AMBA APPLICATION TESTING

Author: Joseph Kibira

**BACKGROUND INFORMATION**

**Based on the above figure, there are 2 possibilities for an app:**

**1) Mobile App for the web application:** It may be possible that the customer has a web app and wants to make a mobile app for the same. Amazon, Jabong, Myntra etc., did the same thing, and the moment mobile apps started creating a boom in the market these commercial companies realized that soon people will move from web apps to mobile apps.

**2) Mobile App as a separate business entity:** I tested a similar logistics app, in order to increase the business it created a separate app for its drivers to facilitate a smooth delivery and removed the paperwork.

The drivers had the app installed on their phones to move the shipment to different stages of delivery and their corresponding status, and delivery updates were reflected in their existing web app.

Other such ***Examples*** where mobile app is a separate entity are WhatsApp, health tracking apps like S Health etc.

**‘To Do’ list before the start of Testing Process**

Unlike a normal testing cycle, for the very first release of a mobile app, there are a few items which need some extra research and planning for a successful delivery of the app.

**Following are some of the activities that are important for a QA to carry out before the actual testing starts:**

**#1)** **Research on OS and Devices:**

Apart from understanding the app requirement, you also need to understand the impact of the various OS versions and the device models. You need to create test cases or scenarios w.r.t the OS and device models.

**#2)** **Test Bed:**

Unlike desktop, web and iOS apps, the test bed creation for android apps is very tricky because of the open Android system. The decision should be taken after discussion with the BA, developers and the product owners to find the target audience.

**#3)** **Test Plan:**

As it is the first release, an approved plan with approved test bed, testing types etc., should be in place. If there is a separate time required for researching functionalities or business rules or if the testing is going to be tricky etc., everything should be covered in the plan.

**#4)** **Automation Tools:**

The automation tools required for mobile apps are very different from that of desktop or web apps. Hence you may need to spend extra time and effort to learn the tool. The time and effort required for the same should be approved by the product owner or customer.

**#5)** **Testing Techniques or Methods:**

Unlike the conventional tests that we normally do, there are few tests that are very specific to mobile apps. Namely field test, GPS test, image capture test, etc. These need to be documented in the test cases along with how these types will be tested.

**Testing Types for Mobile Apps**

**Following types of testing are performed in order to certify an Android application:**

1) Functional Testing:

Testing is normally achieved by user interface initiated test flows. Not just the flow of a use case is tested, but the various business rules are also tested. Testing is done by certifying the requirements. i.e. whether the application is working based on the requirements.

Let us take the **Example** of Zomato app.

When you launch the app, based on the location, the list of restaurants shows up. Now as a QA, you have to test the listing of restaurants.

**This is functionality testing and here you will test functionalities like:**

* Verify whether the restaurant cards are shown w.r.t the location.
* Verify whether the restaurants shown are in the given range (km) of the location.
* Verify whether the review count for each card is in sync with the reviews given.
* Check whether adding a new restaurant for a location reflects in the list.
* Verify whether the restaurants are listed based on the filtering rule of Zomato etc.

2) Android UI Testing:

This is an user-centric testing of the application. In this test phase, items such as visibility of text in various screens of the app, interactive messages, alignment of data, the look and feel of the app for different screens, size of fields etc are tested under this.

**For the same above Example of Zomato, the UI tests would be:**

* Verify whether the cards are shown properly aligned with the mobile screen size.
* Verify whether the image of the restaurant is shown as expected.
* Verify whether the card details, the rating of the restaurant, cuisine type, address etc, are shown properly aligned.
* Verify whether the scrolling of the list is glitch free and the cards are not shown misaligned when a long list of cards is scrolled etc.

3) Compatibility Testing:

This testing is done mostly in the form of two matrices of OS Vs app and Device model Vs app. Usually, a list of supported OS (and sometimes devices) is provided by the product owner or customer.

**The need for this testing is:**

* When an OS like Android is taken into account, 7 base versions are found neglecting the number of patch releases/EPs.
* *Can you guess the types of Android devices currently working all over the globe? 1000? 2000? Wrong!* The correct answer is 24000. There are 24000 types of unique Android devices that are present and active in the world.
* With variations such as these, compatibility testing plays a vital role in certifying an android app.
* We not only need to validate the OS and type of the device but also validate few other features which fall into compatibility testing.  **The features include:**
  + Screen size
  + Screen resolution
  + Network connectivity

4) Interface Testing:

In other words, it is also termed as Integration testing. This testing is done after all the modules of the app are completely developed, tested individually and all the bugs are fixed verified.

Interface testing includes tests like a complete end to end testing of the app, interaction with other apps like Maps, social apps etc, usage of Microphone to enter text, usage of Camera to scan a barcode or to take a picture etc.

**Again considering Zomato, the integration tests would be like:**

* Verify whether the user is able to book a table for a restaurant.
* Verify whether the user is able to view the menu and order food online.
* Verify whether the user is able to avail a PayTM coupon while ordering food.
* Verify whether the user is able to view the location of the [restaurant on Google Maps](https://www.softwaretestinghelp.com/drop-a-pin-in-google-maps/).
* Verify whether the user is able to open the phone dialer and call the restaurant etc.

5) Network Testing:

**The key features of Network Testing include:**

* The app should talk to the intermediate service so as to carry out the process.
* During this testing, request/response to/from the service is tested for various conditions.
* This test is mainly done to verify the response time in which the activity is performed like refreshing data after sync or loading data after login etc.
* This is done for both strong wifi connection and the mobile data network.
* This is an in-house testing.

6) Performance Testing:

Performance of the application under some peculiar conditions are checked.

**Those conditions include:**

* Low memory in the device.
* The battery in extremely at a low level.
* Poor/Bad network reception.

Performance is basically tested from 2 ends, application end, and the application server end

7) Installation Testing:

There are two types of apps on an Android device i.e, Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing needs to be carried out. This is to ensure smooth installation of the application without ending up in errors, partial installation etc.

Upgrade and uninstallation testing are carried out as part of Installation testing.

8) Security Testing:

Privacy and security are the 2 major requirements of an app. However, in Banking, healthcare, this becomes the primary requirement.

Testing of the data flow for encryption and decryption mechanism is to be tested in this phase. Access to stored data is also tested in this phase.

9) Field Testing:

Field testing is done specifically for the mobile data network and not in-house but by going out and using the app as a normal user. This testing is done ‘only’ after the whole app is developed, tested and regressed (for bugs and test cases).

It is basically done to verify the behavior of the app when the phone has a 2G or 3G connection. Field testing verifies if the app is crashing under slow network connection or if it is taking too long to load the information.

You can read more about field testing in our article ***“Importance and necessity of Field Testing for mobile applications”***

10) Interrupt Testing:

This type of testing is also known as Offline Scenario Verification. Conditions where the communication breaks in the middle are called as offline conditions.

**Some of the conditions where interruptions of a network can be tested are as follows:**

* Data cable removal during data transfer process.
* Network outage during the transaction posting phase.
* Network recovery after an outage.
* Battery removal or Power On/Off when it is in the transactional phase.

**Best Practices in Android App Testing**

There are certain factors to be considered while putting a strategy for mobile app testing.

**They are:**

1) Device Selection:

* This is one of the most critical steps before starting the android application testing process.
* Decide which devices are to be taken into account for the testing process.
* The selection is to be done so as to maximize the number of target customers.
* Factors such as OS version, Screen resolutions and Form factors [Tablet or smart phones] play a vital role in the selection phase.
* If required, even the help of Emulators can be taken into account.
* But, Emulators should not replace the physical device testing process.
* Device emulators are cost effective and they come in handy during the initial development phase.
* But, to test the real-life scenarios, physical devices are the must. Both emulators and physical devices are to be used in a balanced manner for an optimized result.

2) Beta Testing of the Application:

* Beta testing is very effective in testing with the real-world users, real devices, actual networks and applications installed in a wide geography.
* This gives a clear picture of the network density, network variations [Wi-Fi, 4G, 3G, and 2G] and the impact on the application.
* Beta testing in the real world is one of its kind and cannot be replicated in a controlled environment.

3) Connectivity:

* Normally, Android applications are connected to the internet for various requirements.
* The connectivity on different devices plays a key role in putting up the strategy.
* Mostly the connectivity is controlled by simulation software which helps in regulating the network speed, latency, and limited connectivity while testing.
* It is said that testing under real network connections is always advisable for real-time result/data.

4) Manual or Automated Testing:

* Though automation testing takes ample amount of time for the first run, it comes in handy when the testing has to be repeated. This also reduces the overall time span of testing during the different development stages.
* Android Automation should be clubbed with Manual testing when the regression testing repetition is high in the application development phase, compatibility testing has to be done for the same application on different OS versions, backward compatibility checkpoints etc.

**Android Testing Framework**

There are a handful of Android testing frameworks that are available.

**In this tutorial, we are going to discuss 3 different types of most commonly used frameworks:**

* Robotium test framework
* Robo-electric test framework
* Appium test framework

1) Robotium Test Framework:

* This framework is used to write sophisticated and robust black box test cases for Android applications.
* It supports both native as well as hybrid clients.
* Functions, system test cases, and User acceptance test cases can be written using this framework.
* Robotium supports Android 1.6 and above and also support for Dialogs, Menus, Activities etc.
* This framework handles multiple Android activities automatically.
* A handful of methods are given as a part of Robotium for interacting with different graphical components of the Android application. ***Some of them are as below:***
  + goBack();
  + getButton();
  + isRadioButtonChecked();
  + searchText(“User”);
  + click on button(“Logout”);

2) Robo-electric Test Framework:

* This framework helps in testing Android applications on the JVM based on the JUnit4 framework.
* It uses Android API.
* This helps in writing test cases and running them on the JVM.
* Under this, all the classes are replaced by something called shadow objects.
* Whenever a method is implemented, Robo-electric internally sends the call to the shadow object.
* Based on the implementation, if a method is implemented by shadow object then a value is returned. Else, NULL is returned.
* Because of the shadow objects and JVM, the execution becomes faster.

3) Appium Test Framework:

* This framework works for native, hybrid and mobile –web apps for Android devices.
* Appium is free to use utility.
* Single API works for both Android as well as the iOS platform. This is one of the frameworks which supports cross-platform testing.
* It uses Selenium Web driver to interact with the Android application.
* Appium supports script writing using a lot of programming languages such as Java, C#, Python, PHP, Ruby etc.

**Conclusion**

Android application automation can be clubbed with manual as per the need. However, if Androind Automation testing is added to the strategy, selecting the correct tool is a big task. While selecting the automation tool factors like Multi-platform support, Test workflow, Price of the tool, service/support etc. should be taken into account.

There are several challenges involved in Android application testing. There are different factors to be taken into account before the Android testing process can really be implemented but once done this becomes a very interesting task.