

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC TẬP CƠ SỞ

Đề tài: “Xây dựng nền tảng chia sẻ video ngắn sử dụng ReactJS và Django”

Giảng viên hướng dẫn : Nguyễn Mạnh Hùng
Họ và tên sinh viên : Phan Văn Tú
Mã sinh viên : B21DCCN754
Lớp : D21CQCN10-B

Hà Nội – 2024

Mục Lục

LỜI MỞ ĐẦU	1
Chương 1: Giới thiệu về nền tảng chia sẻ video ngắn.	2
1.1 Nền tảng chia sẻ video ngắn	2
1.1.1 Giới thiệu nền tảng chia sẻ video ngắn	2
1.1.2 Xu hướng xem video hiện tại	2
1.1.3 Các nền tảng chia sẻ video ngắn phổ biến hiện nay	2
1.2 Mục tiêu và phạm vi của hệ thống	3
1.2.1 Mục tiêu	3
1.2.2 Phạm vi	3
Chương 2: Giới thiệu các công nghệ chính sử dụng trong dự án.	4
2.1 Các công nghệ sử dụng cho phần Front-end	4
2.1.1 HTML/CSS/JavaScript	4
2.1.2 ReactJS	4
2.2 Các công nghệ sử dụng cho phần Back-end	5
2.2.1 Django	5
2.2.2 Xây Dựng API sử dụng Django REST Framework	7
2.2.3 Cơ sở dữ liệu SQLite	8
2.2.4 Json Web Token	9
2.2.5 Công nghệ Video Streaming	10
Chương 3: Triển khai và xây dựng dự án	12
3.1 Phân tích và thiết kế	12
3.1.1. Mô tả hệ thống bằng ngôn ngữ tự nhiên	12
3.1.2 Xác định tác nhân và UseCase	13
3.1.3 Biểu đồ UseCase tổng quát	14
3.1.4 Biểu đồ hoạt động	15
3.1.5 Biểu đồ lớp	18
3.1.6 Thiết kế CSDL	19
3.2 Xây dựng giao diện của hệ thống	20
KẾT LUẬN	26

DANH MỤC HÌNH ẢNH

Hình 1: Biểu đồ UC tổng quát	14
Hình 2: Biểu đồ hoạt động đăng tải video	15
Hình 3: Biểu đồ hoạt động chỉnh sửa thông tin	15
Hình 4: Biểu đồ hoạt động nhắn tin.....	16
Hình 5: Biểu đồ hoạt động duyệt xem video	17
Hình 6: Biểu đồ hoạt động tìm kiếm	18
Hình 7: Biểu đồ hoạt động quản lý video	18
Hình 8: Sơ đồ lớp.....	19
Hình 9: Sơ đồ cơ sở dữ liệu	20
Hình 10 Trang Login	21
Hình 11 Trang đăng ký	21
Hình 12 Trang Home	22
Hình 13 Trang tìm kiếm	22
Hình 14 Trang Upload	23
Hình 15 Trang nhập thông tin video upload.....	23
Hình 16 Trang nhắn tin.....	24
Hình 17 Trang các cuộc hội thoại.....	24
Hình 18 Trang hồ sơ	25
Hình 19 Trang chỉnh sửa hồ sơ.....	25

LỜI MỞ ĐẦU

Hiện nay việc truy cập và chia sẻ nội dung video trở nên ngày càng phổ biến và không thể phủ nhận được vai trò quan trọng của nó trong việc truyền tải thông điệp và giải trí cho mọi người. Với sự phát triển của công nghệ internet, nhu cầu xem video ngắn theo yêu cầu đã trở nên cực kỳ phổ biến. Do đó, việc tạo ra một nền tảng web để cung cấp dịch vụ xem video ngắn không chỉ đáp ứng nhu cầu ngày càng tăng của người dùng mà còn đem lại nhiều tiện ích và trải nghiệm tốt hơn cho họ.

Với việc sử dụng ReactJS vào dự án này, ta có thể tạo ra những ứng dụng web hấp dẫn, nhanh và hiệu quả. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản. Báo cáo này sẽ được chia thành 3 chương chính như sau:

1. Giới thiệu về nền tảng chia sẻ video ngắn.
2. Giới thiệu các công nghệ chính sử dụng trong dự án.
3. Triển khai và xây dựng dự án.

Chương 1: Giới thiệu về nền tảng chia sẻ video ngắn.

1.1 Nền tảng chia sẻ video ngắn

1.1.1 Giới thiệu nền tảng chia sẻ video ngắn

Video ngắn là những nội dung có thể hiện với định dạng video, thường có thời lượng ngắn từ vài giây đến vài phút tùy vào mục đích của người dùng. Đặc biệt, nội dung video ngắn không giới hạn thể loại, lĩnh vực. Cho phép đa dạng hình thức sáng tạo, giúp người xem dễ dàng tương tác và nắm bắt nội dung.

Nền tảng chia sẻ video ngắn là nền tảng trực tuyến cho phép người sử dụng tạo, tải lên, xem và chia sẻ các video ngắn. Cung cấp một giao diện thân thiện dễ sử dụng, tạo ra nội dung hấp dẫn và tương tác với người dùng khác.

Nền tảng chia sẻ video ngắn đã trở thành một phần quan trọng của văn hóa truyền thông xã hội, đem lại cơ hội cho người dùng để thể hiện sự sáng tạo của họ, kết nối với người khác và tiếp cận nội dung giải trí, học tập và chia sẻ thông điệp.

1.1.2 Xu hướng xem video hiện tại

Trước đây Youtube là top những nền tảng xem video phổ biến và có số lượng người dùng đông đảo. Nhưng từ khi có sự ra đời của Tiktok, xu hướng xem video của mọi người đã có sự thay đổi lớn.

Ra mắt thị trường vào năm 2018, nhưng Tiktok đã nhanh chóng vượt mặt các anh lớn trong ngành và thành công thu hút sự chú ý của hầu hết khán giả trẻ trên khắp thế giới. Theo thống kê, vào tháng 4 năm 2022, Tiktok đã vượt Youtube về Tổng thời lượng trung bình mà một người Mỹ trưởng thành dành ra mỗi ngày trên nền tảng.

Trước tình hình đó, ông lớn Youtube đã tung ra nền tảng Youtube Short, một nền tảng xem video ngắn tương tự Tiktok để cạnh tranh xu thế hiện tại.

Các điều trên đã chứng minh thay vì xem các video có thời lượng dài, người dùng hiện nay ưa chuộng các video ngắn, mang tính giải trí cao hơn. Vậy nên ta thấy được video ngắn đang và sẽ tiếp tục là xu hướng sử dụng mạng xã hội hiện tại.

1.1.3 Các nền tảng chia sẻ video ngắn phổ biến hiện nay

Nền tảng video ngắn đã và đang trở nên rất thịnh hành trong thời gian qua. Trong đó, phổ biến nhất chính là TikTok, Facebook Reels và YouTube Shorts, ngoài ra còn một số nền tảng khác.

- Sơ qua về TikTok:

TikTok, một ứng dụng chia sẻ video được thiết kế bởi một công ty công nghệ có trụ sở tại Bắc Kinh tên là ByteDance. Được ra mắt vào năm 2017, đến năm 2018 TikTok có mặt ở hơn 150 thị trường và có 75 ngôn ngữ. Vào tháng 2/2019, TikTok, cùng với Douyin, đã đạt một tỷ lượt tải xuống trên toàn cầu. Một số thông tin bên lề:

- Người dùng đang hoạt động: Hơn một tỷ người.
- Lượt tải xuống: Hơn 3 tỷ lần cài đặt trên toàn thế giới.
- Tốc độ tăng trưởng: Từ tháng 1/2018 đến tháng 7/2020, cơ sở người dùng toàn cầu của TikTok đã tăng 1157,76%, với tốc độ tăng trưởng ở Mỹ là 787,86%.
- Tương tác của người dùng: Thời lượng phiên trung bình gần 11 phút, vượt xa thời gian của Facebook và Instagram.
- Tiêu thụ video: TikTok xem khoảng 1 tỷ video mỗi ngày.

1.2 Mục tiêu và phạm vi của hệ thống

1.2.1 Mục tiêu

- Mục tiêu chính của dự án là tạo ra một trang web trực tuyến nơi người dùng có thể xem và chia sẻ video ngắn về nhiều chủ đề khác nhau.
- Tính năng cần có: Đăng nhập, đăng ký, chỉnh sửa trang cá nhân, duyệt xem video, tải lên, chia sẻ, tìm kiếm, thích, bình luận và nhắn tin ...
- Yêu cầu: giao diện đơn giản thân thiện, có khả năng xem video mượt mà, bảo mật tốt.

1.2.2 Phạm vi

- Xây dựng ứng dụng web hướng tới khả năng responsive trên cả 2 nền tảng mobile và desktop

Chương 2: Giới thiệu các công nghệ chính sử dụng trong dự án.

2.1 Các công nghệ sử dụng cho phần Front-end

2.1.1 HTML/CSS/JavaScript

HTML là viết tắt của từ HyperText Markup Language, hay còn gọi là "Ngôn ngữ Đánh dấu Siêu văn bản", là một ngôn ngữ đánh dấu được thiết kế để tạo nên cấu trúc và các thành phần của các trang Web, hay còn được coi là khung xương hoặc bản vẽ thiết kế của trang Web. Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag và attributes) được bao bọc trong một dấu ngoặc nhọn và được lưu dưới dạng .html hoặc .htm.

CSS là viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Nếu ví HTML như các thành phần để xây nên một căn nhà thì CSS chính là màu sơn cho căn nhà đó. CSS giúp chúng ta thêm style như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc... cho trang HTML đó.

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Bằng cách tương tác với DOM, JavaScript có thể có thể lắng nghe và thay đổi các thành phần trang Web giúp tạo ra các tương tác giữa trang Web và người dùng.

Bằng cách kết hợp giữa HTML, CSS và JavaScript, ta có thể tạo ra phần giao diện trang Web đầy đủ, đẹp và tương tác tốt với người dùng. HTML, CSS và JavaScript được xem là những thành phần không thể thiếu để hình thành nên một website hoạt động ổn định và bắt mắt như hiện nay.

2.1.2 ReactJS

- **Tổng quan về ReactJS**

- **React** là một thư viện của JavaScript được phát triển bởi Facebook theo chuẩn SPA (Single Page Application) được ra mắt vào năm 2013. React mang đến giải pháp để xây dựng giao diện người dùng với tốc độ phản hồi tuyệt vời cùng với phương pháp render website nhanh chóng.

- React cho phép lập trình viên phân tách các giao diện phức tạp thành nhiều thành phần nhỏ lẻ, đơn giản, giúp cho việc phát triển, bảo trì tái sử dụng trở nên hiệu quả.
- React còn cung cấp khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:
 - JSX: JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản.
 - VirtualDOM: Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả
- Ưu điểm:
 - ReactJS tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó.
 - Thân thiện với SEO (Search Engine Optimization): ReactJS là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
 - Khả năng tái sử dụng các components: Các components được tách riêng với nhau nên một khi có sự ảnh hưởng đến một component nó sẽ không ảnh hưởng đến các components khác. Nếu các components đủ linh hoạt nó sẽ được tái sử dụng nhiều lần, giảm thiểu thời gian viết mã.
 - Debug dễ dàng: Facebook đã phát hành 1 Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cũng như quá trình viết mã
 - Dễ học: ReactJS là thư viện của JavaScript và có một cộng đồng lớn hỗ trợ.
- Nhược điểm:
 - Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
 - React khá nặng nếu so với các framework khác: React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

2.2 Các công nghệ sử dụng cho phần Back-end

2.2.1 Django

Django là web framework dựa trên ngôn ngữ lập trình bậc cao Python, khuyến khích sự phát triển nhanh chóng, gọn gàng và thiết kế thực dụng. Django miễn phí và mã nguồn mở.

Bất kỳ ứng dụng web nào cũng đều được cấu tạo từ hai phần: máy chủ và máy khách. Khách hoặc đối tượng truy cập trang web bằng trình duyệt. Khi nhập URL vào trình duyệt của mình, họ sẽ gửi một yêu cầu tới máy chủ web chứa ứng dụng web đang chạy. Các máy chủ xử lý yêu cầu bằng cách sử dụng cơ sở dữ liệu và gửi thông tin ngược về máy khách dưới dạng phản hồi. Mã máy khách hiển thị thông tin cho đối tượng truy cập dưới dạng trang web.

Django quản lý mã cho hệ thống yêu cầu và phản hồi này bằng cách sử dụng kiến trúc Mô hình-Khung nhìn-Mẫu (MVT).

- Model – Dữ liệu bạn muốn thể hiện, thường là dữ liệu trong database.
 - View - Một trình xử lý yêu cầu trả về template và nội dung tương ứng - dựa trên yêu cầu từ người dùng.
 - Template - Một tệp văn bản (như một tệp HTML) chứa bố cục của trang web, với logic về cách hiển thị dữ liệu.
- Một số tính năng của Django:
 - ORM (Object-Relational Mapping): Django cung cấp một cách thuận tiện để tương tác với cơ sở dữ liệu thông qua ORM, giúp việc tạo và thao tác với các model dễ dàng hơn.
 - Hệ thống URL định tuyến: Django cung cấp một cách linh hoạt để xác định các URL và liên kết chúng với các view tương ứng.
 - Hệ thống template: Django cung cấp hệ thống template mạnh mẽ cho việc tạo ra các giao diện người dùng.
 - Bảo mật: Django đi kèm với các tính năng bảo mật như xác thực người dùng, quản lý phiên, và bảo vệ chống tấn công CSRF.
 - Admin site tự động tạo: Django cung cấp một giao diện quản trị tự động được tạo dựa trên các model đã định nghĩa, giúp quản lý dữ liệu dễ dàng hơn.
 - Ưu điểm của Django:
 - Phát triển nhanh chóng: Django cung cấp các tính năng tích hợp sẵn và một cấu trúc dự án rõ ràng, giúp việc phát triển ứng dụng web trở nên nhanh chóng và hiệu quả.
 - Bảo mật: Django đi kèm với các tính năng bảo mật tích hợp sẵn, giúp bảo vệ ứng dụng khỏi các cuộc tấn công phổ biến.
 - Cộng đồng lớn: Django có một cộng đồng phát triển lớn, với nhiều tài nguyên hỗ trợ và các thư viện bổ sung.
 - Độ tin cậy: Django đã được sử dụng rộng rãi trong nhiều dự án lớn và được thử nghiệm thực tế, cho phép nhà phát triển tin tưởng vào sự ổn định và độ tin cậy của nó. Hàng nghìn dự án nguồn mở và trang web tầm cỡ sử dụng Django như: Instagram, Mozilla Firefox, Pinterest, National Geographic.

2.2.2 Xây Dựng API sử dụng Django REST Framework

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP/HTTPS.

Các thành phần của RESTful API:

- API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.
- REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE,... đến một URL để xử lý dữ liệu.
- RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...):

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

Django REST Framework hoặc DRF là một bộ công cụ mạnh mẽ cho việc phát triển các API Web trong các ứng dụng Django. DRF cho phép bạn tạo ra các API RESTful nhanh chóng bằng cách cung cấp các lớp và quy ước được xây dựng sẵn.

DRF có các tính năng chính sau:

- **Serialization:** Serialization cho phép bạn tự động chuyển đổi các mô hình Django thành JSON và ngược lại, làm cho việc truyền dữ liệu giữa máy khách và máy chủ trở nên mượt mà.
- **URL Routing:** DRF cung cấp định tuyến URL giống như router của Django, cho phép bạn liên kết các URL với các chế độ xem API của họ một cách hiệu quả.
- **Xác thực:** DRF cung cấp các phương thức xác thực khác nhau bao gồm xác thực cơ bản, token. Nó cũng hỗ trợ JWT thông qua một gói bên thứ ba.
- **Quyền:** DRF cho phép kiểm soát chi tiết quyền của người dùng.
- **Giới hạn tốc độ/Rate Limiting:** Các tính năng giới hạn tốc độ và tốc độ giúp bạn giới hạn số lượng yêu cầu mà một máy khách có thể thực hiện trong một khoảng thời gian nhất định.
- **Phân trang:** DRF cung cấp hỗ trợ tích hợp sẵn cho việc phân trang các bộ dữ liệu lớn trong các phản hồi API.
- **Browsable API:** DRF có một giao diện HTML thân thiện với người dùng để kiểm tra và khám phá các API trực tiếp từ trình duyệt.

2.2.3 Cơ sở dữ liệu SQLite

Cơ sở dữ liệu có nhiều loại như:

- Cơ sở dữ liệu dạng tập tin (ví dụ: text, ASCII, dbf)
- Cơ sở dữ liệu quan hệ (dạng bảng)
- Cơ sở dữ liệu hướng đối tượng
- Cơ sở dữ liệu bán cấu trúc (xml, json)

Các ngôn ngữ lập trình thường cung cấp các thư viện để làm việc với từng loại cơ sở dữ liệu.

Django hỗ trợ chính thức nhiều cơ sở dữ liệu như PostgreSQL, MariaDB, MySQL, Oracle, SQLite... Ngoài ra còn có một số cơ sở dữ liệu bên thứ 3. Bên cạnh đó, Django còn tích hợp sẵn một cơ sở dữ liệu mang tên SQLite.

- Sơ qua về cơ sở dữ liệu SQLite:

SQLite là một trong những cơ sở dữ liệu phổ biến nhất được tích hợp sẵn trong Django. Nó là một hệ thống quản lý cơ sở dữ liệu phẳng (flat-file database), có nghĩa là dữ liệu được lưu trữ trong một tập tin duy nhất trên đĩa và không yêu cầu một máy chủ cơ sở dữ liệu riêng biệt để hoạt động.

SQLite có kích thước nhỏ, chỉ khoảng 700KB, do D. Richard Hipp tạo ra năm 2000 bằng ngôn ngữ lập trình C.

SQLite hoạt động dưới dạng một thư viện, được nhúng trong một chương trình chứ không theo mô hình client-server. SQLite được sử dụng để thao tác với cơ sở dữ liệu trong các ứng dụng desktop, mobile và website.

SQL có ưu điểm nhẹ, thân thiện với người dùng và có tính di động cao.

Tuy nhiên SQLite có nhược điểm như: Không thích hợp cho các hệ thống có lượng dữ liệu lớn, phát sinh liên tục và tại một thời điểm, chỉ cho phép một người ghi dữ liệu, nhưng với quy mô dự án hiện tại nhỏ nên SQLite là một lựa chọn không tệ và có thể dễ dàng scale lên bằng việc hỗ trợ của Django với các cơ sở dữ liệu khác.

2.2.4 Json Web Token

JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hóa RSA.

Cấu trúc của một JWT là:

```
<base64-encoded header>.<base64-encoded payload>.<base64-encoded signature>
```

JWT gồm 3 thành phần chính:

- Header: gồm 2 phần chính là loại token và thuật toán dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

Ví dụ:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

- Payload: chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

Ví dụ Payload:

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

- Signature: là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    secret)
```

Đặt chúng cùng nhau ta có một JWT

Một trong những tình huống ứng dụng JWT thường gặp, đó là:

- Authentication: Tình huống thường gặp nhất, khi user logged in, mỗi request tiếp đó đều kèm theo chuỗi token JWT, cho phép người dùng có thể truy cập đường dẫn, dịch vụ và tài nguyên được phép ứng với token đó. Single Sign On cũng là một chức năng có sử dụng JWT một cách rộng rãi, bởi vì chuỗi JWT có kích thước đủ nhỏ để đính kèm trong request và sử dụng ở nhiều hệ thống thuộc các domain khác nhau.
- Information Exchange: JSON Web Token cũng là một cách hữu hiệu và bảo mật để trao đổi thông tin giữa nhiều ứng dụng, bởi vì JWT phải được ký (bằng cặp public / private key), bạn có thể chắc rằng người gửi chính là người mà họ nói rằng họ là (nói tóm tắt hơn là không hoặc khó để mạo danh bằng JWT), ngoài ra, chữ ký cũng được tính toán dựa trên nội dung của header và nội dung payload, nhờ đó, bạn có thể xác thực được nội dung là nguyên bản, chưa được chỉnh sửa hoặc can thiệp. Tuy nhiên, một lưu ý hết sức quan trọng là do cấu trúc của JWT đơn giản nên JWT có thể dễ dàng bị decode, do vậy, bạn không nên dùng JWT để transfer các thông tin nhạy cảm.

2.2.5 Công nghệ Video Streaming

Như ta đã thấy, việc xem video trực tuyến gấp phải một vấn đề đó là tốc độ mạng của người dùng. Việc tải và xem video phụ thuộc rất lớn vào tốc độ mạng. Tùy thuộc vào dung lượng của video, việc tải video có thể mất ít nhiều thời gian và quan trọng hơn là video phải tải hết thì người dùng mới bắt đầu xem được. Do đó chúng ta phải tìm giải pháp cho vấn đề này.

Hiện nay có rất nhiều giải pháp có thể áp dụng được, một trong những giải pháp đó là Streaming Video như Dynamic Adaptive Streaming over HTTP (DASH) hoặc HTTP Live Streaming (HLS). Mấu chốt là sử dụng giao thức DASH hoặc HLS cho phép bạn phân chia video thành các phần nhỏ gọi là các đoạn (segment). Mỗi đoạn được tải về một cách độc lập, giúp giảm thiểu thời gian chờ đợi khi bắt đầu phát video và cải thiện trải nghiệm của người dùng.

- HTTP Live Streaming:

HLS (HTTP Live Streaming) là một giao thức truyền phát video trực tuyến phổ biến được phát triển bởi Apple để phát video trên internet. Đây là một trong những giao thức phổ biến nhất được sử dụng cho việc phân phối video

trực tuyến với chất lượng cao trên các thiết bị và trình duyệt khác nhau. Một số ưu điểm:

- Phương pháp Phân phối động (Dynamic Adaptive Streaming): điều chỉnh chất lượng video theo chất lượng mạng của người dùng, tạo ra một trải nghiệm xem video mượt mà.
- Định dạng Video M3U8: sử dụng định dạng M3U8 để chia nhỏ video thành các phần nhỏ hơn,
- Hỗ trợ rộng rãi trên nhiều nền tảng.

Chương 3: Triển khai và xây dựng dự án

3.1 Phân tích và thiết kế

3.1.1. Mô tả hệ thống bằng ngôn ngữ tự nhiên

- **Mục tiêu hệ thống:** Trang web trực tuyến nơi người dùng có thể xem và chia sẻ video ngắn về nhiều chủ đề khác nhau, kết bạn và trò chuyện...
- **Phạm vi hệ thống:** Trang web có hai loại người dùng chính:
 - Khách: được phép duyệt xem các video, tìm kiếm video hoặc người dùng khác
 - Người dùng (yêu cầu đăng nhập): bao gồm tất cả quyền của khách và các chức năng còn lại như: đăng tải video, nhắn tin, tương tác với video (like, comment, ...), theo dõi người dùng, chỉnh sửa trang cá nhân, quản lý video, ...
- **Mô tả hoạt động của một số chức năng chính:**
 - **Người dùng đăng tải video:** Người dùng đăng nhập vào hệ thống => chọn chức năng đăng tải video => hệ thống hiện lên biểu mẫu đăng tải video => người dùng nhập mô tả cho video và chọn video cần đăng tải, sau đó nhấn Upload => hệ thống chuyển video đến server để xử lý và lưu trữ => hệ thống thông báo kết quả và đưa người dùng quay lại trang Home.
 - **Người dùng duyệt xem và tương tác video:** Người dùng đăng nhập vào hệ thống => hệ thống xác thực thông tin và đưa người dùng đến trang Home, tại đây người dùng có thể duyệt xem video bằng cách vuốt trên màn hình hoặc sử dụng các phím mũi tên điều hướng để chuyển đổi giữa các video, click vào màn hình để tiếp tục/tạm dừng video đang phát, tương tác với các nút tương tác tại trang Home như like, share, comment, ... hoặc chuyển đổi duyệt video để xuất hoặc chỉ video của những người mình theo dõi
 - **Người dùng thực hiện tìm kiếm:** Người dùng đăng nhập vào hệ thống => chọn chức năng tìm kiếm => người dùng nhập thông tin muốn tìm kiếm và chọn lọc kết quả theo video hoặc người dùng => hệ thống tiếp nhận thông tin và thực hiện truy vấn => hệ thống trả về kết quả tìm kiếm => người dùng có thể chọn 1 kết quả, nếu là video thì người dùng được chuyển hướng tới trang phát video đó, nếu chọn người dùng thì người dùng sẽ được chuyển đến trang cá nhân của họ.
 - **Người dùng nhắn tin:** Người dùng đăng nhập vào hệ thống, chọn chức năng nhắn tin => hệ thống hiển thị trang danh sách những người nhắn tin gần nhất, tại đây người dùng có thể chọn một đoạn hội thoại để tiếp tục, hoặc tìm kiếm 1 đoạn hội thoại => sau khi chọn người

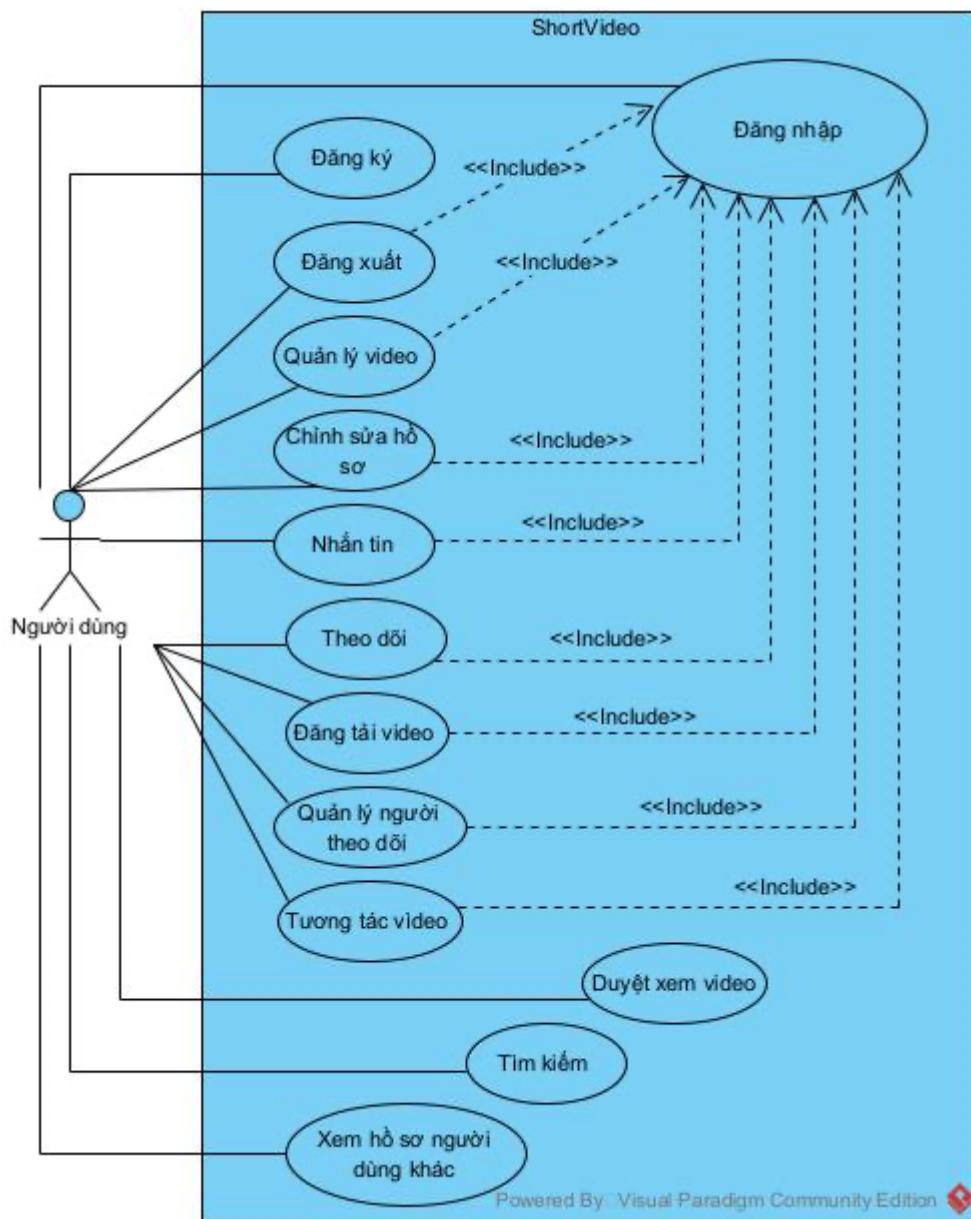
dùng được chuyển đến trang nhắn tin => người dùng thực hiện việc nhắn tin như bình thường.

- Chức năng tại trang cá nhân: Người dùng đăng nhập vào hệ thống => chọn chức năng hồ sơ => hệ thống đưa người dùng tới trang cá nhân, tại đây người dùng có thể thực hiện một số chức năng như xem thông tin bản thân, chỉnh sửa hồ sơ, quản lý người đang theo dõi và được theo dõi, quản lý video của bản thân hoặc những video đã like.

3.1.2 Xác định tác nhân và UseCase

- Tác nhân chính: Người dùng
- Một số UC chính:
 - Đăng nhập: Người dùng cung cấp thông tin đăng nhập (tên đăng nhập và mật khẩu) để truy cập vào hệ thống.
 - Đăng ký: Người dùng tạo một tài khoản mới bằng cách cung cấp thông tin cá nhân yêu cầu.
 - Đăng xuất: Người dùng thoát khỏi tài khoản đang đăng nhập.
 - Tìm kiếm: Người dùng tìm kiếm video hoặc người dùng cụ thể.
 - Theo dõi: Người dùng có thể theo dõi người dùng khác
 - Nhắn tin: Người dùng nhắn tin với người dùng khác
 - Duyệt xem video: Người dùng duyệt xem các video có trên hệ thống.
 - Tương tác video: Người dùng tương tác với nội dung video, bao gồm thích, bình luận và chia sẻ.
 - Đăng tải video: Người dùng tải lên nội dung video của riêng họ lên hệ thống
 - Chỉnh sửa hồ sơ: Chỉnh sửa thông tin trên trang cá nhân
 - Quản lý video: Người dùng quản lý các video của bản thân hoặc video đã thích.
 - Quản lý người theo dõi: Người dùng quản lý những người mình theo dõi.
 - Xem hồ sơ người dùng khác: Người dùng xem thông tin người dùng khác và video của họ

3.1.3 Biểu đồ UseCase tổng quát

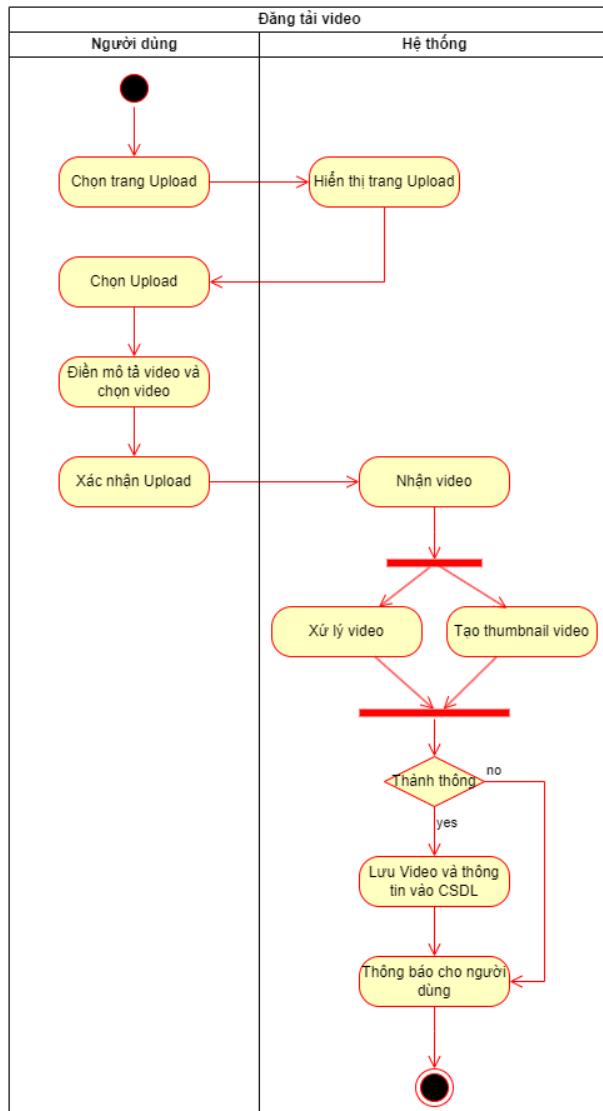


Hình 1: Biểu đồ UC tổng quát

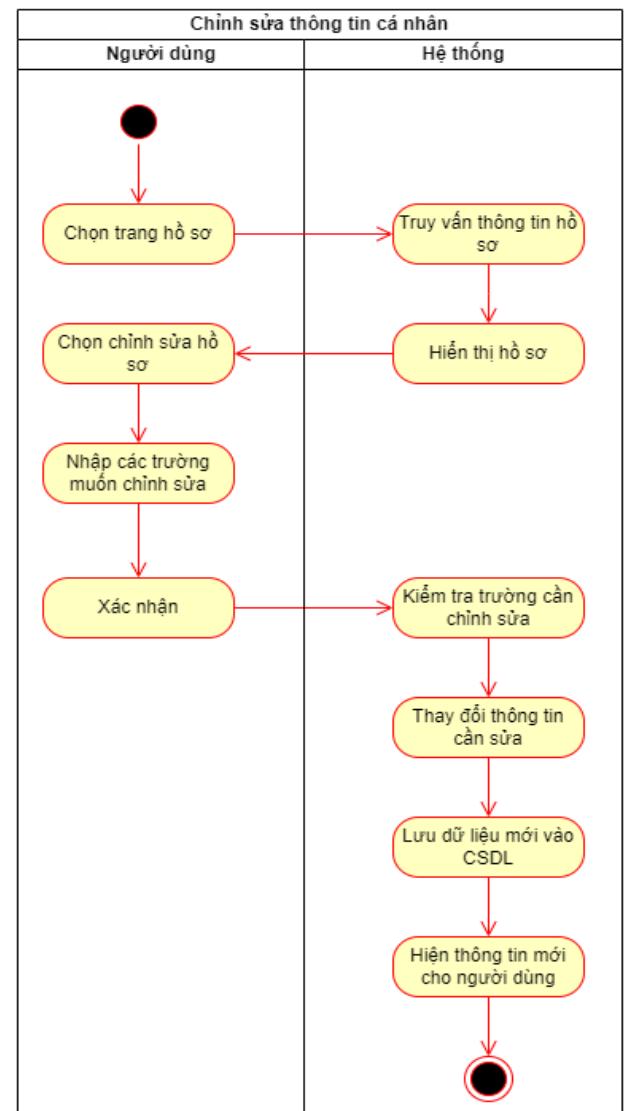
*Mô tả: Ngoài UC tìm kiếm, duyệt xem video, xem hồ sơ người dùng không yêu cầu đăng nhập, các UC còn lại đều yêu cầu đăng nhập để thực hiện.

3.1.4 Biểu đồ hoạt động

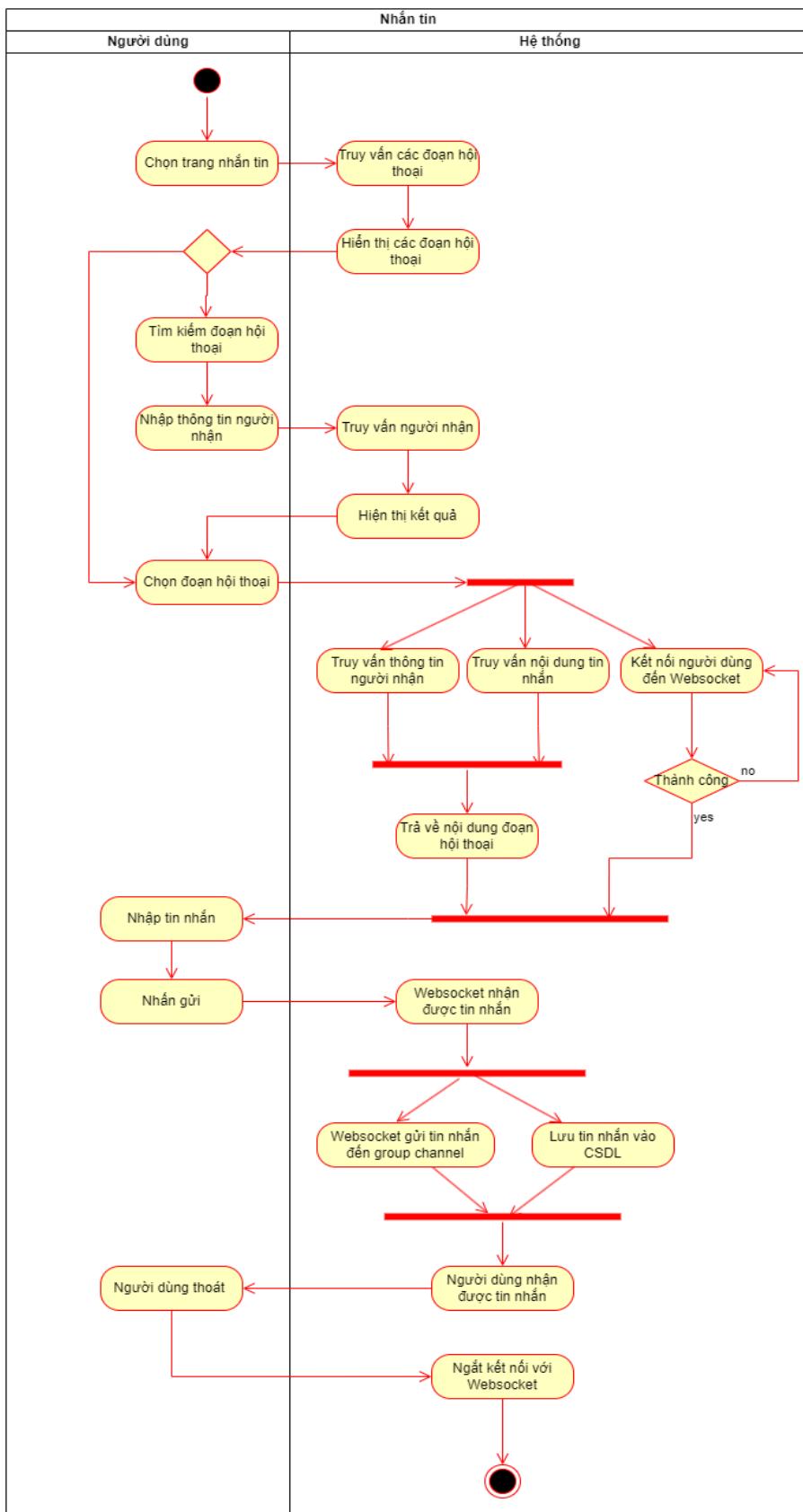
Sau đây là biểu đồ hoạt động của một số chức năng chính có trong trang web:



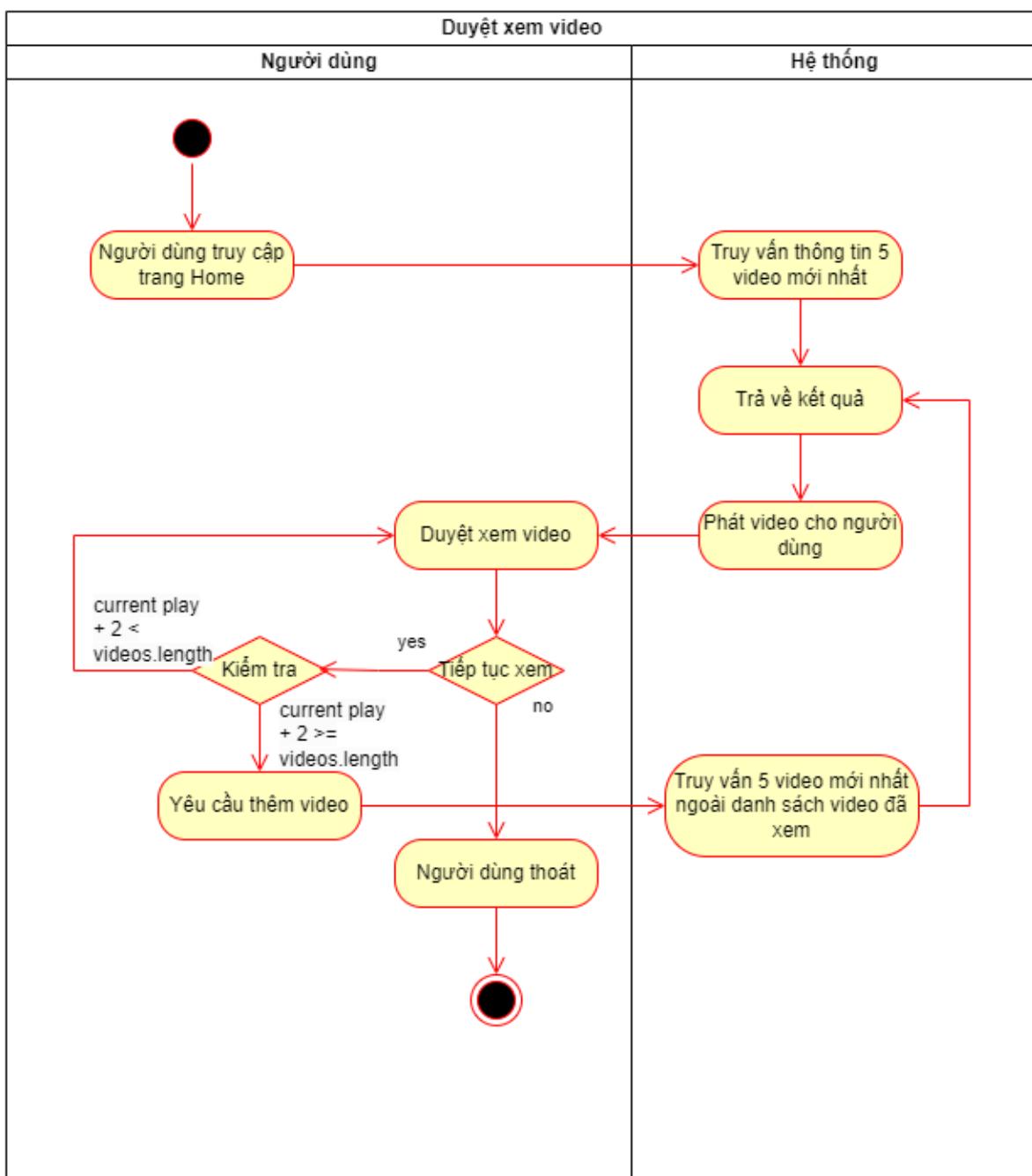
Hình 2: Biểu đồ hoạt động đăng tải video



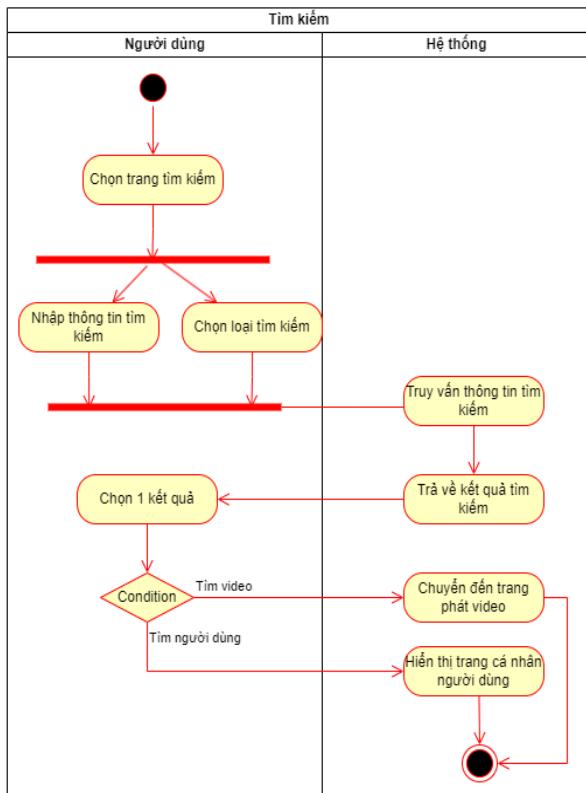
Hình 3: Biểu đồ hoạt động chỉnh sửa thông tin



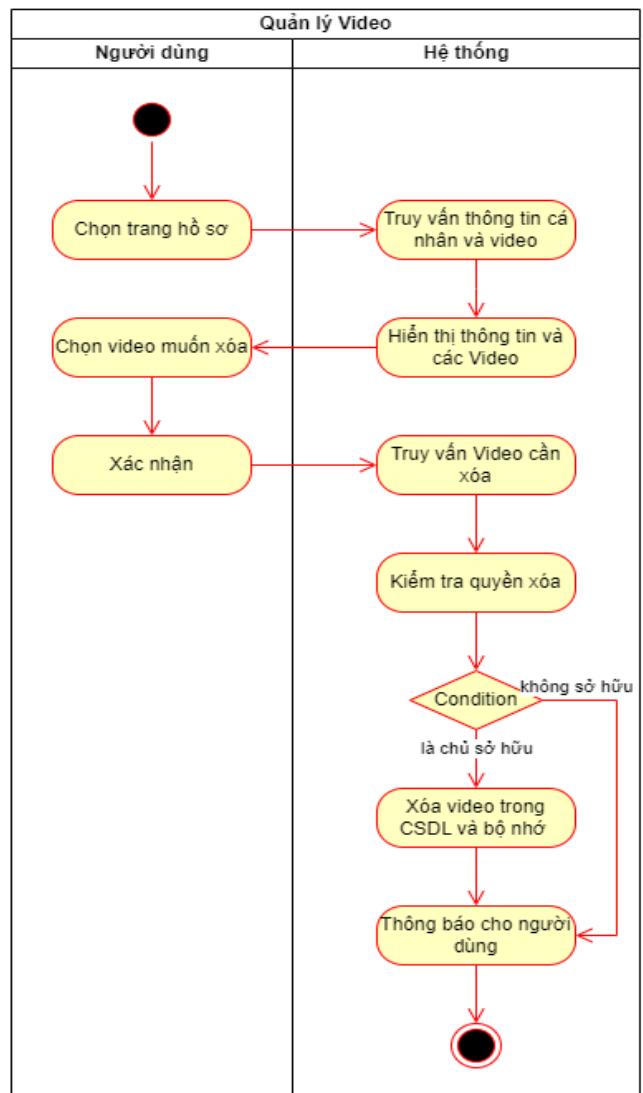
Hình 4: Biểu đồ hoạt động nhắn tin



Hình 5: Biểu đồ hoạt động duyệt xem video



Hình 6: Biểu đồ hoạt động tìm kiếm

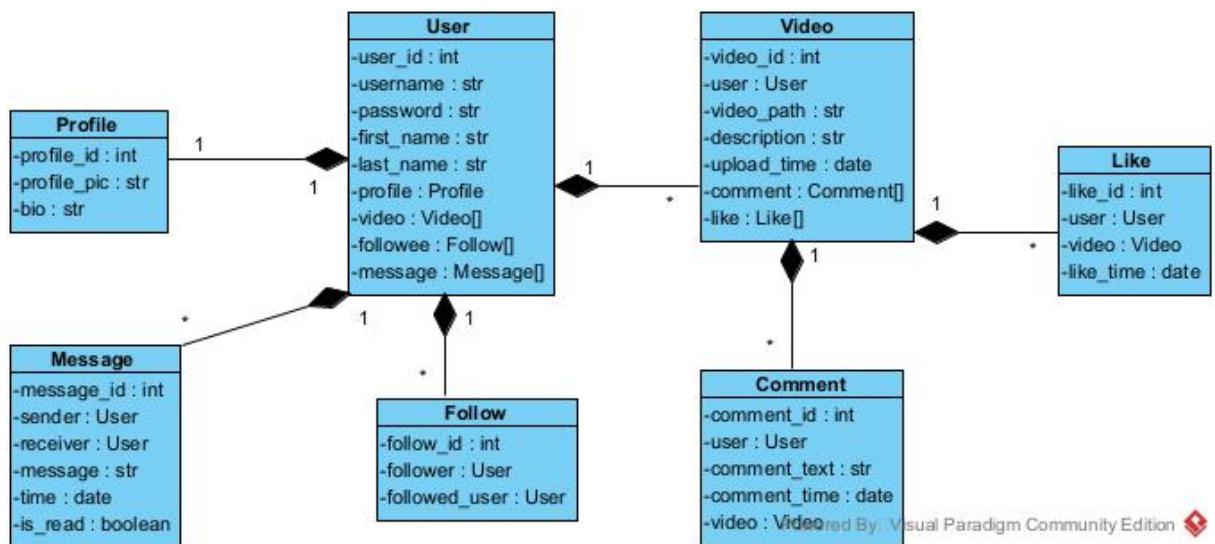


Hình 7: Biểu đồ hoạt động quản lý video

3.1.5 Biểu đồ lớp

- Hệ thống gồm các lớp:
 - User: id, username, password, first_name, last_name
 - Profile: id, profile_pic, bio
 - Video: id, user , video_path, description, upload_time
 - Comment: id, user, video , comment_text, comment_time
 - Like: id, video, user, like_time
 - Follow: id, follower, followed_user
 - Message: id, sender, receiver, message, time, is_read
- Quan hệ giữa các lớp
 - Một User chỉ có 1 Profile và ngược lại: quan hệ giữa User và Profile là 1-1

- Một User có thể có nhiều Video nhưng 1 Video chỉ thuộc về 1 User: quan hệ giữa User và Video là 1-n
- Một Video có thể có nhiều Like nhưng 1 Like chỉ thuộc về 1 Video: quan hệ giữa Video và Like là 1-n
- Một Video có thể có nhiều Comment nhưng 1 Comment chỉ thuộc về 1 Video: quan hệ giữa Video và Comment là 1-n
- Một User có thể có nhiều người follow: quan hệ giữa User và Follow là 1-n
- Một User có thể có nhiều Message: quan hệ giữa User và Message là 1-n
- Quan hệ giữa các thực thể:
 - Profile, Video, Follow, Message là thành phần của User
 - Like, Comment là thành phần của Video

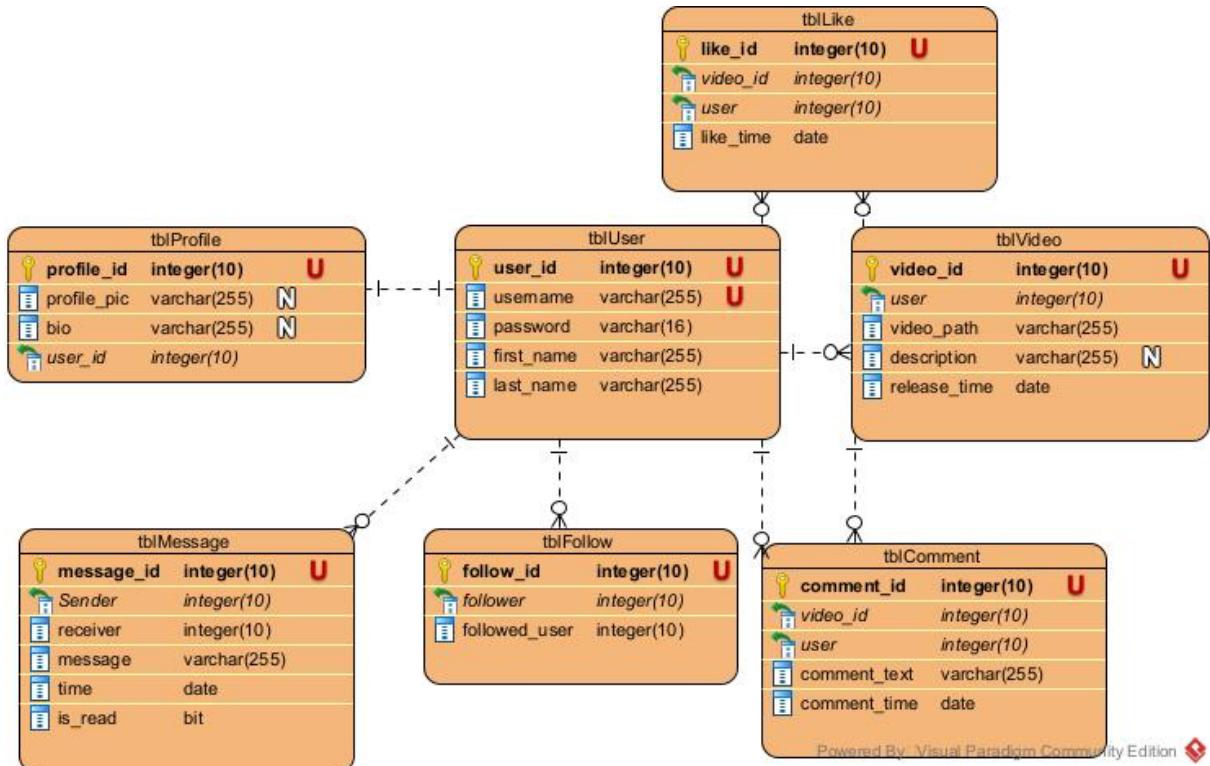


Hình 8: Sơ đồ lớp

3.1.6 Thiết kế CSDL

- Bảng `tblUser`: `user_id` (pk), `username`, `password`, `first_name`, `last_name`
- Bảng `tblProfile`: `profile_id` (pk), `profile_pic`, `bio`, `user_id`
- Bảng `tblVideo`: `video_id` (pk), `user`, `video_path`, `description`, `upload_time`
- Bảng `tblLike`: `like_id` (pk), `video_id`, `user`, `like_time`
- Bảng `tblComment`: `comment_id` (pk), `video_id`, `user`, `comment_text`, `comment_time`
- Bảng `tblFollow`: `follow_id` (pk), `follower`, `followed_user`
- Bảng `tblMessage`: `message_id` (pk), `sender`, `receiver`, `message`, `time`, `is_read`
- Bảng `tblProfile` có khóa ngoại là `user_id`
- Bảng `tblVideo` có khóa ngoại là `owner`

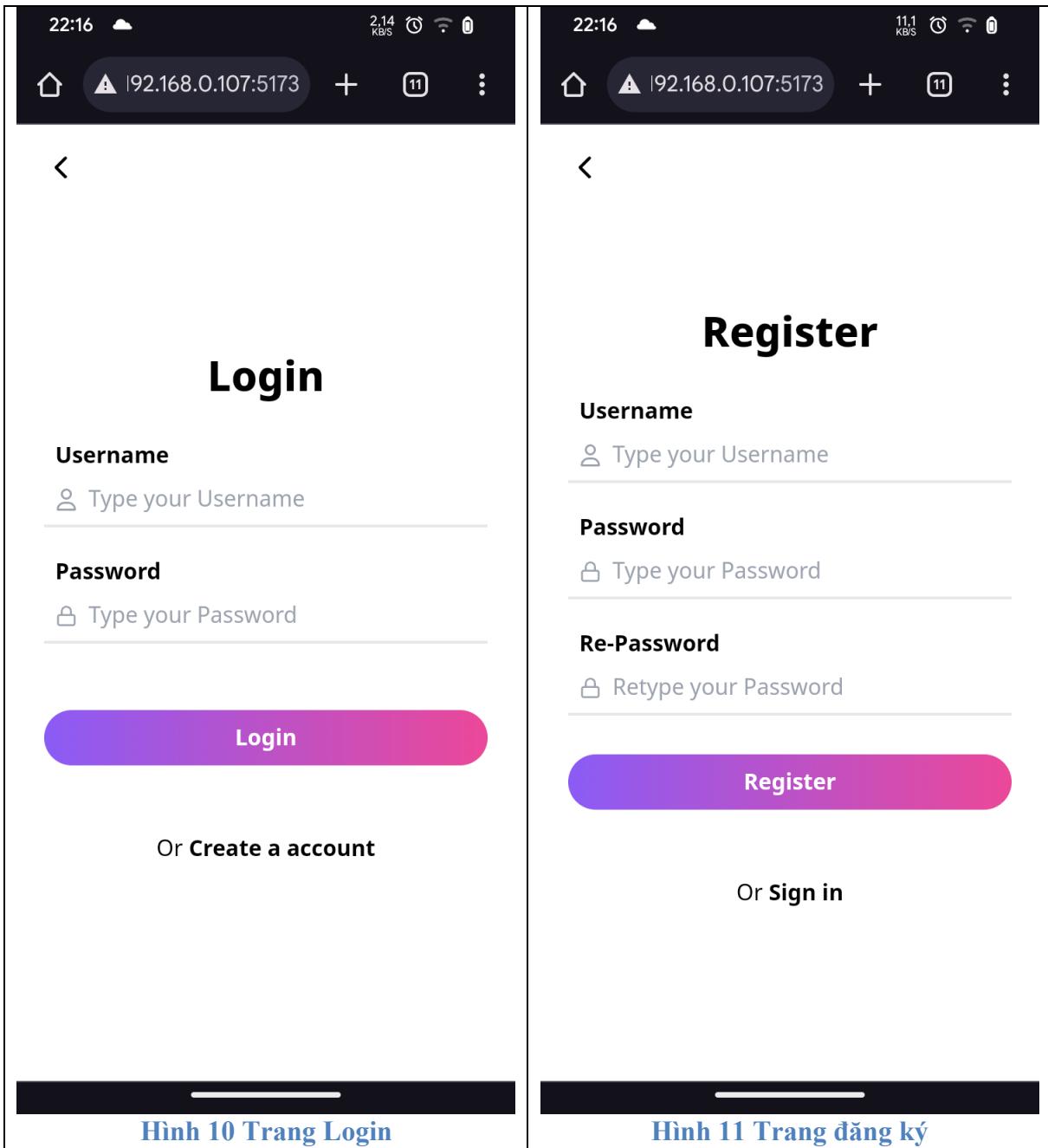
- Bảng tblLike có khóa ngoại là owner và video_id
- Bảng tblComment có khóa ngoại là video_id và owner
- Bảng tblFollow có khóa ngoại là follower
- Bảng tblMessage có khóa ngoại là sender

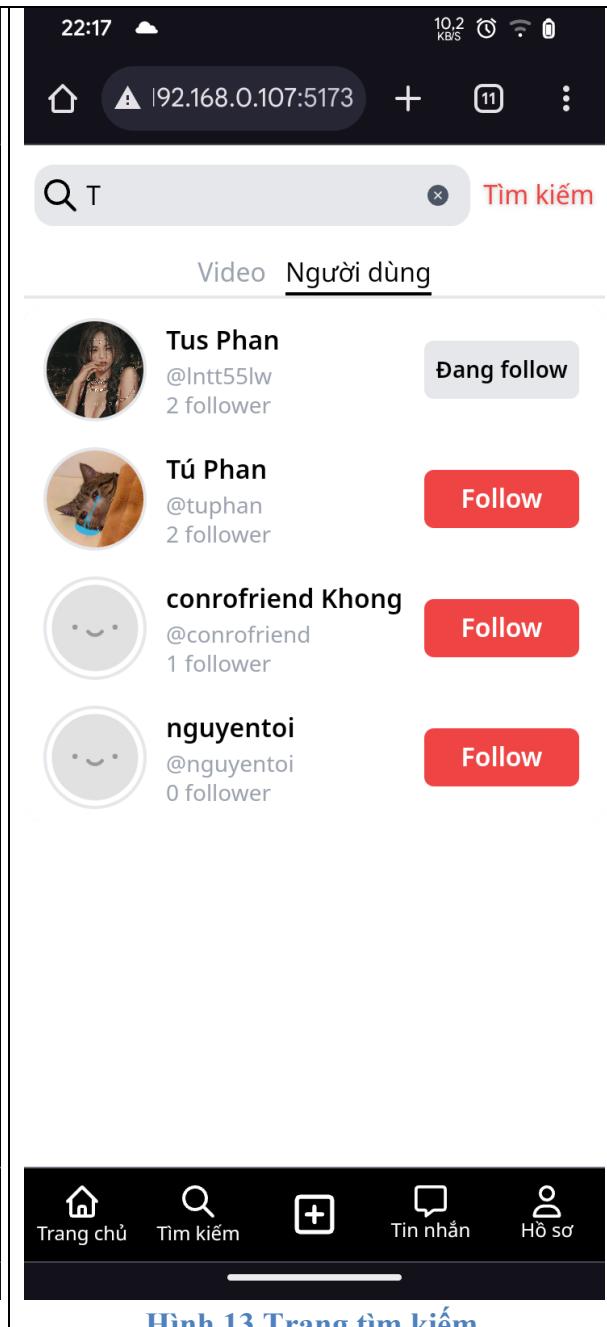
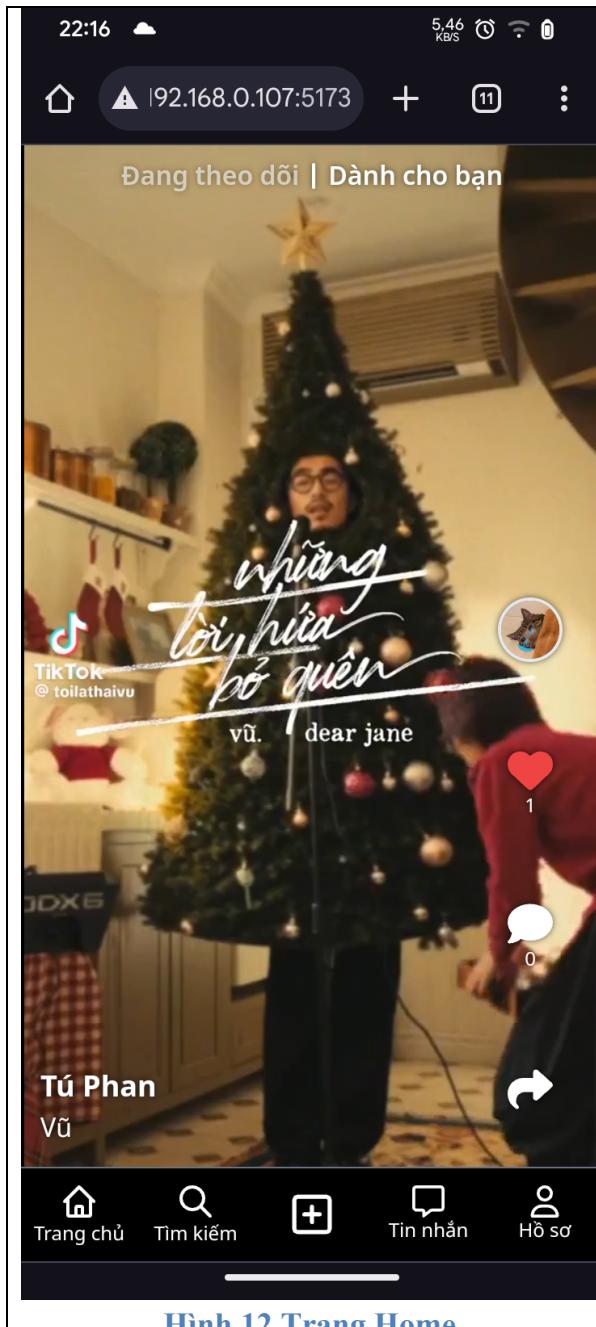


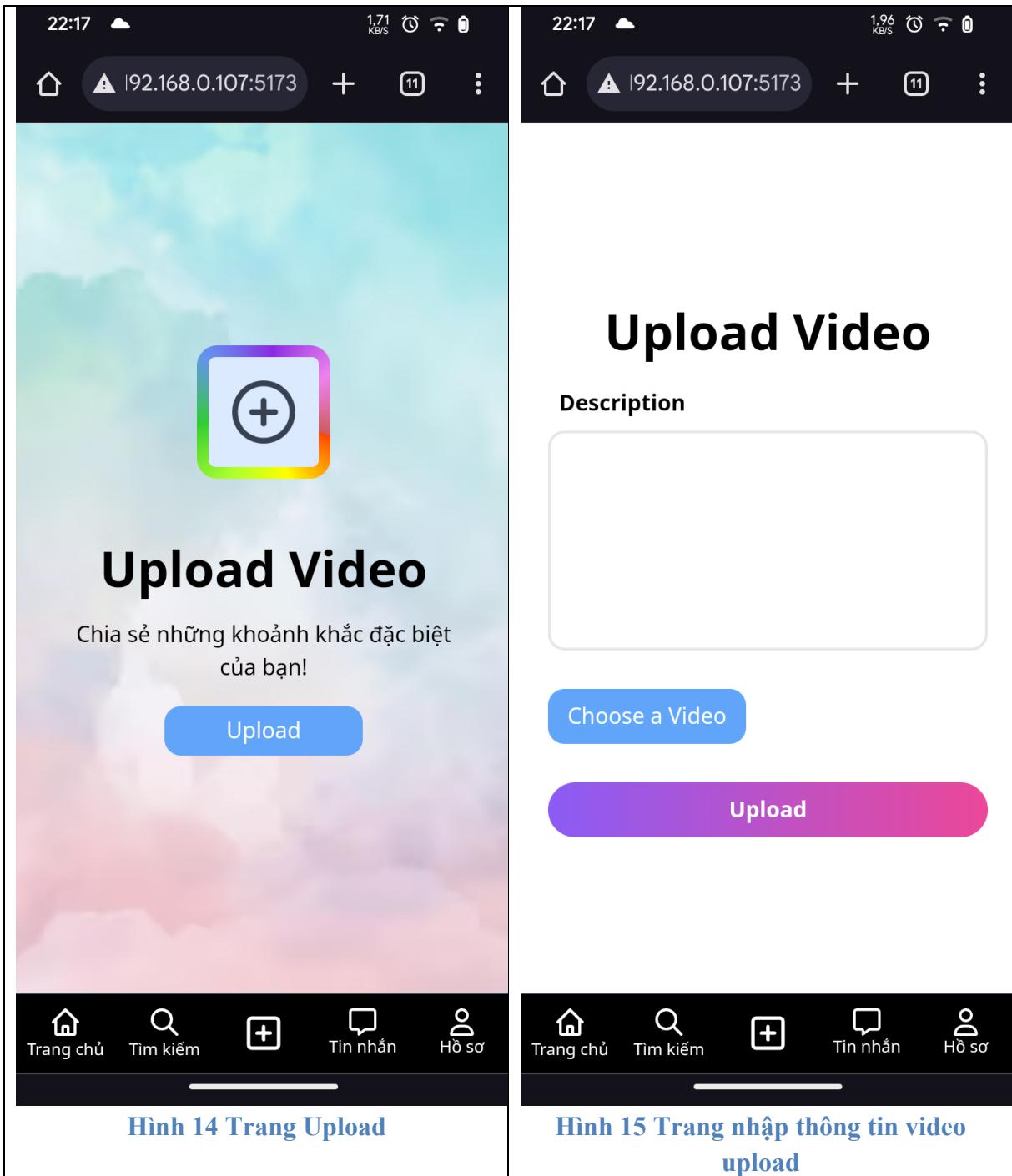
Hình 9: Sơ đồ cơ sở dữ liệu

3.2 Xây dựng giao diện của hệ thống

Dưới đây là một số giao diện của trang web:

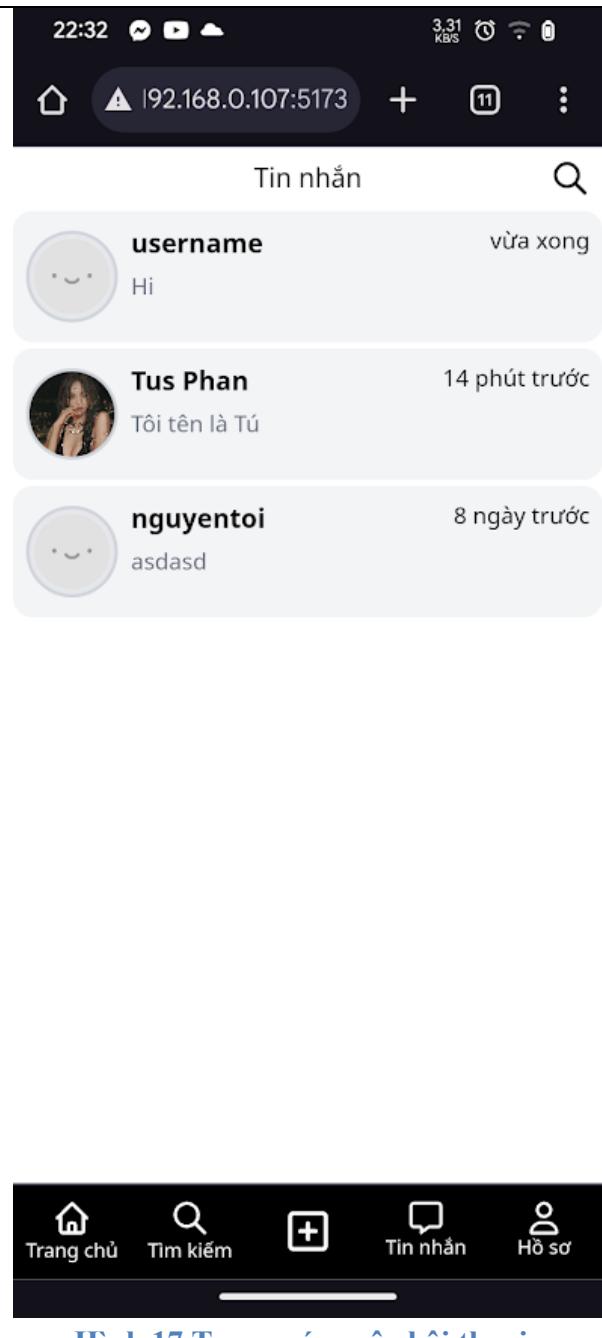




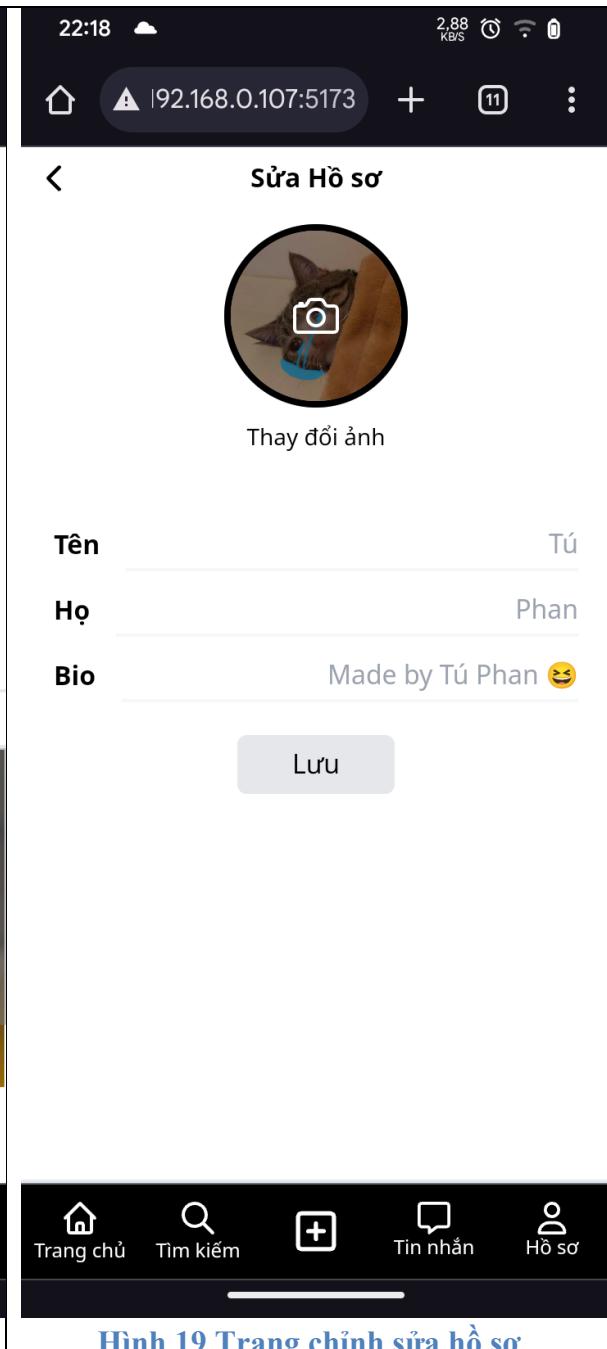
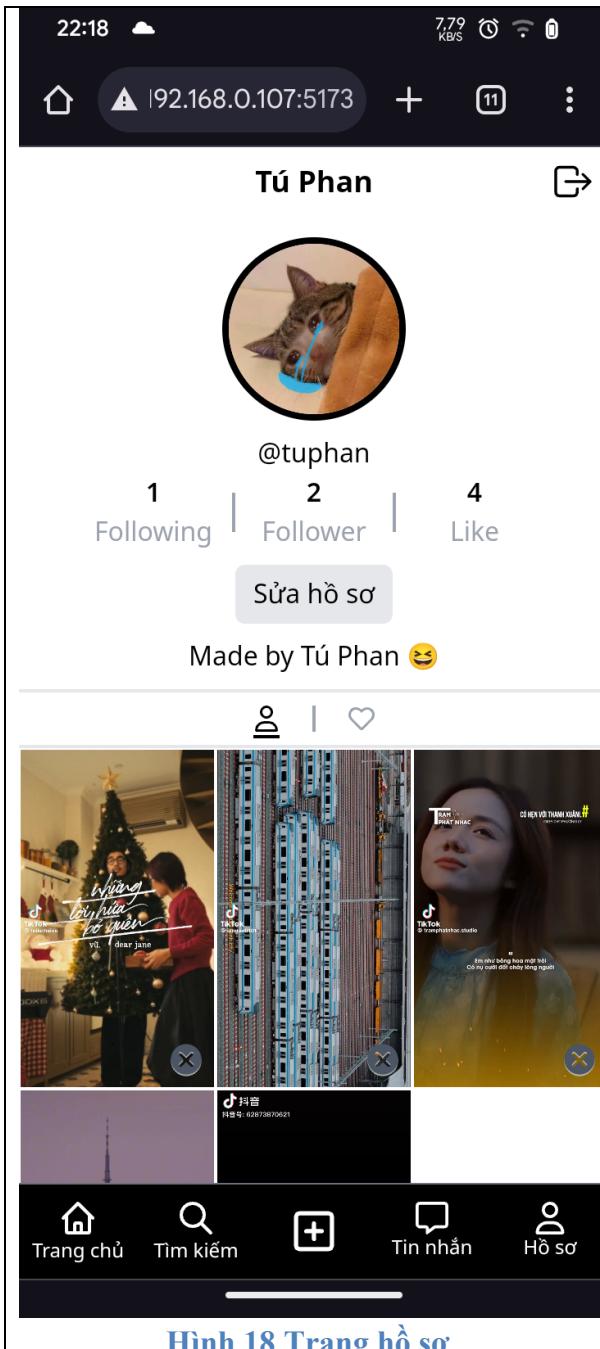




Hình 16 Trang nhắn tin



Hình 17 Trang các cuộc hội thoại



KẾT LUẬN

- **Những đóng góp của luận văn**

Qua quá trình tự tìm tòi và phát triển, luận văn đã hoàn thành trang web xem video ngắn trực tuyến với nhiều chức năng hữu ích, mang tính giải trí cao. Tuy vẫn còn nhiều điểm thiếu xót nhưng chung quy lại kết quả đạt được cũng khá là thành công. Cụ thể trang web cho phép đăng nhập, đăng ký, chỉnh sửa trang cá nhân, duyệt xem video, tải lên, chia sẻ, tìm kiếm, thích, bình luận, nhắn tin, chỉnh sửa trang cá nhân, quản lý video, quản lý theo dõi và nhiều tính năng khác

- **Những hạn chế còn tồn tại**

Trang web còn nhiều nhược điểm hạn chế cần khắc phục như:

- Hiệu năng chưa được tốt: đối với việc xử lý lượng lớn dữ liệu và yêu cầu từ người dùng đồng thời có khả năng gây quá tải hệ thống
 - Chưa bảo mật quá tốt: trang web tuy hỗ trợ tính năng bảo mật nhưng vẫn chưa đủ tốt vì còn lưu token tại localStorage, truyền token qua param để xác thực websocket, ...
 - Tính năng responsive chưa đủ tốt
- **Hướng phát triển tương lai**
 - Tối ưu về hiệu suất
 - Nâng cao tính bảo mật
 - Thiết kế giao diện đẹp hơn
 - Tăng khả năng chịu tải của server
 - Tách việc lưu trữ dữ liệu như video sang dịch vụ lưu trữ bên thứ 3 như cloud