# **BỘ GIÁO DỤC & ĐẠO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**-------------------🖎🕮✍-------------------**

**A logo with a sun and text

Description automatically generated with medium confidence**

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH MOBILE.**

**ĐỂ TÀI:** *Tìm hiểu API vè các thành phó trên thế giới từ nhiều nguồn trên Internet, sau đó xây dựng App liên quan đến lĩnh vực này.*

**Giáo viên hướng dẫn: Trần Đức Minh  
Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã SV** | **Họ & Tên** | **Lớp** |
| 28 | 1571020168 | Lê Văn Mạnh | CNTT 15-01 |

Hà Nội, 08/2024

**BỘ GIÁO DỤC & ĐẠO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**-------------------🖎🕮✍-------------------**

**A logo with a sun and text

Description automatically generated with medium confidence**

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH MOBILE.**

**ĐỂ TÀI:** *Tìm hiểu API vè các thành phó trên thế giới từ nhiều nguồn trên Internet, sau đó xây dựng App liên quan đến lĩnh vực này.*

**Giáo viên hướng dẫn: Trần Đức Minh  
Sinh viên thực hiện:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã SV** | **Họ & Tên** | **Ngày sinh** | **Điểm** | |
| Bằng số | Bằng chữ |
| 28 | 1571020168 | Lê Văn Mạnh | 09/10/2003 |  |  |

Hà Nội, 08/2024

# **LỜI NÓI ĐẦU**

Sau đây là bài báo cáo sơ lược về quy trình xây dựng và phát triển mà em tùm hiểu được mong ***Thầy*** xem và cho ý kiến đánh giá để em nhận ra những khiếm khuyết của mình. Báo cáo đề tài gồm:

* Giới thiệu.
* Chương 1: Mở đầu.
* Chương 2: Cơ sở lý thuyết.
* Chương 3: Triển khai xây dựng chương trình.
* Chương 4: Kết luận.
* Danh mục tham khảo.

Em xin trân thành cảm ơn ***Trần Đức Minh*** đã tận tình hướng dẫn chỉ bảo em hoàn thành đề tài này.

# **LỜI CẢM ƠN**

Để hoàn thành được bài tiểu luận này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường **Đại Học Đại Nam**, những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập.

Đặc biệt, em muốn bày tỏ lòng biết ơn đặc biệt đến giảng viên ***Trần Đức Minh*** - người đã trực tiếp chỉ bảo, hướng dẫn em trong quá trình hoàn thành bài tiểu luận này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc. Những góp ý chân thành của ***Thầy*** đã giúp em làm sâu sắc hơn về vấn đề và cải thiện chất lượng của bài viết.

Trong quá trình thực hiện bài tiểu luận này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất kính mong nhận được ý kiến đóng góp của quý thầy cô để bài tiểu luận được hoàn thiện hơn.

Cuối cùng, em muốn gửi lời cảm ơn đặc biệt đến tất cả những người đã đọc và đánh giá bài báo cáo này. Sự quan tâm và sự phản hồi của ***Thầy*** không chỉ là nguồn động viên mà còn là động lực để em tiếp tục nghiên cứu và phát triển trong tương lai

Một lần nữa, em xin chân thành cảm ơn ***Thầy*** vì sự giúp đỡ và hỗ trợ trong việc hoàn thành bài tiểu luận này.

Em xin chân thành cảm ơn!

**GIỚI THIỆU**

Trong thời đại công nghệ ngày nay, thiết bị di động đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Từ việc gọi điện, nhắn tin đến sử dụng các ứng dụng để làm việc, giải trí và kết nối với thế giới xung quanh, điện thoại di động và các thiết bị thông minh đã trở thành một phần quan trọng của cuộc sống hiện đại.

Môn học Lập trình Mobile chính là cánh cửa mở ra cho việc khám phá và tận dụng sức mạnh của các thiết bị di động này. Trong môn học này, chúng ta sẽ được giới thiệu về cách phát triển ứng dụng cho các nền tảng di động phổ biến như Android và iOS.

Qua các bài học, sinh viên sẽ được làm quen với các ngôn ngữ lập trình như Java, Kotlin (cho Android) và Swift (cho iOS), cùng với các công cụ và framework như Android Studio và Xcode. Chúng ta sẽ học cách xây dựng giao diện người dùng đẹp mắt, tương tác với cơ sở dữ liệu, và triển khai các tính năng đa dạng trong ứng dụng di động.

Ngoài ra, môn học cũng tập trung vào việc hiểu rõ về các xu hướng công nghệ mới trong lĩnh vực lập trình di động và cách áp dụng chúng vào việc phát triển ứng dụng thực tế.

Với sự phát triển không ngừng của thị trường di động, kỹ năng lập trình di động sẽ là một phần quan trọng trong hộp công cụ của các nhà phát triển phần mềm trong tương lai.

1. **Chương 1: MỞ ĐẦU.**
   1. **Tính cấp thiết của đề tài.**

Trong thời đại số hóa ngày nay, việc sử dụng và tận dụng dữ liệu từ các API về các thành phố trên thế giới đang trở thành một phần không thể thiếu trong nhiều ứng dụng và dự án khác nhau. Dữ liệu từ các API này không chỉ cung cấp thông tin về các thành phố, mà còn cung cấp thông tin về thời tiết, múi giờ, điều kiện giao thông, địa điểm du lịch, và nhiều thông tin khác.

**Tính cấp thiết của đề tài này có thể được thấy qua các mặt sau:**

* *Nhu cầu thực tiễn:* Công nghệ di động đã và đang thay đổi cách chúng ta tương tác với thế giới xung quanh, và thông tin về các thành phố là một phần quan trọng trong cuộc sống hàng ngày của chúng ta. Từ việc lập kế hoạch du lịch đến việc xem xét điều kiện thời tiết, người dùng đều cần thông tin chính xác và dễ dàng truy cập về các thành phố trên thế giới.
* *Đa dạng ứng dụng:* Dữ liệu về các thành phố không chỉ hữu ích cho các ứng dụng du lịch mà còn cho các ứng dụng liên quan đến thời tiết, giao thông, giáo dục, và nhiều lĩnh vực khác. Việc có thể truy cập vào các API này giúp phát triển các ứng dụng đa dạng và đáp ứng nhu cầu của đa dạng người dùng.
* *Sự phát triển của công nghệ:* API là một cách hiệu quả và linh hoạt để chia sẻ dữ liệu giữa các ứng dụng và dịch vụ khác nhau. Việc tìm hiểu và sử dụng các API từ nhiều nguồn khác nhau không chỉ giúp phát triển ứng dụng của chúng ta mà còn mở ra cơ hội hợp tác và tích hợp với các dự án khác.

**Phương pháp thực hiện:**

* *Tìm hiểu API:* Bắt đầu bằng việc tìm kiếm và nghiên cứu các API có sẵn trên Internet về các thành phố trên thế giới. Các nguồn thông tin như documentations của API, hướng dẫn sử dụng, và ví dụ về cách sử dụng sẽ giúp bạn hiểu rõ về cách thức hoạt động của các API này và cách trích xuất thông tin từ chúng.
* *Phân tích và lựa chọn:* Dựa trên nhu cầu và yêu cầu của ứng dụng mà bạn muốn phát triển, phân tích và lựa chọn các API phù hợp nhất. Đảm bảo rằng API cung cấp đủ thông tin và tính năng cần thiết cho ứng dụng của bạn.
* *Xây dựng ứng dụng:* Sau khi đã chọn được API, bạn có thể bắt đầu xây dựng ứng dụng của mình. Sử dụng ngôn ngữ lập trình và framework phù hợp để tạo ra giao diện người dùng hấp dẫn và kết nối với API để truy xuất thông tin và hiển thị cho người dùng.
* *Kiểm thử và triển khai:* Cuối cùng, đảm bảo rằng ứng dụng của bạn được kiểm thử kỹ lưỡng trước khi triển khai vào sản xuất. Sửa các lỗi và tối ưu hóa hiệu suất của ứng dụng trước khi phát hành để đảm bảo trải nghiệm tốt nhất cho người dùng.
  1. **Tổng quan đề tài.**

Đề tài này tập trung vào việc tìm hiểu và sử dụng các API liên quan đến các thành phố trên thế giới để xây dựng một ứng dụng di động có liên quan đến lĩnh vực này.

Trong thời đại số hóa hiện nay, việc truy cập và sử dụng dữ liệu từ các API trở thành một phần không thể thiếu trong việc phát triển ứng dụng di động. Với sự phổ biến của các thiết bị thông minh, người dùng ngày càng có nhu cầu cao hơn về thông tin và tiện ích từ các ứng dụng trên điện thoại di động của mình.

Thông qua việc tìm hiểu và sử dụng các API về các thành phố trên thế giới, chúng ta có thể tạo ra những ứng dụng mang lại giá trị cho người dùng, từ việc cung cấp thông tin về thời tiết, địa điểm du lịch, múi giờ, đến thông tin về giao thông, cơ sở hạ tầng, và nhiều lĩnh vực khác.

Trong quá trình này, chúng ta sẽ tìm hiểu về cách sử dụng các API từ nhiều nguồn khác nhau, phân tích và lựa chọn các API phù hợp nhất cho ứng dụng của mình, và xây dựng một ứng dụng di động thực tế và hữu ích.

Bằng cách này, đề tài không chỉ giúp chúng ta áp dụng kiến thức lập trình di động vào thực tế mà còn mở ra cơ hội học hỏi và nâng cao kỹ năng trong việc sử dụng và tích hợp dữ liệu từ các nguồn khác nhau.

* 1. **Mục đích nghiên cứu.**

Tìm hiểu về các API về các thành phố trên thế giới: Nghiên cứu sẽ tập trung vào việc khám phá và hiểu rõ về các API có sẵn trên Internet liên quan đến các thành phố trên thế giới. Điều này bao gồm việc tìm hiểu về cách hoạt động của các API, các endpoints có sẵn, cách truy xuất dữ liệu từ các API này, và các hạn chế và quy định liên quan.

Phát triển kỹ năng lập trình di động và tích hợp dữ liệu: Qua việc xây dựng một ứng dụng di động liên quan đến các thành phố trên thế giới, mục đích của nghiên cứu là phát triển kỹ năng lập trình di động và kỹ năng tích hợp dữ liệu từ các nguồn khác nhau. Điều này bao gồm việc làm quen với các framework và công cụ phát triển di động, viết mã để giao tiếp với các API và hiển thị dữ liệu lấy được lên giao diện người dùng của ứng dụng.

Tạo ra một ứng dụng di động có giá trị thực tiễn: Mục đích cuối cùng của nghiên cứu là tạo ra một ứng dụng di động thực tế và hữu ích cho người dùng. Ứng dụng này sẽ cung cấp thông tin và tiện ích liên quan đến các thành phố trên thế giới, từ việc cung cấp thông tin thời tiết đến địa điểm du lịch và các thông tin hữu ích khác.

Thông qua việc đạt được các mục tiêu trên, nghiên cứu sẽ đóng góp vào việc nâng cao kiến thức và kỹ năng của người tham gia và tạo ra một sản phẩm ứng dụng có giá trị cho cộng đồng người dùng.

* 1. **Đối tượng và phạm vi nghiên cứu.**

**Đối tượng của nghiên cứu này bao gồm:**

* *Người học Lập trình Mobile:* Sinh viên và những người quan tâm đến việc phát triển ứng dụng di động có thể là đối tượng chính của nghiên cứu. Đối tượng này muốn tìm hiểu và phát triển kỹ năng trong việc sử dụng các API và xây dựng ứng dụng di động liên quan đến các thành phố trên thế giới.
* *Người sử dụng ứng dụng di động*: Đối tượng này bao gồm mọi người có nhu cầu sử dụng ứng dụng di động để truy cập thông tin và tiện ích về các thành phố trên thế giới, từ việc cần biết thông tin thời tiết đến việc lập kế hoạch du lịch.

**Phạm vi của nghiên cứu bao gồm:**

* *Tìm hiểu và sử dụng các API về các thành phố trên thế giới:* Nghiên cứu sẽ tập trung vào việc khám phá và hiểu rõ về các API có sẵn trên Internet liên quan đến các thành phố trên thế giới. Điều này bao gồm việc tìm hiểu về cách hoạt động của các API, các endpoints có sẵn, cách truy xuất dữ liệu từ các API này, và các hạn chế và quy định liên quan.
* *Xây dựng ứng dụng di động liên quan đến các thành phố trên thế giới:* Phạm vi của nghiên cứu bao gồm việc phát triển một ứng dụng di động thực tế và hữu ích cho người dùng. Ứng dụng này sẽ cung cấp thông tin và tiện ích liên quan đến các thành phố trên thế giới, từ việc cung cấp thông tin thời tiết đến địa điểm du lịch và các thông tin hữu ích khác.
* *Hướng dẫn và hỗ trợ cho người học:* Nghiên cứu sẽ cung cấp hướng dẫn chi tiết và hỗ trợ cho người học trong việc tìm hiểu và áp dụng kiến thức vào thực tế thông qua việc xây dựng ứng dụng di động.
  1. **Phương pháp nghiên cứu.**

Để đạt được mục tiêu nghiên cứu và giải quyết các vấn đề được đề ra, phương pháp nghiên cứu được sử dụng sẽ bao gồm các bước chính sau:

* *Tìm hiểu và lựa chọn API:* Bắt đầu bằng việc nghiên cứu và khám phá các API có sẵn trên Internet liên quan đến thông tin về các thành phố trên thế giới. Đánh giá và lựa chọn các API phù hợp nhất với mục đích của ứng dụng sẽ là bước quan trọng trong quá trình này.
* *Phát triển ứng dụng di động:* Sử dụng kiến thức về lập trình di động và tích hợp dữ liệu từ API, phát triển một ứng dụng di động có khả năng truy cập và hiển thị thông tin về các thành phố trên thế giới. Bao gồm việc xây dựng giao diện người dùng, lập trình logic và kết nối với các endpoint của API.
* *Kiểm thử và sửa lỗi:* Thực hiện các bước kiểm thử để đảm bảo tính ổn định và đáp ứng của ứng dụng. Sửa lỗi và tối ưu hóa hiệu suất của ứng dụng để đảm bảo trải nghiệm tốt nhất cho người dùng.
* *Đánh giá và cải tiến:* Tiến hành đánh giá về hiệu suất và tính năng của ứng dụng sau khi triển khai. Nhận phản hồi từ người dùng và đề xuất các cải tiến và nâng cấp để cải thiện trải nghiệm người dùng.
* *Tài liệu hóa và báo cáo:* Tài liệu hóa quá trình phát triển ứng dụng, ghi lại các kết quả đạt được và kinh nghiệm học được từ quá trình nghiên cứu. Viết báo cáo kết quả nghiên cứu để chia sẻ kiến thức và kinh nghiệm với cộng đồng.

1. **Chương 2: CƠ SỞ LÝ THUYẾT.**
   1. **Android Studio.**

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA. Chức năng chính của Android Studio là cung cấp các giao diện giúp người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp sau hậu trường. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java và nó sẽ được cài đặt sẵn trên thiết bị của bạn. Khi sử dụng Android Studio thì bạn chỉ cần viết, chỉnh sửa và lưu trữ chúng trên các dự án của mình và các file nằm trong dự án đó. Đồng thời, Android Studio còn cung cấp quyền truy cập vào SDK.



Ngoài ra, bạn có thể xem Android Studio là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android rồi tận dụng được lợi thế của phần cứng gốc. Bạn chỉ cần sử dụng ngôn ngữ lập trình Java để có thể viết chương trình, khi đó Android SDK sẽ có nhiệm vụ kết nối các phần này lại với nhau. Khi đó, Android Studio sẽ kích hoạt để thực hiện chạy code và thông qua trình giả lập hoặc dựa vào bất kỳ phần cứng giúp kết nối với thiết bị. Sau đó, bạn có thể gỡ rối cho chương trình của mình ngay khi nó chạy và nhận phản hồi giúp giải thích các sự cố,...Cho đến hiện nay, Google đã và đang rất nỗ lực để giúp cho Android Studio sẽ trở nên mạnh mẽ và hữu ích hơn. Khi bạn gõ code, nó sẽ giúp bạn cung cấp danh sách gợi ý hoàn thành để giúp người dùng có thể hoàn thiện được dòng code đó. Đây là một trong những chức năng rất hữu ích đề phòng cho trường hợp người dùng không nhớ chính xác cú pháp giúp bạn tiết kiệm thời gian hiệu quả hơn.

* 1. **Flutter.**

Flutter là một bộ công cụ phát triển ứng dụng đa nền tảng đáng kinh ngạc, được giới thiệu bởi Google. Nó sử dụng ngôn ngữ [Dart](https://dart.dev/) để lập trình. Flutter được ra mắt vào năm 2018 với các tính năng còn thiếu của các công cụ phát triển đa nền tảng trước đó. Các ứng dụng được xây dựng với Flutter có thể chạy trên Android, iOS, [Raspberry Pi](https://medium.com/flutter/flutter-on-raspberry-pi-mostly-from-scratch-2824c5e7dcb1) và Google Fuchsia,,một nền tảng phát triển ứng dụng khác do Google phát triển.

Flutter đang được sử dụng bởi các developer, điều đó chứng tỏ rằng có một cái gì đó độc đáo đã tồn tại và thu hút các developer. Dưới đây là một trong những điều đó:

* Khả năng hỗ trợ nhiều API Firebase hơn
* Có công cụ riêng Sửa lỗi
* Tài liệu cải tiến Hỗ trợ phát triển windows
* Hiệu suất được nâng cao
* Công cụ cho Android Studio và Visual Studio Code
* Các tính năng bổ sung như video, biểu đồvà quảng cáo

Kể từ khi Flutter trở nên nổi tiếng, câu hỏi này đã được đưa ra. Trước khi quyết định xây dựng một ứng dụng di động Flutter, mọi người luôn muốn biết rằng nó hỗ trợ phát triển cho loại nào và không cho loại nào. Flutter không phải là lựa chọn cho bạn nếu bạn đang phát triển:

* Các ứng dụng web được phát triển nhanh và tức thì – Các loại ứng dụng web nhỏ.
* Các ứng dụng cần thư viện riêng không phổ biến – Flutter là còn mới và nó không có tất cả các thư viện riêng trong kho lưu trữ của nó, vì vậy nếu bất kỳ thư viện gốc không phổ biến nào được yêu cầu bởi một ứng dụng của bạn, nó có thể làm phức tạp quá trình phát triển ứng dụng của bạn. Trong trường hợp đó, các nhà phát triển sẽ phải thêm các custom và mà chắc chắn sẽ mất thời gian.
* Các ứng dụng giao tiếp với phần cứng bluetooth – Tính năng này có thể được xây dựng riêng cho cả Android và iOS và sau đó được thêm vào ứng dụng Flutter. Tuy nhiên, điều này có thể tốn thời gian lập trình.
  1. **Dart.**

Dart, là một ngôn ngữ lập trình đa mục đích mã nguồn mở, đã được đặt nền móng bởi Google. Đây là một khía cạnh của ngôn ngữ lập trình hướng đối tượng, đặc trưng bởi cú pháp kiểu C. Tinh thần lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp, mở ra khả năng sáng tạo không giới hạn khi đặt lên bàn cân với những ngôn ngữ khác. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn.

Ứng dụng ngôn ngữ Dart trong lập trình di động

Không chỉ dừng lại ở việc phát triển ứng dụng cho một nền tảng, Dart mở ra cánh cửa cho sự sáng tạo trên cả hai nền tảng quan trọng: Android và iOS. Và trong cuộc hành trình này, [Flutter](https://200lab.io/blog/flutter-la-gi/) nổi lên như một công cụ vượt trội, được Google chắp cánh. Flutter, một framework ra đời từ Google, trở thành điểm đặc biệt giúp xây dựng những ứng dụng tuyệt đẹp trên cả hai nền tảng bằng một nền tảng mã nguồn duy nhất.

Smalltalk được phát hành vào năm 1970 và là một trong những ngôn ngữ lập trình hướng đối tượng nguyên bản đầu tiên. Ngày nay ngôn ngữ hướng đối tượng đã trở nên phổ biến và dường như thống trị trong thế giới ngôn ngữ lập trình.

Ý tưởng đằng sau khái niệm lập trình hướng đối tượng rất đơn giản: các chương trình yêu cầu một dạng cấu trúc cụ thể.

Cách rõ ràng nhất để đạt được cấu trúc cụ thể này là sử dụng khái niệm containers. Một ngôn ngữ lập trình có thể được chia thành dữ liệu và những phương thức thực hiện các dữ liệu đó. Dữ liệu cụ thể và những phương thức sẽ được gói trong một số loại container.

Hơn nữa, những container này được tạo ra để trở nên dễ sử dụng. Vì thế chúng không chỉ chứa dữ liệu và phương thức mà chúng cũng chính là những giá trị được chứa vào những container khác và được chuyển dưới dạng tham số cho các hoạt động khác. Trong lập trình hướng đối tượng, những container này được biết như là những đối tượng.



Photo by Martin Shreder / Unsplash

Alan Kay, người phát minh ra Smalltalk, nhận xét rằng theo cách này, đối tượng đơn giản nhất có nguyên tắc cấu tạo giống như một máy tính hoàn chỉnh: nó kết hợp dữ liệu với các phương thức dưới một giao diện được chính thức hóa.

Và bây giờ, mặc dù lập trình hướng đối tượng có thể được tìm thấy trong vô số ngôn ngữ, nhưng có rất ít ngôn ngữ thực sự tuân theo các nguyên tắc do Smalltalk đặt ra.

Dart là một ngôn ngữ lập trình hướng đối tượng thuần túy với mọi giá trị là một đối tượng.

**Ngôn ngữ lập trình Dart: dưới góc nhìn của JavaScript**

Dart là một ngôn ngữ hướng đối tượng đơn giản, clean và dựa trên class. Nó còn có nhiều cấu trúc hơn cả JavaScript - một ngôn ngữ mà nó chịu ảnh hưởng thiết kế khá là nhiều.

Đó là một điều tốt cho các developer những ai yêu thích việc có cấu trúc trong ngôn ngữ lập trình của mình. Các dev có thể dễ dàng tái cấu trúc và xây dựng những ứng dụng web lớn hơn.

Theo như những nhà sáng lập, một trong những thứ họ tập trung vào khi tạo ra ngôn ngữ này đó chính là tính tương thích của nó với web. Vì vậy, một trong những component quan trọng nhất của Dart đó chính là trình biên dịch Dart sang JavaScript. Nó sẽ dịch source code của Dart sang Javascript và đảm bảo rằng bạn sẽ nhận đúng ngữ nghĩa giống như khi bạn chạy nó trên máy ảo của javascript.

**Từ Dart đến Framework Flutter**

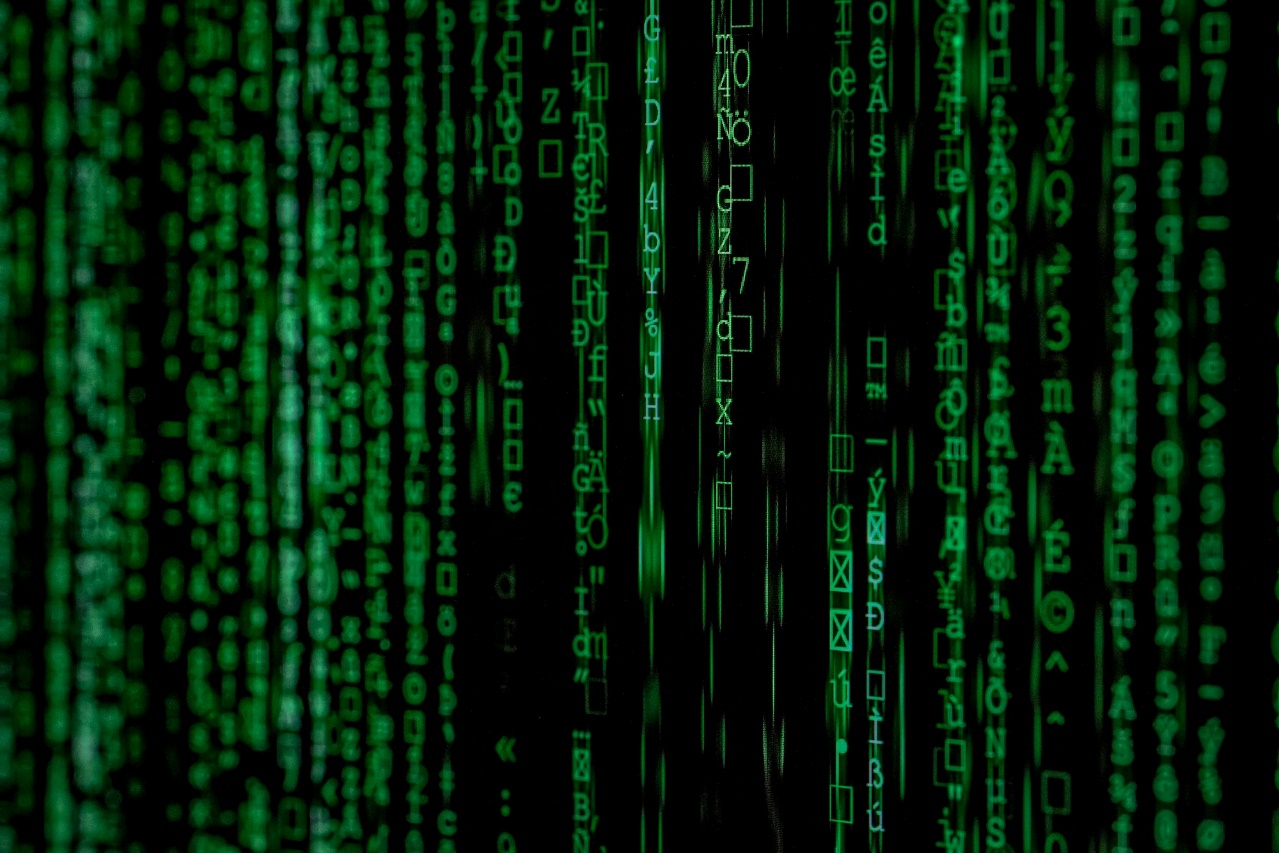


Photo by Markus Spiske / Unsplash

Flutter đã thu hút được sự chú ý của cộng đồng các nhà phát triển bằng cách giới thiệu các style cho phép việc xây dựng UI đẹp hơn và biểu cảm hơn vì thế mà việc code cũng trở nên thú vị hơn nhiều. Nó kết hợp một số khái niệm quen thuộc với những kinh nghiệm phát triển hiện đại như lập trình reactive và widget composition trong khi sử dụng nền tảng Dart làm cơ sở chính cho các hoạt động đó. Nhóm Flutter đã đánh giá nhiều ngôn ngữ khác nhau và cuối cùng họ chọn Dart vì nó phù hợp với cách mà họ xây dựng giao diện người dùng.

**Dưới đây là những lý do tại sao mà ngôn ngữ Dart lại được Flutter lựa chọn để sử dụng:**

* **Tối ưu cho giao diện người dùng:** Hoạt động bất đồng bộ cho phép chương trình của bạn có thể hoàn thành hoàn toàn công việc trong khi chờ hoạt động khác kết thúc. Đây là một số hoạt động bất đồng bộ phổ biến:
* Tìm nạp dữ liệu thông qua mạng
* Viết cơ sở dữa liệu
* Đọc dữ liệu từ file

Hầu hết các máy tính, thậm chí các nền tảng mobile, đều có CPU đa nhân. Để tận dụng hết tất cả các nhân đó, các dev thông thường phải sử dụng các luồng bộ nhớ chia sẻ đồng thời. Các concurrency ở trạng thái chia sẻ có thể dễ bị lỗi và có thể dễ dẫn đến những code khá phức tạp. Thay vì sử dụng luồng, tất cả các code của ngôn ngữ Dart chạy bên trong các isolate. Mỗi isolate có vùng bộ nhớ riêng, đảm bảo rằng không có bất kỳ trạng thái của isolate này có thể truy cập được vào isolate kia.

Ngôn ngữ lập trình này cũng tối ưu cho việc xây dựng giao diện người dùng với những tính năng cho phép mở rộng các collection và tùy chỉnh UI cho từng nền tảng khác nhau.

* **Việc phát triển năng suất hơn:**
* Flutter có tính năng hot reload giúp bạn thử nghiệm, xây dựng UI, thêm tính năng và fix bug một cách nhanh chóng và dễ dàng hơn. Tính năng hot reload hoạt động bằng cách đưa những file source code đã được update vào máy ảo Virtual Machine (VM) của Dart. Sau khi VM cập nhập các class với phiên bản mới nhất của field và function, framework Flutter sẽ tự động tái xây dựng cây widget, cho phép bạn xem các hiệu ứng mà bạn đã thay đổi một cách nhanh chóng hơn.
* Flutter cung cấp phân tích static cho phép bạn phát hiện ra những vấn đề ngay trước khi dòng code đó được thực thi. Nó thực sự là một công cụ mạnh mẽ có thể giúp các dev tránh việc phát sinh bug và đảm bảo được code tuân theo các quy tắc của style.
* **Hiệu năng nhanh trên tất cả nền tảng:**
* Ngôn ngữ Dart có trình biên dịch AOT (Ahead of Time) giúp nó biên dịch nhanh, đúng và native code. Điều này không chỉ đảm bảo giúp Flutter nhanh hơn mà còn đảm bảo rằng hầu như mọi thứ (bao gồm tất cả các widget) đều có thể được tùy chỉnh. Với lý do đó cho nên hầu hết các phần của Flutter đều được viết bằng ngôn ngữ này.
* Với tất cả những cột mốc thành tựu, không thể phủ nhận vai trò và tiềm năng của Dart trong giới lập trình. Dù đã có những bước tiến, Dart vẫn tiếp tục hướng tới tương lai với sự đổi mới và sáng tạo không ngừng. Như một cây cầu nối giữa ý tưởng và hiện thực, Dart tiếp tục mở ra cơ hội cho sự sáng tạo và phát triển trong tương lai của lập trình.
  1. **SDK.**

SDK là viết tắt của Software Development Kit hoặc Devkit. Đây là một tập hợp các công cụ và chương trình phần mềm được các developer sử dụng để tạo ứng dụng cho nền tảng cụ thể.

**Ví dụ:** nếu đang tạo ứng dụng cho thiết bị di động, bạn sẽ cần tải SDK iOS của Apple. Điều này cho phép bạn truy cập vào bất kỳ công cụ iPhone nào bạn muốn sử dụng trong ứng dụng của mình. Chẳng hạn như kết nối với máy ảnh, giao diện Siri hoặc bật thông báo. Tương tự như vậy, nếu bạn muốn triển khai nền tảng của [Auth0](https://auth0.com/) trên cả web app và thiết bị của mình, bạn phải tải xuống SDK iOS, SDK Android và SDK Web.

**Thành phần cơ bản trong SDK :**

* *Code library:* Đây là code mà các developer sử dụng làm cho ứng dụng làm những gì họ muốn.
* *Application programming interfaces (APIs*): Những giao diện này cho phép ứng dụng dễ dàng kết nối với các dịch vụ khác.
* *Integrated development environment (IDE):* Đây là giao diện mà thông qua đó các developer thực hiện việc lập trình. Nó bao gồm một trình biên dich, dịch ngôn ngữ lập trình cấp cao của mã nguồn của SDK sang ngôn ngữ thấp hơn có thể sử dụng để tạo ứng dụng.
* *Documentation:* Đây là những hướng dẫn giải thích cách các developer nên sử dụng code. Tài liệu rõ ràng, kỹ lưỡng là tiêu chí quan trọng cho một SDK tốt.
* *Debugger:* Nó sẽ tự động sửa các lỗi nhỏ, làm trơn tru quá trình phát triển ứng dụng.
* *Code samples:* Những code này giúp các developer bắt đầu bằng cách cung cấp cho họ một số ví dụ về code đang hoạt động và giúp họ sử dụng nó theo cách đơn giản. Một số SDK cũng bao gồm hướng dẫn và các công cụ hỗ trợ khác.

**Các loại SDK:**

* *SDK theo phần cứng:* SDK không chỉ dành cho web và ứng dụng dành cho thiết bị di động; chúng cũng được sử dụng để lập trình trong Internet of Things (loT).
* *Hệ điều hành thiết bị di động:* Như đã đề cập, việc thiết kế một ứng dụng hoạt động trên Android hay Apple thì đều yêu cầu một SDK cho mỗi loại.
* *Ngôn ngữ lập trình cho web app:* Developer SDK cho phép họ tạo web app bằng ngôn ngữ lập trình mà họ chọn. Chẳng hạn như Python, Ruby, JavaScript hoặc PHP.
* *Các SDK nguồn mở:* Các SDK này được sử dụng miễn phí để các developer sửa đổi chúng. Điều này có thể mang lại lợi thế khi nói đến khả năng tùy chỉnh nhưng cũng có thể gây rủi ro bảo mật.
* *SDK độc quyền:* Không giống như DSK nguồn mở, SDK độc quyền hoặc thương mại yêu cầu giấy phép để sử dụng và không cho phép các nhà phát triển thay đổi mã nguồn.

**Lợi ích nổi bật của SDK mang lại:**

Vậy lợi ích nổi bật mà đến từ SDK là gì? SDK iOS và Android là yếu tố chính vì chúng thuộc phạm vi tiếp cận của các nền tảng trên toàn thế giới. Việc không sử dụng SDK Android và SDK iOS sẽ khiến bạn khó thực hiện hiệu quả tất cả các loại báo cáo sự cố như gửi tin nhắn, thu thập số liệu phân tích.

*Khả năng tích hợp nhanh:* SDK Android sẽ tăng tốc hiệu quả quá trình tích hợp. Devkit của bạn sẽ giúp rút ngắn quá trình tích hợp. Vì nó làm cho việc tích hợp với ngăn xếp công nghệ trở nên dễ dàng hơn.

*Hỗ trợ triển khai nhanh chóng và hiệu quả:* Khi xem xét thực tế rằng Android app trung bình sẽ sử dụng khoảng 18,2 SDK của bên thứ ba (cao hơn nếu là game). Bạn sẽ nhanh chóng thấy rằng không developer phần mềm nào có thời gian để viết code và chỉnh sửa mọi thứ ngay từ đầu.

*Kiểm soát và giảm rủi ro:* Bạn có quyền kiểm soát tốt các chi tiết của giao diện người dùng hiển thị trong các app sử dụng SDK của bạn. Điều này không chỉ giúp bạn quản lý các sản phẩm của bạn tích hợp với app khác mà còn cả giao diện của sản phẩm.Với SDK, bạn có quyền kiểm soát tốt hơn các thành phần của giao diện người dùng hiển thị trong các ứng dụng khác. Điều này không những cho phép bạn cách ra lệnh cho sản phẩm của bạn được tích hợp với các ứng dụng khác như thế nào mà còn cả giao diện của sản phẩm. Tất cả điều này giúp đảm bảo các chức năng quan trọng nhất vẫn duy trì an toàn khỏi việc bị giả mạo và có thể làm hỏng trải nghiệm của người dùng.

*Tăng phạm vi tiếp cận đối với sản phẩm/dịch vụ*: Nếu sản phẩm của bạn có giá trị và SDK đi kèm của nó cho phép khá nhiều khả năng tương tác. Bạn sẽ tăng khả năng các công cụ khác muốn tích hợp với sản phẩm của bạn. Điều này dẫn đến việc hiển thị nhiều hơn cho thương hiệu của bạn. SDK là miễn phí cho tất cả mọi người và giúp tạo các Android app tuyệt vời. SDK Android là một nền tảng mã nguồn mở, cho phép các nhà phát triển chia sẻ các công nghệ đáng giá và quý giá ở quy mô lớn.

* 1. **API.**

**API nghĩa là gì?**

API là cụm viết tắt của Giao diện chương trình ứng dụng. Trong ngữ cảnh API, từ “Ứng dụng” đề cập đến mọi phần mềm có chức năng riêng biệt. Giao diện có thể được xem là một hợp đồng dịch vụ giữa 2 ứng dụng. Hợp đồng này xác định cách thức hai ứng dụng giao tiếp với nhau thông qua các yêu cầu và phản hồi. Tài liệu API của các ứng dụng này chứa thông tin về cách nhà phát triển xây dựng cấu trúc cho những yêu cầu và phản hồi đó.

**API hoạt động như thế nào?**

Kiến trúc API thường được giải thích dưới dạng máy chủ và máy khách. Ứng dụng gửi yêu cầu được gọi là máy khách, còn ứng dụng gửi phản hồi được gọi là máy chủ. Như vậy, trong ví dụ về thời tiết, cơ sở dữ liệu của cơ quan thời tiết là máy chủ còn ứng dụng di động là máy khách.

API hoạt động theo 4 cách khác nhau, tùy vào thời điểm, lý do chúng được tạo ra.

* API SOAP
* API Websocket
* API REST

**Tiện ích tích hợp API là gì?**

Tiện ích tích hợp API là các thành phần phần mềm tự động cập nhật dữ liệu giữa máy khách và máy chủ. Một số ví dụ về tiện ích tích hợp API bao gồm khi dữ liệu tự động đồng bộ với đám mây từ thư viện hình ảnh trong điện thoại của bạn hoặc máy tính xách tay của bạn tự động đồng bộ ngày giờ khi bạn đến một múi giờ khác. Các doanh nghiệp cũng có thể sử dụng chúng để tự động hóa nhiều chức năng của hệ thống một cách hiệu quả.

**API có những loại khác nhau nào?**

API được phân loại theo cả kiến trúc và phạm vi sử dụng. Chúng ta đã khám phá các loại kiến trúc API chính, vậy nên hãy cùng xem xét phạm vi sử dụng nhé.

*API riêng:*Đây là những API nội bộ của một doanh nghiệp và chỉ dùng để kết nối các hệ thống cũng như dữ liệu trong doanh nghiệp đó.Những API này dành cho công chúng, có thể được sử dụng bởi bất kỳ ai. Những loại API này có thể yêu cầu sự ủy quyền hay chi phí nào đó, hoặc không yêu cầu.

*API đối tác:* Những API này chỉ dành cho các nhà phát triển bên ngoài được ủy quyền để hỗ trợ những mối quan hệ hợp tác giữa doanh nghiệp với doanh nghiệp.

*API tổng hợp:* Những API này kết hợp hai API khác nhau trở lên để giải quyết những yêu cầu hay hành vi phức tạp của hệ thống.

**Điểm cuối API là gì và vì sao nó lại quan trọng?**

Điểm cuối API là điểm tiếp xúc cuối cùng trong hệ thống giao tiếp của API. Những điểm cuối này bao gồm URL máy chủ, dịch vụ và những địa điểm kỹ thuật số cụ thể khác, từ đây thông tin được gửi đi và tiếp nhận giữa các hệ thống. Điểm cuối API rất quan trọng đối với doanh nghiệp vì 2 lý do chính:

* *Bảo mật:* Điểm cuối API khiến hệ thống dễ bị tấn công. Việc giám sát API để ngăn tình trạng lạm dụng là rất quan trọng.
* *Hiệu năng:* Điểm cuối API, nhất là những điểm cuối có lưu lượng truy cập cao, có thể gây ra tình trạng nghẽn mạng và ảnh hưởng đến hiệu năng hệ thống.

**Làm thế nào để tạo API?**

Việc xây dựng một API mà các nhà phát triển khác sẽ tin tưởng và muốn sử dụng đòi hỏi phải thẩm định kỹ lưỡng và nhiều công sức. Sau đây là 5 bước cần thực hiện để thiết kế API chất lượng cao:

* *Lên kế hoạch cho API:* Thông số kỹ thuật của API, ví dụ như OpenAPI, cung cấp bản thiết kế cho API của bạn. Bạn nên dự liệu trước các tình huống sử dụng khác nhau và đảm bảo rằng API tuân thủ các tiêu chuẩn phát triển API hiện hành.
* *Xây dựng API:* Các nhà thiết kế API dựng nguyên mẫu cho API bằng mã nguyên mẫu. Sau khi đã kiểm thử nguyên mẫu, nhà phát triển có thể tùy chỉnh nguyên mẫu này theo thông số kỹ thuật nội bộ.
* *Kiểm thử API:* Kiểm thử API tương tự như kiểm thử phần mềm và phải được thực hiện để ngăn lỗi và khiếm khuyết. Công cụ kiểm thử API có thể được sử dụng để thử nghiệm khả năng chống đỡ các cuộc tấn công mạng của API.
* *Lập tài liệu cho API:* Mặc dù không cần giải thích gì về API, tài liệu về API đóng vai trò là hướng dẫn để nâng cao tính khả dụng. Các API được lập tài liệu đầy đủ, cung cấp các chức năng và trường hợp sử dụng đa dạng thường phổ biến hơn trong kiến trúc hướng đến dịch vụ.
* *Đưa API ra thị trường:* Tương tự như thị trường bán lẻ trực tuyến Amazon, nhà phát triển có thể mua bán các API khác trên sàn giao dịch API. Bạn có thể niêm yết API để kiếm tiền từ nó.
  1. **Firebase.**

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.

**Các tính năng chính của Firebase:**

* Realtime Database
* Authentication
* Cloud Storage
* Cloud Firestore
* Hosting
* Cloud Functions
* Analytics

1. **Chương 3: TRIỂN KHAI XÂY DỰNG CHƯƠNG TRÌNH.**
   1. **Danh sách các chức năng.**
      1. **Tìm kiếm thông tin.**

Chức năng này cho phép người dùng tìm kiếm thông tin về đồ uống cocktail dựa trên các tiêu chí như tên cocktail hoặc thành phần chính.

* + 1. **Xem chi tiết.**

Người dùng có thể xem chi tiết về một đồ uống cụ thể, bao gồm hình ảnh, tên, thành phần, và mô tả chi tiết về cách làm.

* + 1. **Đăng ký.**

Cho phép người dùng tạo tài khoản mới thông qua Firebase, cung cấp thông tin cá nhân và an toàn thông tin đăng nhập.

* + 1. **Đăng nhập.**

Chức năng đăng nhập thông qua Firebase giúp quản lý tài khoản người dùng, tạo trải nghiệm cá nhân hóa và cho phép thực hiện các hành động đặc biệt.

* + 1. **Thêm vào Danh sách yêu thích.**

Người dùng có thể thêm một đồ uống vào danh sách yêu thích của mình để dễ dàng truy cập và theo dõi những đồ uống họ quan tâm.

* + 1. **Danh sách yêu thích.**

Hiển thị danh sách các đồ uống mà người dùng đã thêm vào danh sách yêu thích, giúp họ quản lý và xem lại những lựa chọn yêu thích của mình.

* 1. **Phác thảo giao diện.**
  2. **Triển khai Android.**

1. **Chương 4: KẾT LUẬN**
   1. **Kết luận.**
   2. **Kết quả đạt được.**

**TÀI LIỆU THAM KHẢO.**

[1]. Firebase (2021), FPT Cloud.

[2]. SDK (2021), Cao Lê Viết Tiến.

[3]. Fluter (2018), David Xuân.

[4]. Android Studio (2021), IT Navi.

[5]. Firebase\_authentication (2024), Trần Đức Minh.