Team BAMM

Project Research

Team Members:

Mark Eames
Andrew Winters
Barry Walters
Mike (Absent)

Contents

Native Applications - Mark Eames

What Are They?

A native application is a software program that is developed for use on a particular platform or device.

The term native app is used to refer to platforms such as Mac and PC, with examples such as the Photos, Mail or Contacts applications that are pre installed and configured on every Apple computer. However, in the context of mobile web apps, the term native app is used to mean any application written to work on a specific device platform.

A native app is built for use on a particular device and its Operating System (OS), it has the ability to use device-specific hardware and software. Native apps can provide optimized performance and take advantage of the latest technology, such as a Global Positioning System (GPS), compared to web apps or mobile cloud apps developed to be generic across multiple systems.

The two main mobile OS platforms are Apple's iOS and Google's Android. Native apps are written in the code preliminarily used for the device and its OS. For example, developers write iOS applications in Objective-C or Swift, while they create Android-native apps in Java.

Native apps work on the device's operating system. In simpler terms, native apps require complete access to all the hardware and functionality of a device and live on a device. Apple provides the XCode, and Google provides Android Studio. These are specially designed integrated development environments (IDE) or software development kit (SDK), a software suite that comprises of a code editor, a compiler, and a debugger. The developers use this for writing and testing software. These increases the efficiency of the app development process by fixing the toughest bugs and reducing development time.

For example, the Facebook application was once written in HTML5 to use the same code for iOS, Android and mobile web. However, the app was slower for iOS users, leading Facebook's app developers to create separate code for iOS. Complex tasks can be rebalanced, such as networking done in the background of the main thread or program, which drives the User Interface (UI).
Source: https://searchsoftwarequality.techtarget.com/definition/native-application-native-app

The App Store was launched in 2008 with 500 apps available. In January Apple's App Store reached 2.2 million apps — last quarter the store grew by 60%.
Google also launched its Play Store (formerly known as Android Market) in 2008. It quickly outgrew Apple's store with 82% of growth, as of January, the total number of apps is over 2.7 million.
Source:https://medium.com/master-of-code-global/app-store-vs-google-play-stores-in-numbers-fd5ba020c195

Advantages and Disadvantages

Advantage

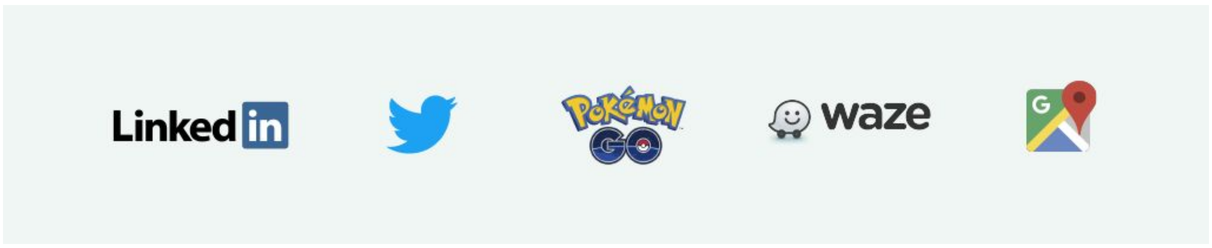
- broad functionalities due to using the capabilities of the underlying device;
- fast and responsive software performance;
- apps work even if there is no internet connectivity;
- push notifications;
- a UI that better matches with user experiences of the OS; and
- quality assurance though ratings in application stores.

Disadvantage

- multiple code bases because each device has its own version of the app;
- the cost for additional developers to build and manage a code base for each platform; and
- lengthy download times, and creating app accounts;
- frequent updates;
- time spent on multiple builds for separate platforms in each feature update.

In summary despite the fact that native apps are fast, smooth, blend in with the device's features perfectly, and can work offline, the challenges like high development cost and a more time-consuming development process makes it a hiccup for the business owners who have less budget and more time constraints.
Source: https://clutch.co/app-developers/resources/pros-cons-native-apps

Companies that use Native Applications



Some of the companies using Native platform development are shown above, they all use varying types of programming software for front and back end development.

How do they code them

- LinkedIn
  Front End - JavaScript
  Back End - Java, JavaScript, Scala
  Source:
  https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites

- Twitter
  Front End - JavaScript
  Back End - C++, Java, Scala, Ruby
  Source:
  https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites

- Pokemon
  Front End - ?
  Back End - Java, C++, C#
  Source:
  https://www.freelancinggig.com/blog/2017/02/28/programming-language-used-code-pokemon-go/

- Waze
  Israel application at origin, hard to find info

- Google Maps
  Language - Python, Java, C++
  Source:
  https://www.google.co.uk/search?ei=uShZXoGPBZij1fAP3vu66AQ&q=programming+language+used+for+google+maps&oq=programming+language+used+for+google+maps&gs_l=psy-ab.3..0.214979.223640..224642...1.2..0.149.1699.18j2......0....1..gws-wiz.......0i71j33i22i29i30j33i10j0i22i30j0i13.nopn9DGgOPQ&ved=0ahUKEwjBz4itv_TnAhWYURUIHd69Dk0Q4dUDCAs&uact=5

PWA (Progressive Web Applications) - Andrew Winters

What are they?

PWA stands for progressive web application.

The chief feature identifying a PWA, as opposed to a native desktop application or traditional web application, is that it attempts to deliver the look, feel, and functionality of a traditional mobile desktop application using fronted technologies (primarily HTML, CSS, and JavaScript)(1). Fundamentally a PWA is a website; however, it must display and perform like a mobile application(2)

As in a traditional website, HTML is responsible for the content of the application; CSS, the design; and JavaScript, the behaviour; of these three, it is JavaScript which has the biggest part to play in delivering the native-app like functionality that is expected of these applications.

Another key expectation of a PWA is that it should be able to deliver at least some level of functionality offline; this is accomplished by caching data during periods of connectedness; the responsibility for this lies with a background program known as a service worker(2)**.**

There are four pivotal features which an application requires to be considered a PWA(2):

A web app manifest:
This is a JSON (JavaScript Object Notation(3)) file providing data to the browser in use about the application and its expected behaviour; this should typically provide the name of the app and the URL at which to locate the app on its being launched(4)**.**

Service workers:
As previously mentioned, these are event-driven(5) programs which run in the background of the application, separate from the script running in the user facing application (the application shell)(8, 10); their main task is to coordinate network requests and cache data for periods of offline activity(6).

An icon:
This allows the user to install the application on their home-screen; different native desktop environments will require icons in different formats; this must be considered during their creation(2).

Being served over HTTPS:
Service workers only work on websites using the HTTPS protocol; this is as opposed to traditional websites which at least have the option of using HTTP(7).

This, ideally, results in an application that is designed to run in all standards-compliant web browsers, possessing a uniform end-user experience across all of them.

<u>Advantages</u>

Given that most users can be assumed to have at least one standards-compliant web browser installed on their device, PWAs allow an application to be made as broadly available as possible. They are not coupled to the capabilities or conventions of any particular platform or hardware; do not require the user to sign up to any platform-specific app store[13]; and do not require the end user to engage in any lengthy installation, configuration, or updating process[1, 2].

A well designed PWA has a realistic chance of working on any platform, desktop or mobile, with only negligible differences[2]. As websites, these applications retain the traditional advantages of websites: they are linkable, do not require the end user to store any great amount of data on their system, and their design can be tailored to the device in use using exactly the same techniques as would be used in traditional webpages.

The technologies used in creating PWAs are the same as those used in developing traditional web applications; therefore, it is simpler to find or train developers with the required skill set; they do not need to become familiar with an entirely new tech stack[9].

As mentioned, the presence of service workers requires the use of HTTPS, thus making mandatory what is already best practice[2]; service workers cache resources, so that the application can continue running while offline[1, 2].

Traditionally, native apps were generally able to outperform browser-based equivalents due to their having direct access to platform-specific features for boosting performance, being more easily integrated with other software native to the platform, and not having the web browser as an additional layer of abstraction between the application and the processor. This meant that browser-applications were generally permitted to handle only trivial functions and were usually a thin layer over much more involved server-side logic.

Now, however, modern web browsers are able to run applications that would be too computation-heavy for many desktops only decades ago; if we consider the typical modern scenario of viewing 3d renderings of scenery and buildings in Google Maps whilst simultaneously streaming multimedia from several open tabs, we begin to appreciate the strength and breadth of modern browser performance. In light of this, PWAs seem like a promising and logical choice for anyone launching a new product.

Disadvantages

Despite what is written above, there are a number of issues that detract from PWAs; many of these stem from the same features as the above mentioned advantages.

Running primarily within the browser, a PWA has limited access to platform-specific facilities and does not integrate as readily with software native to its platform; this is becoming less of an issue as browsers advance technologically; however, this still poses a problem with applications that need specific, particularly low-level, facilities[11, 13].

Being built on top of a web browser, PWAs are inherently more high-level than many equivalent desktop applications; this is not as much of an issue as it used to be; however, there are still applications where the speed and precision of low-level code are required; these are poor candidates for PWA development, and generally require a more traditional approach[11]; it is likely that computation-intensive tasks of the kind found in the automobile industry, for instance, will remain strictly the domain of lower-level code for the foreseeable future. Particularly, when we consider the experience of mobile users, PWAs can have a considerably higher impact on battery consumption[11], which could be off-putting to potential customers.

While browser support is not as much of a concern as it used to be, it is still difficult to be certain that a PWA will enjoy complete support across all devices; indeed, some vendors are slightly averse to PWAs as a concept due to the potential impact to their business model[12]; although, there are indications this is changing[12 - see 2017 update, 13].

Whilst there is some convenience attached to bypassing the app store installation route, PWAs can be offered anywhere on the internet, with the end user having to decide whether or not the source is trustworthy; this has security implications far less likely to arise within officially sanctioned app stores[13].

<u>Real-World Examples</u>[17]

- Alibaba, the world's largest online B2B trading platform, designed a new PWA and subsequently enjoyed a 76% increase in online conversions[14] - they used the high-profile JS framework, vue.js[15].
- Debenhams constructed a PWA with the platform, Mobify[16] - since doing so, they have seen a 40% increase in mobile revenue and a 20% increase in conversions[16].
- Uber, one of tech's most widely recognized names, constructed a PWA, m.uber, to offer an app-like experience to end-users owning devices not supportive of their native client; this demonstrates the increase in usership that can be gained by conversion to PWA; this was built using the framework, Fusion.js[18].
- Forbes launched a PWA in 2017; the increased speed and user-friendliness of this application resulted in their readers' article completion rate growing to 6 times its previous level and their average session time increasing by 100%[19]; their user engagement rate was doubled[17].

References

1. https://en.wikipedia.org/wiki/Progressive_web_application
2. https://hackernoon.com/everything-you-need-to-know-about-progressive-web-app-pwa-6524edbb0c57
3. https://en.wikipedia.org/wiki/JSON
4. https://web.dev/add-manifest/
5. https://en.wikipedia.org/wiki/Event-driven_programming
6. https://love2dev.com/blog/what-is-a-service-worker/
7. https://www.smashingmagazine.com/2018/11/guide-pwa-progressive-web-applications/
8. https://developers.google.com/web/fundamentals/primers/service-workers
9. https://hackernoon.com/the-advantages-of-progressive-web-apps-pwa-versus-native-apps-5o2aq308m
10. https://clutch.co/app-developers/resources/pros-cons-progressive-web-apps
11. https://clutch.co/app-developers/resources/pros-cons-progressive-web-apps
12. https://love2dev.com/blog/apple-encouraging-progressive-web-apps-by-rejecting-apps/
13. https://avengering.com/en/why-use-the-pwa-what-are-advantages-and-disadvantagesof-pwa-part-2/
14. https://developers.google.com/web/showcase/2016/alibaba
15. https://madewithvuejs.com/alibaba
16. https://www.thinkwithgoogle.com/intl/en-gb/success-stories/global-success-stories/debenhams-progressive-web-app-boosts-speed-conversions-and-revenue/
17. https://medium.com/progressivewebapps/best-pwa-examples-for-your-inspiration-261bcb3fab47
18. https://eng.uber.com/web-booking-flow/
19. https://www.e-point.com/_fileserver/item/1501040

Cross Platform Applications - Barry Walters

What are they?

Cross-platform development is the ability to build and deliver apps that can run across multiple device platforms, such as iOS, Android, and the Universal Windows Platform. Beyond mobile, it is the process of creating software, applications, or services that can run on more than one platform or operating system.

Advantages & Disadvantages of Cross Platform Apps.

Advantage

You've decided that your business needs a mobile app.Congratulations. Now one of the crucial questions you must answer is, “Should we build a native or cross-platform app?” If your target audience is split between iOS, Android, and Windows phones, then designing and developing for multiple platforms might make sense. Here's a look at the advantages and drawbacks of investing in cross-platform mobile app development.

Speed
It can be potentially faster to develop a cross platform mobile app rather than a native app for iOS and Android. You can leverage one codebase and customize for multiple platforms instead of creating a new codebase for each platform. Making one cross-platform app functional across all platforms can also be more efficient than building multiple.

Costs
In theory, it’s more cost effective to build a cross platform solution for multiple platforms because it leverages one codebase. This is generally true unless the application requires a lot of customization for each platform.

Simplicity
Updates to your cross-platform app would, naturally, be instantly synced across all platforms and devices. There are also a number of technologies like PhoneGap and Appcelerator offering a cross-platform solution that one team of developers can more easily handle to deploy changes.

Disadvantage

Platform limitations

Each platform has its own unique style and affords certain flexibilities missing in others. This may put you at a disadvantage because you may not be able to leverage the unique functionality and tools of the respective native platform when developing an app for multiple platforms.

User Experience

iPhone and Android operating systems, screen layouts, functions, etc. are different so designing and developing a cross-platform app that offers a good user experience on two or more platforms will undoubtedly be challenging. The majority of successful consumer apps are built on either Native iOS or Android. Let's remember that Facebook started out with a Cross-Platform mobile app but changed directions to Native iOS and Android after realizing the User Experience was not optimal.

Integration challenges

Integrating the app with preferences, local settings, and notification apps can be quite a task. Diverse storage options may also require engaging a third-party cloud service. Clearly, there's lots to consider when deciding whether to build native or cross-platform. Consider these pros and cons when assessing your resources and the level of expertise of your team to make the decision easier.

Development for businesses

Apple has a tradition that every januarys, thay disclose statistics to prove how app stores and os apps performed that year’s announcomments a the staggering popularity and penetration of Iphone apps let’s have a look.

General Information

There are 2.2m apps in the app stores.

On average, rear end 2.2m apps every day released on the app store everyday.

The App store reavend recorded in Q2 2019 is around $$255B which is nearly 80% more than what google play store. App’s business is thriving and Iphone app looking and people’s favourite Businesses are looking to capture the market store in apps would be market a in cavite deal by investing in Iphone application development for businesses.
The ‘App economy’ is a highly lucrative and competitive market for independent software vendors as it potentially offers an easy highway to reach millions of users. However, the mobile application landscape is scattered and an application developer has to publish the software for several different platforms to be able to serve a majority of smartphone users. Therefore, a bunch of cross-development tools have been offered to simplify this workload. In this paper, we present an evaluation framework for comparing different cross-development tools. We use this framework to evaluate Adobe PhoneGap tools against

native development in Android and Windows Phone platforms. The results of a case study reveal that while the cross-platform technique was easy to use, the appearance and usability of the app was mediocre at its best. The business impacts of these are also discussed. Cross-platform mobile development is the creation of software applications that are compatible with multiple mobile operating systems. Originally, the complexity of developing mobile apps was compounded by the difficulty of building out a back-end that worked across multiple platforms.
Xamarin is an open-source platform for building modern and performant applications for iOS, Android, and Windows with . NET. ... This pattern allows developers to write all of their business logic in a single language (or reuse existing application code) but achieve native performance, look, and feel on each platform.

- Titanium. Appcelerator Titanium is one of the top choices available. ...
- Cocos2d. If you're building an app that's for a two-dimensional game, you might want to consider using Cocos2d as your cross platform development tool. ...
- Unity3d. Let's stick with the gaming theme here. ...
- PhoneGap. ...
- Sencha. ...
- Qt. ...
- Corona. ...
- 5app.

Companies using java script platform apps
Mobile together

- Easy to use Designer for sophisticated apps
- Develop full-featured apps in record time (as little as 2-5 days)
- Your design builds both the UI and the mobile app back-end
- Connect to back-end data in databases, XML, HTML,
- Web services, JSON, and more
- Native apps for all relevant mobil
- e platforms and desktops
- Browser-based client for laptop and desktop users
- Mobile apps are created in-house, by your development team
- Build enterprise and app store apps

WE BUILD SCALABLE SMART SCALABLE

We are passionate about creating outstanding software solutions that create value for your businesses and lead to success. We are a web & app development company that turns your ideas into a new driving force of your business. We hire brilliant web and mobile app developers to deliver projects on time and maintain the top-notch code standards.