**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO MÔN HỌC CÔNG NGHỆ LẬP TRÌNH**

**ĐA NỀN TẢNG CHO ỨNG DỤNG DI ĐỘNG**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG ỨNG DỤNG XEM VIDEO**

**CHO ĐIỆN THOẠI DI ĐỘNG**

Giảng viên hướng dẫn: **ThS. Nguyễn Thanh Truyền**

Sinh viên thực hiện: **Nguyễn Văn Hiền**

**Lê Nguyễn Anh Dũng**

**Phan Thanh Hải**

**Trần Bá Minh**

Lớp: **08\_ĐH\_CNPM**

Khóa: **2019 – 2023**

***TP. Hồ Chí Minh, tháng 08 năm 2023***

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO MÔN HỌC CÔNG NGHỆ LẬP TRÌNH**

**ĐA NỀN TẢNG CHO ỨNG DỤNG DI ĐỘNG**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG ỨNG DỤNG XEM VIDEO**

**CHO ĐIỆN THOẠI DI ĐỘNG**

Giảng viên hướng dẫn: **ThS. Nguyễn Thanh Truyền**

Sinh viên thực hiện: **Nguyễn Văn Hiền**

**Lê Nguyễn Anh Dũng**

**Phan Thanh Hải**

**Trần Bá Minh**

Lớp: **08\_ĐH\_CNPM**

Khóa: **2019 – 2023**

***TP. Hồ Chí Minh, tháng 08 năm 2023***

**LỜI MỞ ĐẦU**

Chúng ta đang sống trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu giải trí và xem video trực tuyến trên điện thoại di động ngày càng tăng cao. Để đáp ứng nhu cầu này và mang đến trải nghiệm xem video thuận tiện, đa dạng và tốt nhất cho người dùng, nhóm em xin trân trọng giới thiệu đến thầy dự án "XÂY DỰNG ỨNG DỤNG XEM VIDEO CHO ĐIỆN THOẠI DI ĐỘNG".

Ứng dụng của nhóm em sẽ là nơi tổng hợp và cung cấp các video chất lượng cao từ nhiều nguồn khác nhau, giúp mọi người dễ dàng tìm kiếm và xem những nội dung mà bạn quan tâm. Tích hợp cơ sở dữ liệu Firebase, nhóm em muốn cung cấp cho mọi người trải nghiệm xem video mượt mà và ổn định trên điện thoại di động của mình.

Điểm nổi bật của dự án:

* Danh mục video đa dạng: Ứng dụng sẽ cung cấp danh mục video phong phú, bao gồm nhiều thể loại và chủ đề khác nhau, giúp mọi người dễ dàng lựa chọn nội dung yêu thích.
* Chia sẻ và đánh giá: Ứng dụng cho phép chúng ta chia sẻ video với bạn bè và gia đình cũng như đánh giá nội dung, góp phần xây dựng cộng đồng người dùng tích cực.
* Giao diện người dùng thân thiện: Giao diện được thiết kế đơn giản và dễ sử dụng, giúp bạn tận hưởng trải nghiệm xem video một cách thoải mái.

Với sự hướng dẫn của thầy Nguyễn Thanh Truyền, nhóm em quyết định chọn đề tài: *“Xây dựng ứng dụng xem video cho điện thoại di động”*, nhằm phục vụ nhu cầu giải trí của mọi người và học hỏi điều hay thông qua những video.

**LỜI CẢM ƠN**

Trong suốt quá trình học tập và hoàn thành đồ án này, nhóm em đã nhận được rất nhiều sự hướng dẫn, giúp đỡ quý báu của các thầy cô giáo bộ môn, ban giám hiệu, gia đình và bạn bè.

Với lòng kính trọng và biết ơn sâu sắc nhóm em xin được bày tỏ lời cảm ơn chân thành tới: Ban giám hiệu, Phòng đào tạo, khoa Hệ thống thông tin và Viễn thám trường Đại học Tài Nguyên và Môi Trường đã tạo mọi điều kiện thuận lợi giúp đỡ nhóm em trong quá trình học tập và hoàn thành đồ án này. Cảm ơn thầy Nguyễn Thanh Truyền, người thầy kính mến đã hết lòng giúp đỡ, dạy bảo, động viên nhóm em trong suốt quá trình học tập và hoàn thành bài báo cáo đồ án môn học Công nghệ lập trình đa nền tảng cho ứng dụng di động.

Dù đã cố gắng nhưng không thể tránh khỏi những sai sót, nhóm em rất mong nhận được sự giúp đỡ và đóng góp của các quý thầy cô, các anh chị cùng toàn thể các sinh viên.

Một lần nữa nhóm em xin chân thành cảm ơn, chúc thầy sức khỏe và thành đạt.

**GIỚI THIỆU DỰ ÁN**

Trong bối cảnh ngày càng phát triển của công nghệ và sự lan rộng của thiết bị di động, việc xây dựng ứng dụng đáp ứng nhu cầu xem video trên điện thoại di động đã trở thành một thách thức đầy hứa hẹn và thú vị. Để khám phá và nắm bắt sự phát triển này, đồ án "Xây dựng ứng dụng xem video cho điện thoại di động" đã được nhóm em triển khai và xây dựng.

Trong đồ án này, chúng em đã tập trung vào việc nghiên cứu và phát triển một ứng dụng hiệu quả, dễ sử dụng và đáp ứng đầy đủ nhu cầu của người dùng khi họ mong muốn thưởng thức nội dung video trên các thiết bị di động của mình. Điều quan trọng là tạo ra một trải nghiệm người dùng tốt, đồng thời tối ưu hóa hiệu suất và khả năng tương thích với các nền tảng khác nhau.

Cuốn đồ án này không chỉ tập trung vào khía cạnh kỹ thuật của việc phát triển ứng dụng, mà còn sâu sắc vào việc hiểu và đáp ứng các mong muốn, thói quen xem video của người dùng trong môi trường di động. Chúng em đã áp dụng những kiến thức và kỹ năng thu thập được trong quá trình học tập để thiết kế một giao diện hấp dẫn, dễ dàng điều hướng và phản ánh sự phức tạp của việc xem video trên điện thoại di động.

Cuốn đồ án này không chỉ là kết quả của sự học tập và nỗ lực của các thành viên trong nhóm, mà còn là sản phẩm của sự hỗ trợ và chỉ dẫn từ thầy cô giáo và bạn bè. Chúng em hy vọng rằng cuốn đồ án này sẽ đem lại cái nhìn toàn diện về quá trình xây dựng ứng dụng xem video cho điện thoại di động và góp phần nhỏ bé vào sự phát triển không ngừng của lĩnh vực công nghệ thông tin và trải nghiệm người dùng.

**NHẬN XÉT**

(Của giảng viên môn học)

…………., ngày……tháng……năm……

NGƯỜI NHẬN XÉT

*(ký tên)*

**CÔNG VIỆC THỰC HIỆN**

|  |  |
| --- | --- |
| Thành viên | Nội dung thực hiện |
| Nguyễn Văn Hiền  0850080018 | Tìm hiểu Firebase, TextFromField, ListTitle, SizeBox,  Compressing video, Await, async, showCupertinoDialog  Authentication with Email & Password  Đăng tải video với tiêu đề video, tên bài hát  Gửi lời mời kết bạn, thu hồi lời mời kết bạn, đồng ý kết bạn, từ chối kết bạn, hủy kết bạn |
| Phan Thanh Hải  0850080017 | Tìm hiểu Firebase, Image, Row, CircleAvatar, Await, async  Tương tác video  Tìm kiếm người dùng  Hiển thị lượt thích |
| Trần Bá Minh  0850080083 | Tìm hiểu FirseBase, ListView, BottomNavigation, Navigation, Video, Await, async  Tạo và hiển thị thumbnail cho video  Displaying Videos with Caption  Bình luận video |
| Lê Nguyễn Anh Dũng  0850080011 | Tìm hiểu Firsebase, Container, Column, Divider  Theo dõi người dùng  Hiển thị lược theo dõi, đang theo dõi |

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc142341050)

[1.1. Lý do chọn đề tài 1](#_Toc142341051)

[1.2. Mục tiêu của đề tài: 1](#_Toc142341052)

[1.3. Phạm vi ứng dụng: 2](#_Toc142341053)

[1.4. Đối tượng: 2](#_Toc142341054)

[1.5. Kỹ thuật công nghệ: 2](#_Toc142341055)

[1.6. Tính ứng dụng thực tiễn: 2](#_Toc142341056)

[1.7. Kết luận: 3](#_Toc142341057)

[CHƯƠNG 2: CƠ SỞ LÝ LUẬN 4](#_Toc142341058)

[2.1. Lý thuyết 4](#_Toc142341060)

[2.1.1. Lập trình đa nền tảng là gì? 4](#_Toc142341061)

[2.1.2. Ưu điểm của lập trình đa nền tảng: 4](#_Toc142341062)

[2.1.3. Nhược điểm của lập trình đa nền tảng: 5](#_Toc142341063)

[2.2. Kỹ thuật: 6](#_Toc142341064)

[2.2.1. Ngôn ngữ lập trình Dart 7](#_Toc142341065)

[2.2.2. Flutter 9](#_Toc142341066)

[2.2.3. Android Studio 10](#_Toc142341067)

[2.2.4. Firebase 11](#_Toc142341068)

[2.2.5. Giả lập điện thoại 12](#_Toc142341069)

[CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM 13](#_Toc142341070)

[3.1. Sơ đồ hoạt động. 13](#_Toc142341071)

[3.2. Cách thức kết nối với cơ sở dữ liệu. 17](#_Toc142341072)

[3.3. Cài đặt thực nghiệm. 19](#_Toc142341073)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc142341075)

[4.1. Kết luận 27](#_Toc142341076)

[4.2. Hướng phát triển 28](#_Toc142341077)

[TÀI LIỆU THAM KHẢO 29](#_Toc142341078)

**DANH MỤC HÌNH ẢNH**

Hình 1. Sơ đồ đăng nhập

Hình 2. Sơ đồ đăng ký tài khoản

Hình 3. Sơ đồ tương tác người dùng

Hình 4. Sơ đồ đăng video

Hình 5. Sơ đồ luồng đi của CSDL

Hình 6. Đăng nhập tài khoản

Hình 7. Đăng ký tài khoản

Hình 8. Giao diện Trang chủ

Hình 9. Giao diện Hồ sơ người dùng

Hình 10. Giao diện thêm video

Hình 11. Giao diện Tìm kiếm

Hình 12. Giao diện bình luận

Hình 13. Giao diện theo dõi người dùng

Hình 14. Giao diện chat giữa các người dùng

# **CHƯƠNG 1: TỔNG QUAN**

* 1. **Lý do chọn đề tài**

Hiện nay, với sự đô thị hóa nhanh chóng của xã hội, các ứng dụng để giải trí ngày một nhiều hơn. Việc xem một đoạn video ngắn để giải trí, nắm bắt tình hình, tiếp thu kiến thức, học tập điều hay lẽ phải, cảnh giác trước những chiêu trò lừa đảo,… là một điều vô cùng được mọi người quan tâm và chú trọng. Nắm bắt được tình hình đó, nhóm em quyết định xây dựng một ứng dụng có thể đáp ứng được các nhu cầu kể trên.Vì vậy, dưới sự hướng dẫn của thầy Nguyễn Thanh Truyền, nhóm em quyết định thực hiện đồ án: *“Xây dựng ứng dụng xem video trên điện thoại di động”*.

* 1. **Mục tiêu của đề tài:**
* Phát triển ứng dụng xem video đa nền tảng: Xây dựng một ứng dụng có khả năng hoạt động trên cả hệ điều hành Android và iOS, giúp đáp ứng nhu cầu người dùng sử dụng các thiết bị di động khác nhau.
* Tối ưu trải nghiệm người dùng: Cải thiện trải nghiệm người dùng khi xem video bằng giao diện thân thiện, mượt mà và tùy chỉnh theo sở thích của người dùng.
* Hỗ trợ chất lượng video: Cung cấp tùy chọn xem video với chất lượng khác nhau, từ thấp đến cao, để người dùng có thể chọn lựa phù hợp với tốc độ internet và dung lượng thiết bị.
* Tích hợp tính năng tìm kiếm: Cung cấp tính năng tìm kiếm video nhanh chóng và chính xác, giúp người dùng dễ dàng tìm thấy nội dung mong muốn.
* Quản lý dữ liệu video: Sử dụng Firebase để lưu trữ và quản lý dữ liệu liên quan đến video, bao gồm thông tin về video, người đăng, lượt xem, bình luận, v.v.
* Tích hợp chức năng chia sẻ và bình luận: Cho phép người dùng chia sẻ video với bạn bè và gia đình cũng như thể hiện ý kiến, ý kiến phản hồi thông qua tính năng bình luận.
* Đảm bảo tính bảo mật và ổn định: Xây dựng ứng dụng với các biện pháp bảo mật và kiểm tra kỹ lưỡng để đảm bảo dữ liệu và thông tin người dùng được bảo vệ an toàn.
* Thống kê và phân tích: Cung cấp chức năng thống kê và phân tích dữ liệu, giúp nhà phát triển hiểu rõ hơn về hành vi người dùng và cải thiện trải nghiệm ứng dụng.
  1. **Phạm vi ứng dụng:**
* Phạm vi của đồ án này gói gọn ở các chức năng như đăng nhập, đăng xuất, đăng video, bình luận, thả tương tác, quản lý hồ sơ cá nhân, theo dõi bạn bè, xem video.
* Hướng tới những đối tượng người dùng là những người dùng điện thoại muốn giải trí, muốn học hỏi, tương tác với nhau.
* Các thao tác nghiệp vụ cơ bản của một ứng dụng xem video.
  1. **Đối tượng:**
* Những người quan tâm đến việc giải trí, học hỏi, theo dõi tin tức thông qua video.
  1. **Kỹ thuật công nghệ:**
* Flutter.
* Firebase.
* Dart.
* Android Studio.
  1. **Tính ứng dụng thực tiễn:**

Tiềm năng thương mại: Ứng dụng xem video trên điện thoại di động có tiềm năng phát triển thương mại rất lớn. Có thể tích hợp quảng cáo và các nền tảng kiếm tiền khác nhau để tạo nguồn thu nhập từ ứng dụng. Điều này thu hút sự quan tâm của các doanh nghiệp và nhà quảng cáo.

Trải nghiệm người dùng: Với một giao diện người dùng hấp dẫn và dễ sử dụng, ứng dụng xem video sẽ cung cấp trải nghiệm tốt cho người dùng. Khả năng tùy chỉnh giao diện và chất lượng video sẽ thu hút người dùng và giữ chân họ.

Nền tảng phát triển đa nền tảng: Sử dụng Flutter, ứng dụng có thể được phát triển đồng thời cho nhiều nền tảng như Android, iOS, web và máy tính để bàn. Điều này giúp tiết kiệm thời gian và chi phí phát triển và duy trì ứng dụng.

Kết nối cộng đồng người dùng: Ứng dụng xem video thu hút sự tương tác và chia sẻ nội dung giữa người dùng. Nó tạo ra cộng đồng xem video và góp phần kết nối mọi người với nhau qua nội dung chia sẻ.

Thông tin và giáo dục: Ngoài giải trí, ứng dụng còn có thể chứa các video hướng dẫn, giảng dạy, hoặc thông tin giáo dục hữu ích. Điều này giúp người dùng học hỏi và nâng cao kiến thức..

* 1. **Kết luận:**

Đồ án sau khi hoàn thành có thể đáp ứng đầy đủ các nhu cầu xem để giải trí và học tập, kết nối cộng đồng, chạy được trên các thiết bị di động ở thị trường Việt Nam.

# **CHƯƠNG 2: CƠ SỞ LÝ LUẬN**

1. 1. **Lý thuyết**
      1. **Lập trình đa nền tảng là gì?**

Trong thời đại công nghệ số hiện nay, khi bạn phục vụ cho các đối tượng sử dụng nhiều nền tảng khác nhau đến từ iOS, Android hay Windows Phone, bạn cần phải tạo ra các ứng dụng đa nền tảng (cross-platform hay multi-platform).

Đa nền tảng là một thuật ngữ để chỉ các phần mềm hay phương thức điện toán được vận hành cùng nhau trên nhiều nền tảng. Như vậy, một phần mềm được gọi là đa nền tảng khi và chỉ khi nó có khả năng hoạt động trên nhiều hơn một hệ điều hành hay kiến trúc máy tính. Hiện nay có khá nhiều loại phần mềm đa nền tảng khác nhau được tạo ra nhưng nhìn chung chúng ta có thể phân chúng thành 2 loại chính là:

Loại phần mềm có thể trực tiếp chạy trên bất cứ nền tảng nào mà không cần đến các bước biên dịch/ thông dịch. Để làm được điều này, các lập trình viên cần viết phần mềm bằng một loại ngôn ngữ thông dịch hoặc đã dịch phần mềm trước sang mã bytecode…

Phần mềm đa nền tảng còn lại đơn giản, dễ thiết kế hơn. Tuy nhiên nó yêu cầu có thêm bước thiết kế hoặc biên dịch từng phiên bản cho mỗi nền tảng mà nó hỗ trợ.

Mỗi loại phần mềm đều có ưu nhược điểm riêng phụ thuộc vào mục đích của người dùng. Tuy nhiên, tất cả các phần mềm đa nền tảng trên thị trường đều có một điểm chung là nó có thể chạy trên các nền tảng khác nhau nhưng không thể tự động làm việc trên tất cả kiến trúc mà hệ điều hành đó hỗ trợ.

* + 1. **Ưu điểm của lập trình đa nền tảng:**

Xu hướng lập trình đa nền tảng đã lan rộng ra toàn thế giới bởi nó mang đến nhiều tiện ích nổi bật, một số có thể kể đến như:

Dễ dàng tiếp cận khách hàng: Lợi ích đầu tiên mà ai cũng có thể nhận thấy đó chính là lập trình mobile đa nền tảng giúp tiếp cận khách hàng dễ dàng và nhanh chóng mở rộng thị trường. Ứng dụng của bạn có thể hoạt động trên nhiều nền tảng khác nhau đồng nghĩa với việc độ phủ sóng đang ngày càng mở rộng. Và như thế, không khó để tiếp cận với đối tượng khách hàng mục tiêu. Bên cạnh đó, khi sở hữu các ứng dụng đa nền tảng, bạn sẽ thuận tiện trong việc marketing thương hiệu/ sản phẩm trên các phương tiện truyền thông khác nhau.

Tiết kiệm thời gian và chi phí: Thay vì phải tốn nhiều nhân lực và thời gian trong việc tạo ra nhiều ứng dụng cho các hệ điều hành khác nhau thì giờ đây bạn chỉ cần viết code một lần – quản lý tập trung cho ứng dụng của mình trên tất cả các nền tảng. Không những thế, việc tự do tiếp thị trên các phương tiện truyền thông cũng giúp bạn giảm thiểu chi phí quảng cáo.

Sử dụng công nghệ lập trình cao: Trong khi lập trình đa nền tảng, bạn có thể sử dụng các công nghệ lập trình tiên tiến để nâng cao trải nghiệm người dùng cũng như tăng độ mượt mà cho ứng dụng. Bên cạnh đó thì bạn cũng hoàn toàn có thể sử dụng các công nghệ cũ, quen thuộc để viết phần mềm đa nền tảng. Ví dụ như với Appcelerator – một công cụ đã quá quen thuộc với giới lập trình, bạn có thể viết code trong HTML5 và chuyển đổi sang các nền tảng khác nhau một cách dễ dàng. Đây chính là lợi thế về sự linh hoạt trong lập trình. Với lợi thế này, bạn vừa có thể tối ưu nguồn lực sẵn có và cũng có thể tận dụng chúng cho việc khai thác nâng cao để tạo ra các phần mềm cao cấp hơn!

Tạo ra và duy trì sự đồng bộ hoá: Và một lợi thế khác của lập trình đa nền tảng chính là khả năng đồng bộ hoá “cực đỉnh”. Thay vì phải cập nhật dữ liệu và đồng bộ thủ công giữa ứng dụng của các nền tảng khác nhau thì giờ đây bạn có thể ngay lập tức đồng bộ hoá dữ liệu trên tất cả các nền tảng. Thật tuyệt vời phải không nào? Việc này sẽ giúp tạo ra một mạng lưới thông tin đồng nhất, thuận tiện cho lập trình viên trong quá trình sửa chữa/ nâng cấp/ quản lý phần mềm và cũng mang lại cho người dùng trải nghiệm hoàn hảo hơn.

* + 1. **Nhược điểm của lập trình đa nền tảng:**

Có ưu điểm thì tất nhiên phải có nhược điểm! Bạn không thể làm hài lòng tất cả mọi người và các phần mềm đa nền tảng cũng vậy. Chúng được “sinh ra” với mục đích tương thích với một lượng lớn đối tượng người dùng. Bởi thế việc phát sinh một số hạn chế là không thể tránh khỏi.

Hạn chế về sự linh hoạt: Trước hết, một ứng dụng đa nền tảng chắc chắn sẽ bị hạn chế về sự linh hoạt hơn so với các ứng dụng phục vụ cho một nền tảng nhất định. Mỗi nền tảng sẽ có đặc điểm, yêu cầu, phương thức hoạt động, cách hiển thị và nhu cầu người dùng khác nhau. Những yếu tố này sẽ đặt ứng dụng đa nền tảng vào một phạm vi bất lợi nhất định. Trong khi đó, yêu cầu của người dùng ngày càng khắt khe hơn, bởi thế bạn phải cân nhắc lựa chọn kỹ lưỡng giữa việc lập trình ứng dụng đa nền tảng hay thiết kế nhiều ứng dụng cho các nền tảng khác nhau.

Việc đảm bảo chất lượng gặp khó khăn: Việc thiết kế ra một ứng dụng hoạt động tốt được trên nhiều nền tảng đòi hỏi khả năng và công nghệ lập trình cao. Bởi thế bạn sẽ gặp nhiều khó khăn trong việc đảm bảo chất lượng cho phần mềm. Có rất nhiều yếu tố cần cân nhắc trong quá trình lập trình là:

* Giao diện người dùng khác nhau;
* Khác biệt về công cụ và ngôn ngữ trong mỗi nền tảng API;
* Tích hợp ứng dụng với thiết lập cục bộ giữa các nền tảng;
* Tùy chọn lưu trữ;
* Xử lý yêu cầu từ các bên liên quan…

Tất nhiên bạn có thể sử dụng các công cụ hỗ trợ như Unity, Ramp, PhoneGap, Grapple, Open Plug, Rhomobile, Titanium… để lập trình thuận tiện hơn. Tuy nhiên hãy cân nhắc kỹ mục tiêu và tính năng của ứng dụng trước khi quyết định có sử dụng phương pháp lập trình đa nền tảng hay không.

* 1. **Kỹ thuật:**

Về kỹ thuật để tạo demo cho các chức năng nói trên thì dưới đây là một số các kỹ thuật và công nghệ được áp dụng.

* Ngôn ngữ lập trình: Sử dụng các ngôn ngữ Dart.
* Phần mềm: Sử dụng các phần mềm như Android Studio.
* Framework: Flutter.
* Mô hình: MVC.
* Ngoài ra còn sử dụng nhiều công cụ như Device Manager, GitHub,..
  + 1. **Ngôn ngữ lập trình Dart**



Dart là một ngôn ngữ lập trình phổ biến và đa nền tảng được phát triển bởi Google. Nó được thiết kế để hỗ trợ xây dựng các ứng dụng đa nền tảng, chẳng hạn như ứng dụng web, di động và máy tính cá nhân. Dart được sử dụng chủ yếu cho việc phát triển ứng dụng Flutter - một framework lập trình đa nền tảng rất mạnh mẽ.

* **Lịch sử hình thành và phát triển của Dart có các giai đoạn chính như sau:**

1. Ra đời: Dart được công bố công khai lần đầu tiên tại hội nghị GOTO ở Aarhus, Đan Mạch vào tháng 10 năm 2011. Dart ban đầu được xem như một ngôn ngữ thay thế cho JavaScript trong việc phát triển các ứng dụng web phức tạp.
2. Dartium và Dart SDK: Để hỗ trợ việc phát triển và kiểm tra ứng dụng Dart, Google phát triển một trình duyệt đặc biệt có tên Dartium. Dart SDK cũng được phát hành cùng với Dartium, cung cấp các công cụ cần thiết để viết, biên dịch và chạy các ứng dụng Dart trên nhiều nền tảng.
3. Dart 1.x: Phiên bản Dart 1.x chủ yếu tập trung vào việc phát triển ứng dụng web. Dart được biên dịch thành JavaScript để chạy trên các trình duyệt không hỗ trợ trực tiếp Dart.
4. Dart 2.0 và Flutter: Dart 2.0 là một bước đột phá quan trọng, đi kèm với việc giới thiệu Dart Native Compiler (dart2native) để biên dịch Dart thành mã máy nhanh hơn. Ngoài ra, Flutter - một framework phát triển ứng dụng di động đa nền tảng - cũng được giới thiệu cùng với Dart 2.0. Flutter sử dụng Dart làm ngôn ngữ lập trình chính để xây dựng các ứng dụng di động đẹp và nhanh chóng trên iOS và Android.

* **Cấu trúc của Dart bao gồm các thành phần sau:**

1. Cú pháp: Dart có cú pháp tương tự như nhiều ngôn ngữ lập trình khác như C++, Java hoặc JavaScript. Nó sử dụng dấu chấm phẩy để kết thúc câu lệnh và ngoặc nhọn để định nghĩa phạm vi của khối mã.
2. Kiểu dữ liệu: Dart là một ngôn ngữ kiểu tĩnh, có nghĩa là các biến phải được khai báo với một kiểu cụ thể. Các kiểu dữ liệu thông thường bao gồm số nguyên, số thực, chuỗi, boolean và danh sách.
3. Lớp và đối tượng: Dart hỗ trợ lập trình hướng đối tượng, với khả năng định nghĩa lớp, tạo đối tượng và thực hiện kế thừa. Nó cũng hỗ trợ các tính chất của lập trình hàm như lambda và closures.
4. Bộ điều khiển luồng: Dart cung cấp các cấu trúc điều khiển luồng như if-else, vòng lặp for và while, và câu lệnh điều khiển nhảy như break và continue.
5. Gói và import: Để sắp xếp mã của mình, Dart sử dụng gói và import. Một gói là một thư mục chứa các tệp mã, trong khi import được sử dụng để nhập mã từ các gói khác.

Tổng quan, Dart đã trải qua một quá trình phát triển liên tục từ khi được giới thiệu lần đầu vào năm 2011. Với việc tích hợp mạnh mẽ với Flutter, Dart đã trở thành một công cụ quan trọng trong phát triển ứng dụng di động đa nền tảng.

* + 1. **Flutter**



Flutter là một framework mã nguồn mở được phát triển bởi Google, cho phép lập trình viên xây dựng ứng dụng di động đẹp và chạy mượt trên nhiều nền tảng, bao gồm cả Android và iOS. Flutter được viết bằng ngôn ngữ Dart và hỗ trợ phát triển ứng dụng di động nhanh chóng, dễ dàng và linh hoạt.

* **Lịch sử phát triển của Flutter:**

1. Project Sky: Flutter ban đầu được gọi là "Project Sky" và được bắt đầu bởi công ty nguồn mở Alpha của Google vào năm 2010. Ban đầu, nó là một dự án nghiên cứu với mục tiêu tạo ra một framework UI linh hoạt và tốc độ cao, nhằm giúp phát triển các ứng dụng di động đẹp và nhanh chóng.
2. Flutter: Trong năm 2015, Google công bố Flutter chính thức tại sự kiện Dart Developer Summit. Nó là một framework mã nguồn mở cho việc xây dựng các ứng dụng di động, web và máy tính để bàn từ một mã nguồn duy nhất.
3. Stabilization: Sau khi công bố, Flutter tiếp tục phát triển và trải qua quá trình ổn định hóa và cải tiến. Google đánh giá cao Flutter như là một công cụ phát triển tiện ích và mạnh mẽ, và nó ngày càng được sử dụng rộng rãi trong cộng đồng lập trình di động.
   * 1. **Android Studio**



Android Studio là môi trường phát triển tích hợp (IDE) chính thức của Google dành cho việc phát triển ứng dụng di động Android. Nó cung cấp môi trường toàn diện để phát triển, kiểm thử và triển khai ứng dụng Android dễ dàng và hiệu quả.

* **Lịch sử phát triển của Android Studio:**

1. Eclipse: Trước khi Android Studio ra đời, lập trình viên Android thường sử dụng Eclipse IDE để phát triển ứng dụng di động Android. Tuy nhiên, Eclipse không được tối ưu hoá tốt cho việc phát triển Android và có một số vấn đề về hiệu suất và tích hợp.
2. Android Studio 1.0: Vào năm 2013, Google công bố Android Studio, một IDE mới dành riêng cho phát triển ứng dụng Android. Android Studio được xây dựng trên nền tảng IntelliJ IDEA của JetBrains và được tối ưu hoá cho việc phát triển ứng dụng Android.
3. Gradle: Android Studio sử dụng Gradle làm công cụ quản lý build, giúp đơn giản hóa việc tạo APK và quản lý các phụ thuộc của dự án.
4. Phiên bản mới và cải tiến: Kể từ khi ra mắt, Android Studio đã trải qua nhiều phiên bản và cải tiến liên tục. Google luôn cập nhật và cải tiến các tính năng và công cụ trong Android Studio để hỗ trợ tốt hơn cho các nhà phát triển Android.
   * 1. **Firebase**



Firebase là một nền tảng phát triển ứng dụng di động và web do Google cung cấp. Nó cung cấp các dịch vụ đám mây mạnh mẽ và toàn diện để giúp lập trình viên xây dựng ứng dụng di động và web một cách dễ dàng, nhanh chóng và hiệu quả. Firebase bao gồm một loạt các dịch vụ đa dạng, từ lưu trữ dữ liệu và xử lý người dùng đến phân tích và gửi thông báo đến người dùng.

* **Lịch sử hình thành và phát triển của Firebase như sau:**

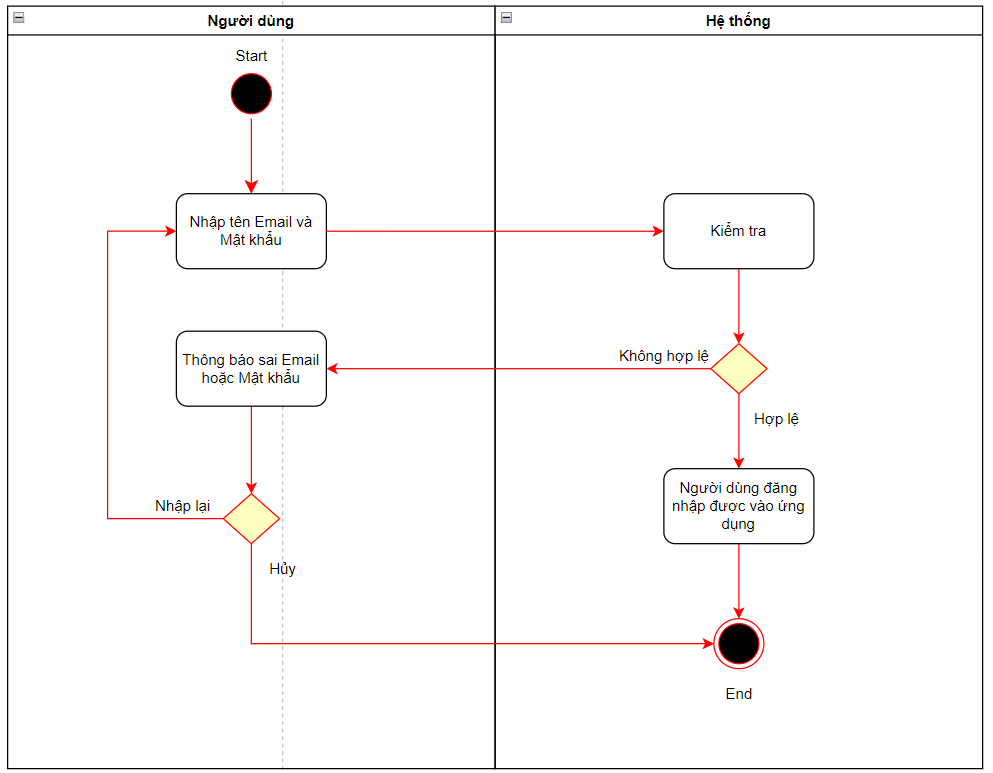
1. 2011: Firebase được thành lập bởi James Tamplin và Andrew Lee tại San Francisco. Ban đầu, Firebase là một startup tạo ra một nền tảng thời gian thực cho việc xây dựng ứng dụng di động và web.
2. 2012: Firebase chính thức ra mắt sản phẩm đầu tiên là Firebase Realtime Database. Đây là một cơ sở dữ liệu thời gian thực có khả năng đồng bộ dữ liệu giữa các thiết bị và máy chủ. Firebase Realtime Database cho phép các ứng dụng di động và web cập nhật dữ liệu một cách tức thì và tự động đồng bộ hóa dữ liệu giữa các thiết bị.
3. 2014: Firebase tiếp tục mở rộng với việc ra mắt các tính năng mới, bao gồm Firebase Authentication (xác thực người dùng), Firebase Hosting (lưu trữ và phân phối ứng dụng web), Firebase Cloud Messaging (gửi thông báo đến các thiết bị di động) và Firebase Remote Config (cấu hình ứng dụng từ xa).
4. 2016: Firebase chính thức được mua lại bởi Google và trở thành một phần của Google Cloud Platform (GCP). Sự kết hợp với GCP giúp Firebase mở rộng phạm vi và khả năng của nó, đồng thời cung cấp sự hỗ trợ mạnh mẽ từ Google.
5. 2017: Firebase mở rộng thêm với việc ra mắt các tính năng mới như Firebase Cloud Firestore (cơ sở dữ liệu tài liệu mô tả được cấu trúc), Firebase ML Kit (máy học cho việc xử lý hình ảnh và văn bản), và nhiều công cụ hỗ trợ phát triển ứng dụng khác.
6. Hiện tại: Firebase tiếp tục phát triển và trở thành một trong những nền tảng phát triển ứng dụng hàng đầu, được sử dụng rộng rãi bởi các nhà phát triển trên toàn thế giới. Firebase không chỉ cung cấp các tính năng và công cụ mạnh mẽ cho việc xây dựng ứng dụng di động và web, mà còn cung cấp sự linh hoạt và tiện ích cho việc quản lý và phát triển ứng dụng trên nền tảng đám mây.
   * 1. **Giả lập điện thoại**

Giả lập điện thoại (hay còn gọi là Emulator) là một phần mềm giả lập các thiết bị di động hoặc máy tính bảng trên máy tính cá nhân. Chức năng chính của giả lập điện thoại là cho phép lập trình viên thử nghiệm và chạy ứng dụng di động trên môi trường phát triển dễ dàng và thuận tiện, mà không cần phải sở hữu thực tế các thiết bị di động.

Giả lập điện thoại cung cấp môi trường ảo giống như thiết bị di động thật, cho phép lập trình viên kiểm tra và thử nghiệm ứng dụng với các tính năng và giao diện tương tự như trên các thiết bị thực. Nó hỗ trợ nhiều phiên bản hệ điều hành di động như Android và iOS, và cho phép bạn chạy ứng dụng với nhiều cấu hình và kích thước màn hình khác nhau.

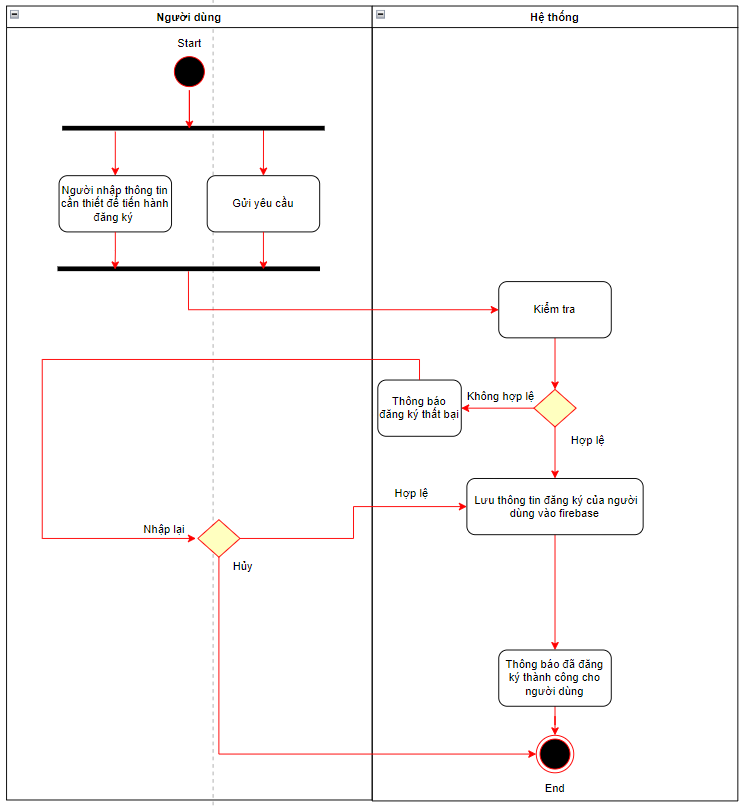
# **CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM**

* 1. **Sơ đồ hoạt động.**
* **Sơ đồ đăng nhập**

****

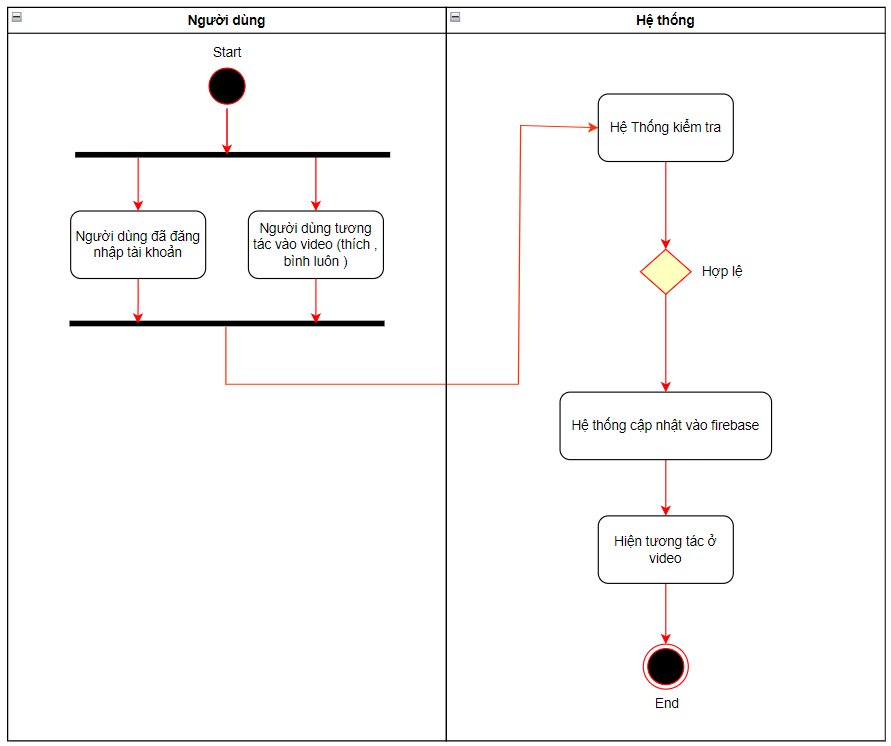
*Hình 1. Sơ đồ đăng nhập*

* **Sơ đồ đăng ký tài khoản**

****

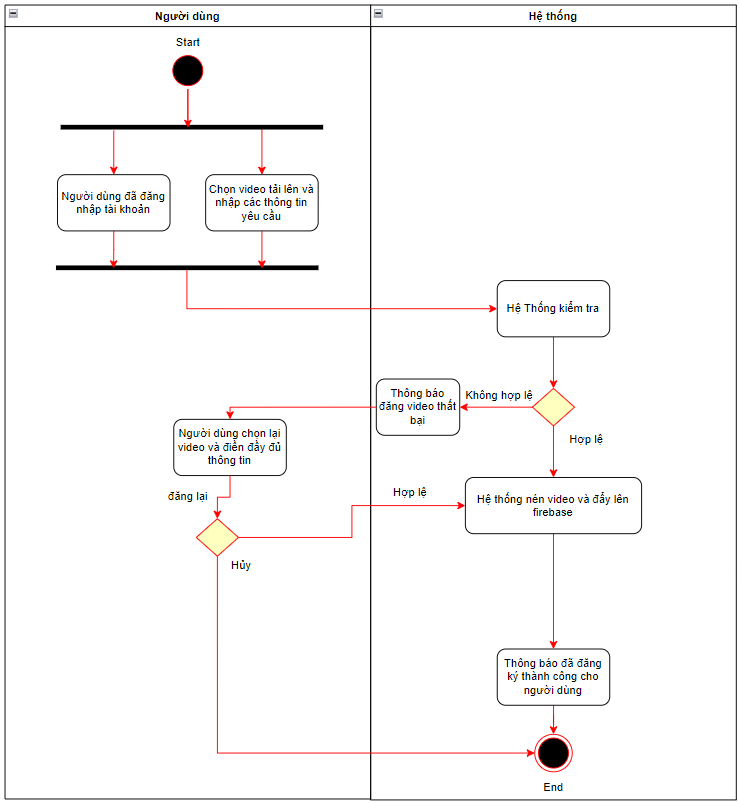
*Hình 2. Sơ đồ đăng ký tài khoản*

* **Sơ đồ tương tác người dùng**

****

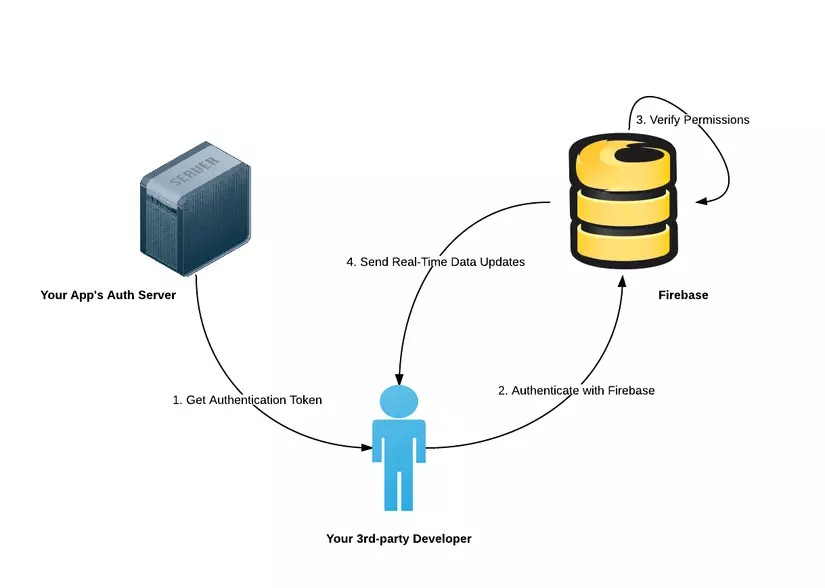
*Hình 3. Sơ đồ tương tác người dùng*

* **Sơ đồ đăng video**

****

*Hình 4. Sơ đồ đăng video*

* 1. **Cách thức kết nối với cơ sở dữ liệu.**

****

*Hình 5. Sơ đồ luồng đi của CSDL*

Your 3rd-party Developer: Đây là bên thứ ba (một người hoặc tổ chức) muốn tích hợp tính năng liên quan đến xác thực và quản lý người dùng từ Firebase vào ứng dụng của họ.

Get Authentication Token: Bên phía 3rd-party Developer sẽ yêu cầu Firebase cung cấp mã thông báo xác thực (authentication token). Điều này thường xảy ra khi họ muốn cho phép người dùng đăng nhập hoặc truy cập các tài nguyên bảo mật trên ứng dụng.

Your App's Auth Server: Đây là phần của ứng dụng của bạn, nơi bạn xây dựng một máy chủ (server) để tương tác với Firebase và xác thực người dùng. Nhiệm vụ của nó là lấy mã thông báo xác thực từ Firebase và sử dụng nó để thực hiện các hoạt động liên quan đến xác thực và quản lý người dùng cho ứng dụng Videoapp.

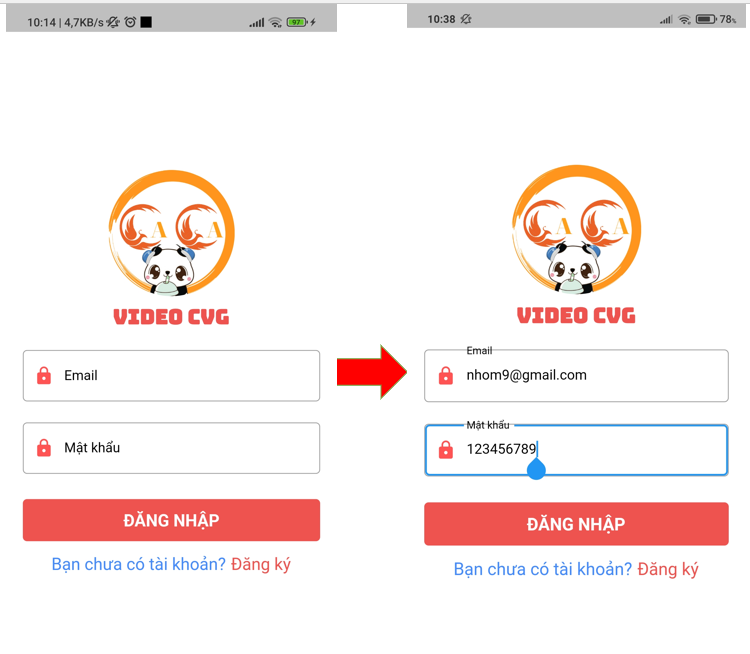
Authenticate with Firebase: Ở bước này, mã thông báo xác thực được truyền từ ứng dụng của bạn đến Firebase để xác minh tính hợp lệ của nó. Firebase sẽ kiểm tra mã thông báo và trả về kết quả xác thực cho ứng dụng Videoapp.

Firebase: Firebase là dịch vụ đám mây của Google cung cấp các tính năng như xác thực người dùng, quản lý cơ sở dữ liệu thời gian thực, quảng cáo, phân tích và nhiều tính năng khác cho ứng dụng di động và web. Trong trường hợp này, Firebase đảm nhận nhiệm vụ xác thực người dùng và cung cấp mã thông báo xác thực.

Verify Permissions: Sau khi mã thông báo xác thực được xác minh, Firebase sẽ kiểm tra quyền truy cập và các loại giới hạn khác của người dùng để đảm bảo rằng họ có quyền thực hiện các hoạt động cụ thể trong ứng dụng.

Send real-time Data Updates: Sau khi xác thực thành công và xác minh quyền truy cập, ứng dụng của bạn có thể gửi và nhận các cập nhật dữ liệu thời gian thực thông qua Firebase, đảm bảo rằng người dùng có trải nghiệm tương tác mượt mà và thú vị.

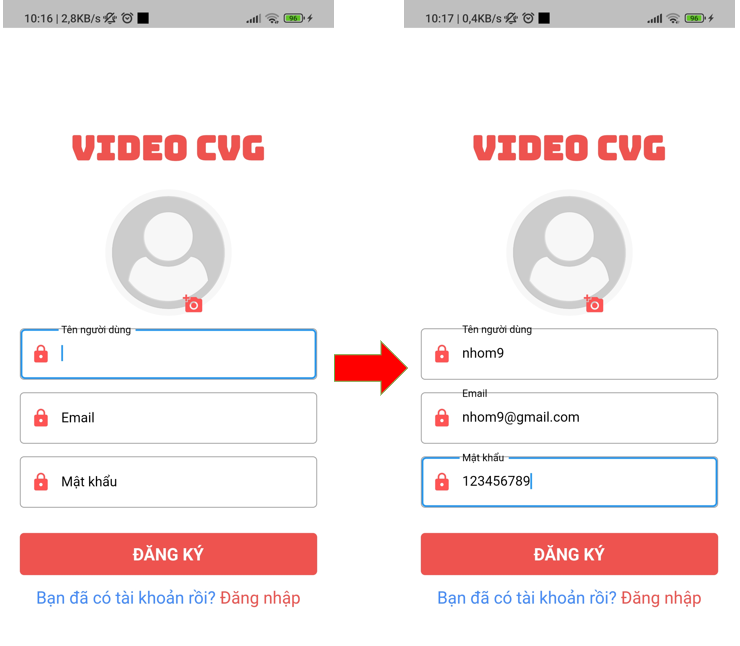
* 1. **Cài đặt thực nghiệm.**
* **Đăng nhập tài khoản**

****

*Hình 6. Đăng nhập tài khoản*

Khi vào giao diện Videoapp của nhóm 9, màn hình đăng nhập là màn hình đầu tiên được hiện ra, nếu người dùng đã có tài khoản thì sẽ tiến hành nhập Email và Mật khẩu và bấm vào nút ĐĂNG NHẬP để vào giao diện trang chủ, nếu chưa có tài khoản thì người dùng có thể bấm vào nút Đăng ký để đi đến giao diện đăng ký và tiến hành tạo một tài khoản cá nhân cho chính bản thân.

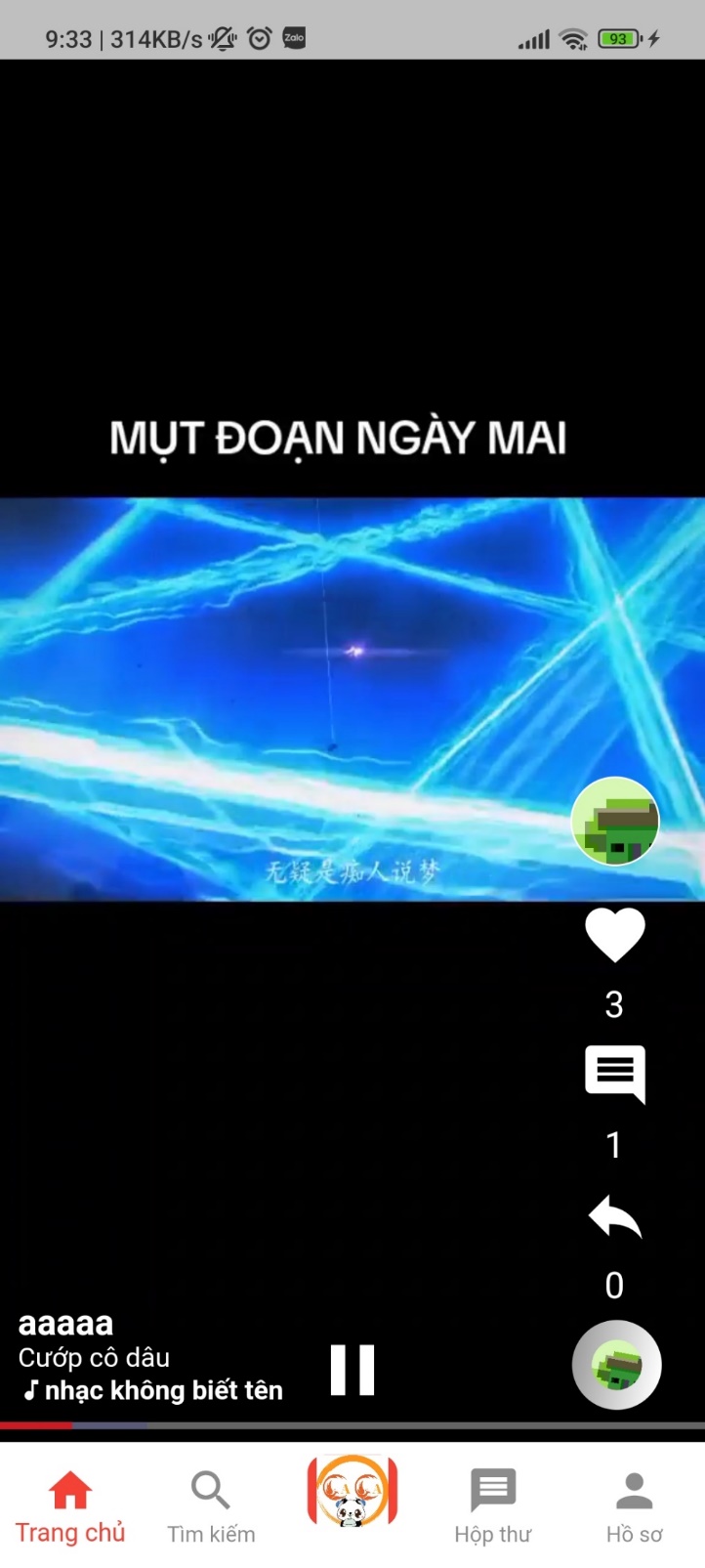
* **Đăng ký tài khoản**

****

*Hình 7. Đăng ký tài khoản*

Khi người dùng đi đến giao diện đăng ký tài khoản. Người dùng nếu chưa có tài khoản, người dùng phải điền Tên người dùng (đây là phần tên sẽ hiển thị trên mỗi video khi người dùng tiến hành đăng video), điền Email và Mật khẩu (Email và Mật khẩu này dùng để đăng nhập vào tài khoản của cá nhân) và chọn một hình ảnh để làm ảnh đại diện cho người dùng. Sau khi bấm vào nút ĐĂNG KÝ, khi đã điền đầy đủ các trường Tên người dùng, Email, Mật khẩu và chọn ảnh, người dùng sẽ được vào giao diện trang chủ và đăng nhập với tài khoản mình đã đăng ký. Khi các trường không đúng định dạng sẽ có các thông báo để yêu cầu người dùng hoàn thành. Nếu người dùng vô tình bấm nhầm qua giao diện đăng ký thì có thể quay lại giao diện đăng nhập bằng các nhấn vào chữ Đăng nhập màu đỏ dưới nút ĐĂNG KÝ.

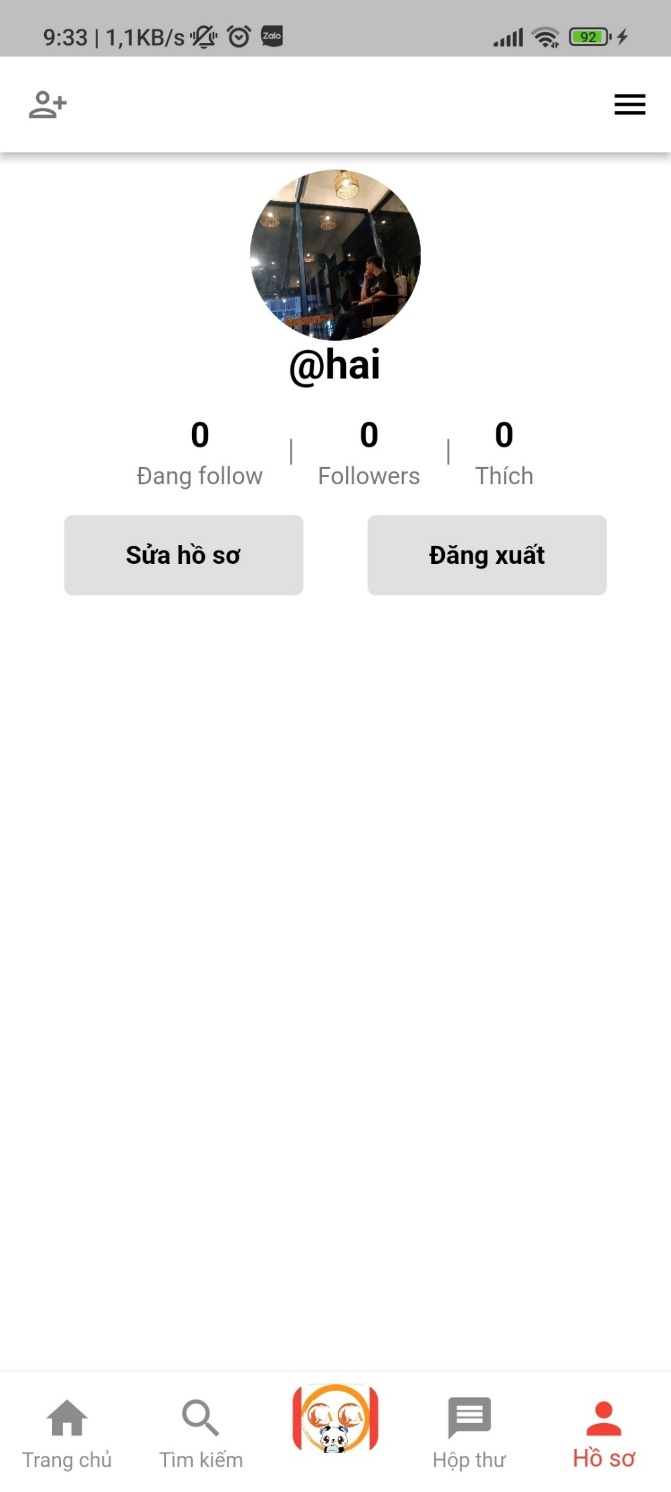
* **Giao diện Trang chủ**

****

*Hình 8. Giao diện trang chủ*

Khi đã đăng ký hoặc đăng nhập thành công sẽ ta sẽ vào giao diện Trang chủ. Trong giao diện này, người dùng có thể xem video, thả các tương tác như thả yêu thích, bình luận vào video, hoặc có thể bấm vào ảnh người dùng để tiến hành theo dõi tài khoản mà mình thích. Khi người dùng thả yêu thích hoặc bình luận thì các con số sẽ được tăng lên theo số lượt yêu thích mà người dùng thực đã tương tác. Người dùng là chủ tài khoản có thể tự thả yêu thích hoặc đăng bình luận trên video mình, lượt yêu thích của mỗi tài khoản chỉ được thả yêu thích 1 lần cho mỗi video, các bình luận thì sẽ không bị giới hạn.

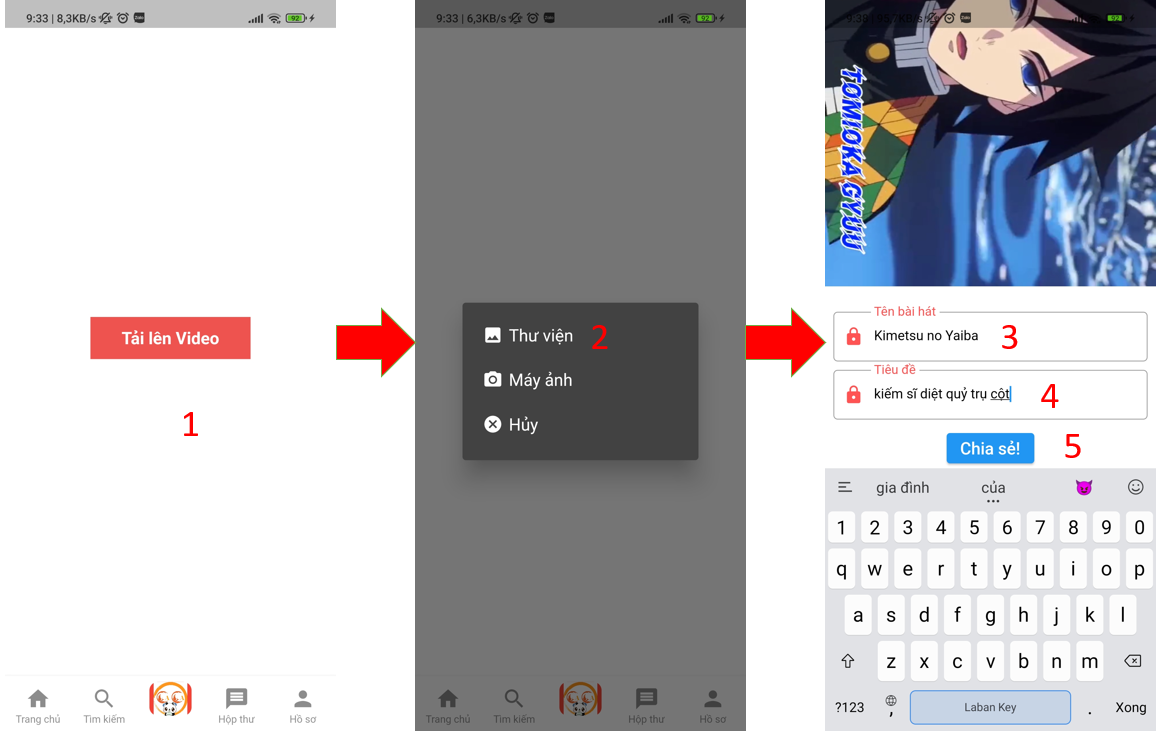
* **Giao diện Hồ sơ người dùng**



*Hình 9. Giao diện hồ sơ người dùng*

Trong giao diện Hồ sơ, sẽ hiển thị các thông tin như tên người dùng, số người mà người dùng đang theo dõi, số người theo dõi tài khoản và tổng số lượng yêu thích của các video. Người dùng có thể bấm vào nút Đăng xuất để có thể thay đổi tài khoản hoặc bấm vào nút Sửa hồ sơ để cập nhật tên người dùng hoặc ảnh đại diện.

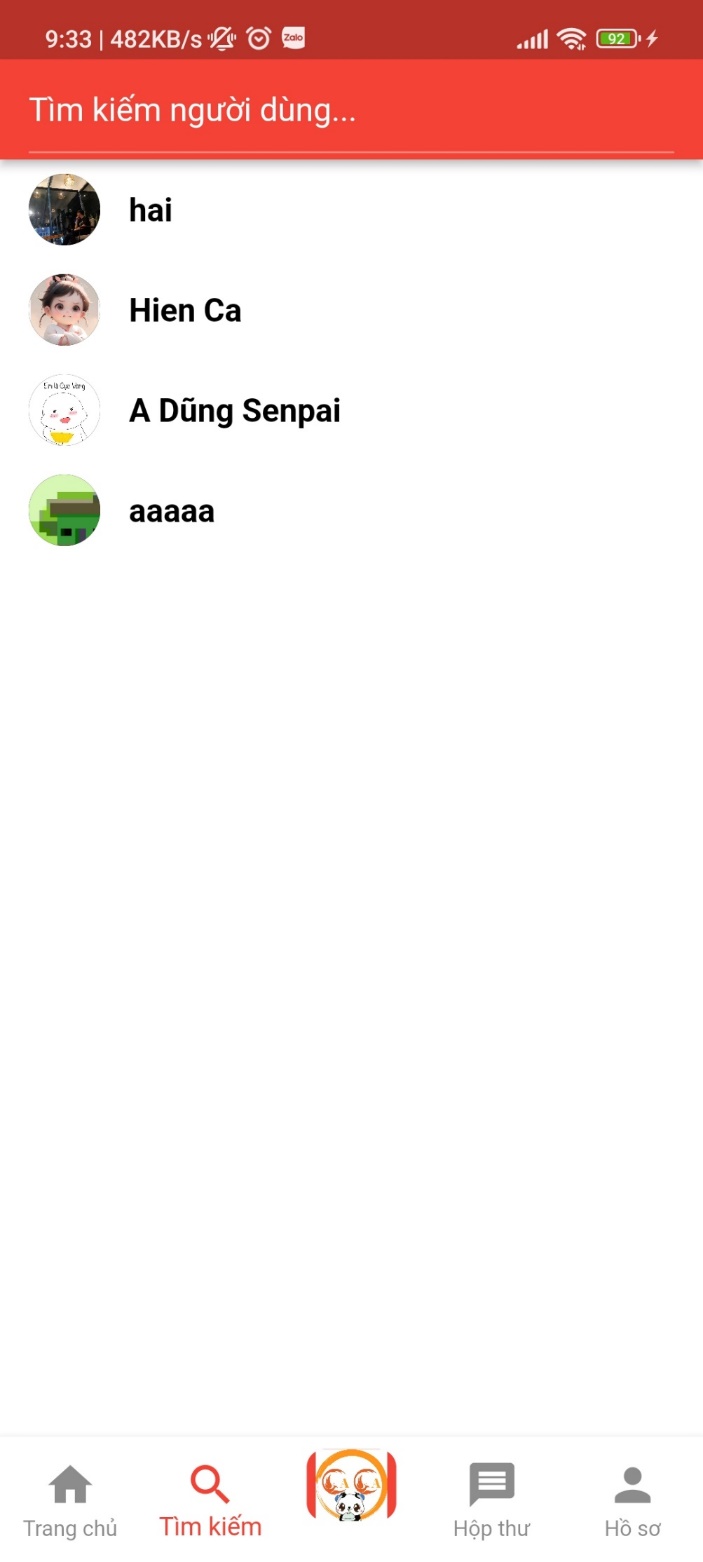
* **Giao diện thêm video**

****

*Hình 10. Giao diện thêm video*

Khi người dùng muốn đăng video thì người dùng chỉ cần bấm vào biểu tượng ở chính giữa phía dưới để tiến hành vào giao diện thêm video. Người dùng phải bấm vào nút Tải lên Video, sau đó tiến hành chọn nơi lưu trữ vào chọn video người dùng muốn đăng tải, sau đó tiến hành nhập Tên bài hát, Tiêu đề cuối cùng là nhấn vào nút Chia sẻ! để tiến hành đăng tải. Người dùng cần làm theo các bước trên để tiến hành đăng video. Sau khi thực hiện xong và bấm vào nút Chia sẻ! ở bước 5 thì người dùng sẽ đợt ít phút để video có thể nén và tiến hành đăng tải lên ứng dụng. Sau khi đăng thành công thì video sẽ được chạy trên trang chủ, người dùng có thể xem video cá nhân bằng cách nhấp vào Hồ sơ để xem.

* **Giao diện Tìm kiếm**

****

*Hình 11. Giao diện tìm kiếm*

Người dùng cần nhập Tên người dùng để tiến hành tìm kiếm, tìm kiếm sẽ được tìm kiếm dựa trên thuật toán tìm kiếm tương đối. Chỉ cần tên gần giống sẽ được hiển thị lên. Ứng dụng sẽ gợi ý một số tên tài khoản đã đăng video trong ứng dụng.

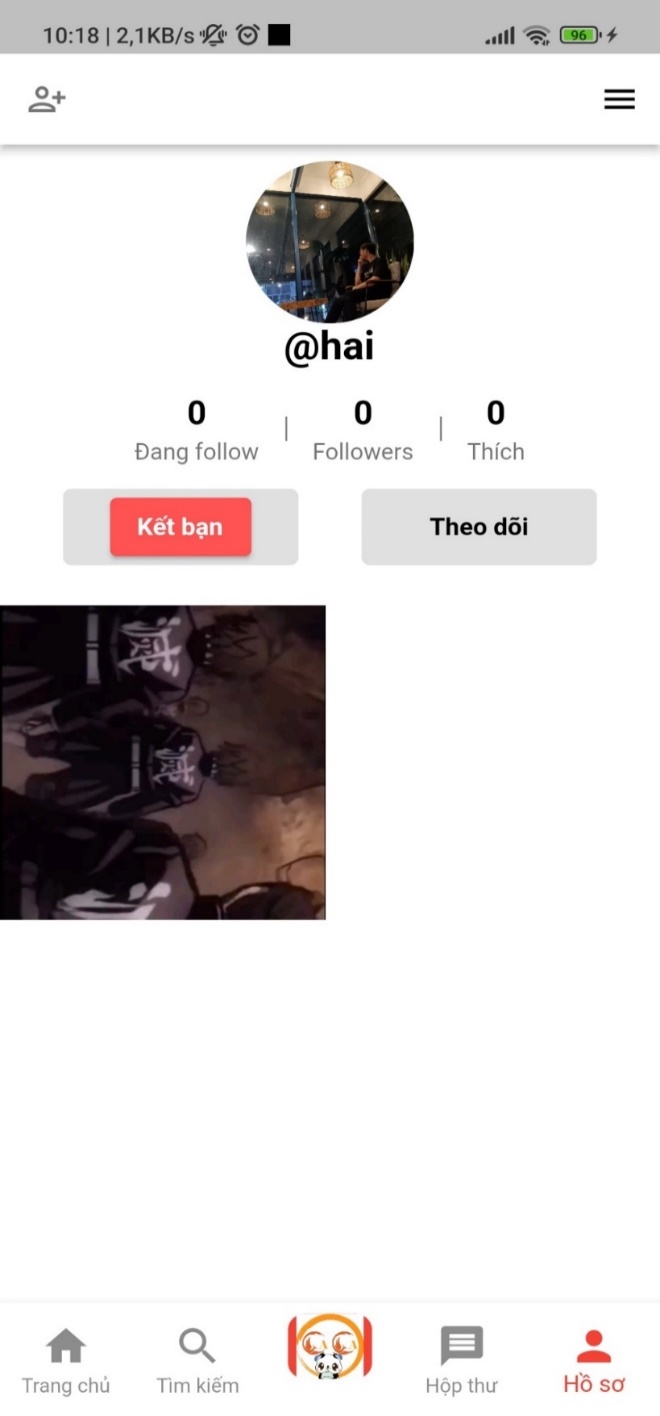
* **Giao diện bình luận**

****

*Hình 12. Giao diện bình luận*

Khi bấm vào biểu tượng Comment ở giao diện Trang chủ sẽ vào được giao diện bình luận. Mỗi video sẽ có một số bình luận khác nhau và sẽ không bị ảnh hướng đến video khác. Tên người dùng sẽ có màu đỏ, bình luận sẽ có màu đen, bên dưới bình luận sẽ hiển thị ngày đã bình luận và số lượt yêu thích của bình luận đó, người dùng có thể nhấp vào biểu tương trái tim để thả yêu thích. Số lượt bình luận của mỗi video sẽ được thay đổi dựa vào số bình luận.

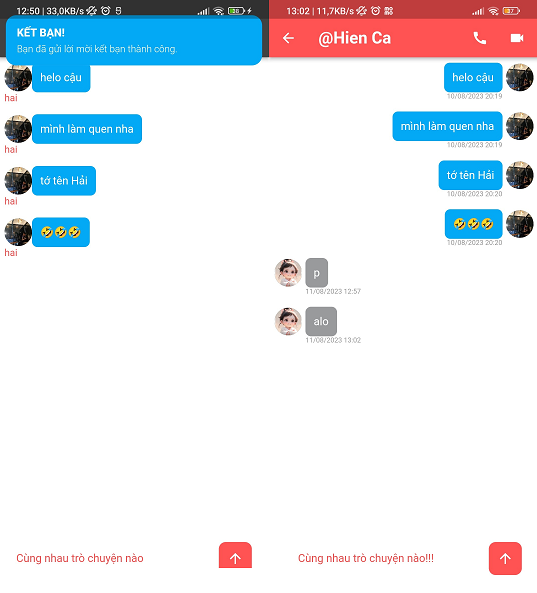
* **Giao diện theo dõi người dùng**

****

*Hình 13. Giao diện theo dõi người dùng*

Khi người dùng muốn theo dõi một tài khoản người dùng mình yêu thích chỉ cần bấm vào biểu tượng người dùng trong mỗi video ở giao diện Trang chủ hoặc vào giao diện Tìm kiếm để tìm người dùng. Sau khi bấm vào các giao diện ấy sẽ hiển thị ra trang của Hồ sơ của người mình quan tâm. Người dùng có thể kết bạn bằng cách nhấp vào nút kết bạn, bấm vào nút Theo dõi để tiến hành theo dõi người đó hoặc đơn giản hơn là vào đó xem những video họ đã đăng.

* **Giao diện chat giữa các người dùng**



Hình 14. Giao diện chat giữa các người dùng.

Ứng dụng Videoapp hỗ trợ cho người dùng tính năng chat với nhau khi người dùng có nhu cầu kết bạn để học hỏi, trò chuyện, làm quen. Đây là một tin năng có thể thu hút người dùng sử dụng Videoapp nhiều hơn.



# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. **Kết luận**

Trong ứng dụng video này, chúng ta đã khám phá một môi trường tương tác thú vị, nơi người dùng có thể tận hưởng việc xem và chia sẻ nhiều loại video khác nhau. Ứng dụng đã cung cấp một loạt các tính năng và trải nghiệm hấp dẫn để đảm bảo rằng người dùng có thể tìm thấy và tương tác với nội dung một cách dễ dàng và thú vị.

Chúng ta đã thấy cách tạo giao diện người dùng trực quan với sự hỗ trợ của Flutter, cho phép người dùng dễ dàng duyệt qua các video và tìm kiếm nội dung yêu thích của họ. Đồng thời, việc tích hợp các chức năng như play, pause và điều chỉnh âm lượng cũng đã giúp tăng cường trải nghiệm xem video.

Một phần quan trọng của ứng dụng là tính năng tương tác với cộng đồng, cho phép người dùng thảo luận, bình luận và chia sẻ ý kiến với nhau. Điều này đã tạo ra một không gian giao tiếp động và thú vị, nơi người dùng có thể kết nối và tương tác với nhau qua các video và bình luận.

Qua quá trình phát triển ứng dụng, chúng ta cũng đã học cách làm việc với dịch vụ đám mây Firebase để lưu trữ và quản lý nội dung video cũng như dữ liệu người dùng. Firebase đã cung cấp một cơ sở hạ tầng mạnh mẽ để xây dựng và triển khai ứng dụng, đồng thời cung cấp tính năng xác thực và bảo mật để đảm bảo an toàn cho người dùng và dữ liệu.

Tóm lại, ứng dụng video này đã mang lại một trải nghiệm đa dạng và thú vị cho người dùng, cho phép họ thư giãn và tận hưởng nội dung video đa dạng. Đồng thời, việc tích hợp tương tác cộng đồng cũng đã thúc đẩy sự kết nối và tương tác giữa người dùng. Từ việc tạo giao diện đến tích hợp dịch vụ đám mây, quá trình phát triển ứng dụng này đã mở ra nhiều cơ hội để tạo ra những ứng dụng hấp dẫn và mang lại giá trị thực sự cho người dùng.

* 1. **Hướng phát triển**

Tối ưu hóa trải nghiệm người dùng (ux): tiếp tục tối ưu hóa giao diện người dùng để đảm bảo trải nghiệm xem video mượt mà và hấp dẫn hơn. cải thiện tính năng điều khiển video, bao gồm chất lượng video, tuỳ chỉnh phụ đề, tốc độ phát lại, và thậm chí cả chế độ xem toàn màn hình.

Tích hợp chức năng xác thực mạnh mẽ: mở rộng tích hợp các phương thức xác thực mạnh mẽ, chẳng hạn như xác thực hai yếu tố (2fa) hoặc xác thực bằng dấu vân tay/nhận diện khuôn mặt để tăng cường bảo mật tài khoản người dùng.

Tạo nền tảng tương tác cộng đồng mạnh mẽ: phát triển chức năng tương tác cộng đồng mạnh mẽ hơn, bao gồm tích hợp hệ thống bình luận, chia sẻ, đánh giá và tạo ra các nhóm người dùng có sở thích chung.

Tích hợp thống kê và phân tích: phát triển tính năng theo dõi và phân tích hiệu suất ứng dụng để hiểu rõ hơn về hành vi người dùng, nội dung phổ biến nhất, thời lượng xem, và nhiều yếu tố khác để tối ưu hóa trải nghiệm người dùng và nội dung.

Hợp tác với nội dung sản xuất và người dùng nội dung: tạo hợp đồng hoặc hợp tác với các nhà sản xuất nội dung, người sáng tạo và người dùng để tạo nội dung hấp dẫn và đa dạng, từ video ngắn đến video dài và nhiều thể loại khác nhau.

Phát triển tiện ích tương tác mạnh mẽ: thêm tính năng tương tác sâu hơn, bao gồm khả năng tạo và lưu danh sách phát, theo dõi người dùng yêu thích, và gợi ý nội dung dựa trên lịch sử xem và sở thích.

Tăng cường bảo mật và quản lý dữ liệu: đảm bảo tính bảo mật cao cho dữ liệu người dùng và nội dung video. xây dựng tính năng quản lý tài khoản để người dùng có thể kiểm soát dữ liệu của người dùng và quyền riêng tư.

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | T. D. Phương, Học lập trình Flutter cùng thực tế, Nhà Xuất Bản Trẻ, 2020. |
| [2] | Đ. M. Tuấn, Flutter và Dart- hướng dẫn phát triển ứng dụng di động, Công Ty Cổ Phần Công Nghệ Trí Tuệ Nhân Tạo, 2020. |
| [3] | H. M. Tuấn, Học Flutter qua ví dụ, Nhà Xuất Bản Đại Học Sư Phạm TpHCM, 2021. |
| [4] | L. Đ. Tuấn, Flutter cho người mới bắt đầu, Công Ty Cổ Phần Sách Và Truyền Thông Việt Nam, 2020. |