TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**DỰ ÁN CÔNG NGHỆ THÔNG TIN 1**

**XÂY DỰNG ỨNG DỤNG NGHE NHẠC VỚI ELECTRON**

*Người hướng dẫn*: **NGUYỄN TRẦN UYÊN THÚY**

*Người thực hiện*: **CAO THẾ KIỆT – 519H0184**

Lớp **: 19H50204**

Khoá  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**DỰ ÁN CÔNG NGHỆ THÔNG TIN 1**

**XÂY DỰNG ỨNG DỤNG NGHE NHẠC VỚI ELECTRON**

Người hướng dẫn: **NGUYỄN TRẦN UYÊN THÚY**

Người thực hiện: **CAO THẾ KIỆT**

Lớp **: 19H50204**

Khoá  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

Đầu tiên, em xin được cảm ơn tập thể các thầy và các cô ở trường Đại học Tôn Đức Thắng đã giảng dạy em trong suốt quãng thời gian 4 năm vừa qua để em có đủ khả năng để đi thực tập ở công ty TNHH-DV tin học Long Bình.

Sau đó, em xin gửi lời cảm ơn đến cô Nguyễn Trần Uyên Thúy – mentor của em tại công ty Long Bình. Cô Thúy tuy có phần nghiêm túc trong công việc, nhưng nhờ có cô ấy đã chỉ dạy cho em nhiều điều từ kinh nghiệm sống cho đến kiến thức lập trình mà em có thể sẽ mất nhiều năm mới có thể tự nhận ra được.

Cuối cùng, em muốn nói lời cảm ơn đến các bạn bè, các đồng nghiệp, những bạn thực tập sinh ở tại Long Bình cùng với em trong suốt khoảng thời gian vừa qua. Họ đã chỉ em về nhiều điều mà em vẫn còn chưa rõ về học phần Dự án Công nghệ thông tin và Kiến tập công nghiệp này.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của cô Nguyễn Trần Uyên Thúy;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 12 tháng 8 năm 2024*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Cao Thế Kiệt*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Nội dung bài báo cáo này bao gồm phần giới thiệu về đề tài, giới thiệu về các công nghệ sử dụng để thực hiện đề tài, các màn hình đã thực hiện và phần kết luận.

# MỤC LỤC

# DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

# DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

# CHƯƠNG 1: TỔNG QUAN

## Giới thiệu

Ứng dụng nghe nhạc [Tên ứng dụng] là nền tảng giải trí hoàn hảo dành cho những người yêu âm nhạc, mang đến trải nghiệm nghe nhạc tuyệt vời mọi lúc, mọi nơi. Với giao diện thân thiện và dễ sử dụng, ứng dụng cho phép người dùng khám phá hàng triệu bài hát thuộc nhiều thể loại khác nhau, từ những bản nhạc hot nhất hiện nay đến những ca khúc kinh điển vượt thời gian.

[Tên ứng dụng] cung cấp tính năng tải nhạc về thiết bị để nghe ngoại tuyến, giúp bạn tận hưởng âm nhạc ngay cả khi không có kết nối internet. Vì tính chất thời gian có hạn nên ứng dụng sẽ thiên về xây dựng một giao diện thân thiệt với người dùng và có thể tạo playlist cá nhân dựa trên sở thích riêng của người dùng.

Với chất lượng âm thanh đỉnh cao, kho nhạc phong phú, và các tính năng thông minh, [Tên ứng dụng] là người bạn đồng hành lý tưởng trên hành trình khám phá và thưởng thức âm nhạc.

## 1.2 Mục tiêu đề tài

Mục tiêu của đề tài này là nghiên cứu, phân tích và phát triển một ứng dụng nghe nhạc với các tính năng hiện đại, đáp ứng nhu cầu của người dùng trong việc tiếp cận và thưởng thức âm nhạc. Cụ thể, đề tài hướng tới các mục tiêu chính sau:

1. Tìm hiểu và áp dụng các công nghệ mới đang được sử dụng để phát triển các ứng dụng web như vue, electron, materialize.

2. Phát triển giao diện người dùng thân thiện và dễ sử dụng: Tạo ra một giao diện hấp dẫn, tối ưu hóa trải nghiệm người dùng với khả năng tùy chỉnh và cá nhân hóa cao.

3. Cung cấp chất lượng âm thanh cao: Đảm bảo chất lượng âm thanh tốt nhất cho người dùng, hỗ trợ nhiều định dạng âm thanh và tùy chọn chất lượng nghe nhạc khác nhau.

4. Hỗ trợ tính năng nghe nhạc ngoại tuyến: Phát triển khả năng tải nhạc về thiết bị để người dùng có thể nghe mà không cần kết nối internet.

Thông qua việc hoàn thành các mục tiêu trên, đề tài hướng tới việc tạo ra một ứng dụng nghe nhạc vượt trội, mang lại trải nghiệm tối ưu cho người dùng và góp phần thúc đẩy sự phát triển của ngành công nghiệp âm nhạc số.

## 1.3 Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu:

* + - * Vue, Electron, Materialize
      * HTML,CSS, JS

### Phạm vi nghiên cứu:

* + - * Tìm hiểu về cách xây dựng giao diện ứng dụng
      * Tìm hiểu nghiệp vụ xử lý
      * Tìm hiểu về công nghệ

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Vue.js

### **2.1.1 Giới thiệu**

Vue.js là một framework JavaScript tiến bộ được sử dụng để xây dựng giao diện người dùng (UI) và các ứng dụng đơn trang (Single-Page Applications - SPA). Được phát triển bởi Evan You và phát hành lần đầu vào năm 2014, Vue.js nổi bật nhờ sự đơn giản và tính linh hoạt của nó.

Vue (phát âm là /vjuː/, giống như view trong tiếng Anh) là một framework linh động (tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu nhằm cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện (view layer), người dùng chỉ cần sử dụng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Đồng thời, nếu kết hợp với các kỹ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng dễ dàng đáp ứng nhu cầu xây dựng các ứng dụng một trang (SPA - Single-Page Applications) có độ phức tạp cao hơn nhiều.

### 2.1.2 Các đặc tính của Vue.js

Vue.js là một framework JavaScript nổi bật với nhiều đặc tính mạnh mẽ và linh hoạt, giúp các nhà phát triển xây dựng ứng dụng web hiện đại một cách dễ dàng và hiệu quả. Dưới đây là các đặc tính chính của Vue.js:

Reactivity (Tính phản ứng):

- Vue.js có hệ thống reactivity tích hợp, cho phép tự động cập nhật giao diện người dùng khi dữ liệu thay đổi. Khi một thuộc tính trong dữ liệu thay đổi, Vue sẽ theo dõi và tự động cập nhật DOM để phản ánh sự thay đổi đó mà không cần sự can thiệp của lập trình viên.

Component-Based Architecture (Kiến trúc dựa trên thành phần):

- Vue.js tổ chức ứng dụng dưới dạng các thành phần (components) độc lập, có thể tái sử dụng. Mỗi thành phần có thể bao gồm giao diện (HTML), logic (JavaScript) và kiểu dáng (CSS) của riêng nó, giúp mã nguồn trở nên gọn gàng và dễ quản lý hơn.

Virtual DOM:

- Vue.js sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện. Thay vì cập nhật trực tiếp DOM thật, Vue.js sẽ tạo một bản sao của DOM dưới dạng Virtual DOM và chỉ thực hiện thay đổi khi cần thiết, giúp cải thiện hiệu suất của ứng dụng.

Directives (Chỉ thị):

- Vue.js cung cấp một số chỉ thị (directives) giúp bạn dễ dàng thao tác với DOM trong template, như `v-bind` để kết nối dữ liệu, `v-if` để điều kiện hiển thị, `v-for` để lặp qua các phần tử, và `v-on` để xử lý sự kiện. Các chỉ thị này giúp việc lập trình trở nên ngắn gọn và dễ hiểu.

Computed Properties (Thuộc tính tính toán):

- Vue.js hỗ trợ computed properties, cho phép bạn tạo ra các thuộc tính phụ thuộc vào các thuộc tính khác và tự động tính toán lại khi các thuộc tính đó thay đổi. Điều này giúp bạn tránh phải viết logic lặp lại và đảm bảo mã nguồn rõ ràng hơn.

Watchers (Trình theo dõi):

- Watchers trong Vue.js cho phép bạn theo dõi các thay đổi trong dữ liệu và thực hiện các hành động tương ứng. Đây là công cụ hữu ích để theo dõi và phản ứng lại các thay đổi trong trạng thái của ứng dụng.

Routing (Điều hướng):

- Vue.js có thư viện Vue Router tích hợp, giúp bạn dễ dàng quản lý các tuyến đường (routes) trong ứng dụng đơn trang (SPA). Vue Router cho phép bạn định nghĩa và điều hướng giữa các trang khác nhau mà không cần tải lại toàn bộ trang web.

State Management (Quản lý trạng thái):

- Với thư viện Vuex, Vue.js cung cấp giải pháp quản lý trạng thái tập trung cho các ứng dụng phức tạp. Vuex cho phép lưu trữ và quản lý trạng thái của ứng dụng tại một nơi duy nhất, đồng thời giúp các thành phần dễ dàng chia sẻ trạng thái với nhau.

Transition/Animation (Chuyển cảnh và hoạt ảnh):

- Vue.js hỗ trợ các hiệu ứng chuyển cảnh và hoạt ảnh, cho phép bạn dễ dàng thêm hiệu ứng khi các thành phần thêm, xóa hoặc thay đổi trạng thái. Các hiệu ứng này có thể được định nghĩa bằng CSS hoặc JavaScript.

Flexibility and Integration (Linh hoạt và tích hợp):

- Vue.js rất linh hoạt và có thể được tích hợp vào các dự án hiện có mà không cần thay đổi cấu trúc. Bạn có thể sử dụng Vue.js cho một phần nhỏ của giao diện người dùng hoặc xây dựng toàn bộ ứng dụng bằng Vue.js.

Ecosystem (Hệ sinh thái):

- Vue.js có một hệ sinh thái phong phú với các công cụ và thư viện chính thức như Vue CLI (dùng để khởi tạo và quản lý dự án), Vue Devtools (dùng để gỡ lỗi), Vue Router, Vuex, và nhiều plugin khác, giúp hỗ trợ toàn diện quá trình phát triển ứng dụng.

Lightweight (Nhẹ nhàng):

- Vue.js có kích thước nhỏ gọn, thường chỉ khoảng 20-30 KB sau khi nén, giúp ứng dụng tải nhanh hơn và tiêu thụ ít băng thông hơn, điều này rất quan trọng đối với các ứng dụng web cần tối ưu hóa tốc độ tải trang.

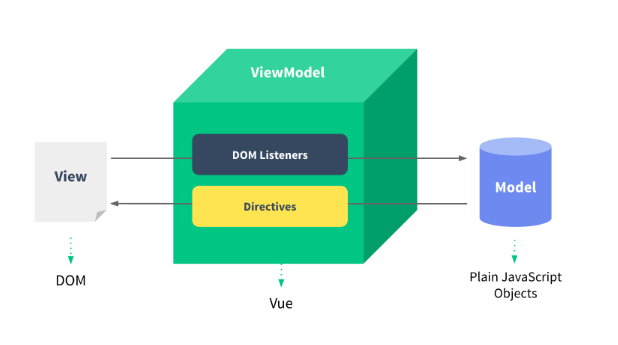


Figure 1: Mô hình xử lý của Vue

### 2.1.3 Ưu điểm và nhược điểm của Vue.js

* **Ưu điểm**

1. Dễ học và sử dụng:

- Vue.js có một cú pháp đơn giản và dễ hiểu, làm cho nó trở nên lý tưởng cho người mới bắt đầu. Ngay cả những người có ít kinh nghiệm với JavaScript cũng có thể nhanh chóng nắm bắt và sử dụng Vue.js để phát triển các ứng dụng web.

2. Tài liệu phong phú:

- Vue.js có tài liệu chính thức rất chi tiết và rõ ràng, giúp các nhà phát triển dễ dàng tìm hiểu và giải quyết các vấn đề. Tài liệu này cũng có nhiều ví dụ và hướng dẫn từng bước, làm cho quá trình học tập trở nên suôn sẻ hơn.

1. Kiến trúc dựa trên thành phần (Component-Based Architecture):

- Kiến trúc dựa trên thành phần của Vue.js giúp chia nhỏ ứng dụng thành các phần độc lập và tái sử dụng, giúp việc phát triển và bảo trì trở nên dễ dàng hơn. Các thành phần có thể được sử dụng lại và kết hợp với nhau để tạo nên các phần phức tạp của ứng dụng.

4. Kích thước nhỏ gọn:

- Vue.js có kích thước nhỏ gọn (khoảng 20-30 KB sau khi nén), giúp tải trang nhanh hơn và tối ưu hóa hiệu suất của ứng dụng, đặc biệt là trên các thiết bị di động hoặc các môi trường mạng chậm.

* **Nhược điểm**

1. Quy mô cộng đồng nhỏ hơn so với các framework lớn khác:

- Mặc dù Vue.js có một cộng đồng phát triển lớn và đang tăng trưởng, nhưng nó vẫn nhỏ hơn so với các framework như React và Angular. Điều này có thể dẫn đến việc ít tài nguyên, thư viện bên thứ ba và plugin có sẵn hơn.

2. Không được hỗ trợ bởi các công ty lớn:

- Vue.js chủ yếu được phát triển bởi cộng đồng và không có sự hỗ trợ mạnh mẽ từ các công ty công nghệ lớn như Google (Angular) hoặc Facebook (React). Điều này đôi khi gây ra lo ngại về sự ổn định và hỗ trợ lâu dài.

3. Quản lý dự án lớn có thể phức tạp:

- Mặc dù Vuex giúp quản lý trạng thái trong các ứng dụng lớn, việc quản lý một dự án rất lớn và phức tạp với nhiều thành phần có thể trở nên khó khăn. Đôi khi, việc phân chia và tổ chức các thành phần trở nên phức tạp, đặc biệt đối với những nhóm phát triển lớn.

4. Hạn chế về tài nguyên và công cụ hỗ trợ:

- So với React hoặc Angular, Vue.js có ít công cụ và tài nguyên hỗ trợ hơn từ bên thứ ba. Điều này có thể là một rào cản đối với những dự án yêu cầu các giải pháp đặc thù hoặc khi cần tìm kiếm các công cụ tối ưu hóa cụ thể.

5. Khả năng việc làm thấp hơn:

- Do Vue.js ít phổ biến hơn React hoặc Angular, số lượng việc làm yêu cầu kỹ năng Vue.js có thể ít hơn. Điều này có thể là một yếu tố quan trọng đối với những người muốn lựa chọn công nghệ dựa trên cơ hội nghề nghiệp.

### 2.1.4 Ứng dụng

Sau cùng, để có cái nhìn ‘trọn vẹn’ về Vue là gì, bạn cần biết về ứng dụng của nó.

* Single-Page Applications (SPAs) - Ứng dụng đơn trang:

- Vue.js là một lựa chọn tuyệt vời để phát triển các ứng dụng đơn trang (SPA). SPAs là những ứng dụng web tải một lần duy nhất, sau đó nội dung được cập nhật động mà không cần tải lại toàn bộ trang. Vue Router kết hợp với Vue.js giúp quản lý điều hướng giữa các trang trong SPA một cách mượt mà và hiệu quả.

* Progressive Web Applications (PWAs) - Ứng dụng web tiến bộ:

- Vue.js có thể được sử dụng để phát triển Progressive Web Applications (PWAs), kết hợp các tính năng của trang web và ứng dụng di động. PWAs hoạt động ngay cả khi ngoại tuyến, có thể cài đặt trên màn hình chính và có hiệu suất nhanh như các ứng dụng gốc. Vue CLI hỗ trợ tích hợp các công cụ cần thiết để phát triển PWAs.

* Ứng dụng quản lý nội dung (CMS) và Blog:

- Vue.js có thể được sử dụng để xây dựng các hệ thống quản lý nội dung (CMS) và blog tương tác. Giao diện người dùng động và tính năng cập nhật tự động của Vue.js giúp tạo ra trải nghiệm người dùng mượt mà khi thêm, sửa hoặc xóa nội dung.

* Dashboards và các công cụ quản lý:

- Vue.js là lựa chọn lý tưởng cho việc xây dựng các dashboard và công cụ quản lý dữ liệu. Với khả năng tương tác động và hệ thống reactivity, Vue.js cho phép hiển thị dữ liệu thời gian thực, biểu đồ, và bảng thống kê một cách nhanh chóng và hiệu quả.

* Ứng dụng thương mại điện tử:

- Vue.js có thể được sử dụng để phát triển các trang web thương mại điện tử. Nó cho phép tạo ra các giao diện người dùng tương tác với giỏ hàng động, hệ thống thanh toán, và quản lý sản phẩm, mang lại trải nghiệm mua sắm trực tuyến linh hoạt và dễ sử dụng.

* Ứng dụng cộng tác và truyền thông xã hội:

- Vue.js có thể được dùng để phát triển các ứng dụng cộng tác, như công cụ làm việc nhóm, nền tảng truyền thông xã hội, và ứng dụng nhắn tin. Với khả năng reactivity, Vue.js giúp cập nhật trạng thái và nội dung theo thời gian thực, hỗ trợ trải nghiệm người dùng tốt hơn.

* Ứng dụng di động với Vue Native:

- Vue.js không chỉ giới hạn ở ứng dụng web mà còn có thể được sử dụng để phát triển ứng dụng di động nhờ vào Vue Native. Vue Native là một framework cho phép viết mã nguồn bằng Vue.js và chuyển đổi sang các ứng dụng di động sử dụng React Native.

* Ứng dụng quản lý trạng thái phức tạp:

- Với Vuex, Vue.js có thể được sử dụng để phát triển các ứng dụng cần quản lý trạng thái phức tạp, như các hệ thống quản lý dự án, công cụ kế toán, hoặc phần mềm ERP. Vuex cung cấp một kho lưu trữ trung tâm để quản lý trạng thái ứng dụng, giúp đồng bộ hóa và chia sẻ dữ liệu giữa các thành phần dễ dàng hơn.

* Tích hợp với các ứng dụng hiện có:

- Vue.js có thể được tích hợp vào các ứng dụng web hiện có để thêm các tính năng mới mà không cần phải xây dựng lại toàn bộ hệ thống. Điều này đặc biệt hữu ích khi cần cải tiến hoặc mở rộng các trang web cũ.

* Ứng dụng học tập và giải trí:

- Vue.js có thể được sử dụng để xây dựng các ứng dụng học tập, như hệ thống quản lý học tập (LMS), ứng dụng học ngôn ngữ, hoặc trò chơi giáo dục. Tính năng tương tác và cập nhật động của Vue.js giúp tạo ra trải nghiệm học tập phong phú và hấp dẫn.

## 2.2 ElectronJS

### 2.2.1 Giới thiệu

* ElectronJS là framework viết ứng dụng đa nền tảng. Tương tự với các nền tảng cross-platform khác (ví dụ: React Native), bạn chỉ cần viết code một lần bằng Javascript là có được ứng dụng chạy trên nhiều nền tảng OS khác nhau.
* ElectronJS hỗ trợ build ứng dụng cho cả 3 OS phổ biến nhất là Window, MacOS, và Linux.
* Ưu điểm quan trọng nhất của ElectronJS là người dùng có thể tiếp tục sử dụng các công nghệ web quen thuộc như Javascript – thậm chí có thể dùng các frontend framework nổi tiếng ReactJS, VueJS… để xây dựng các ứng dụng cho desktop. Điều này khiến người dùng có cảm giác “như được trở về nhà” khi làm các dự án tạo ứng dụng cho desktop.

Về cơ bản ElectronJS gồm 3 thành phần:

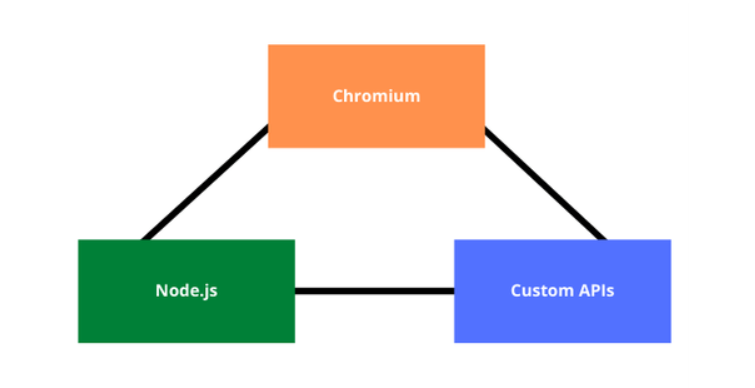


Figure 2: Cấu trúc của ElectronJS

****Chromium:****đây là thành phần chịu trách nhiệm tạo và hiển thị các trang web. Nội dung của trang web được hiển thị trong Electron.js Renderer process. Nhờ chromium mà bạn có quyền truy cập tất cả các API của trình duyệt giống như đang làm việc với một trình duyệt thực thụ vậy.

****Node.js****: Thành phần này cho phép bạn truy cập vào các tài nguyên của hệ thống. Ví dụ như tương tác với tệp tin, các chức năng của hệ điều hành.v.v…

****Custom APIs****: Bao gồm các API cho phép bạn tạo các tính năng chung của một ứng dụng desktop. Ví dụ như: tạo menu, desktop notification, phím tắt.v.v…

### 2.2.2 Các đặc tính của ElectronJS

Electron.js có nhiều đặc tính nổi bật giúp nó trở thành một lựa chọn phổ biến để phát triển ứng dụng desktop đa nền tảng. Dưới đây là các đặc tính chính của Electron.js:

* **Đa nền tảng**

- Electron.js cho phép phát triển các ứng dụng có thể chạy trên Windows, macOS, và Linux mà không cần thay đổi mã nguồn. Điều này giúp tiết kiệm thời gian và công sức trong việc phát triển và bảo trì ứng dụng.

* **Kết hợp Chromium và Node.js**

- Electron.js sử dụng Chromium để render giao diện người dùng, vì vậy bạn có thể sử dụng HTML, CSS, và JavaScript để xây dựng giao diện ứng dụng.

- Node.js được tích hợp vào Electron, cho phép bạn truy cập vào các API hệ điều hành, xử lý file, và các tác vụ máy chủ ngay từ mã JavaScript của mình.

**Sử dụng các công nghệ web**

- Electron.js cho phép bạn tận dụng các công nghệ web hiện đại, chẳng hạn như HTML5, CSS3, và JavaScript. Điều này làm cho việc phát triển ứng dụng trở nên quen thuộc và dễ dàng hơn đối với các nhà phát triển web.

**Khả năng mở rộng với npm**

- Electron.js có thể sử dụng hàng ngàn gói npm, cung cấp một hệ sinh thái phong phú các thư viện và công cụ mà bạn có thể tích hợp vào ứng dụng của mình.

**Hỗ trợ quản lý bộ nhớ và hiệu suất**

- Electron.js cung cấp các công cụ để quản lý bộ nhớ và tối ưu hiệu suất ứng dụng, giúp đảm bảo rằng các ứng dụng chạy mượt mà trên mọi nền tảng.

**Quyền truy cập vào API hệ thống**

- Với Electron.js, bạn có quyền truy cập vào các API hệ thống, cho phép bạn thực hiện các tác vụ như quản lý file, truy cập phần cứng, thông báo hệ thống, và nhiều hơn nữa, điều mà các ứng dụng web thông thường không thể làm được.

**Hỗ trợ cập nhật tự động**

- Electron.js hỗ trợ tính năng cập nhật tự động, cho phép ứng dụng của bạn được cập nhật lên phiên bản mới nhất mà không cần người dùng phải tải về và cài đặt thủ công.

**Debugging và Development Tools**

- Electron.js tích hợp với DevTools của Chromium, cho phép bạn dễ dàng debug, kiểm tra hiệu suất, và phân tích ứng dụng của mình trong quá trình phát triển.

**Cộng đồng mạnh mẽ và tài liệu phong phú**

- Electron.js có một cộng đồng lớn với nhiều tài liệu, hướng dẫn, và ví dụ mã nguồn mở, giúp dễ dàng học hỏi và giải quyết các vấn đề trong quá trình phát triển.

**Ứng dụng thực tiễn**

- Nhiều ứng dụng nổi tiếng được phát triển bằng Electron.js như Visual Studio Code, Slack, Discord, Atom, và nhiều ứng dụng khác, chứng minh khả năng của Electron.js trong việc phát triển các ứng dụng desktop phức tạp và đa nền tảng.

Các đặc tính này làm cho Electron.js trở thành một công cụ mạnh mẽ và linh hoạt cho việc phát triển ứng dụng desktop hiện đại.

### 2.2.3 Ưu điểm và nhược điểm của ElectronJS

**Ưu điểm**

1.Đa nền tảng:

- Electron.js cho phép viết mã một lần và chạy trên nhiều hệ điều hành (Windows, macOS, Linux) mà không cần thay đổi mã nguồn, tiết kiệm thời gian và công sức.

2.Sử dụng công nghệ web quen thuộc:

- Các nhà phát triển web có thể sử dụng kiến thức về HTML, CSS, và JavaScript để phát triển ứng dụng desktop, giúp dễ dàng chuyển đổi từ phát triển web sang phát triển ứng dụng desktop.

3. Truy cập API hệ điều hành:

- Electron.js cho phép truy cập vào các API của hệ điều hành, giúp các ứng dụng có thể thực hiện các tác vụ như quản lý file, thông báo hệ thống, và truy cập phần cứng.

4. Hệ sinh thái phong phú:

- Electron.js tích hợp với Node.js, cho phép sử dụng hàng ngàn thư viện từ npm, mở rộng khả năng của ứng dụng.

5. Cộng đồng và tài liệu hỗ trợ mạnh mẽ:

- Electron.js có một cộng đồng lớn và nhiều tài liệu, giúp nhà phát triển dễ dàng tìm kiếm hỗ trợ, học hỏi và giải quyết vấn đề.

6. Công cụ phát triển và gỡ lỗi tích hợp:

- Electron.js tích hợp các công cụ DevTools của Chromium, giúp dễ dàng gỡ lỗi, tối ưu hóa hiệu suất và phân tích ứng dụng.

7. Tính năng cập nhật tự động:

- Electron.js hỗ trợ cập nhật tự động, giúp người dùng luôn có phiên bản mới nhất của ứng dụng mà không cần thao tác phức tạp.

**Nhược điểm**

1. Kích thước ứng dụng lớn:

- Do Electron.js tích hợp cả Chromium và Node.js trong mỗi ứng dụng, kích thước của ứng dụng thường khá lớn, ngay cả đối với các ứng dụng đơn giản.

2. Tiêu thụ tài nguyên cao:

- Electron.js có thể tiêu tốn nhiều tài nguyên hệ thống hơn so với các ứng dụng được viết bằng các ngôn ngữ lập trình gốc (native languages) như C++ hay Swift. Điều này có thể dẫn đến hiệu suất chậm hơn, đặc biệt là trên các máy tính có cấu hình thấp.

3. Bảo mật:

- Vì Electron.js cho phép truy cập vào các API hệ điều hành, các ứng dụng có thể gặp rủi ro bảo mật nếu không được xây dựng cẩn thận. Cần phải chú ý đến các vấn đề như quản lý quyền và thực hiện các biện pháp bảo mật tốt.

4. Khả năng mở rộng:

- Với các ứng dụng phức tạp và lớn, Electron.js có thể không phải là lựa chọn tối ưu do khó quản lý và tối ưu hóa, đặc biệt là khi cần xử lý nhiều tác vụ nặng về tài nguyên.

5. Hiệu suất trên thiết bị di động:

- Electron.js không phải là công cụ tốt nhất để phát triển ứng dụng cho các thiết bị di động, do hạn chế về hiệu suất và khả năng tối ưu hóa cho phần cứng di động.

6. Sự phụ thuộc vào Chromium:

- Việc sử dụng Chromium có thể dẫn đến các vấn đề về hiệu suất và bộ nhớ, đặc biệt là khi ứng dụng sử dụng nhiều tab hoặc cửa sổ.

### 2.2.4 Ứng dụng

Electron.js được sử dụng rộng rãi trong việc phát triển các ứng dụng desktop đa nền tảng nhờ vào khả năng kết hợp các công nghệ web với quyền truy cập vào các API hệ điều hành. Dưới đây là một số ứng dụng nổi tiếng được xây dựng bằng Electron.js:

**1. Visual Studio Code**

- Mô tả: Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn miễn phí và phổ biến do Microsoft phát triển. Nó hỗ trợ nhiều ngôn ngữ lập trình, cung cấp các tính năng mạnh mẽ như gỡ lỗi, tích hợp Git, và có thể mở rộng với các tiện ích mở rộng (extensions).

- Lý do sử dụng Electron.js: Sử dụng Electron.js giúp VS Code có thể chạy trên nhiều hệ điều hành khác nhau, từ Windows, macOS đến Linux, với một mã nguồn duy nhất.

**2. Slack**

- Mô tả: Slack là một công cụ giao tiếp và làm việc nhóm trực tuyến, cho phép người dùng trò chuyện, chia sẻ tệp và tích hợp với nhiều dịch vụ khác nhau như Google Drive, Trello, và GitHub.

- \*\*Lý do sử dụng Electron.js\*\*: Slack cần một giao diện người dùng hiện đại và khả năng chạy trên nhiều nền tảng khác nhau, điều mà Electron.js có thể cung cấp.

**3. Discord**

- Mô tả: Discord là một ứng dụng chat tập trung vào cộng đồng game thủ, cho phép người dùng tạo các máy chủ chat riêng biệt, trò chuyện qua văn bản, giọng nói và video.

- Lý do sử dụng Electron.js: Nhờ Electron.js, Discord có thể cung cấp một trải nghiệm người dùng đồng nhất trên các nền tảng khác nhau và dễ dàng cập nhật tính năng mới.

**4. Atom**

- Mô tả: Atom là một trình soạn thảo mã nguồn mở do GitHub phát triển, nổi tiếng với khả năng tùy biến cao và tích hợp tốt với Git.

- Lý do sử dụng Electron.js: Atom được xây dựng để trở thành một trình soạn thảo hackable, và Electron.js cho phép dễ dàng mở rộng và tùy chỉnh thông qua các gói (packages).

**5. Trello**

- Mô tả: Trello là một công cụ quản lý dự án và công việc trực tuyến, sử dụng bảng (board) và thẻ (card) để tổ chức và quản lý nhiệm vụ.

- Lý do sử dụng Electron.js: Electron.js cho phép Trello có một ứng dụng desktop đầy đủ chức năng, cung cấp trải nghiệm tương tự như trên trình duyệt web nhưng với khả năng làm việc offline và quản lý thông báo tốt hơn.

**6. WhatsApp Desktop**

- Mô tả: Ứng dụng desktop của WhatsApp cung cấp khả năng nhắn tin, gọi điện và chia sẻ tệp tương tự như phiên bản di động, nhưng với sự tiện lợi của một ứng dụng desktop.

- Lý do sử dụng Electron.js: Electron.js giúp WhatsApp duy trì một ứng dụng desktop nhẹ và đa nền tảng, đồng bộ hóa tốt với phiên bản di động.

1. **Figma**

- Mô tả: Figma là một công cụ thiết kế giao diện người dùng (UI/UX) dựa trên đám mây, cho phép làm việc cộng tác trong thời gian thực.

- Lý do sử dụng Electron.js: Figma sử dụng Electron.js để cung cấp một ứng dụng desktop với hiệu suất cao, đồng thời tích hợp liền mạch với các tính năng dựa trên đám mây.

**8. Postman**

- Mô tả: Postman là một công cụ API phổ biến cho phép các nhà phát triển thiết kế, thử nghiệm và quản lý các API.

- Lý do sử dụng Electron.js: Electron.js giúp Postman cung cấp một ứng dụng desktop đa nền tảng, giúp nhà phát triển có thể dễ dàng làm việc với API từ mọi nơi.

**9. Skype**

- Mô tả: Skype là một ứng dụng gọi video và nhắn tin phổ biến, được sử dụng rộng rãi cho cả mục đích cá nhân và công việc.

- Lý do sử dụng Electron.js: Sử dụng Electron.js giúp Skype duy trì một giao diện người dùng nhất quán và tích hợp tốt với các tính năng của hệ điều hành trên mọi nền tảng.

**10. GitHub Desktop**

- Mô tả: GitHub Desktop là một ứng dụng giúp người dùng quản lý các kho mã nguồn GitHub của mình, cung cấp giao diện người dùng thân thiện cho các tác vụ như commit, pull request, và merge.

- Lý do sử dụng Electron.js: Electron.js giúp GitHub Desktop dễ dàng triển khai trên nhiều hệ điều hành, cung cấp tính năng đồng nhất và khả năng mở rộng mạnh mẽ.

## 2.3 Materialize CSS

### 2.3.1 Giới thiệu

Materialize là một framework front-end hiện đại, dựa trên Material Design. Để sử dụng trong các dự án web, nó cung cấp tùy chọn của cả hai CSS cũng như các tập tin nguồn SCSS, cùng với JavaScript, biểu tượng Material Design và font chữ Roboto. Thành phần bao gồm những cái cơ bản như hệ thống lưới, các biểu mẫu, các nút, thanh điều hướng và thẻ.

### 2.3.2 Các đặc tính của Materialize

* Material Design

- Dựa trên nguyên tắc Material Design của Google: Materialize tuân theo các hướng dẫn về thiết kế Material Design, mang đến giao diện người dùng hiện đại, trực quan và dễ sử dụng.

* Responsive Design

- Hỗ trợ responsive: Materialize tự động điều chỉnh bố cục và các thành phần giao diện để phù hợp với nhiều kích thước màn hình khác nhau, từ máy tính để bàn đến thiết bị di động.

* Grid System

- Hệ thống lưới 12 cột: Giúp dễ dàng bố trí các thành phần trên trang web một cách linh hoạt và đồng đều.

* Pre-styled Components

- Các thành phần giao diện có sẵn: Materialize cung cấp nhiều thành phần giao diện có sẵn như nút, biểu mẫu, thẻ, thanh điều hướng, modals, dropdowns, và nhiều thành phần khác. Điều này giúp giảm bớt thời gian phát triển.

* Customizable

- Dễ dàng tùy chỉnh: Các thành phần trong Materialize có thể dễ dàng tùy chỉnh để phù hợp với yêu cầu của dự án. Bạn có thể thay đổi màu sắc, kiểu chữ và các yếu tố khác một cách linh hoạt.

* JavaScript Components

- Hỗ trợ các thành phần động: Materialize đi kèm với nhiều thành phần JavaScript giúp tạo ra các hiệu ứng động như sliders, modals, tabs, tooltips, v.v., mà không cần phải viết nhiều mã JavaScript từ đầu.

* Icons and Typography

- Biểu tượng và font chữ: Materialize tích hợp biểu tượng Material Design và font chữ Roboto, giúp tạo nên giao diện đồng bộ và hiện đại.

* Cross-browser Compatibility

- Tương thích trình duyệt: Materialize đảm bảo giao diện hoạt động ổn định trên nhiều trình duyệt phổ biến như Chrome, Firefox, Safari, và Internet Explorer.

* Documentation

- Tài liệu chi tiết: Materialize cung cấp tài liệu phong phú, chi tiết và dễ hiểu, hỗ trợ các nhà phát triển trong việc triển khai và tùy chỉnh framework này.

* Community and Support

- Cộng đồng và hỗ trợ: Materialize có một cộng đồng người dùng lớn và nhiều tài nguyên hỗ trợ, giúp các nhà phát triển dễ dàng tìm kiếm giải pháp cho các vấn đề phát sinh.

Những đặc tính này giúp Materialize trở thành một lựa chọn tốt cho việc phát triển các ứng dụng web hiện đại với giao diện người dùng đẹp và thân thiện.

### 2.3.3 Ưu điểm và nhược điểm

**Ưu điểm**

* Thiết kế hiện đại và nhất quán:

- Materialize tuân theo các nguyên tắc của Material Design, mang lại giao diện người dùng hiện đại, trực quan và nhất quán trên tất cả các thiết bị và nền tảng.

* Thân thiện với người dùng:

- Các thành phần UI của Materialize được thiết kế để dễ dàng sử dụng và hiểu, giúp cải thiện trải nghiệm người dùng.

* Responsive Design:

- Materialize hỗ trợ responsive design, giúp giao diện tự động điều chỉnh để phù hợp với các kích thước màn hình khác nhau, từ máy tính để bàn đến thiết bị di động.

* Component sẵn có:

- Framework cung cấp nhiều thành phần giao diện sẵn có như buttons, forms, cards, modals, và nhiều thành phần khác, giúp giảm thời gian phát triển.

* Dễ dàng tùy chỉnh:

- Các thành phần của Materialize có thể dễ dàng tùy chỉnh theo yêu cầu của dự án, từ màu sắc đến kiểu chữ, cho phép tạo ra giao diện độc đáo.

* Tài liệu phong phú:

- Materialize đi kèm với tài liệu chi tiết và dễ hiểu, hỗ trợ các nhà phát triển trong việc triển khai và tùy chỉnh các thành phần của framework.

* Tích hợp biểu tượng và font chữ:

- Materialize tích hợp sẵn các biểu tượng Material Design và font chữ Roboto, giúp duy trì sự đồng bộ và thẩm mỹ cho giao diện.

**Nhược điểm**

* Hạn chế trong tùy chỉnh sâu:

- Mặc dù Materialize cho phép tùy chỉnh, nhưng nếu bạn muốn thực hiện các thay đổi sâu rộng hoặc tạo ra một giao diện hoàn toàn khác biệt, bạn có thể gặp khó khăn do sự phụ thuộc vào các nguyên tắc Material Design.

* Kích thước lớn:

- Materialize đi kèm với nhiều thành phần và tính năng, có thể làm tăng kích thước của trang web. Điều này có thể ảnh hưởng đến hiệu suất tải trang, đặc biệt trên các kết nối internet chậm.

* Cộng đồng nhỏ hơn so với các framework khác:

- So với các framework như Bootstrap, cộng đồng người dùng và tài nguyên hỗ trợ của Materialize nhỏ hơn, có thể làm hạn chế khả năng tìm kiếm giải pháp cho các vấn đề phát sinh.

* Sự phụ thuộc vào Material Design:

- Do Materialize dựa trên Material Design, nó có thể giới hạn khả năng sáng tạo của bạn nếu bạn muốn thiết kế giao diện theo phong cách khác.

* Khả năng tương thích:

- Một số tính năng hoặc thành phần của Materialize có thể gặp vấn đề về tương thích với các trình duyệt cũ hoặc không được hỗ trợ đầy đủ.

* Cập nhật không thường xuyên:

- Tần suất cập nhật của Materialize không đều đặn như một số framework khác, có thể dẫn đến việc thiếu các tính năng mới hoặc sửa lỗi kịp thời.

### 2.3.4 Ứng dụng

Materialize có thể được ứng dụng rộng rãi trong nhiều loại dự án web khác nhau nhờ vào các tính năng linh hoạt và thiết kế hiện đại của nó. Dưới đây là một số ứng dụng phổ biến của Materialize:

1. Xây dựng các trang web doanh nghiệp:

- Giao diện chuyên nghiệp: Materialize giúp tạo ra giao diện trang web doanh nghiệp chuyên nghiệp, sạch sẽ và dễ sử dụng, phù hợp với các công ty muốn có một trang web hiện đại và dễ quản lý.

- Tính năng tương tác: Các thành phần như modals, tooltips, dropdowns giúp tạo ra các trang web có nhiều tương tác, nâng cao trải nghiệm người dùng.

2. Thiết kế các ứng dụng web (Web Apps):

- Giao diện đồng nhất: Đối với các ứng dụng web cần có giao diện đồng nhất trên nhiều nền tảng, Materialize cung cấp một framework mạnh mẽ và linh hoạt để xây dựng giao diện người dùng nhất quán.

- Component sẵn có: Việc sử dụng các component như forms, buttons, và navbars giúp tiết kiệm thời gian phát triển và đảm bảo giao diện thân thiện với người dùng.

3. Phát triển trang landing page:

- Thiết kế đẹp mắt: Materialize là một lựa chọn lý tưởng để tạo các trang landing page có thiết kế đẹp mắt và hiện đại, thu hút người dùng ngay từ cái nhìn đầu tiên.

- Tùy chỉnh dễ dàng: Các công cụ tùy chỉnh của Materialize cho phép bạn dễ dàng điều chỉnh giao diện để phù hợp với thương hiệu hoặc mục tiêu của chiến dịch marketing.

1. Xây dựng các trang blog hoặc portfolio cá nhân:

- Thiết kế đơn giản và trực quan: Với Materialize, bạn có thể dễ dàng tạo ra các trang blog hoặc portfolio cá nhân với giao diện đẹp mắt và dễ sử dụng.

- Responsive design: Khả năng responsive của Materialize đảm bảo trang web của bạn sẽ hiển thị tốt trên mọi thiết bị, từ máy tính đến điện thoại di động.

5. Hệ thống quản lý nội dung (CMS) hoặc bảng điều khiển quản trị (Admin Dashboards):

- Giao diện quản trị: Materialize có thể được sử dụng để xây dựng các giao diện quản trị (admin dashboards) với các thành phần quản lý nội dung như biểu đồ, bảng, và form, giúp người dùng quản lý dữ liệu một cách hiệu quả.

- Dễ dàng mở rộng: Framework này dễ dàng mở rộng và tùy chỉnh, phù hợp cho việc phát triển các hệ thống quản lý nội dung (CMS).

6. Thiết kế các trang thương mại điện tử (eCommerce):

- Giao diện người dùng thân thiện: Materialize cung cấp các thành phần cần thiết để xây dựng giao diện người dùng cho các trang thương mại điện tử, từ danh sách sản phẩm đến giỏ hàng và thanh toán.

- Tích hợp biểu tượng và hiệu ứng: Sử dụng các biểu tượng Material Design và các hiệu ứng động giúp trang web trở nên sinh động và hấp dẫn hơn.

7. Dự án giáo dục và học tập:

- Tài liệu dễ hiểu: Với tài liệu chi tiết và dễ hiểu, Materialize là một công cụ tuyệt vời để các sinh viên hoặc người mới học lập trình sử dụng trong các dự án học tập.

- Phát triển kỹ năng thiết kế: Việc sử dụng Materialize giúp người học nắm vững các nguyên tắc cơ bản của thiết kế giao diện người dùng và phát triển kỹ năng thiết kế web.

Materialize là một lựa chọn lý tưởng cho những dự án cần giao diện đẹp, hiện đại và tuân theo các nguyên tắc của Material Design. Nó phù hợp cho cả những người mới bắt đầu và những nhà phát triển có kinh nghiệm muốn tạo ra các trang web hoặc ứng dụng web nhanh chóng và hiệu quả.

# CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG

## 3.1 Phân tích nghiệp vụ

### 3.1.1 Đặc tả Actor

|  |  |
| --- | --- |
| **TÊN TÁC NHÂN** | **VAI TRÒ** |
| **Người dùng** | Người sử dụng ứng dụng nghe nhạc. |
| **Hệ thống(Tệp nhạc)** | Hệ thống lưu trữ nội bộ hoặc bên ngoài từ đó các tệp nhạc được tải vào. |

### 3.1.2 Đặc tả tác vụ

|  |  |
| --- | --- |
| **TÊN TÁC VỤ** | **VAI TRÒ** |
| Phát nhạc | Người dùng có thể phát một bài nhạc đã chọn. |
| Tạm dừng nhạc | Người dùng có thể tạm dừng bài nhạc đang phát. |
| Tua nhạc | Người dùng có thể tua (bỏ qua hoặc tua lại) trong bài nhạc đang phát. |
| Thêm nhạc vào thư viện | Người dùng có thể thêm các tệp nhạc từ hệ thống vào thư viện nhạc của ứng dụng. |
| Xem thư viện nhạc | Người dùng có thể duyệt danh sách tất cả các bài nhạc có sẵn trong thư viện. |
| Tạo danh sách phát | Người dùng có thể tạo danh sách phát bằng cách chọn các bài nhạc cụ thể từ thư viện. |
| Phát từ danh sách | Người dùng có thể phát một bài nhạc từ danh sách phát đã chọn. |
| Xóa nhạc/ danh sách phát | Người dùng có thể xóa một bài nhạc hoặc danh sách phát khỏi thư viện. |
| Xem những bài đã phát gần đây | Người dùng có thể xem danh sách các bài nhạc đã phát gần đây. |
| Bài kế tiếp/ bài trước | Người dùng có thể bỏ qua đến bài kế tiếp hoặc quay lại bài trước trong danh sách phát hoặc hàng đợi. |
| Lưu file nhạc | Người dùng có thể lưu các hồ sơ, danh sách phát và cài đặt thư viện hiện tại. |
| Tải file nhạc | Người dùng có thể tải các hồ sơ, danh sách phát và cài đặt thư viện đã lưu trước đó. |

### 3.1.3 Mô hình Usecase



Figure 3: Mô hình Usecase của ứng dụng

### 3.1.4 Đặc tả Usecase

#### Phát nhạc

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Phát nhạc | **Mã** | UC01 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mô tả** | Cho phép người dùng có thể phát một bài nhạc đã chọn. | | |
| **Tác nhân** | Người dùng | **Trigger** | Người dùng bấm vào nút play (phát nhạc). |
| **Tiền sự kiện** | Bài nhạc đã có sẵn trong thư viện. | | |
| **Hậu sự kiện** | Bài nhạc được phát. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tác nhân** | | **Hệ thống** | |
| **Luồng chính** | | | |
| 1 | Người dùng chọn bài nhạc từ danh sách nhạc. |  |  |
|  |  | 2 | Hệ thống tải bài nhạc được chọn. |
|  |  | 3 | Hệ thống bắt đầu phát bài nhạc. |
| **Luồng sự kiện thay thế** | | | |
|  |  | 4 | Nếu bài nhạc không tồn tại hoặc có lỗi trong khi tải, hệ thống thông báo lỗi. |

Table 1: Usecase phát nhạc

#### Tạm dừng nhạc

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Tạm dừng nhạc | **Mã** | UC02 |
| **Mô tả** | Cho phép người dùng tạm dừng bài nhạc đang phát. | | |
| **Tác nhân** | Người dùng | **Trigger** | Người dùng bấm vào nút pause( Tạm dùng). |
| **Tiền sự kiện** | Bài nhạc đang được phát. | | |
| **Hậu sự kiện** | Bài nhạc tạm dừng | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tác nhân** | | **Hệ thống** | |
| **Luồng chính** | | | |
| 1 | Người dùng nhấn nút tạm dừng. |  |  |
|  |  | 2 | Hệ thống tạm dừng phát nhạc. |

#### **Tua nhạc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Tua nhạc | **Mã** | UC03 |
| **Mô tả** | Cho phép người dùng tua đến một vị trí cụ thể trong bài nhạc. | | |
| **Tác nhân** | Người dùng | **Trigger** | Người dùng bấm hoặc kéo đến vị trí bất kì trên thanh tua. |
| **Tiền sự kiện** | Bài nhạc đang được phát. | | |
| **Hậu sự kiện** | Bài nhạc được phát từ vị trí mới được chọn. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tác nhân** | | **Hệ thống** | |
| **Luồng chính** | | | |
| 1 | Người dùng kéo thanh tua đến vị trí mong muốn. |  |  |
|  |  | 2 | Hệ thống cập nhật vị trí phát nhạc theo yêu cầu. |

1. **Thêm nhạc vào thư viện**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Thêm nhạc vào thư viện | **Mã** | UC04 |
| **Mô tả** | Cho phép người dùng thêm một tệp nhạc từ hệ thống vào thư viện nhạc. | | |
| **Tác nhân** | Người dùng | **Trigger** | Click chuột phải vào bài hát mong muốn sau đó sẽ hiện thư viện nhạc và bấm chọn. |
| **Tiền sự kiện** | Người dùng có quyền truy cập vào hệ thống tệp nhạc. | | |
| **Hậu sự kiện** | Tệp nhạc được thêm vào thư viện. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng chọn tệp nhạc cần thêm. |  |  |
|  |  | 2 | Hệ thống kiểm tra tệp nhạc và thêm vào thư viện. |
| Luồng sự kiện thay thế: | | | |
|  |  | 3 | Nếu tệp nhạc không hợp lệ, hệ thống hiển thị thông báo lỗi. |

#### Xem thư viện nhạc

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Xem thư viện nhạc | **Mã** | UC05 |
| **Mô tả** | Cho phép người dùng duyệt qua danh sách các bài nhạc có trong thư viện. | | |
| **Tác nhân** | Người dùng | **Trigger** | Click vào phần Playlists phía bên trái app để xem thư viện nhạc đã được tạo. |
| **Tiền sự kiện** | Thư viện đã có các tệp nhạc. | | |
| **Hậu sự kiện** | Người dùng có thể xem và chọn các bài nhạc từ thư viện. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng mở thư viện nhạc. |  |  |
|  |  | 2 | Hệ thống hiển thị danh sách các bài nhạc có sẵn. |

#### Tạo danh sách phát

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Tạo danh sách phát | **Mã** | UC06 |
| **Mô tả** | Cho phép người dùng tạo một danh sách phát mới từ các bài nhạc trong thư viện. | | |
| **Tác nhân** | Người dùng | **Trigger** | Click vào biểu tượng ≡ ở góc phải màn hình và bấm chọn create playlist |
| **Tiền sự kiện** | Thư viện đã có các tệp nhạc. | | |
| **Hậu sự kiện** | Danh sách phát mới được tạo và lưu. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng chọn tạo danh sách phát mới. |  |  |
| 2 | Người dùng chọn các bài nhạc từ thư viện để thêm vào danh sách phát. |  |  |
|  |  | 3 | Hệ thống lưu danh sách phát mới. |

1. **Phát nhạc từ danh sách phát**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Phát nhạc từ danh sách phát | **Mã** | UC07 |
| **Mô tả** | Cho phép người dùng chọn một bài nhạc từ danh sách phát để phát. | | |
| **Tác nhân** | Người dùng | **Trigger** | Bấm vào nút play trên danh sách phát để phát nhạc. |
| **Tiền sự kiện** | Danh sách phát đã được tạo và có các bài nhạc. | | |
| **Hậu sự kiện** | Bài nhạc được phát từ danh sách phát. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng mở một danh sách phát. |  |  |
| 2 | Người dùng chọn một bài nhạc từ danh sách để phát. |  |  |
|  |  | 3 | Hệ thống tải và phát bài nhạc đã chọn. |

1. **Xóa nhạc/ danh sách phát**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Xóa nhạc/ danh sách phát | **Mã** | UC08 |
| **Mô tả** | Cho phép người dùng xóa một bài nhạc hoặc danh sách phát khỏi thư viện. | | |
| **Tác nhân** | Người dùng | **Trigger** | Click chuột phải vào bài hát hoặc danh sách phát sau đó bấm dấu X để xóa. |
| **Tiền sự kiện** | Thư viện hoặc danh sách phát có nội dung để xóa. | | |
| **Hậu sự kiện** | Bài nhạc hoặc danh sách phát bị xóa khỏi thư viện. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng chọn bài nhạc hoặc danh sách phát cần xóa. |  |  |
|  |  | 2 | Hệ thống xác nhận việc xóa. |
|  |  | 3 | Hệ thống xóa bài nhạc hoặc danh sách phát khỏi thư viện. |
| Luồng sự kiện thay thế: | | | |
| 4 | Nếu người dùng hủy yêu cầu xóa, hệ thống giữ nguyên trạng thái hiện tại. |  |  |

1. **Xem những bài hát đã phát gần đây**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Xem những bài hát gần đây | **Mã** | UC09 |
| **Mô tả** | Cho phép người dùng xem danh sách các bài nhạc đã phát gần đây. | | |
| **Tác nhân** | Người dùng | **Trigger** | Recently played hiện ở góc trái app. |
| **Tiền sự kiện** | Người dùng đã phát nhạc trước đó. | | |
| **Hậu sự kiện** | Người dùng có thể xem và phát lại các bài nhạc từ danh sách gần đây. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng mở mục "Recently Played". |  |  |
|  |  | 2 | Hệ thống hiển thị danh sách top 10 các bài nhạc đã phát gần đây. |

1. **Lưu file nhạc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Lưu file nhạc và danh sách phát hiện tại. | **Mã** | UC10 |
| **Mô tả** | Cho phép người dùng lưu hồ sơ và danh sách phát hiện tại. | | |
| **Tác nhân** | Người dùng | **Trigger** | Bấm chuột phải vào bài nhạc hoặc danh sách phát và bấm save. |
| **Tiền sự kiện** | Hồ sơ và danh sách phát đã được tạo và muốn lưu lại. | | |
| **Hậu sự kiện** | Hồ sơ và danh sách phát được lưu. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng chọn lưu hồ sơ. |  |  |
|  |  | 2 | Hệ thống lưu lại hồ sơ, danh sách phát và cài đặt. |

1. **Tải file nhạc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Tải lại( load) file nhạc | **Mã** | UC11 |
| **Mô tả** | Cho phép người dùng tải lại hồ sơ và danh sách phát đã lưu trước đó. | | |
| **Tác nhân** | Người dùng | **Trigger** | Bấm vào nút load để khôi phục lại bài hát. |
| **Tiền sự kiện** | Hồ sơ và danh sách phát đã được lưu trước đó. | | |
| **Hậu sự kiện** | Hồ sơ và danh sách phát được khôi phục. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tác Nhân | | Hệ Thống | |
| Luồng chính: | | | |
| 1 | Người dùng chọn tải hồ sơ. |  |  |
|  |  | 2 | Hệ thống tải lại hồ sơ, danh sách phát và cài đặt từ bộ nhớ. |