BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing symbol, logo, text, yellow

Description automatically generated=====\*\*\*=====**

**BÁO CÁO BÀI TẬP LỚN HỌC PHẦN**

**THỰC TẬP CƠ SỞ NGÀNH**

**ĐỀ TÀI**

**TÌM HIỂU FLUTTER VÀ ỨNG DỤNG VÀO XÂY DỰNG APP NGHE NHẠC**

**Giảng viên hướng dẫn: Vũ Việt Thắng**

**Lớp: 20222IT6055002 Nhóm: 09**

**Thành viên trong nhóm Mã sinh viên**

1. **Nguyễn Tiến Tú Anh 2021604282**
2. **Chử Thị Ánh 2021607049**
3. **Nguyễn Bình Dương 2021605757**
4. **Nguyễn Diệu Linh 2021603421**
5. **Nguyễn Thị Kim Ngân 2021604905**

***Hà Nội - 2023***

# MỤC LỤC

[MỤC LỤC 1](#_Toc137631270)

[DANH MỤC HÌNH ẢNH 3](#_Toc137631271)

[DANH MỤC BIỂU ĐỒ 4](#_Toc137631272)

[LỜI MỞ ĐẦU 5](#_Toc137631273)

[LỜI CẢM ƠN 7](#_Toc137631274)

[CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU 8](#_Toc137631275)

[1.1. Giới thiệu về Flutter 8](#_Toc137631276)

[1.1.1. Flutter là gì? 8](#_Toc137631277)

[1.1.2. Tại sao lại là Flutter? 8](#_Toc137631278)

[1.1.3. Đặc điểm nổi bật 9](#_Toc137631279)

[1.1.4. Kết luận 13](#_Toc137631280)

[1.2. Tìm hiểu về ngôn ngữ Dart 13](#_Toc137631281)

[1.2.1. Dart là gì? 13](#_Toc137631282)

[1.2.2. Lịch sử 14](#_Toc137631283)

[1.2.3. Tại sao lại sử dụng Dart? 14](#_Toc137631284)

[1.2.4. Những điểm chính cần nhớ 15](#_Toc137631285)

[1.2.5. Tính năng của Dart 15](#_Toc137631286)

[1.3. Visual Studio Code 19](#_Toc137631287)

[1.3.1. Visual studio code là gì? 19](#_Toc137631288)

[1.3.3. Lý do bạn nên sử dụng Visual Studio Code 20](#_Toc137631289)

[1.3.3.1. Hoạt động đa nền tảng 20](#_Toc137631290)

[1.3.3.2. Đơn giản trong việc chỉnh sửa, xây dựng và gỡ lỗi 20](#_Toc137631291)

[1.3.3.3. Sự hỗ trợ của cộng đồng GitHub 21](#_Toc137631292)

[1.3.3.4. Kiến trúc cực mạnh mẽ và có thể mở rộng 21](#_Toc137631293)

[1.3.3.5. Khởi chạy ứng dụng Flutter 22](#_Toc137631294)

[1.4. Figma 24](#_Toc137631295)

[1.4.1. Figma là gì? 24](#_Toc137631296)

[1.4.2. Tại sao lại là Figma? 25](#_Toc137631297)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 27](#_Toc137631298)

[2.1. Đặc tả yêu cầu hệ thống 27](#_Toc137631299)

[2.1.1. Yêu cầu chức năng 27](#_Toc137631300)

[2.1.2. Yêu cầu phi chức năng 27](#_Toc137631301)

[2.2. Phân tích và thiết kế 28](#_Toc137631302)

[2.2.1. Biểu đồ Use-case 28](#_Toc137631303)

[2.2.2. Biểu đồ hoạt động 28](#_Toc137631304)

[2.2.3. Biểu đồ tuần tự 32](#_Toc137631305)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 35](#_Toc137631306)

[3.1. Giao diện người dùng 35](#_Toc137631307)

[3.1.1. Giao diện đăng ký, đăng nhập 35](#_Toc137631308)

[3.1.2. Giao diện chính 36](#_Toc137631309)

[3.1.3. Giao diện tìm kiếm 37](#_Toc137631310)

[3.1.4. Các giao diện liên quan 38](#_Toc137631311)

[3.1.5. Giao diện phát nhạc 39](#_Toc137631312)

[3.2. Giao diện quản trị 40](#_Toc137631313)

[CHƯƠNG 4. KẾT QUẢ 41](#_Toc137631315)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Hình ảnh mô tả Flutter chạy trên máy ảo 11](#_Toc137497183)

[Hình 1.2 Hình ảnh mô tả UI Flutter 13](#_Toc137497184)

[Hình 1.3 Giao diện Visual Studio Code 21](#_Toc137497185)

[Hình 1.4 Các phím tắt trên VS Code 23](#_Toc137497186)

[Hình 1.5 Hình minh họa chọn trình duyệt để khởi động 24](#_Toc137497187)

[Hình 1.6 Hình ảnh giao diện của app 25](#_Toc137497188)

[Hình 1.7 Chạy lệnh flutter devices 25](#_Toc137497189)

[Hình 1.8 Hình ảnh giao diện app 26](#_Toc137497190)

[Hình 1.9 Hình ảnh giao diện của trang web figma 27](#_Toc137497191)

[Hình 3.1 Giao diện đăng ký, đăng nhập 38](#_Toc137497192)

[Hình 3.2 Giao diện chính 39](#_Toc137497193)

[Hình 3.3 Giao diện tìm kiếm 40](#_Toc137497194)

[Hình 3.4 Các giao diện liên quan 41](#_Toc137497195)

[Hình 3.5 Giao diện phát nhạc 42](#_Toc137497196)

[Hình 3.6 Giao diện chính 43](#_Toc137497197)

# 

# 

# 

# 

# 

# 

# 

# 

# DANH MỤC BIỂU ĐỒ

[Biểu đồ 2.1 Biểu đồ use case tổng quan 31](#_Toc137496723)

[Biểu đồ 2.2 Biểu đồ hoạt động đăng ký tài khoản 32](#_Toc137496724)

[Biểu đồ 2.3 Biểu đồ hoạt động đăng nhập 33](#_Toc137496725)

[Biểu đồ 2.4 Biểu đồ hoạt động thêm bài hát 33](#_Toc137496726)

[Biểu đồ 2.5 Biểu đồ hoạt động thêm album 34](#_Toc137496727)

[Biểu đồ 2.6 Biểu đồ hoạt động thêm chủ đề 34](#_Toc137496728)

[Biểu đồ 2.7 Biểu đồ hoạt động tìm kiếm 35](#_Toc137496729)

[Biểu đồ 2.8 Biểu đồ hoạt động sửa thông tin tài khoản 35](#_Toc137496730)

[Biểu đồ 2.9 Biểu đồ tuần tự đăng nhập 36](#_Toc137496731)

[Biểu đồ 2.10 Biểu đồ tuần tự thêm bài hát 37](#_Toc137496732)

[Biểu đồ 2.11 Biểu đồ tuần tự sửa thông tin người dùng 38](#_Toc137496733)

# 

# 

# 

# 

# LỜI MỞ ĐẦU

Lịch sử phát triển công nghệ thông tin đã chỉ ra một vấn đề là phần cứng càng ngày càng tốt lên và có xu hướng rẻ đi, còn phần mềm càng ngày càng “tệ đi” và có xu hướng đắt lên. Vì sao lại có “nghịch lý” này? Câu trả lời rất đơn giản là vì con người luôn có nhiều “đòi hỏi”. Đòi hỏi về phần cứng là một điều rõ ràng, nhưng với phần mềm thì sao? Thực ra con người cần nhiều và rất nhiều từ phần mềm, cần sự phục vụ của phần mềm trong mọi lĩnh vực, cần sự nhanh chóng và hiệu quả tính toán. Việc “tệ đi” không phải theo chiều hướng xấu đi mà do mong đợi của con người cao hơn, chính vì thế chi phí để xây dựng phần mềm càng trở nên đắt hơn.

Khi tiến hành một nghiên cứu, chúng ta đặt ra những câu hỏi, tìm kiếm những thông tin mới, và mong muốn tìm ra những kết luận có giá trị. Nghiên cứu không chỉ giúp chúng ta hiểu rõ hơn về một chủ đề cụ thể, mà còn giúp đưa ra những giải pháp hiệu quả cho những vấn đề được đặt ra. Qua quá trình nghiên cứu, chúng ta có thể khám phá ra những sự thật mới, đưa ra những suy luận và kết luận có tính khả thi và đáng tin cậy.

Xuất phát từ thực tiễn đó, từ tầm quan trọng, cộng với niềm đam mê phát triển lĩnh vực khoa học công nghệ phần mềm, nhóm 9 chúng tôi đã tổng hợp những tài liệu cần thiết cho học phần thực tập cơ sở ngành, nhằm đáp ứng cho công tác giảng dạy và học tập tại nhà trường và các bạn đọc yêu thích lĩnh vực phần mềm. Cấu trúc của tài liệu được xây dựng với 2 chương lớn, gồm 5 phần nhỏ, cách thức xây dựng dựa trên hướng tiếp cận phát triển phần mềm.

**CẤU TRÚC CỦA TÀI LIỆU TỔNG HỢP**

Tài liệu được xây dựng với 2 chương, cách thức xây dựng dựa trên hướng tiếp cận và phát triển.

Chương 1 – Tổng quan về nội dung nghiên cứu. Nội dung của chương trình bày về flutter, ngôn ngữ dart và visual code. Chúng ta sẽ được tìm hiểu kĩ hơn về 3 mục này trong những mục nhỏ.

Mục 1.1: Giới thiệu về flutter............................................................................ xx

Mục 1.2: Giới thiệu ngôn ngữ dart.................................................................... xx

Mục 1.3: Giới thiệu về phần mềm Visual Studio Code..................................... xx

Chương 2 – Giới thiệu về cơ sở dữ liệu mySQL. Đó là các vấn đề liên quan tới cơ sở dữ liệu cần dùng khi sử dụng cũng như mySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

**HƯỚNG TIẾP CẬN CỦA TÀI LIỆU**

Mặc dù là tài liệu và trình bày về cách thức sử dụng phần mềm đơn thuần, hướng tiếp cận để xây dựng nội dung tập trung vào thực hành. Phương pháp tiếp cận đòi hỏi trong mọi vấn đề từ khái niệm ban đầu đến kỹ thuật sau cùng đều phải được chỉ ra từng bước, mỗi bước được mô tả và có thể có những minh họa làm sáng tỏ. Phương pháp tiếp cận thực hành giúp người đọc nhanh chóng n hiểu được vấn đề hơn, tuy vậy để xây dựng được phương pháp này là công việc khá khó khăn cho nhóm tác giả. Vẫn trên quan điểm hiểu nhanh vấn đề và hướng tiếp cận này, giáo trình cấu trúc sau mỗi chương là một danh sách các câu hỏi lý thuyết và bài tập thực hành. Tài liệu được xây dựng là môn thực tập cơ sở ngành, với cách thức trình bày và xây dựng nội dung của nhập môn nên giáo trình sẽ cung cấp nhiều khái niệm (từ đơn giản đến phức tạp) trong môn học. Bên cạnh đó là các quy trình, các phương pháp, các vấn đề kỹ thuật liên quan tới việc tạo dựng và phát triển chương trình. Do đó, bất kể ai cũng có thể đọc được tài liệu này, nếu có sự hiểu biết căn bản về lĩnh vực công nghệ thông tin và lĩnh vực xây dựng sản phẩm phần mềm.

# 

# 

# LỜI CẢM ƠN

Chúng em xin được gửi lời cảm ơn sâu sắc đến quý thầy/cô và toàn bộ thành viên trong nhóm nghiên cứu đã dành thời gian, nỗ lực và tâm huyết để thực hiện tài liệu nghiên cứu này. Công sức của các thành viên trong nhóm cũng đã mang lại những kết quả quan trọng và giá trị cho cộng đồng khoa học cũng như xã hội.

Cảm ơn những nỗ lực tạo điều kiện không ngừng của quý thầy/cô và các thành viên trong nhóm để tìm kiếm các giải pháp và kết quả chính xác nhất cho các câu hỏi nghiên cứu. Cảm ơn các thành viên trong nhóm đã thể hiện tinh thần trách nhiệm và sự chuyên nghiệp cao độ trong suốt quá trình nghiên cứu, từ việc thu thập dữ liệu đến phân tích và đánh giá kết quả. Trong quá trình tìm hiểu và thực hiện báo cáo nhóm chúng em không thể tránh khỏi những thiếu hụt và sai sót , vì vậy nhóm rất mong được nghe sự nhận xét và ý kiến đánh giá của thầy cô dành cho nhóm.

Lời cuối cùng chúng em xin chúc thầy cô luôn dồi dào sức khỏe, hạnh phúc và thành công trong cuộc sống!

Chúng em xin chân thành cảm ơn.

**Nhóm tác giả**

Nhóm 9

# CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## **1.1. Giới thiệu về Flutter**

### **1.1.1. Flutter là gì?**

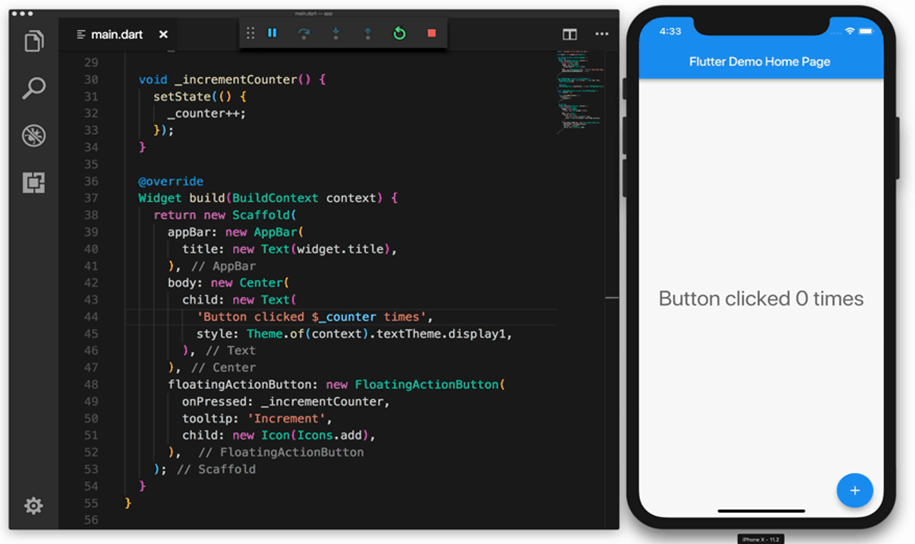
* Flutter là mobile UI framework của Google để tạo ra các giao diện chất lượng cao trên iOS và Android trong khoảng thời gian ngắn. Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức. Flutter còn là một SDK ( Software Development Kit) phát triển ứng dụng di động nguồn mở được tạo ra bởi Google. Nó được sử dụng để phát triển ứng ứng dụng cho Android và IOS, cũng là phương thức chính để tạo ứng dụng cho Google Fuchsia.
* Flutter hoàn toàn miễn phí và cũng là mã nguồn mở.
* Trước khi nói về điểm cộng và điểm trừ của Flutter, thì cũng giống như bất kỳ sản phẩm phần mềm nào khác, Flutter vẫn luôn được cải thiện từng ngày và cộng đồng của nó hiện đang phát triển rất nhanh. Gần đây vào tháng 3 năm 2021, Flutter đã công bố bản cập nhật ổn định lớn nhất của mình, Flutter 2.0, bao gồm tính ổn định hơn, cải tiến tính năng và hơn thế nữa. Bạn có thể thấy rõ tốc độ phát triển của Flutter trong hình bên trên, điều này khiến Flutter trở thành framework đa nền tảng phổ biến thứ 2 vào năm 2020 sau React Native.

### **1.1.2. Tại sao lại là Flutter?**

* Nếu bạn đang tìm kiếm các phương pháp thay thế để phát triển ứng dụng Android, bạn nên cân nhắc thử Flutter của Google, một framework dựa trên ngôn ngữ lập trình Dart.
* Các ứng dụng được xây dựng với Flutter hầu như không thể phân biệt với những ứng dụng được xây dựng bằng cách sử dụng Android SDK, cả về giao diện và hiệu suất. Hơn nữa, với những tinh chỉnh nhỏ, chúng có thể chạy trên thiết bị iOS.
* Chạy ở 60 fps, giao diện người dùng được tạo ra với Flutter thực thi tốt hơn nhiều so với những ứng dụng được tạo ra với các framework phát triển đa nền tảng khác chẳng hạn như React Native và Ionic. Một số lí do khiến bạn có thể hứng thú với Flutter:
* Flutter sử dụng Dart, một ngôn ngữ nhanh, hướng đối tượng với nhiều tính năng hữu ích như mixin, generic, isolate, và static type.
* Flutter có các thành phần UI của riêng nó, cùng với một cơ chế để kết xuất chúng trên nền tảng Android và iOS. Hầu hết các thành phần giao diện người dùng, đều sẵn dùng, phù hợp với các nguyên tắc của Material Design.
* Các ứng dụng Flutter có thể được phát triển bằng cách 4 sử dụng IntelliJ IDEA, một IDE rất giống với Android Studio.

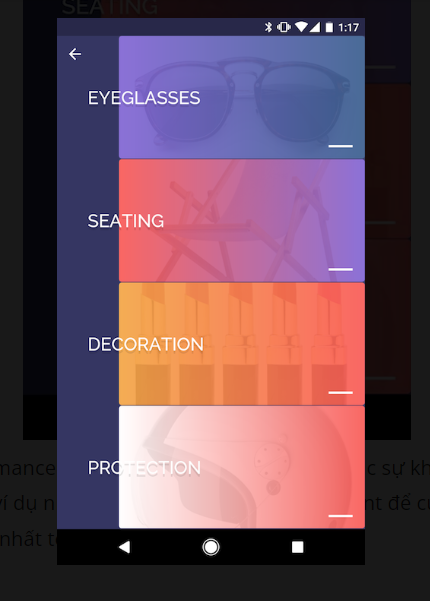
### **1.1.3. Đặc điểm nổi bật**

* Fast Development: Tính năng Hot Reload hoạt động trong milliseconds để hiện thị giao diện tới bạn. Sử dụng tập hợp các widget có thể customizable để xây dựng giao diện trong vài phút. Ngoài ra Hot Reload còn giúp bạn thêm các tính năng, fix bug tiết kiệm thời gian hơn mà không cần phải thông qua máy ảo, máy android hoặc iOS



Hình 1.1 Hình ảnh mô tả Flutter chạy trên máy ảo

* Expressive and Flexible UI: Có rất nhiều các thành phần để xây dựng giao diện của Flutter vô cùng đẹp mắt theo phong cách Material Design và Cupertino, hỗ trợ nhiều các APIs chuyển động, smooth scrolling...



Hình 1.2:Hình ảnh mô tả UI Flutter

* Native Performance: Các widget của Flutter kết hợp các sự khác biệt của các nền tảng ví dụ như scrolling, navigation, icons, font để cung cấp một hiệu năng tốt nhất tới iOS và Android.

**Tóm lại:**

Ưu và nhược điểm của Flutter: Bây giờ chúng ta hãy xem những ưu và nhược điểm quan trọng nhất của Flutter để giúp quyết định của bạn trở nên chính xác hơn khi theo học Flutter.

* Ưu điểm:

+ Mạnh về hiệu ứng, hiệu suất ứng dụng rất cao.

+ Giao tiếp gần như trực tiếp với hệ thống

+ Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại (tương tự JS, Python, Java), compiler linh động khi dùng AOT (cho sản phẩm cuối) và JIT (cho quá trình phát triển với hot reload)

+ Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển. Các bộ đo lường chỉ số hiệu suất được hỗ trợ sẵn giúp lập trình viên kiểm soát tốt hiệu suất của ứng dụng.

+ Có thể dùng để xây dựng các nền tảng gắn vào ứng dụng native để tăng hiệu suất.

Bên cạnh những ưu điểm như vậy thì Flutter còn có những nhược điểm:

* Nhược điểm:

+ Hầu hết bộ render UI đã được viết lại, không liên quan đến UI sẵn có trong UI Framework native nên memory sẽ dùng khá nhiều. Bên cạnh đó, UI không đi cùng với OS mà được phát triển riêng.

+ Thường xuyên bổ sung các kiến thức về ngôn ngữ DART: Có rất ít lập trình viên hiểu biết rõ về Dart. Có nhiều trường hợp sau khi học xong sẽ dính liền luôn với Dart trong mảng phát triển các ứng dụng trên mobile, chứ không thể linh động như Python, JS có thể qua lại giữa AI, back,..

+ Mô hình dữ liệu rất mới: Nếu đã quen với Redux thì phải sẽ mất kha khá thời gian để học hỏi thêm về mô hình dữ liệu của Flutter mặc dù nó không quá khó.

### **1.1.4. Kết luận**

Nếu bạn là một cá nhân hay một doanh nghiệp nhỏ và đang muốn đẩy nhanh tiến độ dự án trong thời gian ngắn, thì Flutter là một sự lựa chọn hoàn toàn hợp lý. Tuy nhiên, nếu dự án của bạn phức tạp hơn và quy mô lớn hơn hoặc bạn đang có kế hoạch thực hiện nó lâu dài thì hãy chọn React Native thay vì Flutter. Vì sau này mọi thứ sẽ trở rất khó khăn nếu như bạn chọn Flutter.

## **1.2. Tìm hiểu về ngôn ngữ Dart**

* Dart là ngôn ngữ lập trình mới, được phát triển bởi Google, hiện đã được chấp thuận bởi tổ chức Ecma. Dart được sử dụng để xây dựng các loại ứng dụng: web, server, di động (IOS và Android với công cụ Flutter).
* Dart là ngôn ngữ hướng đối tượng với các cú phạm theo dạng C-style, nó hỗ trợ các khái niệm lập trình như interface, class.

### **1.2.1. Dart là gì?**

* Dart là một ngôn ngữ lập trình hiện đại có mục đích chung, cấp cao, được phát triển ban đầu bởi Google. Đây là ngôn ngữ lập trình mới xuất hiện vào năm 2011, nhưng phiên bản ổn định của nó đã được phát hành vào tháng 6 năm 2017. Dart không quá phổ biến vào thời điểm đó, nhưng nó đã trở nên phổ biến khi được sử dụng bởi Flutter.
* Dart là một ngôn ngữ lập trình động, dựa trên lớp, hướng đối tượng với phạm vi đóng và từ vựng. Về mặt cú pháp, nó khá giống với Java, C và JavaScript. Nếu bạn biết bất kỳ ngôn ngữ lập trình nào trong số này, bạn có thể dễ dàng học ngôn ngữ lập trình Dart.
* Dart là một ngôn ngữ lập trình mã nguồn mở được sử dụng rộng rãi để phát triển ứng dụng di động, ứng dụng web hiện đại, ứng dụng máy tính để bàn và Internet of Things (IoT) bằng cách sử dụng khung Flutter. Nó cũng hỗ trợ một số khái niệm nâng cao như giao diện, mixin, lớp trừu tượng, tổng thể trường và giao diện kiểu. Nó là một ngôn ngữ biên dịch và hỗ trợ hai loại kỹ thuật biên dịch.
  + AOT (Ahead of Time) – Nó chuyển đổi mã Dart sang mã JavaScript được tối ưu hóa với sự trợ giúp của trình biên dịch dar2js và chạy trên tất cả các trình duyệt web hiện đại. Nó biên dịch mã tại thời điểm xây dựng.
  + JOT (Just-In-Time) – Nó chuyển đổi mã byte trong mã máy (mã gốc), nhưng chỉ mã cần thiết.

### **1.2.2. Lịch sử**

Dart được tiết lộ lần đầu tiên trong hội nghị GOTO vào tháng 10 – 12 tháng 10 năm 2011 tại Aarhus, Đan Mạch. Ban đầu nó được thiết kế bởi Lars và Kespa và được phát triển bởi Google. Phiên bản 1.0 đầu tiên của Dart được phát hành vào ngày 14 tháng 11 năm 2013, nhằm mục đích thay thế JavaScript. Vào tháng 7 năm 2014, ấn bản đầu tiên của ngôn ngữ Dart đã được Ecma International thông qua tại Đại hội đồng lần thứ 107 của tổ chức này. Phiên bản đầu tiên đã bị chỉ trích do sự cố trên web và kế hoạch này đã bị loại bỏ vào năm 2015 với bản phát hành 1.9 của Dart. Phiên bản thứ hai của Dart 2.0 được phát hành vào tháng 8, bao gồm một hệ thống âm thanh. Phiên bản gần đây Dart 2.7 được bổ sung thêm phương thức mở rộng, cho phép chúng tôi thêm bất kỳ loại chức năng nào.

### **1.2.3. Tại sao lại sử dụng Dart?**

Chúng tôi xác định các đặc điểm của Dart ở điểm sau:

* Dart là một ngôn ngữ độc lập với nền tảng và hỗ trợ tất cả các hệ điều hành như Windows, Mac, Linux, v.v.
* Nó là một ngôn ngữ mã nguồn mở, có nghĩa là nó có sẵn miễn phí cho tất cả mọi người. Nó đi kèm với giấy phép BSD và được công nhận bởi tiêu chuẩn ECMA.
* Nó là một ngôn ngữ lập trình hướng đối tượng và hỗ trợ tất cả các tính năng của oops như kế thừa, giao diện và các tính năng kiểu tùy chọn.
* Dart rất hữu ích trong việc xây dựng các ứng dụng thời gian thực vì tính ổn định của nó.
* Dart đi kèm với trình biên dịch dar2js để truyền mã Dart thành mã JavaScript chạy trên tất cả các trình duyệt web hiện đại.
* Máy ảo Dart độc lập cho phép mã Dart chạy trong môi trường giao diện dòng lệnh.

### **1.2.4. Những điểm chính cần nhớ**

Trước khi học **Dart**, chúng ta nên ghi nhớ những khái niệm này. Những khái niệm này được đưa ra dưới đây:

* Mọi thứ trong Dart được coi như một đối tượng bao gồm, số, Boolean, hàm, v.v. giống như Python. Tất cả các đối tượng kế thừa từ lớp Đối tượng.
* Công cụ Dart có thể báo cáo hai loại sự cố trong khi mã hóa, cảnh báo và lỗi. Cảnh báo là dấu hiệu cho thấy mã của bạn có thể có một số vấn đề, nhưng nó không làm gián đoạn quá trình thực thi của mã, ngược lại lỗi có thể ngăn chặn việc thực thi mã.
* Dart hỗ trợ gõ âm thanh. Chúng ta sẽ tìm hiểu về điều này trong phần hướng dẫn tiếp theo.
* Dart hỗ trợ các kiểu chung chung, như List <int> (danh sách các số nguyên) hoặc List <dynamic> (danh sách các đối tượng thuộc bất kỳ kiểu nào).

### **1.2.5. Tính năng của Dart**

Dart là một ngôn ngữ lập trình hướng đối tượng, mã nguồn mở, chứa nhiều tính năng hữu ích. Đây là ngôn ngữ lập trình mới và hỗ trợ một loạt các tiện ích lập trình như giao diện, bộ sưu tập, lớp, kiểu gõ động và tùy chọn. Nó được phát triển cho máy chủ cũng như trình duyệt. Dưới đây là danh sách các tính năng quan trọng của Dart:

* Mã nguồn mở
* Nền tảng độc lập
* Hướng đối tượng
* Dồng nhất
* Thư viện mở rộng
* Dễ học
* Biên dịch linh hoạt
* Nhập an toàn
* Các đối tượng
* Hỗ trợ trình duyệt
* Cộng đồng(community)

**Mã nguồn mở**

Dart là một ngôn ngữ lập trình mã nguồn mở, có nghĩa là nó có sẵn miễn phí. Nó được phát triển bởi Google, được phê duyệt bởi tiêu chuẩn ECMA và đi kèm với giấy phép BSD.

**Nền tảng độc lập**

Dart hỗ trợ tất cả các hệ điều hành chính như Windows, Linux, Macintosh, v.v. Dart có Máy ảo riêng được gọi là Dart VM, cho phép chúng tôi chạy mã Dart trong mọi hệ điều hành.

**Hướng đối tượng**

Dart là một ngôn ngữ lập trình hướng đối tượng và hỗ trợ tất cả các khái niệm oops như lớp, kế thừa, giao diện và các tính năng gõ tùy chọn. Nó cũng hỗ trợ các khái niệm nâng cao như mixin, abstract, các lớp, hệ thống kiểu chung được sửa đổi và mạnh mẽ.

**Đồng nhất**

Dart là một ngôn ngữ lập trình không đồng bộ, có nghĩa là nó hỗ trợ đa luồng sử dụng Isolates. Các vùng cách ly là các thực thể độc lập có liên quan đến các luồng nhưng không chia sẻ bộ nhớ và thiết lập giao tiếp giữa các quá trình bằng cách truyền thông điệp. Thông điệp nên được nối tiếp nhau để tạo hiệu quả truyền thông. Việc tuần tự hóa thông báo được thực hiện bằng cách sử dụng một ảnh chụp nhanh được tạo ra bởi đối tượng đã cho và sau đó truyền đến một vùng cách ly khác để giải mã.

**Thư viện mở rộng**

Dart bao gồm nhiều thư viện tích hợp hữu ích bao gồm SDK (Bộ phát triển phần mềm), lõi, toán học, không đồng bộ, toán học, chuyển đổi, html, IO, v.v. Nó cũng cung cấp cơ sở để tổ chức mã Dart thành các thư viện với không gian tên riêng. Nó có thể sử dụng lại bằng câu lệnh nhập.

**Dễ học**

Như chúng ta đã thảo luận trong phần trước, học Dart không phải là nhiệm vụ của Hercules vì ​​chúng ta biết rằng cú pháp của Dart tương tự như Java, C #, JavaScript, kotlin, v.v. nếu bạn biết bất kỳ ngôn ngữ nào trong số này thì bạn có thể học Dart dễ dàng.

**Biên dịch linh hoạt**

Dart cung cấp sự linh hoạt để biên dịch mã và nhanh chóng. Nó hỗ trợ hai loại quy trình biên dịch, AOT (Ahead of Time) và JIT (Just-in-Time). Mã Dart được truyền bằng ngôn ngữ khác có thể chạy trong các nhà sản xuất web hiện đại.

**Nhập An toàn**

Dart là ngôn ngữ an toàn kiểu, có nghĩa là nó sử dụng cả kiểm tra kiểu tĩnh và kiểm tra thời gian chạy để xác nhận rằng giá trị của một biến luôn khớp với kiểu tĩnh của biến, đôi khi nó được gọi là kiểu gõ âm thanh.

Mặc dù loại là bắt buộc, nhưng chú thích loại là tùy chọn vì loại nhiễu. Điều này làm cho mã dễ đọc hơn. Ưu điểm khác của ngôn ngữ an toàn kiểu chữ là khi chúng ta thay đổi phần mã, hệ thống sẽ cảnh báo chúng ta về sửa đổi mà chúng ta đã sửa trước đó.

**Các đối tượng**

Dart coi mọi thứ như một đồ vật. Giá trị gán cho biến là một đối tượng. Các hàm, số và chuỗi cũng là một đối tượng trong Dart. Tất cả các đối tượng kế thừa từ lớp Đối tượng.

**Hỗ trợ trình duyệt**

Dart hỗ trợ tất cả các trình duyệt web hiện đại. Nó đi kèm với trình biên dịch dart2js để chuyển đổi mã Dart thành mã JavaScript được tối ưu hóa phù hợp với tất cả các loại trình duyệt web.

**Cộng đồng**

Dart có một cộng đồng lớn trên toàn thế giới. Vì vậy, nếu bạn gặp vấn đề trong khi viết mã thì rất dễ dàng tìm được trợ giúp. Nhóm các nhà phát triển chuyên dụng đang làm việc để nâng cao chức năng của nó.

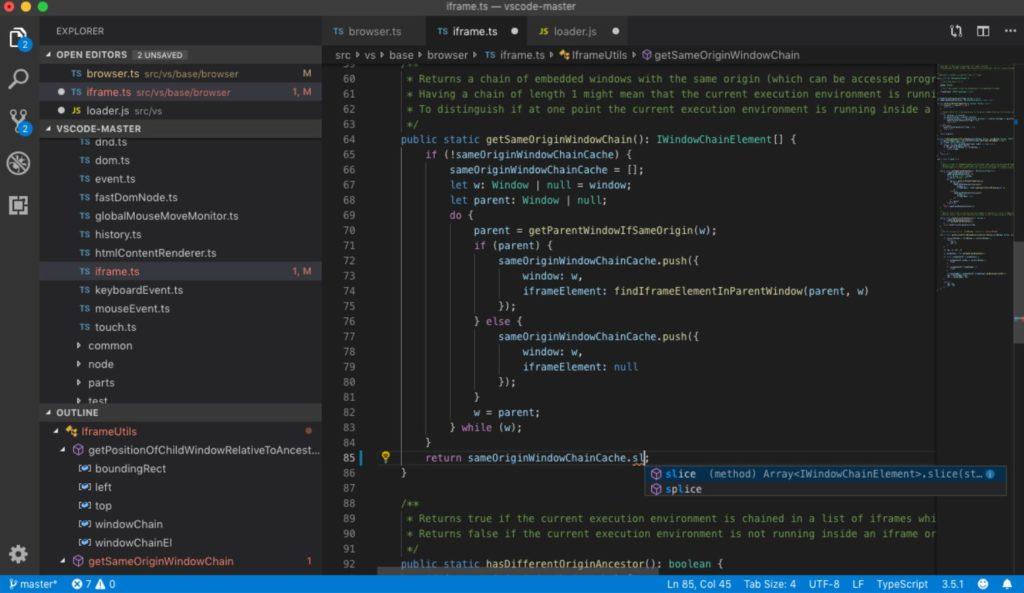
Ở đây chúng ta đã thảo luận về các tính năng thiết yếu của ngôn ngữ Dart. Chúng tôi sẽ bổ sung thêm các khái niệm về ngôn ngữ Dart trong các bài hướng dẫn sắp tới.

## **1.3. Visual Studio Code**

### **1.3.1. Visual studio code là gì?**

Đối với nhiều lập trình viên, trình soạn thảo code là một công cụ không thể thiếu trong quá trình xây dựng các ứng dụng web. Ưu điểm của chúng là dung lượng khá nhẹ, ít tốn RAM và được tích hợp các tính năng gần giống với IDE. Trong đó, Visual Studio Code là cái tên quen thuộc nhất với dân IT hiện nay.

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ trên 3 nền tảng là Windows, Linux và macOS được phát triển bởi Microsoft. Nó hỗ trợ cho JavaScript, Node.js và TypeScript, cũng như cung cấp một hệ sinh thái mở rộng vô cùng phong phú cho nhiều ngôn ngữ lập trình khác.



Hình 1.3 Giao diện Visual Studio Code

Visual Studio Code là trình soạn thảo mã nguồn mở hoạt động mạnh mẽ trên nhiều nền tảng. Có thể nói, VS Code là sự kết hợp giữa một trình soạn thảo mã nguồn với những công cụ phát triển mạnh mẽ như Git, Debug hay Syntax Highlighter.

### **1.3.3. Lý do bạn nên sử dụng Visual Studio Code**

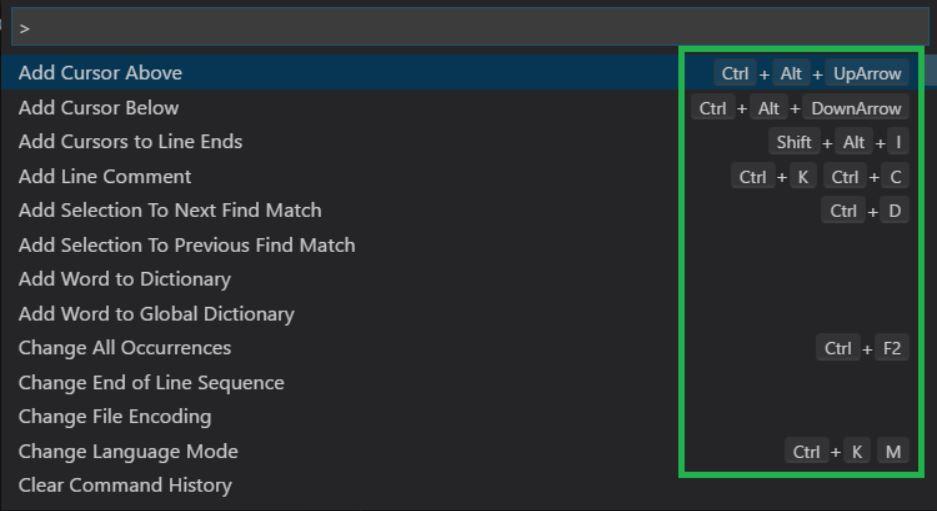
#### **1.3.3.1. Hoạt động đa nền tảng**

VS Code được thiết kế để có thể chạy tốt trên macOS, Linux và Windows. Vì thế, bạn có thể thoải mái sử dụng trình soạn thảo này mà không cần lo lắng về vấn đề lựa chọn nền tảng tương thích.

#### **1.3.3.2. Đơn giản trong việc chỉnh sửa, xây dựng và gỡ lỗi**

Như đã nói ở trên, tốc độ là một trong những ưu thế vượt trội của VS Code. Nó có thể hỗ trợ rất nhiều các loại ngôn ngữ lập trình khác nhau như Java, Python, C++, C#, Golang hay PHP. Điều này cho phép bạn có thể ứng dụng rất nhiều tính năng thú vị như đánh dấu cú pháp, tự động thụt lề, đối sánh dấu ngoặc,… cho công việc của mình.

Hệ thống phím tắt sinh động, trực quan giúp thao tác trở nên nhanh chóng hơn, từ đó năng suất và hiệu quả lập trình cũng được tăng lên. Ngoài ra, tính năng nhắc mã IntelliSense chuyên nghiệp trên VS Code sẽ tự động phát hiện đoạn mã không đầy đủ và thông báo cho lập trình viên. Hoặc nó có thể tự động bổ sung cú pháp còn thiếu khi bạn quên không khai báo biến.



Hình 1.4 Các phím tắt trên VS Code

Gỡ lỗi cũng là một tính năng nổi bật của VS Code. Nó giúp bạn xem mã nguồn, kiểm tra các các biến, Call Stack và thực thi lệnh trong bảng điều khiển. Bên cạnh đó, trình soạn thảo này còn hỗ trợ Git phục vụ cho việc kiểm soát nguồn mà không đòi hỏi phải rời khỏi trình chỉnh sửa.

#### **1.3.3.3. Sự hỗ trợ của cộng đồng GitHub**

Visual Studio Code là một dự án mã nguồn mở nên bạn hoàn toàn có thể tham gia đóng góp và tìm kiếm sự giúp đỡ từ cộng đồng GitHub.

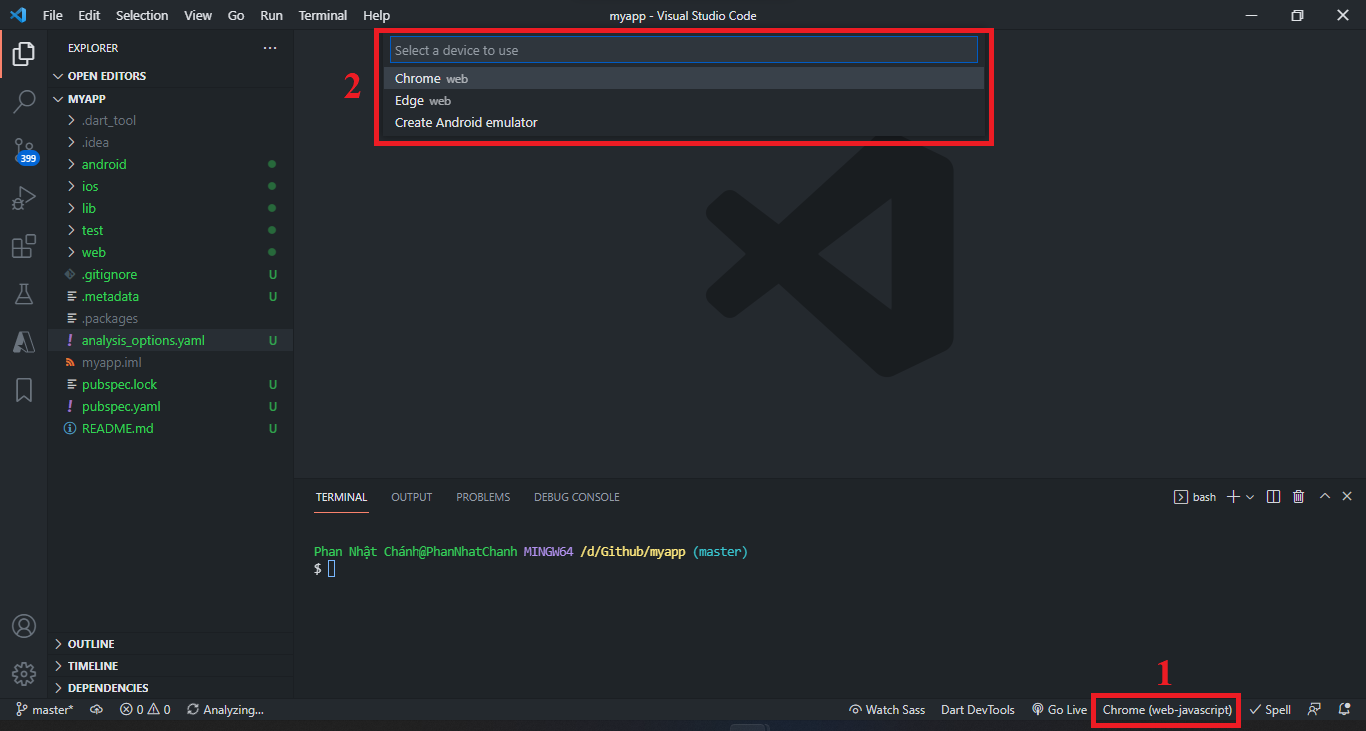
#### **1.3.3.4. Kiến trúc cực mạnh mẽ và có thể mở rộng**

VS Code được xây dựng dựa trên Electron, kết hợp các công nghệ web như Node.js và JavaScript với tốc độ và sự linh hoạt của các ứng dụng gốc. Ngoài ra, kiến trúc dịch vụ công cụ của VS Code giúp nó tích hợp được với các công nghệ hỗ trợ Visual Studio như Roslyn cho .NET, TypeScript, công cụ gỡ lỗi Visual Studio…

Nhờ những ưu điểm nổi bật của mình, VS Code trở thành một trong những sự lựa chọn hàng đầu của rất nhiều lập trình viên, giúp đảm bảo tiến độ và đạt hiệu quả công việc cao. Nếu bạn đang tìm kiếm câu trả lời cho thắc mắc **Visual Studio Code là g**ì và vì sao nên dùng trình soạn thảo mã nguồn này thì đây chính là bài viết dành cho bạn.

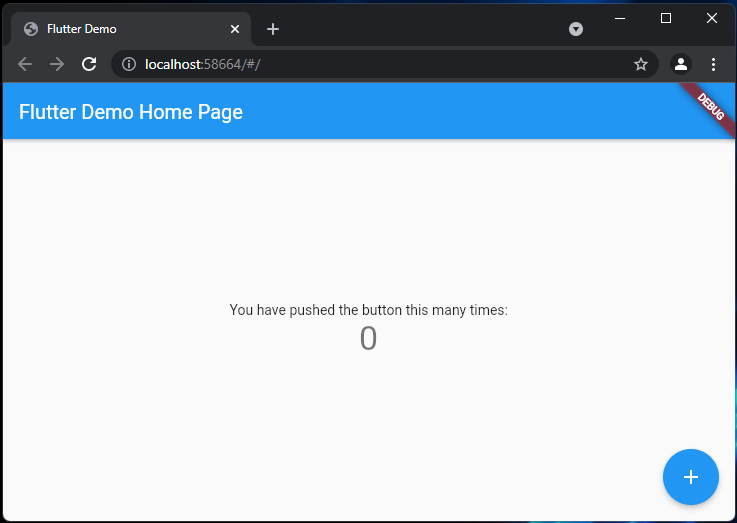
#### **1.3.3.5. Khởi chạy ứng dụng Flutter**

* **Cách 1: Sử dụng VS Code** Mở VS Code và vào **File -> Open Folder...** để mở thư mục chứa dự án của bạn. Sau đó, hãy nhấp vào tên thiết bị ở dưới cùng bên phải và chọn trình duyệt để khởi động (chọn Chrome) như hình dưới đây:



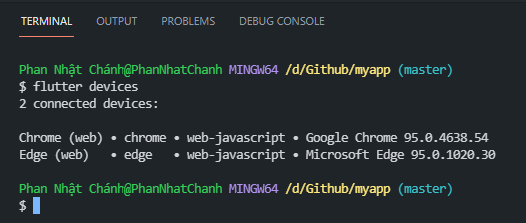
Hình 1.5 Hình minh họa chọn trình duyệt để khởi động

Cuối cùng, hãy khởi động ứng dụng của bạn bằng một phím tắt F5 và ứng dụng đã được khởi chạy trên thiết bị được chỉ định



Hình 1.6 Hình ảnh giao diện của app

* **Cách 2: Chạy ứng dụng Flutter bằng dòng lệnh** Bạn có thể kiểm tra các thiết bị chạy được ứng dụng flutter bằng cách chạy dòng lệnh **flutter devices**

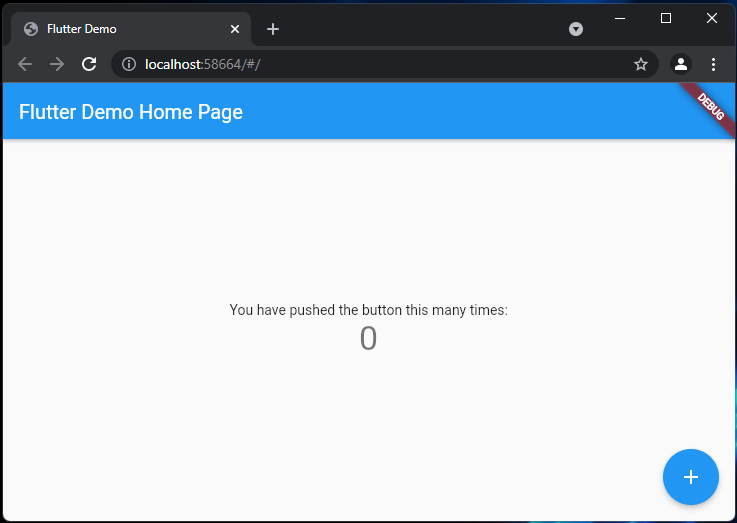


Hình 1.7 Chạy lệnh flutter devices

Tiếp theo, bạn có thể bắt đầu ứng dụng của mình với dòng lệnh **flutter run** (chỉ định trình duyệt để khởi động ứng dụng là Chrome).

**flutter run --device-id chrome**

Ứng dụng đã được khởi chạy trên thiết bị được chỉ định



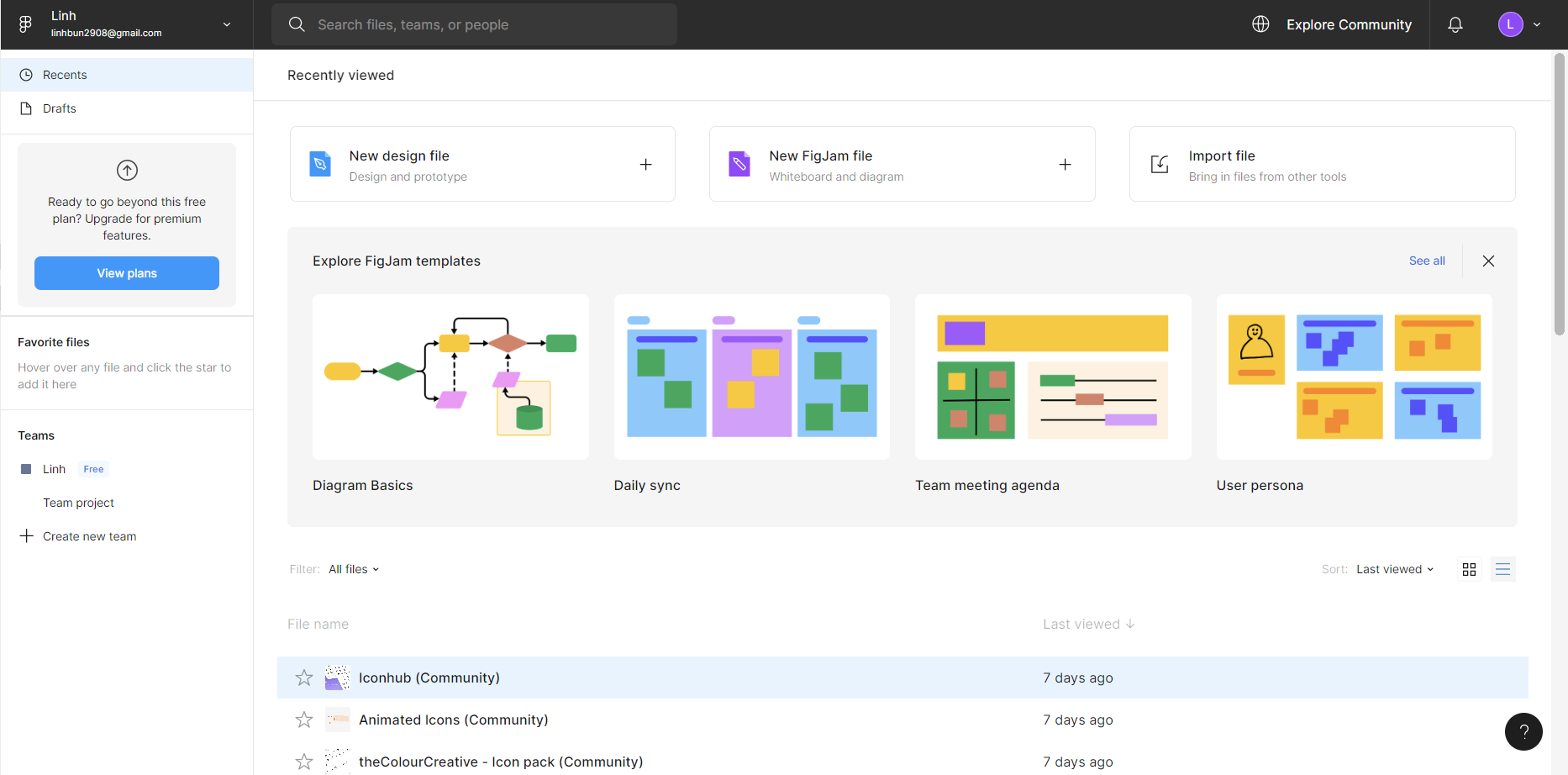
Hình 1.8 Hình ảnh giao diện của app

## **1.4. Figma**

### **1.4.1. Figma là gì?**

Figma là một [phần mềm biên tập đồ họa vector](https://vi.wikipedia.org/w/index.php?title=Ph%E1%BA%A7n_m%E1%BB%81m_bi%C3%AAn_t%E1%BA%ADp_%C4%91%E1%BB%93_h%E1%BB%8Da_vector&action=edit&redlink=1) và [dựng prototype](https://vi.wikipedia.org/w/index.php?title=D%E1%BB%B1ng_nguy%C3%AAn_m%E1%BA%ABu_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1) (nguyên mẫu). Figma chủ yếu hoạt động trên nền web, dù có một số tính năng ngoại tuyến trên các phiên bản ứng dụng desktop cho hệ điều hành M[acOS](https://vi.wikipedia.org/wiki/MacOS) và [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows). Một tiện ích của Figma là Figma Mirror cho [Android](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)) và [iOS](https://vi.wikipedia.org/wiki/IOS) giúp người dùng có thể xem các prototype Figma trên thiết bị di động. Bộ tính năng của Figma tập trung hỗ trợ công việc thiết kế [giao diện người dùng](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_ng%C6%B0%E1%BB%9Di_d%C3%B9ng) và [trải nghiệm người dùng](https://vi.wikipedia.org/wiki/Tr%E1%BA%A3i_nghi%E1%BB%87m_ng%C6%B0%E1%BB%9Di_d%C3%B9ng), chú trọng khả năng cộng tác theo thời gian thực.

Điểm đặc biệt ở Figma là các công cụ được cung cấp miễn phí hoàn toàn và được lưu trữ thông qua hệ thống lưu trữ đám mây. Figma được xem là phiên bản Google Docs dành cho dân thiết kế, cho phép nhiều người cộng tác trực tuyến trong một file. Các [lập trình viên](https://careerbuilder.vn/en/talentcommunity/wiki-career/lap-trinh-vien-la-gi-cong-viec-cua-mot-lap-trinh-vien.35A51E5A.html) có thể cùng chỉnh sửa, thảo luận, theo dõi bản thiết kế ở cùng một thời gian, ở các phía máy chủ khác nhau.



Hình 1.9 Hình ảnh giao diện của trang web

### **1.4.2. Tại sao lại là Figma?**

Ngoài tính năng cung cấp bộ công cụ miễn phí, Figma còn có những ưu điểm nổi bật sau:

**Theo dõi hoạt động trong thời gian thực**

Mỗi người dùng có thể truy cập vào dự án chung trong cùng một thời gian, được thể hiện bằng ảnh avatar trên thanh công cụ. Mọi người có thể biết được các thành viên đang làm gì, đang ở vị trí nào trên trang bằng cách click vào ảnh avatar. Với tính năng này, Figma sẽ giúp các nhóm làm việc từ xa trở nên dễ dàng hơn. Các thành viên có thể theo dõi thao tác của người hướng dẫn, giúp kiểm tra tiến độ công việc và thực hiện điều chỉnh kịp thời nếu có sai sót xảy ra theo thời gian thực.

Figma còn có tiện ích là sẽ gửi thông báo khi file có thay đổi để tất cả các thành viên có thể cập nhật được thông tin mới nhất.

**Khả năng tương thích cao**

Là phần mềm dựa trên nền tảng website nên Figma có khả năng tương thích với nhiều hệ điều hành như Windows, Linux, [macOS](https://vieclamit.careerbuilder.vn/en/advices/macos-monterey-ban-chinh-thuc-va-ios-15-1-se-duoc-phat-hanh-vao-ngay-25-10.35A5F262.html),… Vì thế, đây là một trong những công cụ thiết kế giúp designer có thể truy cập và thao tác dễ dàng.

Figma còn có thêm phiên bản ứng dụng dùng trên điện thoại để người dùng có thể dễ dàng thao tác bất cứ nơi đâu.

**Có nhiều công cụ plugin mạnh mẽ**

Kho plugin cực kỳ đa dạng và hữu ích giúp người dùng có thể khắc phục được những lỗi nhỏ và làm tăng hiệu suất làm việc. Kho plugin tương đối ít lỗi và việc quản lý màu sắc, nội dung, hình ảnh vô cùng dễ dàng.

**Thiết kế nhiều layout trong một sản phẩm**

Figma cũng tích hợp tính năng quản lý nhiều artboard cùng một lúc. Tính năng này giúp các designer có thể sáng tạo nhiều layout có các kích thước khác nhau trên cùng một sản phẩm mà không cần phải tạo thêm một tệp mới.

**Xuất được đa dạng file ảnh cực sắc nét**

Figma thiết kế hình ảnh dưới dạng vector vì thế chất lượng hình ảnh tốt, không bị giảm khi thay đổi kích thước ảnh. Hình ảnh xuất ra được lưu trữ dưới các định dạng khác nhau như: PNG, SVG, JPG hay là PDF nhưng chất lượng hình ảnh vẫn cực sắc nét.

**Hỗ trợ lưu trữ đám mây**

Toàn bộ các thiết kế được lưu trữ trên hệ thống đám mây điện tử để lưu trữ và dễ dàng chỉnh sửa dữ liệu. Trường hợp các thiết kế đang thực hiện nhưng bị ngắt kết nối, hoặc thiết bị lỗi thì các thông tin thiết kế sẽ được lưu trữ trên hệ thống [điện toán đám mây](https://vieclamit.careerbuilder.vn/advices/dien-toan-dam-may-co-the-la-co-the-la-muc-tieu-cong-pha-tiep-theo-cua-alibaba.35A5B44C.html). Các dữ liệu trên Figma sẽ lấy lại được với một vài thao tác đơn giản. Bạn không cần quá lo lắng khi ổ cứng đầy dung lượng hay quên sao lưu thì Figma sẽ tự động thực hiện sao lưu giúp bạn. Người dùng cũng có thể chia thành các mục lưu trữ để dễ dàng quản lý và tránh những rủi ro không mong muốn xảy ra.

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## **2.1. Đặc tả yêu cầu hệ thống**

### **2.1.1. Yêu cầu chức năng**

* User có thể đăng nhập, đăng ký vào ứng dụng đồng thời có các liên kết tài khoản từ số điện thoại, Google, Facebook,…
* Ứng dụng có hiển thị các mục đã tìm kiếm, các top play và my playlist cho người nghe.

### **2.1.2. Yêu cầu phi chức năng**

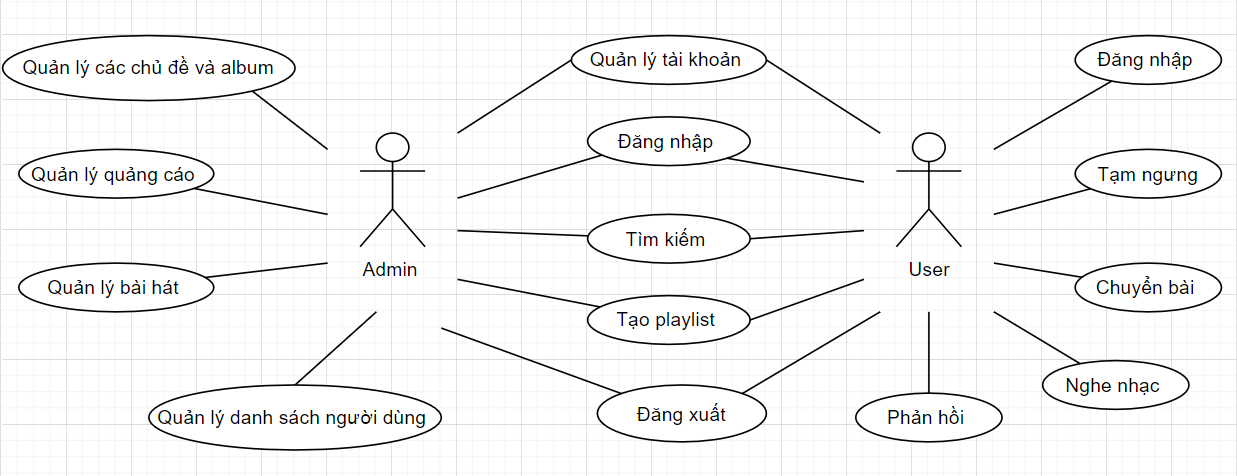
* App hỗ trợ cứu hộ hay trang quản trị hoạt động 24/24h, giao diện(UI) dễ nhìn có thiện cảm với người dùng ngoài ra các vấn đề thao tác của người dùng cần đơn giản dễ thao tác tránh cảm giác gây ức chế khi người dùng thao tác.
* Ổn định, xử lý nhanh, Tốc độ tải trang hay thực hiện công việc không được quá lâu, hiện các thông báo khi kết thúc một thao tác công việc nào đó.
* Trang web có thể tương tích với các trình duyệt khác nhau các thiết bị truy cập khác nhau.
* An toàn, bảo mật:Đảm bảo an toàn thông tin người dùng. Các thao tác nghiệp vụ của người quản trị chỉ thực hiện được khi có quyền.
* Bảo trì nhanh chóng, dễ dàng. Hệ thống đáng tin cậy và đáp ứng đúng với nhu cầu người dùng.
* App có khả năng xử lý số lượng người dùng cần thiết mà không có bất kỳ sự suy giảm nào về hiệu suất.

## **2.2. Phân tích và thiết kế**

### **2.2.1. Biểu đồ Use-case**

Từ các yêu cầu về chức năng, chúng ta xác định 2 Actor chính:

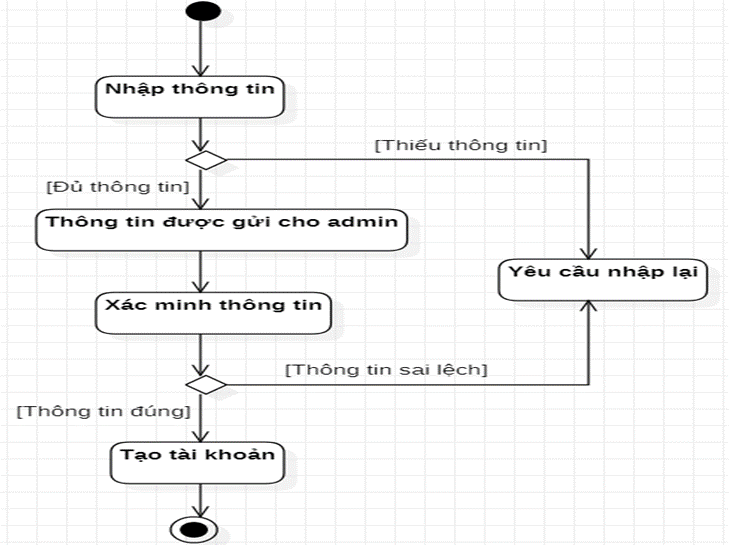
* Quản lý(Administrator): có quyền cao nhất có thể sử dụng mọi chức năng của hệ thống.
* Người dùng(User): Có quyền chỉnh sửa thông tin tài khoản và sử dụng các chức năng cơ bản của ứng dụng như tìm kiếm, nghe, tạo playlist nhạc ưa thích,…



Biểu đồ 2.1 Biểu đồ use case tổng quan

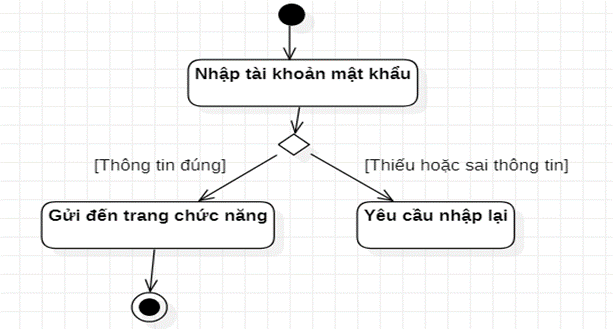
### **2.2.2. Biểu đồ hoạt động**

* Đăng ký tài khoản



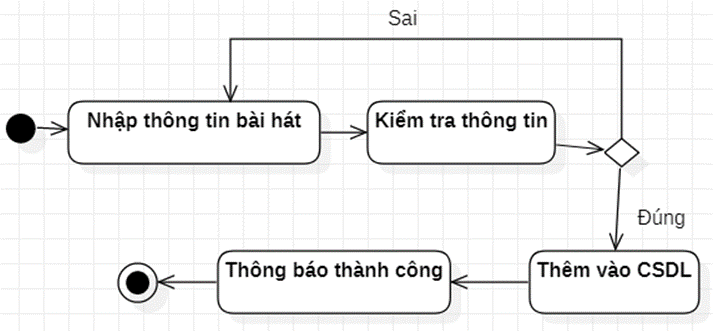
Biểu đồ 2.2 Biểu đồ hoạt động đăng ký tài khoản

* Đăng nhập

**

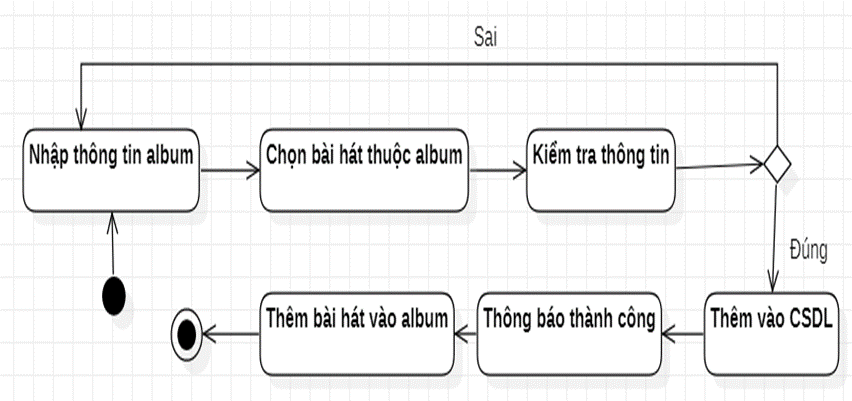
Biểu đồ 2.3 Biểu đồ hoạt động đăng nhập

* Thêm bài hát

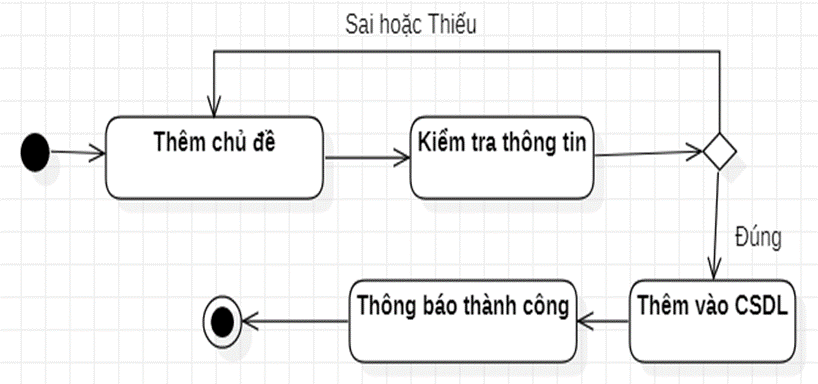
**

Biểu đồ 2.4 Biểu đồ hoạt động thêm bài hát

* Thêm album

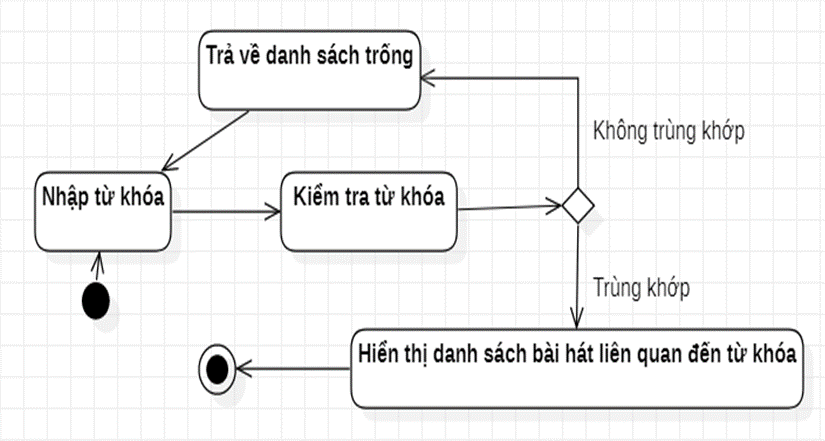
Biểu đồ 2.5 Biểu đồ hoạt động thêm album

* Thêm chủ đề

**

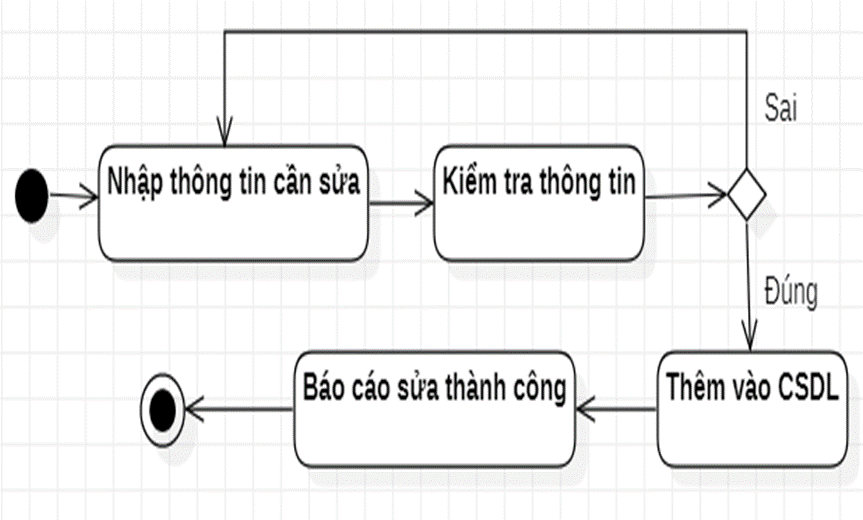
Biểu đồ 2.6 Biểu đồ hoạt động thêm chủ đề

* Tìm kiếm

**

Biểu đồ 2.7 Biểu đồ hoạt động tìm kiếm

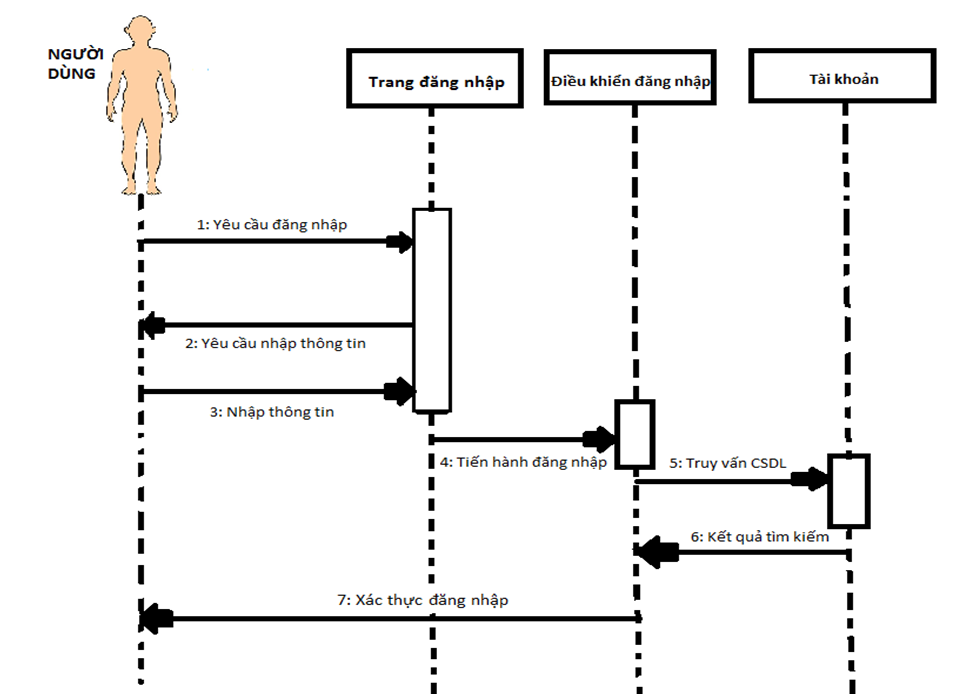
* Sửa thông tin tài khoản

**

Biểu đồ 2.8 Biểu đồ hoạt động sửa thông tin tài khoản

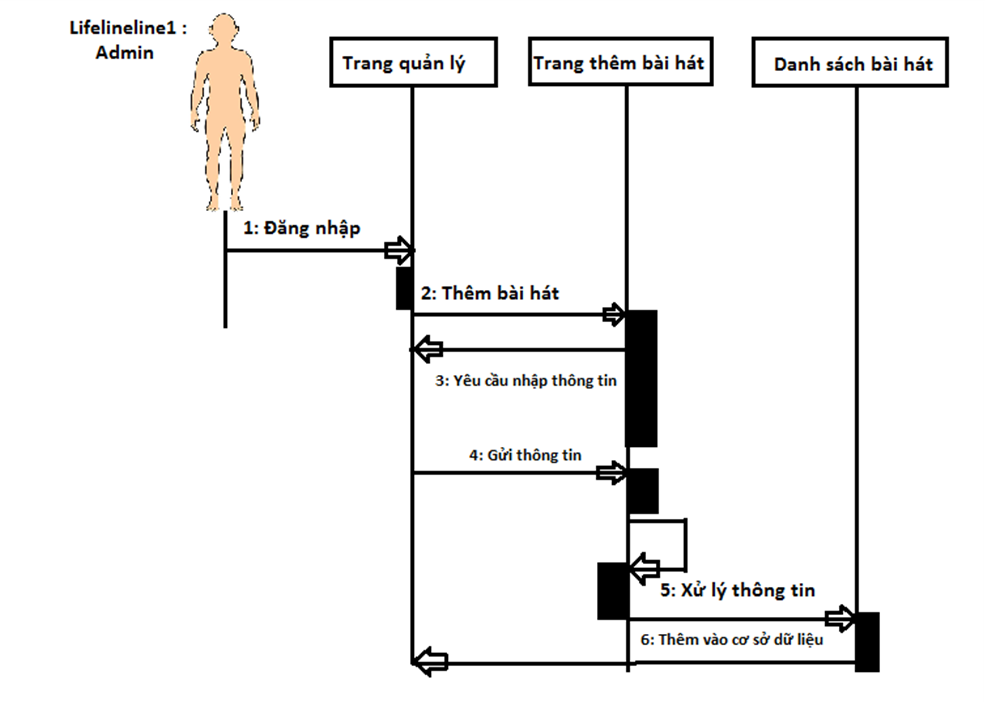
### **2.2.3. Biểu đồ tuần tự**

* Biểu đồ tuần tự đăng nhập

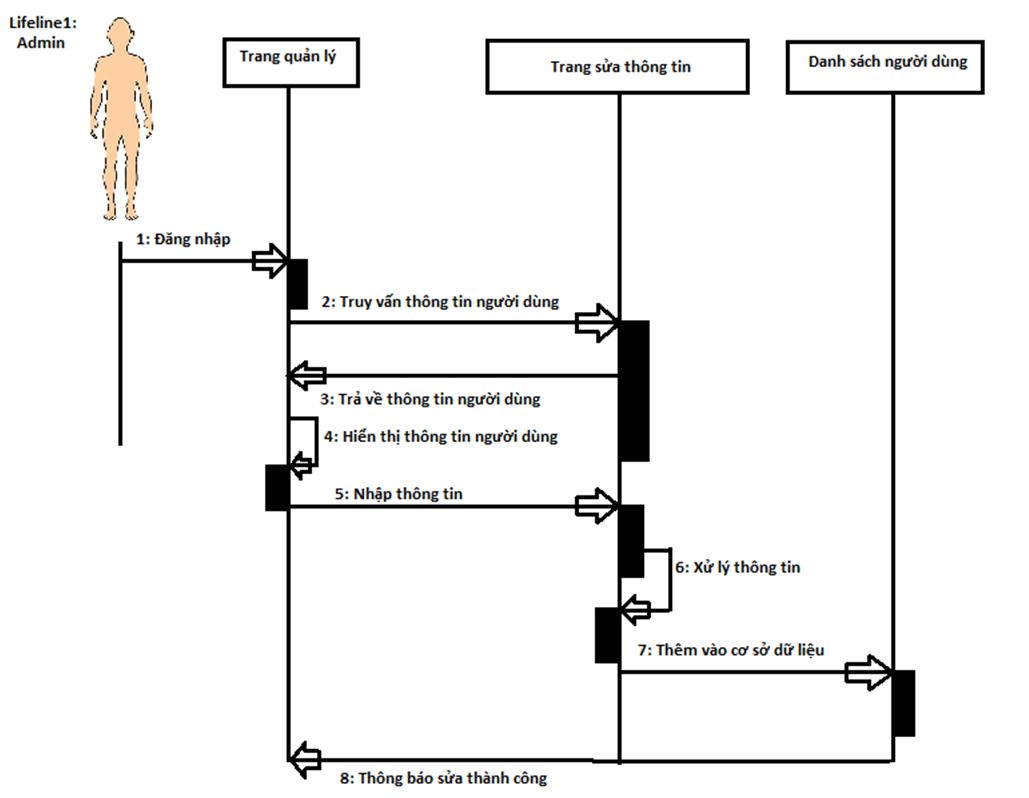
****

Biểu đồ 2.9 Biểu đồ tuần tự đăng nhập

* Biểu đồ tuần tự thêm bài hát

****

Biểu đồ 2.10 Biểu đồ tuần tự thêm bài hát

* Biểu đồ tuần tự ****

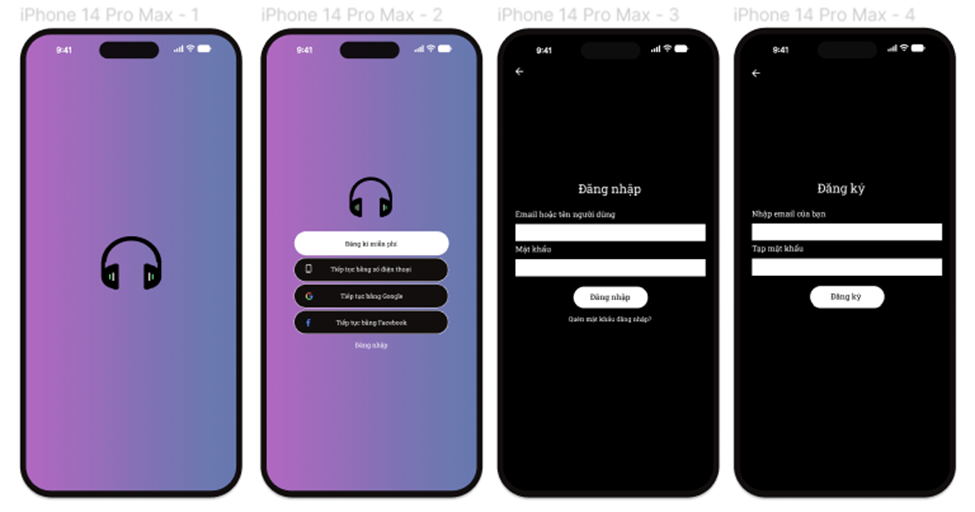
Biểu đồ 2.11 Biểu đồ tuần tự sửa thông tin người dùng

# CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG

## **3.1. Giao diện người dùng**

### **3.1.1. Giao diện đăng ký, đăng nhập**

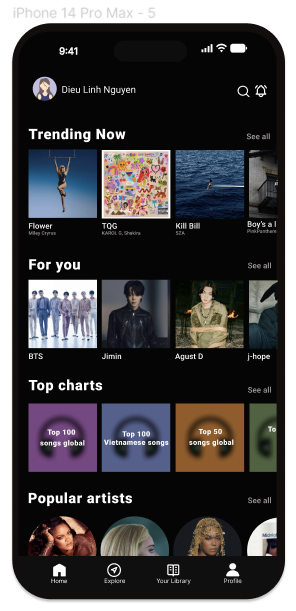
Tại giao diện này hiển thị các cách mà người dùng có thể bắt đầu sử dụng được app nghe nhạc. Người dùng có thể dễ dàng đăng kí, đăng nhập tài khoản cá nhân để bắt đầu nghe nhạc.



Hình 3.1 Giao diện đăng ký, đăng nhập

### **3.1.2. Giao diện chính**

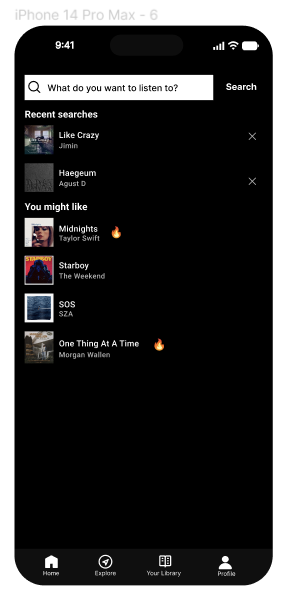
Tại giao diện này có danh sách các bài hát đang thịnh hành, bài hát đề xuất cũng như bảng xếp hạng các xu hướng nghe nhạc hiện nay. Người dùng có thể dễ dàng chọn và nghe các bài hát.



Hình 3.2 Giao diện chính

### **3.1.3. Giao diện tìm kiếm**

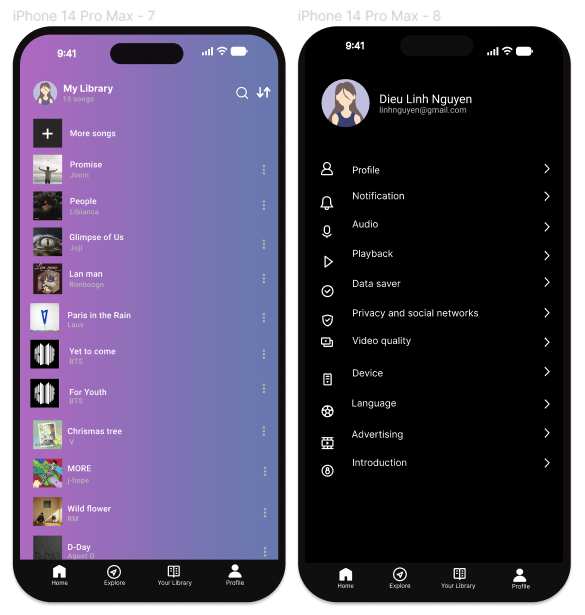
Tại giao diện này, người dùng nhập từ bàn phím từ khóa vào sau đó nhấn nút tìm kiếm. Hệ thống sẽ trả về kết quả tìm kiếm liên quan đến từ khóa.



Hình 3.3 Giao diện tìm kiếm

### **3.1.4. Các giao diện liên quan**

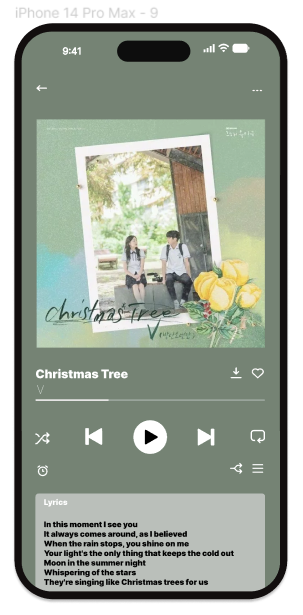
Nếu người dùng yêu thích bài nhạc nào có thể đưa vào mục yêu thích để tạo ra 1 album nhạc của riêng mình. Ngoài ra người dùng cũng có thể thay đổi một số tác vụ theo ý mình.



Hình 3.4 Các giao diện liên quan

### **3.1.5. Giao diện phát nhạc**

Tại giao diện này, người dùng có thể chọn chuyển bài (lùi về trước hoặc tiến về sau), tạm ngưng (hoặc tiếp tục), bài hát ngẫu nhiên (hoặc theo thứ tự bảng chữ cái) và vòng lặp.

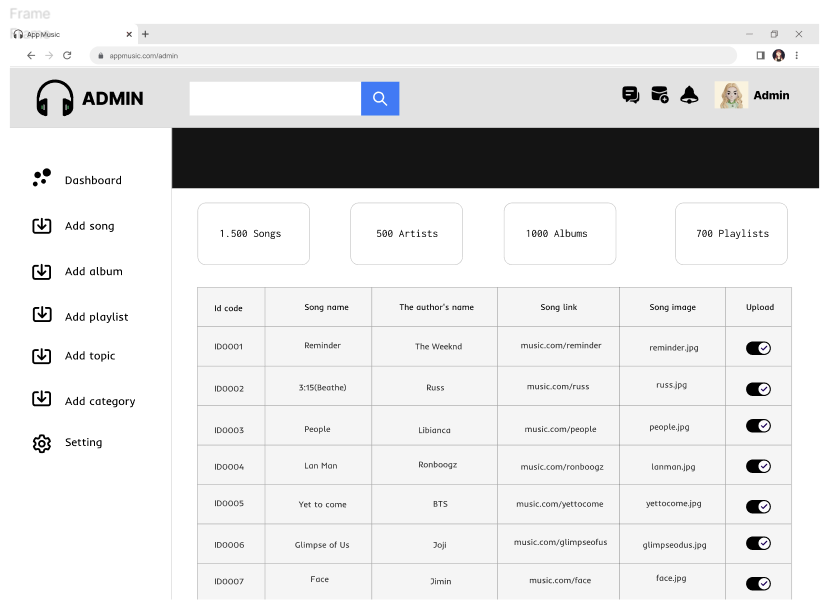


Hình 3.5 Giao diện phát nhạc

## **3.2. Giao diện quản trị**

### Giao diện chính

Giao diện này cho phép hiển thị chính xác thông tin của các bài hát hiện có trong cơ sở dữ liệu. Admin có thể lựa chọn các nút điều hướng ứng với mỗi chức năng để mở các giao diện khác.



Hình 3.6 Giao diện chính

# CHƯƠNG 4. KẾT QUẢ

[❖](https://docs.google.com/document/d/16J81B1F1g7LuhQlGVO69lO9o9yIFaaQF/edit#heading=h.32hioqz) Nội dung đã thực hiện:

* Quá trình xây ứng dụng nghe nhạc trực tuyến của chúng tôi đã là một hành trình đầy thú vị và đầy thách thức. Từ những ý tưởng ban đầu cho đến sản phẩm hoàn chỉnh, chúng tôi đã nỗ lực để tạo ra một sản phẩm khá chỉnh chu và dễ sử dụng.
* Bắt đầu bằng việc lên ý tưởng về mục tiêu và phạm vi của ứng dụng. Chúng tôi đã thực hiện các cuộc họp và nghiên cứu thị trường để hiểu rõ nhu cầu của khách hàng và xu hướng nghe nhạc trực tuyến của các đối tượng khác nhau. Điều này đã giúp chúng tôi xác định được mục tiêu chính của sản phẩm là tạo ra một nền tảng trực tuyến để thưởng thức những bài hát yêu thích và tạo ra một trải nghiệm tốt cho người sử dụng.
* Tiếp theo, chúng tôi đã tiến hành thiết kế giao diện cho ứng dụng. Sự tương tác giữa người dùng và giao diện là yếu tố quan trọng để tạo ra một ứng dụng hoàn thiện để sử dụng.
* Cuối cùng, chúng tôi đã hoàn thiện và tiến hành kiểm tra ứng dụng để đem đến cho người dùng trải nghiệm tốt nhất.

[❖](https://docs.google.com/document/d/16J81B1F1g7LuhQlGVO69lO9o9yIFaaQF/edit#heading=h.32hioqz) Hướng phát triển trong tương lai:

Trong báo cáo này, chúng tôi tập trung vào việc thiết kế ứng dụng thân thiện với khách hàng. Trong tương lai chúng tôi sẽ tiếp tục phát triển nó để nâng cao hiệu quả và tính hữu dụng cho ứng dụng. Hướng phát triển mà chúng tôi hướng tới sau này là sử dụng các yêu cầu chức năng use case. Use case là một công cụ phân tích và mô tả người dùng và hệ thống tương tác để đạt được mục tiêu cụ thể. Bằng cách sử dụng use case, chúng tôi có thể xác định và triển khai các chức năng mới và cải tiến hiện có để cung cấp trải nghiệm tốt hơn người sử dụng. Dưới đây là một số ví dụ về cách sử dụng use case để phát triển website:

* Use case đăng nhập
* Use case đăng ký
* Use case quản lý bài hát yêu thích
* Use case thêm bài hát cho tác giả

**Tài liệu tham khảo**

(1) *Tổng quan về Flutter, cài đặt flutter và chạy chương trình đầu tiên – Phan Nhật Chánh*, 2021. <https://phannhatchanh.com/blog/cai-dat-flutter-va-chay-chuong-trinh-dau-tien>

(2) *Why Visual Studio Code?* (n.d.). Visual Studio Code from [https://code.visualstudio.com/docs/editor/whyvscode#](https://code.visualstudio.com/docs/editor/whyvscode)

(3) *Git - Documentation*. (n.d.). Git SCM from <https://git-scm.com/doc>

(4) *Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial*. (2022, February 24). YouTube from <https://www.youtube.com/watch?v=VPvVD8t02U8>

(5) *Từ gà tới pro Git và Github trong 20 phút - Tự học Git siêu tốc*. (2019, November 5). YouTube from <https://youtu.be/1JuYQgpbrW0>

(6) *Figma là gì? Ứng dụng, ưu điểm, cách sử dụng Figma*. (n.d.). CareerBuilder from <https://careerbuilder.vn/vi/talentcommunity/figma-la-gi-ung-dung-uu-diem-cach-su-dung-figma.35A520B7.html>

(7) *Figma*. (n.d.). YouTube from <https://www.youtube.com/channel/UCQsVmhSa4X-G3lHlUtejzLA>

(8) *Flutter documentation | Flutter* (n.d.). Flutter from

<https://docs.flutter.dev/>