**IOS App Testing: A Beginners Guide with a Practical Approach**

Mobiles are true assets in this era, in concurrence with the industrial revolutions the mobile platforms have taken a whole significant transition to make the customers more amazed to experience the features that it offers. We have plethora of mobile vendors available in the market but if we ever interested in knowing the platforms or the operating systems that these are built on top of, we have a few types available which are as below

* **Android**
* **IOS**
* **Windows**
* **Symbian**
* **Blackberry**

Among the list of operating system’s Android and IOS are the two buzzing OS’s. In this blog we are going to focus on the IOS platform and its testing methodologies.

To give a gist about IOS, it’s a dedicated platform for Apple devices such a Iphone, Ipad, Ipod, apple watches..etc. This operating system is specifically designed by Apple and they don’t license this to be installed on other mobile devices unlike Android.

It is fact that everything that’s developed should be tested, in the same way when an app that’s related to IOS or the devices that runs IOS must be tested before it goes live. Let’s understand the way to test and the best practices to follow for proper IOS testing

**IOS APP testing**

We deal with so many applications in our day in and day out, most of the applications we come across are intended to support multiple platforms to hit the target market.

There are three kinds of applications that we must aware

**Web based application**

These applications are designed in way that these will only operate on browser, in our discussion the apps that operate in safari browser. We might have to do a http or https call to experience the application renders web page.

**Native application**

These applications are designed in a way that these are specific to the operating system and must be installed locally to make use of them (eg: vlc media, anti virus software..etc)

**Hybrid application**

These applications extend their operations from both the browser and as well as being an installed app on top of the operating system eg: twitter, facebook..etc

Now, one must be thinking od

How do we test them?

How do we deem that the web responsive design is proper?

Do we consider them testing on the real device or any simulated software?

Well, to answer for above questions for a proper IOS app testing one must consider testing them on real devices and consider testing on the emulators

Let’s see some of the testing which are done using real device

**UI/UX testing**

User interface quality is the cutting edge for any application to be successful. Below are the some elements that have to be thoroughly tested as part of this UI/UX testing

**Inputs**

We feed inputs to an application in lot of means a few such examples are by soft touching, prolonged touching, scrolling and accessing the buttons present on the UI. We as a user should be focusing that all the design is apt and the application is accepting the inputs without having any problem.

**Hard/Soft keys**

There are native apps which would respond to the hard panel buttons such as home button, volume keys. We need to test that these buttons function on the app well

**Screens**

Attractive screens with good design will always grab the attention of the user. We should have a hawk eye on the each and every field that’s available and validate the color of the element, width and position of the element and its workflow. We can greatly differentiate the UI between the conventional android and the IOS. We tend to see IOS behaves the other way in most of the cases, hence careful observation is must.

Eg: for a notification pop up the confirm and cancel options displays in the opposite order than that of andriod

**Messages & Notifications**

When we are working with a particular application we likely receive messages and notifications, we should be test these properly and ensure the application doesn’t crash during these events.

As a tester while validating the screens, we should not just looks for bugs but also call out if something is weird and it has to be shifted or colored differently as that enhances the user experience (UX) of the applications

**Security testing**

There shouldn’t be any compromised nature while assuring the amount of security that we offer as that’s where we gain the trust of the customers. We should test all the positive and negative scenarios to ensure the system is not exposed to hackers.

One good examples of this case would be checking if the data is encrypted before it is being sent to the payment gateway.

**System testing**

System testing is done only when the complete application available, in this testing the end-to-end flow of a business use case is tested. System testing can be done via emulator and via real devices.

When we test through the emulator, we must understand the fact that we are not testing the real system, this approach is followed to avoid the cost of buying all the devices

But as a best practice we must be testing them on real devices to ensure all the scenarios are working, the real advantage of testing using the real device is we can experience the look and feel of the applications as well as how various device characteristics behave with the application

Eg: let’s assume that the application we are testing is an online shopping site and the application allow us to make payments using different payment gateways, which will work with the corresponding bank vendor

In the above context system testing is all about validating all the screens that are involved in the whole transaction i.e. right from the searching a product, placing an order then making the payment and acknowledging that the order was successfully placed.

Thus far we have seen the types of tests that are carried out using a mobile phone, now let’s take a look at testing types those can be managed via emulators

**Unit testing:**

By the definition unit testing is done on specific standalone module to ensure that piece of code is working in all possible scenarios, this is in general done by the developers.

Eg: testing of one particular screen UI/UX

Since we don’t have the real system or the whole application built we can consider testing this using a simulator, so that without any hardware dependency we can test this for different types of IOS devices

**Integration testing**

In this level we will integrate two more components then test to see if the app communicates to the other system well, this can again be handled through emulator without having to rely on the hardware components

**UI testing**

UI/UX can also be done using the emulators, this approach saves us from the cost as we can avoid buying all the different types of devices that are available. The only fact that we should concerned about here is we are not testing the real system, we may fail in the field though we would have noticed everything went well.

All the above mentioned testing types were done manually despite the fact that they were done using real device or emulator. But keeping the importance of automation testing in mind, we should consider automating the as much as we can

**Automated Testing**

We all know how crucial the automation testing is in the testing world. It saves us lot of effort and the cost. The IOS mobile app testing can also be done via automation we have popular tools called Appium which helps us to automate the testing.

Types of testing that are automated

**Regression**

Mostly the regression test cases are automated to ensure the previous features are not broke due to the new implementations

**Compatibility**

There are a few devices that run IOS, we can test the application on different platform using the automation so that we save lot of manual effort.

**Best practices to adopt**

Nothing is perfect during initial stages, best practices are not the guidelines those were penned before in hand rather those are the learning from the previous outcomes. We should conduct a proper retrospection then understand where to get better, then implement them in the process as best practices. Following are some of the best methodologies to adopt while testing the IOS application

**Bring in more automation**

Automation is vital in the current era given its importance. As a team we should always focus on the areas that are handled manually and try to bring in more automation

**Use real device to test**

Though usage of all mobile types is cost oriented but still we can achieve lot more reliability and can understand the field scenarios better and then test. This approach will help in avoiding the bug leakages

**Capture screens and remote logs**

Logs are essential as the developer needs them to understand the root cause for a particular problem. Hence capturing logs and screenshots always help to decipher the problem then fix accordingly.

In a very high level the IOS app testing no way different from the mobile application testing strategy, but this article mainly focuses on the IOS apps, hope this has some interesting content in it.

Happy learning ☺