

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống xem và chia sẻ video âm nhạc

HOÀNG TRUNG PHONG

phong.ht184171@sis.hust.edu.vn

Ngành Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: TS. Nguyễn Khánh Phương

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 8/2023

LỜI CẢM ƠN

Đầu tiên, em xin được gửi lời cảm ơn sâu sắc tới TS. Nguyễn Khánh Phương, giảng viên bộ môn Khoa học máy tính - Trường Công nghệ thông tin và Truyền thông - Đại học Bách khoa Hà Nội đã tận tình giúp đỡ, hướng dẫn và có những góp ý vô cùng quý báu cho em trong quá trình thực hiện đồ án tốt nghiệp. Sự chỉ bảo đầy tâm huyết từ cô đã giúp em có thêm nhiều kiến thức và động lực để hoàn thành tốt nhất đồ án tốt nghiệp của mình.

Em cũng xin được gửi lời cảm ơn tới toàn thể các thầy cô trong Đại học Bách khoa Hà Nội nói chung và Trường Công nghệ thông tin và Truyền thông nói riêng, đã tận tình chỉ dạy em trong suốt thời gian học tập. Những bài học bổ ích của thầy cô chính là hành trang vô giá, giúp ích cho em trong quá trình làm việc, phát triển sau này.

Cuối cùng, em xin được gửi lời cảm ơn tới gia đình, bạn bè, những người đã luôn bên cạnh, hỗ trợ và động viên em trong suốt những chặng đường học tập cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thời đại số hóa hiện nay, việc xem và chia sẻ video âm nhạc trên web đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Tuy nhiên, với số lượng lớn video âm nhạc có sẵn trên web, việc tìm kiếm và lựa chọn nội dung phù hợp với sở thích và gu âm nhạc của người dùng trở nên khó khăn. Hơn thế, trải nghiệm người dùng cũng là một yếu tố quan trọng. Người dùng luôn yêu cầu một giao diện thân thiện, tốc độ tải video nhanh chóng và khả năng phát lại mượt. Đây là những yếu tố quan trọng để tạo ra một trải nghiệm tuyệt vời cho người dùng. Đồng thời, tính năng tương tác và chia sẻ nội dung cũng cần được xem xét để tạo sự kết nối và tương tác giữa người dùng và ca sĩ. Vì vậy, mục tiêu của đồ án là tạo ra ứng dụng giải quyết được vấn đề trên, giúp người dùng tìm ra được những video âm nhạc phù hợp nhất với sở thích nghe nhạc và có trải nghiệm nghe nhạc liền mạch. Ngoài ra, ứng dụng còn cung cấp tính năng cho ca sĩ có thể quản lý chất lượng video và người đăng ký thông qua các phân tích thống kê. Sản phẩm của đồ án được xây dựng dựa trên công nghệ Ruby với framework Ruby on Rails và VueJS.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	6
2.2.1 Biểu đồ use case tổng quát	6
2.2.2 Biểu đồ usecase phân rã Quản lý bình luận.....	7
2.2.3 Biểu đồ use case phân rã Tương tác với video	7
2.2.4 Biểu đồ use case phân rã Quản lý video	8
2.2.5 Biểu đồ use case phân rã Quản lý kênh.....	8
2.2.6 Biểu đồ use case phân rã Quản lý playlist	9
2.3 Đặc tả chức năng	10
2.3.1 Đặc tả use case Đăng tải video	10
2.3.2 Đặc tả use case Bình luận video	11
2.3.3 Đặc tả use case Tạo playlist	12
2.3.4 Đặc tả use case Tìm kiếm video	13
2.3.5 Đặc tả use case Đặt lịch đăng tải video	14
2.3.6 Đặc tả use case Xem video xem sau	15
2.4 Yêu cầu phi chức năng	16
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	17
3.1 Ruby on Rails.....	17

3.2 VueJS	17
3.3 Vuetify	18
3.4 MySQL	19
3.5 Amazon S3	19
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ	21
4.1 Thiết kế kiến trúc.....	21
4.1.1 Lựa chọn kiến trúc phần mềm	21
4.1.2 Thiết kế tổng quan.....	23
4.1.3 Thiết kế chi tiết gói	24
4.2 Thiết kế chi tiết.....	25
4.2.1 Thiết kế giao diện	25
4.2.2 Thiết kế lớp	28
4.2.3 Thiết kế cơ sở dữ liệu	35
4.3 Xây dựng ứng dụng.....	46
4.3.1 Thư viện và công cụ sử dụng.....	46
4.3.2 Kết quả đạt được	47
4.3.3 Minh họa các chức năng chính	47
4.4 Kiểm thử.....	58
4.4.1 Kiểm thử chức năng đăng tải video	58
4.4.2 Kiểm thử chức năng Tạo danh sách phát.....	59
4.4.3 Kiểm thử chức năng tìm kiếm video và ca sĩ.....	60
4.5 Triển khai	60
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	62
5.1 Sử dụng kiến trúc 3 lớp kết hợp với Single Page Application	62
5.1.1 Đặt vấn đề	62
5.1.2 Giải pháp và kết quả đạt được.....	62

5.2 Phát triển chức năng đặt lịch phát hành video cho ca sĩ	63
5.2.1 Đặt vấn đề	63
5.2.2 Giải pháp và kết quả đạt được.....	63
5.3 Phương pháp tính toán mức độ thịnh hành của video.....	66
5.3.1 Đặt vấn đề	66
5.3.2 Giải pháp và kết quả đạt được.....	66
5.4 Cải thiện chức năng cập nhật lượt xem của video.....	67
5.4.1 Đặt vấn đề	67
5.4.2 Giải pháp và kết quả đạt được.....	67
5.5 Xây dựng hệ thống các chức năng gợi ý video cho người dùng.....	69
5.5.1 Đặt vấn đề	69
5.5.2 Giải pháp và kết quả đạt được.....	69
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	71
6.1 Kết luận	71
6.2 Hướng phát triển.....	71
TÀI LIỆU THAM KHẢO.....	72

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát	6
Hình 2.2	Biểu đồ use case phân rã Quản lý bình luận	7
Hình 2.3	Biểu đồ use case phân rã Tương tác với video	7
Hình 2.4	Biểu đồ use case phân rã Quản lý video	8
Hình 2.5	Biểu đồ use case phân rã Quản lý kênh	9
Hình 2.6	Biểu đồ use case phân rã Quản lý playlist	9
Hình 3.1	Ruby on Rails	17
Hình 3.2	VueJS	18
Hình 3.3	Vuetify	18
Hình 3.4	MySQL	19
Hình 3.5	Amazon S3	20
Hình 4.1	Kiến trúc ba lớp (Three-layer)	22
Hình 4.2	Biểu đồ thiết kế tổng quan của hệ thống	23
Hình 4.3	Biểu đồ chi tiết gói chức năng xem, đặt lịch, chỉnh sửa và xóa video	25
Hình 4.4	Thiết kế giao diện danh sách video	26
Hình 4.5	Thiết kế giao diện xem video	27
Hình 4.6	Thiết kế giao diện xem danh sách phát	27
Hình 4.7	Thiết kế giao diện thông kê cho ca sĩ	28
Hình 4.8	Lớp VideosController	28
Hình 4.9	Lớp VideoService	30
Hình 4.10	Lớp UploadVideoWorker	31
Hình 4.11	Biểu đồ trình tự ca sĩ tạo và đặt lịch phát video	32
Hình 4.12	Biểu đồ trình tự ca sĩ cập nhật thông tin video	33
Hình 4.13	Biểu đồ trình tự ca sĩ xóa video	34
Hình 4.14	Sơ đồ thực thể liên kết	35
Hình 4.15	Thiết kế chi tiết các bảng	36
Hình 4.16	Màn hình trang chủ của người dùng	48
Hình 4.17	Màn hình danh sách video xem sau	48
Hình 4.18	Màn hình danh sách video đã thích	49
Hình 4.19	Màn hình lịch sử video đã xem	49
Hình 4.20	Màn hình lịch sử tìm kiếm	50
Hình 4.21	Màn hình danh sách video của ca sĩ đã đăng ký	50
Hình 4.22	Màn hình danh sách video thịnh hành	51

Hình 4.23 Màn hình danh sách video playlist video	51
Hình 4.24 Màn hình xem video	52
Hình 4.25 Màn hình xem video từ danh sách phát	52
Hình 4.26 Màn hình kênh của ca sĩ	53
Hình 4.27 Màn hình cài đặt thông tin cá nhân	53
Hình 4.28 Màn hình thống kê lượt xem	54
Hình 4.29 Màn hình phân tích người xem	54
Hình 4.30 Màn hình phân tích chất lượng nội dung video	55
Hình 4.31 Màn hình danh sách người đăng ký	55
Hình 4.32 Màn hình danh sách tải video	56
Hình 4.33 Màn hình danh sách video	56
Hình 4.34 Màn hình xem chi tiết và chỉnh sửa video	57
Hình 4.35 Màn hình danh sách bình luận	57
Hình 4.36 Màn hình cài đặt thông tin ca sĩ	58
Hình 5.1 Kiến trúc hệ thống sử dụng background job	63
Hình 5.2 Giao diện quản lý và giám sát trạng thái các job	64
Hình 5.3 Quá trình lưu job vào hằng đợi	65
Hình 5.4 Quá trình lấy job từ hằng đợi để xử lý	65
Hình 5.5 Quy trình cập nhật lượt xem thông thường	68
Hình 5.6 Quy trình cập nhật lượt xem khi sử dụng event-driven job	68
Hình 5.7 Danh sách video và playlist hệ thống gợi ý đến người dùng	70

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đánh giá ưu nhược điểm của Youtube và Spotify	4
Bảng 2.2	Đặc tả use case Đăng tải video	10
Bảng 2.4	Đặc tả use case Bình luận video	11
Bảng 2.5	Đặc tả use case Tạo playlist	12
Bảng 2.6	Đặc tả use case Tìm kiếm video	14
Bảng 2.7	Đặc tả use case Đặt lịch đăng tải video	14
Bảng 2.8	Đặc tả use case Xem video xem sau	15
Bảng 4.1	Bảng user	37
Bảng 4.2	Bảng video	37
Bảng 4.3	Bảng category	38
Bảng 4.4	Bảng singer	38
Bảng 4.5	Bảng comment	39
Bảng 4.6	Bảng feeling	39
Bảng 4.7	Bảng hash_tag	39
Bảng 4.8	Bảng history	40
Bảng 4.9	Bảng member_notification	40
Bảng 4.10	Bảng own_playlist_video	41
Bảng 4.11	Bảng own_playlist	41
Bảng 4.12	Bảng playlist_video	42
Bảng 4.13	Bảng playlist	42
Bảng 4.14	Bảng reply	42
Bảng 4.15	Bảng singer_reply	43
Bảng 4.16	Bảng singer_notification	43
Bảng 4.17	Bảng subscribe	44
Bảng 4.18	Bảng user_playlist	44
Bảng 4.19	Bảng video_hash_tag	45
Bảng 4.20	Bảng watch_later_video	45
Bảng 4.21	Bảng playlist_by_topic	45
Bảng 4.22	Bảng playlist_by_topic_video	46
Bảng 4.23	Danh sách thư viện và công cụ sử dụng	46
Bảng 4.24	Thống kê một số thông tin về ứng dụng	47
Bảng 4.25	Các trường hợp kiểm thử cho chức năng Đăng tải video	58
Bảng 4.26	Các trường hợp kiểm thử cho chức năng Tạo danh sách phát .	59

Bảng 4.27 Các trường hợp kiểm thử cho chức năng Tìm kiếm video và ca sĩ	60
Bảng 4.28 Chi tiết về môi trường triển khai ứng dụng	61

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CSDL	Cơ sở dữ liệu
CSS	Ngôn ngữ định kiểu theo tầng (Cascading Style Sheets)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
MVC	Mẫu kiến trúc phần mềm (Model-View-Controller)
ORM	Ánh xạ quan hệ đối tượng (Object-Relational Mapping)
SPA	Ứng dụng một trang (Single Page Application)
UI	Giao diện người dùng (User Interface)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Âm nhạc đóng vai trò vô cùng quan trọng trong cuộc sống của chúng ta. Nó không chỉ là một loại hình giải trí mà còn là một phương tiện biểu đạt cảm xúc và tạo nên sự kết nối giữa các con người. Âm nhạc còn có vai trò thúc đẩy sáng tạo và tự do biểu đạt. Nó là nguồn cảm hứng để tạo ra những tác phẩm mới, giai điệu và lời ca sáng tạo. Âm nhạc không giới hạn và cho phép mọi người tự do biểu đạt bản thân và khám phá sự sáng tạo của mình.

G&Me-một trang web chuyên khảo sát, nghiên cứu thị trường tại Việt Nam đã thực hiện một bài khảo sát về nhu cầu nghe nhạc của người Việt Nam trên 1500 người trong độ tuổi từ 18 - 44. Kết quả bài khảo sát chỉ ra rằng: 75% số người khảo sát nghe nhạc hàng ngày và gần 97% số người khảo sát nghe nhạc hàng tuần. Số liệu cho thấy được nhu cầu nghe nhạc hàng ngày, hàng tuần là rất lớn. Và xu hướng nghe nhạc trực tuyến là một trong những xu hướng nghe nhạc phát triển mạnh và đầy tiềm năng với sự phát triển của công nghệ như hiện nay.

Một trong những ứng dụng nghe nhạc và xem video trực tuyến lớn nhất hiện nay là Youtube và Spotify. YouTube là một nền tảng chia sẻ video trực tuyến được ra mắt vào năm 2005. Nó cho phép người dùng tải lên, xem và chia sẻ video miễn phí trên mạng. Với hàng tỷ video có sẵn trên nền tảng này, YouTube đã trở thành một trong những trang web phổ biến nhất trên Internet. Ưu điểm lớn nhất của Youtube là sự đa dạng nội dung. YouTube cung cấp một loạt các nội dung đa dạng, bao gồm âm nhạc, phim ảnh, video hài, tin tức, hướng dẫn, vlog và nhiều thể loại giải trí khác. Người dùng có thể tìm thấy hầu hết mọi thứ mà họ quan tâm và thưởng thức nội dung theo sở thích cá nhân. Vì nội dung quá đa dạng nên Youtube chưa thực sự chuyên sâu dưới vai trò là một ứng dụng nghe nhạc trực tuyến vì còn hạn chế ở chức năng gọi ý nghe nhạc cho người dùng. Spotify là một dịch vụ nghe nhạc trực tuyến được ra mắt vào năm 2008. Nó cho phép người dùng truy cập và nghe hàng triệu bài hát từ các nghệ sĩ khác nhau trên mạng. Spotify có một ứng dụng di động và máy tính để bàn, cho phép người dùng nghe nhạc mọi lúc, mọi nơi. Spotify cũng sở hữu thư viện âm nhạc rộng lớn và cung cấp các tính năng gợi ý và phân loại thông minh, giúp người dùng khám phá và khám phá những bài hát mới dựa trên sở thích cá nhân. Nó sử dụng thuật toán để đề xuất nhạc mà người dùng có thể thích dựa trên lịch sử nghe nhạc và sở thích cá nhân. Tuy nhiên, hạn chế của Spotify là chỉ mới hỗ trợ nghe nhạc trực tuyến và chưa có phần video.

Từ những nhu cầu thực tiễn và ưu nhược điểm của một số ứng dụng được nêu ra

ở trên, yêu cầu đặt ra là phát triển một hệ thống chia sẻ, xem video âm nhạc.

1.2 Mục tiêu và phạm vi đề tài

Sản phẩm của đồ án là một phần mềm dạng thức web, cung cấp các tính năng tiện ích cho người dùng và ca sĩ: Ca sĩ tạo và tải lên những video âm nhạc; ca sĩ quản lý video, quản lý kênh, quản lý bình luận; người dùng có thể xem video và tương tác với video bằng cách react và bình luận; người dùng có thể tương tác với ca sĩ bằng cách đăng ký kênh của họ. Bằng các chức năng của hệ thống, ứng dụng có thể gợi ý ra các video phù hợp với nhu cầu và sở thích của người dùng, tạo ra các trải nghiệm nghe nhạc cho họ, giải quyết được những vấn đề của hai ứng dụng trên.

1.3 Định hướng giải pháp

Để đáp ứng các yêu cầu đặt ra và tạo thuận lợi cho việc tiếp cận của người dùng, hệ thống sẽ được xây dựng trên nền tảng web. Ngoài ra, để cải thiện trải nghiệm người dùng khi sử dụng hệ thống trên nền web, hệ thống sẽ được xây dựng theo hướng Single Page Application (SPA). Khi xảy ra việc chuyển trang hay thay đổi dữ liệu hiển thị, SPA sẽ chỉ cập nhật lại phần nội dung bị thay đổi, mà không cần gọi đến server để tải lại toàn bộ trang, kể cả những phần không cần thiết. Điều này giúp giảm bớt gánh nặng cho server, đồng thời làm giảm đáng kể thời gian phản hồi tới người dùng, từ đó cải thiện trải nghiệm người dùng đối với hệ thống. Từ những định hướng trên, hệ thống sẽ được xây dựng dựa trên các công nghệ VueJS và Ruby on Rails cùng với hệ quản trị CSDL MySQL. Phần frontend của hệ thống sẽ sử dụng VueJS, một thư viện JavaScript được dùng trong việc phát triển giao diện người dùng, cho phép xây dựng các giao diện theo hướng Single Page Application và dựa trên các component. Phần backend của hệ thống sẽ sử dụng Ruby on Rails. Nó là một Framework cho phép phát triển ứng dụng Web được base dựa trên ngôn ngữ lập trình Ruby. Ruby là một ngôn ngữ lập trình mã nguồn mở, linh hoạt, với một sự nổi bật về sự đơn giản dễ dùng và hữu ích. Nó có cú pháp rõ ràng, tự nhiên dễ đọc và dễ dàng để viết. Các màn hình frontend sẽ tương tác kết nối với backend thông qua các RESTful API [7]. Ngoài ra, hệ thống cũng sử dụng thư viện Vuetify để tận dụng các UI component đã được xây dựng sẵn, giúp tăng tốc độ phát triển, tăng tính thẩm mỹ và tính nhất quán trong việc phát triển giao diện.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2: Khảo sát bài toán và yêu cầu hệ thống. Chương này tập trung vào việc khảo sát bài toán và xác định các yêu cầu đặt ra cho hệ thống. Khảo sát này bao gồm việc nghiên cứu về tình hình hiện tại, các vấn đề cần giải quyết, và nhu

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

cầu của người dùng. Dựa trên những khảo sát này, chương này sẽ phân tích và đưa ra những chức năng cần thiết của hệ thống, đồng thời xây dựng quy trình nghiệp vụ chính để đáp ứng các yêu cầu đó.

Chương 3: Công nghệ sử dụng. Chương này giới thiệu chi tiết về các công nghệ được sử dụng trong quá trình phát triển hệ thống.

Chương 4: Kết quả thực nghiệm. Chương này trình bày chi tiết về thiết kế hệ thống dựa trên yêu cầu chức năng, quy trình nghiệp vụ và các công nghệ đã được trình bày trong chương trước. Chương này cũng đưa ra kết luận về những kết quả đạt được sau quá trình phát triển, kiểm thử và triển khai hệ thống.

Chương 5: Giải pháp đóng góp. Chương này trình bày những đóng góp nổi bật của bản thân trong suốt quá trình thực hiện ĐATN cũng như những khó khăn gặp phải; cách giải quyết, giải pháp đã áp dụng trong hệ thống.

Chương 6: Kết luận. Chương này kết luận nội dung đồ án, đồng thời đưa ra các đề xuất, hướng phát triển trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Hiện nay, với nhu cầu xem, nghe nhạc trực tuyến ngày càng cao, các ca sĩ và người nghe nhạc luôn mong muốn tìm được một phần mềm chia sẻ video âm nhạc đến với họ để thỏa mãn được nhu cầu giải trí. Họ mong muốn có một ứng dụng giúp họ:

- Quản lý các video đăng tải, thống kê lượt view và bình luận
- Thống kê giúp ca sĩ các phân tích kỹ thuật về lượt xem, chất lượng người dùng, chất lượng video
- Gợi ý các video và danh sách phù hợp với nhu cầu, sở thích nghe nhạc của người dùng
- Quản lý video yêu thích, playlist yêu thích
- Cho phép người dùng tìm kiếm ca sĩ, video âm nhạc
- Cho phép người dùng tương tác với video

Một số ứng dụng nghe, chia sẻ âm nhạc phổ biến, được sử dụng rộng rãi hiện nay phải kể đến Youtube và Spotify. YouTube là một nền tảng truyền thông xã hội được sử dụng để chia sẻ và xem video. Người dùng có thể tạo kênh cá nhân và tải lên các video mà họ muốn chia sẻ với mọi người. YouTube cũng cung cấp cho người dùng khả năng tìm kiếm, xem và tương tác với các video khác trong cộng đồng trên toàn thế giới. Spotify là một dịch vụ phát nhạc trực tuyến cho phép người dùng nghe nhạc từ hàng triệu bài hát, album và nghệ sĩ khác nhau. Người dùng có thể truy cập Spotify thông qua ứng dụng di động, trình duyệt web hoặc ứng dụng trên máy tính để bàn. Spotify cung cấp các tính năng như tìm kiếm nhạc theo thể loại, tạo danh sách phát cá nhân, chia sẻ nhạc với bạn bè và khám phá những bài hát mới. Bảng 2.1 đánh giá một số ưu nhược điểm của hai phần mềm này:

Bảng 2.1: Đánh giá ưu nhược điểm của Youtube và Spotify

	Youtube	Spotify
Nhược điểm	<ul style="list-style-type: none">Chức năng gợi ý chưa đưa ra được nhiều playlist phù hợp với người dùngChức năng khám phá âm nhạc chưa thật sự tốt	<ul style="list-style-type: none">Chỉ mới cung cấp dịch vụ nghe nhạc, chưa hỗ trợ videoChưa phát triển các chức năng tương tác với bài hát như reaction, bình luận

Bảng 2.1 – Còn nữa ở trang tiếp

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Bảng 2.1 – Tiếp tục từ trang trước

Ưu điểm	
	<ul style="list-style-type: none">Sở hữu thư viện video khổng lồ đáp ứng nhu cầu của người nghe nhạcĐa nền tảng từ điện thoại Android, iOS, trình duyệt web cho đến TV thông minh và ô tôGiao diện trực quan và dễ sử dụngTrình phát nhạc cung cấp đầy đủ tính năng cần thiết như gợi ý ca khúc tiếp theo, mở Radio bài hát hay tải xuốngĐem lại trải nghiệm sử dụng dễ dàng và tối giản. Các bảng chọn được thiết kế rõ ràng, đơn giản và dễ dàng làm quen với những người mới.Trình phát nhạc bao gồm rất nhiều tính năng khác nhau, nổi bật trong số đó là chia sẻ đa thiết bị, hẹn giờ hay tạo playlistKhả năng gợi ý nhạc được đánh giá cao, ứng dụng có thể phân tích giai điệu hay thói quen nghe nhạc của người dùng, từ đó gợi ý các bài hát tiếp theo một cách tương đối chính xác.

Thông qua việc khảo sát các yêu cầu, mong muốn của người dùng và quá trình phân tích, đánh giá các tính năng, ưu điểm, nhược điểm của một số phần mềm hiện có, chúng ta đã xác định được các tính năng quan trọng, cần có của hệ thống. Hệ thống xem và chia sẻ video âm nhạc sẽ hướng tới phục vụ hai nhóm đối tượng là người dùng và ca sĩ, cụ thể như sau:

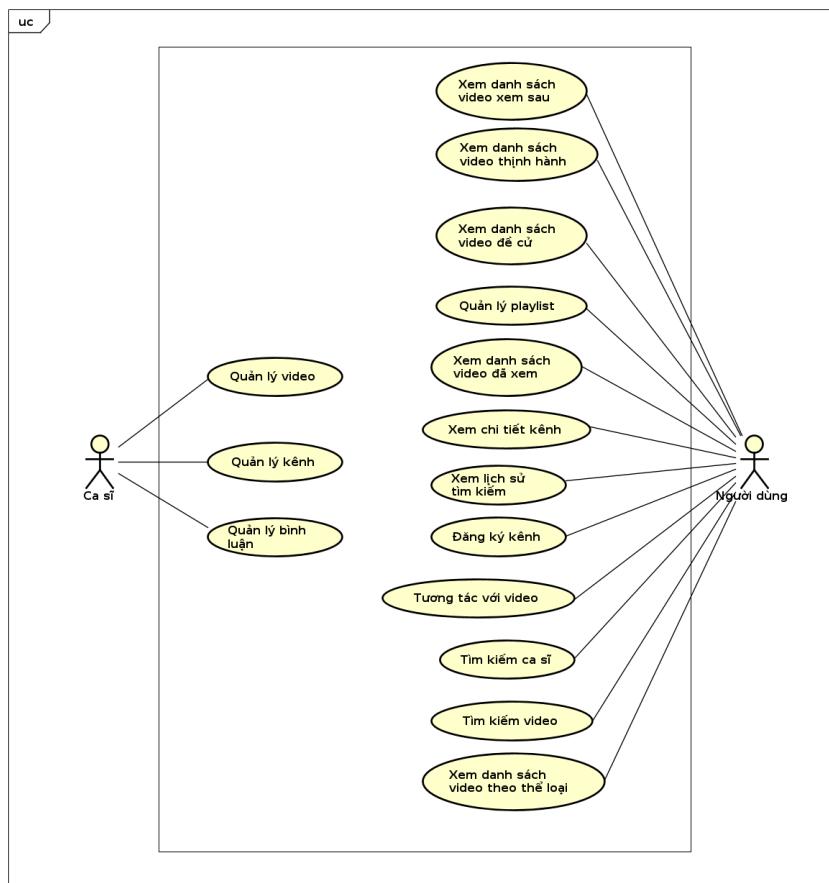
- Ca sĩ có thể thực hiện các chức năng liên quan đến việc:
 - Quản lý video âm nhạc: tạo, chỉnh sửa, xóa video, đặt lịch phát hành, quản lý views
 - Quản lý kênh, bình luận của người dùng
 - Quản lý người đăng ký: hiển thị danh sách người đăng ký
 - Xem các phân tích thống kê về lượt xem, người xem và chất lượng video
- Người dùng có thể thực hiện các chức năng liên quan đến việc:
 - Xem video: xem danh sách video thịnh hành, video xem sau, video đã like, lịch sử video
 - Tương tác với video: Xem chi tiết video, react bình luận video, lưu lại video, tải xuống video
 - Tìm kiếm video, tìm kiếm ca sĩ
 - Quản lý playlist: Tạo playlist, thêm, xóa video vào playlist, chỉnh sửa

thông tin playlist

- Xem danh sách playlist do hệ thống gợi ý

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát



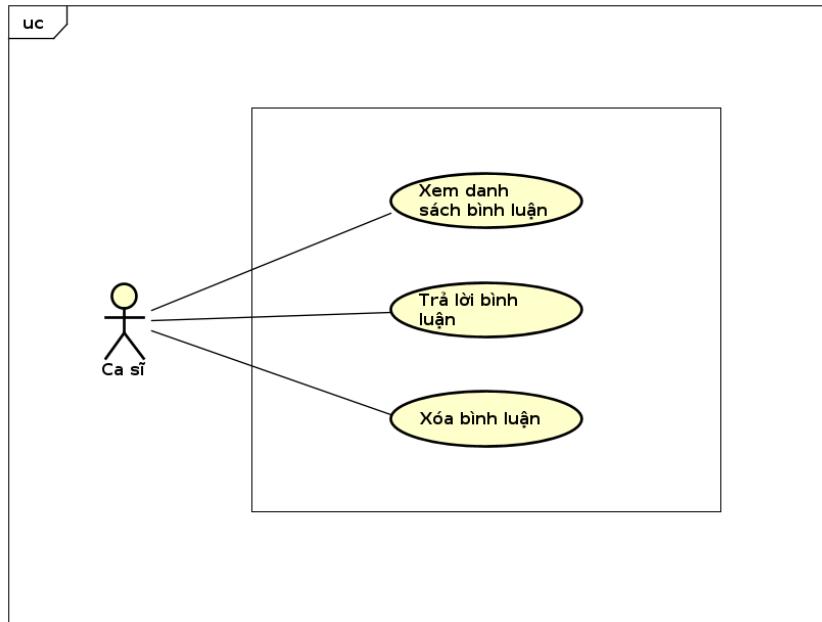
Hình 2.1: Biểu đồ use case tổng quát

Hình 2.1 là biểu đồ use case tổng quát của hệ thống. Hệ thống bao gồm hai tác nhân chính:

- Ca sĩ: Là người cung cấp dịch vụ nghe nhạc bằng cách tải lên những video âm nhạc để phục vụ nhu cầu giải trí nghe nhạc của người dùng. Mỗi ca sĩ có thể thực hiện các chức năng liên quan đến quản lý video đã đăng tải, quản lý kênh và người đăng ký. Ngoài ra, ca sĩ có thể tương tác với người dùng bằng cách trả lời bình luận của người dùng.
- Người dùng: Là người đăng ký vào hệ thống, xem và thưởng thức những video mà ca sĩ đăng tải. Mỗi người dùng có thể xem video ở những thể loại khác đa dạng khác nhau, xem những video do hệ thống gợi ý dựa trên nhu cầu, sở thích, lịch sử nghe nhạc của họ. Ngoài ra, người dùng còn có thể xem video theo danh sách phát, tương tác bình luận với dưới mỗi video, tìm kiếm video

theo tên, ...

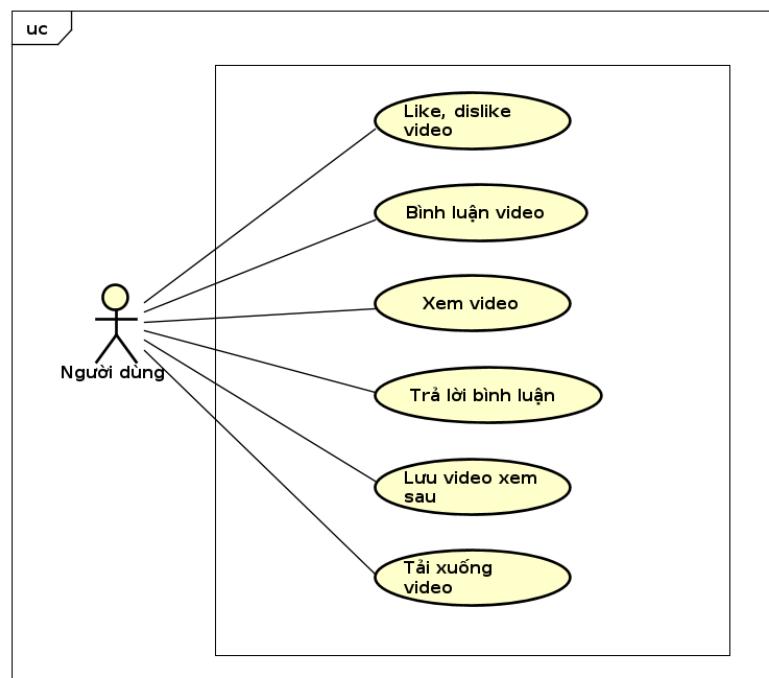
2.2.2 Biểu đồ usecase phân rã Quản lý bình luận



Hình 2.2: Biểu đồ use case phân rã Quản lý bình luận

Hình 2.2 là biểu đồ phân rã use case Quản lý bình luận. Ca sĩ có thể thực hiện các chức năng liên quan đến quản lý bình luận bao gồm xem danh sách bình luận; trả lời từng bình luận hoặc xóa bình luận của mỗi video do mình quản lý.

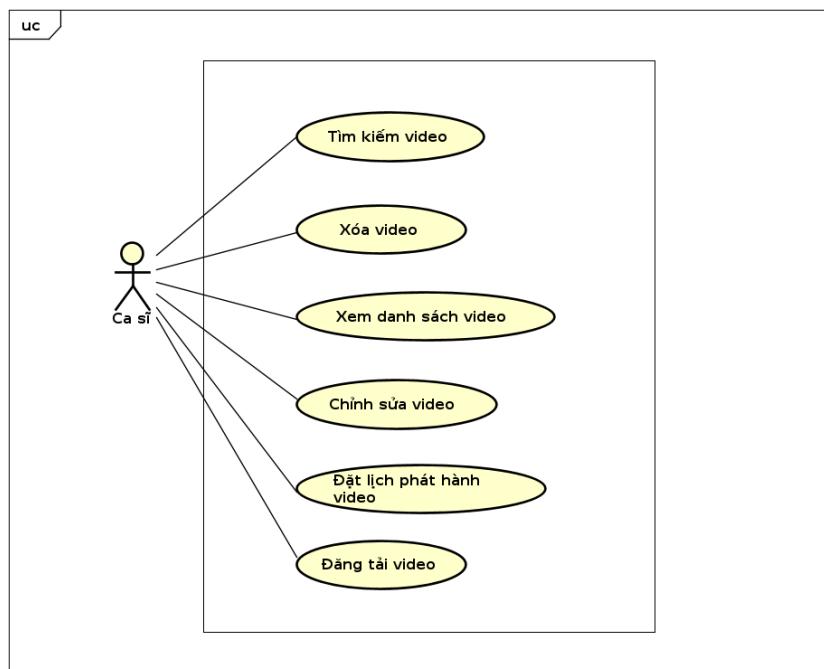
2.2.3 Biểu đồ use case phân rã Tương tác với video



Hình 2.3: Biểu đồ use case phân rã Tương tác với video

Hình 2.3 là biểu đồ phân rã use case Tương tác với video. Người dùng có thể thực hiện các chức năng liên quan đến việc tương tác với video như react like, dislike video; bình luận dưới mỗi video; trả lời các bình luận. Người dùng còn có thể lưu lại video yêu thích để xem sau, tải xuống video.

2.2.4 Biểu đồ use case phân rã Quản lý video

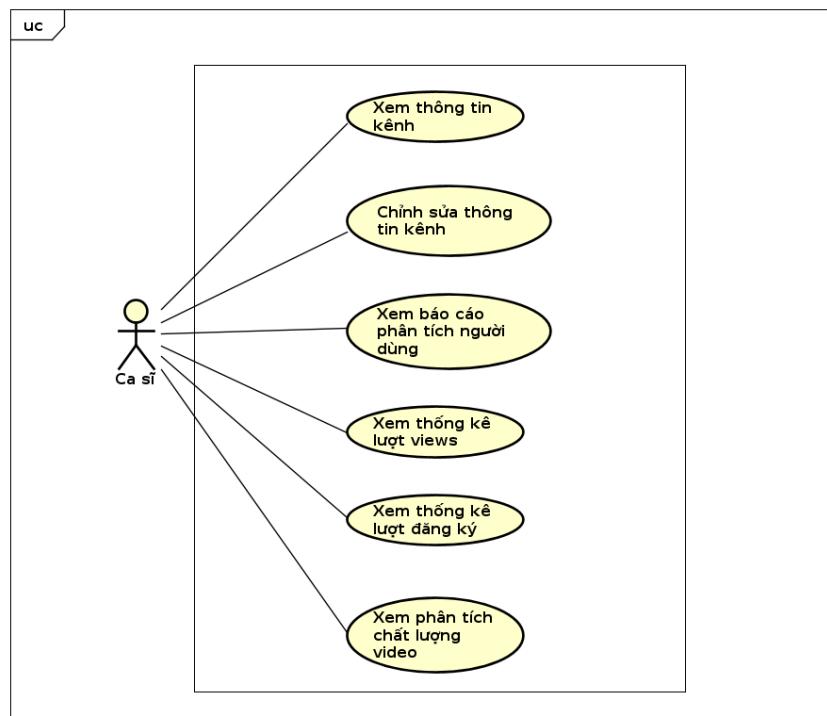


Hình 2.4: Biểu đồ use case phân rã Quản lý video

Hình 2.4 là biểu đồ phân rã use case Quản lý video. Ca sĩ có thể thực hiện các chức năng liên quan đến quản lý video bao gồm xem danh sách video; Đăng tải, xóa, chỉnh sửa video; Đặt lịch phát hành video; Tìm kiếm video,...

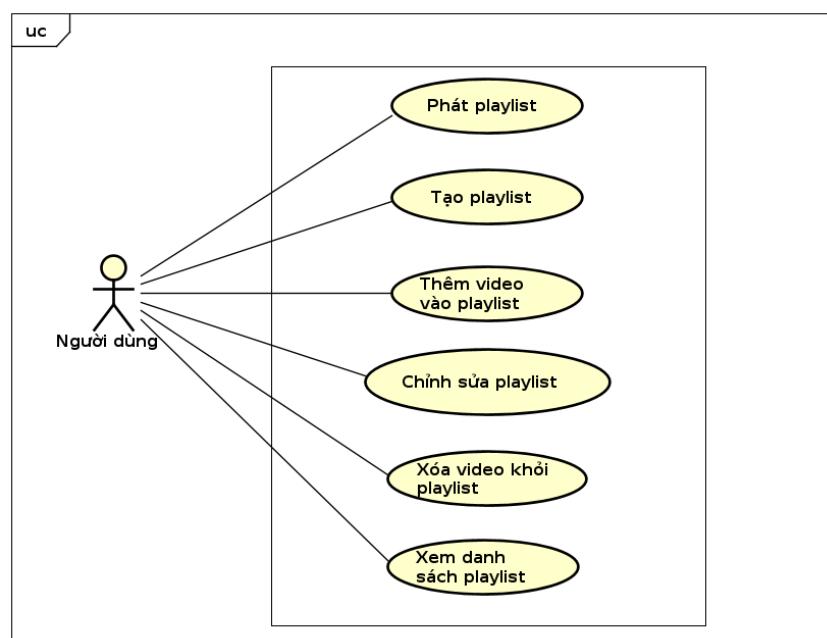
2.2.5 Biểu đồ use case phân rã Quản lý kênh

Hình 2.5 là biểu đồ phân rã use case Quản lý kênh. Ca sĩ có thể thực hiện các chức năng liên quan đến quản lý kênh bao gồm xem thông tin channel; chỉnh sửa thông tin kênh như mô tả kênh, tên kênh, ảnh đại diện,...; thống kê tương quan về kênh như thể loại, ngày tạo kênh,...; thống kê lượt views theo từng tuần, tháng, năm; thống kê lượt đăng ký.



Hình 2.5: Biểu đồ use case phân rã Quản lý kênh

2.2.6 Biểu đồ use case phân rã Quản lý playlist



Hình 2.6: Biểu đồ use case phân rã Quản lý playlist

Hình 2.6 là biểu đồ phân rã use case Quản lý playlist. Người dùng có thể thực hiện các chức năng liên quan đến quản lý playlist bao gồm xem danh sách playlist; tạo mới một playlist gồm những trường thông tin như tiêu đề, mô tả, trạng thái, video,...; chỉnh sửa thông tin playlist; thêm video vào playlist; xóa video khỏi

playlist.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Đăng tải video

Bảng 2.2: Đặc tả use case Đăng tải video

Mã use case	UC001		
Tên use case	Upload video		
Tác nhân	Ca sĩ		
Tiền điều kiện	Tác nhân đăng nhập thành công		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	Ca sĩ	chọn upload video từ màn hình trang quản lý
	2	Hệ thống	hiển thị giao diện upload video
	3	Ca sĩ	chọn file video để tải lên
	4	Hệ thống	kiểm tra xem ca sĩ đã tải lên đúng định dạng và kích thước file hay chưa
	5	Hệ thống	upload file thành công, hiển thị giao diện nhập thông tin video
	6	Ca sĩ	nhập các trường thông tin cho video và bấm submit
	7	Hệ thống	kiểm tra các trường thông tin đã đủ và hợp lệ hay chưa
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	thông báo người dùng cần nhập lại file video
	7a	Hệ thống	thông báo người dùng cần nhập trường thông tin còn thiếu hoặc chưa hợp lệ
Hậu điều kiện	Một video mới được cập nhật vào CSDL, video sẽ được xuất hiện tại trang chủ của người dùng nếu trạng thái lúc tạo của nó là public		

Dữ liệu đầu vào

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
Tiêu đề	Tiêu đề của video	Có	Độ dài lớn hơn 10 và không quá 100 ký tự	Psychic(Official Video) ft. Jack Harlow

Bảng 2.3 – Còn nữa ở trang tiếp

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Bảng 2.3 – Tiếp tục từ trang trước

Mô tả	Mô tả của video	Có	Độ dài lớn hơn 30 và không quá 800 ký tự	
Thể loại	Thể loại của video	Có		Pop
File video	Tệp video chứa các nội dung âm nhạc	Có	Tệp có định dạng mp4 và kích thước bé hơn 15MB	videomusic.mp4
Hashtag	Các chủ đề mà video liên quan	Có	Là một trong danh sách hashtag đã có trong hệ thống	Gym, Study,...
Trạng thái	Trạng thái của video	Có	Là một trong ba trạng thái đã có trong hệ thống	Public, private, schedule
File thumbnails	Tệp chứa hình ảnh thu nhỏ của video	Có	tệp có định dạng png	thumbnails-video1.png
Thời gian đặt lịch đăng	Chỉ có khi video được đặt lịch phát hành	Có	Định dạng YYYY-MM-DD HH:mm:ss và sau thời gian hiện tại	2023-10-01 09:00:00

2.3.2 Đặc tả use case Bình luận video

Bảng 2.4: Đặc tả use case Bình luận video

Mã use case	UC002
Tên use case	Comment video
Tác nhân	Người dùng
Tiền điều kiện	Tác nhân đăng nhập thành công

Bảng 2.4 – Còn nữa ở trang tiếp

Bảng 2.4 – Tiếp tục từ trang trước

	STT	Thực hiện bởi	Hành động
Luồng sự kiện chính (thành công)	1	Người dùng	chọn và bấm vào video từ màn hình danh sách video
	2	Hệ thống	hiển thị giao diện xem video
	3	Người dùng	tiến hành bình luận dưới video và bấm nút comment
	4	Hệ thống	kiểm tra độ dài chuỗi bình luận có vượt quá độ dài cho phép hay chưa
	5	Hệ thống	lưu vào cơ sở dữ liệu, thông báo thành công và hiển thị tại màn hình xem video
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	thông báo quá độ dài cho phép, và cho người dùng cần nhập lại chuỗi bình luận
Hậu điều kiện	Một bình luận mới được cập nhật vào CSDL, bình luận sẽ được hiển thị bên dưới mỗi video khi người dùng bấm vào xem video		

2.3.3 Đặc tả use case Tạo playlist

Bảng 2.5: Đặc tả use case Tạo playlist

Mã use case	UC003
Tên use case	Create playlist
Tác nhân	Người dùng
Tiền điều kiện	Tác nhân đăng nhập thành công

Bảng 2.5 – Còn nữa ở trang tiếp

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Bảng 2.5 – Tiếp tục từ trang trước

Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	chọn và bấm thêm video vào danh sách từ màn hình danh sách video
	2	Hệ thống	hiển thị giao diện thêm video vào danh sách
	3	Người dùng	chọn tạo mới playlist
	4	Hệ thống	hiển thị màn hình tạo mới playlist
	5	Người dùng	tiến hành nhập các trường thông tin của playlist
	6	Hệ thống	kiểm tra thông tin người dùng nhập đã đầy đủ và hợp lệ hay chưa
	7	Hệ thống	lưu playlist vào cơ sở dữ liệu, lưu video vào playlist vừa mới tạo
	8	Hệ thống	thông báo tạo playlist thành công đến người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	6a	Hệ thống	thông báo các trường thông tin chưa hợp lệ, và cho người dùng cần nhập lại
Hậu điều kiện	Một danh sách playlist mới được cập nhật vào CSDL, danh sách này được hiển thị bên thanh công cụ mỗi khi người dùng đăng nhập thành công		

Dữ liệu đầu vào

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Tiêu đề	Tiêu đề danh sách phát	Có	Có độ dài lớn hơn 3 và không quá 50 ký tự	Playlist for relaxing
2	Mô tả	Mô tả danh sách phát	Có	Có độ dài lớn hơn 3 và không quá 255 ký tự	Listening playlist when feel relax
3	Trạng thái	Trạng thái danh sách phát	Có	Là một trong 3 trạng thái public, private, unlisted	Public, Private, Unlisted
4	Video	Video của danh sách	Có	Là các video có trong thư viện video của hệ thống	

2.3.4 Đặc tả use case Tìm kiếm video

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Bảng 2.6: Đặc tả use case Tìm kiếm video

Mã use case	UC004		
Tên use case	Search video		
Tác nhân	Người dùng		
Tiền điều kiện	Không có		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	chọn và nhập tên video từ thanh tìm kiếm
	2	Hệ thống	lưu lại kết quả tìm kiếm dựa trên thông tin người dùng nhập
	3	Hệ thống	tìm kiếm video dựa trên thông tin người dùng nhập
	4	Hệ thống	hiển thị danh sách video tại trang kết quả tìm kiếm
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	1a	Người dùng	bấm vào biểu tượng ghi âm, ghi lại thông tin tìm kiếm bằng cách sử dụng search voice
	1b	Người dùng	chọn video hoặc ca sĩ từ danh sách lịch sử tìm kiếm
	3a	Hệ thống	Thông báo không tìm thấy video cần tìm kiếm
Hậu điều kiện	Lưu lại lịch sử tìm kiếm vào CSDL và hiển thị danh sách video tại trang kết quả nếu có video phù hợp		

2.3.5 Đặc tả use case Đặt lịch đăng tải video

Bảng 2.7: Đặc tả use case Đặt lịch đăng tải video

Mã use case	UC005
Tên use case	Schedule video
Tác nhân	Ca sĩ
Tiền điều kiện	Tác nhân đăng nhập thành công

Bảng 2.7 – Còn nữa ở trang tiếp

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Bảng 2.7 – Tiếp tục từ trang trước

Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	Ca sĩ	chọn upload video từ màn hình trang quản lý
	2	Hệ thống	hiển thị giao diện upload video
	3	Ca sĩ	chọn file video để tải lên
	4	Hệ thống	kiểm tra xem ca sĩ đã tải lên đúng định dạng và kích thước file hay chưa
	5	Hệ thống	upload file thành công, hiển thị giao diện nhập thông tin video
	6	Ca sĩ	nhập các trường thông tin cho video ,chọn status cho video là schedule
	7	Ca sĩ	nhập thời điểm muốn phát hành video
	9	Hệ thống	lưu video theo thông tin ca sĩ vừa nhập, hiển thị thông báo đặt lịch thành công
	9	Hệ thống	lưu video id và thời gian đặt lịch vào hằng đợi để tiến hành xử lý background job
	10	Hệ thống	đến thời gian phát hành, tiến hành chạy hàm trong hằng đợi, cập nhật trạng thái của video thành public
	11	Hệ thống	thông báo thông tin video vừa phát hành đến người dùng đăng ký kênh
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	thông báo người dùng cần nhập lại file video
	6a	Hệ thống	thông báo ca sĩ cần nhập trường thông tin còn thiếu
	7a	Hệ thống	thông báo ca sĩ cần nhập thời gian phát hành lớn hơn hiện tại
Hậu điều kiện	Một video mới được cập nhật vào CSDL, video có trạng thái private cho đến thời điểm phát hành sẽ cập nhật thành public		

2.3.6 ĐẶC TẢ USE CASE XEM VIDEO XEM SAU

Bảng 2.8: Đặc tả use case Xem video xem sau

Mã use case	UC006
Tên use case	Watch later video
Tác nhân	Người dùng
Tiền điều kiện	Tác nhân đăng nhập thành công

Bảng 2.8 – Còn nữa ở trang tiếp

Bảng 2.8 – Tiếp tục từ trang trước

Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	chọn xem video sau tại thanh công cụ
	2	Hệ thống	hiển thị màn hình danh sách video sau
	3	Người dùng	bấm vào video muốn xem từ danh sách video
	4	Hệ thống	chuyển hướng đến trang xem video
	5	Hệ thống	lưu lại lịch sử xem video của người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Người dùng	bấm vào nút phát tất cả video trong danh sách xem sau
Hậu điều kiện	Lưu lại lịch sử xem vào CSDL		

2.4 Yêu cầu phi chức năng

Ứng dụng web nghe, xem và chia sẻ video âm nhạc cần đảm bảo các yêu cầu phi chức năng sau:

- **Hiệu năng:** Hệ thống có thể phục vụ 200 người dùng đồng thời. Thời gian đáp ứng tối đa của hệ thống là 3s ở thời điểm bình thường và 6s lúc cao điểm
- **Tính khả dụng:** Giao diện đơn giản, thao tác dễ dàng, cho phép người dùng dễ dàng học và sử dụng hệ thống
- **Tính bảo mật:** Dữ liệu cần được xác thực và bảo mật
- **Tính bảo trì:** Hệ thống cần phải được xây dựng sao cho dễ dàng bảo trì, mở rộng quy mô và chức năng mới

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Ruby on Rails

Ruby on Rails[1], thường được gọi là Rails, là một framework phát triển ứng dụng web được viết bằng ngôn ngữ lập trình Ruby. Nó cung cấp một cách tiếp cận mô hình MVC để xây dựng và triển khai ứng dụng web một cách nhanh chóng và hiệu quả.



Hình 3.1: Ruby on Rails

Rails tuân thủ các nguyên tắc thiết kế và chuẩn mực tốt nhất trong cộng đồng phát triển phần mềm. Nó áp dụng các quy ước và quy tắc đặt tên để giữ cho mã nguồn dễ đọc, dễ hiểu và dễ bảo trì. Điều này giúp giảm thiểu sự phức tạp và tăng tính nhất quán trong việc phát triển ứng dụng. Hơn nữa, Rails tích hợp tốt với các công nghệ và công cụ phổ biến khác như JavaScript, HTML, CSS và các thư viện JavaScript như jQuery và Bootstrap. Điều này giúp người phát triển xây dựng giao diện người dùng phong phú và tương tác mạnh mẽ. Ngoài ra, Rails được thiết kế để tăng tốc độ phát triển ứng dụng. Nó cung cấp các công cụ và thư viện giúp giảm thiểu việc lặp lại mã và tăng cường hiệu suất phát triển. Cú pháp đơn giản và dễ hiểu của Ruby cũng giúp làm giảm thời gian và công sức cần thiết để viết mã. Rails cung cấp khả năng linh hoạt và mở rộng, cho phép bạn tùy chỉnh và mở rộng ứng dụng theo nhu cầu cụ thể. Nó hỗ trợ các công nghệ như RESTful API, routing tự động, và hỗ trợ ORM.

3.2 VueJS

Vue.js [2] là một framework linh động dùng để xây dựng giao diện người dùng. Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện, người dùng chỉ cần dùng thư viện lõi của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn.

Vue sở hữu một hệ sinh thái vững chắc nên có thể cung cấp một số add-ons rất

**Hình 3.2:** VueJS

hữu ích cho việc xây dựng một ứng dụng frontend điển hình nhất. Nó có thể bao gồm: vue-router, vuex, vue-test-utils, vue-dev-tools, vue-cli,... Core Vue sở hữu tính năng tối thiểu bởi khả năng tập trung vào việc render giao diện cho người dùng và các tương tác của nó. Chính vì vậy, nó sẽ cung cấp tối thiểu những tính năng cần thiết cho việc thiết kế và xây dựng kiến trúc. Nó có thể loại bỏ được các tính năng không cần thiết ra khỏi thư viện trong lõi Vue.js và đảm bảo cho framework nhỏ gọn và mềm dẻo hơn.

3.3 Vuetify

Vuetify[3] là một framework UI dựa trên Vue.js, được thiết kế để phát triển ứng dụng web đẹp và có trải nghiệm người dùng tốt. Vuetify cung cấp một tập hợp các thành phần giao diện và các hệ thống tạo giao diện giúp xây dựng ứng dụng web một cách nhanh chóng và dễ dàng.

**Hình 3.3:** Vuetify

Vuetify cung cấp các thành phần giao diện được thiết kế theo nguyên tắc Material Design của Google. Điều này đảm bảo rằng các ứng dụng sử dụng Vuetify sẽ có giao diện đẹp mắt, hiện đại và tuân thủ các tiêu chuẩn thiết kế. Vuetify được xây dựng trên Vue.js, một framework JavaScript mạnh mẽ cho phát triển ứng dụng web. Vì vậy, việc tích hợp Vuetify vào dự án Vue.js hiện có là rất dễ dàng và không tạo ra những xung đột hoặc khó khăn. Ngoài ra, Vuetify cung cấp một bộ sưu tập

lớn các thành phần giao diện như nút, thanh điều hướng, bảng, biểu đồ, hộp thoại và nhiều hơn nữa. Ngoài ra, nó cũng cung cấp một hệ thống tạo giao diện mạnh mẽ để giúp bạn tạo bố cục cho trang web của mình một cách dễ dàng và linh hoạt. Cuối cùng, Vuetify có thể tích hợp dễ dàng với các công nghệ khác như Vuex (quản lý trạng thái ứng dụng), Vue Router (định tuyến trang) và Axios (gửi yêu cầu HTTP). Điều này giúp xây dựng các ứng dụng web phức tạp và có khả năng mở rộng.

3.4 MySQL

MySQL[4] là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu theo hình thức quản lý các mối liên hệ giữa chúng.



Hình 3.4: MySQL

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được coi là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyen, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

3.5 Amazon S3

Amazon S3 hay Amazon Simple Storage Service[5] là một dịch vụ lưu trữ đối tượng trên đám mây được cung cấp bởi Amazon Web Service. Dịch vụ Amazon S3 mang đến cho người dùng hệ thống lưu trữ có thể thay đổi theo quy mô, tính khả dụng dữ liệu, cùng với đó là độ bảo mật đi kèm hiệu năng hoạt động hàng đầu hiện

nay.



Hình 3.5: Amazon S3

Người dùng có thể tạo và đặt tên cho bucket chứa dữ liệu trong Amazon S3. Dữ liệu được lưu trữ đa dạng thể loại với phạm vi không giới hạn. Với mỗi dữ liệu được lưu trữ sẽ là một object và người dùng có thể tải dữ liệu lên tùy thích với dung lượng chứa tối đa 5TB. Bên cạnh việc tải dữ liệu lên bucket để lưu trữ, người dùng còn có thể tải dữ liệu xuống thiết bị bất cứ lúc nào. Hoặc chia sẻ dữ liệu cho những người dùng khác nằm trong Amazon S3 bucket cùng tải dữ liệu về. Trong Amazon S3 còn cung cấp khả năng mã hóa dữ liệu mà người dùng lưu trữ theo 2 cách là Mã hóa ở client side hoặc Mã hóa ở server side. Trong trường hợp dữ liệu hỏng, hệ thống vẫn lưu trữ lại nhiều bản sao hỗ trợ khôi phục lại. Đồng thời, mỗi bản chỉnh sửa cũng đều được lưu trữ lại để sử dụng mỗi khi cần thiết. Trong phạm vi đồ án, ứng dụng sử dụng dịch vụ Amazon S3 kết hợp chức năng active storage trong Rails để xử lý lưu trữ video và hình ảnh.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

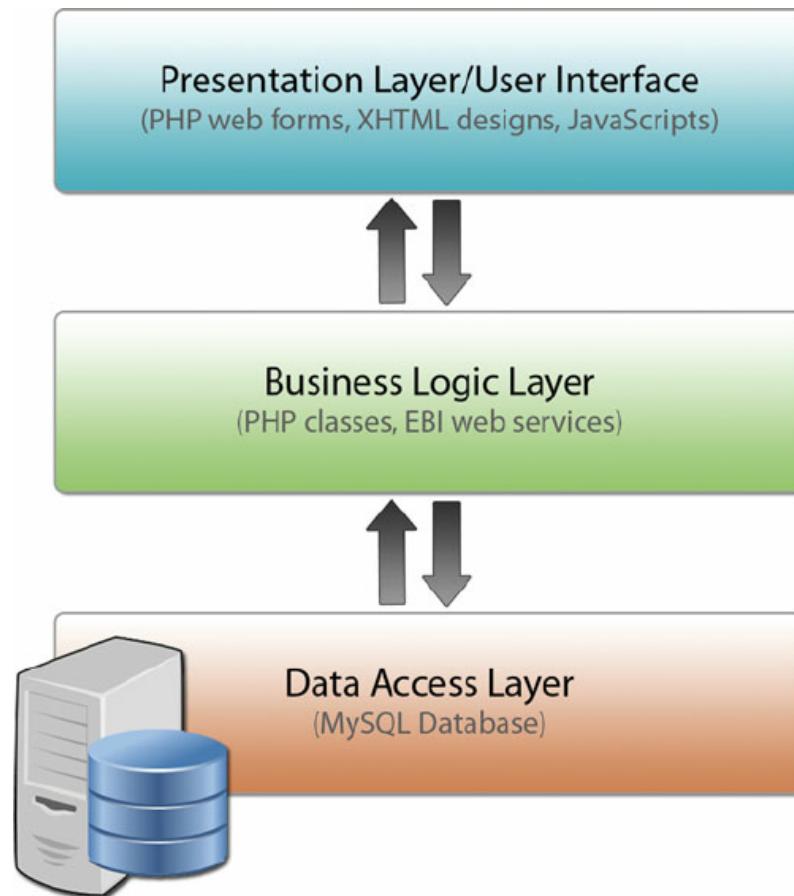
4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Hệ thống được xây dựng theo kiến trúc ba lớp (Three-Layer Archirecure). Kiến trúc này bao gồm tầng trình diễn (Presentation Layer), tầng logic nghiệp vụ (Business Logic Layer) và tầng truy cập dữ liệu (Data Access Layer). Cụ thể như sau:

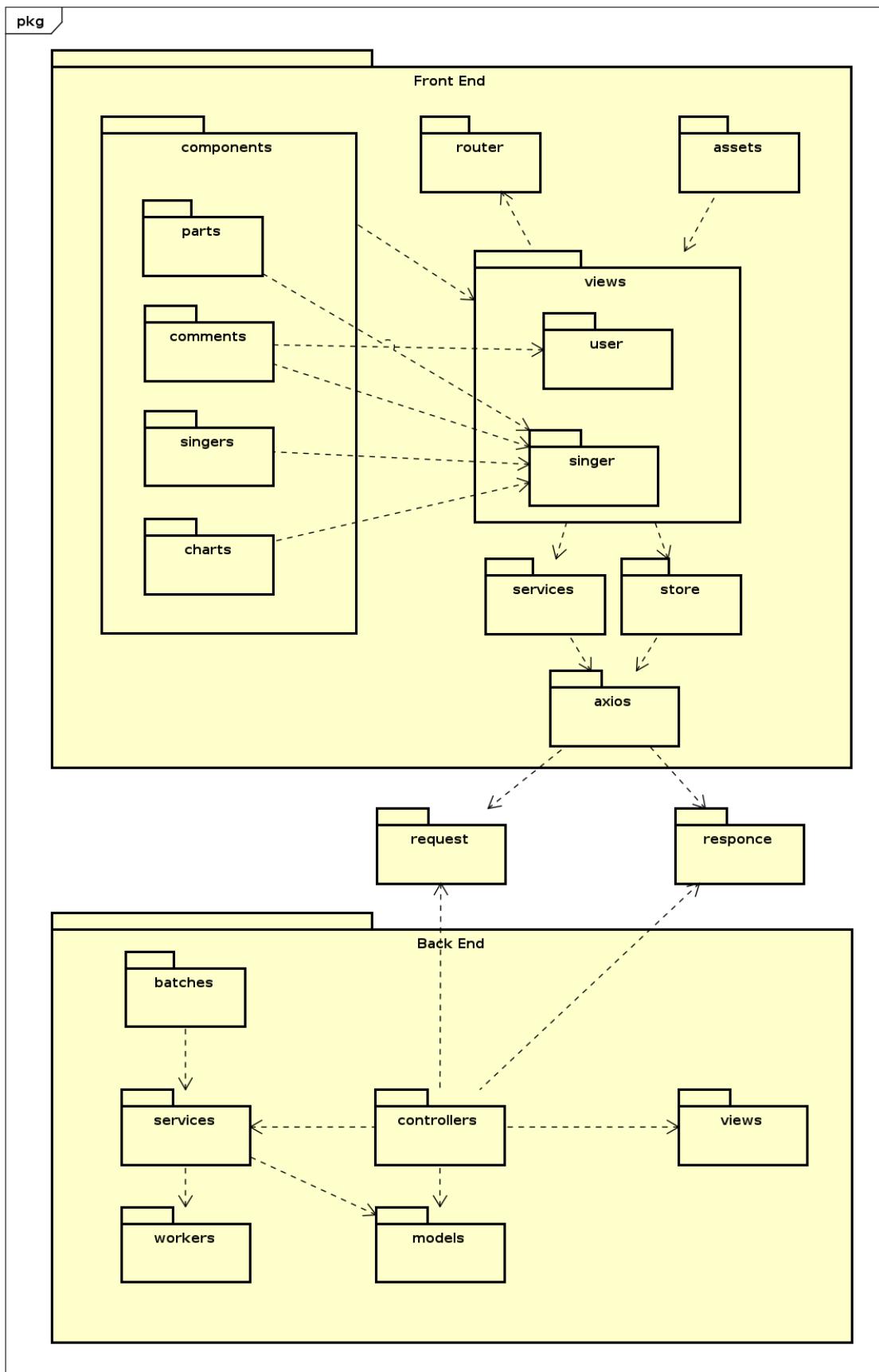
- **Tầng trình diễn:** Tầng này có nhiệm vụ chính là giao tiếp với người dùng. Nó gồm các thành phần giao diện (winform, webform, ...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL). Các giao diện web thường được phát triển sử dụng HTML, CSS, JavaScript với sự hỗ trợ của các framework, thư viện frontend, dưới dạng Single Page Application (SPA) hoặc Multiple Page Application (MPA). Trong đồ án này, tầng trình diễn được xây dựng dưới dạng Single Page Application sử dụng thư viện VueJS.
- **Tầng logic nghiệp vụ:** Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL. Ngoài ra, đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer. Tầng logic nghiệp vụ thường được phát triển sử dụng các ngôn ngữ như Java, Ruby, PHP,... với sự hỗ trợ của các framework backend. Trong đồ án này, tầng logic nghiệp vụ được xây dựng sử dụng ngôn ngữ Ruby và framework Ruby on rails.
- **Tầng truy cập dữ liệu:** Tầng này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,...).

Hình 4.1 mô tả kiến trúc ba lớp. Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính. Ngoài ra, ưu điểm của kiến trúc này là dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong một lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình. Chẳng hạn, thay đổi từ giao diện desktop sang giao diện web sẽ không yêu cầu thay đổi các tầng bên dưới.



Hình 4.1: Kiến trúc ba lớp (Three-layer)

4.1.2 Thiết kế tổng quan



Hình 4.2: Biểu đồ thiết kế tổng quan của hệ thống

Hệ thống được thiết kế thành hai phần riêng biệt backend và frontend.

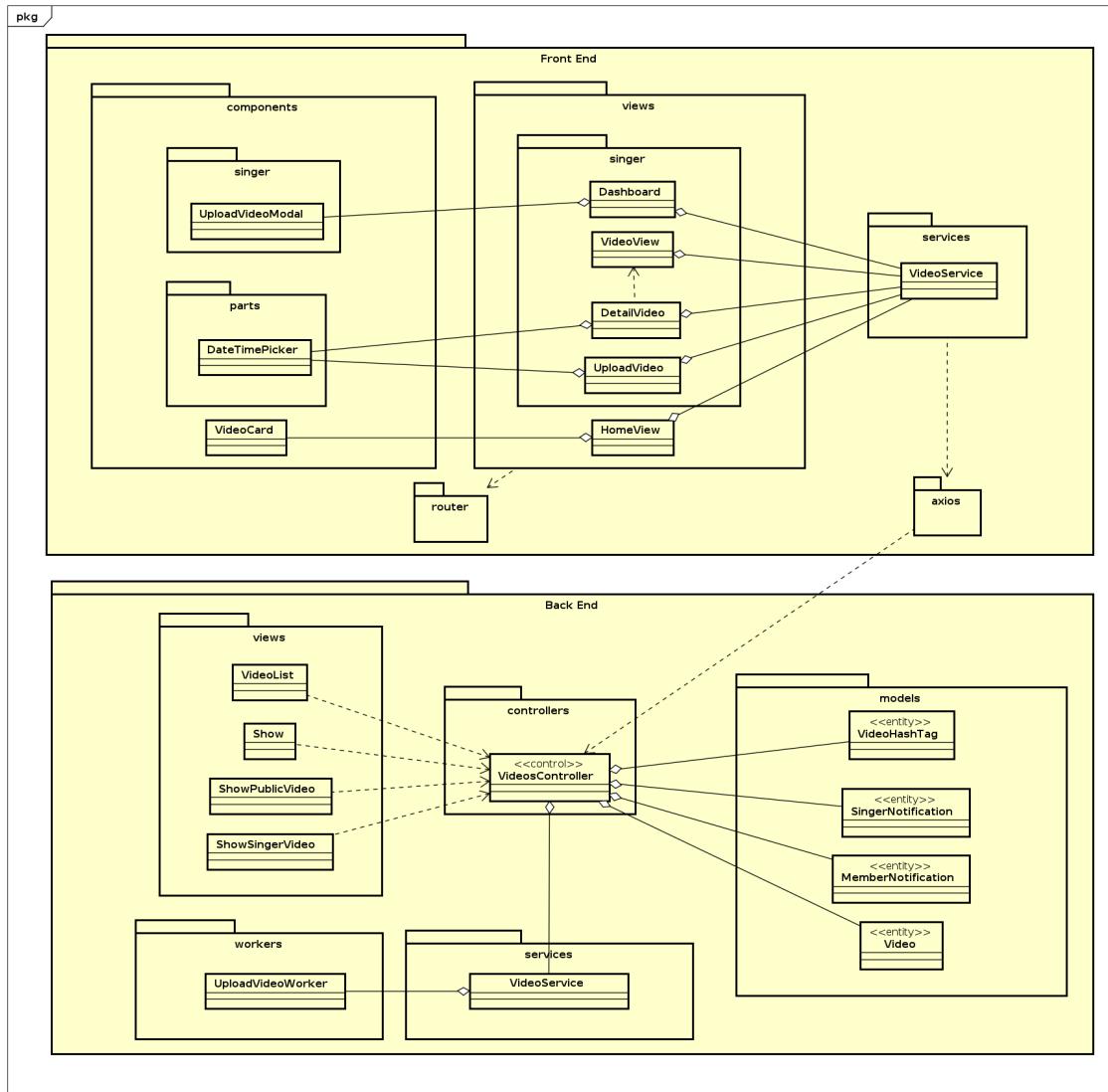
- Kiến trúc Frontend có bốn phần chính là components, views, service và store. Hệ thống sử dụng VueJs và Vuex để quản lý các trạng thái của ứng dụng. Người dùng sẽ tương tác trực tiếp với thành phần views - là tập hợp các UI components để thay đổi các trạng thái của ứng dụng, khi đó View sẽ gọi đến các hành động được thiết kế sẵn trong Services hoặc các hành động trong Store, request API lên server. Khi nhận được dữ liệu từ server sẽ tiến hành cập nhật lại dữ liệu cho người dùng.
- Kiến trúc Backend được xây dựng bằng Ruby trên framework Ruby on Rails bao gồm bốn phần chính: Controllers, Services, Model, Views. Phía backend được xây dựng như một ứng dụng API endpoint cung cấp danh sách API cho client. Khi có yêu cầu từ client, yêu cầu được bộ router điều hướng đến controller phù hợp. Tại đây, Controller sẽ gọi đến các Services để xử lý yêu cầu, Services sẽ tác động, lấy dữ liệu thông qua Models và trả lại Views bằng kiểu dữ liệu json, kết thúc luồng xử lý request.

4.1.3 Thiết kế chi tiết gói

Hình 4.3 là biểu đồ chi tiết gói của chức năng xem danh sách, đặt lịch, chỉnh sửa, xóa video, cụ thể như sau:

- Gói components bao gồm những thành phần như : UploadVideoModal (modal tạo mới video), DateTimePicker (component chọn ngày giờ) và VideoCard.
- Gói views trong front end bao gồm những thành phần như : DashBoard (Màn hình trang chủ), VideoView (Màn hình danh sách video), DetailVideo (Màn hình chi tiết video), UploadVideo (Màn hình tải lên video), HomeView (Màn hình danh sách video cho người dùng).
- Gói services trong font end gồm có VideoService dùng để gọi đến các hàm request lên server.
- Gói views trong back end bao gồm những thành phần như : VideoList, Show, ShowPublicVideo, ShowSingerVideo dùng để thiết kế cấu trúc các trường dữ liệu để trả về front end.
- Gói models gồm những thành phần như : VideoHashTag (chủ đề video), SingerNotification (thông báo đến ca sĩ), MemberNotification (thông báo đến người dùng), Video.
- Gói services trong back end gồm có VideoService dùng để xử lý các tác vụ liên quan đến chức năng lập lịch phát hành video.

- Gói workers gồm có UploadVideoWorker dùng để xử lý các tác vụ liên quan đến chức năng đặt lịch phát hành video.



Hình 4.3: Biểu đồ chi tiết gói chức năng xem, đặt lịch, chỉnh sửa và xóa video

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

a, Đặc tả thông tin màn hình ứng dụng

Các thông số màn hình mà ứng dụng hướng tới như sau:

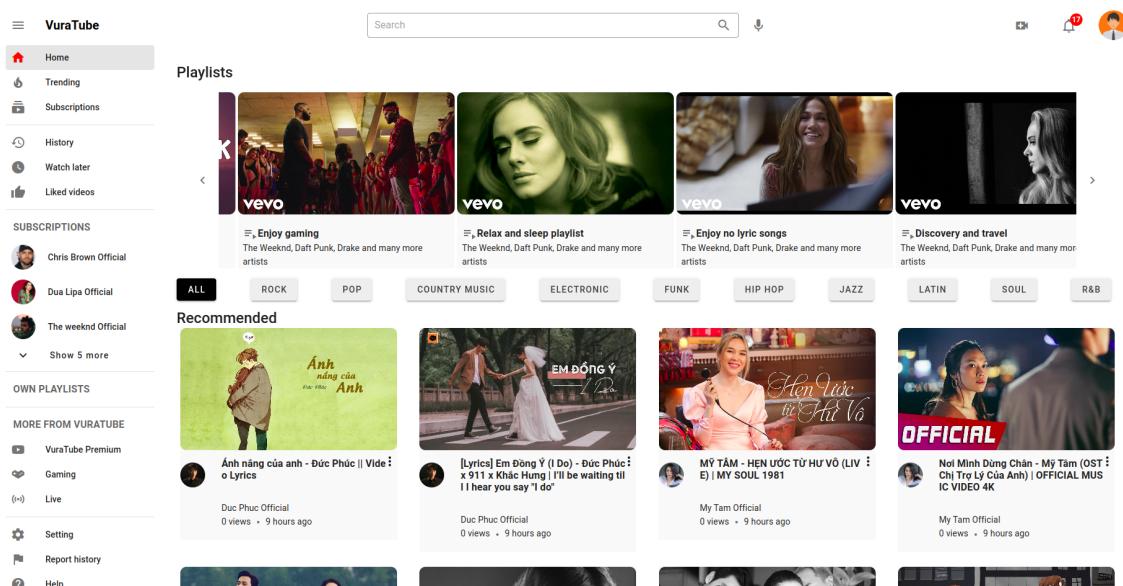
- Độ phân giải màn hình là 1600 x 900 và 1920 x 1080
- Kích thước màn hình là 16 inch
- Màu sắc sử dụng là hệ màu RGB và RGBA
- Màu sắc chủ đạo là màu trắng

Một số thông nhất, chuẩn hóa khi thiết kế giao diện:

- Ở mỗi màn hình luôn có thanh sidebar ở bên trái, thanh navbar được thiết kế trên cùng, ở giữa thường là nội dung hiển thị. Thanh tìm kiếm được đặt trên cùng ở giữa cùng với thanh navbar
- Các nút phải có kích thước phần đệm (padding), kích thước phần viền (border), màu viền và bo viền thống nhất. Màu nền và màu chữ của nút phải đảm bảo tính tương phản: nếu màu nền trắng, phải sử màu chữ đen hoặc xám; nếu màu nền không phải màu trắng (chẳng hạn đỏ, xanh, tím) thì sử dụng màu chữ trắng. Ngoài ra, màu nền của nút nên thể hiện được loại chức năng hoặc mức độ tác động mà hành động trên nút có thể gây ra (chẳng hạn, các nút xóa nên có nền màu đỏ)
- Các icon thu thập, tổng hợp từ thư viện Material Design Icons của Pictogrammers
- Các thành phần giao diện cần được bố trí hợp lý, bố cục cân đối, hài hòa
- Thiết kế giao diện hướng tới sự đơn giản, thân thiện với người dùng

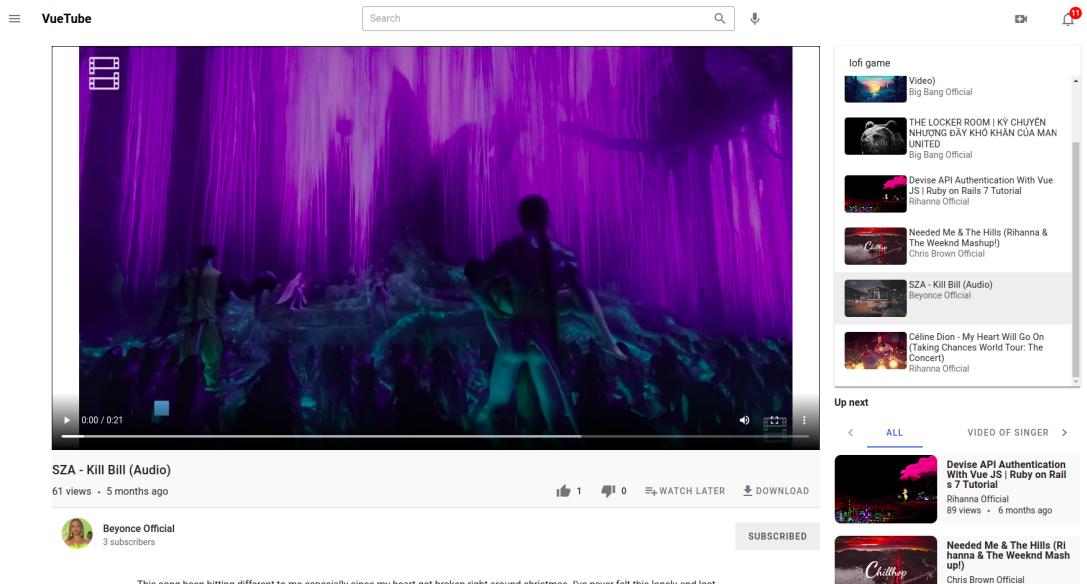
b, Thiết kế giao diện cho một số chức năng

Các hình 4.4, 4.5, 4.6, 4.7 dưới đây minh họa thiết kế giao diện cho một số chức năng quan trọng của hệ thống.

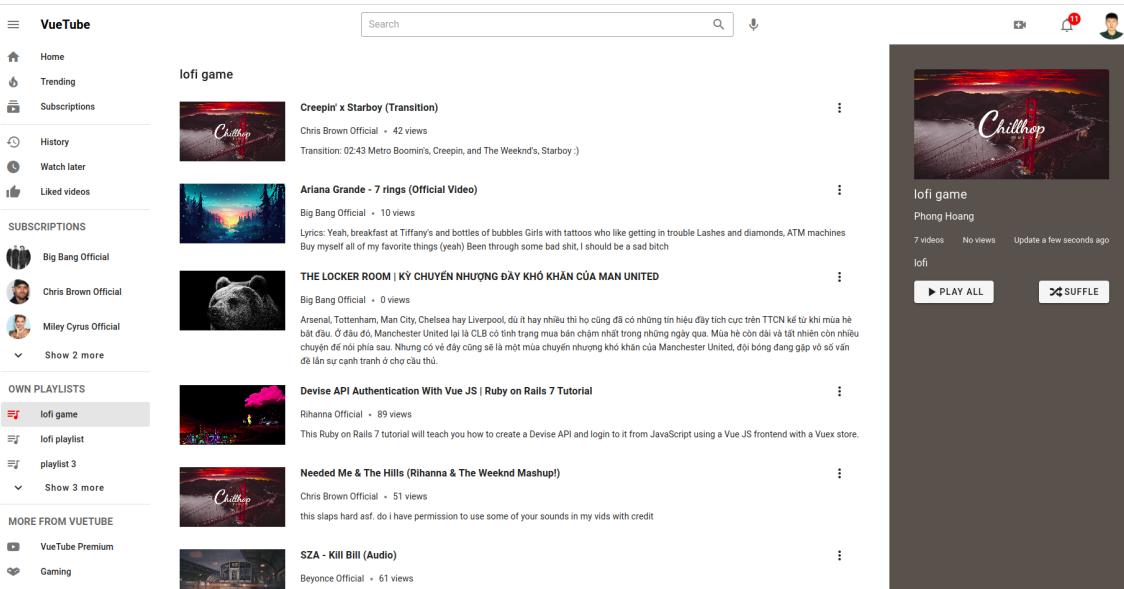


Hình 4.4: Thiết kế giao diện danh sách video

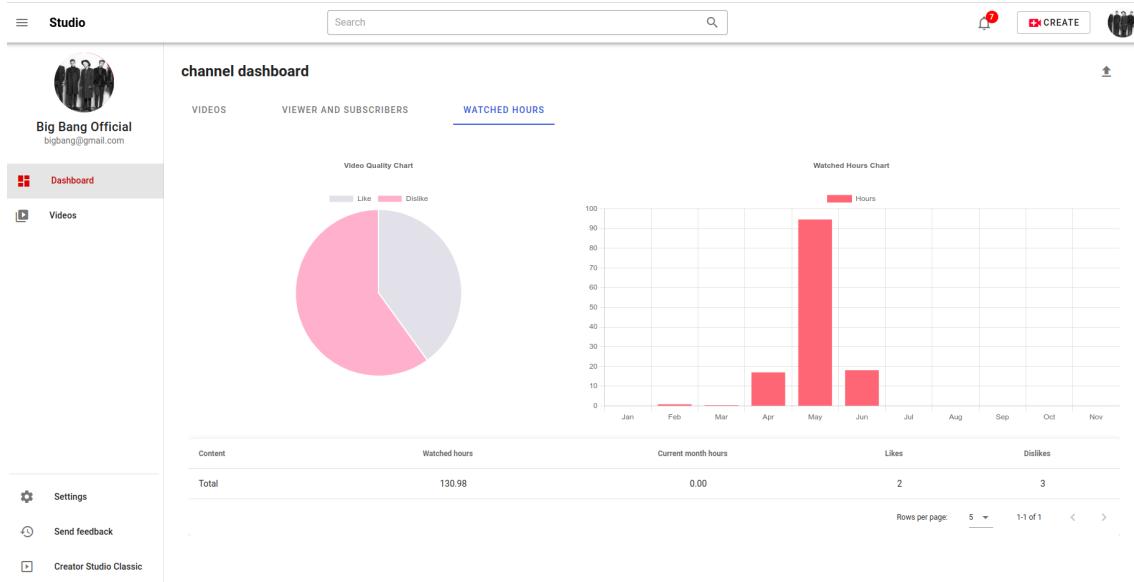
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.5: Thiết kế giao diện xem video



Hình 4.6: Thiết kế giao diện xem danh sách phát



Hình 4.7: Thiết kế giao diện thông kê cho ca sĩ

4.2.2 Thiết kế lớp

Do đồ án hạn chế số trang, phần này sẽ trình bày thiết kế lớp cho một số chức năng gồm đặt lịch, sửa, xóa video của ca sĩ.

a, Lớp VideosController

VideosController	
- videos : array	
- video : object	

+ index() : void
+ show() : void
+ create() : void
+ update() : void
+ destroy() : void
+ handle_send_notification(video : object, title : string) : void
+ handle_send_scheduling_video_notification(video : object, title : string, singer_id : int) : void
- handle_schedule_upload_video(video : object, schedule_at : date_time) : void
+ update_thumbnails() : void
- set_video() : void
- video_params() : void

Hình 4.8: Lớp VideosController

Attribute:

1. videos: Là mảng các video khi request đến hàm index()
2. video: Là video có kiểu dữ liệu object.

Operation:

1. index

Mô tả: Hàm lấy danh sách videos

Parameter: Không có

Return: ResponseEntity với status HttpStatus.OK và body là một Collection của các video

*Exception:*Exception: Không có

2. show

Mô tả: Hàm dùng để lấy ra video theo id

Parameter: Không có

Return: ResponseEntity với status HttpStatus.OK và body là một object chính là video vừa tìm thấy

Exception: Không có

3. create

Mô tả: Tạo mới một video vào CSDL

Parameter:

- attachment (mp4 file) :File video vừa upload
- category_id: ID thể loại video

Return: ResponseEntity với status HttpStatus.OK và body là video vừa tạo xong nếu tạo mới thành công. Ngược lại, trả về ReponseEntity với status HttpStatus.INTERNAL_SERVER_ERROR nếu xảy ra lỗi trong quá trình tạo video mới.

Exception: Không có

4. update

Mô tả: Cập nhật thông tin video

Parameter: Thông tin video như : tiêu đề, mô tả, Id thể loại, trạng thái, thời gian đặt lịch (nếu có), id ca sĩ

Return: Đối tượng Video vừa cập nhật nếu cập nhật thành công. Ngược lại, trả về ReponseEntity với status HttpStatus.INTERNAL_SERVER_ERROR nếu xảy ra lỗi trong quá trình cập nhật video

Exception: Không có

5. destroy

Mô tả: Xóa video

Parameter: Id của video muốn xóa

Return: Không có

Exception: ResourceNotFoundException nếu không tồn tại video với id đã cung cấp

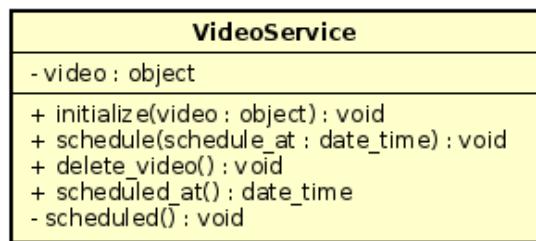
6. video_params

Mô tả: Hàm trả về danh sách tham số của video, có nhiệm vụ validate và lọc các params không phù hợp

Parameter: Không có

Return: Danh sách tham số của video

Exception: Không có



Hình 4.9: Lớp VideoService

Attribute:

1. video: Là video có kiểu dữ liệu object.

Operation:

1. initialize

Mô tả: Hàm khởi tạo một instance mới

Parameter: video

Return: Không trả về giá trị, chỉ khởi tạo và thiết lập trạng thái cho đối tượng

Exception: Không có

2. schedule

Mô tả: Lập lịch phát hành cho video, lưu thông tin video và thời điểm phát hành vào hằng đợi

Parameter: schedule_at: Thời điểm phát hành video

Return: Không có

Exception: Không có

3. delete_video

Mô tả: Xóa job chứa video ra khỏi hằng đợi. Hàm được khởi chạy khi có yêu cầu xóa video

Parameter: Không có

Return: Không có

Exception: Không có

UploadVideoWorker
+ perform(video_id : int) : void + scheduled_job(video_id : int) : void + delete_job(video_id : int) : void - target_members_to_sent_notification(video : object) : void - handle_send_notification(video : object) : void - handle_send_notification_for_singer(video : object) : void

Hình 4.10: Lớp UploadVideoWorker

Operation:

1. perform

Mô tả: Hàm xử lý mỗi khi lấy job từ hằng đợi ra. Hàm có mục đích cập nhật trạng thái của video và gửi thông báo đến người dùng, ca sĩ.

Parameter: video_id: ID của video

Return: Không có

Exception: Không có

2. scheduled_job

Mô tả: Hàm xử lý lấy ra job đã được lập lịch trong queue

Parameter: video_id: ID của video

Return: job chứa thông tin video id và thời gian lập lịch

Exception: Không có

3. delete_job

Mô tả: Hàm xử lý xóa job của video ra khỏi hằng đợi.

Parameter: video_id: ID của video

Return: Không có

Exception: Không có

4. handle_send_notification_for_singer

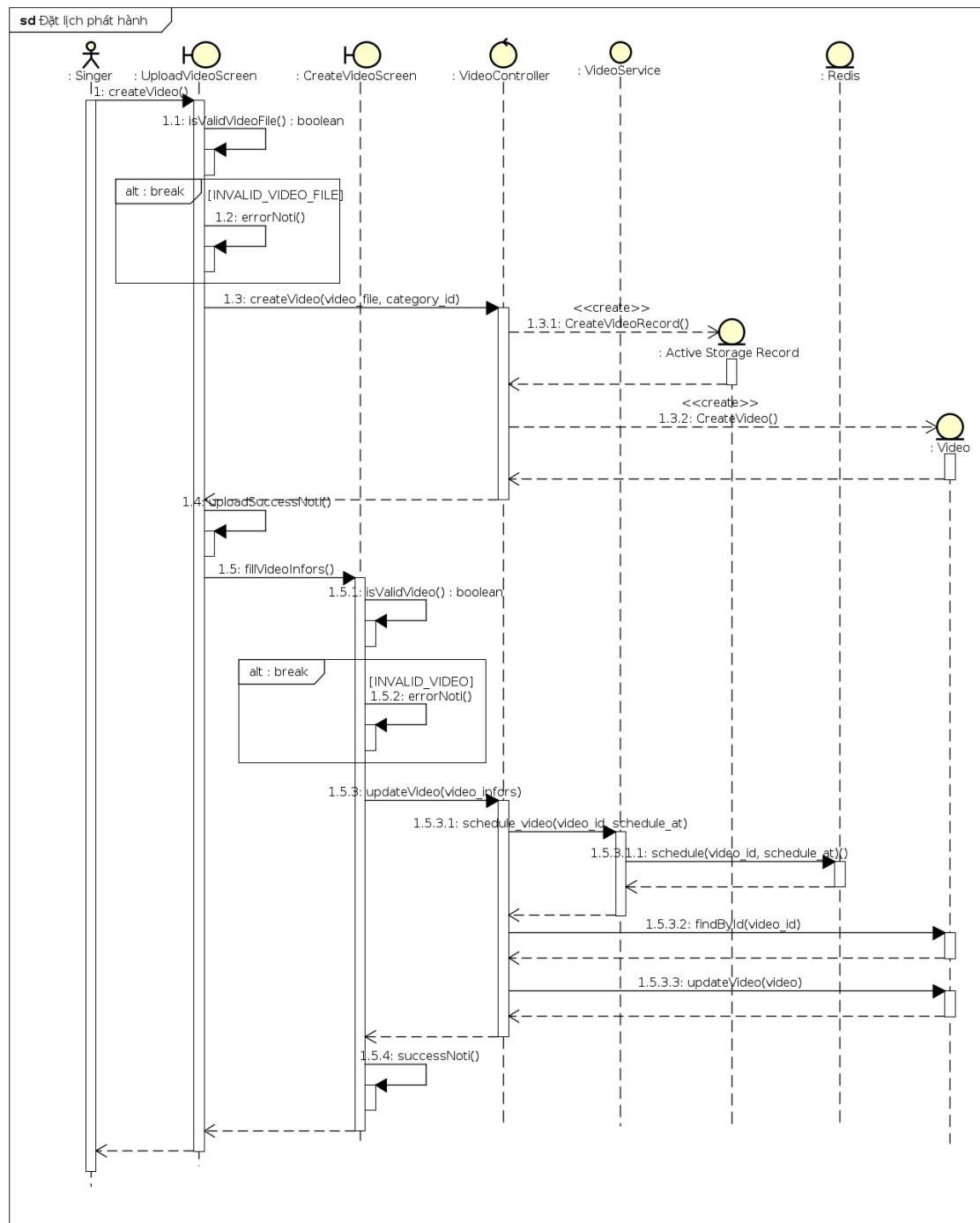
Mô tả: Hàm gửi thông báo cho ca sĩ sau khi phát hành xong video

Parameter: video

Return: Không có

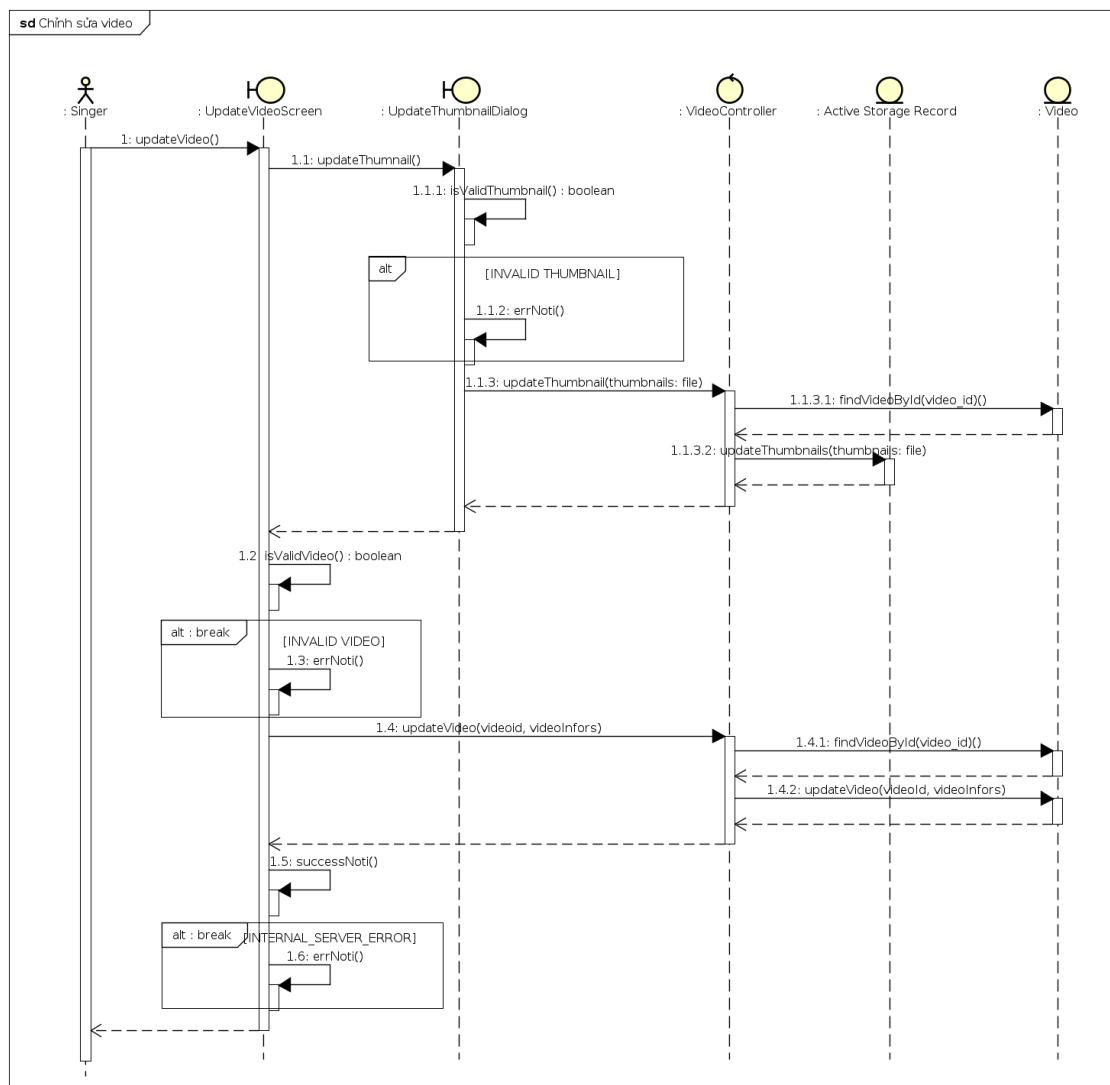
Exception: Không có

Để minh họa cho thiết kế lớp, hình 4.11, 4.12, 4.13 là biểu đồ trình tự cho các use case đặt lịch, sửa và xóa video cho ca sĩ:

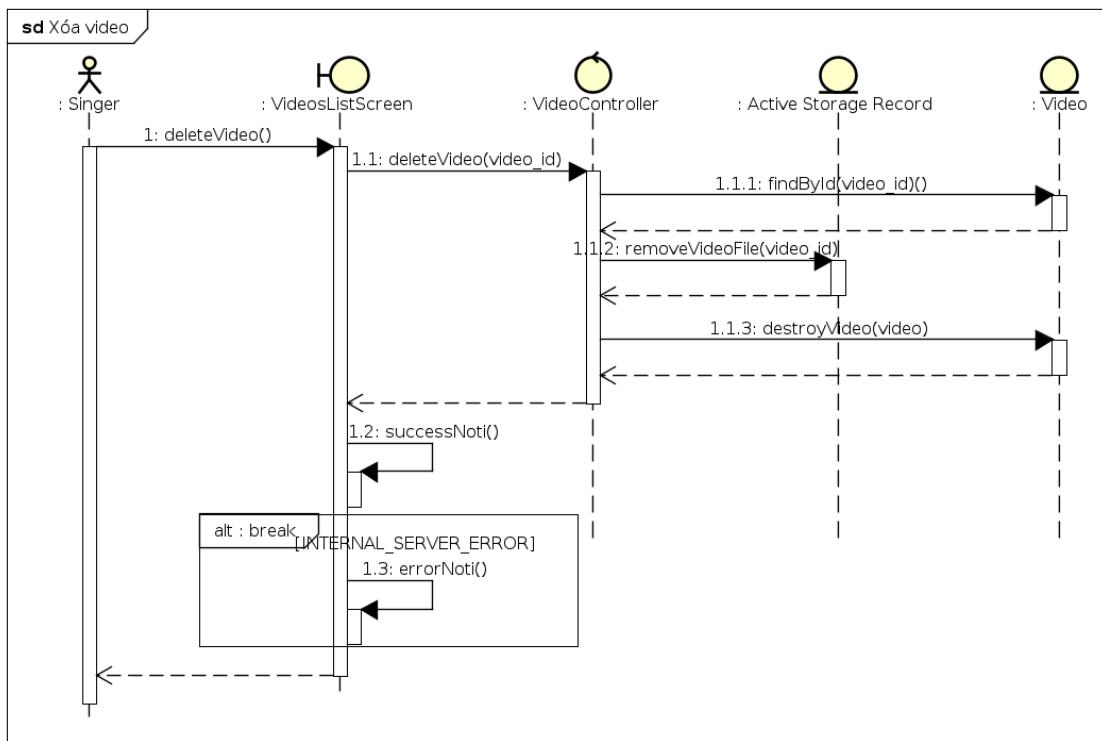


Hình 4.11: Biểu đồ trình tự ca sĩ tạo và đặt lịch phát video

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

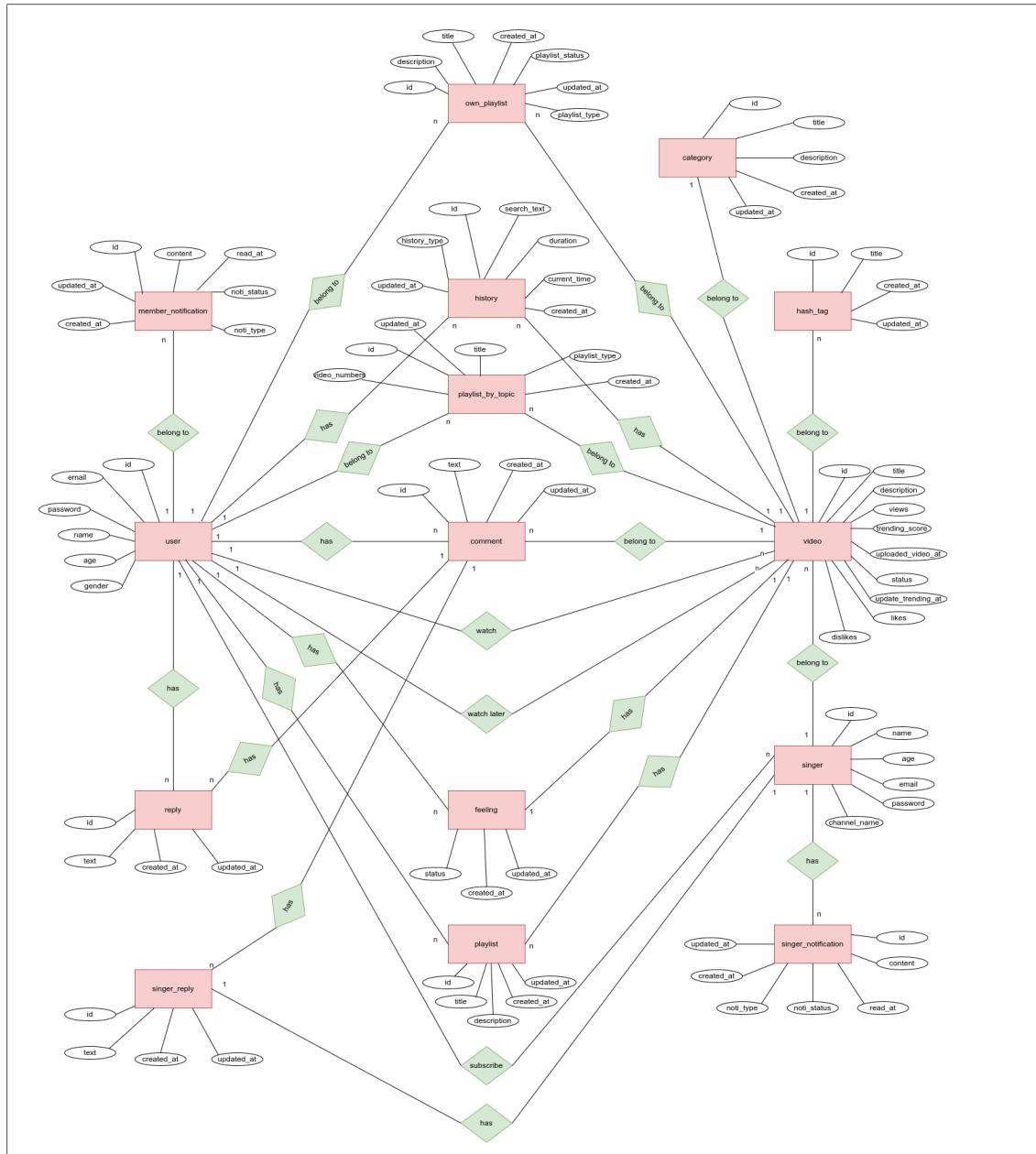


Hình 4.12: Biểu đồ trình tự ca sĩ cập nhật thông tin video



Hình 4.13: Biểu đồ trình tự ca sĩ xóa video

4.2.3 Thiết kế cơ sở dữ liệu

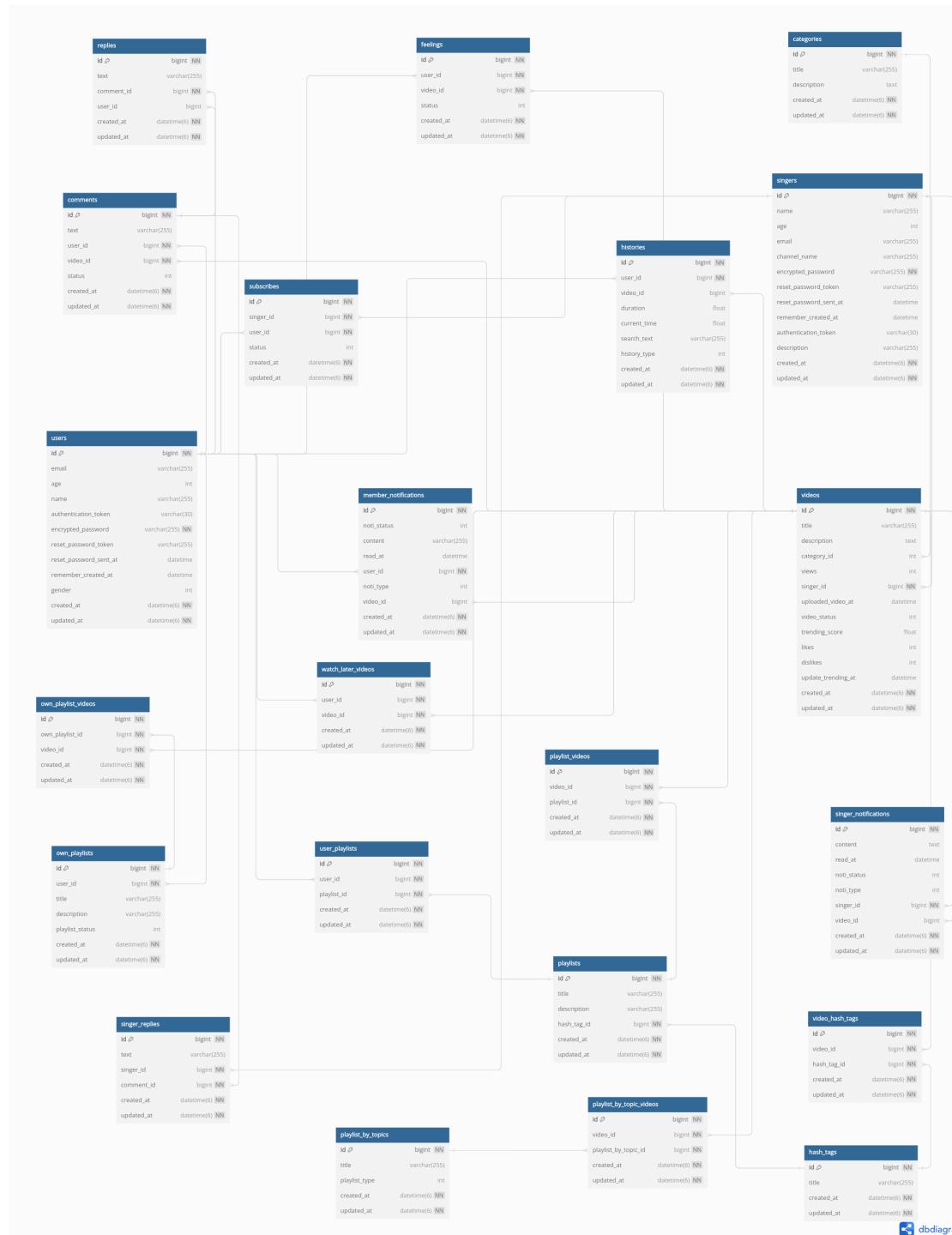


Hình 4.14: Sơ đồ thực thể liên kết

Hình 4.14 mô tả sơ đồ thực thể liên kết cho hệ thống. Sơ đồ bao gồm 15 thực thể: user (người dùng), singer (ca sĩ), video, category (thể loại), comment (bình luận), feeling (cảm xúc), hash_tag, history (lịch sử), member_notification (thông báo người dùng), own_playlist (danh sách phát riêng), reply (phản hồi bình luận), singer_reply (phản hồi của ca sĩ), playlist (danh sách phát), playlist_by_topic (danh sách phát theo chủ đề), singer_notification (thông báo của ca sĩ)

Từ sơ đồ thực thể liên kết, dễ thấy mỗi thể loại âm nhạc có thể bao gồm nhiều video, mỗi video thuộc về một ca sĩ và có thể có nhiều hashtag. Mỗi danh sách phát (playlist) tập hợp nhiều video có cùng hashtag. Mỗi người dùng có thể xem

video, tương tác với thực thể ca sĩ bằng cách đăng ký kênh. Người dùng còn có xem lịch sử video (history), xem các video xem sau (watch_later), react, bình luận và trả lời bình luận dưới mỗi video. Ngoài ra, người dùng có thể tạo các danh sách phát cho bản thân (own_playlist). Ứng dụng còn xây dựng hệ thống thông báo cho người dùng và ca sĩ. Dựa vào sơ đồ thực thể liên kết, ta có thiết kế chi tiết các bảng trong CSDL như hình 4.15



Hình 4.15: Thiết kế chi tiết các bảng

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Sau đây là đặc tả chi tiết cho mỗi bảng trong CSDL:

1. *Bảng user*

Bảng 4.1: Bảng user

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã người dùng
email	varchar(255)	Có	Unique	Email của người dùng
name	varchar(255)	Có	Không	Tên người dùng
encrypted_password	varchar(255)	Có	Không	Mật khẩu
gender	int	Có	Không	Giới tính
age	int	Có	Không	Tuổi của người dùng
reset_password_token	varchar(255)	Không	Không	Token đặt lại mật khẩu
reset_password_sent_at	datetime	Không	Không	Thời gian đặt lại mật khẩu
remember_created_at	datetime	Không	Không	Thời điểm tạo remember
authentication_token	varchar(30)	Không	Không	Token xác thực
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

2. *Bảng video*

Bảng 4.2: Bảng video

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã video
title	varchar(255)	Có	Không	Tiêu đề của video
description	text	Có	Không	Mô tả video
category_id	int	Có	Khóa ngoại tham chiếu tới cột id của bảng category	Thể loại video
views	int	Có	Không	Lượt xem
single_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng singer	Mã ca sĩ
dislikes	int	Có	Không	Lượt không thích
update_trending_at	datetime	Có	Không	Thời điểm cập nhật danh sách video xu hướng
uploaded_video_at	datetime	Không	Không	Thời điểm đăng tải video

Bảng 4.2 – Còn nữa ở trang tiếp

Bảng 4.2 – Tiếp tục từ trang trước

video_status	int	Có	Không	Trạng thái video: nhận một trong các giá trị public, private, scheduling
trending_score	float	Có	Không	Điểm xu hướng
likes	int	Có	Không	Lượt thích
created_at	datetime(6)	Có	Không	Thời điểm tạo
update_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật video

3. *Bảng category*

Bảng 4.3: Bảng category

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã thể loại
title	varchar(255)	Có	Không	Tên thể loại
description	text	Có	Không	Mô tả thể loại
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

4. *Bảng singer*

Bảng 4.4: Bảng singer

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã ca sĩ
email	varchar(255)	Có	Unique	Email của ca sĩ
name	varchar(255)	Có	Không	Tên ca sĩ
encrypted_password	varchar(255)	Có	Không	Mật khẩu
description	varchar(255)	Có	Không	Mô tả kênh
age	int	Có	Không	Tuổi của ca sĩ
channel_name	varchar(255)	Có	Không	Tên channel
reset_password_token	varchar(255)	Không	Không	Token đặt lại mật khẩu
reset_password_sent_at	datetime	Không	Không	Thời gian đặt lại mật khẩu
remember_created_at	datetime	Không	Không	Thời điểm tạo remember
authentication_token	varchar(30)	Không	Không	Token xác thực
created_at	datetime(6)	Có	Không	Thời điểm tạo

Bảng 4.4 – Còn nữa ở trang tiếp

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Bảng 4.4 – Tiếp tục từ trang trước

updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng
------------	-------------	----	-------------------	------------------------------

5. Bảng comment

Bảng 4.5: Bảng comment

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã bình luận
text	varchar(255)	Có	Không	Nội dung bình luận
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng user	ID người dùng
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng video	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

6. Bảng feeling

Bảng 4.6: Bảng feeling

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã feeling
status	int	Có	Không	Trạng thái feeling
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng user	ID người dùng
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng video	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

7. Bảng hash_tag

Bảng 4.7: Bảng hash_tag

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
----------------	--------------	----------	-----------	-------

Bảng 4.7 – Còn nữa ở trang tiếp

Bảng 4.7 – Tiếp tục từ trang trước

id	bigint	Có	Khoá chính	Mã hash tag
title	varchar(255)	Có	Không	Tên hash tag
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

8. *Bảng history*

Bảng 4.8: Bảng history

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã lịch sử
search_text	varchar(255)	Có	Không	Nội dung tìm kiếm
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng user	ID người dùng
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng video	ID video
history_type	int	Có	Không	Thể loại lịch sử
duration	float	Có	Không	Thời lượng video
current_time	float	Có	Không	Thời lượng xem
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

9. *Bảng member_notification*

Bảng 4.9: Bảng member_notification

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã thông báo
content	varchar(255)	Có	Không	Nội dung thông báo
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng user	ID người dùng
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng video	ID video
noti_type_id	int	Có	Không	Loại thông báo
noti_status	int	Có	Không	Trạng thái của thông báo

Bảng 4.9 – Còn nữa ở trang tiếp

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Bảng 4.9 – Tiếp tục từ trang trước

read_at	datetime(6)	Có	Không	Thời điểm đọc thông báo
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

10. *Bảng own_playlist_video*

Bảng 4.10: Bảng own_playlist_video

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã own playlist video
own_playlist_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng own_playlist	ID danh sách phát
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng video	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

11. *Bảng own_playlist*

Bảng 4.11: Bảng own_playlist

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã danh sách phát riêng
title	varchar(255)	Có	Không	Tên danh sách phát riêng
description	varchar(255)	Có	Không	Mô tả về danh sách phát riêng
playlist_status	int	Có	Không	Trạng thái của danh sách phát riêng
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột id của bảng user	ID người dùng
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

12. *Bảng playlist_video*

Bảng 4.12: Bảng playlist_video

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã own playlist video
playlist_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>playlist</i>	ID danh sách phát
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

13. *Bảng playlist*

Bảng 4.13: Bảng playlist

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã danh sách phát riêng
title	varchar(255)	Có	Không	Tên danh sách phát riêng
description	varchar(255)	Có	Không	Mô tả về danh sách phát riêng
hash_tag_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>hash_tag</i>	ID hash tag
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

14. *Bảng reply*

Bảng 4.14: Bảng reply

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã danh sách phát riêng
text	varchar(255)	Có	Không	Tên danh sách phát riêng

Bảng 4.14 – Còn nữa ở trang tiếp

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Bảng 4.14 – Tiếp tục từ trang trước

comment_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>comment</i>	Mô tả về danh sách phát riêng
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>user</i>	ID người dùng
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

15. *Bảng singer_reply*

Bảng 4.15: Bảng *singer_reply*

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
<i>id</i>	bigint	Có	Khoa chính	Mã danh sách phát riêng
<i>text</i>	varchar(255)	Có	Không	Tên danh sách phát riêng
<i>comment_id</i>	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>comment</i>	Mô tả về danh sách phát riêng
<i>singer_id</i>	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>singer</i>	ID ca sĩ
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

16. *Bảng singer_notification*

Bảng 4.16: Bảng *singer_notification*

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
<i>id</i>	bigint	Có	Khoa chính	Mã thông báo
<i>content</i>	varchar(255)	Có	Không	Nội dung thông báo
<i>singer_id</i>	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>singer</i>	ID ca sĩ
<i>video_id</i>	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video

Bảng 4.16 – Còn nữa ở trang tiếp

Bảng 4.16 – Tiếp tục từ trang trước

noti_type	int	Có	Không	Loại thông báo
noti_status	int	Có	Không	Trạng thái của thông báo
read_at	datetime(6)	Có	Không	Thời điểm đọc thông báo
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

17. *Bảng subscribe*

Bảng 4.17: Bảng subscribe

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã đăng ký
status	int	Có	Không	Trạng thái đăng ký
user_id	varchar(255)	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>user</i>	ID người dùng
singer_id	varchar(255)	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>singer</i>	ID ca sĩ
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

18. *Bảng user_playlist*

Bảng 4.18: Bảng user_playlist

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã playlist video
playlist_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>playlist</i>	ID danh sách phát
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>user</i>	ID người dùng
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

19. *Bảng video_hash_tag*

Bảng 4.19: Bảng video_hash_tag

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã playlist video
playlist_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>playlist</i>	ID danh sách phát
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

20. *Bảng watch_later_video*

Bảng 4.20: Bảng watch_later_video

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã playlist video
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>user</i>	ID người dùng
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

21. *Bảng playlist_by_topic*

Bảng 4.21: Bảng playlist_by_topic

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
id	bigint	Có	Khoá chính	Mã own playlist video
playlist_by_topic_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>playlist_by_topic</i>	ID danh sách phát

Bảng 4.21 – Còn nữa ở trang tiếp

Bảng 4.21 – Tiếp tục từ trang trước

video_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

22. *Bảng playlist_by_topic_video*

Bảng 4.22: Bảng *playlist_by_topic_video*

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Ràng buộc	Mô tả
<i>id</i>	bigint	Có	Khoá chính	Mã playlist video
video_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>video</i>	ID video
user_id	bigint	Có	Khóa ngoại tham chiếu tới cột <i>id</i> của bảng <i>user</i>	ID người dùng
created_at	datetime(6)	Có	Không	Thời điểm tạo
updated_at	datetime(6)	Có	Sau thời điểm tạo	Thời điểm cập nhật cuối cùng

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Một số công cụ, framework, library được sử dụng trong quá trình xây dựng đồ án được trình bày trong bảng 4.23 dưới đây.

Bảng 4.23: Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual studio code	https://code.visualstudio.com/
Hệ quản trị cơ sở dữ liệu MySQL	Mysql Workbench 8.0	https://dev.mysql.com/downloads/workbench/
Ngôn ngữ lập trình back-end	Ruby	https://www.ruby-lang.org
Ngôn ngữ lập trình front-end	Javascript	https://www.javascript.com/
Framework lập trình back-end	Ruby on Rails	https://rubyonrails.org/

Bảng 4.23 – Còn nữa ở trang tiếp

Bảng 4.23 – Tiếp tục từ trang trước

Framework lập trình front-end	Vuejs	https://vuejs.org/
Thư viện UI-component	Vuetify 2.6.0	https://vuetifyjs.com/
Vẽ biểu đồ UML	Astah UML 8.5.0	https://astah.net/products/astah-uml/
Vẽ sơ đồ ERD	draw.io 21.5.0	https://app.diagrams.net/

4.3.2 Kết quả đạt được

Ứng dụng đã được xây dựng thành công đảm bảo các chức năng như ban đầu đề ra, đáp ứng nhu cầu xem, nghe nhạc của người dùng và nhu cầu chia sẻ âm nhạc của ca sĩ. Người dùng có thể vừa nghe và xem những video âm nhạc do ứng dụng gợi ý theo nhiều thể loại khác nhau. Ca sĩ có thể đăng tải những video đến người dùng. Giao diện ứng dụng đơn giản, đẹp mắt và tương thích tốt trên giao diện web cho các màn hình 14, 15 và 16 inch.

Một số thông tin chính về ứng dụng được thống kê trong bảng 4.24

Bảng 4.24: Thống kê một số thông tin về ứng dụng

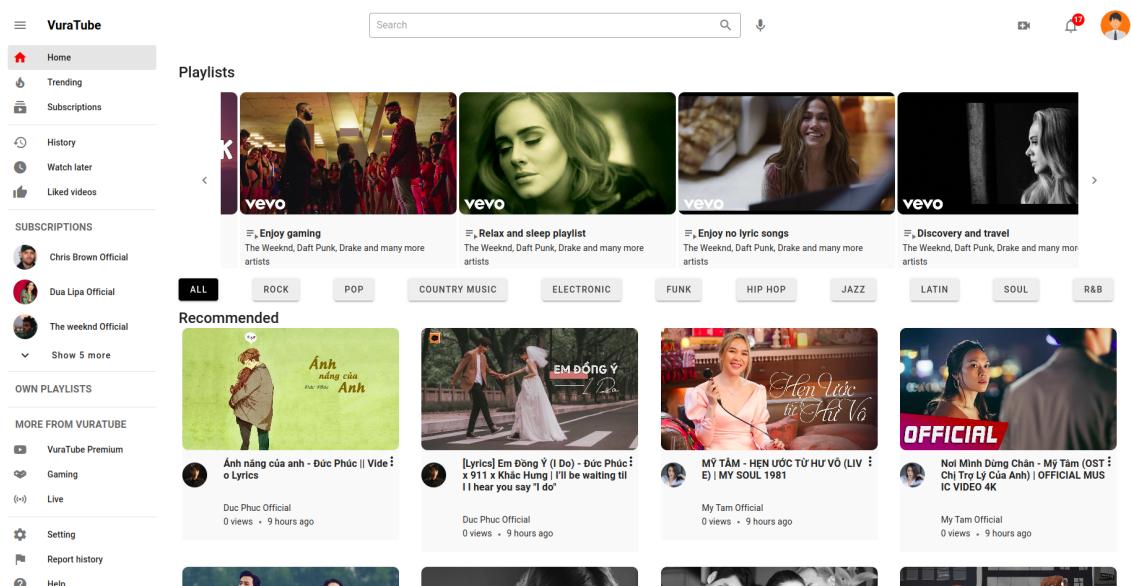
Thông tin	Thông kê
Số dòng code	14217
Số file	179
Số bảng trong CSDL	22
Dung lượng mã nguồn	755MB

4.3.3 Minh họa các chức năng chính

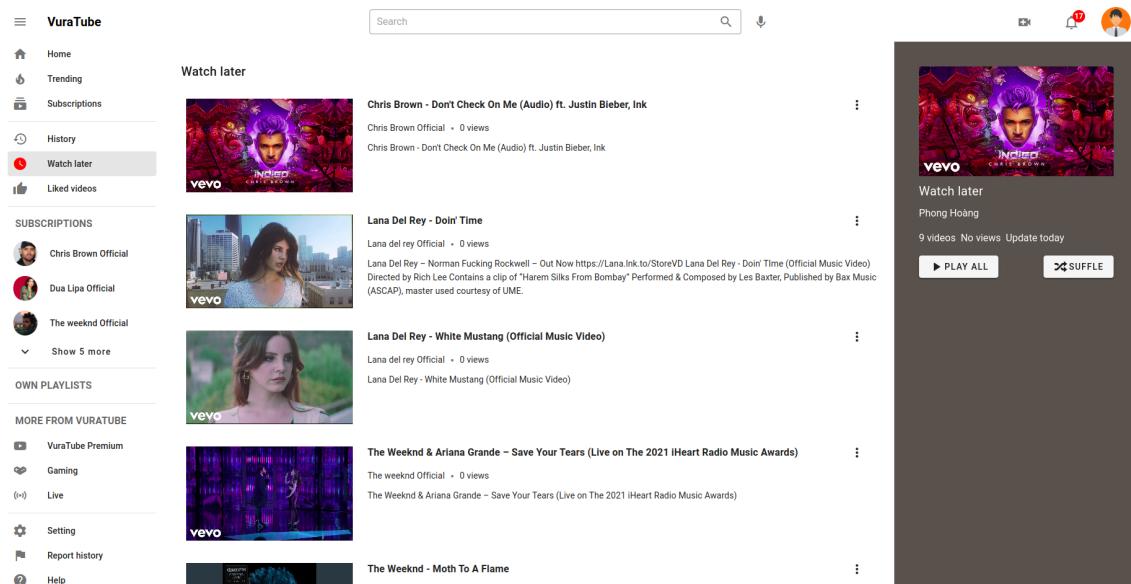
a, Màn hình danh sách video cho người dùng

- Hình 4.16 là màn hình trang chủ của người dùng. Trên cùng là thanh tìm kiếm video và ca sĩ. Người dùng có thể nhập thông tin video hoặc ca sĩ để tìm kiếm. Ở trang chủ này, hệ thống hiển thị danh sách các playlist do hệ thống gợi ý cho người dùng, các danh sách này bao gồm các video có cùng hashtag hay chủ đề nào đó. Ngoài ra, ở trang chủ còn hiển thị các video cụ thể. Người dùng có thể click vào để xem video, lưu video xem sau, thêm video vào danh sách phát do họ tạo ra.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



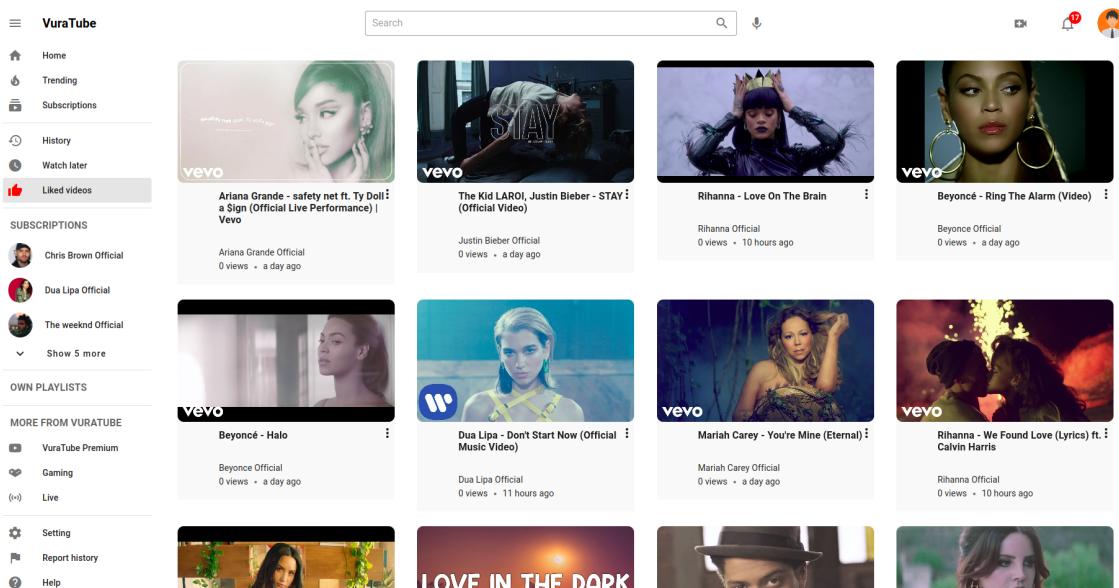
Hình 4.16: Màn hình trang chủ của người dùng



Hình 4.17: Màn hình danh sách video xem sau

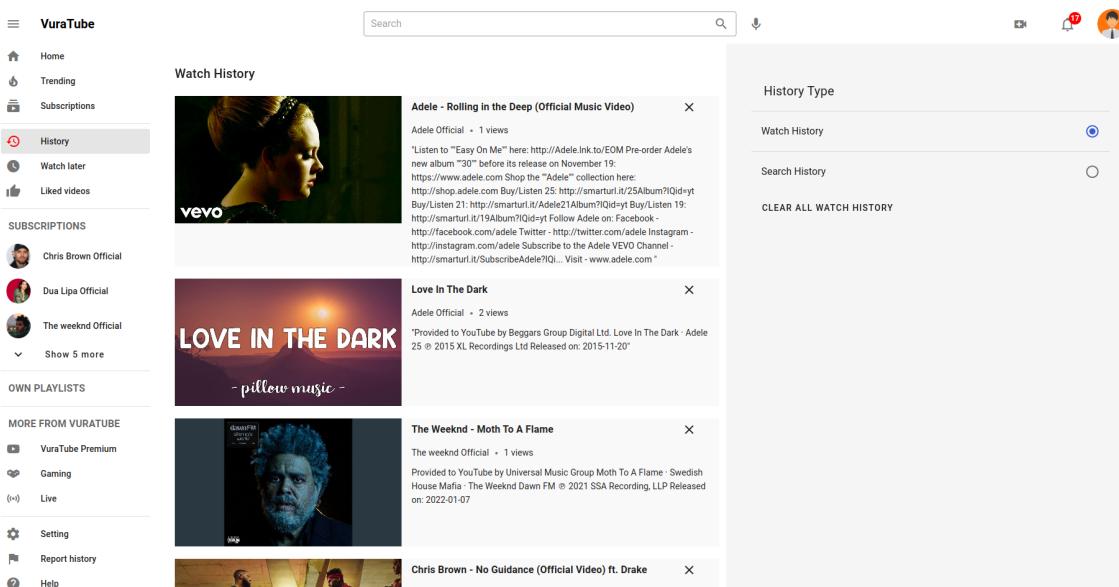
- Hình 4.17 là màn hình danh sách video xem sau của người dùng. Người dùng lưu lại các video này để xem lại. Tại màn hình này hiển thị chi tiết thông tin về danh sách như tổng số video, tên người dùng,... Người dùng có thể sắp xếp lại thứ tự video để xem. Khi người dùng bấm vào nút phát, hệ thống sẽ điều hướng đến màn hình xem video cùng danh sách phát này.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.18: Màn hình danh sách video đã thích

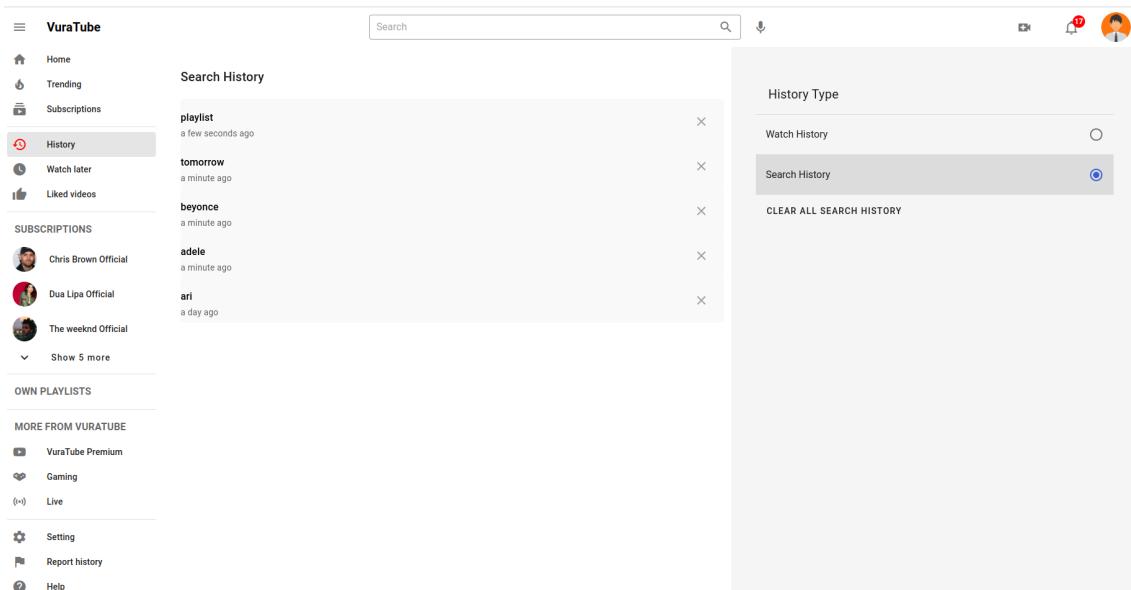
- Hình 4.18 là màn hình danh sách video đã thích. Người dùng có thể bấm vào để xem video hoặc bấm vào icon để thêm video vào danh sách xem sau hoặc danh sách phát do người dùng tạo ra.



Hình 4.19: Màn hình lịch sử video đã xem

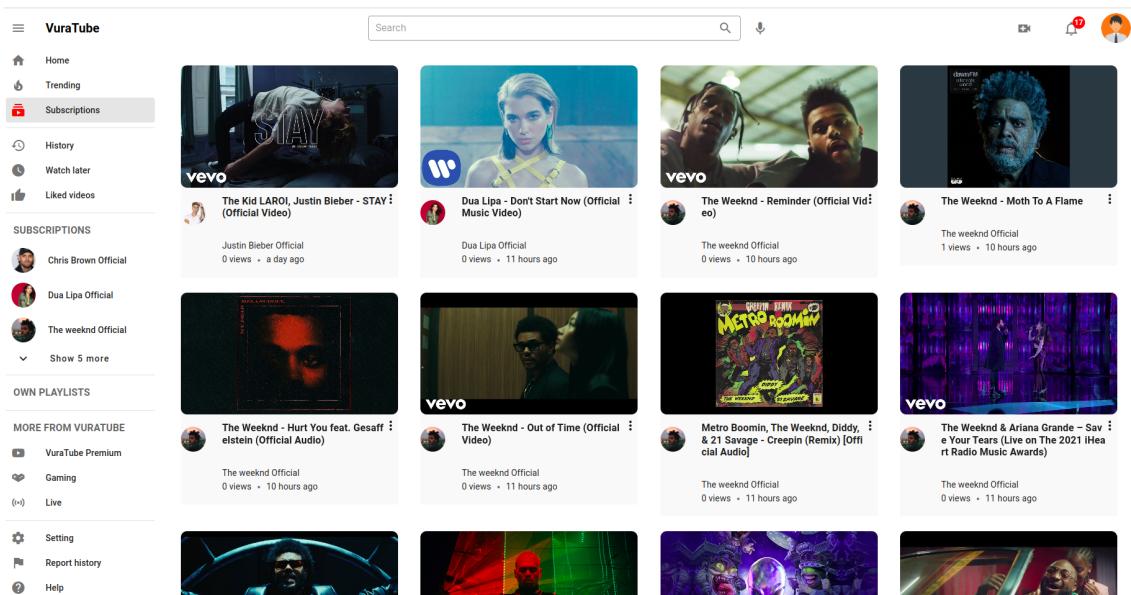
- Hình 4.19 là màn hình lịch sử video đã xem. Thuật toán xác định người dùng đã xem video được giải thích chi tiết ở chương 5.4. Tại màn hình này, người dùng có thể bấm vào từng video để xem, hoặc xóa video khỏi lịch sử khi bấm vào biểu tượng xóa bên phải mỗi video.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.20: Màn hình lịch sử tìm kiếm

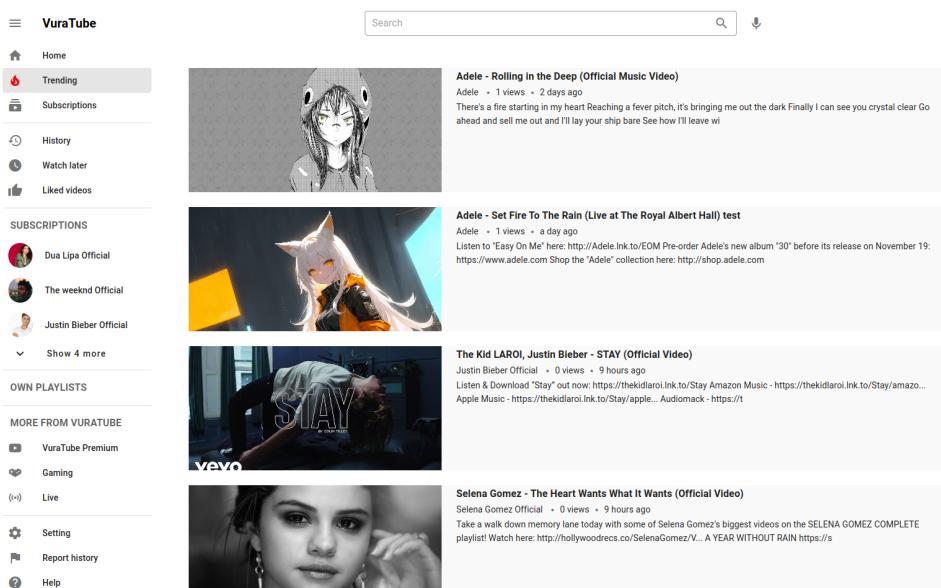
- Hình 4.20 là màn hình lịch sử tìm kiếm của người dùng. Người dùng có thể xóa lịch sử tìm kiếm.



Hình 4.21: Màn hình danh sách video của ca sĩ đã đăng ký

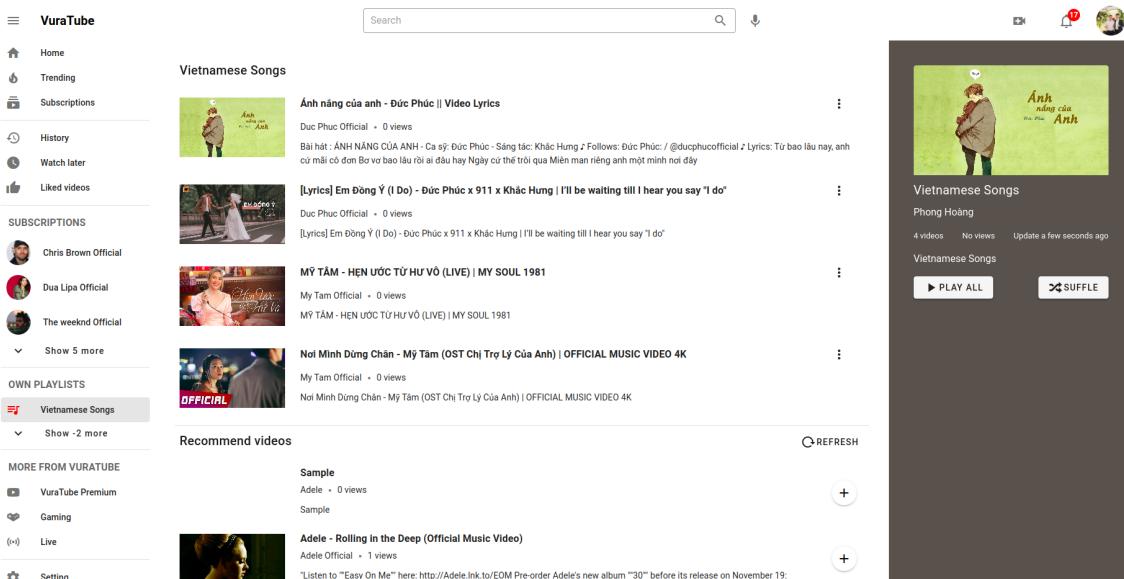
- Hình 4.21 là màn hình danh sách video của ca sĩ mà người dùng đã đăng ký. Người dùng có thể bấm vào từng video để xem.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.22: Màn hình danh sách video thịnh hành

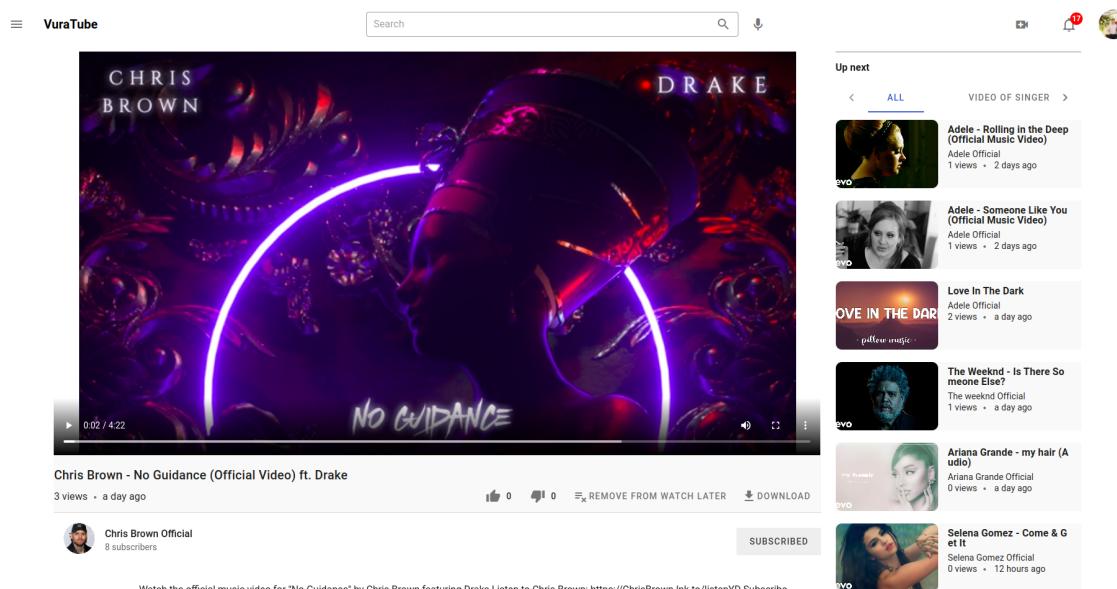
- Hình 4.22 là màn hình danh sách video thịnh hành. Thuật toán xác định video thịnh hành được trình bày chi tiết tại chương 5.3.



Hình 4.23: Màn hình danh sách video playlist video

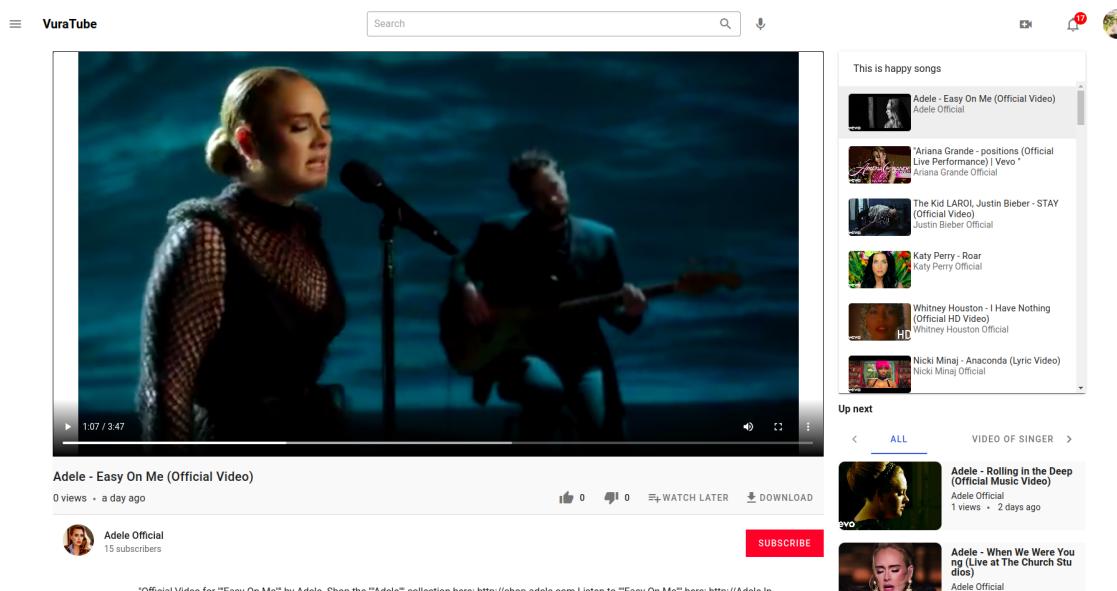
- Hình 4.23 là màn hình danh sách video trong playlist do người dùng tự tạo. Người dùng có thể trộn bài và phát danh sách phát. Hệ thống gợi ý những video cùng thể loại với video danh sách video đã tạo, người dùng có thể làm mới những video gợi ý và thêm nó vào danh sách phát.

b, Các màn hình xem video và kênh ca sĩ



Hình 4.24: Màn hình xem video

- Hình 4.24 là màn hình xem video. Tại màn hình này, người dùng có thể vừa xem video, vừa tương tác với nó bằng cách like, dislike, bình luận dưới mỗi video. Ngoài ra, người dùng có thể đăng ký channel của ca sĩ. Sau khi xem xong video, hệ thống sẽ tự động chuyển video tiếp theo. Thuật toán gợi ý video tiếp theo được giải thích ở chương 5.5.

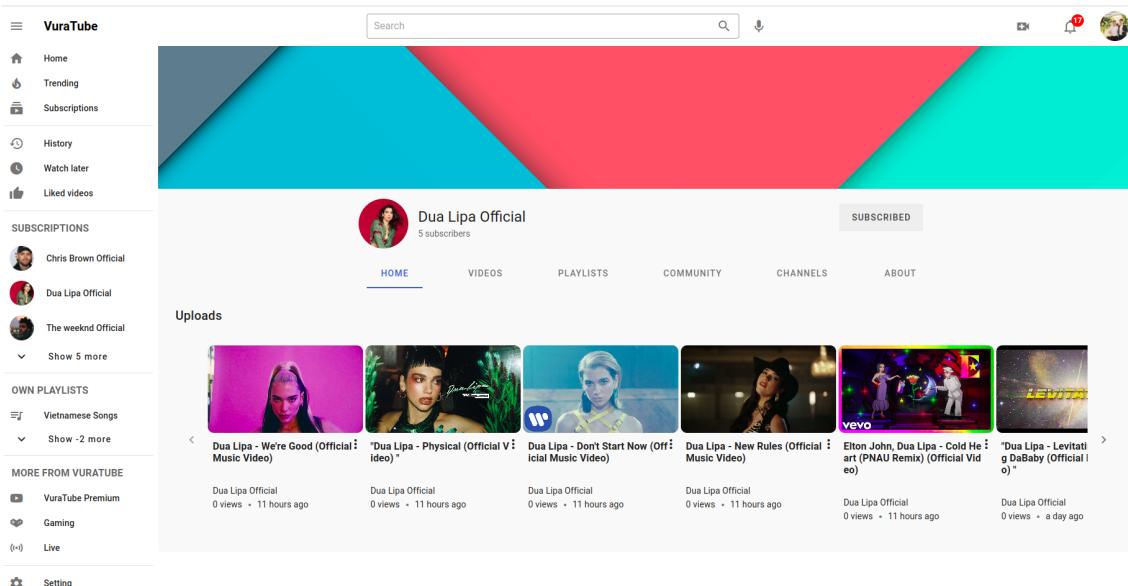


Hình 4.25: Màn hình xem video từ danh sách phát

- Hình 4.25 là màn hình xem video từ danh sách phát. Màn hình này có các chức năng tương tự màn hình xem video thông thường, tuy nhiên sau khi phát xong

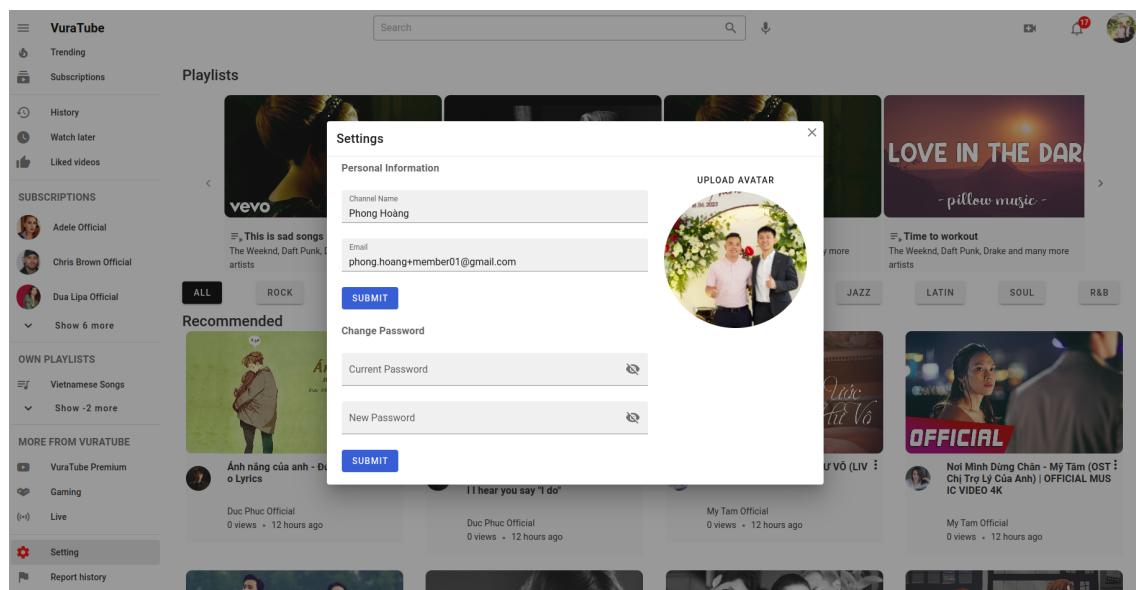
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

video hiện tại sẽ phát tiếp video có trong danh sách phát. Có ba loại danh sách phát: danh sách do hệ thống gợi ý, danh sách phát do người dùng tự tạo, danh sách phát xem sau.



Hình 4.26: Màn hình kênh của ca sĩ

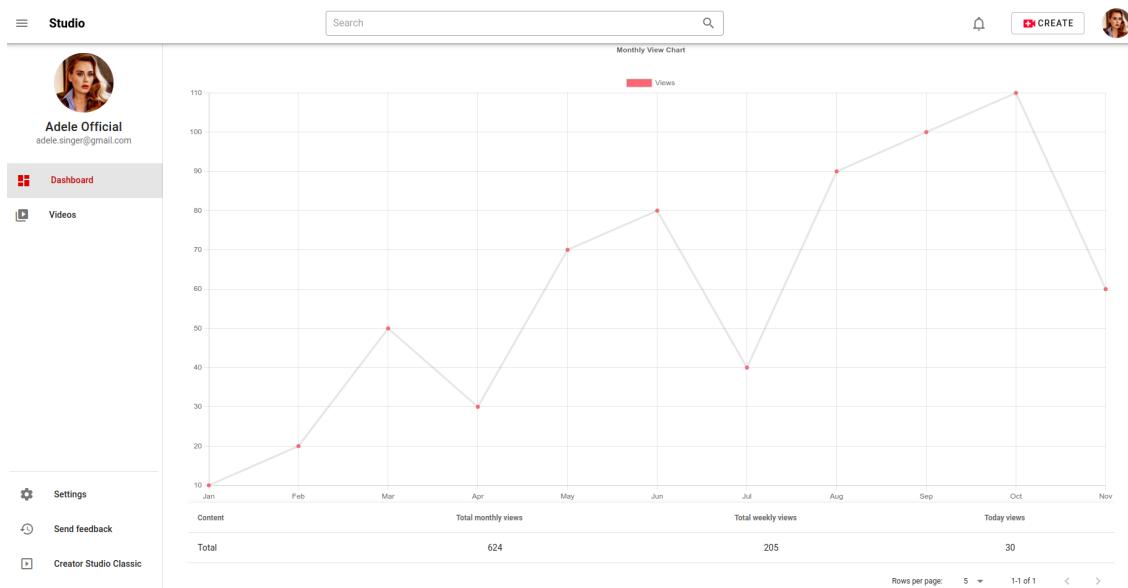
- Hình 4.26 là màn hình kênh của ca sĩ. Màn hình này hiển thị thông tin về channel của ca sĩ như số lượng người đăng ký, danh sách video của ca sĩ.



Hình 4.27: Màn hình cài đặt thông tin cá nhân

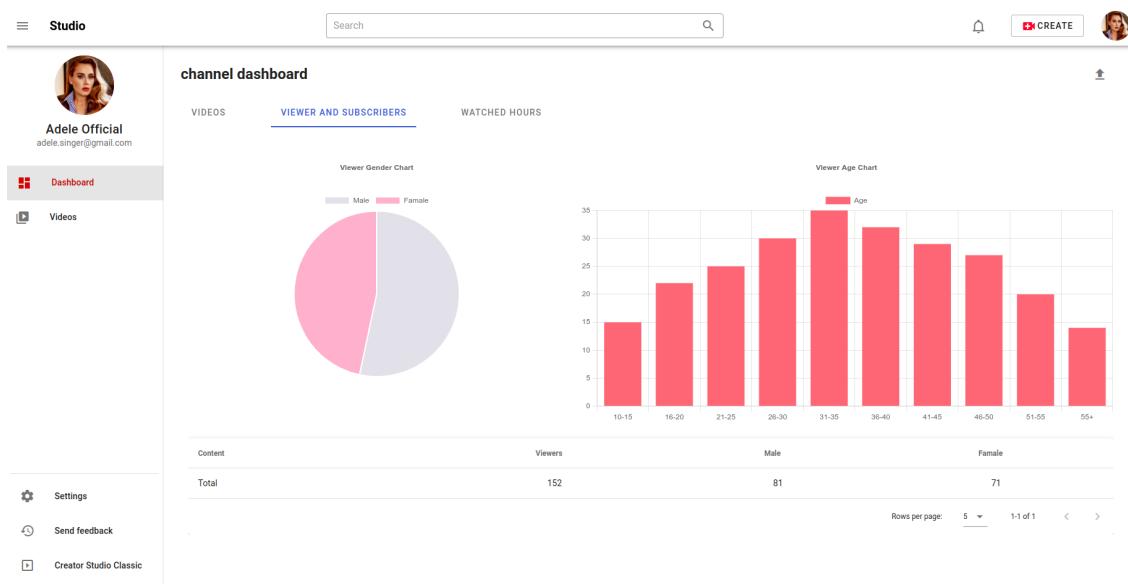
- Hình 4.27 là cài đặt thông tin cá nhân. Tại đây, người dùng có thể cập nhật tên, mật khẩu và ảnh avatar.

c, Các màn hình thống kê cho ca sĩ



Hình 4.28: Màn hình thống kê lượt xem

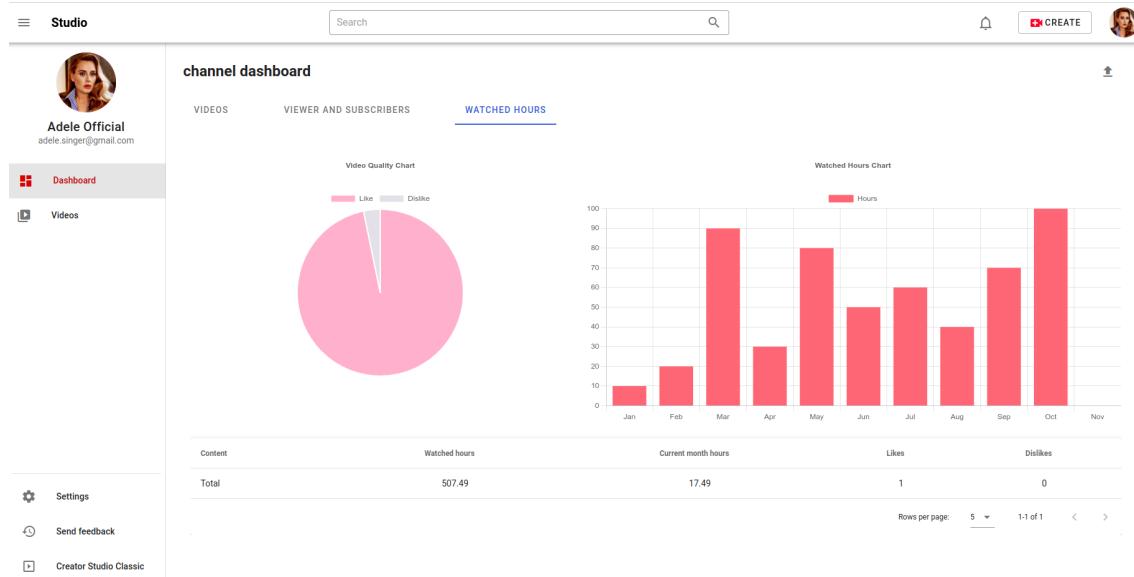
- Hình 4.28 là màn hình thống kê lượt xem của kênh. Hệ thống cung cấp số liệu hiển thị thống kê hàng tuần, tháng, năm từ đó ca sĩ có thể thấy được tốc độ tăng views ở mỗi thời điểm.



Hình 4.29: Màn hình phân tích người xem

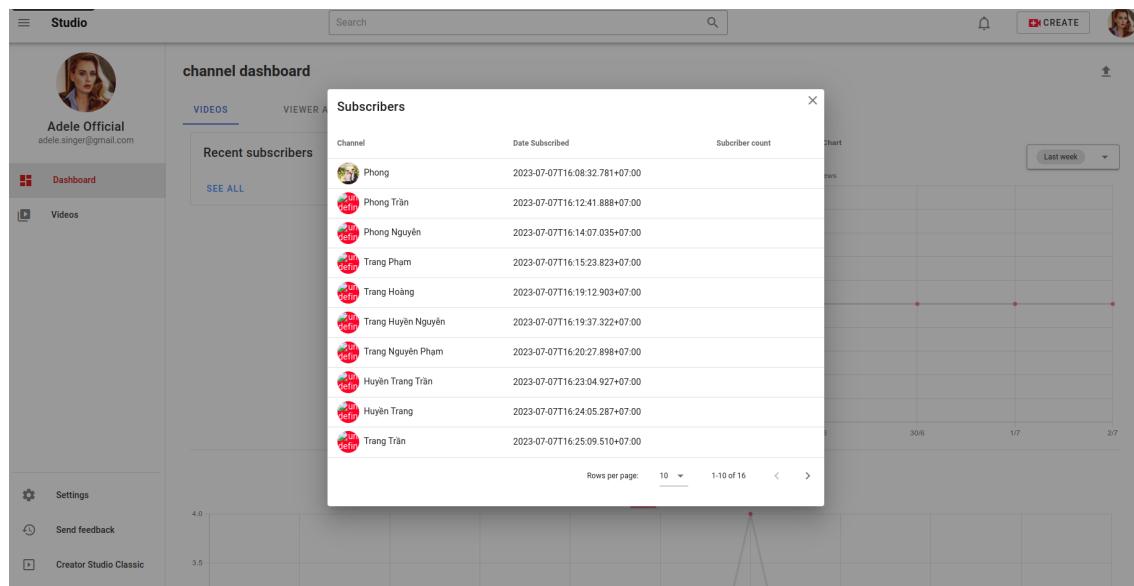
- Hình 4.29 là màn hình phân tích người xem. Hệ thống sẽ phân tích người dùng trên những khía cạnh như số lượng người xem là nam, nữ, tỷ lệ giới tính. Ngoài ra, hệ thống phân tích số lượng người dùng ở từng độ tuổi.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.30: Màn hình phân tích chất lượng nội dung video

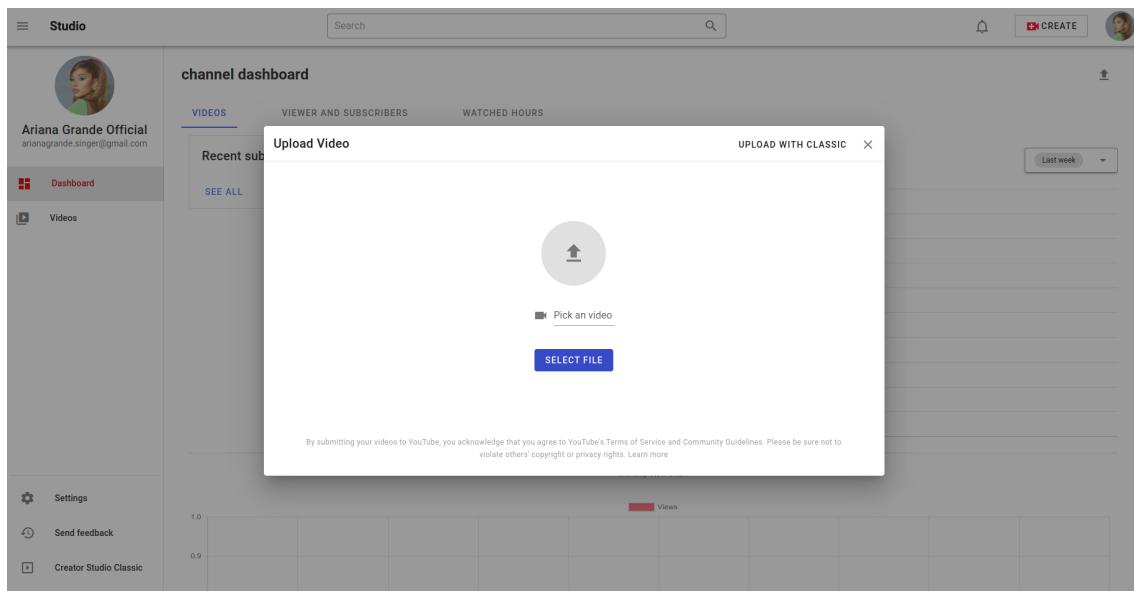
- Hình 4.30 là màn hình phân tích chất lượng nội dung video. Tại đây, ca sĩ có thể biết được tổng số lượng like trên số lượng dislike của video là bao nhiêu, tổng thời gian xem video theo từng tháng.



Hình 4.31: Màn hình danh sách người đăng ký

- Hình 4.31 là màn hình danh sách người đăng ký. Danh sách gồm những thông tin tên và avatar người đăng ký.

d, Các màn hình quản lý video



Hình 4.32: Màn hình danh đăng tải video

- Hình 4.32 là màn hình đăng tải video. Ca sĩ tiến hành upload file, sau khi upload xong, hệ thống sẽ điều hướng đến màn hình nhập các trường thông tin khác của video như tiêu đề, thể loại, mô tả, ảnh thumbnail, hashtag.

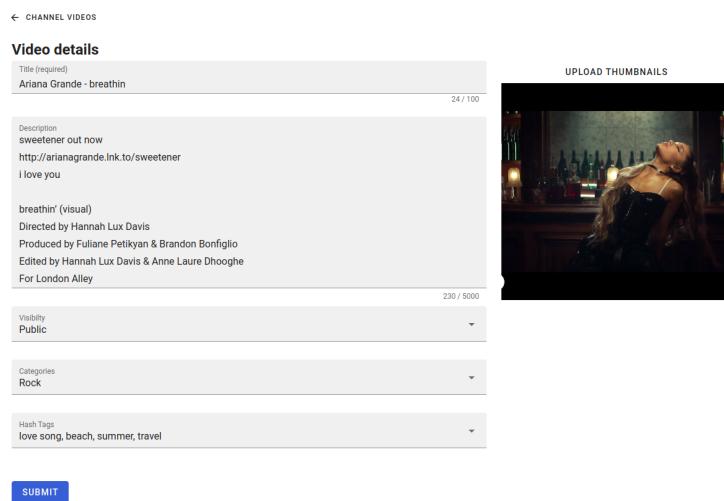
channel videos					
	UPLOADS	LIVE	Filter		
Video	Visibility	Views	Comments	Likes (vs. dislikes)	Actions
Ariana Grande - breathin	is_public	0	0	0 0	
Ariana Grande, Doja Cat - motive (audio)	is_public	0	0	0 0	
Ariana Grande - break up with your girlfriend, I'm bored (Official Video)	is_public	0	0	0 0	
Ariana Grande - thank u, next (Official Lyric Video)	is_public	0	0	0 0	
Ariana Grande - Greedy	is_public	0	0	0 0	
Ariana Grande, The Weeknd - Love Me Harder (Lyric Video)	is_public	0	0	0 0	
Ariana Grande, The Weeknd - off the table (audio)	is_public	0	0	0 0	
Ariana Grande - my hair (Audio)	is_public	0	0	0 0	
Ariana Grande - safety net ft. Ty Dolla \$ign (Official Live Performance) Vevo	is_public	0	0	0 0	
"Ariana Grande - positions (Official Live Performance) Vevo"	is_public	0	0	0 0	

Hình 4.33: Màn hình danh sách video

- Hình 4.33 là màn hình danh sách các video của ca sĩ trong hệ thống. Tại màn hình này mô tả rõ các thông tin của từng video như tiêu đề, lượt xem, lượt bình luận, trạng thái. Ca sĩ có thể tìm kiếm thông tin video bằng thanh tìm kiếm ở trên bảng. Ngoài ra ca sĩ có thể click vào các biểu tượng bên phải của từng

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

video để chỉnh sửa, xem và xóa video.



Hình 4.34: Màn hình xem chi tiết và chỉnh sửa video

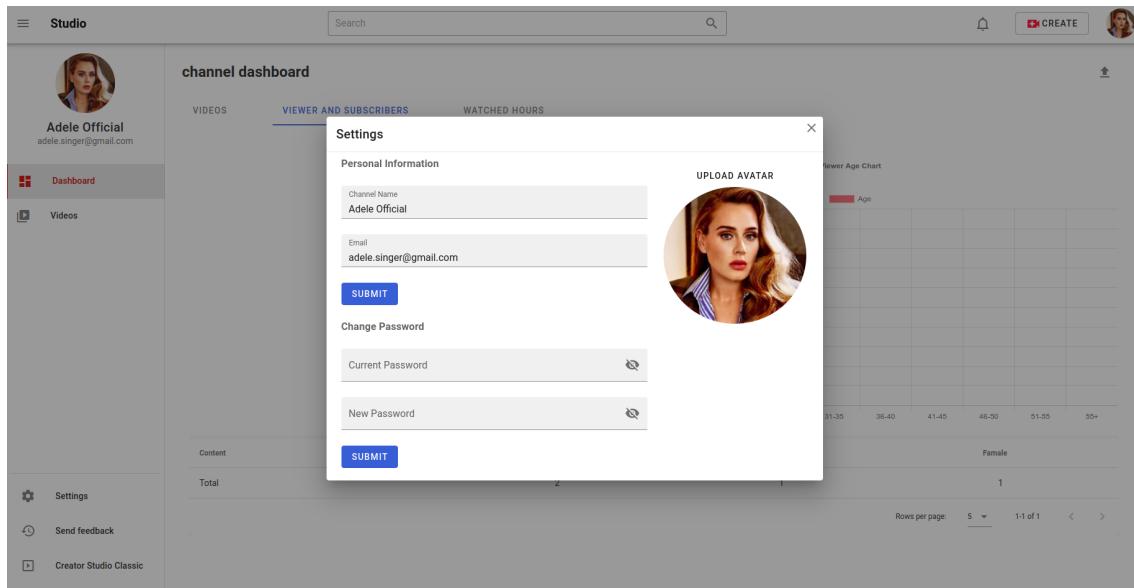
- Hình 4.34 là màn hình xem chi tiết và chỉnh sửa video. Ca sĩ có thể chỉnh sửa hầu hết các trường thông tin như tiêu đề, thể loại, mô tả, ảnh thumbnail, ... ngoại trừ file video.

Comment	Video	Commented At	User	Status	Actions
Someone Like You is nice song !!	Adele - Someone Like You (Official Music Video)	a minute ago	Phong Hoàng	Unanswered	
before its release on November 19: https://www.adele.com	Adele - Someone Like You (Official Music Video)	7 minutes ago	Huyền Trang	Unanswered	
so cute video songss	Adele - Set Fire To The Rain (Live at The Royal Albert Hall)	8 minutes ago	Huyền Trang	Unanswered	
unbelievable video songs	Adele - I Drink Wine (Official Lyric Video)	11 minutes ago	Phong Hoàng	Unanswered	
"Listen to "Easy On Me" here: http://Adele.Ink.to/EOM Pre-order Adele's	Adele - Rolling in the Deep (Official Music Video)	7 minutes ago	Huyền Trang	Answered	
nice video i ever hear	Adele - Skyfall (Official Lyric Video)	6 minutes ago	Huyền Trang	Answered	

Hình 4.35: Màn hình danh sách bình luận

- Hình 4.35 là màn hình danh sách bình luận. Tại màn hình này, ca sĩ có thể xem được nội dung bình luận về video của người dùng, gồm những thông tin như nội dung bình luận, tên người dùng, tên video, trạng thái của bình luận: đã được ca sĩ trả lời hay chưa.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.36: Màn hình cài đặt thông tin ca sĩ

- Hình 4.36 là màn hình cài đặt thông tin ca sĩ. Tại màn hình này, ca sĩ có thể thay đổi tên channel, password và ảnh avatar.

4.4 Kiểm thử

4.4.1 Kiểm thử chức năng đăng tải video

Bảng 4.25: Các trường hợp kiểm thử cho chức năng Đăng tải video

Mục đích	Đầu vào	Đầu ra mong muốn	Kết quả thực tế	Đánh giá
Kiểm thử trường hợp chọn sai kiểu file	Chọn file đuôi .png cho trường video	Thông báo chọn sai kiểu file	Thông báo chọn sai kiểu file	Đạt
Kiểm thử bỏ trống trường tiêu đề	Bỏ trống trường tiêu đề	Thông báo lỗi "Trường bắt buộc nhập"	Thông báo lỗi "Trường bắt buộc nhập"	Đạt
Kiểm thử độ dài trường mô tả	Mô tả video chứa 1000 ký tự	Thông báo lỗi "Giới hạn ký tự cho trường mô tả, dưới 800 ký tự"	Thông báo lỗi "Giới hạn ký tự cho trường mô tả, dưới 800 ký tự"	Đạt
Kiểm thử trường hợp chọn file img có kích thước không đạt chuẩn	Chọn file thumbnail có kích thước 800*800	Thông báo lỗi "Bắt buộc chọn file lớn hơn 1280*800"	Thông báo lỗi "Bắt buộc chọn file lớn hơn 1280*800"	Đạt

Bảng 4.25 – Còn nữa ở trang tiếp

Bảng 4.25 – Tiếp tục từ trang trước

Kiểm thử thời gian đặt lịch phát hành lớn hơn thời điểm hiện tại	Chọn trường thời gian đặt lịch với thời điểm trong quá khứ	Thông báo thời gian không hợp lệ, yêu cầu người dùng nhập lại	Thông báo thời gian không hợp lệ, yêu cầu người dùng nhập lại	Đạt
Kiểm thử trường hợp bỏ trống status cho video	Bỏ trống trường status	Thông báo trường status bắt buộc, yêu cầu ca sĩ nhập	Thông báo trường status bắt buộc, yêu cầu ca sĩ nhập	Đạt
Kiểm thử nhập đúng và đủ các trường thông tin	Nhập đúng và đủ các trường	Thông báo đăng tải video thành công và chuyển về màn hình dashboard	Thông báo đăng tải video thành công và chuyển về màn hình dashboard	Đạt

Chức năng này được kiểm thử sử dụng kỹ thuật phân vùng tương đương. Có 7 vùng tương đương cho dữ liệu đầu vào của chức năng đăng tải video, bao gồm 1 vùng tương đương hợp lệ và 6 vùng tương đương không hợp lệ. Từ đó, ta đưa ra được 7 trường hợp kiểm thử như bảng 4.25 trong đó 6 trường hợp kiểm thử đầu tiên tương ứng với vùng tương đương không hợp lệ. Trường hợp kiểm thử cuối cùng tương đương với vùng tương đương hợp lệ.

4.4.2 Kiểm thử chức năng Tạo danh sách phát

Bảng 4.26: Các trường hợp kiểm thử cho chức năng Tạo danh sách phát

Mục đích	Đầu vào	Đầu ra mong muốn	Kết quả thực tế	Đánh giá
Kiểm thử trường hợp bỏ trống trường tiêu đề	BỎ TRỐNG TRƯỜNG TIÊU ĐỀ	Thông báo "Trường bắt buộc, yêu cầu người dùng nhập"	"Trường bắt buộc, yêu cầu người dùng nhập"	Đạt
Kiểm thử trường hợp số ký tự của tiêu đề dài hơn 256 ký tự	NHẬP TIÊU ĐỀ CHỨA 300 KÝ TỰ	Thông báo tiêu đề không hợp lệ, dài hơn 256 ký tự	Thông báo tiêu đề không hợp lệ, dài hơn 256 ký tự	Đạt
Kiểm thử trường hợp bỏ trống trường trạng thái playlist	BỎ TRỐNG TRƯỜNG TRẠNG THÁI PLAYLIST	Thông báo "Trường bắt buộc, yêu cầu người dùng nhập"	Thông báo "Trường bắt buộc, yêu cầu người dùng nhập"	Đạt
Kiểm thử trường hợp nhập đúng và đủ các trường	NHẬP ĐÚNG VÀ ĐỦ CÁC TRƯỜNG	Thông báo tạo danh sách thành công, chuyển đến trang chủ	Thông báo tạo danh sách thành công, chuyển đến trang chủ	Đạt

Chức năng này được kiểm thử sử dụng kỹ thuật phân vùng tương đương. Có 4 vùng tương đương cho dữ liệu đầu vào của chức năng đăng tải video, bao gồm 1 vùng tương đương hợp lệ và 3 vùng tương đương không hợp lệ. Từ đó, ta đưa ra

được 4 trường hợp kiểm thử như bảng 4.26 trong đó 3 trường hợp kiểm thử đầu tiên tương ứng với vùng tương đương không hợp lệ. Trường hợp kiểm thử cuối cùng tương đương với vùng tương đương hợp lệ.

4.4.3 Kiểm thử chức năng tìm kiếm video và ca sĩ

Bảng 4.27: Các trường hợp kiểm thử cho chức năng Tìm kiếm video và ca sĩ

Mục đích	Đầu vào	Đầu ra mong muốn	Kết quả thực tế	Đánh giá
Kiểm thử trường hợp nhập 1 chuỗi ký tự hợp lệ	Nhập chuỗi ký tự "Adele"	Chuyển hướng đến trang danh sách kết quả tìm kiếm, hiển thị kết quả	Chuyển hướng đến trang danh sách kết quả tìm kiếm, hiển thị kết quả	Đạt
Kiểm thử trường hợp nhập chuỗi có ký tự đặc biệt	Nhập chuỗi có ký tự đặc biệt	Chuyển hướng đến trang kết quả tìm kiếm, thông báo không tìm thấy kết quả hợp lệ	Chuyển hướng đến trang kết quả tìm kiếm, thông báo không tìm thấy kết quả hợp lệ	Đạt
Kiểm thử trường hợp nhập chuỗi dài hơn số ký tự cho phép	Nhập chuỗi có 300 ký tự	Hiển thị thông báo nhập quá 256 ký tự, yêu cầu người dùng nhập lại	Hiển thị thông báo nhập quá 256 ký tự, yêu cầu người dùng nhập lại	Đạt
Kiểm thử trường hợp nhập các từ khóa để tìm kiếm các video có cùng chủ đề	Nhập từ khóa "happy song"	Chuyển hướng đến trang kết quả tìm kiếm, hiển thị video có liên quan đến từ khóa	Chuyển hướng đến trang kết quả tìm kiếm, hiển thị không có kết quả phù hợp	Không đạt

Chức năng này được kiểm thử sử dụng kỹ thuật phân vùng tương đương. Có 4 vùng tương đương cho dữ liệu đầu vào của chức năng đăng tải video, bao gồm 1 vùng tương đương hợp lệ và 3 vùng tương đương không hợp lệ. Từ đó, ta đưa ra được 4 trường hợp kiểm thử như bảng 4.27 trong đó trường hợp kiểm thử đầu tiên tương ứng với vùng tương đương hợp lệ, đạt kết quả như kỳ vọng. 2 trường hợp kiểm thử tiếp theo tương ứng với vùng tương đương không hợp lệ, đạt kết quả như kỳ vọng. Tuy nhiên, trường hợp kiểm thử cuối cùng ứng với vùng tương đương không hợp lệ không đạt kết quả như kỳ vọng. Nguyên nhân là do hệ thống hiện tại chỉ xử lý tìm kiếm theo tên của video và ca sĩ, nên việc tìm kiếm các video có liên quan đến từ khóa không đưa ra được video phù hợp.

4.5 Triển khai

Ứng dụng bên phía back end được triển khai lên AWS EC2[6] dưới sự hỗ trợ của gem Capistrano[7] trong Rails. Dưới đây là các bước triển khai :

1. Tạo một instance EC2, phân quyền người dùng truy cập.
2. Cài đặt các package cần thiết cho hệ thống: Cài đặt rvm, ruby, mysql, redis, ...
3. Ở local, cài đặt cấu hình Capistrano, cài đặt các gem cần thiết, config file deploy
4. Cấu hình Nginx và Puma trên server.

Bảng 4.28 là chi tiết môi trường triển khai ứng dụng:

Bảng 4.28: Chi tiết về môi trường triển khai ứng dụng

Thông số	Mô tả
Domain	vuratube.site
RAM	2GB
CPU	Architecture: x86_64 CPU op-mode(s): 32-bit, 64-bit Byte Order: Little Endian CPU(s): 1 On-line CPU(s) list: 0 Thread(s) per core: 1 Model name: Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz L1d cache: 32 KiB L1i cache: 32 KiB L2 cache: 256 KiB L3 cache: 30 MiB
HĐH	Ubuntu 20.04.6 LTS
Package	rbenv: 1.2.0-67-g3112172 ruby: 3.1.2p20 mysql: 8.0.33-0ubuntu0.20.04.2 redis: 7.0.11 nginx: 1.18.0

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

5.1 Sử dụng kiến trúc 3 lớp kết hợp với Single Page Application

5.1.1 Đặt vấn đề

Khi sử dụng kiến trúc MVC truyền thống và xây dựng giao diện ứng dụng dưới dạng Multiple Page Application (MPA), toàn bộ phần xử lý logic nghiệp vụ và phần giao diện đều nằm ở phía server. Server ngoài việc xử lý các nghiệp vụ chính, còn phải chịu trách nhiệm nhận, xử lý render và phản hồi cho các yêu cầu tải trang, điều hướng giữa các trang. Ngoài ra, khi chuyển trang hay có sự thay đổi dữ liệu hiển thị, các trang đều phải tải lại toàn bộ, kể cả những phần không cần thiết. Những điều này làm tăng thêm gánh nặng phía server, tiêu tốn tài nguyên của server cũng như băng thông mạng (do tăng kích thước thông điệp phản hồi), đồng thời làm tăng thời gian phản hồi yêu cầu người dùng, dẫn đến trải nghiệm không tốt ở người dùng.

5.1.2 Giải pháp và kết quả đạt được

Để giải quyết những hạn chế nêu trên, hệ thống đã được xây dựng sử dụng kiến trúc 3 lớp. Điều này cho phép chúng ta có thể phân chia hệ thống thành từng lớp, mỗi lớp đảm nhận một chức năng khác nhau và riêng biệt. Dựa vào sự phân chia này, chúng ta có thể tách biệt phần giao diện với phần xử lý logic nghiệp vụ, và phát triển nó dưới dạng Single Page Application (SPA) sử dụng VueJS. Với SPA, các trang được build vào một trang HTML duy nhất và được tải lên trong lần tải trang đầu tiên, các logic render hay điều hướng giữa các trang đều được thực hiện ở phía client. Khi chuyển trang hay có sự thay đổi về dữ liệu hiển thị, hệ thống cũng chỉ cần lấy dữ liệu mới thông qua các lời gọi RESTful API và cập nhật lại giao diện, mà không cần thiết phải tải lại toàn bộ trang. Nhờ đó, chúng ta giảm được gánh nặng cho phía server, giảm thời gian phản hồi và cải thiện trải nghiệm người dùng.

Ngoài ra, việc sử dụng kiến trúc 3 lớp cũng đem lại nhiều lợi ích và sự linh hoạt trong quá trình phát triển và triển khai. Nhờ đã tách biệt giữa giao diện và phần xử lý nghiệp vụ, chúng ta có thể lựa chọn triển khai hai phần này trên cùng một máy chủ hoặc trên các máy chủ khác nhau. Điều này làm cho việc triển khai linh hoạt hơn, thuận lợi cho việc mở rộng hệ thống sau này. Hệ thống được chia thành từng lớp, mỗi lớp có chức năng riêng biệt nên dễ bảo trì, dễ thay đổi và có tính tái sử dụng.

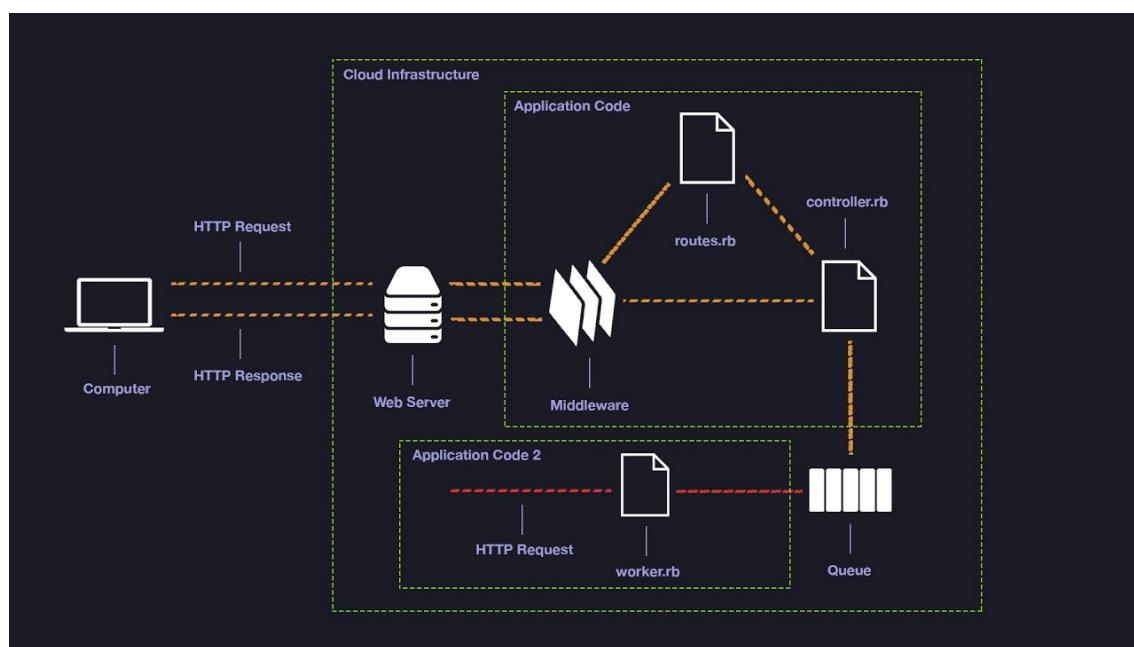
5.2 Phát triển chức năng đặt lịch phát hành video cho ca sĩ

5.2.1 Đặt vấn đề

Với các chức năng như hiện tại, người dùng có thể đăng tải video ngay lập tức hoặc đặt lịch đăng tải vào thời điểm trong tương lai. Chức năng đăng tải ngay tại thời điểm hiện tại được thực hiện thông qua luồng request-response thông thường. Tuy nhiên, chức năng đặt lịch phải được thực hiện ngoài luồng request-response mà phải được xử lý ở luồng riêng biệt và không ảnh hưởng đến tốc độ xử lý.

5.2.2 Giải pháp và kết quả đạt được

Từ những bài toán đó, công nghệ xử lý background-job đã được áp dụng và xử lý vấn đề. Hệ thống sử dụng gem Sidekiq[8] để xử lý các tác vụ background job.



Hình 5.1: Kiến trúc hệ thống sử dụng background job

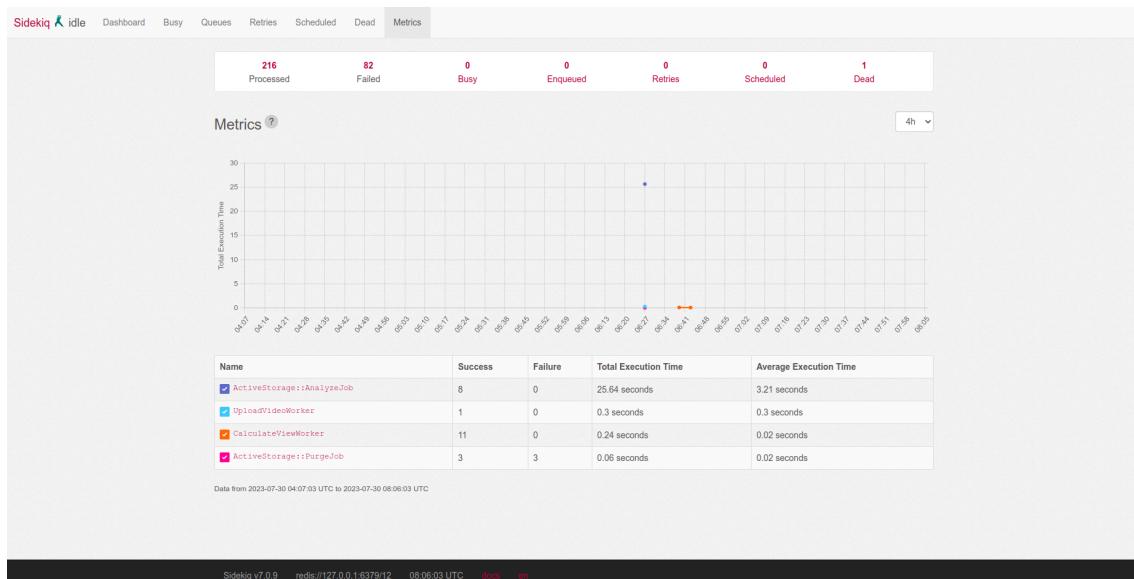
Khi sử dụng Sidekiq làm công cụ quản lý background job trong hệ thống, kiến trúc hệ thống được xây dựng như sau:

- **Job Queue:** Sidekiq sử dụng Redis làm hàng đợi (queue) để lưu trữ các công việc chạy nền. Các công việc được thêm vào hàng đợi Redis khi cần thực hiện và sau đó được xử lý theo thứ tự đến khi hoàn thành.
- **Sidekiq Worker:** Sidekiq cung cấp các worker để xử lý các công việc từ hàng đợi Redis. Mỗi worker lấy công việc từ hàng đợi và thực hiện nhiệm vụ tương ứng. Người phát triển có thể tạo các worker riêng biệt cho các loại công việc khác nhau, và Sidekiq sẽ tự động phân bổ công việc cho các worker.
- **Redis:** Redis là một cơ sở dữ liệu key-value được sử dụng bởi Sidekiq để lưu

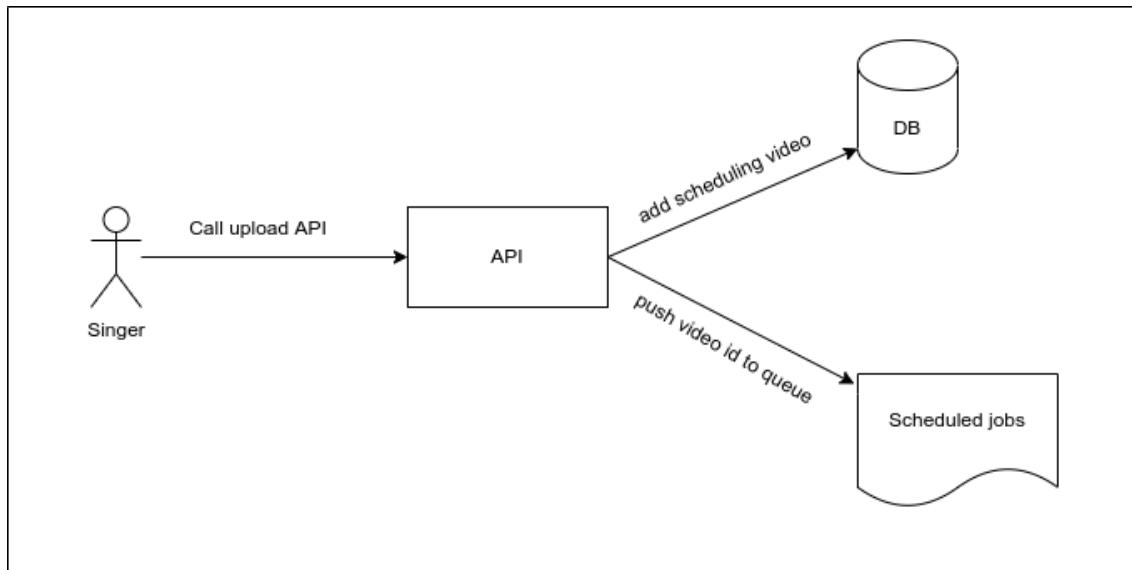
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

trữ hàng đợi và thông tin về các công việc. Redis cũng đóng vai trò là message broker để truyền thông tin giữa hàng đợi và các worker.

- **Database:** Cơ sở dữ liệu này thường được sử dụng để lưu trữ thông tin về công việc trước và sau khi chúng được xử lý bởi Sidekiq.
- **Monitoring và Retry Mechanism:** Sidekiq cung cấp giao diện quản lý và giám sát để theo dõi trạng thái của hàng đợi và các công việc. Ngoài ra, Sidekiq cũng hỗ trợ cơ chế tự động thử lại (retry) các công việc thất bại theo cách được cấu hình trước.
- **Scalability và Load Balancing:** Sidekiq cho phép mở rộng hệ thống bằng cách thêm các worker để xử lý các công việc tăng lên. Load balancing có thể được triển khai để phân phối công việc đồng đều giữa các worker.
- **Scheduling:** Sidekiq hỗ trợ lập lịch công việc chạy định kỳ bằng cách sử dụng Sidekiq Scheduler.

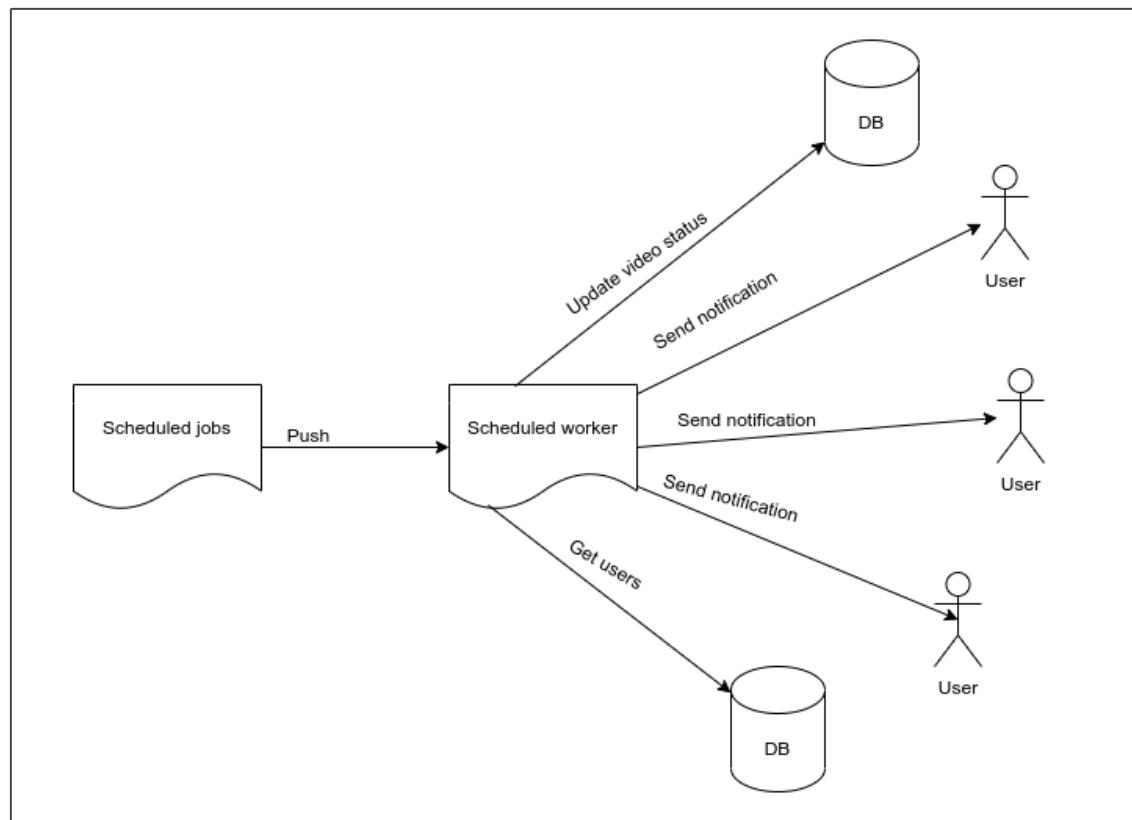


Hình 5.2: Giao diện quản lý và giám sát trạng thái các job



Hình 5.3: Quá trình lưu job vào hằng đợi

Hình 5.3 mô tả quá trình lưu job vào hằng đợi. Đầu tiên, ca sĩ tiến hành điền đúng và đủ các thông tin của video, chọn trạng thái của video và scheduling, sau đó gọi request lên hệ thống. Hệ thống sẽ lưu video vào CSDL, và lưu thời điểm phát hành video và video id vào redis, đẩy job vào hằng đợi.



Hình 5.4: Quá trình lấy job từ hằng đợi để xử lý

Hình 5.4 mô tả quá trình lấy job từ hằng đợi để xử lý. Vào đúng thời điểm đã lập

lịch, sidekiq server sẽ lấy job ra từ hằng đợi, khởi động worker và gọi đến hàm xử lý, cập nhập trạng thái video sang thành public, lấy danh sách người đăng ký và gửi thông báo đến họ.

Như vậy, việc sử dụng background job với sự hỗ trợ của gem sidekiq và redis đã giải quyết được vấn đề và xây dựng thành công chức năng đặt lịch phát hành video, làm đa dạng hóa cách thức đăng tải video cho ca sĩ.

5.3 Phương pháp tính toán mức độ thịnh hành của video

5.3.1 Đặt vấn đề

Chức năng hiển thị danh sách video âm nhạc đang thịnh hành là một trong những chức năng quan trọng của ứng dụng, giúp người dùng tiếp cận được nguồn nội dung mới mẻ, bắt kịp được xu hướng nghe nhạc ngày nay. Vì vậy, đồ án đề xuất một thang điểm nhằm xác định mức độ thịnh hành của các video.

5.3.2 Giải pháp và kết quả đạt được

Mức độ thịnh hành được tính toán dựa trên những yếu tố sau:

- **Thời gian video có mặt trên hệ thống:** Video có thời gian có mặt trên hệ thống càng lâu thì mức độ thịnh hành càng giảm. Điểm thời gian (giả sử ký hiệu là $time_score$) của $video_i$ được xác định như sau :

$$time_score_i = \frac{time_i - time_{min}}{time_{max} - time_{min}}$$

Trong đó: $time_i$ là thời gian tạo $video_i$

$time_{max}$ là thời gian tạo của video gần đây nhất

$time_{min}$ là thời gian tạo của video lâu nhất trong hệ thống

- **Tổng lượng view của video:** Video có lượng view tỉ lệ thuận với điểm thịnh hành. Điểm view (giả sử ký hiệu là $view_score$) của $video_i$ được xác định bằng công thức:

$$view_score_i = \frac{view_i}{max_view}$$

Trong đó: $view_i$ là lượt view của $video_i$, max_view là lượt view lớn nhất

- **Chất lượng nội dung video:** Video có lượt like càng lớn và lượt dislike càng ít thì chất lượng nội dung video càng cao. Điểm chất lượng (giả sử ký hiệu là

quality_score) của *video_i* được xác định như sau:

$$quality_score = \frac{quality_scale_i}{quality_scale_{max}}$$

Trong đó:

$$quality_scale_i = \frac{like_numbers_i}{dislike_numbers_i}$$

$$quality_scale_{max} = \max(quality_score_1, \dots, quality_score_n)$$

Cuối cùng , điểm thịnh hành (giả sử ký hiệu là *trending_score*) của *video_i* được xác định bằng công thức:

$$trending_score_i = time_score_i + view_score_i + quality_score_i$$

Như vậy, bằng việc thực thi các cron job để cập nhật điểm thịnh hành hàng ngày, người dùng có thể coi đây như là thước đo để đánh giá và biết được những video nào đang thịnh hành.

5.4 Cải thiện chức năng cập nhật lượt xem của video

5.4.1 Đặt vấn đề

Thuộc tính lượt xem của video được coi là yếu tố quan trọng nhất trong việc đánh giá chất lượng kênh của ca sĩ cũng như ảnh hưởng rất lớn đến mức độ thịnh hành giữa các video. Vì vậy, việc xác định lượt xem video của người dùng như thế nào là hợp lệ là điều cần quan tâm khi xây dựng hệ thống.

5.4.2 Giải pháp và kết quả đạt được

a, Phương pháp xác định lượt xem của video

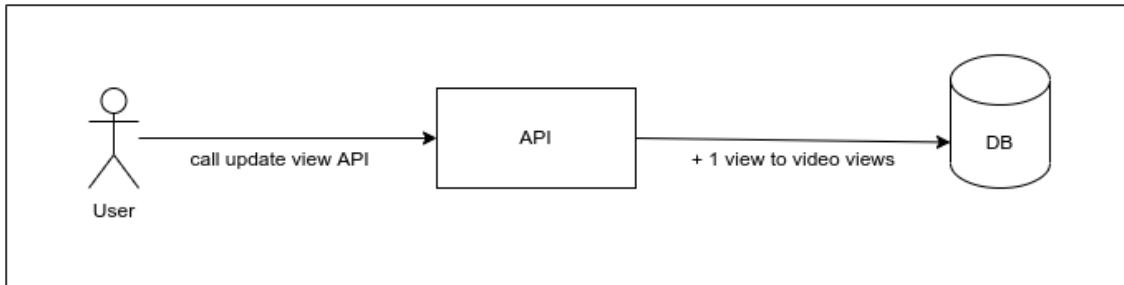
Lượt xem của video được xác định dựa trên thời gian người dùng xem video, cụ thể như sau :

- Nếu video thể loại ngắn (có thời lượng video dưới 60s) thì được tính là thêm 1 view nếu người dùng xem hết toàn bộ thời lượng video.
- Nếu video có thời lượng từ 60s đến 240s thì người dùng phải xem trên 70% thời lượng mới được tính là thêm 1 view
- Nếu video có độ dài trên 240s thì người dùng phải xem trên 50% thời lượng

mới được tính là thêm 1 view

b, Cải thiện chức năng cập nhật lượt xem bằng event-driven job

Với chức năng cập nhật lượt xem, cách đơn giản và phổ biến nhất là gọi yêu cầu cập nhật đến server, kiểm tra xem lượt xem có hợp lệ hay không, rồi gọi request cập nhật trường views của bảng video trong CSDL. Quy trình được mô tả ở hình 5.5

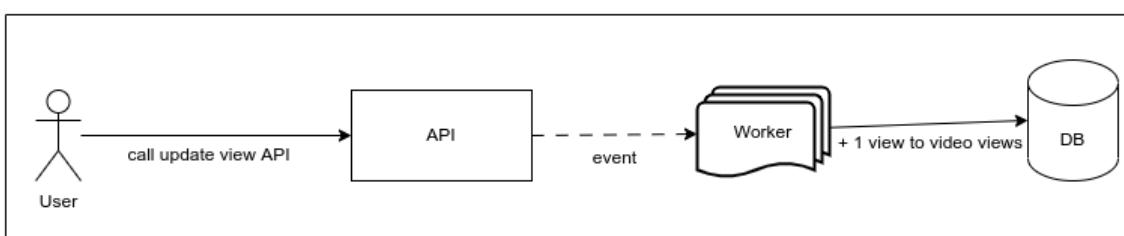


Hình 5.5: Quy trình cập nhật lượt xem thông thường

Tuy nhiên, quy trình trên gặp phải một số vấn đề như:

- **Blocking IO:** Việc thêm một view vào database làm chậm phản hồi đến người dùng, mặc dù người dùng không cần thiết phải chờ hành động này.
- **Performance:** Khi số lượng người xem video lớn, một thời điểm hệ thống phải thực thi nhiều truy vấn cùng lúc cập nhật lượt xem vào CSDL làm tăng thời gian thực thi và giảm performance của cả hệ thống.

Để giải quyết vấn đề trên, hệ thống sử dụng các công việc nền (background job) cụ thể là event-driven job, với sự hỗ trợ của gem sidekiq. Thay vì API gọi trực tiếp cập nhật CSDL, xử lý cập nhật được đẩy vào job queue, sau đó sẽ có các worker chờ sẵn sàng để thực hiện. Quy trình xử lý được mô tả ở hình 5.6



Hình 5.6: Quy trình cập nhật lượt xem khi sử dụng event-driven job

Như vậy, việc sử dụng event-driven job đã giải quyết những vấn đề sau:

- **Non-blocking IO:** Việc cập nhật view bây giờ đã gần như không ảnh hưởng tới thời gian response của người dùng do các job được xử lý đồng bộ và thời gian để đẩy job queue thường nhỏ hơn nhiều so với thời gian truy vấn cập nhật

vào CSDL.

- **Throttling:** Các job được phân bổ cho từng worker nên tại mỗi thời điểm chỉ có số lượng nhỏ các câu truy vấn được thực hiện, giúp giảm gánh nặng cho việc truy vấn vào CSDL.

5.5 Xây dựng hệ thống các chức năng gợi ý video cho người dùng

5.5.1 Đặt vấn đề

Chức năng gợi ý video là một trong những chức năng không thể thiếu đối với hệ thống xem và chia sẻ video âm nhạc. Nó giúp người dùng khám phá những video mới trong thư viện âm nhạc, mở ra cơ hội để mở rộng sự hiểu biết và thưởng thức âm nhạc. Ngoài ra, hệ thống gợi ý video dựa trên sở thích và lịch sử xem của người dùng, tạo ra những gợi ý cá nhân hóa. Điều này giúp người dùng tận hưởng nội dung phù hợp với sở thích cá nhân và tạo ra một trải nghiệm cá nhân hóa trên nền tảng xem và chia sẻ video.

5.5.2 Giải pháp và kết quả đạt được

Hiểu được vai trò của chức năng này, đồ án đã tính toán và xây dựng ba chức năng liên quan đến gợi ý video đó là:

Một là, xây dựng chức năng hiển thị danh sách video gợi ý sau khi người dùng nghe xong video hiện tại: Danh sách video này bao gồm những video được xác định như sau:

- Hệ thống truy vấn lấy ra danh sách 10 video mới phát hành của ca sĩ
- Hệ thống truy vấn lấy ra danh sách 5 video có lượt xem nhiều nhất của ca sĩ
- Hệ thống truy vấn lấy ra danh sách 10 videos vừa xem của người dùng
- Hệ thống truy vấn lấy ra danh sách 10 videos ngẫu nhiên có cùng thể loại với video đang xem hiện tại.

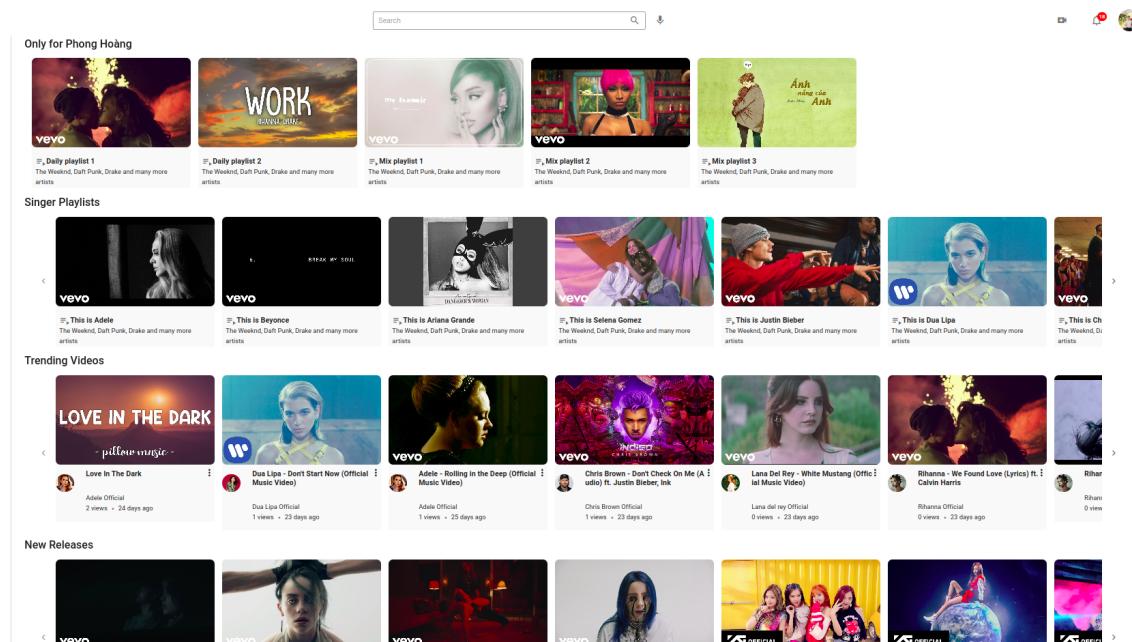
Sau đó, hệ thống trộn các video được xác định như trên, kết quả thu được danh sách video gợi ý cho người dùng sau khi nghe xong bài hát hiện tại.

Hai là, xây dựng các danh sách phát cho người dùng khám phá: Hệ thống tạo ra những danh sách phát (playlist) dành cho tất cả người dùng dựa trên những đặc điểm của ứng dụng xem video đó là :

- Danh sách phát gồm những video của từng ca sĩ.
- Danh sách phát gồm top 20 video đang thịnh hành
- Danh sách phát gồm 20 video mới phát hành
- Danh sách phát theo từng chủ đề (hash tag)

Ba là, xây dựng các danh sách phát dành riêng cho từng người dùng: Có hai loại danh sách phát dành riêng cho người dùng, bao gồm :

- Danh sách phát trộn (mix playlist): Hệ thống truy vấn và lấy ra được ba thể loại nhạc mà người dùng hay nghe nhất. Từ đó lấy ra các video thuộc ba thể loại đó. Tiếp theo, với mỗi người dùng tạo ra ba mix playlist, mỗi danh sách gồm 10 video được lấy ngẫu nhiên từ danh sách video trên. Với việc chọn ra ngẫu nhiên 10 video trong tập video có trước, hệ thống sẽ tạo ra được những danh sách phát có sự đa dạng về nội dung video, giúp tăng trải nghiệm nghe nhạc cho người dùng.
- Danh sách phát hằng ngày (daily playlist): Danh sách bao gồm mỗi loại 15 video người dùng xem nhiều nhất kết hợp video thịnh hành và video mới phát hành.



Hình 5.7: Danh sách video và playlist hệ thống gợi ý đến người dùng

Như vậy, việc tạo ra sự đa dạng của những danh sách phát, video gợi ý dựa trên lịch sử và thói quen nghe nhạc của người dùng, dựa trên các thuộc tính sẵn có của video về mức độ thịnh hành, thời gian phát hành đã làm tăng trải nghiệm nghe nhạc cho người dùng. Họ có thể vừa xem lại những video yêu thích, vừa có thể khám phá ra kho tàng video thú vị của ứng dụng. Hệ thống sử dụng chức năng cron jobs với sự hỗ trợ của gem whenever[9] để tự động hóa các tác vụ liên quan đến cập nhật danh sách phát hàng ngày, hàng tuần. Nhờ vậy, các video cung cấp đến người dùng phù hợp và bám sát nhu cầu, sở thích của họ hơn.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Với các mục tiêu đặt ra ban đầu, kết quả đạt được của đồ án là một module phần mềm dạng thức Web Application cung cấp các tính năng, tiện ích cho ca sĩ chia sẻ video âm nhạc đến với người dùng. Phần mềm được thiết kế với usecase và bảng cơ sở dữ liệu đã đủ đáp ứng được các chức năng quan trọng, cần thiết của một hệ thống chia sẻ video âm nhạc. Ca sĩ có thể đăng tải và chia sẻ các nội dung video âm nhạc, quản lý video và bình luận của người dùng. Ngoài ra, hệ thống còn cung cấp các phân tích liên quan đến lượt xem của video, nhu cầu của người dùng và chất lượng nội dung video. Người dùng có thể tìm kiếm xem video âm nhạc tùy theo sở thích nghe nhạc của mình, tự tạo ra các danh sách phát theo chủ đề tùy thích. Hệ thống cũng có giao diện dễ dùng, tối giản để làm tăng trải nghiệm xem và nghe nhạc của người dùng.

Qua việc thực hiện đồ án này, bản thân đã học hỏi, hiểu biết hơn về quy trình phát triển phần mềm, từ thiết kế, phát triển, kiểm thử đến bước triển khai ứng dụng. Từ những bài toán đặt ra của hệ thống, em đã có thể phân tích bài toán, đưa ra các giải pháp và học hỏi, áp dụng được những công nghệ mới để giải quyết vấn đề. Từ đó, năng lực giải quyết vấn đề và năng lực tự học được mài giũa và nâng cao.

6.2 Hướng phát triển

Đây là một đề tài thú vị, áp dụng được nhiều công nghệ mới và có tính thực tiễn cao. Tuy nhiên, do sự hạn chế về mặt thời gian và nhân sự, hệ thống vẫn còn hạn chế về mặt chức năng. Trong tương lai, ứng dụng có thể được cải tiến và bổ sung thêm một số chức năng để phục vụ ca sĩ và người dùng, đó là:

- Phát triển chức năng tự động kiểm duyệt nội dung video, video có nội dung vi phạm tiêu chuẩn cộng đồng sẽ không được chia sẻ đến người dùng.
- Cải thiện chức năng gợi ý video cho người dùng: Hiện tại đã có gợi ý nhưng chưa thực sự bám sát được sở thích và thói quen nghe nhạc của họ.
- Thêm chức năng kiểm tiền và thanh toán cho ca sĩ.
- Đa dạng hóa thời lượng và thể loại video: video ngắn, video trong 24h, post-card
- Cho phép người dùng chia sẻ video yêu thích trên nhiều nền tảng mạng xã hội.
- Thêm chức năng quảng cáo, nâng cấp tài khoản người dùng.

TÀI LIỆU THAM KHẢO

- [1] *Ruby on rails framework*. [Online]. Available: <https://guides.rubyonrails.org/>.
- [2] *Vuejs framework*. [Online]. Available: <https://vuejs.org/guide/introduction.html>.
- [3] *Vuetify ui library*. [Online]. Available: <https://vuetifyjs.com/>.
- [4] *Mysql*. [Online]. Available: <https://dev.mysql.com/doc/>.
- [5] *Amazon s3*. [Online]. Available: <https://docs.aws.amazon.com/s3/index.html>.
- [6] *Amazon ec2*. [Online]. Available: <https://docs.aws.amazon.com/ec2/index.html>.
- [7] *Capistrano gem*. [Online]. Available: <https://capistranorb.com/>.
- [8] *Sidekiq gem*. [Online]. Available: <https://sidekiq.org/>.
- [9] *Whenever gem*. [Online]. Available: <https://www.rubydoc.info/gems/whenever>.