Mobile Development (CM3050)

Couurse Notes

Felipe Balbi

September 21, 2021

Contents

Week 1

Key Concepts

- Understand the limitations and advantages of different platforms
- Discuss the elements of apps you enjoy
- Understand the course structure

1.006 Getting started on this module

React Native is an Open Source framework created by Facebook which lets us JavaScript to produce native code for different operating systems. Some of the benefits are:

1. Single language for multiple platforms
2. Ease and flexibility of JavaScript
3. Near-native performance
4. Heavily used in industry

1.103 The pathway from developer to consumer

The process to moving an app from developer to consumer is roughly laid out below.

1. Ideation

   The idea for an app has to come from somewhere

2. Storyboarding

   Drawing an idea is usually the simplest form of conveying that idea

3. Prototyping

   Helps us understanding challenges that may arise from implementing the idea

4. Feedback and Production

   Using our prototype, we can gather feedback to understand if the app fulfills its purpose

5. Testing

   Automated testing and manual testing is an important part of guaranteeing stability and quality of the app

6. Approval and Release

   After testing, apps go through an approval process by e.g. Google and Apple before they can be released to end users

7. On sale

   Apps are finally available in the App Store and can be purchased by anyone

1.201 Multiple codebases

There are two major mobile OSes:

- iOS ($\approx 26.5\%$ market share)
- Android ($\approx 72.9\%$ market share)

This means that developers usually have to write the same app twice — once for iOS with Switch of Objective-C, and once for Android using Kotlin or Java —; needless to say this creates the possibility of minor incompatibilities between the two versions of the app.

The alternative to writing two apps is to write a single app in a common language that can be compiled for both platforms.

Native Apps Written specifically for a single OS

Hybriid Apps Written in a common language

1.203 Native apps vs hybrid apps

React Native, while using a common language for all platforms, is much closer to a native application. During compilation of the project, React Native renderas our views with native code on the target platform thus giving us the best of both worlds.

In table 1 we list pros and cons of Native Apps. Similarly, table 2 shows the same comparison for Hybrid Apps.

Table 1: Native Apps Pros & Cons

Pros	**Cons**
Fast	Lengthy and Complex to build
Device-level APIs	No cross-compatibility
Native GUI Elements	

Table 2: Hybrid Apps Pros & Cons

Pros	**Cons**
Cross-compatibility	Slower perforance
Larger market reach	No device-level APIs
Faster to write	

1.301 What is React Native?

React Native is a framework written in JavaScript that allows us to build code for multiple platforms.

1.302 What is Expo?

Expo helps manage a React Native project.

1.305 Javascript arrow notation

A traditional JavaScript function looks like this:

```
function (a) {
  return a * 2;
}
```

We can convert it Arrow Notation like so:

```
(a) => {
  return a * 2;
}
```

Single line functions can be further simplified by removing braces and the `return` keyword:

```
(a) => a * 2;
```

Next, the parenthesis around `(a)` can also be removed:

```
a => return a * 2;
```

Functions without arguments or with more than one argument must keep parenthesis around arguments:

```
() => 42;
(a, b) => a**2 + 2*a*b + b**2;
```

1.306 Creating a new React Native project and overview of the file structure

Create a new project with:

```
$ expo init --npm
```

Choose a name for the application and choose the `blank` template. Hitting `ENTER` will create a new directory with the name of the application and all dependencies already installed. When it completes, we can change into the new application's directory and run:

```
$ expo start
```