùng nhấn vào yêu thích phim thì những bộ phim mà người dùng đã yêu thích sẽ lưu vào danh mục danh sách phim mà mình đã yêu thích



Hình 30 Giao diện yêu thích và danh sách phim yêu thích

#### 4.8 Giao diện đánh giá và danh sách phim đã đánh giá

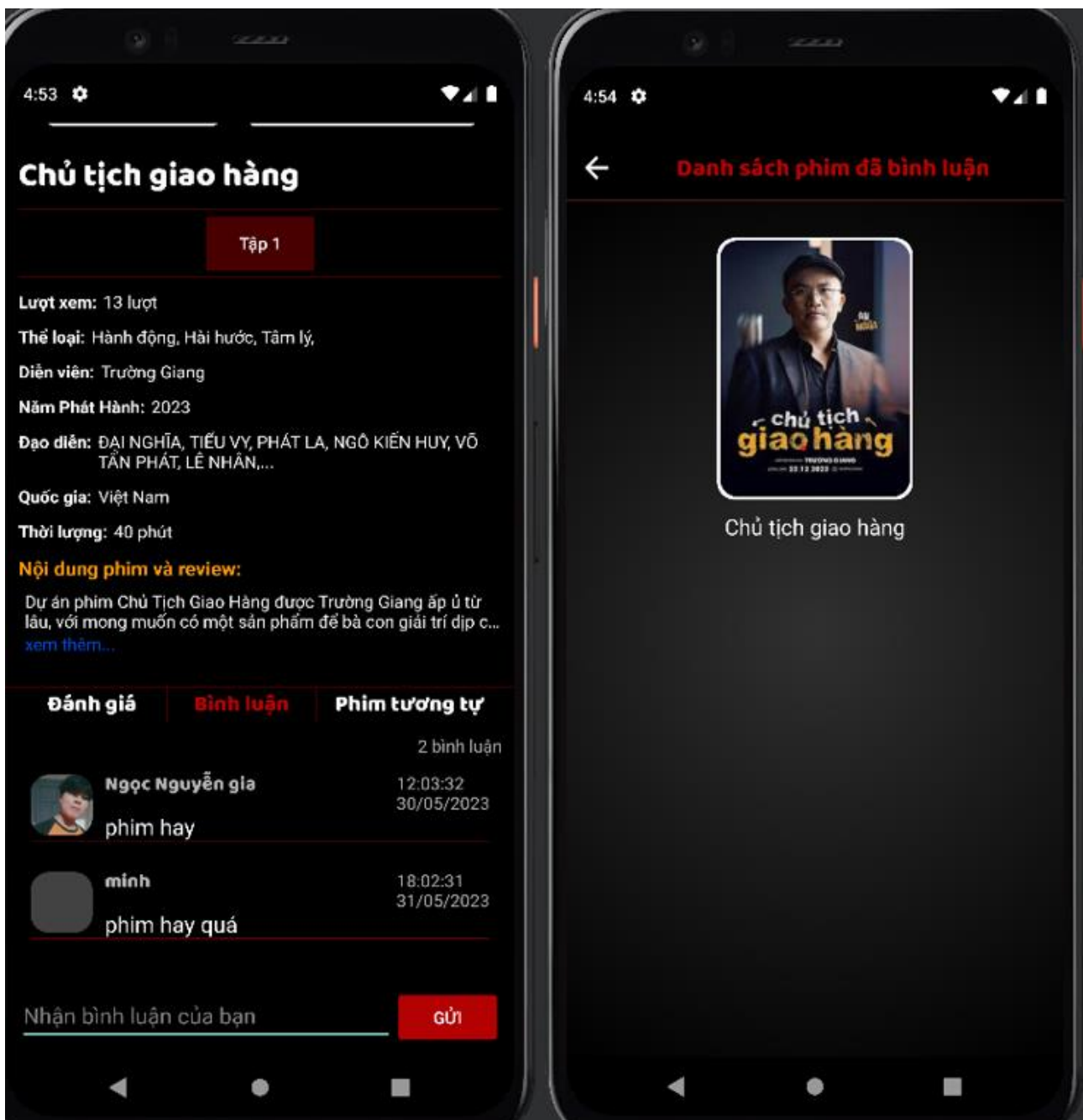
Khi người dùng đánh giá phim thì những bộ phim mà người dùng đã đánh giá sẽ lưu vào danh mục danh sách phim mà mình đã đánh giá



Hình 31 Giao diện yêu đánh giá và danh sách đánh giá

#### 4.9 Giao diện bình luận và danh sách phim đã bình luận

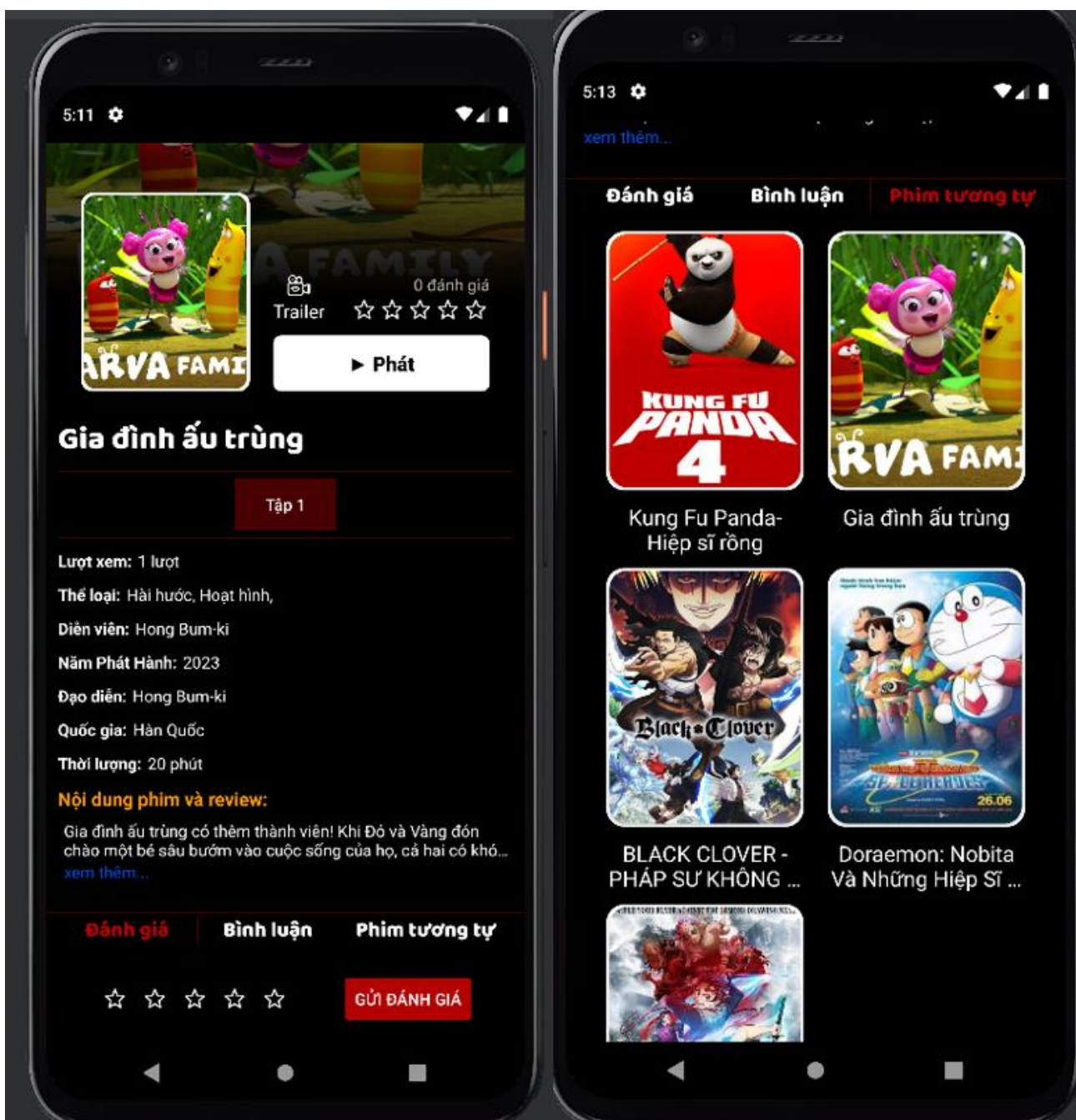
Khi người dùng bình luận một bộ phim mà mình đã xem thì những bộ phim mà người dùng đã bình luận sẽ lưu vào danh sách các phim mà người dùng đã bình luận



Hình 32 Giao diện bình luận và danh sách phim đã bình luận

#### 4.10 Giao diện đề xuất phim

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



Hình 33 Giao diện đề xuất phim



#### 4.11 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 34 Giao diện xem phim*

# KẾT LUẬN VÀ KIẾN NGHỊ

## 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.

Về mặt chức năng, hệ thống hỗ trợ các tính năng cơ bản trong việc giúp người dùng thuận tiện trong việc mua hàng, tích hợp thanh toán trực tuyến. Đối với app quản lý giúp quản trị viên dễ dàng quản lý thông tin.

## 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. Đưa ra những bộ phim tương tự, phim gợi ý cho người dùng chưa đa dạng. 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. Sử dụng quá nhiều thư viện có sẵn.

## Hướng phát triển

- Khắc phục những nhược điểm trên.
- Phát triển thêm ứng dụng có thể đăng kí tài khoản VIP, và thanh toán trực tiếp trên ứng dụng.
- Tạo thêm ứng dụng giúp quản trị viên có thể cập nhật thông tin phim trên ứng dụng
- Phát triển một ứng dụng trên nền tảng di động (android, ios) để nâng cao trải nghiệm của khách hàng, phục vụ nhu cầu sử dụng ứng dụng trên thiết bị di động.
- Phát triển thêm các công cụ gợi ý phim tốt hơn cho người dùng.
- Ứng dụng các thuật toán học máy để dự đoán phim hay xem nhiều nhất.

## TÀI LIỆU THAM KHẢO

- [1] Tác giả: Bill Phillips, Chris Stewart, Kristin Marsicano, và Joe Hummel, Android Programming: - The Big Nerd Ranch Guide (4th Edition), Nhà xuất bản: Big Nerd Ranch Guides (2019).
- [2] Tác giả: Neil Smyth - Firebase Essentials for Android, Nhà xuất bản: Payload Media (2018).
- [3] Tác giả: Suresh Kumar Gorakala. - Building Recommender Systems: An Introduction, Nhà xuất bản: Apress (2018).
- [4] Tác giả: Charu Aggarwal - Recommender Systems: The Textbook . Nhà xuất bản: Springer (2018).
- [5] <https://developer.android.com/guide?hl=vi>
- [6] <https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/>
- [7] <https://developer.android.com/studio/write/firebase?hl=vi>
- [8] <https://viblo.asia/p/tim-hieu-so-luoc-ve-firebase-Eb85oeOmZ2G>
- [9] <https://freetuts.net/tai-lieu-hoc-java-android-kotlin-lo-trinh-hoc-android-2243.html>
- [10] <https://www.javatpoint.com/android-tutorial>
- [11] <https://www.w3schools.com/sql/default.asp>
- [12] [https://www.tutorialspoint.com/firebase/firebase\\_overview.htm](https://www.tutorialspoint.com/firebase/firebase_overview.htm)
- [13] [https://github.com/Giangoc12333/DATN\\_APPXEMPHIM](https://github.com/Giangoc12333/DATN_APPXEMPHIM)