**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG XEM PHIM**

Nhóm sinh viên thực hiện: **22IT172** – Châu Văn Mẫn

Giảng viên hướng dẫn: **KS.NGUYỄN THỊ THANH THÚY**

***Đà Nẵng, 12 tháng 5 năm 2024***

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG XEM PHIM**

Nhóm sinh viên thực hiện: **22IT172** – Châu Văn Mẫn

Giảng viên hướng dẫn: **KS.NGUYỄN THỊ THANH THÚY**

***Đà Nẵng, 2 tháng 5 năm 2023***

**LỜI MỞ ĐẦU**

*Trong đời sống hiện đại ngày nay, ứng dụng công nghệ thông tin và việc tin học hoá được xem là một trong những yếu tố mang tính quyết định trong hoạt động của chính phủ, các tổ chức, cũng như các công ty, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.*

*Sự phát triển mạnh về kỹ thuật máy tính và mạng điện tử. Các công nghệ mới lần lược ra đời. Giờ đây, mọi việc liên quan đến thông tin trở nên thật dễ dàng cho người sử dụng: chỉ cần có một điện thoại có kết nối internet và từ khoá về vấn đề bạn đang tìm kiếm thì gần như lập tức cả thế giới về vấn đề bạn đang tìm kiếm sẽ hiện ra, có đầy đủ thông tin, hình ảnh và thậm chỉ có cả âm thanh nếu bạn cần. Bằng internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí được giảm thiểu thấp hơn với các hình thức truyền thống. Và phổ biến nhất hiện nay đó chính là nền tảng mobile. Có rất nhiều lĩnh vực ứng dụng trên nền tảng mobile như là quản lý nhân sự, thương mại điện tử, y tế, giáo dục và đào tạo, vui chơi giải trí,... Trong đồ án này, với mục đích xây dựng một ứng dụng xem phim, em xây dựng một ứng dụng cho phép người dùng có thể xem phim miễn phí trên ứng dụng đó. Ứng dụng được lập trình bằng ngôn ngữ kotlin với Firebase, một nền tảng đám mây do Google cung cấp.*

**LỜI CẢM ƠN**

*Để làm ra sản phẩm này, em đã nhận được sự hỗ trợ, giúp đỡ rất nhiều từ giảng viên hướng dẫn – cô KS.Nguyễn Thị Thanh Thuý. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến cô và tất cả các bạn trong nhóm đã cùng nhau giúp đỡ, cùng nhau nghiên cứu trong suốt quá trình thực hiện đề tài. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, bản báo cáo này không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến từ các thầy cô, từ đó chúng em có thể bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn cho các dự án sau này. Em xin chân thành cảm ơn!*

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN:**

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc134078062)

[1.1 Tổng quan đề tài 1](#_Toc134078063)

[1.1.1 Tên đề tài 1](#_Toc134078064)

[1.1.2 Lý do chọn đề tài 1](#_Toc134078065)

[1.1.3 Vấn đề cần giải quyết 2](#_Toc134078066)

[1.1.4 Nội dung cần thực hiện 2](#_Toc134078067)

[1.2 Nhiệm vụ thực hiện đề tài 3](#_Toc134078068)

[1.2.1 Phương pháp thực hiện đề tài 3](#_Toc134078069)

[1.2.2 Kết quả 3](#_Toc134078070)

[1.3 Cấu trúc đồ án 4](#_Toc134078071)

[CHƯƠNG 2. NGHIÊN CỨU TỔNG QUAN 5](#_Toc134078072)

[2.1 Công cụ hỗ trợ 5](#_Toc134078073)

[2.2 Tìm hiểu về các công nghệ cần sử dụng 7](#_Toc134078074)

[2.2.1 Giới thiệu về Kotlin 7](#_Toc134078075)

[2.2.2 Ưu điểm và nhược điểm của Kotlin 7](#_Toc134078076)

[2.2.3 Giới thiệu về Firebase 8](#_Toc134078077)

[2.2.4 Ưu điểm và nhược điểm của Firebase 10](#_Toc134078078)

[2.3 Tổng kết 11](#_Toc134078079)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 11](#_Toc134078080)

[3.1 Phân tích 11](#_Toc134078081)

[3.1.1 Chức năng 11](#_Toc134078082)

[3.2 Biểu đồ Use-case của hệ thống 13](#_Toc134078083)

[3.2.1 Đặc tả Use-case: Tìm kiếm 13](#_Toc134078084)

[3.2.2 Đặc tả Use-case: Xem phim 13](#_Toc134078085)

[3.2.3 Đặc tả Use-case: Đăng nhập 14](#_Toc134078086)

[3.2.4 Đặc tả Use-case: Đăng ký 14](#_Toc134078087)

[3.2.5 Đặc tả Use-case: Đánh giá 14](#_Toc134078088)

[3.2.6 Đặc tả Use-case: Xóa lịch sử 15](#_Toc134078089)

[3.2.7 Đặc tả Use-case: Thay đổi tài khoản 15](#_Toc134078090)

[3.3 Biểu đồ hoạt động 16](#_Toc134078091)

[3.3.1 Biểu đồ hoạt động tìm kiếm 16](#_Toc134078092)

[3.3.2 Biểu đồ hoạt động xem phim 17](#_Toc134078093)

[3.3.3 Biểu đồ hoạt động đăng nhập 18](#_Toc134078094)

[3.3.4 Biểu đồ hoạt động đăng ký 19](#_Toc134078095)

[3.3.5 Biểu đồ hoạt động đánh giá 20](#_Toc134078096)

[3.3.6 Biểu đồ hoạt động xóa lịch sử 21](#_Toc134078097)

[3.3.7 Biểu đồ hoạt động thêm yêu thích 22](#_Toc134078098)

[3.3.8 Biểu đồ hoạt động thay đổi tài khoản 23](#_Toc134078099)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc134078100)

[4.1 Kết luận 24](#_Toc134078101)

[4.1.1 Kết quả đạt được 24](#_Toc134078102)

[4.1.2 Hạn chế 24](#_Toc134078103)

[4.2 Một số hình ảnh ứng dụng 25](#_Toc134078104)

**DANH MỤC HÌNH ẢNH**

[**Hình 1.1:** Cơ sở dữ liệu đám mây Firebase 3](#_Toc134077931)

[**Hình 2.1:** Giao diện Android Studio 6](#_Toc134077932)

[**Hình 2.2:** Giao diện quản lý cơ sở dữ liệu Firebase 6](#_Toc134077933)

[**Hình 2.3:** Ngôn ngữ Kotlin 7](#_Toc134077934)

[**Hình 2.4:** Firebase 10](#_Toc134077935)

[**Hình 3.1** Tổng quan chức năng của người dùng 12](#_Toc134077936)

[**Hình 3.3** Biểu đồ Use-case của hệ thống 13](#_Toc134077937)

[**Hình 3.4** Biểu đồ hoạt động tìm kiếm 16](#_Toc134077938)

[**Hình 3.5** Biểu đồ hoạt động xem phim 17](#_Toc134077939)

[**Hình 3.6** Biểu đồ hoạt động đăng nhập 18](#_Toc134077940)

[**Hình 3.7** Biểu đồ hoạt động đăng ký 19](#_Toc134077941)

[**Hình 3.8** Biểu đồ hoạt động đánh giá 20](#_Toc134077942)

[**Hình 3.9** Biểu đồ hoạt động xóa lịch sử 21](#_Toc134077943)

[**Hình 3.10:** Biểu đồ hoạt động thêm yêu thích 22](#_Toc134077944)

[**Hình 3.11** Biểu đồ hoạt động thay đổi tài khoản 23](#_Toc134077945)

# GIỚI THIỆU

## Tổng quan đề tài

### Tên đề tài

* Xây dựng ứng dụng xem phim

### Lý do chọn đề tài

* Sự phát triển của ngành công nghiệp giải trí trực tuyến: Trong những năm gần đây, nhu cầu xem phim trực tuyến đã tăng đáng kể, và nền tảng phát trực tuyến như Netflix, Amazon Prime Video, Disney+ và Hulu đã trở thành phổ biến với người dùng. Việc xây dựng một ứng dụng xem phim sẽ giúp đáp ứng nhu cầu ngày càng tăng của người dùng.
* Thị trường tiềm năng: Thị trường xem phim trực tuyến vẫn đang tiếp tục phát triển, đặc biệt là ở các quốc gia đang phát triển. Việc phát triển một ứng dụng xem phim sẽ mang lại tiềm năng thị trường lớn và cơ hội kinh doanh.
* Công nghệ đang phát triển: Công nghệ phát triển ngày càng tiên tiến, điều này đồng nghĩa với việc xây dựng ứng dụng xem phim trực tuyến có thể được phát triển để đáp ứng các yêu cầu của người dùng về chất lượng hình ảnh, âm thanh và tính năng.
* Tính cạnh tranh: Với nhiều ứng dụng xem phim trực tuyến hiện có trên thị trường, việc phát triển một ứng dụng xem phim có thể giúp tăng tính cạnh tranh và giúp bạn thu hút được nhiều người dùng.
* Kết nối cộng đồng: Ứng dụng xem phim có thể là một cách để kết nối cộng đồng người dùng, cho phép họ trao đổi, thảo luận và chia sẻ các ý kiến ​​về các bộ phim đang hot.Vấn đề cần giải quyết
* Vì những lý do trên, xây dựng ứng dụng xem phim là một đề tài hấp dẫn và có tiềm năng phát triển trong tương lai.

### Vấn đề cần giải quyết

* Khi xây dựng ứng dụng xem phim, có một số vấn đề cần giải quyết để đảm bảo ứng dụng được hoạt động tốt và đáp ứng được nhu cầu của người dùng. Dưới đây là một số vấn đề cần giải quyết:
* Tốc độ tải trang: Việc tải phim nhanh và mượt là một yếu tố quan trọng để người dùng có thể trải nghiệm tốt hơn. Do đó, cần phải tối ưu hóa tốc độ tải trang để đảm bảo người dùng không bị gián đoạn khi xem phim.
* Tính bảo mật: Việc bảo vệ thông tin người dùng và thông tin tài khoản của họ là rất quan trọng. Cần phải đảm bảo rằng ứng dụng có các biện pháp bảo mật đủ mạnh để bảo vệ thông tin của người dùng.
* Tìm kiếm và phân loại phim: Để giúp người dùng dễ dàng tìm kiếm và lựa chọn phim, cần phải có các tính năng tìm kiếm và phân loại phim dễ sử dụng và hiệu quả.
* Cập nhật phim mới nhất: Việc cập nhật liên tục các bộ phim mới nhất sẽ giúp ứng dụng luôn hấp dẫn và thu hút được nhiều người dùng.
* Độ phân giải và chất lượng phim: Để đáp ứng được nhu cầu người dùng, ứng dụng cần hỗ trợ độ phân giải và chất lượng phim tốt, đảm bảo người dùng có trải nghiệm xem phim tốt nhất.

### Nội dung cần thực hiện

* Xây dựng một ứng dụng với giao diện đơn giản, dễ sử dụng, dễ dàng thu hút được người dùng.
* Xây dựng các chức năng cơ bản của một ứng dụng xem phim cho người dùng dễ dàng sử dụng.

## Nhiệm vụ thực hiện đề tài

### Phương pháp thực hiện đề tài

* Nắm vững các kiến thức cơ bản của ngôn ngữ kotlin, sử dụng thành thạo công cụ Android Studio và các tính năng ở trong Firebase. Hiểu rõ cách quản lý dữ liệu trong Firebase



Hình .: Cơ sở dữ liệu đám mây Firebase

### Kết quả

* Ứng dụng được thiết kế đơn giản, đẹp mắt, dễ sử dụng. Nội dung được bố trí một cách khoa học
* Mục tiêu của đồ án là có thể áp dụng vào thực tế. Ứng dụng sẽ cung cấp các chức năng cơ bản cần có của một ứng dụng xem phim cho người dùng, thuận tiện cho việc trải nghiệm của người dùng.

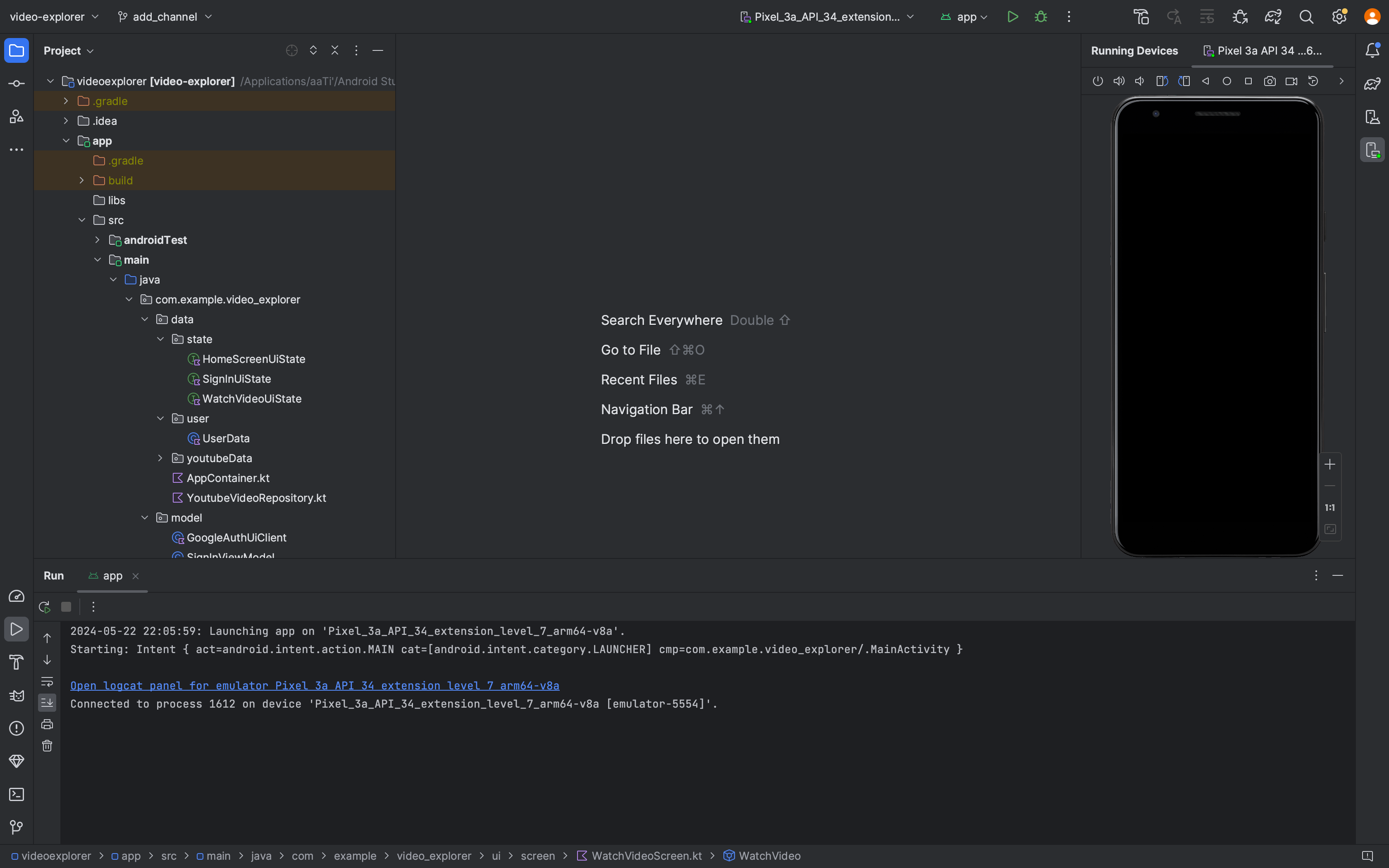
## Cấu trúc đồ án

* Đồ án gồm 4 phần chính:
* Chương 1: Giới thiệu: giới thiệu tổng quan về đề tài, phương pháp thực hiện và mục tiêu của đề tài.
* Chương 2: Nghiên cứu tổng quan: trình bày kiến thức tổng quan về các công nghệ được sử dụng để xây dựng một ứng dụng.
* Chương 3: Phân tích thiết kế hệ thống: trình bày các chức năng và giao diện của ứng dụng.
* Chương 4: Kết luận và hướng phát triển: Tổng kết lại đề tài và đưa ra các phương hướng phát triền.

# NGHIÊN CỨU TỔNG QUAN

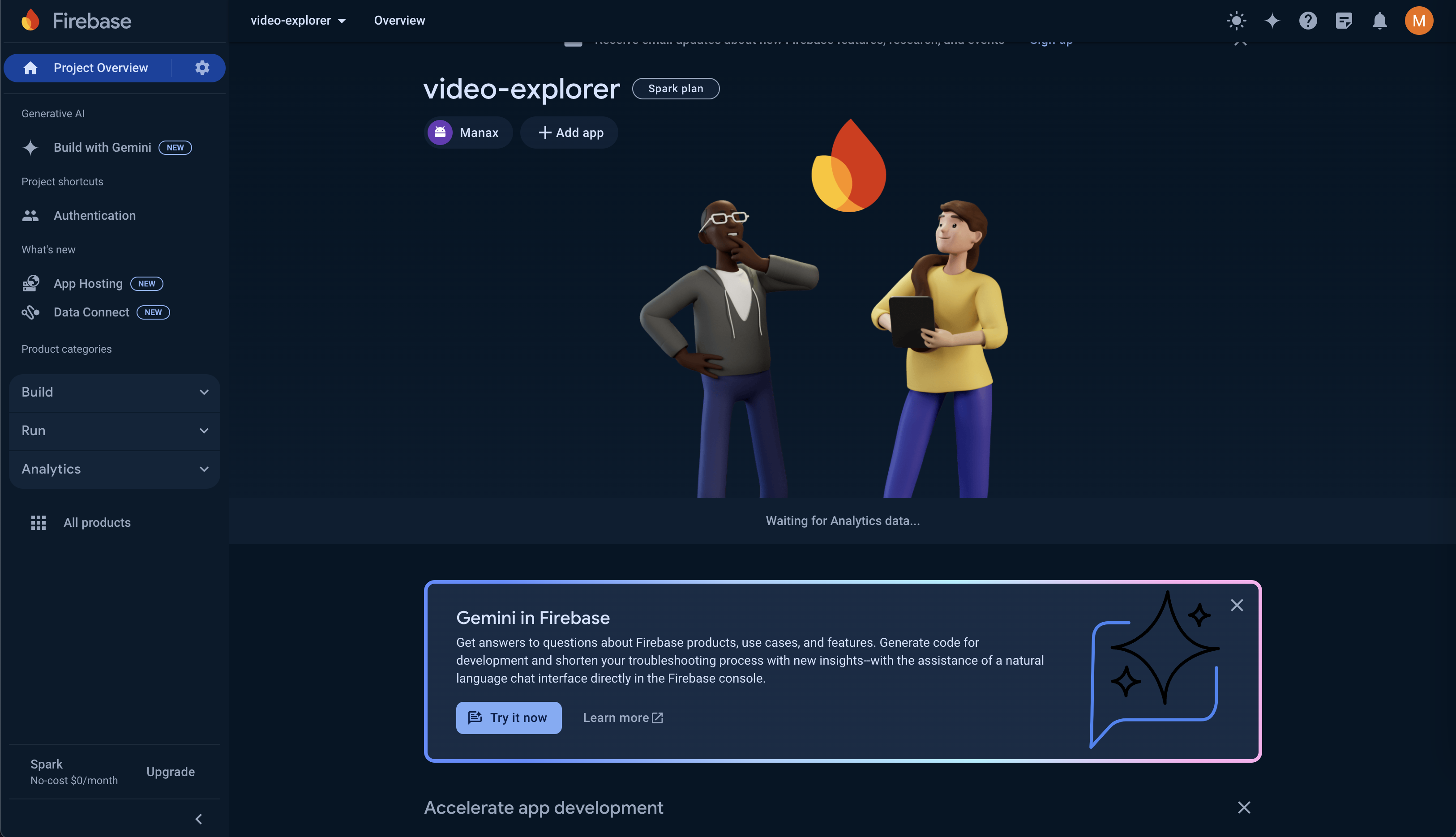
## Công cụ hỗ trợ

* Công cụ được sử dụng để xây dựng ứng dụng là Android Studio.
* Android Studio là một môi trường phát triển tích hợp (IDE) dành cho việc phát triển ứng dụng di động chạy trên hệ điều hành Android. Nó được phát triển bởi Google và là IDE chính thức để phát triển ứng dụng Android.
* Android Studio được xây dựng trên nền tảng của IntelliJ IDEA, một IDE được sử dụng phổ biến trong lập trình Java. Android Studio cung cấp cho các nhà phát triển một môi trường đầy đủ các công cụ cần thiết để phát triển ứng dụng Android, bao gồm một trình biên dịch, trình gỡ lỗi, trình quản lý phiên bản, trình phân tích cú pháp, thiết kế giao diện, mô phỏng thiết bị, và nhiều tính năng khác.
* Android Studio cũng tích hợp với Android SDK (Software Development Kit), bao gồm các thư viện, tài liệu và công cụ hỗ trợ phát triển ứng dụng Android. Nó cũng cung cấp cho các nhà phát triển khả năng kiểm tra ứng dụng trên nhiều thiết bị khác nhau và định dạng màn hình khác nhau.



Hình .: Giao diện Android Studio

* Về phần công cụ cơ sở dữ liệu: Công cụ được sử dụng để xây dựng, quản lý cơ sở dữ liệu Firebase.



Hình .: Giao diện quản lý cơ sở dữ liệu Firebase

## Tìm hiểu về các công nghệ cần sử dụng

### Giới thiệu về Kotlin



Hình .: Ngôn ngữ Kotlin

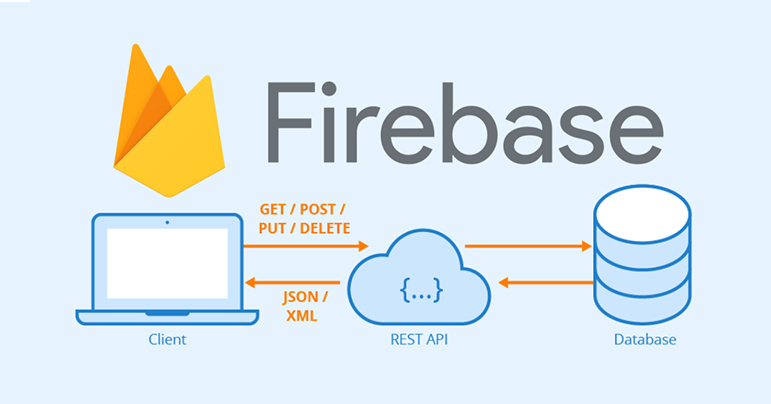
* Kotlin là một ngôn ngữ lập trình đa nền tảng (multi-platform) được phát triển bởi JetBrains, cùng với các công cụ lập trình như IntelliJ IDEA, PhpStorm và PyCharm. Kotlin được thiết kế để tránh một số lỗ hổng bảo mật và vấn đề khác trong Java và cung cấp cho lập trình viên nhiều tính năng hiện đại hơn để viết mã.
* Kotlin hỗ trợ viết mã cho nhiều nền tảng, bao gồm Android, JVM, JavaScript và Native. Điều này cho phép lập trình viên sử dụng cùng một mã để phát triển ứng dụng cho nhiều nền tảng khác nhau, giảm thiểu thời gian và chi phí phát triển.
* Kotlin ngày càng trở nên phổ biến trong cộng đồng lập trình viên và được sử dụng rộng rãi trong các dự án phát triển ứng dụng trên Android và trên các nền tảng khác.

### Ưu điểm và nhược điểm của Kotlin

* **Ưu điểm** của Kotlin:
* Cú pháp đơn giản: Kotlin có cú pháp đơn giản, dễ đọc và dễ viết hơn so với Java. Vì vậy, các lập trình viên có thể dễ dàng học và sử dụng Kotlin mà không mất nhiều thời gian.
* An toàn hơn: Kotlin loại bỏ một số lỗ hổng bảo mật trong Java và cung cấp nhiều tính năng để giúp lập trình viên tránh lỗi như null-pointer exceptions (NPE).
* Đa nền tảng: Kotlin hỗ trợ viết mã cho nhiều nền tảng, cho phép lập trình viên sử dụng cùng một mã cho các ứng dụng Android, JVM, JavaScript và Native.
* Tính mở rộng: Kotlin được phát triển để có thể mở rộng và cải tiến dễ dàng. Bất kỳ ai có thể tham gia phát triển Kotlin, tạo ra các thư viện mới và phát triển các tính năng mới cho Kotlin.
* Tính tương thích: Kotlin có thể sử dụng chung với mã nguồn Java, cho phép các lập trình viên chuyển đổi từ Java sang Kotlin dễ dàng.
* **Nhược điểm** của Kotlin:
* Thời gian chuyển đổi: Mặc dù Kotlin có thể sử dụng chung với mã nguồn Java, nhưng việc chuyển đổi từ Java sang Kotlin vẫn mất thời gian.
* Tính năng tùy chọn: Kotlin có nhiều tính năng tùy chọn, điều này có thể khiến việc học và sử dụng Kotlin trở nên phức tạp đối với những người mới bắt đầu.
* Kích thước tệp: Tệp .kt của Kotlin có kích thước lớn hơn so với tệp .java tương ứng. Điều này có thể làm cho quá trình tải ứng dụng chậm hơn.

### Giới thiệu về Firebase

* Firebase là một nền tảng công nghệ điện toán đám mây (cloud computing platform) được cung cấp bởi Google, được thiết kế để giúp phát triển ứng dụng web và di động dễ dàng hơn. Firebase cung cấp các công cụ để phát triển và quản lý các ứng dụng như cơ sở dữ liệu, đăng nhập và xác thực, lưu trữ, push notification, analytics, testing, hosting và nhiều tính năng khác.
* Firebase được phát triển trên nền tảng đám mây của Google, nó cho phép các nhà phát triển tập trung vào việc phát triển ứng dụng mà không phải lo lắng về việc quản lý cơ sở hạ tầng phức tạp. Firebase được tích hợp sẵn với các công nghệ của Google như Google Cloud Platform, Google Analytics, Google Ads và nhiều sản phẩm khác, giúp các nhà phát triển tạo ra các ứng dụng có khả năng tích hợp cao và mang lại trải nghiệm người dùng tốt hơn.
* Firebase cũng có thể tích hợp với nhiều ngôn ngữ lập trình khác nhau như JavaScript, Swift, Java, Kotlin, C++ và nhiều ngôn ngữ lập trình khác.
* Các tính năng chính của Firebase bao gồm:
* Realtime Database: Cơ sở dữ liệu thời gian thực cho các ứng dụng web và di động.
* Authentication: Đăng nhập và xác thực người dùng thông qua các tài khoản mạng xã hội và các nhà cung cấp khác.
* Cloud Storage: Lưu trữ tệp dữ liệu trên đám mây, bao gồm tệp hình ảnh, âm thanh, video và các tệp khác.
* Cloud Messaging: Push notification để giữ liên lạc với người dùng.
* Hosting: Dịch vụ lưu trữ trang web để phát triển và triển khai ứng dụng web.
* Analytics: Theo dõi và phân tích dữ liệu về lượng truy cập, sử dụng và tương tác của người dùng với ứng dụng.
* Test Lab: Công cụ để thử nghiệm và kiểm tra ứng dụng trên các thiết bị khác nhau.
* Với những tính năng đầy đủ và linh hoạt, Firebase trở thành một công cụ hữu ích cho các nhà phát triển ứng dụng để phát triển, triển khai và quản lý các ứng dụng web và di động một cách dễ dàng và hiệu quả.

Hình .: Firebase

### Ưu điểm và nhược điểm của Firebase

* **Ưu điểm** của Firebase:
* **Dễ dàng sử dụng**: Firebase cung cấp cho các nhà phát triển một giao diện đơn giản và dễ sử dụng, giúp cho việc triển khai các tính năng như đăng nhập, lưu trữ dữ liệu, push notification, hosting,... trở nên dễ dàng hơn.
* **Tính khả dụng cao**: Firebase được cung cấp dưới dạng dịch vụ đám mây, giúp cho ứng dụng của bạn luôn sẵn sàng hoạt động trên nhiều thiết bị khác nhau và không bị gián đoạn do sự cố về phần cứng hoặc phần mềm.
* **Bảo mật**: Firebase cung cấp nhiều tính năng bảo mật để bảo vệ dữ liệu của người dùng, bao gồm cơ chế xác thực, mã hóa dữ liệu, kiểm tra phân quyền và giám sát.
* **Tích hợp tốt**: Firebase có thể tích hợp tốt với nhiều ngôn ngữ lập trình khác nhau như JavaScript, Swift, Java, Kotlin, C++ và nhiều ngôn ngữ khác, giúp cho các nhà phát triển có thể dễ dàng sử dụng Firebase để phát triển ứng dụng của mình.
* **Giá cả phù hợp**: Firebase cung cấp các gói dịch vụ khác nhau với các mức giá phù hợp cho nhiều đối tượng khác nhau, từ các ứng dụng nhỏ cho đến các ứng dụng lớn.
* **Nhược điểm** của Firebase
* **Tính linh hoạt hạn chế**: Firebase cung cấp các tính năng cơ bản và được giới hạn trong một số trường hợp, do đó các nhà phát triển có thể cần phải tìm kiếm các giải pháp khác để đáp ứng nhu cầu của ứng dụng của mình.
* **Khả năng mở rộng có hạn**: Firebase có một số hạn chế trong việc mở rộng các tính năng của ứng dụng khi số lượng người dùng tăng lên đáng kể.
* **Hiệu suất có thể bị giảm**: Vì Firebase là một dịch vụ đám mây, nó có thể bị ảnh hưởng bởi tình trạng mạng

## Tổng kết

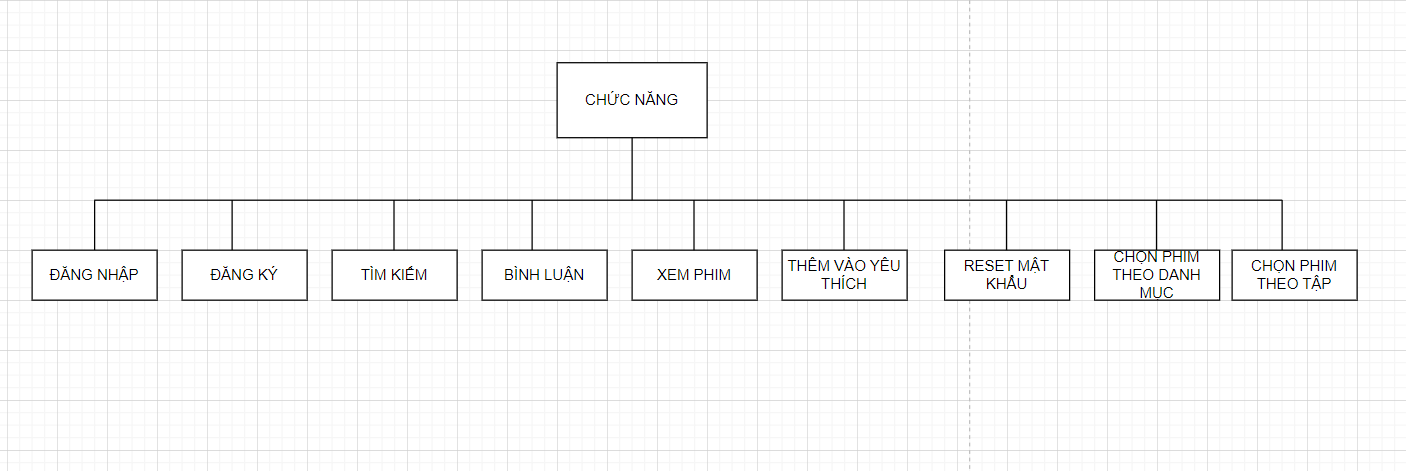
* Để xây dựng được một ứng dụng xem phim hoàn chỉnh, người thực hiện cần nằm rõ vấn đề sau:
* Xây dựng được giao diện ứng dụng thông qua các công nghệ như Kotlin.
* Nắm vững kiến thức về lập trình Back-end và Front-end theo ngôn ngữ Kotlin để phát triển sản phẩm.
* Nắm vững kiến thức về công nghệ điện toán đám mây(Firebase), sử dụng được các tính năng trong Firebase.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Phân tích

### Chức năng

* Dưới góc độ người dùng, Ứng dụng gồm có các chức năng sau:



Hình . Tổng quan chức năng của người dùng

* Đăng nhập: Người dùng đăng nhập vào trang web để thực hiện các chức năng ví dụ như mua sản phẩm
* Đăng ký: Người dùng đăng ký tài khoản để sử dụng các dịch vụ mua hàng của Website
* Tìm kiếm: Người dùng có thể tìm kiếm phim hoặc tìm kiếm theo danh mục bằng bộ lọc
* Bình luận: Người dùng có thể bình luận phim mà người dùng đã xem
* Xem phim: Người dùng có thể xem phim trực tiếp trên ứng dụng, có thể thực hiện các thao tác như tua phim, xoay màn hình, dừng/ phát phim
* Thêm vào yêu thích: Người dùng có thể thêm phim vào danh sách yêu thích để theo dõi các tập tiếp theo
* Reset mật khẩu: Người dùng có thể reset mật khẩu khi quên mật khẩu thông qua Email/ Tên đăng nhập của người dùng
* Chọn phim theo danh mục: Người dùng có thể chọn phim theo danh mục như Truyền hình, Thể thao, Hoạt hình…
* Chọn phim theo tập: Người dùng có thể chọn từng tập phim trực tiếp dễ dàng trực tiếp trên ứng dụng

## Biểu đồ Use-case của hệ thống



Hình . Biểu đồ Use-case của hệ thống

### Đặc tả Use-case: Tìm kiếm

* Tên ca sử dụng: Tìm kiếm
* Mô tả ngắn gọn: Use-case này cung cấp chức năng tìm kiếm tên phim hoặc là tìm kiếm bằng giọng nói hoặc có thể tìm kiếm theo danh mục.
* Tác nhân: Người dùng (Khách hàng)
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng ấn vào icon “Tìm kiếm”
* Ứng dụng sẽ tạo ra một trang để hiển thị kết quả tìm kiếm.
* Ứng dụng sẽ hiển thị ra phim nếu tìm thấy kết quả.
* Còn nếu không tìm thấy kết quả thì sẽ không hiển thị nội dung gì.

### Đặc tả Use-case: Xem phim

* Tên ca sử dụng: Xem phim
* Mô tả ngắn gọn: Use-case này cung cấp chức năng cho người dùng xem phim khi click vào các bô phim và tập phim mình muốn xem.
* Tác nhân: Người dùng (Khách hàng)
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng ấn vào nút “Xem phim” hoặc là bộ phim nào đó

### Đặc tả Use-case: Đăng nhập

* Tên ca sử dụng: Đăng nhập
* Mô tả ngắn gọn: Use-case này cung cấp chức năng đăng nhập vào ứng dụng để người dùng có thể sử dụng các chức năng khác trong ứng dụng.
* Tác nhân: Người dùng (Khách hàng)
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng ấn vào Đăng nhập thì hệ thống sẽ đưa ra 1 cái form cho người dùng nhập thông tin tài khoản.
* Sau khi người dùng nhập xong thông tin tài khoản và ấn đăng nhập thì hệ thống sẽ đưa người dùng đến trang chủ chính.

### Đặc tả Use-case: Đăng ký

* Tên ca sử dụng: Đăng ký
* Mô tả ngắn gọn: Use-case này cung cấp chức năng chức năng đăng ký tài khoản nếu như người dùng chưa có tài khoản
* Tác nhân: Người dùng (Khách hàng)
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng ấn vào Đăng ký thì hệ thống sẽ đưa ra 1 cái form cho người dùng nhập thông tin tài khoản.
* Sau khi người dùng nhập xong thông tin tài khoản và ấn đăng ký mà hợp lệ thì hệ thống sẽ thông báo thành công và đưa người dùng đến trang đăng nhập.

### Đặc tả Use-case: Đánh giá

* Tên ca sử dụng: Bình luận
* Mô tả ngắn gọn: Use-case này cung cấp chức năng đánh giá bộ phim mà mình đang xem để Admin biết được cảm nhận của người dùng và cải thiện ứng dụng.
* Tác nhân: Người dùng (Khách hàng)
* Luồng cơ bản:
* Sau khi người dùng xem 1 một bộ phim dưới đó sẽ có đánh giá về bộ phim đó.
* Khi người dùng để lại lời đánh giá và ấn vào nút gửi bên cạnh thì đánh giá đó sẽ hiển thị dưới bình luận của bộ phim đó.

### Đặc tả Use-case: Xóa lịch sử

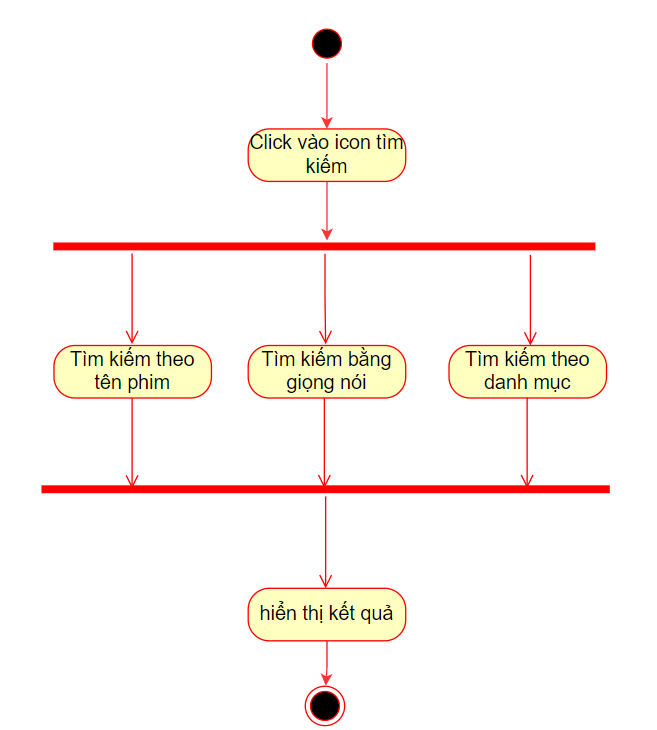
* Tên ca sử dụng: Thêm bộ phim vào yêu thích
* Mô tả ngắn gọn: Use-case này cung cấp chức năng thêm yêu thích 1 bộ phim nào đó trên ứng dụng.
* Tác nhân: Người dùng
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng nhấn vào icon thêm yêu thích.
* Khi người dùng ấn vào thêm yêu thích, bộ phim ấy sẽ được lưu trữ vào 1 trang khác.

### Đặc tả Use-case: Thay đổi tài khoản

* Tên ca sử dụng: Thay đổi tài khoản
* Mô tả ngắn gọn: Use-case này cung cấp chức năng thay đổi thông tin người dùng và thay đổi mật khẩu.
* Tác nhân: Người dùng
* Luồng cơ bản:
* Use-case bắt đầu khi người dùng ấn vào thanh menu bên cạnh thì ứng dụng sẽ hiển thị ra thông tin tài khoản của người dùng.
* Sau đó người dùng có thể chọn chức năng để thay đổi thông tin theo mong muốn.

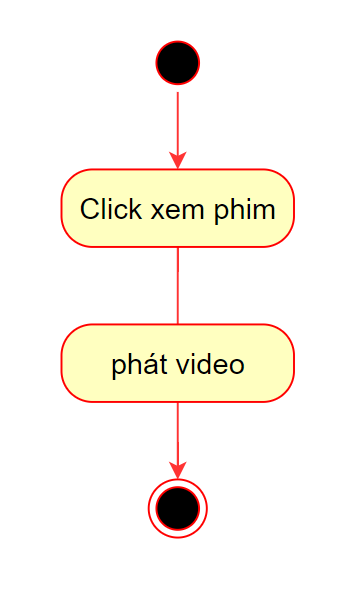
## Biểu đồ hoạt động

### Biểu đồ hoạt động tìm kiếm



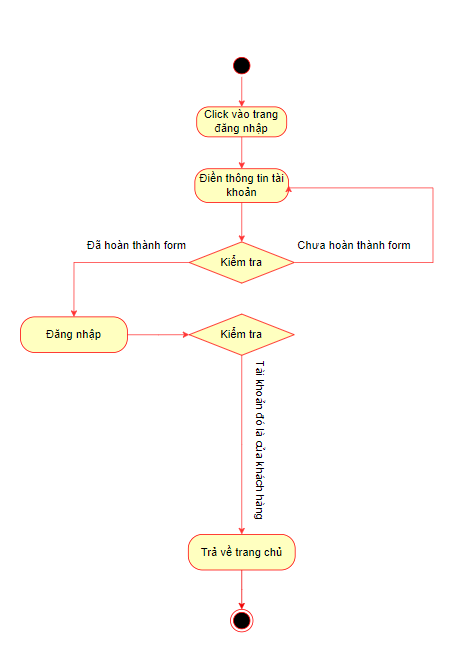
Hình . Biểu đồ hoạt động tìm kiếm

### Biểu đồ hoạt động xem phim



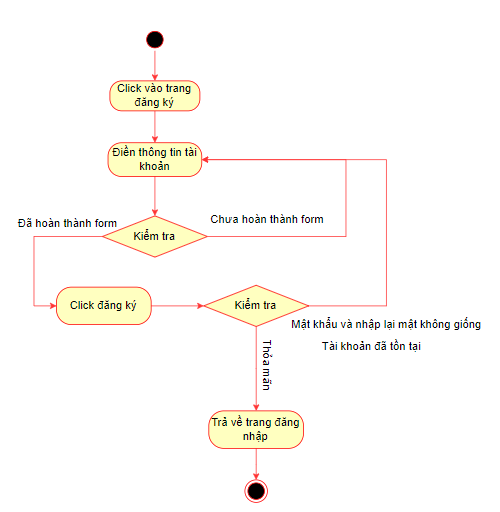
Hình . Biểu đồ hoạt động xem phim

### Biểu đồ hoạt động đăng nhập



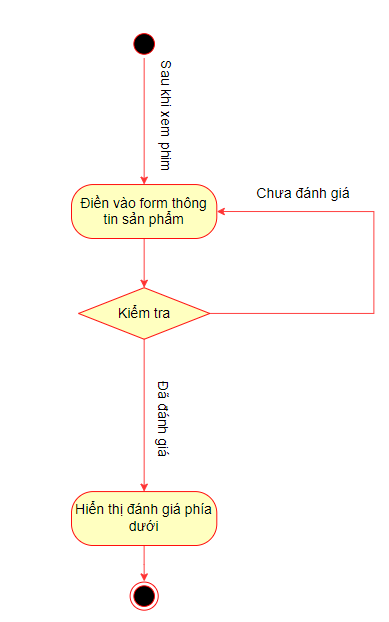
Hình . Biểu đồ hoạt động đăng nhập

### Biểu đồ hoạt động đăng ký



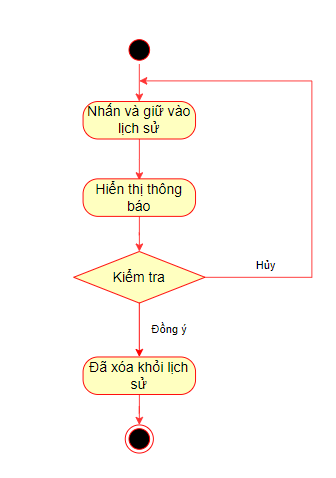
Hình . Biểu đồ hoạt động đăng ký

### Biểu đồ hoạt động đánh giá



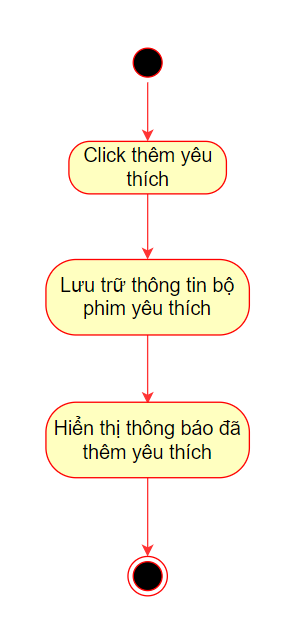
Hình . Biểu đồ hoạt động đánh giá

### Biểu đồ hoạt động xóa lịch sử



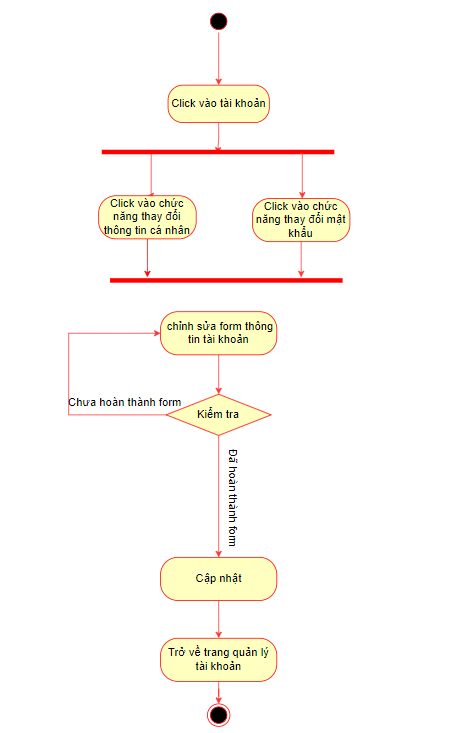
Hình . Biểu đồ hoạt động xóa lịch sử

### Biểu đồ hoạt động thêm yêu thích



Hình .: Biểu đồ hoạt động thêm yêu thích

### Biểu đồ hoạt động thay đổi tài khoản



Hình . Biểu đồ hoạt động thay đổi tài khoản

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Kết quả đạt được

* Hoàn thiện được giao diện ứng dụng xem phim bằng ngôn ngữ kotlin.
* Hoàn thiện được các chức năng cơ bản của một ứng dụng xem phim bằng ngôn ngữ kotlin.

### Hạn chế

* Tiếp tục hoàn thiện các chức năng nâng cao, bổ sung các tính năng mới cho ứng dụng.
* Nâng cấp phần giao diện sao cho hợp với từng thời điểm và dễ dàng sử dụng.
* Cải thiện lại chất lượng xem khi xem phim, tải lại tập mới,…

## Một số hình ảnh ứng dụng





