Part 1: Theoretical Questions

David Chong

1. Mobile Testing Concepts:

a. **What are the main differences between testing an Android app and an iOS app?**

iOS is less customizable because of the Apple restriction in its enviroment, as it’s closed, but in the other hand there are way less devices and SO versions, because of that the testing on Apple devices is not too wide so it’s relatively easier to ensure the quality in a good amount of these devices.

Android compared to iOS is more open and customizable but the quantity of devices with Android SO is extremely huge, a lot of brands like Samsung, LG, Google Pixel, etc, use Android, so the control in the SO features is less wider than in Android, because of that, could be some bugs/errors in one specific device due to the company itself SO personalization, it could be something problematic for the assurance of the quality. Also the amount of devices make it impossible to test all of them.

b. What are some common challenges in mobile automation, and how can they be addressed?

Related to the previous answer, the amount of devices and different SO versions become this a little bit hard a problematic to include in the testing process.

Another relevant challenge is the dynamic elements.

c. What are the differences between real devices, emulators, and cloud-based mobile testing solutions?

Real devices always will give the best results because it’s the device that the user will be working with, but also it’s expensive to have a good amount of them to test, and it could suddely break down.

Emulators as Xcode and Android Studio ones, are more easly to use, mantain and cheaper to use, with only one computer it’s possible to test a lot of devices.

Cloud-based also offer a good amount of devices like the local emulators but depending on the amount of usage it could be way more expensive but the access to them are easly and doesn’t required too much configuration on your own machine.

2. Appium & WebdriverIO:

a. **Explain the purpose of desired capabilities in Appium.**

These are like the basic/inicial configuration of the devices that will be used, telling to Appium what device, SO, SO version, and where the app is located, etc. With the right information, appium can access to the rigth device, and execute the testing.

b. How does Appium interact with native, hybrid, and web apps?

With the native apps, it works directly with the elements of each platform using the right framework like UIAutomator and XCUITest.

For the hybrid ones, it interacts with the native part and also with the web part like in a browser.

With web apps, works similar than Selenium with the web apps but in the mobile device.

c. What are the advantages of using WebdriverIO for mobile automation?

It’s relatively simple and clear to use, and can be integrated with more languages even if its based on JS. Also the integration with Appium makes it a very robust tool to use during the mobile testing.

3. Test Automation Strategy:

a. How do you handle dynamic elements in mobile apps?

With the Xpath selectors that doesn’t depend on static values.

b. What are the best practices for structuring a mobile automation test framework?

Use the POM to have a more organized project, with the logic and the tests separeted. It gives more control and the reusability of the code is easier.

c. How would you implement retry logic for flaky tests in WebdriverIO?

Using the “retries” to ensure that the test will be running again even if the first try fail.