
Mobile

This is CS50. Harvard University. Fall 2015.

Cheng Gong

Table of Contents

1. Mobile 1

1. Mobile

- Today we'll look at how we might build mobile apps, and today there are three main ways: hybrid apps, native apps, and web apps.
- We talked about HTML and CSS, and we remembering having the issue of having to make sure our text and layout was still readable on a much smaller screen.
- But for more complicated layouts, we would want to manually control how the text resizes and how the layout changes.
- Harvard's mobile application (from the App Store) has a layout like this:

- But their mobile website, `m.harvard.edu`, looks the same:

- So this is an example of a hybrid app, where the main application is written as a webpage with languages like HTML, CSS, and JavaScript, and there are tools and frameworks that create an iOS or Android app that can be downloaded and run normally but has this webpage embedded, as though it were using the built-in browser but hiding the address bar.
- A native application, on the other hand, is written specifically for the device and is either in Java for Android or Swift or Objective-C for iOS.
- A hybrid app is useful because, in a place like Japan where about half the phones are running iOS and half are running Android, we only need to write a mobile website once and use a tool to generate two hybrid apps that work on both devices.

- And we could just make our websites and tell our users to go to that URL, but having an app and an icon helps users remember our website.
- And a web app might not have as much functionality, since the browser might not have access to features like location or the camera, but many hybrid app frameworks have enough code to connect those features again.
- Three popular frameworks for creating hybrid apps include PhoneGap, React Native, and Titanium.
- But hybrid apps are slower, because the UI is written in JavaScript which needs to be converted by the browser to native code. Some libraries used to write this code include jQuery Mobile and Sencha Touch, and they make it easier to create pages that look like iOS or Android interfaces.
- Choosing what devices to support is also related to what our target market is like, and what hardware that they have.
- Android, for example, has many versions of the operating system running on its devices, so apps written for newer versions might not work as well on older versions without certain features or functions.
- iOS, on the other hand, has 84% of its devices running the latest version, which Apple accomplishes by making both the hardware and software, so all the new hardware has the latest software pre-installed. And automatic updates are important in that it gives everyone security fixes.
- Christina, MBA '14, joins us from Tripadvisor to talk about their experience in creating a mobile app.
- At first, they had a hybrid app that was slow and buggy, but now they have one codebase for an iOS app in Objective-C, and while there isn't a native app for Android yet they still have to consider the different versions.
- And since the design of user interface elements like buttons and boxes in iOS and Android are different, they also need to consider the look and feel of their app for each system.
- The back-end API that serves content and data is shared across all platforms.
- And tablets need entirely different layouts too, since they don't want to just make everything same as a phone but twice as big. So they decided to think about features as modules and rearrange the modules on the screen depending on the device.

- They do have a Windows Phone app, but it's a very basic hybrid app that just has the mobile website embedded.
- And an advantage of a native app is that content can be easily pre-downloaded and viewed without internet later, which is important for travelers.
- Since there are lots of screen sizes now too, in addition to testing various sizes, they also try to make their designs responsive, and resize themselves to some extent.
- Because mobile user growth is so rapid, TripAdvisor used to have a separate mobile development team but has now moved them to the entire engineering team, since some decisions end up applying to all platforms.
- While being a non-technical PM on a technical team might be difficult at times, learning more about technology problems through reading blogs or talking to engineers casually can be helpful.
- TripAdvisor doesn't have a product that's particularly cutting-edge, but they are experimenting with machine learning to make suggestions and process large numbers of text reviews, etc. For example, they can look for keywords in reviews to automatically determine if a hotel has a pool, even if that information isn't provided elsewhere.
- And that's all for this year!