

The C-Suite Guide to Mobile Technologies for mHealth Development



Medical Web Experts™

January 2017



Executive Summary

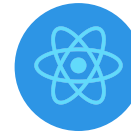
mHealth technologies are forming a more integral part of many healthcare organizations' health IT roadmap. This Guide is designed to concisely present relevant information explaining the benefits, limitations and differences between three different technologies available for mHealth application development. This guide covers:



**Native Application
Development**



**Cross-platform
Application Development**



**React Native
Application Development**

The information presented is intended to serve as an overview and starting point for determining the right set of development technologies for your mHealth application.

Introduction

Mobile technologies have significantly impacted almost all consumer-driven industries over the last decade. As healthcare continues to integrate IT and digital solutions to improve outcomes, reduce costs, and engage patients, mHealth applications will continue to play an important role in the IT and digital strategies of healthcare delivery networks.



Over **165,000 mHealth** applications on the market



iOS mHealth applications **grew 106%** from 2013 to 2015



Only **2% of mHealth** applications are integrated with provider health IT systems

Source: QuintilesIMS

In this guide, Medical Web Experts provides a brief overview of the available technologies for building mHealth applications, and an unbiased assessment of the strengths and weaknesses of each.

Native Applications

Native applications run directly on a mobile device without a browser component. They are built for a specific operating system (iOS, Android, or Windows), which allows access to device features like the camera, GPS, and gesture functions (swipe, enlarge etc.).

🍏 **iOS - Objective-C/Swift**

🤖 **Android - Java**

🪟 **Windows - C#**



The two most common platforms for native application development are Android (55.3% market share) and iOS (43.5% market share). For Android development, Java is the core programming language used. For iOS, Objective-C and Swift are the core languages, with the focus shifting towards Swift.

Source: 9to5Mac.com

Benefits of a Native App

✓ Performance

Better performance than a non-native application, because the native app runs directly on the device operating system.

✓ UI consistency

Excellent user experience with UI/UX design that is optimized for each operating system.

✓ Device Access

Easily able to access device features for complex functionalities -- for example, your app can access the GPS so that a patient can find your nearest care location.

Drawbacks of a Native App

✓ Cost

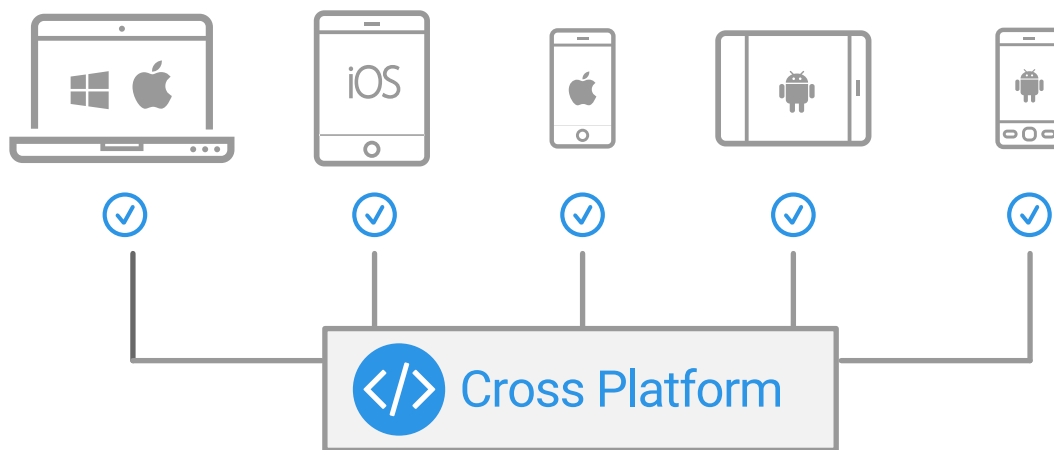
Higher cost, because developers are needed for both Android and iOS, and code written for one system can't be used for the other.

✓ Limited data aggregation

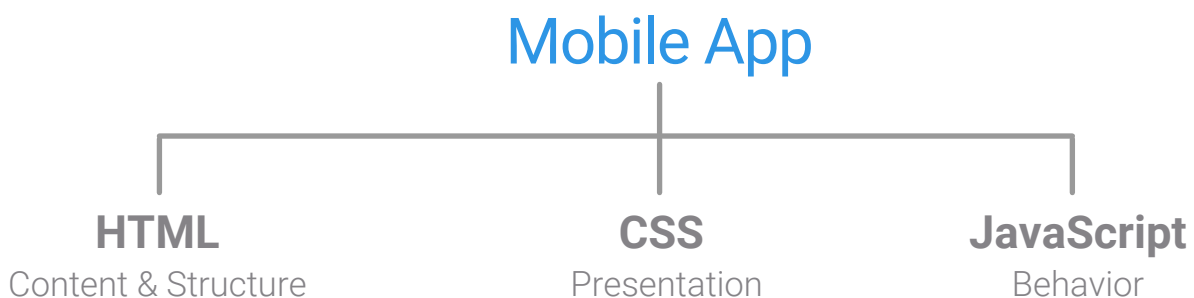
Limited data aggregation capabilities across other Health IT systems. If your organization is looking to leverage its mHealth application for patient data analytics, native applications may present limitations.

Cross Platform Applications

Cross platform applications are those that allow web apps to run on mobile. They do this by wrapping the web application in something called a “**native container**” -- an additional layer of technology that allows web applications to run as mobile applications. This native container also allows access to an array of device features similar to those available to a native app. However, this access is heavily reliant on third party plugins that are often out-of-date or nonexistent.

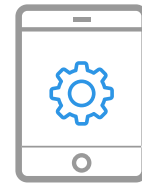


These **cross platform applications** allow greater flexibility for web developers -- and they present an opportunity for software development teams to create mobile applications with the same set of technologies they are already using to develop for the web. They improve efficiency by quickly allowing any development project to serve as both a web and mobile application.



Perhaps the biggest virtue of choosing to develop cross platform mobile applications is that the **HTML, CSS and JavaScript** that your web developers are already writing are the languages they will use to create mobile apps. So the same code your team is using for your website can also be used to create a mobile application.

Tools like Ionic, Cordova, PhoneGap and Xamarin are widely used options that provide the native container necessary to allow your web application to run on mobile devices but have it look like an app. As such, it can be downloaded through the app store and use plugins to access device features using these tools.



Ionic
Cordova
PhoneGap
Xamarin

Benefits of cross platform application development



Web applications can easily be turned into mobile applications.



One coding language for web and mobile apps.



Reduced cost, as your existing web development team can also create your mobile applications.

Drawbacks of cross platform applications



Inferior UI/UX is an issue for cross platform frameworks because they can't access operating system-specific design elements for user interface creation. This means that the UI/UX can fall short of user expectations. Ionic and other new tools are making improvements in this area, but there is still room for improvement.



Lowest performance in terms of battery usage, processing speed, and user experience, as the container adds an extra layer of technology at runtime. Every time an app is run, the native container also opens your browser in the background.



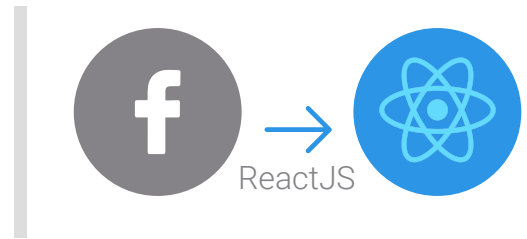
Little access to device features like the camera and GPS, as the only way to access these is through plugins. These are often out-of-date, unreliable or nonexistent.



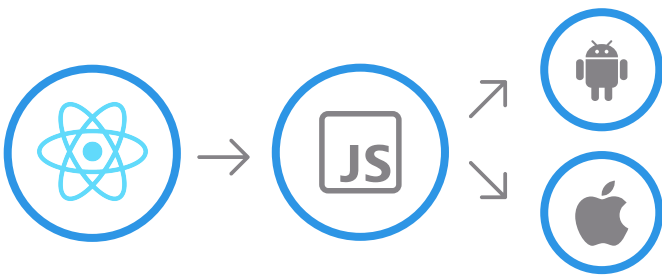
Cannot be used offline. Requires internet connection in order for the native container to connect with the web browser, which enables the app to run.

React Native

React Native is a new generation of mobile development technology created by Facebook, based on the the popular ReactJS library for web development. It has grown continuously in its usage and adoption since early 2015.

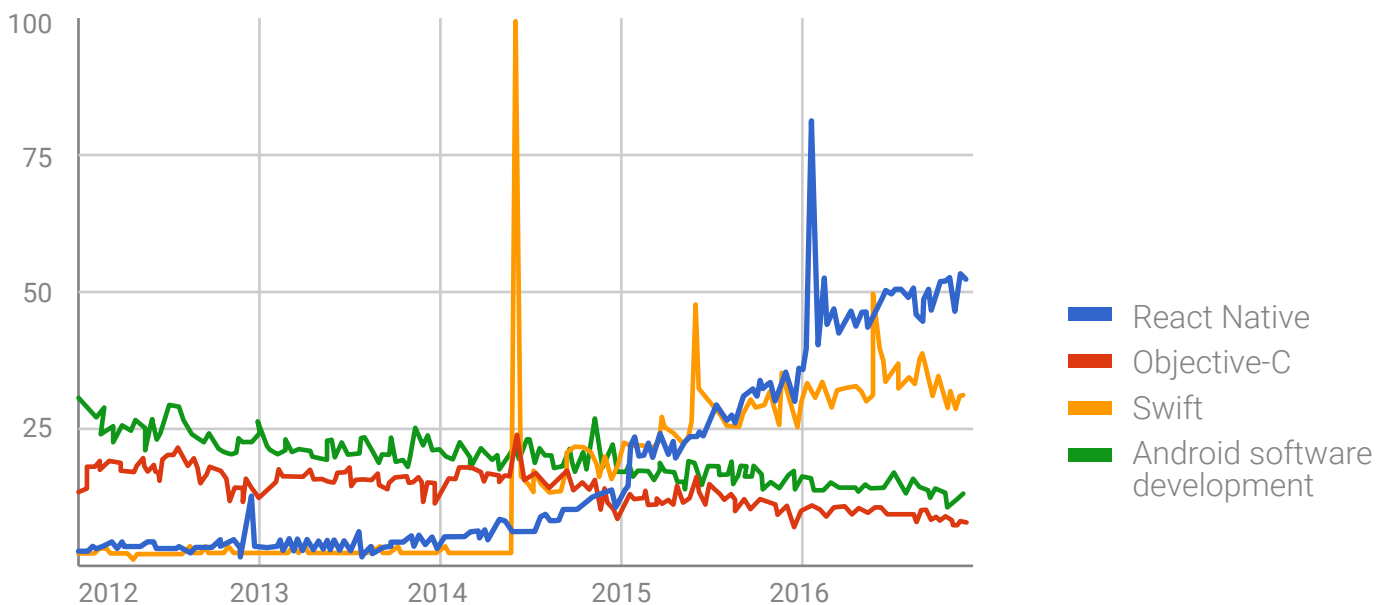


The platform allows developers the ability to create UI/UX with visuals that are standard for each operating system, translating to a much better user experience compared to cross platform applications. **React Native** also allows robust access to device features while allowing web developers to create a single code base in JavaScript that can be used for both Android and iOS.

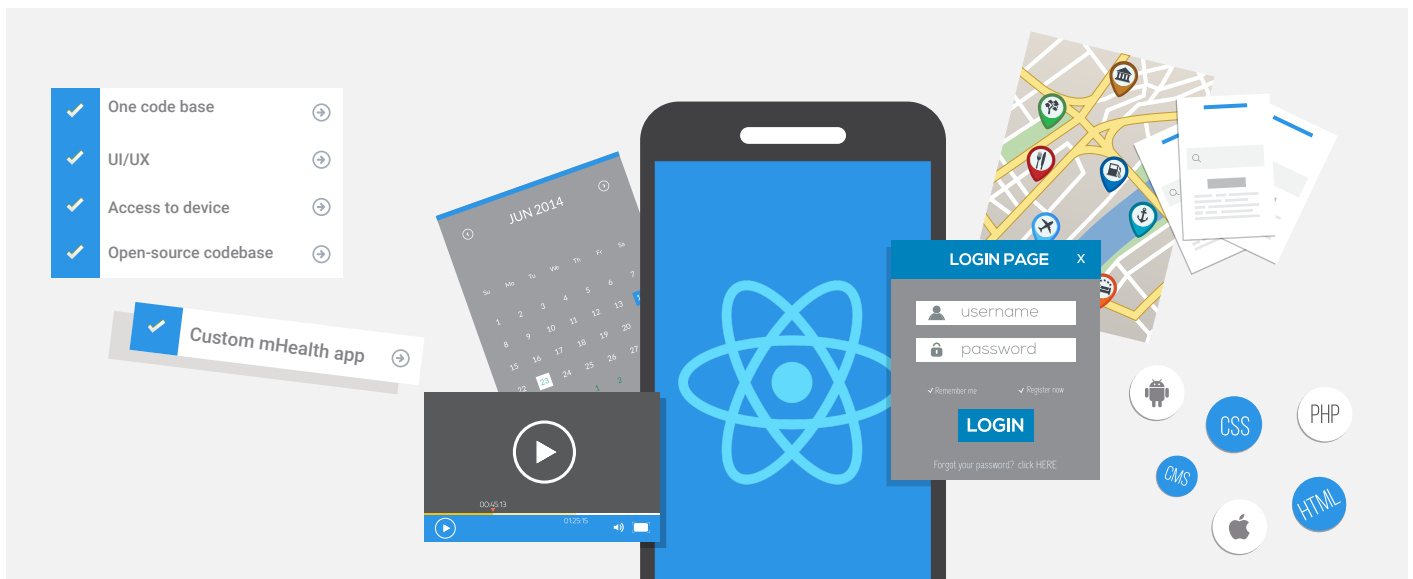


In essence, **React Native** combines the cost-effectiveness of cross platform mobile development with the platform and operating system-specific capabilities of native mobile app development -- the best of both worlds, if you will.

React Native: (United States), Objective-C: (United States), Swift: (United States), Android software development: (United States)

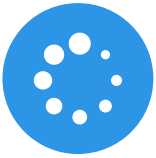


Benefits of React Native development



- ✓ **One code base** across all platforms
- ✓ Ability to create **native application UI/UX**
- ✓ Access to device features, like the camera and GPS
- ✓ **Reduced development costs** compared with native app development, as one set of technologies can be used across both iOS and Android
- ✓ **Open-source codebase** that is well supported and constantly evolving, with a robust developer community contributing to the code and the stability and functionality of the platform.
- ✓ **Increased versatility** as custom mHealth app data can be submitted to a custom administrative portal for population health, care coordination, data analytics and more.
- ✓ **Increased data sharing and aggregation capabilities.** Backend API can be integrated with other existing Health IT systems and applications across the organization.

Drawbacks of React Native development



Faster performance than cross platform, but still slower than native applications. The React engine -- which generates operating system-native code from JavaScript -- adds an extra layer of technology between the application code and the device operating system.



Platform stability that still has not reached a v. 1.0. Because this new platform is still in beta, there are frequent changes and updates. This means that, for now, application maintenance can be resource intensive and your developers need to keep up with new releases and updates.



As with all technologies, native, cross platform, and React Native mobile apps all have their pros and cons. One won't necessarily always be the right choice. Instead, you should choose your app development technology based on the functionalities you will need for your new mHealth application.

Medical Web Experts is a team of trusted industry leaders that has been providing healthcare professionals with top notch web and mobile solutions since 2003.

To learn more about our custom mhealth application development work, get in touch to start a conversation.



Medical Web Experts™

[CONTACT US](#)