**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO CUỐI KỲ**

ĐỒ ÁN MÔN

ĐỒ ÁN CHUYÊN NGÀNH

**Đề tài: Xây dựng ứng dụng quản lý âm nhạc với spotify API**

*Giảng viên hướng dẫn:* ***Mai Trọng Khang***

*Sinh viên thực hiện:* ***16520062 – Vũ Ngọc Duy Anh***

Thành phố Hồ Chí Minh, tháng 6 năm 2019

# LỜI MỞ ĐẦU

Với sự phát triển rầm rộ của công nghệ thông tin, hiện nay số người sử dụng internet ở Việt Nam đã tăng lên rất cao, vì thế nhu cầu giải trí đa phương tiện đã thu hút được sự quan tâm rất lớn trong những năm gần đây, đặc biệt là nhu cầu nghe nhạc trực tuyến.  
 Hiện nay, có rất nhiều ứng dụng, phần mềm trực tuyến về âm nhạc được chia sẻ rộng rãi trên internet. Điển hình như: Youtube, Spotify, NhacCuaTui, ZingMP3, ... Tất cả đều là những trang web và phần mềm ứng dụng chia sẻ âm nhạc, bài hát nổi tiếng trên thế giới.

Đồ án với tên gọi: “Xây dựng ứng dụng quản lý âm nhạc với spotify API”, tận dụng triệt để API đã được công bố từ Spotify, phục vụ đặc biệt cho việc nghiên cứu, phát triển ứng dụng quản lý âm nhạc độc quyền từ Spotify (<https://developer.spotify.com/>). Đồ án này sẽ kế thừa và tạo ra những thứ mới mẻ, nhằm tăng tính sáng tạo, thư giản cho người sử dụng ứng dụng, tạo tiền đề cho việc nâng cấp, khai thác triệt để và phát triển thêm những tính năng thú vị khác trong tương lai.

Ứng dụng này hy vọng sẽ đem đến cho người sử dụng những trải nghiệm tuyệt vời, thư giản sau những ngày làm việc mệt mỏi. Đây là ứng dụng mang tính phi thương mại, được sự cho phép của spotify toàn quyền sử dụng API cho mục đích nghiên cứu và học tập. Xin chân thành cảm ơn Spotify, và đặc biệt gửi lời cảm ơn sâu sắc đến quý thầy/ cô giảng viên đã hướng dẫn giúp tôi hoàn thành sản phẩm này.

**NHẬN XÉT CỦA GIẢNG VIÊN**

**Mục lục**

[LỜI MỞ ĐẦU 2](#_Toc42863941)

[Chương 1: Tổng quan đề tài 5](#_Toc42863942)

[Đặt vấn đề 5](#_Toc42863943)

[Giải pháp 5](#_Toc42863944)

[Mục tiêu đề tài 5](#_Toc42863945)

[Phạm vi đề tài 5](#_Toc42863946)

[Bố cục báo cáo 6](#_Toc42863947)

[Chương 2: Tổng quan về Spotify API 6](#_Toc42863948)

[Giới thiệu 6](#_Toc42863949)

[Cách đăng ký ứng dụng sử dụng Spotify API thông qua Spotify developer 7](#_Toc42863950)

[Các loại xác thực API 8](#_Toc42863951)

[Luồng xác thực được sử dụng trong dự án và nguyên nhân được lựa chọn 12](#_Toc42863952)

[Phạm vi (Scopes) trong vấn đề xác thực người dùng thông qua Spotify API 12](#_Toc42863953)

[Chương 3: Tổng quan về Spotify Web Playback SDK 12](#_Toc42863954)

[Giới thiệu 12](#_Toc42863955)

[Vấn đề lưu ý trong xác thực: 12](#_Toc42863956)

[Chương 4: Tổng quan về ứng dụng 13](#_Toc42863957)

[Cấu trúc 13](#_Toc42863958)

[Một số chức năng tiêu biểu 13](#_Toc42863959)

[Quy trình phát triển phần mềm: 13](#_Toc42863960)

[Giới thiệu 13](#_Toc42863961)

[Chương 5: Tổng quan về Spring boot 14](#_Toc42863962)

[Lịch sử hình thành 14](#_Toc42863963)

[Ưu điểm của Spring 16](#_Toc42863964)

[Chương 6: Tổng quan front end frameworks & libraries 18](#_Toc42863965)

[Bootstrap 18](#_Toc42863966)

[jQuery 19](#_Toc42863967)

[Chương 7: Tổng quan về các công nghệ và công cụ liên quan 21](#_Toc42863968)

[Chương 8: Hình ảnh thực tế từ ứng dụng 22](#_Toc42863969)

[Chương 9: Tham khảo 22](#_Toc42863970)

# Chương 1: Tổng quan đề tài

## Đặt vấn đề

Ngày nay trong nhịp sống hối hả của con người thì việc giành thời gian để giải trí sau những giờ làm việc căng thẳng trở nên rất cần thiết. Những lo lắng về việc sử dụng đĩa CD, USB chứa những bài hát mình yêu thích, những nỗi quan tâm về việc phải giành thời gian đi tới những cửa hàng băng đĩa xa nhà. Những điều đó thật bất tiện, đôi khi càng làm cho mọi người cảm thấy mệt mỏi hơn.

## Giải pháp

Chúng tôi đã đề ra một giải pháp, một trang web về âm nhạc được công bố, giúp ích rất nhiều trong việc tận hưởng âm nhạc ngay tại nhà, ngay tại bất cứ đâu bạn muốn, chỉ cần thông qua kết nối internet là có thể dễ dàng nghe những bài hát mới nhất được cập nhật từ hệ thống spotify.

## Mục tiêu đề tài

Xây dựng website quản lý âm nhạc với spotify api, sẽ đáp ứng được những yêu cầu sau:

* Chức năng quản lý nghệ sĩ, nghệ sĩ yêu thích, nghệ sĩ nghe nhiều nhất.
* Chức năng quản lý bài hát.
* Chức năng quản lý bộ sưu tập.
* Chức năng quản lý những bài hát gần đây.
* Chức năng quản lý album.
* Chức năng quản lý bài hát đang được phát sóng.
* Cung cấp trình phát nhạc trực tuyến.

## Phạm vi đề tài

Việc xây dựng website sẽ tập trung hỗ trợ vào việc cung cấp một hệ thống âm nhạc trực tuyến như: dễ dàng tra cứu bài hát yêu thích, quản lý những bài hát, .... Phạm vi ứng dụng chỉ sử dụng như một ứng dụng phi thương mại, đã được cấp phép sử dụng API Spotify dành cho nhà phát triển phục vụ việc nghiên cứu.

## Bố cục báo cáo

* Chương 1: Tổng quan về đề tài

Giới thiệu tổng quan nội dung đề tài. Nội dung chương 1 bao gồm: lý do lựa chọn đề tài, nội dung và phạm vi của đề tài, bố cục của quyển báo cáo.

* Chương 2: Tổng quan về Spotify API

Giới thiệu tổng quan về Spotify API từ Spotify developer website và cách sử dụng.

* Chương 3: Tổng quan về Spotify Web Playback SDK

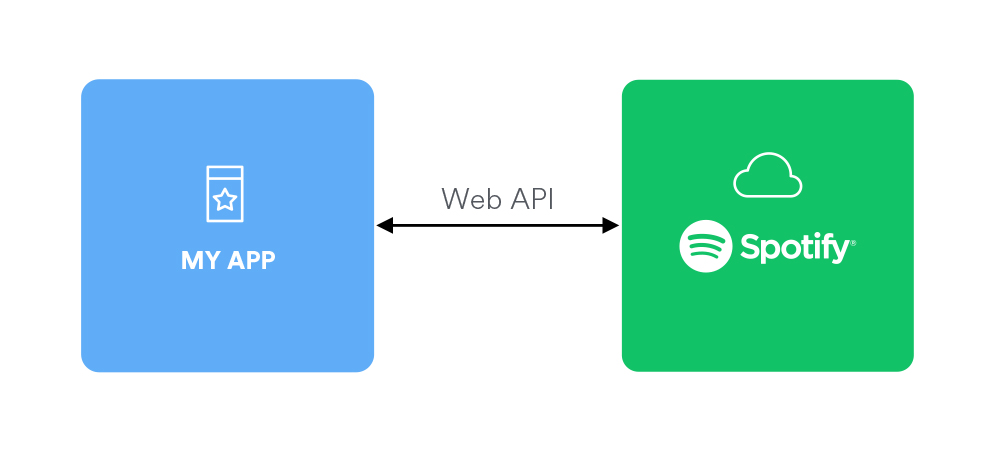
Giới thiệu tổng quan về Web Playback SDK từ Spotify deveporer website và cách sử dụng.

* Chương 4: Tổng quan về Ứng dụng
* Chương 5: Tổng quan về Spring boot – phía Backend
* Chương 6: Tổng quan Front end frameworks & libraries – phía Frontend
* Chương 7: Tổng quan về các ứng dụng và công nghệ liên quan
* Chương 8: Hình ảnh thực tế từ đề tài
* Chương 9: Tham khảo

# Chương 2: Tổng quan về Spotify API

## Giới thiệu

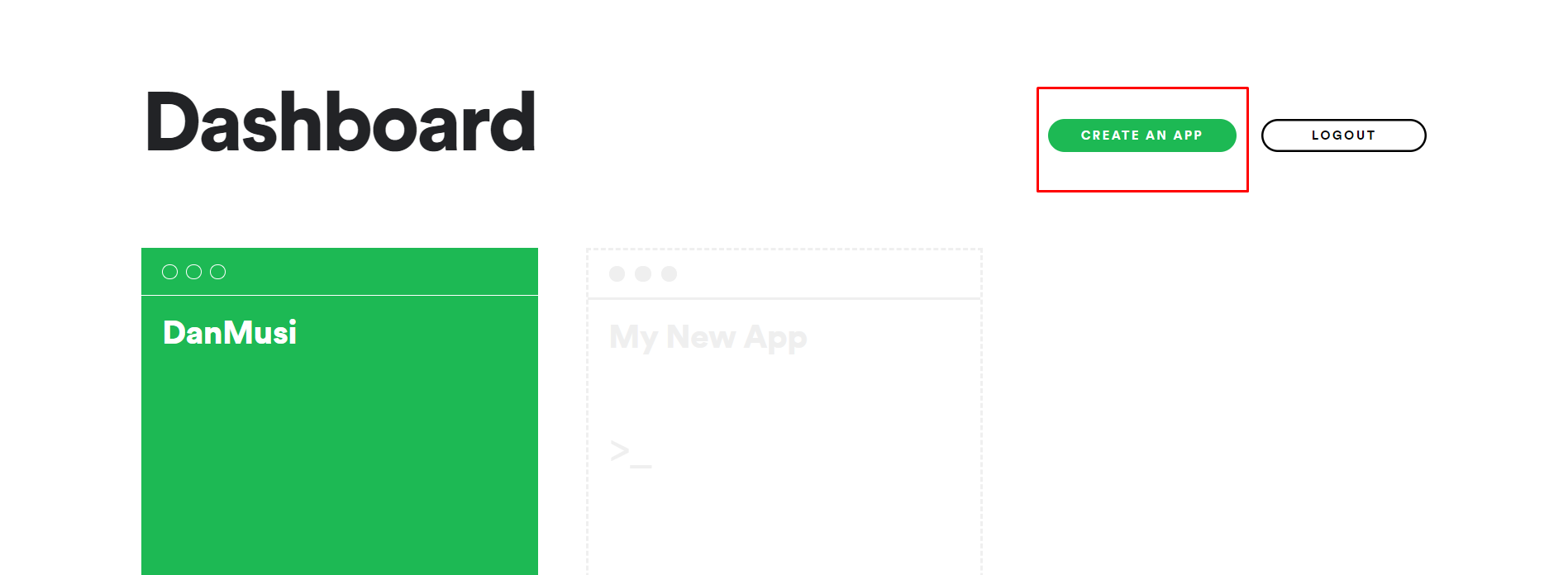
Dựa theo nguyên tắc REST cơ bản, API của spotify sẽ trả về dưới dạng JSON, trả về thông tin nghệ sĩ, albums, bài hát, ... từ Spotify Data. Web API cũng có thể cung cấp truy cập dữ liệu người dùng cá nhân liên quan thông qua việc xác thực API.



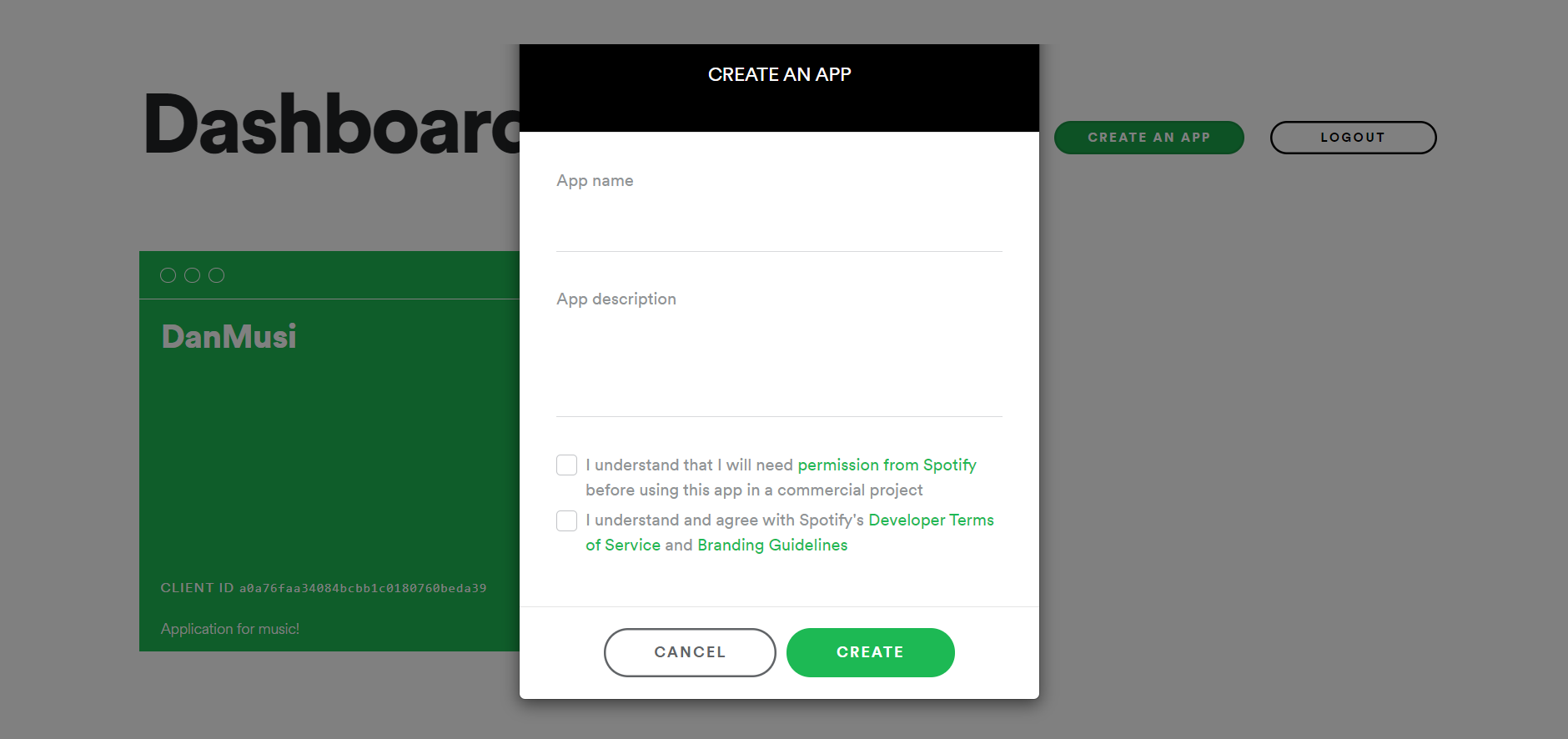
Cách đăng ký ứng dụng sử dụng Spotify API thông qua Spotify developer.

Truy cập vào liên kết : “<https://developer.spotify.com/dashboard/>” và nhấn nút “Login”

* **Nhấn nút “Create an app” để tạo mới một ứng dụng**



* **Nhập thông tin cần thiết của ứng dụng:**



* **Lưu lại thông tin Client ID và Client Secret:**

**Client ID**: là định danh duy nhất của ứng dụng.

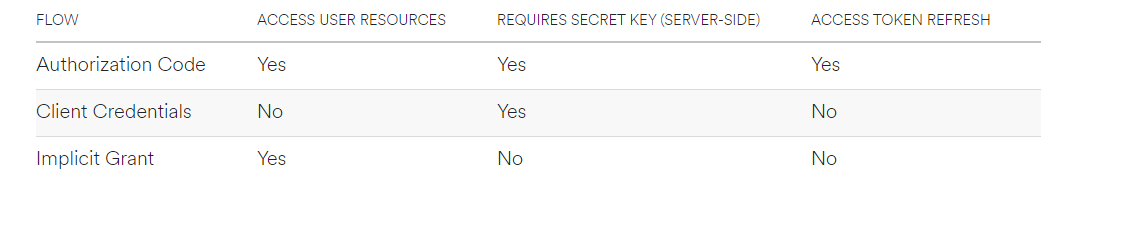
**Client Secret**: là chìa khóa mà bạn vượt qua trong các cuộc gọi an toàn cho tài khoản Spotify và dịch vụ Web API. Luôn luôn lưu trữ khóa bí mật của khách hàng một cách an toàn; không bao giờ tiết lộ công khai! Nếu bạn nghi ngờ rằng chìa khóa bí mật đã bị xâm nhập, tái tạo nó ngay lập tức bằng cách nhấn vào liên kết trên chỉnh sửa cài đặt “EDIT SETTINGS”



## Các loại xác thực API

Có 3 loại luồng xác thực trong khi làm việc với Spotify API:

* [**Authorization Code**](https://developer.spotify.com/documentation/general/guides/authorization-guide/#authorization-code-flow)
* [**Implicit Grant**](https://developer.spotify.com/documentation/general/guides/authorization-guide/#implicit-grant-flow)
* [**Client Credentials Flow**](https://developer.spotify.com/documentation/general/guides/authorization-guide/#client-credentials-flow)

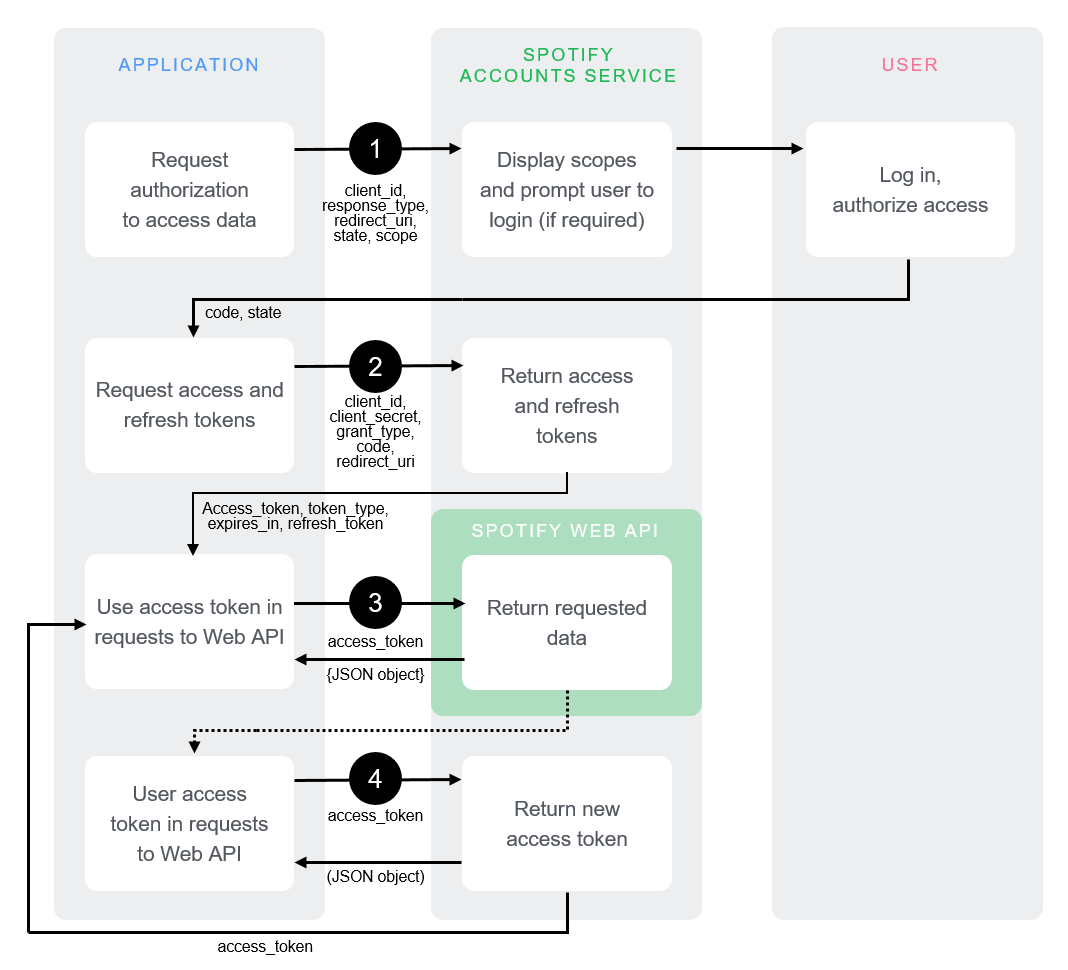


**Authorization Code Flow**

Thích hợp cho các ứng dụng trong đó người dùng cấp quyền một lần duy nhất, có thể kéo dài. Nó cung cấp một access\_token có thể được làm mới. Vì việc trao đổi mã token liên quan đến việc gửi client\_secret của bạn, thực hiện điều này trên một vị trí an toàn, giống như một dịch vụ phụ trợ, chứ không phải từ một khách hàng như một trình duyệt hoặc từ một ứng dụng di động.

**Điều bạn làm**: Nhắc người dùng của bạn đến một trang web nơi họ có thể chọn để cấp cho bạn truy cập vào dữ liệu của họ.

**Điều bạn nhận được:** Một access token và một refresh token.

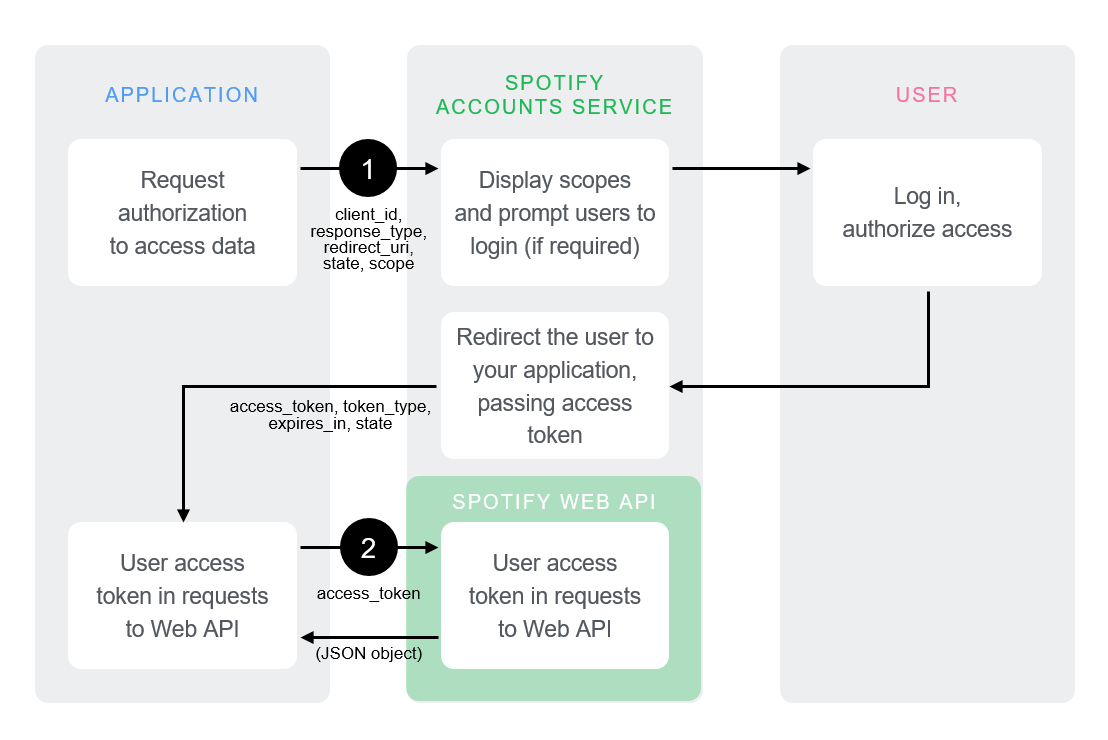


### **Implicit Grant Flow**

Implicit Grant Flow là dành cho client được thực hiện hoàn toàn sử dụng JavaScript và chạy trong trình duyệt của chủ sở hữu tài nguyên của bạn. Bạn không cần thực hiện một ứng dụng server-side để sử dụng nó. Giới hạn cho các yêu cầu được cải thiện nhưng không cung cấp refresh token.

**Điều bạn làm**: Bạn điều hướng người dùng đến dịch vụ tài khoản của Spotify.

**Điều bạn nhận được**: Access token.



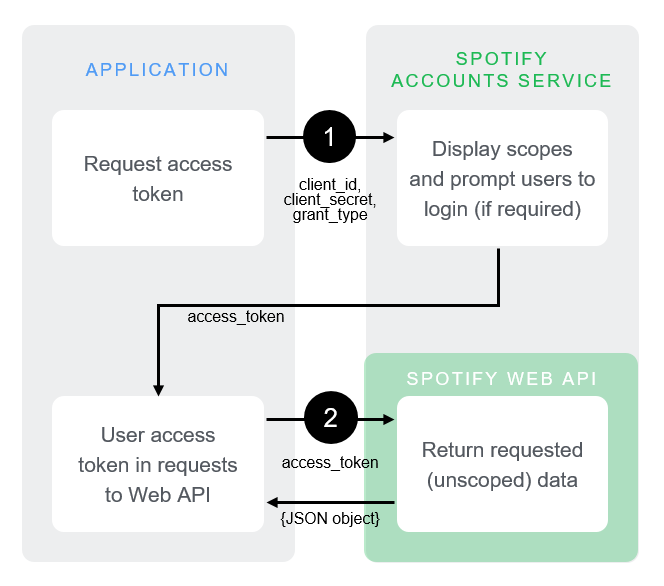
### **Client Credentials Flow**

Client Credentials Flow được sử dụng trong ứng dụng server-to-server. Chỉ điểm cuối(endpoints) không truy cập thông tin người dùng có thể được truy cập.

**Lưu ý**: Luồng này cho phép bạn có thể xác thực yêu cầu của bạn tới Spotify Web API và đạt được một giới hạn tốc độ cao hơn mức bạn có thể nhận được mà không cần xác thực. Tuy nhiên, luồng này không bao gồm ủy quyền và do đó không thể truy cập quản lý thông tin cá nhân của người dùng.

**Điều bạn làm**: Đăng nhập với Client ID và Secret Key.

**Điều bạn nhận được**: Access token.



## Luồng xác thực được sử dụng trong dự án và nguyên nhân được lựa chọn

Trong dự án này, phương thức xác thực được lựa chọn là **Authorization Code.** Bởi vì dự án này cần duy trì sự ủy quyền bởi người dùng trong một thời gian dài (Chỉ ủy quyền một lần).

* **Client Credentials** KHÔNG được lựa chọn bởi vì đây là luồng xác thực không được ủy quyền để có thể truy cập vào dữ liệu cá nhân của người dùng và không thể làm mới access\_token.
* **Implicit Grant** KHÔNG được lựa chọn bởi vì đây là luồng xác thực ngắn hạn, nó sẽ yêu cầu người dùng ủy quyền một lần nữa khi access\_token hết hạn.

## Phạm vi (Scopes) trong vấn đề xác thực người dùng thông qua Spotify API

Để sử dụng nền tảng Spotify, bạn cần làm quen với scopes. Scopes được cung cấp cho người dùng Spotify đang sử dụng ứng dụng bên thứ 3 sự chắc chắn rằng chỉ có thông tin mà họ chọn có thể chia sẻ hoặc được chia sẻ, đảm bảo rằng không có gì hơn.

Scopes bao gồm: Images, Spotify Connect, Playback, Users, Playlists, Library, Listening History và Follow.

# Chương 3: Tổng quan về Spotify Web Playback SDK

## Giới thiệu

Web Playback SDK là thư viện client-side javascipt cho phép người dùng tạo mới một “player” và phát bất kì bài hát nào từ Spotify trong trình duyệt thông qua Encrypted Media Extensions.

Sử dụng Web Playback SDK thông qua Spotify API, chúng ta có thể phát bài hát, phát album, tạm dừng, tăng âm lượng bài hát đang được phát,....

Web Playback SDK chỉ hoạt động trên một thiết bị trong cùng một thời điểm với cùng một tài khoản (Ví dụ: sử dụng tài khoản spotify nghe nhạc trên thiết bị điện thoại và laptop cùng một lúc).

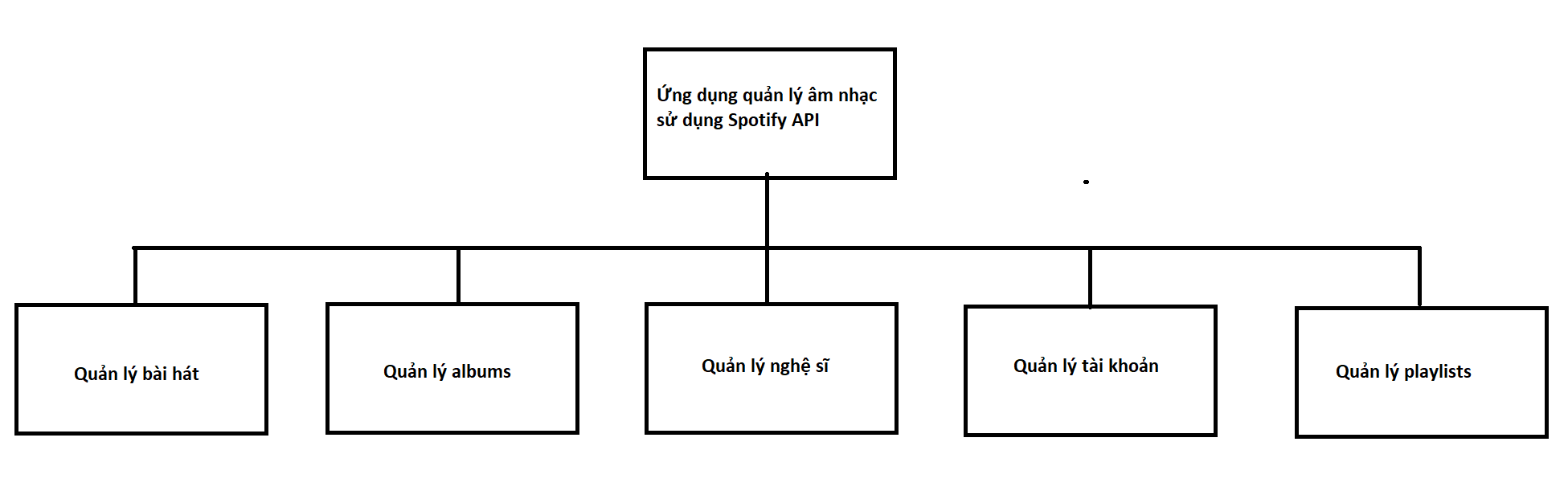
## Vấn đề lưu ý trong xác thực:

Vấn đề xác thực được tuân theo giống phần trên, nhưng có một số điểm cần lưu ý:

* Các phạm vi được yêu cầu: "streaming", "user-read-email", "user-read-private"
* Access Token sẽ hết hạn trong vòng 1 tiếng.
* Yêu cầu tài khoản xác thực phải là tài khoản premium
* Web Playback SDK chỉ làm việc trong client-side javascipt. NodeJS là không được hỗ trợ.

# Chương 4: Tổng quan về ứng dụng

## Cấu trúc



## Một số chức năng tiêu biểu

* Hiển thị danh sách những bài hát, albums mới nhất được cập nhật từ Spotify API.
* Hiển thị danh sách những nghệ sĩ mới, nghệ sĩ thường xuyên nghe, nghệ sĩ liên quan đến thể loại âm nhạc của bản thân.
* Hiển thị danh sách những bài hát, playlists đã lưu.
* Phát bài hát trực tiếp, chuyển tới hoặc trở về bài hát trước đi kèm chức năng trộn bài hát và lặp lại.
* Quản lý tài khoản cá nhân.
* Tìm kiếm tên nghệ sĩ, bài hát và albums theo thời gian thực.

## Quy trình phát triển phần mềm:

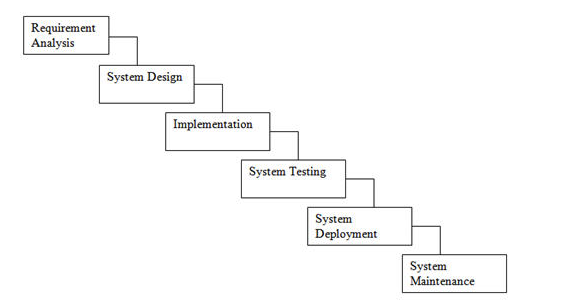
Mô hình phát triển phần mềm được áp dụng trong đồ án này là “Mô hình thác nước – Water fall”

## Giới thiệu

Mô hình thác nước là một ví dụ của mô hình Sequential (Tuần tự). Trong mô hình này, hoạt động phát triển phần mềm được chia thành các giai đoạn khác nhau và từng giai đoạn bao gồm hàng loạt các nhiệm vụ và có các mục tiêu khác nhau.

Mô hình Thác nước là giai đoạn đầu trong quá trình SDLC. Trên thực tế, nó là mô hình đầu tiên được sử dụng rộng rãi trong ngành công nghiệp phần mềm. Nó được chia thành các pha, đầu ra của một pha trở thành đầu vào của pha tiếp theo. Nó là giai đoạn bắt buộc được hoàn thành trước khi bắt đầu giai đoạn tiếp theo. Nói tóm lại, không có sự chồng chéo nào trong mô hình thác nước.

Trong thác nước, sự phát triển của một pha chỉ bắt đầu khi giai đoạn trước hoàn thành. Do tính chất này, mỗi giai đoạn của mô hình thác nước phải được xác định khá chính xác. Các giai đoạn chuyển từ mức cao xuống mức thấp hơn, giống như một thác nước nên mô hình này được đặt tên là mô hình thác nước.



## Chương 5: Tổng quan về Spring boot

## Lịch sử hình thành

Spring là một application framework mã nguồn mở, được giới thiệu vào năm 2002. Rod Johnson đã đưa ra ý tưởng này từ kinh nghiệm làm việc với kiến trúc J2EE. Ông đã viết cuốn sách với tiêu đề: “J2EE Development without EJB” để giới thiệu khái niệm trình chứa hạng nhẹ (lightweight container) với lý luận: EJB có giá trị của nó nhưng không phải lúc nào cũng cần thiết và phù hợp cho tất cả ứng dụng. Theo Rod Johnson thì EJB không phức tạp nhưng nó cố gắng cung cấp giải quyết cho các vấn đề phức tạp. EJB được phát triển chính cho các giao dịch, đối tượng được phân phát ở trên các máy khác. Nhưng một số Enterprise Project không có mức độ phức tạp và vẫn sử dụng EJB và thậm chí các ứng dụng đơn giản lại trở nên phức tạp. Trong trường hợp này thì Spring là một lựa chọn.

Các framework phổ biến khác như Struts, JSF, Tapestry …là các framwork tốt cho tầng Web nhưng khi chúng ta sử dụng framework này, chúng ta phải cung cấp thêm framework khác để giải quyết tầng enterprise mà tích hợp tốt với các framework này. Spring làm giảm bớt vấn đề này bằng cách cung cấp một framework toàn diện bao gồm:

- Core bean container

- MVC framework

- AOP integration framework

- JDBC integration framework

- EJB integration framework

- Các mặt mạnh trong kiến trúc Spring bao gồm:

- Spring tổ chức các tầng trung gian một cách hiệu quả, mà không quan tâm liệu người dùng có sử dụng EJB hay không. Spring sẽ thăm dò xem có nên để người dùng tự quyết định nếu người dùng chỉ dùng Struts hoặc các framework khác với các API của J2EE. Và thành phần được coi là quan trọng nhất trong tầng trung gian, các dịch vụ quản lý cấu hình của Spring có thể đưa vào một tầng kiến trúc nào đó của một môi trường thực thi bất kỳ.

- Spring sẽ giúp loại bỏ tình trạng lan tràn Singleton trong nhiều dự án. Theo một số nhà nghiên cứu, đây là một vấn đề lớn, làm suy giảm khả năng kiểm thử và tính hướng đối tượng.

- Spring sẽ khiến cộng đồng dần quên đi nhu cầu sử dụng các định dạng file thuộc tính thiết kế thông qua việc quản lý cấu hình nhất quán trong các ứng dụng và các dự án. Với IoC (Inversion of Control) và DI (Dependency Injection), người dùng chỉ cần quan sát các thuộc tính JavaBean của lớp cho các tham số khởi tạo mà không cần phải đọc Javadoc hay mã nguồn.

- Spring giúp công việc lập trình trở nên hiệu quả hơn bằng cách giảm chi phí lập trình giao diện, mà không phải là các lớp.

- Spring được thiết kế để các ứng dụng phụ thuộc ở mức tối thiểu vào các hàm API. Phần lớn các đối tượng giao dịch trong các ứng dụng Spring đều không có sự phụ thuộc nào nào Spring.

- Các ứng dụng xây dựng trên nền Spring rất dễ kiểm thử đơn vị.

- Spring biến việc sử dụng EJB là một tùy chọn thay vì là thành phần bắt buộc trong kiến trúc ứng dụng. Người dùng có thể chọn cài đặt các chuẩn thương mại như POJO hoặc EJB cục bộ mà không ảnh hưởng tới mã gọi.

- Spring còn trợ giúp giải quyết nhiều vấn đề mà không cần EJB. Spring cung cấp một công cụ thay thế EJB phù hợp với các ứng dụng. Ví dụ, Spring sử dụng AOP để phân phối các quản lý khai báo mà không dùng container EJB, thậm chí không cần JTA, nếu chỉ cần làm việc với một cơ sở dữ liệu.

- Spring cung cấp một framework nhất quán để truy cập cơ sở dữ liệu, dù người dùng sử dụng JDBC hay một số sản phẩm ánh xạ O/R như TopLink, Hibernate hoặc một cài đặt JDO. Spring sử dụng một mô hình lập trình đơn giản, nhất quán trong nhiều lĩnh vực, biến công cụ này trở thành một “loại hồ gắn kết” kiến trúc lý tưởng. Có thể thấy rõ điều này khi so sánh với với JDBC, JMS, JavaMail, JNDI và nhiều hàm API quan trọng khác.

## Ưu điểm của Spring

✓ **Quản lý phiên.**

Spring giúp kiểm soát hiệu quả, dễ dàng, và an toàn các đơn vị công việc như các  
phiên làm việc Hibernate hay TopLink. Mã liên quan chỉ sử dụng riêng công cụ ORM  
thường cần phải sử dụng kèm một đối tượng “Session” để kiểm soát tính hiệu quả và giao  
dịch thích hợp. Spring có thể tạo mới và liên kết một phiên tới luồng hiện tại một cách  
trong suốt, bằng cách sử dụng hoặc hướng tiếp cận kiểu lá chắn AOP, hoặc sử dụng một  
lớp bao “template” tường minh tại mức mã Java. Do vậy, Spring giải quyết được nhiều  
vấn đề liên quan tới sử dụng ảnh hưởng tới nhiều người dùng của công nghệ ORM.  
✓ **Quản lý tài nguyên.**

Các ngữ cảnh ứng dụng (application context) Spring có thể kiểm soát vị trí và cấu hình của các Hibernate SessionFactories, các nguồn dữ liệu JDBC cũng như các nguồn tài nguyên liên quan khác. Do vậy, Spring Framework cho phép các giá trị này được dễ dàngquản lý và thay đổi.

✓ **Quản lý giao dịch tích hợp.**

Spring cho phép bạn bao mã ORM bằng một lá chắn phương thức AOP, hoặc một lớp bao “template” tường minh tại mức mã Java. Trong bất kỳ trường hợp nào, ngữ nghĩa giao dịch đều được kiểm soát giúp bạn, và việc kiểm soát giao dịch thích hợp trong trường hợp xảy ra các ngoại lệ cũng được quan tâm. Như chúng ta sẽ đề cập sau đây, bạn cũng nhận được lợi ích của khả năng sử dụng và tráo đổi các bộ quản lý giao dịch khác nhau, mà không bị ảnh hưởng tới mã ORM liên quan. Ngoài ra, một lợi ích khác mà Spring đem lại là mã JDBC liên quan có thể hoàn toàn tích hợp với mã ORM, trong hầu hết các công cụ hỗ trợ ORM. Điều này rất hữu ích cho việc kiểm soát tính năng không tuân theo ORM.

✓ **Bao đóng ngoại lệ.**

Như được đề cập ở trên, Spring có thể bao đóng các ngoại lệ từ tầng ORM, chuyển đổi chúng từ các ngoại lệ độc quyền của các hãng thành một tập các ngoại lệ trừu tượng thời gian chạy. Điều này cho phép bạn kiểm soát hầu hết các ngoại lệ bền bỉ nhất, đó là những ngoại lệ không có khả năng khôi phục, chỉ ở trong các tầng thích hợp, mà không làm ảnh hưởng tới các cụm catch/throw và các khai báo ngoại lệ được sử dụng thường xuyên. Bạn có thể vẫn bẫy và kiểm soát các ngoại lệ ở bất kỳ nơi nào cần thiết. Lưu ý rằng các ngoại lệ JDBC (bao gồm cả các đặc trưng cụ thể của DB) cũng được chuyển đổi tới cùng hệ thống cấp bậc, nghĩa là bạn có thể thực hiện một số thao tác với JDBC bên trong một mô hình lập trình nhất quán.

✓ **Tránh sự bó buộc của các nhà cung cấp.**

Các giải pháp ORM có các đặc tính hiệu năng khác, và không có giải pháp nào thỏa mãn được tất cả. Thay vào đó, bạn có thể sẽ nhận thấy rằng tính năng cụ thể lại không phù hợp với một cài đặt sử dụng công cụ ORM của bạn. Vì vậy, một giải pháp khôn ngoan là tách kiến trúc của bạn khỏi các cài đặt phụ thuộc công cụ. Nếu bạn cần chuyển tới một cài đặt khác vì các lý do liên quan tới chức năng, hiệu năng, hay các mối quan tâm khác, thì việc sử dụng Spring hiện tại có thể khiến việc chuyển đổi được dễ dàng hơn rất nhiều. Sự trừu tượng của Spring trong Transactions và Exceptions của công cụ ORM, cùng với hướng tiếp cận IoC cho phép bạn dễ dàng chuyển đổi trong các đối tượng ánh xạ/DAO cài đặt chức năng truy cập dữ liệu khiến nó dễ cô lập tất cả các mã phụ thuộc vào ORM cụ thể trong một lĩnh vực của ứng dụng mà không làm ảnh hưởng tới sức mạnh của công cụ ORM. Ứng dụng mẫu PetClinic đi kèm với Spring demo lợi ích của tính khả chuyển mà Spring đem lại thông qua việc cung cấp các biến sử dụng JDBC, Hibernate, TopLink và Apache OJB để cài đặt tầng tồn tại lâu dài (pesistence layer).

✓ **Dễ dàng kiểm thử.** Sự chuyển đổi hướng tiếp cận điều khiển của Spring khiến nó dễ dàng chuyển đổi cài đặt và vị trí của nguồn tài nguyên như các session factory của Hibernate, các nguồn dữ liệu, trình quản lý giao diện, và các cài đặt đối tượng ánh xạ (nếu cần thiết). Việc này giúp dễ dàng hơn rất nhiều trong việc cô lập và kiểm thử từng đoạn mã liên quan một cách tách biệt.

## Chương 6: Tổng quan front end frameworks & libraries

## Bootstrap

**1. Bootstrap là gì?**

Bootstrap là một framework HTML, CSS, và JavaScript cho phép thiết kế phát triển responsive web mobile. Bootstrap cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn.

Bootstrap bao gồm các HTML templates, CSS templates và JavaScript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong Bootstrap có thêm các plugin Javascript trong nó giúp cho việc thiết kế responsive dễ dàng hơn và nhanh chóng hơn.

**2. Lịch sử Bootstrap**

Bootstrap được phát triển bởi Mark Otto và Jacob Thornton tại Twitter. Nó được xuất bản dưới dạng mã nguồn mở vào tháng 8 năm 2011 trên GitHub. Tính đến thời điểm này nó đã phát triển được 8 năm. Bản Bootstrap mới nhất bây giờ là Bootstrap 4 với nhiều cải tiến đáng giá so với những phiên bản trước.

**3. Tại sao nên sử dụng Bootstrap?**

Dễ sử dụng: Nó đơn giản vì được dựa trên HTML, CSS và JavaScript nên chỉ cẩn có kiến thức cơ bản về 3 ngôn ngữ front-end này là có thể sử dụng tốt Bootstrap.

Tính năng Responsive: Bootstrap xây dựng sẵn *grid system* giúp việc thiết kế website chạy được trên cả các thiết bị di động, tablet, và desktop dễ dàng.

Mobile: Trong Bootstrap, thiết kế website theo hướng mobile-first (ưu tiên cho thiết bị di dộng) là một trong các phần cốt lõi của framework.

Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera) nhưng lưu ý là nó không tương thích với Internet Explore phiên bản quá cũ (<= 9).

## jQuery

* + 1. **Giới thiệu jQuery**

jQuery là một thư viện JavaScript đa tính năng, nhỏ gọn, nhanh, được tạo bởi John Resig vào năm 2006 với một phương châm hết sức ý nghĩa: Write less, do more - Viết ít hơn, làm nhiều hơn.

jQuery đơn giản hóa việc duyệt tài liệu HTML, xử lý sự kiện, hoạt ảnh và tương tác Ajax để phát triển web nhanh chóng. Các phân tích web đã chỉ ra rằng, jQuery là thư viện JavaScript được triển khai rộng rãi nhất.

jQuery là một bộ công cụ JavaScript được thiết kế để đơn giản hóa các tác vụ khác nhau bằng cách viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery:

**Thao tác DOM** − jQuery giúp dễ dàng lựa chọn các phần tử DOM để traverse (duyệt) một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle.

**Xử lý sự kiện** − jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler.

**Hỗ trợ AJAX** − jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.

**Hiệu ứng động** − jQuery đi kèm với rất nhiều hiệu ứng động đẹp mà bạn có thể sử dụng cho các website của mình.

**Gọn nhẹ** − jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped).

**Được hỗ trợ hầu hết bởi các trình duyệt hiện đại** − jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+.

**Cập nhật và hỗ trợ các công nghệ mới nhất** − jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

* + 1. **Tại sao jQuery là cần thiết?**

Vì sao chúng ta cần sử dụng jQuery? Có gì khác biệt khi sử dụng jQuery thay vì AJAX/JavaScript? Câu trả lời đó là chính là:

- jQuery rất nhanh và có thể mở rộng.

- jQuery tạo điều kiện cho người dùng viết UI liên quan đến function codes trong số lines ít nhất có thể.

- jQuery cải thiện hiệu suất của một ứng dụng.

- Các ứng dụng web tương thích với trình duyệt có thể được phát triển

- jQuery sử dụng hầu hết các tính năng mới của các trình duyệt mới.

Trong số rất nhiều JavaScript frameworks, jQuery là phổ biến nhất và mở rộng nhất. Nhiều công ty lớn sử dụng jQuery có thể kể tới là: Microsoft, Google, IBM, Netflix…

## Chương 7: Tổng quan về các công nghệ và công cụ liên quan

**- Spring Boot**:

**Spring Boot** là một dự án phát triển bởi **JAV**(ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng

**- Spring Security**:

Spring Security là một framework tập trung vào việc cung cấp khả năng xác thực và phân quyền cho ứng dụng Java.

**- Spring Data JPA**:

JPA (Java Persistence API) là 1 giao diện lập trình ứng dụng Java, nó mô tả cách quản lý các mối quan hệ dữ liệu trong ứng dụng sử dụng Java Platform.

JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table/các mối quan hệ giữa các table trong database sang các class/mối quan hệ giữa các object.**- Hibernate**:

[Hibernate](https://stackjava.com/framework/hibenrate-framework/series-hibernate-phan-1-gioi-thieu-ve-hibernate-framework.html) là 1 ORM (Object Relational Mapping) framework cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không cần quan tâm đến loại database sử dụng, SQL…

**- HikariCP:**

[**HikariCP**](https://brettwooldridge.github.io/HikariCP/) giúp tạo ***database connection pool*** trong Java. JDBC connection pool có hiệu năng cao, rất nhẹ, được phát triển bởi [Brett Wooldridge](https://github.com/brettwooldridge) (năm 2012) và vẫn đang được cập nhật liên tục.

**- Thymeleaf**:

**Thymeleaf**là một **Java XML/XHTML/HTML5 Template Engine** nó có thể làm việc với cả hai môi trường Web và môi trường không phải Web. Nó phù hợp hơn khi được sử dụng để phục vụ **XHTML/HTML5** trên tầng **View**(View Layer) của ứng dụng **Web**dựa trên kiến trúc **MVC**. Nó có thể xử lý bất kỳ một file **XML**nào, thậm trí trên các môi trường offline (Không trực tuyến). Nó hỗ trợ đầy đủ để tương tác với **Spring Framework.**

**- Boostrap**:

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn.

**- Postman**:

Công cụ giúp kiểm thử các API.

**- Eclipse**:

Công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi IBM.

**- Visual Studio Code**:

Công cụ Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn.

**- Google Chrome:**

Google Chrome là trình duyệt web phổ biến nhất hiện nay, hỗ trợ trong việc thiết kế giao diện website, debug javascript, …

## Chương 8: Hình ảnh thực tế từ ứng dụng

## Chương 9: Tham khảo

**Spotify API**: <https://developer.spotify.com/documentation/web-api/quick-start/>

**Spotify Web Playback SDK**: <https://developer.spotify.com/documentation/web-playback-sdk/quick-start/>

**Spring boot**: <https://spring.io/guides/gs/spring-boot/>

**Bootstrap**: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

**Jquery**: <https://api.jquery.com/>