**PHẦN MỞ ĐẦU**

1. Tính cấp thiết của đề tài

Du lịch là một nghành công nghiệp không khói. Bước vào thế kỷ 21 ngành du lịch ngày càng có những thay đổi rõ ràng, do sự tác động của nhiều yếu tố. Do vậy để phát triển du lịch cần phải có sự thay đổi trong phương pháp quản lý và tiếp cận trong phát triển du lịch. Trong quá trình phát triển chúng ta cần phải thấy được vai trò của các đối tượng, các thành phần và vị trí của nó trong phát triển du lịch của toàn ngành. Đi đôi với du lịch là dịch vụ và sự quảng bá. Để người dùng có thể tiếp cận một cách dễ dàng đến các địa điểm du lịch khi đi đến bất kì đâu, em tập trung vào phát triển một ứng dụng điện thoại giúp phát hiện ra các địa điểm du lịch bằng vị trí hiện tại của người dùng. Ứng dụng cho phép người dùng xem được các địa điểm nổi tiếng mà địa điểm hiện tại của mình đang đứng, đi đôi với đó là các dịch vụ khách sạn, nhà nghỉ, quán ăn. Ứng dụng còn cho phép người dùng là các chủ nhà hàng, khách sạn, chủ khu du lịch cập nhật địa điểm lên một cách dễ dàng. Từ đó giúp người du lịch và ngành du lịch càng tiến lại gần nhau hơn.

1. Mục tiêu của đề tài

Đề tài hướng đến việc sử dụng công nghệ vào ngành du lịch, giúp cho sự tiếp cận của khách hàng, người dùng đến các địa điểm du lịch, quán ăn, nhà nghỉ một cách dễ dàng và tối ưu và tiết kiệm nhất.

3. Đối tượng và phạm vi

a. Đối tượng nghiên cứu

Đối tượng nghiên cứu ở đâu đó là các địa điểm du lịch, khách sạn, quán ăn và sự cần thiết của con người đến chúng.

b. Phạm vi nghiên cứu

Phạm vi tại các địa điểm cụ thể rồi mở rộng ra cả thành phố Hà Nội, cả nước.

1. Phương pháp nghiên cứu

Phương pháp khảo sát địa điểm Phương pháp tổng hợp dữ liệu, thông tin Dùng các công nghệ để xây dựng app mobile, ngôn ngữ Dart, framework Flutter, SQL server, GPS, ASP.NET Core.

**Chương 1: Cơ sở lý thuyết**

* 1. Đặt vấn đề



Hình 1.1 Mạng lưới du lịch

Ở nhiều Quốc gia du lịch đóng góp một phần đáng kể trong tổng thu nhập hàng năm, đặc biệt tại Việt Nam du lịch được đánh giá là một trong 3 ngành kinh tế mũi nhọn được nhà nước chú trọng đầu tư cơ sở hạ tầng, không ngừng phát triển và đóng góp rất lớn và nền kinh tế đất nước. Nhưng thực tế người du lịch vẫn bị thụ động hoặc phải mất nhiều thời gian để tìm kiếm những địa điểm du lịch và dịch vụ đi kèm, thay vì phải bỏ ra một số tiền lớn để thuê những hướng dẫn viên du lịch thì với chuyên ngành công nghệ thông tin em có ý tưởng nhằm thực tế hóa nhu cầu của khách du lịch giúp họ tiếp cận một cách dễ dàng hơn. Em lựa chọn chủ đề cho đồ án tốt nghiệp là :“Xây dựng ứng dụng mobile hỗ trợ tìm kiếm các địa điểm du lịch”.

Phạm vi đồ án:

Ứng dụng mobile hỗ trợ tìm kiếm các địa điểm du lịch có chức năng sau:

* Quản trị viên: Thêm, sửa, xóa, cập nhật liên tục, kiểm soát các địa điểm du lịch và dịch vụ đi kèm. Phạm vi áp dụng : các địa điểm du lịch tại Hà Nội sau đó sẽ cập nhật dần đến những phạm vi rộng hơn. Phương pháp nghiên cứu và hướng giải quyết: Bằng phương pháp tìm hiểu lý thuyết, thực nghiệm, tìm hiểu tình hình thực tế tại các địa điểm du lịch ở Hà Nội và các địa điểm quán ăn, khách sạn, nhà nghỉ gần đó để hình thành luồng nghiệp vụ cho ứng dụng và áp dụng các kiến thức đẫ được giảng dạu cho phù hợp với yêu cầu đã đặt ra.
* Người dùng: được chia làm hai loại, thứ nhất là người đi du lịch chỉ có nhu cầu xem và đánh giá các địa điểm, dịch vụ. Thứ hai là các chủ sở hữu dịch vụ đi kèm xung quanh các địa điểm du lịch ( nhà nghỉ, quán ăn, khách sạn,…)
  + Người dùng thứ nhất: được phép xem, viết đánh giá, sửa đánh giá, xóa đánh giá các địa điểm du lịch.
  + Người dùng thứ hai: được phép thêm, sửa, xóa, viết giới thiệu các địa điểm của họ. Ngoài ra họ hoàn toàn có quyền của người dùng thứ nhất đó là xem và viết, sửa, xóa đánh giá các địa điểm du lịch.

1.2. Giới thiệu về ASP.NET CORE

1.2.1. Tổng quan về ASP.NET CORE

ASP.NET Core được xem là tập hợp các thư viện chuẩn như một Framework sử dụng để xây dựng các ứng dụng web. Đây không phải là phiên bản tiếp nối của ASP.NET, ASP.NET Core là cái tên được xây dựng và phát triển mới mẻ lại từ đầu để trở nên linh hoạt, nhanh chóng và chạy trên nhiều nền tảng khác nhau. ASP.NET Core có sự thay đổi lớn về kiến trúc, mang đến kết quả ngắn gọn, phân chia Module tốt hơn rất nhiều.

Định nghĩa “ASP.NET Core” : “ASP.NET Core là một khuôn khổ web mã nguồn mở miễn phí được kế thừa từ ASP.NET và phát triển bởi Microsoft. Đây là một mô-đun chạy trên cả .NET Framework đầy đủ và .NET Core đa nền tảng. Tuy nhiên, ASP.NET Core phiên bản 3 chỉ có hiệu lực trên .NET Core và không có tính năng hỗ trợ .NET Framework”.



Hình 1. 2 Tìm hiểu về ASP.NET Core

Vào khoảng 16 năm trước, bản phát hành đầu tiên của ASP.NET được ra đời và trở thành một phần của .NET Framework. Kể từ đó, hàng triệu lập trình viên ưa chuộng và sử dụng nó để xây dựng các ứng dụng web tuyệt vời và Microsoft trong những năm đó cũng đã phát triển, cải tiến thêm nhiều tính năng mới.

Sau nhiều năm sử dụng ASP.NET để phát triển web, Framework có những thay đổi tiến hoá đều đặn và mang đến cho chúng ta một hậu duệ tuyệt hảo, mới mẻ là ASP.NET Core.

ASP.NET Core được biết đến với một số thay đổi mới mẻ lớn về kiến trúc, đó là kết quả từ việc học hỏi rất nhiều từ các Framework Module hóa khác. Hiện nay, ASP.NET Core không còn phụ thuộc cố định vào System.Web.dll nữa mà tập hợp các gói, các module hay còn được gọi là các Nuget Packages.

Đặc điểm của ASP.NET Core:

Có thể thấy, ASP.NET Framework tồn tại bền bỉ trên hệ thống Windows từ lâu thì ASP.NET Core mới bắt đầu ra đời để hỗ trợ các tính năng cho .NET Framework như Windows Forms, WPF. ASP.NET Core có thể hoạt động trên bất kỳ nền tảng nào mà đảm bảo có thể khắc phục được những hạn chế của .NET Framework – chạỵ duy nhất trên nền tảng Windows. ASP.NET Core được thiết kế để tối ưu và cung cấp tốt cho các Development Framework triển khai trên đám mây hoặc chạy on-promise.

Graphical user interface, website

Description automatically generated

Hình 1. 3 ASP.NET Core in a Nutshell

Về cơ bản, ASP.NET Core được tạo thành từ một thư viện bản Net. .NET Standard 2.0 có thể chạy ở bất cơ nào mà nó có thể hỗ trợ được. Đồng thời, ASP.NET Core trở thành một mã nguồn mở. Đây là sự thay đổi lớn và vô cùng quan trọng mà trước đây khó có lập trình viên nào nghĩ đến. Chính vì thế, ASP.NET Core thu hút đông đảo các lập trình viên sử dụng để xây dựng các trang web.

ASP.NET Core là một phiên bản thiết kế của ASP.NET 4.x có sự thay đổi kiến trúc giúp Framework nhẹ nhàng hơn và kết hợp cả tính Module nhiều hơn nữa. Với một nền tảng có hiệu suất và tính tương thích hiệu quả, các lập trình viên vẫn trung thành xây dựng các web bằng ASP.NET Core.

1.2.2. Ứng dụng của ASP.NET CORE

- Xây dựng web UI và web API sửng dụng ASP.NET Core MVC

- Phát triển client-side

1.2.3. Ưu điểm của ASP.NET CORE

- Hỗ trợ đa nền tảng: các ứng dụng ASP.NET Core có thể chạy trên Windows, Linux, Mac. Vì vậy bạn khoog cần phải xây dựng các ứng dụng khác nhau cho các nền tải khác nhau bằng các framework khác nhau

- Nhanh: ASP.NET Core không còn phụ thuộc vào System.web.dll cho giao tiếp giữa máy chủ và trình duyệt. ASP.NET Core cho phát them các gói mà chúng ta cần cho ứng dụng của mình. Điều này làm giảm đường ống yêu cầu và cải thiện năng xuất mở rộng.

- IoC Container: tích hợp sẵn IoC Container tích hợp để xử lý phụ thuộc tự động giúp nó có thể duy trì và kiểm tra được.

Tích hợp với các framework UI hiện đại: nó cho phép bạn sử dụng và quản lý các framework UI hiện đại như AngularJS, ReactJS, Umber, Bootstrap, … bằng cách sử dụng Bower (trình quản lý gói cho web).

- Lưu trữ: ững dụng web ASP.NET Core có thể lưu trữ trên nhiều nền tảng với bất kỳ máy chủ nào như IIS, Apahe… Nó không chỉ phụ thuộc vào IIS như một .NET Framework tiêu chuẩn.

- Chia sẻ mã: Nó cho phép bạn xây dựn một thư viện lớp có thể sử dụng với các khung .NET khác nhau như .NET framework 4.x hoặc Mono. Do đó, một cơ sở mã duy nhất có thể được chia sẻ trên các khung.

- Kích thước triển khai nhỏ hơn: Ứng dụng ASP.NET Core chạy trên .NET Core nhỏ hơn .NET Framework đầy đủ. Vì vậy, ứng dugnfj chỉ sử dụn một phần của .NET CoreFX sẽ có kích thước triển khai nhỏ hơn. Điều này làm giảm kích thức triển khai.

1.2.4 Nhược điểm của ASP.NET CORE

Ngoài những ưu điểm nổi trội mang đầy tính thuyết phục như đã kể trên thì hiện tại, ASP.NET cũng có một số nhược điểm tương đối lớn, chưa thể khắc phục được như: không hỗ trợ cho các thiết bị sử dụng hệ điều hành Linux, không hỗ trợ Visual studio trong quá trình viết code. Bên cạnh đó, do có khá nhiều ưu điểm hấp dẫn như vậy nên ASP.NET có phí sử dụng khá cao, không phù hợp sử dụng cho các doanh nghiệp nhỏ hay các cá nhân.

1.3 Giới thiệu về SQL Server

SQL Server hoặc Microsoft SQL Server là phần mềm được tạo bởi Microsoft. Nó là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS), mang đến công cụ quản lý (giao diện GUI, ngôn ngữ truy vấn SQL,…). Nó có nhiều nền tảng bao gồm: ASP.NET, C# xây dựng Winform vì nó hoạt động hoàn toàn độc lập.

1.3.1 Ưu điểm:

* Có thể cài đặt nhiều phiên bản Microsoft SQL cho một máy chủ.
* Duy trì riêng biệt các môi trường (sản xuất, phát triển và thử nghiệm).
* Giảm vấn đề tạm thời trên cơ sở dữ liệu.
* Tách biệt những đặc quyền bảo mật.
* Duy trì máy chủ dự phòng.

1.3.2 Nhược điểm:

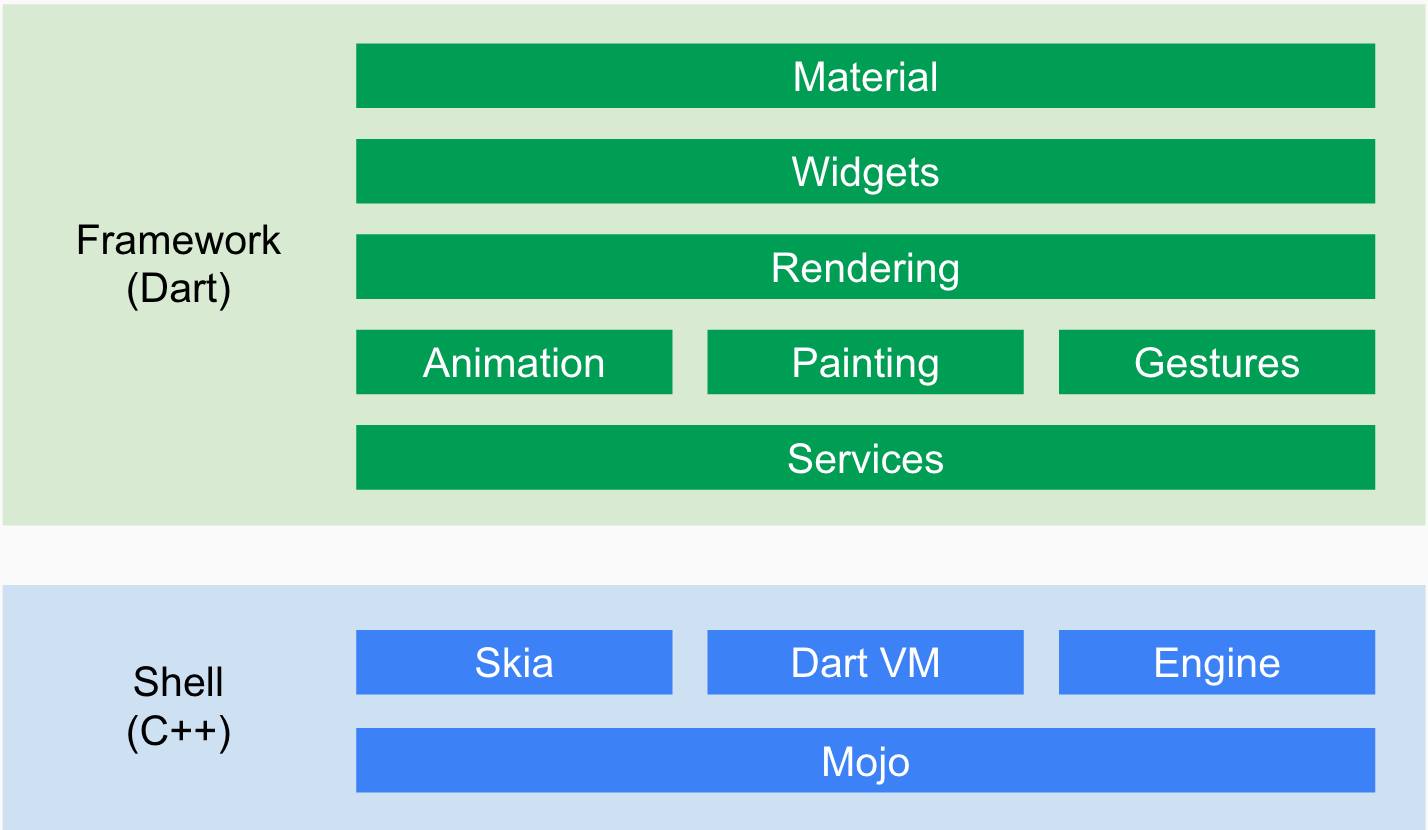
* SQL Server chỉ có thể hoạt động trên hệ điều hành Windows.
* Phải thanh toán phí thường niên nếu bạn muốn chạy nhiều CSDL (database).

Hiện nay, SQL ngày càng đóng vai trò hết sức quan trọng và nó thường được sử dụng để tạo website động. SQL có thể được dùng như sự kết hợp cả hai cơ sở dữ liệu và website. Trong cơ sở dữ liệu trên máy chủ SQL sẽ truy cập thông tin, kết quả hiển thị trên website nếu như người dùng yêu cầu. Thậm chí, SQL còn là công cụ cập nhật thông tin cho CSDL. Tuy nhiên, SQL lại rất dễ sử dụng, không đòi hỏi bạn phải biết về lập trình hay là chuyên gia tin học.

1.4 Giới thiệu về Flutter

**Flutter** được phát triển nhằm giải quyết bài toán thường gặp trong mobile là Fast Development và Native Performance. Nếu như React Native chỉ đảm bảo Fast Development và code native thuần chỉ đảm bảo Native Performance thì Flutter làm được cả 2 điều trên.

Sử dụng DART. Một ngôn ngữ lập trình hướng đối tượng do Google phát triển. DART là một static type language nên nó là AOT (Ahead of Time), compile xong hết rồi mới chạy. Trong khi đó nó cũng là JIT (Just in Time) giống như các dynamic type language. Khi dev thì nó sử dụng JIT để hỗ trợ Hot Load và build release thì dùng AOT để tối ưu hiệu năng như một native code bình thường. Lại magic nữa.



Hình 1.4 Ngôn ngữ sử dụng của Flutter

Ngoài ra DART cũng hướng tới việc trở thành một ngôn ngữ có thể chạy trên nhiều platform khác nhau, nó cũng có máy ảo (VM) làm nhiệm vụ dịch source code sang bytecode như Java. Hiện tại DART có thể transpile ra các ngôn ngữ khác như JS để chạy trên trình duyệt. Cái này không liên quan mobile lắm tuy nhiên đây cũng là 1 ưu điểm lớn của ngôn ngữ này nên không thể bỏ qua.

Vậy còn native module ? Khác với JS Bridge, Flutter “nói chuyện” với các native module bằng chính các native interface. Mặc dù vẫn được gọi là “bridge”, tuy nhiên nó nhanh hơn rất nhiều và gân như không bị “thắc cổ chai” như React Native. Ngoài ra các module này được kiến trúc theo “plugin”, các module viết cho Flutter phải tuân thủ các rule trong này. Cá nhân mình thấy viết native module cho Flutter rất tự nhiên, không cần phải học các syntax macro C/C++ như RN.

Theo doc của Flutter Engine, có tới 4 threads (runners) được sử dụng trong app: Platform Task Runner, UI Task Runner, GPU Task Runner và IO Task Runner. Các threads này độc lập và không share memory với nhau, chúng giao tiếp với nhau thông qua channels… tới đây ai fan golang sẽ hiểu và rất thích pattern này.

Về document: sở hữu bộ doc phải nói là không bỏ sót thứ gì. Đi từ cài đặt, hướng dẫn viết app cơ bản cho tới CI/CD, [debug](https://topdev.vn/blog/?s=debug), test và profiling. Bộ profiling của Flutter cũng cực kì hay dùng để đo đạc các chỉ số về performance khá chi tiết.

Các ví dụ và kiến trúc ứng dụng: Có hẳn 1 repository trên Github đủ hết các example cho Redux, mvc, mvu…

1.4.1 Ưu điểm của Flutter

Phát triển ứng dụng nhanh chóng: Tính năng “hot reload” của nó giúp bạn nhanh chóng và dễ dàng thử nghiệm, xây dựng giao diện người dùng, thêm tính năng và sửa lỗi nhanh hơn. Trải nghiệm tải lại lần thứ hai, mà không làm mất trạng thái, trên emulator (giả lập), simulator (mô phỏng) và device (thiết bị) cho iOS và Android.

UI (User Interface – giao diện người dùng) đẹp và biểu cảm: Thỏa mãn người dùng của bạn với các tiện ích (widget) đẹp mắt theo thiết kế thô (Material Design) và Cupertino (iOS-flavor), các API chuyển động phong phú, cuộn tự nhiên mượt mà và tự nhận thức được nền tảng.

Truy cập các tính năng và SDK native: Làm cho ứng dụng của bạn trở nên sống động với API của nền tảng, SDK của bên thứ ba và native code. Nó cho phép bạn sử dụng lại mã Java, Swift và ObjC hiện tại của mình và truy cập các tính năng và SDK native trên iOS và Android.

Phát triển ứng dụng thống nhất: Flutter có các công cụ và thư viện để giúp bạn dễ dàng đưa ý tưởng của mình vào cuộc sống trên iOS và Android. Nếu bạn chưa có kinh nghiệm phát triển trên thiết bị di động, thì Flutter là một cách dễ dàng và nhanh chóng để xây dựng các ứng dụng di động tuyệt đẹp. Nếu bạn là một nhà phát triển iOS hoặc Android có kinh nghiệm, có thể sử dụng Flutter cho các view (MVC) và tận dụng nhiều code Java / Kotlin / ObjC / Swift hiện có.

1.4.2 Nhược điểm của Flutter

Mặc dù có nhiều ưu điểm, song qua thử nghiệm, Flutter cũng có những yếu điểm nhất định.

Flutter vẫn chưa thật hoàn thiện. Vì là một ứng dụng mới nên Flutter vẫn chưa đạt mức hoàn hảo. Thực tế, nhiều tính năng nâng cao của Flutter vẫn chưa được hỗ trợ hơn thế nữa, nhiều thư viện chưa được thử nghiệm chính thức còn tồn tại hạn chế khi so sánh với các bản sao gốc (như là Google Maps).

Dart khá "non nớt". Về cơ bản Dart khá giống với Swift và Kotlin, nhưng có ít tính năng hơn, hoặc những tính năng hiện có chưa được toàn diện. Các ứng dụng Flutter khá "nặng". Chúng chiếm nhiều dung lượng và mất nhiều thời gian để tải xuống hoặc cập nhật. Giao diện không giống 100% so với phiên bản gốc.

Về cơ bản, Flutter không tạo ra các thành phần gốc mà sao chép không hoàn toàn các thiết kế Material Design của Android và các thành phần riêng của iOS bằng thư viện Cupertino. Thư viện này sẽ hiển thị, đặc biệt với các phiên bản hệ thống chứa các trường văn bản hoặc các nút - những thành phần biến đổi bên ngoài nhưng không thay đổi bên trong Flutter.

Hướng dẫn phát triển ứng dụng Flutter chưa được đồng nhất, điều này có thể gây nhiều khó khăn khi xây dựng các phần mềm mang tính phức tạp. Framework thay đổi nhanh chóng gây khó khăn cho việc duy trì mã.