**TÀI LIỆU ĐẶC TẢ YÊU CẦU PHẦN MỀM**

**(Software Requirements Specification - SRS)**

**Xây dựng website nghe nhạc trực tuyến**

**Môn học: Công nghệ Phần mềm**

**Lớp học phần: INT2208E 23**

**Giảng viên: PGS.TS Đặng Đức Hạnh**

**Nhóm sinh viên thực hiện:**

| **Cao Đức Chung** | **22028015** |
| --- | --- |
| **Đàm Quang Đạt** | **22028026** |
| **Nhâm Ngọc Đức** | **22028096** |
| **Nguyễn Mạnh Quân** | **22028171** |
| **Lê Văn Thắng** | **22028313** |

**Hà Nội, ngày 27 tháng 3 năm 2024**

# **Mục lục**

[Mục lục 2](#_5ou6pybvdkfc)

[Bảng ghi nhận thay đổi 2](#_bzdt1rvkbht)

[1. Giới thiệu 3](#_vc17j1ghx0ns)

[1.1. Mục đích 3](#_oe54h1yikkjd)

[1.2. Phạm vi 3](#_fxdrrsz96d0y)

[1.3. Tài liệu tham khảo 3](#_4yxrv6r25o2g)

[1.4. Bố cục tài liệu 3](#_iz8as5yv0ddj)

[1.5. Các thuật ngữ, từ viết tắt 4](#_i48cgcf3zw2a)

[2. Các yêu cầu chung 5](#_34ctu3eaxic3)

[2.1. Yêu cầu về môi trường 5](#_gncj6ci4k7u5)

[2.2. Yêu cầu về giao diện 5](#_6fp2xzlxhhkp)

[2.3. Yêu cầu về người dùng 5](#_ofb8jiwfjur7)

[3. Yêu cầu chức năng 6](#_gpv68073soyr)

[3.1. Các tác nhân 6](#_l8baqb8vl1do)

[3.2. Biểu đồ use-case tổng quan 7](#_oxup25r2tmyu)

[3.3. Biểu đồ phân rã use-case của guest user 8](#_neu78m1114n1)

[3.4. Biểu đồ phân rã use-case của tài khoản thường 9](#_qialewrei604)

[3.5. Biểu đồ phân rã use-case của tài khoản nghệ sĩ 10](#_py36ie7u1n3n)

[3.6. Biểu đồ phân rã use-case của tài khoản Admin 11](#_4c41cfh1gbx9)

[3.7. UC Stories 12](#_ojr103okleph)

[4. Yêu cầu phi chức năng 14](#_z6d3kvu8o3ig)

[4.1. Giao diện người dùng 14](#_mlg55aynxzzi)

[4.2. Bảo mật 14](#_thrm90e3vucc)

[4.3. Ràng buộc 15](#_rpmrrz47q71u)

# **Bảng ghi nhận thay đổi**

| Tên | Thời gian | Mục đích thay đổi | Phiên bản |
| --- | --- | --- | --- |
| Cao Chung | 22/04/2024 | Bổ sung UC stories, chỉnh sửa biểu đồ phân rã | 1.1 |
| Lê Thắng | 25/04/2024 | Chỉnh sửa font, căn chỉnh lại lề 2 bên | 1.2 |

# **Giới thiệu**

## *1.1. Mục đích*

Tài liệu này được tạo nên với mục đích cung cấp cái nhìn tổng quan về các yêu cầu và thành phần của dự án.

## *1.2. Phạm vi*

Tài liệu đặc tả yêu cầu phần mềm này được soạn nhằm phục vụ dự án xây dựng website nghe nhạc trực tuyến.

Cung cấp một nền tảng trực tuyến cho phép người dùng tìm kiếm, phát nhạc, tạo danh sách phát cá nhân cũng như chia sẻ nhạc với bạn bè. Đồng thời người dùng có thể phát hành, chia sẻ các sản phẩm âm nhạc của họ với cộng đồng.

## *1.3. Tài liệu tham khảo*

Các slides bài giảng trên courses.uet.vnu.edu.vn

Mẫu tài liệu đặc tả yêu cầu phần mềm chuẩn IEEE.

## *1.4. Bố cục tài liệu*

Tài liệu gồm 4 phần chính:

* Phần 1 - Giới thiệu: Giới thiệu tổng quan về tài liệu và dự án.
* Phần 2 - Yêu cầu chung của hệ thống: Liệt kê danh sách các yêu cầu cùng những mô tả cơ bản.
* Phần 3 - Các yêu cầu chức năng: Mô tả chung về hệ thống, các tác nhân trong hệ thống và các biểu đồ thể hiện chức năng.
* Phần 4 - Các yêu cầu phi chức năng: Báo cáo chi tiết về framework nhóm sử dụng.

## *1.5. Các thuật ngữ, từ viết tắt*

| Admin | Administrator: Người quản trị. Thuật ngữ này dùng để chỉ một vai đối với hệ thống |
| --- | --- |
| HTML | Hypertext Markup Language |
| Use-case | Mô tả tương tác của người dùng và hệ thống |
| Guest user | Tài khoản khách, khi người dùng hệ thống chưa đăng nhập |
|  |  |
|  |  |

# **Các yêu cầu chung**

## *2.1. Yêu cầu về môi trường*

| Mã nghiệp vụ | Mô tả yêu cầu |
| --- | --- |
| F001.01 | Cổng thông tin hoạt động trên các hệ điều hành thông dụng |
| F001.02 | Hoạt động tốt trên các trình duyệt Firefox, Google Chrome, Microsoft Edge, Opera,... |

## *2.2. Yêu cầu về giao diện*

| Mã nghiệp vụ | Mô tả yêu cầu |
| --- | --- |
| F002.01 | Trực quan, dễ hiểu, dễ sử dụng |
| F002.02 | Tốc độ phản hồi thấp < 0.5s |
| F005.03 | Các thao tác cơ bản để xuất bản thông tin được thực hiện nhanh chóng |

## *2.3. Yêu cầu về người dùng*

| Mã nghiệp vụ | Mô tả yêu cầu |
| --- | --- |
| F003.01 | Người dùng có kiến thức về các thao tác máy tính cơ bản |

# **Yêu cầu chức năng**

## *3.1. Các tác nhân*

Hệ thống gồm có các tác nhân:

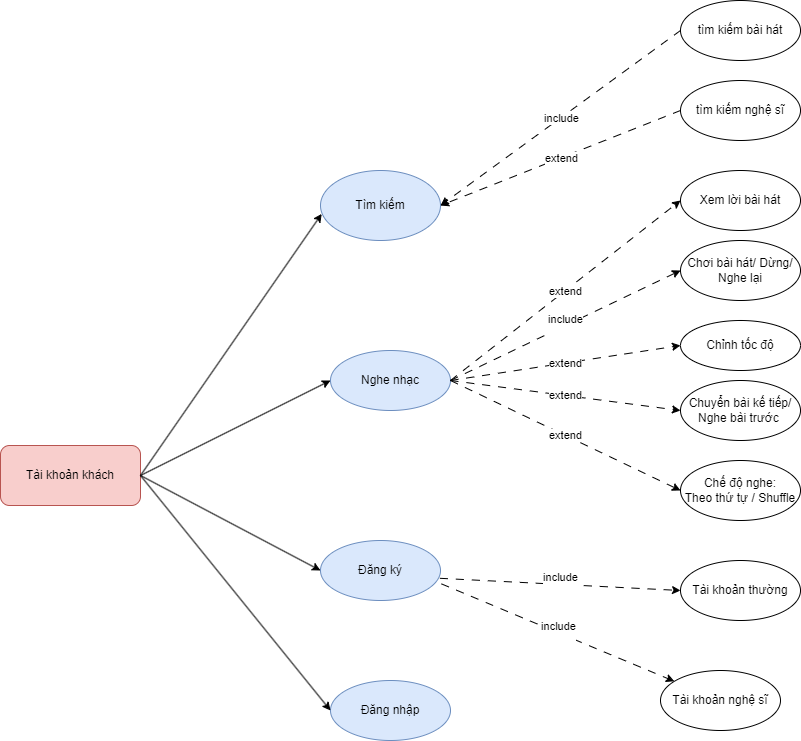
* **Người dùng khách (Guest user)**: có thể tìm kiếm, nghe nhạc nhưng không thể tạo playlist hay tương tác với các tài khoản khác trên hệ thống, cũng như không thể phát hành nhạc.
* **Tài khoản thông thường**: có thể tìm kiếm, nghe nhạc, tạo playlist và tương tác với các tài khoản khác.
* **Tài khoản nghệ sĩ**: có thể thao tác như người dùng thông thường và phát hành sản phẩm âm nhạc trên hệ thống.
* **Admin**: Có toàn bộ chức năng thông thường, thêm vào đó có khả năng tiếp nhận các báo cáo từ người dùng, gỡ bài hát và xử lý tài khoản vi phạm.

## 

## *3.2. Biểu đồ use-case tổng quan*

## 

## *3.3. Biểu đồ phân rã use-case của guest user*



## *3.4. Biểu đồ phân rã use-case của tài khoản thường*

## *3.5. Biểu đồ phân rã use-case của tài khoản nghệ sĩ*

## 

## 

## 

## *3.6. Biểu đồ phân rã use-case của tài khoản Admin*

## 

## 

## *3.7. UC Stories*

* Tài khoản khách:
* SCOTT là một nền tảng nghe nhạc trên mạng internet dành cho người dùng. Đối với đối tượng người dùng khách khi mới trải nghiệm, họ có thể tìm kiếm các bài hát dựa trên tên bài hát hoặc tên các nghệ sĩ đăng tải bài hát lên nền tảng.
* Khi nghe nhạc nhạc, người dùng có thể sử dụng các công cụ như xem lời bài hát, chỉnh tốc độ, nghe lại, nghe theo thứ tự hoặc chuyển các bài hát luân phiên. Đồng thời, tài khoản khách có quyền nâng cấp lên tài khoản thường hoặc tài khoản nghệ sĩ.
* Với người dùng thường, họ sẽ có thêm những chức năng khác như tạo playlist riêng. Đây là một sở thích của đa số người sử dụng nền tảng để có thể nghe một lượt các bài hát yêu thích mà không cần tìm kiếm liên tục. Tài khoản thường có thể ấn vào nút “thêm vào playlist” khi đang nghe bài nhạc đó. Nền tảng cũng sẽ cho phép sự xuất hiện của nhiều playlist để người đăng ký tài khoản có thể tạo ra nhiều nhóm bài hát yêu thích khác nhau. Tài khoản thường sẽ được phép chỉnh sửa hồ sơ cá nhân như họ tên, ngày tháng năm sinh, các nghệ sĩ yêu thích,... Bên cạnh đó, họ có thể tương tác với tài khoản các người dùng đã đăng ký khác hoặc nâng cấp lên tài khoản nghệ sĩ để có các chức năng đặc biệt hơn. Tài khoản thường cũng có quyền thay đổi mật khẩu và đăng xuất để trở về giao diện người dùng khách.
* Tài khoản nghệ sĩ có chức năng đặc biệt hơn tài khoản thường là việc được phát hành nhạc. Dự án này sẽ tạo thêm nhiều cơ hội để các nghệ sĩ bán chuyên có thể đem lại những bài hát chất lượng trên nền tảng. Bên cạnh đó, nghệ sĩ có thể ấn vào mục “xem số liệu thống kê” ở mỗi bài hát mình đăng tải để biết thêm về những thông tin như tổng số lượt nghe nhạc trong ngày hoặc trong tháng.
* Admin là tài khoản có chức năng tối cao nhất và chỉ được sử dụng bởi đội ngũ phát triển dự án. Admin sẽ có quyền nhận phản hồi, báo cáo từ người dùng đã đăng ký thông qua mục chat với Admin. Bên cạnh đó, Admin được phép tiến hành gỡ các bài hát có nội dung không chuẩn mục hoặc đã bị các người dùng báo cáo. Cuối cùng, Admin có khả năng xử lý các tài khoản vi phạm trên nền tảng.

## 

# **Yêu cầu phi chức năng**

## *4.1. Giao diện người dùng*

* **Xử lý biểu mẫu**: Django hỗ trợ xử lý biểu mẫu web một cách dễ dàng và hiệu quả, bao gồm xác thực dữ liệu và bảo mật.
* **Giao diện quản trị**: Django cung cấp một giao diện quản trị sẵn có, giúp Admin quản lý dữ liệu một cách tiện lợi.
* **Hỗ trợ front-end**: Django cũng linh hoạt trong việc tích hợp với các front-end frameworks như React để tạo ra hiệu suất hoạt động cao cho ứng dụng.
* **Tạo API**: Django REST Framework cho phép xây dựng các API chất lượng để cung cấp dữ liệu cho các ứng dụng front-end.
* **Tối ưu hóa hiệu suất**: Django giúp tối ưu hiệu suất của bộ nhớ cache, cơ sở dữ liệu và giao tiếp không đồng bộ.

## *4.2. Bảo mật*

* **Xác thực và ủy quyền**: Việc quản lý thông tin người dùng, đăng nhập và đăng ký thông qua token là tính năng nổi bật của Django.
* **Bảo mật**: Django cung cấp tính năng mã hóa mật khẩu và kiểm tra dữ liệu đầu vào. Điều này giúp phòng tránh một số cuộc tấn công như SQL injection và cross-site scripting (XSS).
* **Bảo mật giao diện người dùng**: Django hỗ trợ CSRF (Cross-Site Request Forgery) và clickjacking protection, giúp ngăn chặn các cuộc tấn công từ phía giao diện người dùng.
* **Quản lý session và cookie**: Django cung cấp các công cụ để quản lý session và cookie để bảo vệ người dùng các cuộc tấn công như session fixation và session hijacking.
* **Bảo mật cấu hình**: Django cho phép cấu hình các thiết lập bảo mật như HTTPS, CORS (Cross-Origin Resource Sharing), và CSP (Content Security Policy) giúp giảm thiểu rủi ro từ tấn công mạng.

## *4.3. Ràng buộc*

* **Ràng buộc cấp quyền truy cập**: Django cung cấp hệ thống quản lý người dùng và phân quyền linh hoạt, cho phép Admin xác định chính xác người dùng nào nào có quyền truy cập vào các phần riêng biệt của ứng dụng.
* **Ràng buộc dữ liệu**: Django cho phép định nghĩa ràng buộc dữ liệu như giá trị tối đa/tối thiểu và các quy tắc kiểm tra dữ liệu trước khi lưu vào cơ sở dữ liệu. Điều này giúp ngăn chặn việc lưu trữ dữ liệu không hợp lệ hoặc không mong muốn.
* **Ràng buộc duy nhất** : Django cho phép định nghĩa các trường dữ liệu duy nhất, nghĩa là giá trị của trường đó phải là duy nhất trong toàn bộ bảng. Điều này giúp tránh tình trạng lưu trữ các dữ liệu trùng lặp.