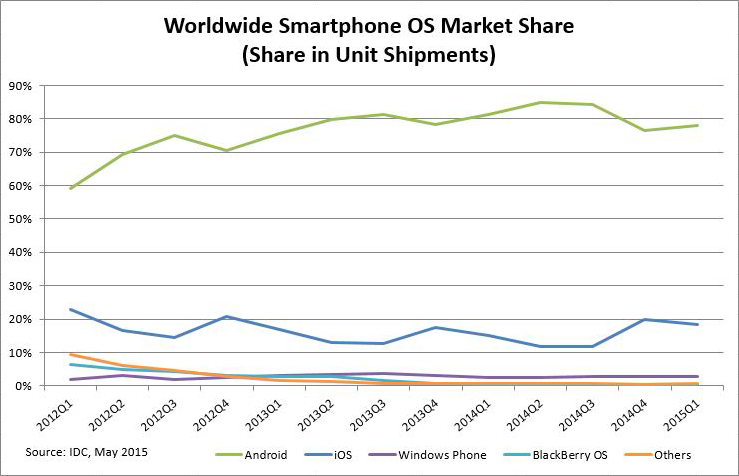
CHƯƠNG 1 : TỔNG QUAN

* 1. Tổng quan về vấn đề được nghiên cứu

Trong khoảng thời gian gần đây, lập trình di động đang là một ngành hot. Các mẩu tin tuyển dụng gần nhất mình đọc thường tuyển Android developer, iOS developer, … với mức lương khá cao, không thua kém gì [lập trình web](https://toidicodedao.com/2015/08/18/nhung-ki-nang-can-co-cua-mot-web-developer/) hay lập trình hệ thống nhúng. Ngoài ra, nếu biết cách lập trình ứng dụng, bạn cũng có thể làm freelance, hoặc tự phát triển ứng dụng và kiếm tiền thông qua ứng dụng của mình.

**Thế chân vạc trong trận chiến Mobile**

Trên thị trường ứng dụng di động hiện nay, 3 hệ điều hành chiếm thị phần cao nhất là : Android, iOS và Window phone, tiếp sau là 1 số hệ điều hành khác như BlackBerry… Trong phạm vi bài viết, mình chỉ phân tích về 3 OS đứng đầu là**Android, iOS**và**Windows Phone** nhé.



**Android – Kẻ chiếm miếng bánh lớn nhất**

Theo biểu đồ, ta dễ nhận ra Android luôn **chiếm hơn 70% thị phần** của mảng di động. Ứng dụng Android được viết bằng **ngôn ngữ Java**, do đó các bạn lập trình viên Java có thể dễ dàng chuyển hướng qua mảng này.

Lập trình viên Android cũng đang là mục tiêu được các công ty săn đón. Các mẩu tin tuyển dụng Android developer **chiếm tỉ trọng lớn nhất** trong số các tin tuyển dụng của mảng mobile. Thuở còn làm đồ án tốt nghiệp, có 1 ông trong nhóm mình chưa biết gì về Android. Mình và ổng tự học và làm 2 tháng thực hiện đồ án, vừa xong đồ án thì ổng đi PV lập trình Android trong 1 công ty và được nhận luôn.

Android có quá nhiều device với đủ kích cỡ màn hình, cùng với vô số phiên bản (Từ 2.0 cho tới 4.4). Điều này gây khá nhiều khó khăn cho lập trình viên khi viết app : Cần phải test đủ thứ, đảm bảo ứng dụng tương thích với nhiều device, không bị lỗi giao diện, v…v.

Nếu bạn muốn đi theo con đường viết ứng dụng kiếm tiền, đưa ứng dụng lên Google Store, bạn sẽ phải mua 1 tài khoản Android Developer. Phí tài khoản này là **25$/năm**.



**iOS – Vị vua không ngai**

Theo biểu đồ, iOS chỉ chiếm **20% thị phần, bằng 1/4 so với Android**. Tuy nhiên nghe đồn là doanh thu của Apple Store lại cao hơn Google Play Store. Nguyên nhân là do người dùng iOS chơi sang hơn, chịu khó bỏ tiền mua ứng dụng hơn so với người dùng Android.

Số lượng tuyển dụng iOS ít hơn Android, tuy nhiên lương cho lập trình viên iOS lại **nhỉnh hơn bên Android chút đỉnh**. Lý do không phải vì iOS tốt hơn Android, mà chỉ đơn thuần là qui luật cung cầu: Lập trình viên iOS hiếm hơn lập trình viên Android nên họ có giá cao hơn.

Để tiếp cận iOS, bạn cần máy ảo hoặc máy Mac để cài hệ điều hành MacOS. Ứng dụng iOS được viết bằng ngôn ngữ Objective-C (Giống C nhưng có thêm OOP) hoặc Swift. Việc code và debug trên iOS phức tạp hơn Android. Bạn phải cài đặt Xcode, **mua tài khoản Apple Developer** mới có thể test ứng dụng và đưa ứng dụng lên Apple Store. Bộ phận kiểm duyệt của Apple Store cũng khắt khe hơn Google Play Store, nhiều khi bạn phải chờ khá lâu để ứng dụng của mình được duyệt.

Nếu làm ở công ty, bạn sẽ được cũng cấp tài khoản Apple Developer cũng như device để test. Nếu muốn tự viết, bạn sẽ phải tự trả **100$/năm** cho tài khoản Apple Developer, và mất thêm 1 khoản kha khá để mua thiết bị (iPhone, iPad) về test.



**Windows Phone – Kẻ sinh sau đẻ muộn**

Windows Phone đã chậm chân khi gia nhập thị trường di động, nơi Android và iOS đã làm mưa làm gió khá lâu. Mặc dù Microsoft đã có một số chính sách hỗ trợ devloper, hệ thống ứng dụng trên Window App Store vẫn còn khá **nghèo nàn và nhàm chán** (Mình tìm app Google Map mà còn không có).

Thú thật, mình chả thấy công ty nào tuyển lập trình viên Windows Phone cả. Hầu như các công ty đều o bế cho ứng dụng trên Android, iOS trước rồi mới đến Windows Phone. Vì Windows Phone được viết bằng [**ngôn ngữ C#**](https://toidicodedao.com/tag/c-hay-ho/) kết hợp với XAML, các lập trình viên C# có thể thử sức ở mảng này.

Cá nhân mình từng code cả Android lẫn Windows Phone thì thấy Windows Phone dễ code hơn, debug nhanh và tiện hơn. Với Android, nếu không có device, ta phải debug trên máy ảo, chạy rất chậm… máy ảo của Window Phone lại *rất mượt và nhanh*.

Nếu muốn viết app kiếm tiền, mình nghỉ các bạn nên thử Window Phone, vì những lý do sau (Nhớ cầu trời sau này Window App Store có nhiều người dùng hơn nhé, hiện tại ít người dùng nên chắc khả năng thu tiền lại cũng không cao đâu):

* Apple Store và Play Store đã có rất nhiều ứng dụng, tính cạnh tranh rất cao. Ngược lại, bạn ít khi gặp phải sự cạnh tranh trên Window Store.
* Microsoft đưa ra khá nhiều chính sách hỗ trợ Windows Phone, có thể trong tương lai sẽ thu hút nhiều người dùng hơn.
* Account Window Phone Developer có giá rất rẻ, chỉ có 19$ và **dùng mãi mãi.**



Hiện nay, có 3 hướng chính để phát triển một [ứng dụng di động](https://toidicodedao.com/2015/09/01/tong-quan-ve-lap-trinh-ung-dung-di-dong-phan-1/), đó là: **Web App, Native App và Hybrid App**. Mỗi hướng sẽ cần những [kĩ năng](https://toidicodedao.com/2015/08/18/nhung-ki-nang-can-co-cua-mot-web-developer/) riêng, có những ưu nhược điểm riêng, sẽ được nói rõ hơn bên dưới.



**Web App**

Hướng Mobile Web thường được áp dụng khi các bạn đã có sẵn một website đang hoạt động. Ta sẽ **tạo thêm 1 trang web riêng cho mobile**, sử dụng HTML, CSS, một số framework hỗ trợ mobile và responsive (Bootstrap, jQuery Mobile, Materialize). Người dùng sẽ trang web dành cho mobile để dùng ứng dụng.

Các xử lý khác liên quan đến [backend](https://toidicodedao.com/2015/08/18/nhung-ki-nang-can-co-cua-mot-web-developer/) như database sẽ được thực hiện phía trên server. Với một số [công nghệ](https://toidicodedao.com/2015/06/25/cach-tiep-can-1-ngon-ngucong-nghe-moi-phan-2/) như [AngularJS](https://toidicodedao.com/2015/07/07/tutorial-tao-ung-dung-chat-voi-50-dong-code-firebase-va-angularjs/), một trang web có thể **giống y hệt một ứng dụng di động thật sự**. Mình có viết 1 game đơn giản trên nền web, bạn hãy thử vào bằng di động, nó cũng khá giống 1 ứng dụng thật sự: <http://hoangphpetprojects.github.io/LadyboyChallenge/>

**Ưu điểm**

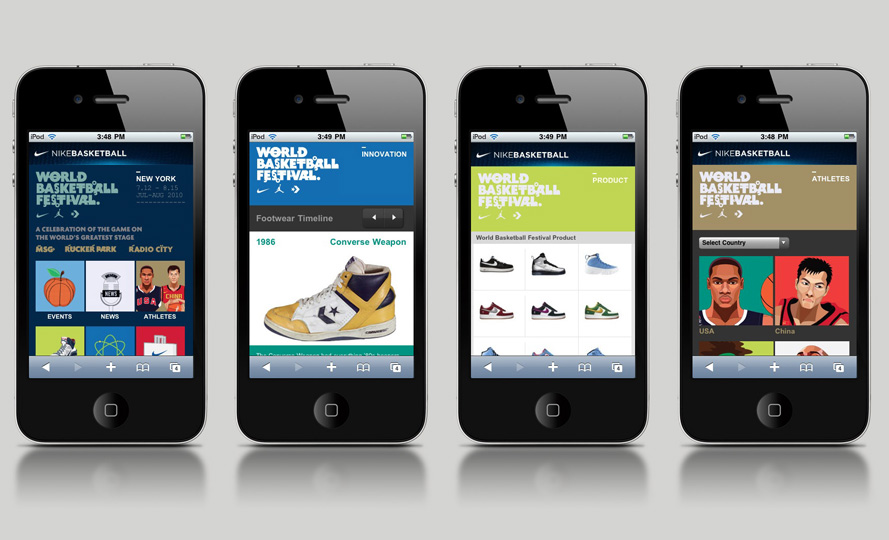
* Chỉ cần có kiến thức về web là viết được
* Viết một lần, chạy được trên mọi hệ điều hành
* Người dùng không cần phải cài app, có thể vào thẳng trang web
* Không cần phải thông qua App Store, **tiết kiệm tiền**
* Dễ nâng cấp (Chỉ việc nâng cấp web là xong)

**Nhược điểm**

* Với một số máy đời cũ, Web App sẽ bị bể giao diện, hiển thị sai, hoặc [javascript](https://toidicodedao.com/tag/javascript-sida/) không chạy.
* **Performance chậm**
* Không thể tận dụng được các tính năng của di động: Push notification, chụp hình, nghiêng máy, định vị GPS…

**Kĩ năng cần có**

* Kiến thức HTML, CSS, Javascript cơ bản.
* Kiến thức về một số framework responsive/mobile như: jQuery Mobile, Bootstrap, …
* Một số framework javascript để viết Single Page Application: [AngularJS](https://toidicodedao.com/2015/07/07/tutorial-tao-ung-dung-chat-voi-50-dong-code-firebase-va-angularjs/), EmberJS, …



**Native App**

Viết Native App nghĩa là lập trình viên**sẽ sử dụng IDE, SDK mà nhà sản xuất cung cấp** để lập trình ra một ứng dụng, build ứng dụng đó thành file cài và gửi lên App Store để kiểm duyệt. Người dùng sẽ phải tìm ứng dụng trên App Store, tải về máy và chạy.

Đây là hướng phát triển được áp dụng nhiều nhất, điển hình là game Flappy Bird của Nguyễn Hà Đông. Với những ứng dụng game, xử lý ảnh, cần tính toán nhiều, **Native App là lựa chọn duy nhất**.

Với những hệ thống lớn, cần đồng bộ, ta vẫn phải **viết phần back-end trên server**. Server sẽ đưa ra một số API. Native app lấy dữ liệu về máy, truyền dữ liệu lên server thông qua các API này.

**Ưu điểm**

* Tận dụng được**toàn bộ những tính năng của device**: Chụp ảnh, nghiêng máy, rung, GPS, notification.
* Có thể chạy được offline.
* Performance rất nhanh, vì code native sẽ được chạy trực tiếp.
* Là lựa chọn duy nhất cho các ứng dụng game, xử lý hình ảnh hay video …

**Khuyết điểm**

* Cần cài đặt nặng nề (Eclipse, XCode, Android SDK, …), khó tiếp cận.
* Với mỗi hệ điều hành, ta phải viết một ứng dụng riêng. **Khó đảm bảo sự đồng bộ giữa các ứng dụng** (1 button trên Android sẽ khác 1 button trên iOS, pop cũng khác).
* Cần phải**submit app lên App Store**, mỗi lần update phải thông báo người dùng.
* Code mệt và lâu hơn so với Mobile Web.

**Kĩ năng cần có**

* Ngôn ngữ lập trình: Java cho Android, Objective-C hoặc Swift cho iOS, [C#](https://toidicodedao.com/tag/c-hay-ho/) cho Windows Phone.
* Kiến thức chuyên sâu về ứng dụng: View, Action, Adapter trong Android …
* Cách xây dựng Web Serivce, Restful API, cách gọi API từ device, …



**Hybrid App**

Hybrid App **kết hợp những ưu điểm** của Mobile Web và Native App. Ta xây dựng một ứng dụng bằng HTML, CSS, Javascript, chạy trên WebView của mobile. Tuy nhiên, Hybrid App vẫn có thể tận dụng những tính năng của device: chụp hình, GPS, rung, ….

Hybrid App sẽ được viết dựa trên một **cross-platform framework**: Cordova, Phonegap, Titanium, …. Ta sẽ gọi những chức năng của mobile thông qua API mà framework này cung cấp, dưới dạng Javascript. Bạn chỉ cần viết một lần, những framework này sẽ tự động dịch ứng dụng này ra các file cài đặt cho Android, iOS và Windows Phone.

Một số ứng dụng **không quá nặng về xử lý**, cần tận dụng chức năng của device sẽ chọn hướng phát triển này. Đây là một game dạng [hybrid app](https://toidicodedao.com/2015/09/03/tong-quan-ve-lap-trinh-ung-dung-di-dong-phan-2/) mình viết bằng [ionic](https://toidicodedao.com/2016/01/12/tutorial-viet-ung-dung-di-dong-mot-cach-de-dang-voi-ionic-framework/)Platform, các bạn dùng Android có thể cài vào chơi thử nhé: <https://dl.dropboxusercontent.com/u/46157401/ladyboy-debug.apk>

**Ưu điểm**

* Chỉ cần biết HTML, CSS, JS (Thế nên mình mới khuyên các bạn nên học [Javascript](https://toidicodedao.com/2015/01/20/hoc-ngon-ngu-lap-trinh-nao-bay-gio-phan-cuoi/)).
* Viết một lần, chạy được trên nhiều hệ điều hành
* Tận dụng được các chức năng của device.

**Khuyết điểm**

* Không ổn định, **khó debug**. Framework sẽ dịch code của bạn thành code native, việc sửa lỗi ứng dụng khá khó vì bạn không biết code sẽ được dịch ra như thế nào.
* Performance chậm.
* Cần cài đặt nhiều thứ (Titanium, Cordova đều bắt phải cài đặt SDK này nọ thì mới build ứng dụng được).

**Kiến thức cần biết**

* HTML, CSS, Javscript cơ bản.
* Cách dùng một số framework CSS, Javascript: jQuery Mobile, Ionic Framework, AngularJS, Bootstrap, …
* Kiến thức về các **cross-platform framework**: Titanium, Cordova, Phonegap.
* Cách xây dựng Web Serivce, Restful API, cách gọi API từ device, … (Hybrid app cũng sẽ kết nối với server thông qua API như Native App).

