

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



HỌC PHẦN: PHẦN MỀM
MÃ NGUỒN MỞ

Phần mềm Spotify

Nhóm sinh viên thực hiện:

Họ và tên	MSSV
Văn Tuấn Kiệt	3122410202
Mai Phúc Lâm	3122410207
Nguyễn Đức Duy Lâm	3122410208
Nguyễn Hữu Lộc	3122410213
Hồ Hưng Lộc	3122410219
Nguyễn Đình Thông	3122410400

Giáo viên hướng dẫn: ThS. Từ Lăng Phiêu

TP.HCM, 2025

Mục lục

Lời cảm ơn	2
1 GIỚI THIỆU	3
1.1 Giới thiệu sơ lược về đề tài	3
1.2 Lý do chọn đề tài	3
1.3 Mục tiêu	4
2 ĐẶC TẢ YÊU CẦU PHẦN MỀM	5
2.1 Yêu cầu chức năng	5
2.1.1 Người dùng	5
2.1.2 Quản lý	6
2.2 Yêu cầu phi chức năng	6
3 KIẾN TRÚC PHẦN MỀM	8
3.1 Mô hình ứng dụng	8
3.2 Công cụ	9
3.3 Kiến trúc phần mềm	11
4 Sơ đồ ERD	13

Lời cảm ơn

Trong suốt quá trình học tập môn Phần mềm mã nguồn mở và thực hiện phần mềm “Spotify”, chúng em xin gửi lời cảm ơn chân thành nhất đến tất cả những người đã hỗ trợ và đồng hành cùng chúng em trong suốt chặng đường này.

Trước tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc đến giảng viên Từ Lăng Phiêu, người đã tận tình hướng dẫn chúng em từ những bước đầu tiên cho đến khi phần mềm được hoàn thành. Thầy không chỉ cung cấp những kiến thức quý báu mà còn tạo điều kiện thuận lợi để chúng em nghiên cứu, học hỏi và áp dụng vào thực tế. Sự hướng dẫn tận tâm, những góp ý sắc sảo cùng kinh nghiệm thực tế mà thầy chia sẻ đã giúp chúng em củng cố nền tảng chuyên môn vững chắc, đồng thời vượt qua những khó khăn trong quá trình phát triển phần mềm.

Bên cạnh đó, chúng em cũng xin gửi lời cảm ơn chân thành đến các thành viên trong nhóm, những người đã luôn hỗ trợ, đồng hành và chia sẻ kinh nghiệm trong suốt quá trình thực hiện dự án. Sự hợp tác chặt chẽ, tinh thần làm việc nhóm hiệu quả cùng những trải nghiệm thực tế từ các bạn đã giúp chúng em làm quen với quy trình làm việc chuyên nghiệp, tạo nền tảng quan trọng cho con đường phát triển sau này.

Mặc dù đã nỗ lực hết mình để hoàn thành phần mềm, nhưng do thời gian và kiến thức còn hạn chế, chắc chắn vẫn còn những thiếu sót. Chúng em rất mong nhận được sự góp ý từ quý thầy cô và các bạn để có thể tiếp tục cải thiện và hoàn thiện sản phẩm hơn nữa. Những ý kiến đóng góp của thầy cô sẽ là nguồn động lực to lớn, giúp chúng em rút ra bài học kinh nghiệm quý báu và không ngừng phát triển trong tương lai.

Chúng em xin chân thành cảm ơn!

Chương 1

GIỚI THIỆU

1.1 Giới thiệu sơ lược về đề tài

Spotify là một nền tảng phát nhạc trực tuyến hàng đầu thế giới, cung cấp cho người dùng quyền truy cập vào hàng triệu bài hát, podcast và video từ các nghệ sĩ trên toàn cầu. Ra mắt vào năm 2008, Spotify đã thay đổi cách mọi người tiếp cận và thưởng thức âm nhạc, chuyển từ việc sở hữu bản ghi vật lý sang mô hình phát trực tuyến tiện lợi và hợp pháp. Spotify hoạt động trên nhiều thiết bị, bao gồm máy tính, điện thoại di động, máy tính bảng, loa thông minh, TV và ô tô, cho phép người dùng nghe nhạc mọi lúc, mọi nơi. Nền tảng này cung cấp cả phiên bản miễn phí với quảng cáo và phiên bản trả phí (Spotify Premium) với nhiều tính năng nâng cao như nghe nhạc offline, chất lượng âm thanh cao hơn và không có quảng cáo. Spotify cũng nổi tiếng với khả năng cá nhân hóa trải nghiệm người dùng thông qua các playlist được tạo tự động dựa trên thói quen nghe nhạc, như "Discover Weekly" và "Daily Mix".

1.2 Lý do chọn đề tài

rước khi Spotify xuất hiện, ngành công nghiệp âm nhạc đối mặt với nhiều thách thức, đặc biệt là vấn đề vi phạm bản quyền do việc chia sẻ nhạc trái phép trên các nền tảng như Napster. Daniel Ek và Martin Lorentzon, những người sáng lập Spotify, nhận thấy cần thiết phải tạo ra một dịch vụ âm nhạc trực tuyến hợp pháp, cung cấp trải nghiệm nghe nhạc chất lượng cao và thuận tiện, đồng thời đảm bảo quyền lợi cho các nghệ sĩ và nhà sản xuất. Mục tiêu của họ là cung cấp một giải pháp thay thế hấp dẫn hơn so với việc tải nhạc bất hợp pháp, bằng cách mang đến cho người dùng quyền truy cập tức thì vào một thư viện âm nhạc khổng lồ với chất lượng cao.

1.3 Mục tiêu

Mục tiêu chính của Spotify là "democratize audio"—dân chủ hóa việc tiếp cận âm thanh. Điều này có nghĩa là cung cấp cho người dùng trên toàn thế giới quyền truy cập dễ dàng và hợp pháp vào kho nội dung âm thanh phong phú, đồng thời tạo cơ hội cho các nghệ sĩ, dù lớn hay nhỏ, tiếp cận với khán giả toàn cầu mà không cần thông qua các kênh phân phối truyền thống. Spotify cũng đặt mục tiêu không ngừng cải thiện trải nghiệm người dùng thông qua việc ứng dụng công nghệ tiên tiến như trí tuệ nhân tạo để cá nhân hóa nội dung và đề xuất âm nhạc phù hợp với sở thích của từng cá nhân.

Ngoài ra, Spotify còn hướng đến việc mở rộng hệ sinh thái âm thanh của mình bằng cách tích hợp podcast và audiobook, biến nền tảng này thành điểm đến toàn diện cho mọi nhu cầu nghe của người dùng. Điều này không chỉ tăng cường giá trị cho người dùng mà còn tạo thêm nguồn thu nhập cho các nhà sáng tạo nội dung.

Tóm lại, Spotify được phát triển với mục tiêu cung cấp một giải pháp nghe nhạc trực tuyến hợp pháp, chất lượng cao và thuận tiện, đồng thời hỗ trợ các nghệ sĩ tiếp cận khán giả rộng rãi hơn. Nền tảng này không ngừng đổi mới và mở rộng để đáp ứng nhu cầu ngày càng cao của người dùng và thị trường âm nhạc toàn cầu.

Chương 2

ĐẶC TẢ YÊU CẦU PHẦN MỀM

2.1 Yêu cầu chức năng

2.1.1 Người dùng

- **Đăng ký và đăng nhập:** Người dùng có thể tạo tài khoản và đăng nhập vào hệ thống. Người dùng có thể đặt lại mật khẩu nếu quên mật khẩu.
- **Tìm kiếm bài hát:** Người dùng có thể tìm kiếm bài hát bằng tên bài hát, tên nghệ sĩ và có thể tìm kiếm album bằng tên bài hát, tên nghệ sĩ.
- **Phát nhạc:** Người dùng có thể phát bài hát, album, và danh sách phát yêu thích.
- **Danh sách phát:** Người dùng có thể thêm bài hát vào danh sách phát cá nhân của mình hay tải xuống.
- **Quản lý thư viện cá nhân:** Người dùng có thể xem và chỉnh sửa danh sách bài hát, bài hát yêu thích.
- **Khám phá nội dung:** Người dùng có thể xem các bài hát, album, và nghệ sĩ mới, cũng như tìm kiếm các bài hát theo thể loại.
- **Đánh giá nghệ sĩ:** Người dùng có thể đánh giá và để lại bình luận về các nghệ sĩ hoặc bài hát.

- **Quản lý thông tin cá nhân:** Người dùng có thể xem và chỉnh sửa thông tin cá nhân của mình.
- **Theo dõi nghệ sĩ:** Người dùng có thể theo dõi các nghệ sĩ yêu thích để nhận thông báo về các bài hát mới.
- **Thanh toán:** Người dùng có thể thanh toán cho các gói dịch vụ cao cấp qua các phương thức thanh toán trực tuyến.

2.1.2 Quản lý

- **Quản lý người dùng:** Quản lý có thể xem danh sách người dùng, thêm, cập nhật, và tìm kiếm người dùng theo tên, email, hoặc số điện thoại.
- **Quản lý nội dung:** Quản lý có thể thêm, sửa, xóa các bài hát, album, và danh sách phát, đồng thời quản lý các nghệ sĩ.
- **Quản lý danh sách phát chung:** Quản lý có thể tạo và duy trì các danh sách phát chung cho người dùng.
- **Thống kê:** Quản lý có thể thống kê số lượng người dùng, lượt nghe, và các bài hát phổ biến theo thời gian.
- **Quản lý hệ thống:** Quản lý có thể xem và quản lý thông tin của hệ thống, bao gồm các cấu hình và cài đặt.
- **Quản lý báo cáo:** Quản lý có thể tạo và xem các báo cáo chi tiết về hoạt động của người dùng và hệ thống.

2.2 Yêu cầu phi chức năng

Hiệu suất

- **Tìm kiếm nhanh chóng:** Các thao tác tìm kiếm phải có thời gian phản hồi dưới 5 giây.
- **Xử lý giao dịch đồng thời:** Hệ thống phải có khả năng xử lý 500 giao dịch đồng thời mà không gặp phải độ trễ.

Tính bảo mật

- **Mã hóa SSL cho thanh toán:** Đảm bảo các giao dịch tài chính được bảo vệ thông qua SSL/TLS.
- **Xác thực hai yếu tố (2FA):** Sử dụng xác thực OTP hoặc sinh trắc học để tăng cường bảo mật khi người dùng thực hiện giao dịch nhạy cảm.

Tính sẵn có

- **Hoạt động 24/7:** Hệ thống cần phải luôn sẵn sàng với mức độ uptime 99.9

Khả năng sử dụng

- **Giao diện thân thiện:** Giao diện người dùng phải đơn giản và dễ sử dụng, với các tính năng quan trọng hiển thị rõ ràng và dễ tìm.
- **Hỗ trợ đa ngôn ngữ:** Hệ thống cần cung cấp hỗ trợ cho nhiều ngôn ngữ và đảm bảo khả năng truy cập cho người dùng có nhu cầu đặc biệt.

Tính tương thích

- **Tương thích với các trình duyệt:** Ứng dụng cần phải tương thích với các trình duyệt web phổ biến như Chrome, Cốc cốc, và Edge.
- **Tương thích với thiết bị di động:** Giao diện cần phải responsive và tương thích với các hệ điều hành di động (iOS, Android).

Tính bảo trì

- **Dễ bảo trì và mở rộng tính năng:** Cần áp dụng kiến trúc modular để dễ dàng thêm hoặc thay đổi chức năng mà không ảnh hưởng đến toàn bộ hệ thống.
- **CI/CD:** Sử dụng quy trình CI/CD để tự động triển khai và cập nhật hệ thống mà không cần downtime.

Chương 3

KIẾN TRÚC PHẦN MỀM

3.1 Mô hình ứng dụng

Frontend:

- **ReactJS:** React là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook vào năm 2013, nhằm xây dựng giao diện người dùng cho các ứng dụng web. React cho phép các lập trình viên phát triển các thành phần giao diện (components) một cách hiệu quả, dễ bảo trì và tái sử dụng. Với React, người dùng có thể trải nghiệm giao diện mượt mà và dễ dàng tương tác với ứng dụng.

Backend:

- **Python - Django:** Django là một framework Python mạnh mẽ giúp phát triển các ứng dụng web nhanh chóng và dễ dàng hơn. Django cung cấp các tính năng như ORM (Object Relational Mapping) để quản lý cơ sở dữ liệu, giúp việc phát triển các API RESTful trở nên đơn giản và thuận tiện. Nó cũng hỗ trợ bảo mật và quản lý người dùng, giúp dễ dàng xử lý các yêu cầu API của người dùng.

Hệ quản trị cơ sở dữ liệu:

- **MySQL:** là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS), thuộc quyền sở hữu của Oracle, được sử dụng để quản lý và lưu trữ dữ liệu. Nó sử dụng SQL (Structured Query Language) làm ngôn ngữ chính để truy vấn và thao tác với cơ sở dữ liệu. MySQL phổ biến trong các ứng dụng web, đặc biệt là các ứng dụng sử dụng kiến trúc LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Các ứng dụng web lớn nhất như

Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

Giao thức truyền thông:

- **WebSocket:** là một giao thức truyền thông giúp cho việc thiết lập kênh truyền thông hai chiều giữa máy chủ và máy khách. WebSocket hoạt động bằng cách thiết lập kết nối HTTP liên tục với máy chủ và sau đó nâng cấp nó lên kết nối websocket hai chiều bằng cách gửi Upgrade header. WebSocket được hỗ trợ trong hầu hết các trình duyệt web hiện đại và cho các trình duyệt không hỗ trợ, chúng tôi có các thư viện cung cấp dự phòng cho các kỹ thuật khác như Comet và HTTP Long Polling.

Dịch vụ bên thứ ba:

- **AWS S3:** AWS S3 (Amazon Simple Storage Service) là một dịch vụ lưu trữ đối tượng (object storage) dựa trên đám mây, được Amazon Web Services (AWS) ra mắt vào năm 2006, nhằm cung cấp giải pháp lưu trữ dữ liệu an toàn, bền bỉ và có khả năng mở rộng cao. S3 cho phép người dùng lưu trữ bất kỳ loại dữ liệu nào, từ hình ảnh, video, đến tài liệu, dưới dạng các đối tượng (objects) trong các "bucket"(thùng chứa), với dung lượng tối đa lên đến 5TB cho mỗi đối tượng. Dịch vụ này đảm bảo độ bền 99.999999999

3.2 Công cụ

PyCharm:

- PyCharm là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) được thiết kế chuyên biệt cho lập trình Python, do JetBrains phát triển và ra mắt lần đầu tiên vào tháng 2 năm 2010. PyCharm cung cấp các tính năng mạnh mẽ như gợi ý mã thông minh, kiểm tra lỗi thời gian thực, tích hợp công cụ quản lý phiên bản (Git, SVN), và hỗ trợ phát triển web với các framework như Django và Flask. IDE này có hai phiên bản: Community (miễn phí, mã nguồn mở) và Professional (trả phí, với nhiều tính năng nâng cao). PyCharm được sử dụng rộng rãi trong các dự án phát triển phần mềm, đặc biệt là các ứng dụng Python, và tương thích với nhiều nền tảng như Windows, macOS, và Linux.

Visual Studio Code (VS Code):

- Visual Studio Code là một trình soạn thảo mã nguồn nhẹ nhưng mạnh mẽ, hỗ trợ đa nền tảng như Windows, macOS và Linux. Với các tính năng như hoàn thành mã thông minh, tích hợp Git, và hỗ trợ nhiều tiện ích mở rộng, VS Code được sử dụng để phát triển frontend của dự án Instagram clone với ReactJS. Công cụ này giúp lập trình viên dễ dàng viết mã, gỡ lỗi, và quản lý các thư viện như Ant Design, Redux, và TailwindCSS một cách hiệu quả.

Postman:

- Postman là một công cụ phổ biến được sử dụng để kiểm thử và làm việc với các API, đặc biệt là API kiểu REST. API đóng vai trò quan trọng trong việc kết nối các thành phần của ứng dụng, và Postman giúp đơn giản hóa quá trình gọi và kiểm tra API mà không cần viết mã. Công cụ này hỗ trợ tất cả các phương thức HTTP như GET, POST, PUT, PATCH, DELETE, và cho phép lưu lại lịch sử các yêu cầu (request) để sử dụng lại khi cần. Trong dự án Instagram clone, Postman được sử dụng để kiểm tra các API như đăng bài post (POST /api/post), lấy danh sách bài post (GET /api/post), và gửi tin nhắn.

MySQL Workbench:

- MySQL Workbench là một công cụ quản lý cơ sở dữ liệu MySQL, cung cấp giao diện đồ họa để phát triển, thiết kế và quản lý cơ sở dữ liệu một cách trực quan. Công cụ này cho phép dễ dàng tạo, chỉnh sửa cơ sở dữ liệu, thực hiện các thao tác như đảo ngược (reverse engineering) để tạo mô hình từ cơ sở dữ liệu hiện có, và chuyển tiếp (forward engineering) để triển khai mô hình thành cơ sở dữ liệu. Trong dự án Instagram clone, MySQL Workbench được sử dụng để thiết kế và quản lý cơ sở dữ liệu MySQL, bao gồm các bảng như users, posts, comments, và messages.

GitHub:

- GitHub là một nền tảng quản lý mã nguồn phổ biến, cho phép các lập trình viên chia sẻ, cộng tác và quản lý phiên bản mã nguồn một cách hiệu quả. Sự phát triển của GitHub bắt đầu vào ngày 19 tháng 10 năm 2007, và trang web chính thức được ra mắt vào tháng 4 năm 2008 bởi Tom Preston-Werner, Chris Wanstrath, và PJ Hyett. Microsoft đã mua lại GitHub vào tháng 6 năm 2018. Trong dự án Instagram clone, GitHub được sử dụng để lưu trữ mã nguồn, quản lý phiên bản, và hỗ trợ làm việc nhóm giữa các thành viên phát triển frontend và backend.

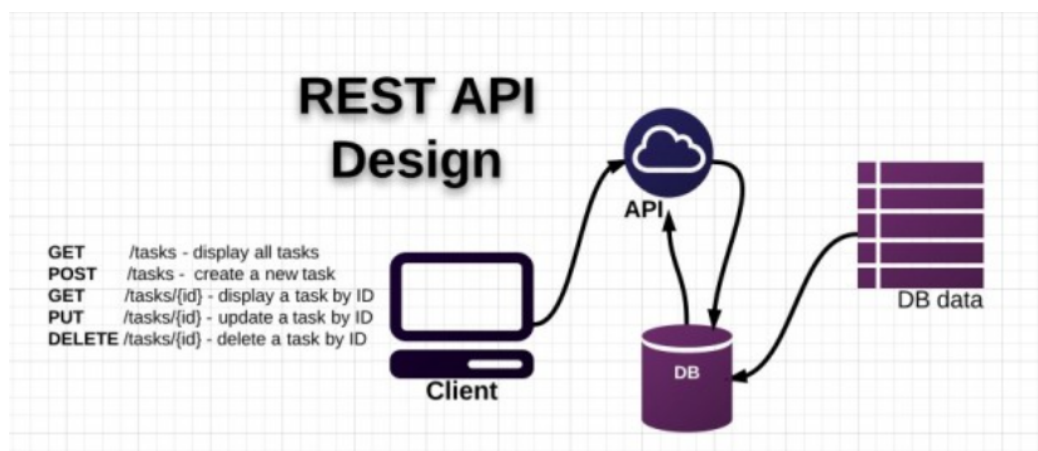
Deploy: Amazon EC2 và VPS:

- **Amazon EC2:** Amazon EC2 là dịch vụ đám mây của Amazon Web Services (AWS), cho phép người dùng thuê máy chủ ảo (instances) để triển khai và chạy ứng dụng. EC2 mang lại sự linh hoạt, khả năng mở rộng và tính sẵn sàng cao cho các ứng dụng web. Với Clone Spotify, EC2 sẽ được sử dụng để triển khai phần backend và giúp ứng dụng có thể mở rộng khi lượng người dùng tăng.
- **VPS:** VPS là một máy chủ ảo được phân chia từ một máy chủ vật lý duy nhất. Mỗi VPS có hệ điều hành riêng biệt, giúp triển khai ứng dụng và quản lý tài nguyên như một máy chủ độc lập. Với Clone Spotify, VPS sẽ được sử dụng cho việc triển khai các dịch vụ khác ngoài backend, như các ứng dụng phụ trợ hoặc lưu trữ dữ liệu không đụng đến hệ thống chính.

3.3 Kiến trúc phần mềm

RESTful API:

- **Khái niệm:** RESTful API là một tiêu chuẩn được sử dụng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) nhằm mục đích quản lý các tài nguyên (resource) một cách hiệu quả. Nó tập trung vào các tài nguyên hệ thống như tệp văn bản, hình ảnh, âm thanh, video, hoặc dữ liệu động, bao gồm các trạng thái tài nguyên được định dạng và truyền tải thông qua giao thức HTTP. RESTful API cho phép các ứng dụng giao tiếp với nhau thông qua các phương thức HTTP chuẩn, giúp đơn giản hóa việc truy cập và xử lý dữ liệu trên các nền tảng khác nhau.



Diễn giải các thành phần

- **API (Application Programming Interface):** API là một tập hợp các quy tắc và cơ chế cho phép một ứng dụng hoặc thành phần tương tác với một ứng dụng hoặc thành phần khác. API đóng vai trò như một cầu nối, giúp các ứng dụng trao đổi dữ liệu với nhau một cách dễ dàng. Dữ liệu trả về từ API thường được định dạng ở các kiểu phổ biến như JSON hoặc XML, phù hợp để tích hợp vào các ứng dụng web hoặc di động.
- **REST (REpresentational State Transfer):** REST là một kiểu kiến trúc (architectural style) được sử dụng để thiết kế API, dựa trên việc chuyển đổi trạng thái biểu diễn của tài nguyên. REST tận dụng các phương thức HTTP đơn giản như GET, POST, PUT, DELETE để thực hiện các thao tác trên tài nguyên. Thay vì sử dụng một URL để xử lý thông tin phức tạp, REST gửi các yêu cầu HTTP đến một URL cụ thể nhằm truy xuất hoặc chỉnh sửa dữ liệu, giúp giao tiếp giữa các hệ thống trở nên trực quan và hiệu quả.
- **RESTful API:** RESTful API là một tiêu chuẩn thiết kế API cho các ứng dụng web, tập trung vào việc quản lý các tài nguyên (resource) và cho phép các ứng dụng khác nhau (web, mobile, v.v.) giao tiếp với nhau. Chức năng quan trọng nhất của RESTful API là quy định cách sử dụng các phương thức HTTP (như GET để lấy dữ liệu, POST để tạo mới, PUT để cập nhật, DELETE để xóa) và cách định dạng URL để truy cập các tài nguyên. RESTful API không quy định logic mã nguồn của ứng dụng và không bị giới hạn bởi ngôn ngữ lập trình, do đó bất kỳ ngôn ngữ hoặc framework nào (như Java với Spring Boot, JavaScript với Node.js) cũng có thể được sử dụng để thiết kế một RESTful API.

Sơ đồ ERD