

TÓM TẮT LUẬN VĂN TỐT NGHIỆP

Đề tài: Tìm hiểu đánh giá các framework phát triển ứng dụng di động đa nền tảng.

Hiện nay theo thống kê thế giới có khoảng hơn 2.3 tỉ điện thoại thông minh với hai hệ điều hành phổ biến nhất là iOS và Android (hoặc các hệ điều hành tùy biến từ hệ điều hành Android). Doanh thu từ ứng dụng trên hai hệ điều hành iOS và Android xấp xỉ 61 tỉ USD. Vì vậy, hệ sinh thái di động trở thành thị trường màu mỡ mà không một công ty hay các nhà phát triển nào muốn bỏ qua. Các doanh nghiệp hay các lập trình viên đều muốn đưa sản phẩm của mình đến càng nhiều người dùng càng tốt, trong khi đó vẫn đảm bảo chất lượng của sản phẩm mà không tốn quá nhiều chi phí trong việc phát triển. Khi các lập trình viên có hứng thú với các nền tảng di động, muốn tạo nên các ứng dụng tuyệt vời trên các thiết bị di động, chắc chắn sẽ có người phân vân trong việc lựa chọn nền tảng công nghệ mà họ nên lựa chọn để đáp ứng nhu cầu của bản thân. Vì lý do đó, khoá luận sẽ đưa ra một cái nhìn tổng quan và chi tiết về các công nghệ phát triển ứng dụng trên nền tảng di động. Bên cạnh đó, khoá luận cũng đưa ra các so sánh về các cách tiếp cận phát triển ứng dụng di động khác nhau dựa trên các tiêu chí để các nhà phát triển có thể lựa chọn phương án phù hợp.

Mục tiêu chính của khoá luận là so sánh các phương pháp phát triển ứng dụng di động dựa trên các tiêu chí đánh giá cần thiết hiện nay để có thể đưa một ứng dụng di động thành công ra cộng đồng.

:

Luận văn cung cấp thông tin về các cách phát triển ứng dụng di động tập trung vào hai bộ khung phát triển ứng dụng là Ionic và Xamarin. Đầu tiên sẽ giới thiệu các cách phát triển ứng dụng di động được sử dụng hiện nay. Sau đó, khóa luận sẽ lần lượt giới thiệu từng bộ khung phát triển Ionic và Xamarin về các tính năng, kiến trúc, các đặc điểm nổi bật cũng như ưu điểm và nhược điểm của chúng. Tiếp theo nghiên cứu sẽ đưa ra bảng so sánh khả năng của từng bộ khung phát triển khi so sánh với việc phát triển ứng dụng native. Sau đó sẽ đi sâu hơn vào các tiêu chí mà các nhà phát triển nên quan tâm để lựa chọn bộ khung phát triển phù hợp. Cuối cùng dựa vào các tiêu chí so sánh, luận văn sẽ đưa ra một số khuyến nghị đến các nhà phát triển trong việc lựa chọn bộ khung phát triển phù hợp với nhu cầu

Hiện nay có khá nhiều bộ khung phát triển ứng dụng đa nền tảng như Cordova, PhoneGap, Ionic, Xamarin, C++ Builder,... Nhưng tựu chung lại thì có hai trường phái chính trong việc xây dựng các bộ khung phát triển ứng dụng đa nền tảng web và native. Trong đó thì Ionic và Xamarin là hai nền tảng phổ biến. Ionic sử dụng công nghệ web trong khi Xamarin sử dụng công nghệ native.

Ionic là một bộ khung phát triển ứng dụng di động mã nguồn mở được ra đời vào năm 2013, được phát triển bởi công ty Drifty. Được lấy cảm hứng từ sự thành công của nền tảng web trên desktop, các nhà phát triển mong muốn Ionic cũng đạt được thành công tương tự trên các nền tảng di động.

Được xây dựng dựa trên nền tảng AngularJS và Apache Cordova (mặc định), Ionic cung cấp công cụ để phát triển các ứng dụng hybrid trên các hệ điều hành di động sử dụng các công nghệ Web như CSS, HTML5, SASS và Javascript. Ứng dụng được xây dựng dựa vào các công nghệ web như trên và phân phối thông qua các chợ ứng dụng mặc định trên các thiết bị nhờ sự trợ giúp của một trình đóng gói bản địa (Native wrapper).

Ionic có thể được xem như một bộ khung phát triển giao diện cho front-end. Nó sẽ chịu trách nhiệm về giao diện và cách người dùng tương tác với ứng dụng. Bên cạnh đó, Ionic còn hỗ trợ rất nhiều các thành phần native trong các nền tảng di động, cung cấp sẵn một số animation có sẵn. Có một điểm khác biệt của Ionic với các framework khác là các thành phần giao diện trong Ionic có vẻ ngoài và cách hoạt động rất giống với các thành phần giao diện tương tự trong các hệ điều hành di động và tất nhiên điều này diễn ra hoàn toàn tự động, nhà phát triển không cần phải chỉnh sửa bất cứ thứ gì. Thêm nữa, Ionic có một số đặc điểm nổi bật như mã nguồn mở, hiệu suất tương đối tốt, sử dụng AngularJS và Cordova để hỗ trợ trong việc phát triển ứng dụng, sử dụng kiến trúc plugin để hỗ trợ các lập trình viên tích hợp các tính năng vào ứng dụng. Tuy nhiên, Ionic cũng có một số nhược điểm như đối với các ứng dụng yêu cầu cao về hiệu suất thì Ionic chưa đáp ứng được, phụ thuộc nhiều vào các plugin, không có sự cam kết chất lượng giữa đơn vị chủ quản và người sử dụng, việc tùy biến tương đối khó khăn do yêu cầu cao về mặt kỹ năng.

Bộ khung phát triển ứng dụng đa nền tảng Xamarin, tiền thân là dự án mã nguồn mở Mono, được giới thiệu vào ngày 16/5/2011. Xamarin ban đầu thuộc công ty Xamarin, tuy nhiên đã được tập đoàn Microsoft mua lại vào 24/2/2016 và được chuyển đổi thành phần mềm mã nguồn mở sau đó.

Xamarin là bộ khung phát triển ứng dụng cho phép các nhà phát triển xây dựng ứng dụng trên các nền tảng Android, iOS và Windows sử dụng một ngôn ngữ lập trình chính là C#. Xamarin cung cấp các lớp thư viện, runtime thực thi trên cả ba nền tảng iOS, Android và Windows Phone, trong khi vẫn biên dịch native (không sử dụng các trình thông dịch) và đảm bảo hiệu suất ứng dụng kể cả các ứng dụng yêu cầu khả năng xử lý nặng như các ứng dụng trò chơi. Mặc dù không sử dụng các ngôn ngữ lập trình tương ứng trên các hệ điều hành khác nhau nhưng Xamarin được xem như là một bộ khung phát triển ứng dụng gốc (native framework). Xamarin có một số đặc điểm nổi bật đáng chú ý như việc liên kết tốt với các thư viện native, tương tác với các ngôn ngữ lập trình native trên các nền tảng, hiệu suất ứng dụng tương đương với các ứng dụng native, tận dụng được số lượng lớn các thư viện .NET có sẵn. Tuy nhiên Xamarin cũng có một số nhược điểm như tồn tại một số lỗi, hạn chế kỹ thuật do kiến trúc của Xamarin, sự phình to của ứng dụng khi tích hợp các thư viện so với ứng dụng thông thường.

Để lựa chọn một bộ công cụ để phát triển một ứng dụng di động, lập trình viên cần quan tâm đến một số các điểm quan trọng như các thành phần giao diện, trải nghiệm người dùng, việc thiết kế layout để tương thích với nhiều

thiết bị với nhiều độ phân giải và kích cỡ, cộng đồng sử dụng, sự hỗ trợ các dịch vụ của bên thứ ba, hỗ trợ đa luồng, kiểm thử.

- Về giao diện và trải nghiệm người dùng, cả Ionic và Xamarin đều cung cấp một số lượng các thành phần giao diện có sẵn có khả năng tự động thay đổi hình thái tương ứng với từng nền tảng cụ thể. Tuy nhiên, Xamarin hỗ trợ tốt việc tùy biến hành vi và vẻ ngoài của các thành phần có sẵn hơn Ionic nhờ việc cung cấp sẵn tài liệu hướng dẫn người sử dụng. Đối với các animation, thì Ionic và Xamarin đều hỗ trợ một số các animation có sẵn, cho phép nhóm các animation vào thành chuỗi các animation. Ionic bắt buộc phải sử dụng các animation sử dụng GPU để đảm bảo hiệu suất ứng dụng
- Về layout, cả hai bộ khung phát triển đều cung cấp các giải pháp cho phép các lập trình viên thiết kế giao diện một cách linh động để hiển thị tốt trên nhiều thiết bị khác nhau. Tuy nhiên, việc thiết kế bằng cách kéo thả thì Ionic khá hạn chế nếu như lập trình viên chỉ sử dụng phiên bản miễn phí của trình thiết kế giao diện kéo thả đi kèm, trong khi Xamarin hỗ trợ rất tốt với IDE mặc định
- Về cộng đồng phát triển, Xamarin có vẻ vượt trội hơn Ionic khi một kết quả so sánh cho thấy có hơn 40000 kết quả gắn với tag 'Xamarin' so với 20000 kết quả gắn tag 'Ionic' trên trang hỏi đáp nổi tiếng nhất trong giới lập trình viên StackOverFlow
- Hiện nay các lập trình viên thường tích hợp rất nhiều các dịch vụ của các bên thứ ba cung cấp như các dịch vụ quảng cáo, các dịch vụ phân tích hay

báo lỗi để hỗ trợ việc phát triển. Ionic và Xamarin đều hỗ trợ một số dịch vụ như Fabric, Admob, Facebook Audience Network, Mopub, Appmoeal. Trong đó, việc triển khai các dịch vụ được hỗ trợ sẵn của Ionic tương đối đơn giản hơn so với Xamarin khi các nhà phát triển không phải viết riêng từng đoạn mã xử lý cho từng nền tảng. Tuy nhiên, nếu như các lập trình viên muốn tích hợp các dịch vụ chưa được hỗ trợ sẵn thì Xamarin có lợi thế nhờ việc tương tác tốt với các thư viện native trong khi Ionic thì trừ khi các lập trình viên có kiến thức tốt về cả web và native trên từng nền tảng thì các lập trình viên bình thường sẽ bị phụ thuộc vào số ít các lập trình viên có kinh nghiệm xây dựng nên các plug in để tích hợp các dịch vụ đấy.

- Đối với một số ứng dụng đặc biệt như trò chơi, các nhà phát triển thường tìm đến các game engine để phát triển như cocos, SpriteKit, Unity, Unreal engine. Tuy nhiên các framework phát triển ứng dụng cơ bản như Ionic và Xamarin đều có thể phát triển các ứng dụng này ở một mức độ vừa phải nhờ việc kết hợp với các engine khác như Phaser.io kết hợp với Ionic, Cocos Sharp kết hợp với Xamarin.
- Số lượng người dùng của một ứng dụng di động có thể lên đến hàng tỉ người. Vì vậy, các ứng dụng di động hiện nay có thể phải xử lý rất nhiều dữ liệu để có thể cung cấp được thông tin cho người dùng. Điều này yêu cầu khả năng tính toán rất lớn trong khi các thiết bị di động có thể không đủ đáp ứng. Các nhà phát triển ứng dụng di động phải tìm cách để đảm bảo việc xử lý dữ liệu không ảnh hưởng gì

đến trải nghiệm liên tục của người dùng. Một trong các cách xử lý vấn đề này là chia một tác vụ lớn thành nhiều tác vụ nhỏ và chạy nó trên nhiều luồng khác nhau. Vì vậy, việc hỗ trợ đa luồng cũng rất quan trọng trong việc tìm kiếm một bộ khung phát triển phù hợp cho việc phát triển ứng dụng di động. Cả Xamarin và Ionic đều hỗ trợ đa luồng, tuy nhiên Ionic có một số điểm yếu trong việc hỗ trợ đa luồng do bản chất ngôn ngữ Javascript là một ngôn ngữ đơn luồng. Ionic sử dụng WebWorker để thực thi các tác vụ ở các luồng phụ, nhưng có một số hạn chế khi ví dụ như không thể truy cập đến các thành phần DOM từ trang web (vì nó không thread safe), không thể truy xuất các biến toàn cục và các hàm Javascript từ trang web, không thể gọi hàm alert hay confirm, các đối tượng như window, document và parent không thể truy cập trong web worker. Trong khi đó, Xamarin hỗ trợ rất tốt đa luồng nhờ khả năng có thể sử dụng các API .Net hoặc sử dụng các API native trong việc tạo, quản lý các luồng.

- Hiện nay, các lập trình viên tham gia rất nhiều vào quá trình kiểm thử ứng dụng, không còn chỉ phụ thuộc vào đội ngũ đảm bảo chất lượng. Họ sử dụng TDD, Unit Test, các công cụ kiểm thử tự động để thực thi các ca kiểm thử, đảm bảo ứng dụng hoạt động đúng theo mong muốn của họ. Cả Xamarin và Ionic đều có các thư viện hỗ trợ việc thực thi kiểm thử tự động như Karma và Jasmine hỗ trợ kiểm thử đơn vị, Protractor hỗ trợ kiểm thử chấp nhận trên Ionic; Nunit, MonoTouch.NunitLite hỗ trợ kiểm thử đơn vị trên Xamarin, Xamarin.UITest

hỗ trợ kiểm thử giao diện trên Xamarin. Một lần nữa Ionic lại tiếp tục thể hiện tính đa nền tảng nhiều hơn Xamarin ở tiêu chí này khi lập trình viên không phải viết lại các ca kiểm thử trên từng nền tảng như Xamarin.

Tổng kết lại, cả Ionic và Xamarin đều là những bộ khung phát triển ứng dụng di động đa nền tảng tốt và có những lợi thế nhất định của mình. Nếu như Xamarin mạnh về khả năng tương tác với các thư viện native, hiệu năng và khả năng tùy biến thì Ionic mạnh về tính nhất quán cần có của một bộ khung phát triển đa nền tảng. Việc lựa chọn bộ khung phát triển phù hợp phụ thuộc vào yêu cầu của ứng dụng và nền tảng công nghệ của lập trình viên. Nếu như ứng dụng yêu cầu không quá phức tạp, không cần xử lý quá nhiều dữ liệu, chỉ sử dụng các dịch vụ phổ biến, cần có thời gian phát triển nhanh, chi phí hạn chế thì Ionic là một lựa chọn rất tốt. Trong khi đó nếu như ứng dụng yêu cầu hiệu năng tốt, cần xử lý nhiều tác vụ nặng, cần tùy biến nhiều trên từng nền tảng hoặc sử dụng nhiều các thư viện native thì Xamarin là lựa chọn tốt hơn.