Mobile App Testing: Challenges, Strategy and Approaches

I. P. S. Sethi
Senior Technical Director
National Informatics Centre
New Delhi

Vaibhav Agarwal
Technical Director
National Informatics Centre
New Delhi

Shivakant Shukla
System Analyst/Scientist-B
National Informatics Centre
New Delhi

ABSTRACT

The Mobile app is growing day by day. A larger number of mobile apps are available on various play store to download and use. Mobile Application Testing is a process by which a developed application of a mobile device is tested for its functionality, consistence & usability. It can be automated and manual. It almost goes through the normal testing process but the testing techniques may vary for mobile devices.

Performing Mobile App Testing is quite different from enterprise testing and website testing. Mobile App Testing needs to be performed on variety of software platform, versions, hardware, network condition and what makes it more complex are the various devices, fast changing mobile OS, the frequent introduction on new devices, and unpredictability of simulators.

The main objective of this paper is to find out various challenges in testing the mobile app and pointing out how it is different than web based software application. The paper also lists the strategy and approaches which need to be considered for mobile app testing

General Terms

Mobile App Testing, Software Testing, Device Emulators, Browser Emulators ,Operating System Emulators

Keywords

Mobile App Testing, Software Testing

1. INTRODUCTION

Testing of Mobile Application(s) is a process by which a developed application of a mobile device is tested for its functionality, consistence & usability. It can be automated and manual. It almost goes through the normal testing process but the testing techniques may vary for mobile devices.

Now a days, market of mobile devices is booming and probably it may be continued for another 5-10 years and hence websites and apps which are mobile friendly are exploding like anything in the market. Each and every business entity is trying to get out the maximum benefits of this mobile technology. With the exponential increase for mobile usage has led to mobile testing. Businesses are focusing on creating mobile strategies and roadmap before implementing the application for their users. It became essential to develop an App with adequate / effective features and functionalities which are customer oriented and beneficial to the customer as well as business units. But it is more critical to have an effective testing plan and strategy before any app is developed. As client is usually in millions for a certain product – and a product with bugs is never appreciated. It often results in monetary losses, legal issue and irreparable brand image damage.

Testing the mobile application is not straight forward as testing the web based or desktop software application due to various factors like multiplicity of devices, OS and their versions etc. The need was felt that a paper should be written which gives the holistic view of the various challenges in mobile app testing and how to devise a strategy to test the mobile app. The paper also focuses on approaches of mobile app testing.

2. TYPES OF MOBILE APP

a) Native apps: A native application is created for use on a platform like mobile and tablets. These applications get installed on the mobile device and often delivered from the Appstore.

b) Mobile web apps: Web applications as one accessed using a browser on the mobile device or server-side apps to access websites on mobile using different browsers like chrome, Firefox by connecting to a mobile network or wireless network like WIFI

c) Hybrid apps: These are combinations of native app and web app. Hybrid embeds a browser in a native application. These also run on devices, online or offline.

3. MOBILE APP TESTING VIS-À-VIS WEB BASED SOFTWARE TESTING

More than a decade ago, software testing was almost the exclusive domain of desktops and laptops. Now, it has become an umbrella term that applies to a range of platforms from PCs and browsers to smart phones, tablets, and even wearable technology.

Despite this proliferation of new devices and interfaces the fundamental building blocks of software testing are same due to universal testing methodologies, adaptive nature of software testers and the convergence of technologies allows the tester to “essentially” test for the same bugs and functionality (although not always) but despite these similarities, the testing for mobile and testing for the Web are not the same.

Some of the major differences are listed below

3.1 Limited screen size

The most obvious difference is the screen size. Responsive design is relatively easy to code for desktop and laptop browsers – most of which come with predefined ratios anyway.

By contrast, mobile devices are much smaller. Aligning images and text becomes a real challenge – especially when you factor in features like portrait orientation (i.e. the ability to rotate a mobile device and have the image flip accordingly).

Worse still, there is so much more variation – even when dealing with the same manufacturer.

For example, the iPhone 5 has a 4” display, whereas the iPhone 6 is 4.7” on the diagonal. When you add in the iPhone 6 Plus (5.5”), iPad Mini (7.9”), and iPad standard (9.7”), it becomes harder and harder to code and test mobile applications that look “good” on all screens.

3.2 Storage and RAM

Screens are not the only spatial constraint mobile software testers face. You also must factor in the limited storage and processing power of today’s mobile devices. Even high capacity phones can quickly fill up as users download apps and multimedia.

In the browser world, such constraints are moot. Desktop storage is essentially unlimited (measured in terabytes). And cloud-based storage is easy to increase, even if this requires charging higher prices to end-users.

3.3 Internet Access

With the exception of a few off-line browser applications (e.g. Gmail), Web-based software always requires an Internet connection.

Mobile apps may or may not need online access. When Internet is needed, however, mobile software testers must factor in 3G and 4G – in addition to normal Wi-Fi.

3.4 Input Interface

Another major difference is how users interface with mobile applications versus Web-based software. With the latter, it is usually with keyboards and with mobile applications, testers must factor in touch screens, USB connections, and even voice recognition.

3.5 Performance Speed

There is one area in which Web-based software testers face a disadvantage i.e. speed

According to Google, delays of 400 milliseconds or longer are enough to scare away today’s increasingly impatient online browsers. You may have an easier time testing applications for the Web, but your product can still fail if it doesn’t deliver as promised in 399 milliseconds or less.

Mobile users tend to be more forgiving. If they’ve taken the time to buy, install, and configure an application, they’ll stick around a little bit longer.

3.6 Supported platforms and devices mean you have more combinations to test

Desktop apps were usually targeted for specific platforms and it was relatively easy to access those platforms. Web based applications made it a bit more challenging by adding another dimension: browsers.

Mobile applications take complexity of supported platforms to the next level by adding devices. Ensuring that mobile apps are working on all type of devices (smart phone, tablets, and phablets) supplied by major brands (various models from Samsung, Sony, Nokia, HTC, Apple, etc.) and on all the platforms (iOS, Android, Windows, BlackBerry, etc.) is challenging. On top of that, new devices are hitting market so often that it becomes impossible to cover all the major devices.

3.7 Adaptability and limited space mean screen size is changing constantly

Pretty much, all the major players are changing screen sizes of their phones, tablets, and phablets to figure out what works or in response to the competition. How applications adapts themselves for various screen sizes, layout, and configuration is a challenging task.

Apart from adaptability to different screen sizes, mobile applications have to deal with the limited screen size. Limited screen size means that user can not be given many different options on a single screen – usability, similar experience, on-screen help, inability to use search or other applications easily, etc. – poses different challenges and as a tester one need to think beyond what is developed and always think of who will use it and in what circumstances.

3.8 Complex user interaction means more than one way to do everything

User interaction in desktop and browser based applications was pretty much limited to mouse and keyboard. Mobile applications on the other hand are trying to make user interaction as fluid as possible. We had touch screen and with new phones from Samsung, you can just wave your hand to give commands. Devices are smart enough to understand complex gestures, eye movement, direction, tilt, movement, acceleration, GPS coordinates, surroundings, sound, and so on.

In the desktop and browser world, applications were straightforward. They were either desktop or web applications. However, with the adoption and support of HTML5, the applications are merging. On all the mobile devices, it is not difficult to find HTML5 applications, native applications and hybrid applications. Testing for hybrid applications would be different from testing native applications.

4. TYPES OF MOBILE APP TESTING

Some or all of the following Testing types may be performed depending on your mobile testing requirements

- Functional Testing
- Performance Testing
- Security testing
- Usability Testing
- Compatibility Testing
- Recoverability Testing
- Laboratory Testing
- Memory Leakage Testing
- Interrupt Testing
- Installation testing
- Certification Testing
- Location Testing
- Outdated Software Testing
- Load Testing
- Black box Testing
- Crowd Sourced Testing

4.1 Functional testing:

Functional testing ensures that the application is working as per the requirements. Most of the test conducted for this is driven by the user interface and call flow

4.2 Performance testing:

This testing process is undertaken to check the performance and behavior of the application under certain conditions such as low battery, bad network coverage, low available memory, simultaneous access to application's server by several users and other conditions. Performance of an application can be affected from two sides: application's server side and client's side. Performance testing is carried out to check both.

4.3 Security testing:

To check for vulnerabilities to hacking, authentication and authorization policies, data security, session management and other security standards.

The fundamental objective of security testing is to ensure that the application's data and networking security requirements are met as per guidelines. .

Security and data privacy aren't optional. Users are worried about their data and credentials being exposed through vulnerable applications that happen all too often.

Applications should encrypt username and passwords when authenticating the user over a network. One way to test security related scenarios is to route your mobile's data through a proxy server like OWASP Zed Attack Proxy and look for vulnerabilities.

4.4 Usability testing:

Usability testing is carried out to verify if the application is achieving its goals and getting a favorable response from users. This is important as the usability of an application is its key to commercial success (it is nothing but user friendliness). Another important part of usability testing is to make sure that the user experience is uniform across all devices. This section of testing hopes to address the key challenges of the variety of mobile devices and the diversity in mobile platforms/OS, which is also called device fragmentation. One key portion of this type of usability testing is to be sure that there are no major errors in the functionality, placement, or sizing of the user interface on different devices.

Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

4.5 Compatibility testing:

Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired.

The following are the most prominent areas for compatibility testing.

1. To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible
2. To ensure that the text is readable for all users for the application
3. To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed

4.6 Recoverability Testing

1. Crash recovery and transaction interruptions
2. Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
3. Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
4. The validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection.

4.7 Laboratory Testing:

Laboratory testing, usually carried out by network carriers, is done by simulating the complete wireless network. This test is performed to find out any glitches when a mobile application uses voice and/or data connection to perform some functions.

4.8 Memory Leakage Testing:

Memory leakage happens when a computer program or application is unable to manage the memory it is allocated resulting in poor performance of the application and the overall slowdown of the system. As mobile devices have significant constraints of available memory, memory leakage testing is crucial for the proper functioning of an application

4.9 Interrupt Testing:

An application while functioning may face several interruptions like incoming calls or network coverage outage and recovery. The different types of interruptions are:

- Incoming and Outgoing SMS and MMS
- Incoming and Outgoing calls
- Incoming Notifications
- Battery Removal
- Cable Insertion and Removal for data transfer
- Network outage and recovery
- Media Player on/off
- Device Power cycle

An application should be able to handle these interruptions by going into a suspended state and resuming afterwards.

An application should be capable to hold these interruptions by going into a suspended state and restarting afterwards.

4.10 Installation testing:

Certain mobile applications come pre-installed on the device whereas others have to be installed from the store. Installation testing verifies that the installation process goes smoothly without the user having to face any difficulty. This testing process covers installation, updating and uninstalling of an application

4.11 Certification Testing:

To get a certificate of compliance, each mobile device needs to be tested against the guidelines set by different mobile platforms.

4.12 Location Testing:

Connectivity changes with network and location, but you can't mimic those fluctuating conditions in a lab. Only in Country non automated testers can perform comprehensive usability and functionality testing.

4.13 Outdated Software Testing:

Not everyone regularly updates their operating system. Some Android users might not even have access to the newest version. Professional Testers can test outdated software.

4.14 Load Testing:

When many users all attempt to download, load, and use your app or game simultaneously, slow load times or crashes can occur causing many customers to abandon your app, game, or website. In-country human testing done manually is the most effective way to test load.

4.15 Black box Testing:

This type of testing doesn't include the internally coding logic of the application. Tester tests the application with functionality without peering with internally structure of the application. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

4.16 Crowd Sourced Testing:

In recent years, crowd sourced testing has become popular as companies can test mobile applications faster and cheaper using a global community of testers. Due to growing diversity of devices and operating systems as well as localization needs, it is difficult to comprehensively test mobile applications with small in-house testing teams. A global community of testers provides easy access to different devices and platforms. A globally distributed team can also test it in multiple locations and under different network conditions. Finally, localization issues can be tested by hiring testers in required geographies. Since real users using real devices test the application, it is more likely to find issues faced by users under real world conditions.

5. CHALLENGES IN MOBILE APP TESTING

It is enormously challenging to verify enterprise applications across myriads of handsets with different screen sizes and operating system flavors.

Mobile Testing needs to be performed on variety of software platform, versions, hardware, network condition and what makes it more complex are the various devices, fast changing mobile OS, the frequent introduction on new devices, and unpredictability of simulators.

5.1 Diversity in Mobile Platforms/OSes:

There are different mobile operating systems in the market. The major ones are Android, iOS, Symbian, Windows Phone, and BlackBerry (RIM). Each operating system has its own limitations. Testing a single application across multiple devices running on the same platform and every platform poses a unique challenge for testers.

5.2 Device Availability:

Access to the right set of devices when there is an ever-growing list of devices and operating system versions is a constant mobile application testing challenge. Access to devices can become even more challenging if testers are spread across different locations.

5.3 Mobile network operators:

There are over 400 mobile network operators in the world; out of which some are CDMA, some GSM, whereas others use less common network standards like FOMA, and TD-SCDMA. Each network operator uses a different kind network infrastructure and this limits the flow of information.

5.4 Scripting:

The variety of devices makes executing a test script (scripting) a key challenge. As devices differ in keystrokes, input methods, menu structure and display properties single script does not function on every device.

5.5 Choosing how to test:

There are two main ways of testing mobile applications: testing on real devices or testing on emulators. Unfortunately, neither method is flawless. Emulators often miss issues that can only be caught by testing on real devices, but because of the multitude of different devices in the market, real devices can be expensive to purchase and time-consuming to use for testing.

5.6 Real Testing Device:

Testing on real device allows you to run your mobile applications and checks its functionality. Real device Testing assures you that your application will work smoothly in customer handsets

5.7 Emulators:

Emulator is a software program that allows your mobile to imitate the features of another computer or mobile software you want them to imitate by installing them to your computer or Mobile.

Emulators are available and are widely used for both manual and automated mobile application testing. They are powerful tools for developing mobile applications; they have unique features which enable them to provide a rich set of development tools as well as an integrated debugging environment and in most cases are available for free from the vendors. They fall into three main categories:

5.7.1 *Device emulators*

These are generally provided by device manufacturers and simulate the actual device. Device emulators take the place of the actual hardware and can provide an incredibly beneficial test environment.

5.7.2 *Browser emulators*

These simulate mobile browser environments. Whilst useful for determining the functionality available in a particular mobile browser, they are useless for device-specific testing.

5.7.3 *Operating System Emulators*

Microsoft provides emulators for Windows Mobile, and Google provides an emulator for Android. These run within a simulated mobile device environment and provide access to applications running within the operating system, e.g. a Web browser.

However, by definition, emulators are not identical to the real target environment. This happens in the world of aviation, where pilots are trained on emulators to test situations that they cannot simulate using real aircrafts. What is the difference between an emulator and a simulator? Emulation is a process of mimicking outwardly observable behaviour to match an existing target. The internal state of the emulation mechanism does not have to accurately reflect the internal state of the target which it is emulating. Simulation, on the other hand, does involve modelling this underlying state. The end result of a good simulation is that the simulation model will emulate the target which it is simulating.

Since mobile applications are used on real handsets and not emulators, clearly the closer you get to the actual platform during the QA process, the better quality you will achieve. On the other hand, using only real devices for development and testing may not make sense for every application. The trade-off in terms of convenience and cost versus the need for risk management and quality must be carefully considered in the testing strategy. Obviously, if no budget or logistical factors exist, using a real device is always a preferable option for testing.

5.8 Compatibility:

It is necessary to test the Compatibility: Suppose an Application can work on the high resolution and it doesn't work on the fewer lower resolution.

5.9 Should be able to Pick up the Phone:

It is necessary to check: While executing the app Application should be able to pick up call.

5.10 Variety of Mobile Devices:

Mobile devices differ in screen input methods (QWERTY, touch, normal) with different hardware capabilities.

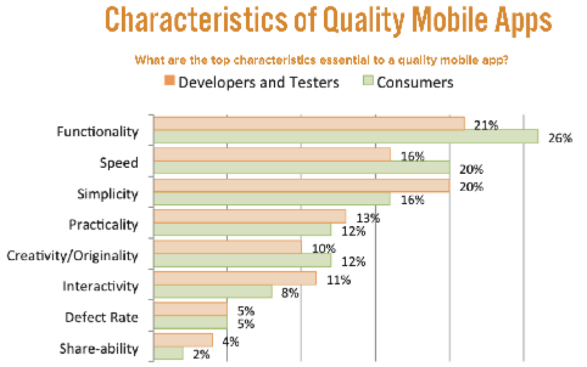

Fig1: Çharacteristics of Quality Mobile Apps
Source: SmartBear's State of Mobile 2014 webinar

6. MOBILE APP TESTING STRATEGY

6.1 Mobile Device Selection

The leading concern before beginning mobile app testing activities is to choose the ways of testing the app. This primary testing method can be a difficult decision to make as it corresponds directly to the market and reach for your app.

Within device selection, there are two choices to be made: selecting the gadget model or choosing between emulators and substantial devices. The factors below require consideration during device selection:

- **OS Version:** test your mobile app on all stable OS versions
- **Screen Resolution:** use a mix of different screens to test by size and resolution
- **Form Factor:** if the app is compatible with smartphones and tablets, test for form factors

As needed, numerous other factors such as memory size, connectivity options, etc. need to be accounted for while selecting the device model.

6.2 Emulators vs. Physical Devices

You can also decide among physical devices or emulators. Predominantly in the opening stages of development, device emulators are extremely useful because they assist rapid and efficient testing, especially in an agile development environment.

Device emulators are also very cost-effective. Mobile device emulators are very useful for basic application functionality testing and during feature development. They provide excellent options for network bypass, a pseudo-live environment, and test scripting languages.

Using mobile device emulators does not mean that you should avoid using physical devices altogether. Testing on physical devices is imperative; it allows the understanding of application activities in real-life scenarios. Mobile device testing is all about using a right mix of emulators and physical devices to get the best results, quickly, and efficiently.

Physical device testing leads to beta testing as an extremely helpful method of mobile app testing which gives you admittance to real-world testers, real devices, actual networks, and a wider geographic coverage.

Beta testing is a major area where emulators fail in comparison to physical mobile devices. Beta testing gives you a chance to test your mobile app for factors like:

- Network density
- How the app behaves on specific devices
- How real-world users interact with the app
- Different battery states on the devices
- Multiple networks (Wi-Fi, 4G, 3G, etc.)

The real-world testing environment in beta testing is nearly impossible to create in a test lab.

6.3 Network Connectivity

Network connectivity significantly affects mobile applications. The majority of mobile applications depend on network connectivity to work correctly. Therefore, testing mobile apps in actual network environments is necessary to get the real picture of the application's behaviour.

There is a multitude of network simulation tools available to test mobile apps against network speeds, bandwidths limitations, connection drop outs, and more. These network simulation offerings add exceptional value to the testing activities. For a mobile app to win, it needs to deliver consistent performance across diverse network environments.

6.4 Manual vs. Automated

Automation is key to successful regression testing during development stages. However, automated testing requires a substantial amount of initial investment.

Therefore, test automation should be done only in scenarios where:

- The application is growing
- The mobile development lifecycle is long
- The scale and frequency of regression testing are high
- A significant portion of test cases includes obtainable functionality test cases

With automation, mobile application testing for the following becomes easy:

- Verifying application compatibility with newly released operating systems
- Validating backward compatibility during application upgrades

Including manual testing scenarios and interoperability testing are very important when it comes to mobile application testing strategy. When testing a mobile application, the testing team ought to test the various events which may occur when the application is being executed – Incoming calls, SMSs, low battery, alerts such as emails and roaming. Lately, along with growing technologies, mobile users also take advantage of location-based and voice related apps – these rely on much more sophisticated use cases, which can also be tested manually.

Manual checks are a necessary part of mobile application testing, because there are conditions that aren't able to be reproduced or scripted with emulators. Further, automated testing leaves out chance and human error conditions that can create pivotal ah-ha moments for an application. There's no winner in manual versus automated testing for mobile applications. They're both required.

7. APPROACHES FOR MOBILE APP TESTING

7.1 Mobile App Testing on Cloud:

To test mobile applications use of the Cloud testing is one of the most successful choices. Using Cloud-based approach you can achieve the Native and Hybrid Mobile Apps automation testing.

Cloud-based application testing delivers the following benefits:

- Highly synchronized and pre-configured architectures
- Reduction in defects associated with unstable test configurations
- Saves businesses from setting up on-premise test conditions that take lots of time
- No additional needs for advanced tools, server settings, licensing
- Supports complex applications, which solves the problems of testing in-house
- Scalability to leverage an application's capacity threshold
- Results in real-time, which means defects can be analyzed while tests run

Cloud-based mobile application testing reduces time to market (TTM) and significantly augment testing competence.

7.2 Lab-based Approach:

In Lab-based automation testing, you need to set up different tools in a testing environment and leverage simulators or physical devices to automate testing using different tools. In the market there are different capability tools are available for a lab-based approach like EggPlant, Jamo, See Test, ZAP, and SilkTest etc. These automation tools can be used for continuous automation testing or to accomplish regression testing functionalities. Few of them also supports the non-functional testing features related to mobile apps.

7.3 Open Source/ Platform-specific Tools:

In QA automation testing strategy the testing to check platform-specific tools reliability should be carried out at least once. There are different mobile testing tools available in the market like Appium, Calabash, Frank, MonkeyTalk, iOS UI Automation, Robotium, iOS-driver, Ui Automator, KeepItFunctional and Selendroid etc. But before starting testing on Open Source/Platform-specific Tools you should have added programming skills to your team. Also, the mobile device simulators and emulators can be used for testing.

8. CONCLUSION

All above testing methodologies prove that users can trust on applications that come with mobile devices, all applications are completely tested with many testing methodologies.

In this paper we have mentioned “Types of Mobile Application Testing”. Considering the significant role the mobile applications plays, nowadays, in our day to day life, testing of these applications are going to evolve and thus they require a lot of testing to make them work as required. Testing in both the simulator/emulator as well as the real world devices is necessary to maintain strong standards and quality assurance.

Careful deliberation of both the pros and cons of mobile emulators and real devices, it would be worthwhile to reach at the conclusion that the optimal mobile testing solution for enterprises is neither putting all the eggs into the basket of the real devices nor putting them into the emulator but rather what we need is an optimum combination of both.

However, to avoid the costly scenario of releasing a business-critical application with defects, enterprises need to ensure that they perform the major part of their mobile testing on real devices before the application goes into production.

Each organization needs to strategize and plan carefully to determine at what stage to introduce real devices, they also need to decide how many devices are sufficient to cover market needs, and what could be the best possible option to adopt for managing those devices.

Best practices would indicate that actual development should use emulators (and a few reference real handsets) in order to speed up the debugging of the application during the coding phase, while sanity, Smoke Testing, performance, interoperability and network feasibility and Regression Testing should be done on real handsets.

9. ACKNOWLEDGMENTS

Our sincere thanks to Shri. Deepak Chandra Misra, Dy. Director General-NIC New Delhi who have supported us to write this paper.

10. REFERENCES

[1] Huang, J., and Gong, Y., "Remote Mobile Test System: A Mobile Phone Cloud for Application Testing", 2012 IEEE 4th International Conference on Cloud Computing Technology and Science

[2] https://smartbear.com/learn/software-testing/what-is-mobile-testing/ last accessed in mar 2018

[3] https://en.wikipedia.org/wiki/Mobile_application_testing last accessed in mar 2018

[4] http://developer.android.com/reference/android/test/AndroidTest-Case.html last accessed in mar 2018

[5] https://testobject.com/wp-content/uploads/ ebook_mobile_app_testing.pdf last accessed in mar 2018

[6] Ravi Ramchandra Nimbalkar,Mobile Application Testing and Challenges, International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064 Volume 2 Issue 7, July 2013

[7] Mohammed Akour , Bouchaib Falah Ahmad A. Al-Zyoud, Salwa Bouriat, Mobile Software Testing: Thoughts, Strategies, Challenges, and Experimental Study, *International Journal of Advanced Computer Science and Applications, Vol. 7, No. 6, 2016*

[8] Ahmed, Maryam & Ibrahim, Rosziati. (2014). A Comparative Study of Web Application Testing and Mobile Application Testing. Lecture Notes in Electrical Engineering. 315. 10.1007/978-3-319-07674-4_48.

[9] Anne Kaikkonen, Aki Kekäläinen, Mihael Cankar, Titti Kallio, and Anu Kankainen. 2005. Usability testing of mobile applications: a comparison between laboratory and field testing. Journal of Usability Studies, volume 1, number 1, pages 4-16.

[10] https://www.qualitestgroup.com/white-papers/mobile-emulators-vs-real-devices last accessed in mar 2018

[11] Justin Mifsud ,2016,Usability Testing Of Mobile Applications: A Step-By-Step Guide, https://usabilitygeek.com/usability-testing-mobile-applications/ last accessed in feb 2018

[12] Pallavi Bhuarya1, Shruti Nupur, Anuradha Chatterjee3, Rajesh Singh Thakur,"Mobile Application Testing: Tools & Challenges", International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 5 Issue 10 Oct. 2016, Page No. 18679-18681

[13] Abhinav Kahuria,Anu Gupta,,"Challenges in Anroid Application Developemnet:A Case Study ", International Journal of Computer Science and Mobile Computing ISSN 2320 –088X, Vol. 4, Issue. 5, May 2015, pg.294–299

[14] Anureet Kaur,"Review of Mobile Applications Testing with Automated Techniques",International Journal of Advanced Research in Computer and Communication Engineering ISSN (Online) 2278-1021 ISSN (Print) 2319 5940,Vol. 4,Issue 10 , October 2015.