International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

Performance Testing Framework for Software Mobile Applications

Mohd Abdul Hakeem[1], Mohammed Abdul Razack Maniyar[2], Prof. Mohd Khalid Mubashir Uz Zafar[3]

Team Lead, Accenture, Hyderabad, India[1]

Ph.D. Research Scholar, Department of Physics, Dravidian University, Kuppam, India[2]

Professor, Maulana Azad National Urdu University, Hyderabad, India[3]

ABSTRACT: Performance testing of Mobile applications is an important activity in the Mobile application development life cycle. A thorough testing of the mobile application with respect to its performance has to be done before it is goes live. This can be effectively done if there exists a Performance testing framework specific to software Mobile applications. This paper basically discusses on the Mobile testing framework which caters the End-to-End Performance testing needs of Mobile applications. The Framework is more generic with respect to its feasibility and efficiency for various technology platforms. The various areas and modules of performance testing like profiling, diagnostics, load generation, analysis and reporting are included in this framework. An effort is made to integrate all these modules and cater the performance bottlenecks identification needs in a single 'Go' or 'flow'. This framework also extends it features to better fit in every environments and also compatible with different technology, platforms and tools. Freeware's or open source tools can also be used in this framework, thereby making it a freeware framework for the performance testing of various mobile applications or products.

KEYWORDS: Performance Testing, Mobile Applications, Performance testing Framework, Performance Testing Tools, Mobile Testing Framework, Mobile testing tools.

I. INTRODUCTION

In the recent years there is seen a tremendous advancements in the industry of Mobile technology. A lot of technology evolution has been seen in the past few years in this area and lot many number of mobile applications and products were introduced in the market and now it is in the hands of each and every people. The statistical figure of the users of these mobile applications has also seen a steep rise and now it has become a part and parcel of everyone's life. With the evolution of technology, a lot of mobile applications and products were introduced into the market. With the increase in the user base of these applications, it is now becoming very much necessary to keep an eye on the quality of these applications and products. For a mobile application or product to make a place in market, it has to maintain a good quality and satisfy the customers or end users. As people have become more quality conscious, it is now necessary to take into consideration each and every factor and feature of such applications. Apart from the mandatory or general functional or UI testing, it is necessary to go for the Non-Functional testing, especially the performance testing, which would help in gauging the performance of the applications and thereupon fine-tune the performance of the mobile applications or products by addressing the identified bottlenecks.

A lot of studies were conducted in this area of Mobile application development and testing. Many proposed strategies, theories and approaches were also presented in support of these applications. Many approaches and strategies were also suggested and proposed by many research scholars for Performance testing of Mobile applications. But most of them were done and addressed the Functional areas of the applications and its testing. Very few or limited studies were done in the Non-Functional Testing area, especially the Performance testing for the mobile applications. Many frameworks were presented for Mobile performance testing as well, but due to the rapid advancement in the technology and with the introduction of many tools in market, the studies become limited and failed to address the new technological pattern or designs. A lot of gaps were observed and it failed to cater the performance testing needs and requirements to the full extent, thereby failed in impressing the customers or end users with respect to the quality and seamless performance of the mobile application or product. Also, the new advanced technology tools which crept in market in recent times can also be used to avoid compatibility issues with the newer versions or releases of the software environment.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

This paper is an attempt to address the gaps identified due to the technology updates and at the same time fulfils the current needs and demands with respect to the performance of the mobile applications or products. The framework designed is a generic one and can be used in various environments and more customizable. It includes all the modules which are necessary for the Mobile Applications End-to-End activity. It facilitates in driving through different modules like, load generation, profiling, diagnostics, monitoring, analysis and reporting in an integrated fashion. Varieties of tools belonging to these areas were also introduced and its use is also explained. The feasibility, compatibility and multi-facet features of this framework are better explained with the case studies.

II. RELATED WORK

Shiping C et al. [1] have proposed a general purpose testing framework for the performance testing of software applications. The proposed framework can be used for simple, small, complicated and large scale performance testing. It also facilitates the testing (Performance testing) by separating the software application logic from the very common performance testing functionalities. The proposed study was prototyped on Java and .NET platforms which can be used for various performance testing projects. Three different entities named- Configurations & Scripts, Test drivers and Test Results are presented in this model. These three entities interact with each other. The configurations and scripts are provided as input and the test driver generates a specific amount of load based on the inputs provided. The test driver after finishing the test collates the results and the final summarized report is generated. This performance testing framework has been implemented on Java and .NET platforms and was demonstrated with a couple of examples. The metrics used while demonstrating the prototype are, Response Times, Latency, Throughput and other resource utilities like- CPU, Memory, Disk I/O and network traffic. To separate the application logics from other components, IRunner is used, which is a common interface. This interface pulls the common requirements for testing a variety of applications or technologies.

Peeysh P et al. [2] have studied on various frameworks across different platforms with their features and other characteristics for the mobile application testing. A comparative study was done on five important and top listed mobile application frameworks like Espresso, Robotium, Appium, Calabash and UIAutomator. The technical feasibility was studied along with their pros and cons. Different factors like, Operating System, Scripting language; test creation tools, Supported API levels and the Community were taken into account while carrying out the comparative study. The study finally concluded with the statement that choosing and applying the best framework for the mobile application testing is a very difficult task as every framework has different features and it also has some constraints. The study says that the following features must be considered for choosing a best suitable framework. The features include- High Scalability, Selective test script execution, Supporting test execution of the test scripts on the device without any connectivity to the machine, Multi-device testing, Functionality on both simulator/emulator and device, Supporting Data Driven Testing (DDT) and other reusable functions, Robust, Extensible to support test automation of Native and web applications across different platforms like Android and iOS, Reducing overall costs to the customer and provision of detailed reporting along with screenshots, logs and pass/fail details.

Bakhtiar A et al. [3] have presented a study to answer various questions related to mobile applications testing techniques, whether to go ahead with Manual or Automated Testing. This study talks about the tests which were executed for both, Manual and Automated case studies. Monkey Talk was used to identify errors and bugs in these case studies. It would be difficult for a software tester to decide which testing technique has to be applied, either to apply Manual testing techniques or automated testing techniques. To plan this, the tester has to do an investigation on tool's considering the limitations and also by considering the objectives of the test. A distinctive study was done between Manual and Automated testing for Mobile applications. The test levels and testing scopes were also defined for the Mobile application testing. The test environments were also defined for the real device and for the simulator and the simulator-based approaches were also defined in this study. The open source tool, Monkey Talk from Gorilla Logic which is operatable on both iOS and Android was used for case studies. The study finally concluded saying that for obtaining quality and satisfactory feedback on Mobile applications throughout the applications development process, it is necessary for the software testers to adopt effective models, methods and techniques. It recommends for both Manual and Automated testing of Mobile applications to cope with the fundamental necessity. It however concludes with the statement that automated testing is one of the most efficient method to guarantee the quality and performance of the mobile application.

Vikrant N et al. [4] have presented a framework called 'Appstrument' which is a unified framework for instrumenting the Mobile applications to make them ready for the functional, accessibility and Performance testing. The presented framework has the feature of allowing instrumenting the mobile application to make it ready for either

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

a single category of testing or a combination or 2 or more of these categories with multiple optional features. The Appstrument presented was deployed and tested on iOS and Android platforms and could succeed in instrumenting a sizeable number of applications and effectively playback the user defined test cases automatically to collect relevant and metrics corresponding to each category of the testing. The architecture of this framework has 5 different components – 1. Appstrument Test Case Repository (TR) 2. Appstrument App Centre (AC) 3. Appstrument Instrumentation Module (IM) 4. Appstrument Playback Service (PS) and 5. Appstrument Reporting Engine (RE). Following categories of applications were tested using this proposed framework – Android-Native, Android-Hybrid, iOS-Native and iOS Hybrid applications.

A.A. Menegassi and A.T. Endo [5] have proposed the mechanisms to develop automated tests for cross-platform mobile applications. To set up in multiple configurations, a two reference device approach was adopted – one running on the Android platform and the other running on iOS platform. A prototype tool, 'x-PATeSCO' was developed to support this proposed approach. This approach was also evaluated with Nine Cross-Platform mobile applications comparing the locating strategies in the six real devices. The approach has three main steps – 1. Device Selection, 2. UI element selection and test case definition and 3.Single Test Engine. The approach proposed is finally implemented in a prototype developed, x-PATeSCO (Cross Platform App test script Recorder), which is based on the open source framework, Appium which is used to automate tests in Native, Web and Hybrid mobile applications. Appium being a cross-platform makes it possible to automate tests for iOS and Android platforms using the Selenium WebDriver API. An experimental evaluation was conducted to compare the eight locating strategies, six individual expressions and two combined strategies. One of the limitations identified during evaluation is that x-PATeSCO only implements a sub-set of all events that can be performed in mobile applications, so it is not possible to fully evaluate its expressiveness.

Vijaya Shetty S and Sarojadevi H [6] have presented study on the performance evaluation of Mobile and cloud applications. The data transactions between the Android mobile application and the online database were profiled and analysed for the performance issues. The profiling methodology uses open source profilers which is a cost effective one. To achieve a better performance, these profiling results can be used for optimizing the cloud based android applications. To achieve a good performance using the DDMS (Dalvik Debug Monitoring System) profiler and New Relic profiler, the SCALIBS (a kind of book store application) is optimized. The methodology used adopted the following steps – (a) Creating an online database in a registered server (b) Creating an android apk by using the developer android tools (c) Running the apk (Scalibs) in the android devices (d) Analysing the performance of Scalibs using DDMS and New Relics and (e) Collecting the results and analysing the application till the desired level of performance is achieved. Apache Jmeter, which is an open source tool, has also been used in this study as a performance evaluation tool.

Shengqian Y et al. [7] have proposed an approach for testing the poor responsiveness in Android Mobile applications. They have presented a test amplification approach which exposes and quantifies the root causes of responsiveness issues. The proposed technique seems to be highly effective in identifying these types of performance bottlenecks in the Android applications. Contribution of the study includes- Test amplification criteria, Test amplification and execution and evaluation with case studies. Four case studies were presented based on the failing test cases in this study. The case studies presented are, Connectbot, K-9 Mail Client, VL Media Player and Astrid. An important question raised by this study is as to how to proactively prevent the important performance bottleneck (responsiveness issues) through the design principles and patterns and with the help of automated techniques like-code transformation techniques. The study also suggests that the responsiveness issues or defects often occur in Android mobile applications regularly and this proposed technique suggested can be highly effective and helpful in identifying it.

Dong-Han H et al. [8] have studied upon the Frameworks and models for identifying as well as organizing usability impact factors of mobile phones. They have proposed a conceptual framework which talks about five different views reflecting different aspects of the user interactions with the mobile phones from which various impact factor models can be derived. The five different views presented are – Product view, User view, Dynamic view, Interaction view and Execution view. A hierarchal model was also developed which organizes the usability factors in terms of the goal-means relations. The usefulness of the proposed framework and model was studied through two case studies. Also, a set of checklists were also developed to measure the usability of the mobile phones which helps in improving and increasing the practicability of the framework. The hierarchal model presented has five different levels – Usability (Quality in use), Usability Indicator, Usability Criteria, Usability Property and Usability Data. The study finally concluded saying that there are a lot of factors which affects the usability of the mobile phones. Taking into account all these factors in a unified way, the usability has to be designed and evaluated.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

Anita A and Kire J [9] have done a performance evaluation of mobile applications and examined the applicability of Computational offloading. The study also details on the analysis of the utilization of the resources and its execution time on the mobile devices. The study has concluded saying that the tasks with constant complexity can better be executed on the device. These identified tasks are simple and also local time is far shorter that the offloaded computation. The common thing noticed in these tasks is that they use very small amount of RAM and CPU on the targeted mobile device. The memory usage was seen below 30% and the CPU below 50%. The mobile application performance metrics used in this study are – Memory Usage, Delay, CPU usage, Battery lifetime and Network. The study was done on Samsung, Huawei and LG mobile devices and the values for the metrics were captured and analysed.

Sadiq M et al. [10] have done a survey on the most common preferred performance testing tools. A matrix is proposed in this study based on the survey to verify the practical implementation of the tools for the Performance testing activity. This way, the gaps were identified and guidelines were provided for identifying and developing good performance testing tools. Different parameters used in this study include – Efficiency, Integrity, Reliability, Survivability and Usability. The automated Performance tools used in this survey are Grinder, Apache Jmeter, Silk Performer, Mercury Interactive LoadRunner, IBM Rational Performance Tester, Open STA and HP LoadRunner. To measure the performance effectiveness of the tools, the most common parameters used for the study are – Response Time, Throughput, Latency, Scalability, Resource Utilization and Security. A comparative study was done on these tools taking into consideration the parameters mentioned above. The performance parameters used or proposed in this study adhere to the IEEE and ISO standards.

Mendez-Porras A et al. [11] have presented a review on the automated testing of Mobile applications. As software testing and automated testing of Mobile applications is very much needed to ensure the quality of the applications, this study mainly focuses on the best practise needed for carrying out the automated Performance Testing of the Mobile applications. The study contributes the following for the Mobile application testing- (a) It helps in identifying the main approaches for the automated testing of mobile applications and the main research trends over time (b) It also analyses the available evidence on the automated testing of mobile applications regarding its usefulness and accuracy. The study says that there are many tools proposed for the testing of mobile applications, but most of them are not available online for download. And also, for the available downloaded tools, there is a lot of complexity in its use and the user manual is also somewhat not user friendly. And some tools are more specific and can be used to test only limited features of the mobile applications. It says that a systematic literature review of evidence has to be done to determine the experimental designs and the use of metrics to provide empirical evidence in the studies related to the mobile applications testing.

Jozef Goetz and Michael Ruvalcaba [12] have evaluated the performance of Mobile applications on different platforms. The study presents a comparative review of creating mobile applications on different platforms using two IDEs (Integrated Development Environments) and one common IDE. The two different platforms used in this study are Android and iOS. The mobile applications created were evaluated based on the criteria points – Execution Time, Application size, Lines of Code, Memory usage, CPU usage, Data usage and the ease of IDE. The testing sequence used in the comparison in this study is – (a) Launching the application on the device (b) Running the application at the default settings (c) Recording the time results from the results screen (d) Repeating from step ‘b’ at least four times and keeping a track of the measurements available from the IDE (e) Changing the amount of simulations per execution (f) Repeating from step ‘b’ again until 20,000 simulations per execution is completed (f) Gathering the data from the devices and finally (g) Analysing the data.

III. PERFORMANCE TESTING FRAMEWORK HIGHLIGHTS FOR MOBILE APPLICATIONS

The Performance Testing Framework for the Mobile applications targets the four main areas (Fig.1.). They are Device Performance, Web Application Performance, Servers or Resource Performance and Network Performance.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

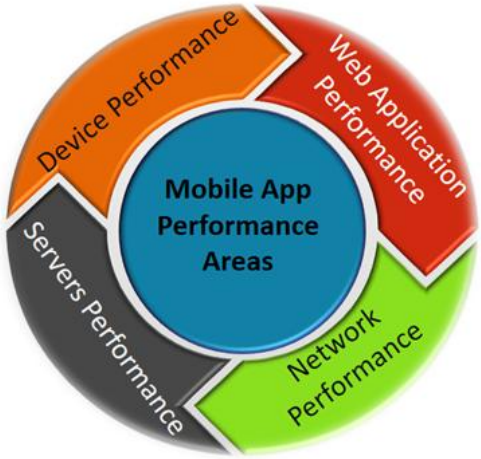

Fig.1. Four main target areas of Mobile application Performance Testing Framework

(1) *Device Performance:*
Under device performance testing, the properties tested are – Application Start up Response time, Memory consumption of the device, Battery Performance (Identification of Battery drainers), Software or Hardware variation on different devices, testing applications which are run in background, Wake lock detections and Sensors testing.

(2) *Web Application Performance:*
Under Web Application performance testing area, the properties tested are – Response times for various transactions, Janks in the UI, Application performance on different browsers and versions of OS and Business Transaction rate.

(3) *Servers or Resources Performance:*
The different servers and resources involved in the architecture and set up are tested for the properties like, Requests per second (RPS), Time to Last Byte (TTLB), Time to First Byte (TTFB), Throughput, Server CPU, Server Memory and Server Storage.

(4) *Network Performance:*
Under the Network performance area, the properties tested include, Latency, Throttle Network conditions and the performance on different networks – Edge, 3G, 4G etc.

Following Hardware and Software configuration is required for the host/test machine (Fig.2.).

Hardware	**Windows**
Minimum RAM	8 GB
Recommended RAM	**16 GB**
Minimum CPU	Intel i5
Software	**Windows**
OS	64 bit version of Microsoft Windows 8, 8.1,10
Mobile	**Android**
Mobile OS	API 17 (4.2.x) and above

Fig.2. Hardware and Software configuration details for Test Environment set up.

Following mobile Platforms with their file extensions can be used in this Mobile application Performance testing framework (Fig.3.). The three important and popular mobile platforms- Android, iPhone & Windows Mobile can be used in this framework.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

Mobile Platforms	File Extensions Used
Android	.apk file
iPhone	.ipa file
Windows Mobile	.cab file

Fig.3. Mobile Platforms with their file extensions.

The different types of tools used in the proposed Mobile application Performance testing framework are given below.

(a) Android Emulator Tool.
(b) Wake Lock Detector tool.
(c) Profiling Tool
(d) Network Monitoring tool
(e) Power Usage Monitoring tools
(f) Diagnostics Tools
(g) Android Monitor tool
(h) Mobile App optimizing tool

1. 'ADB' (Android Debug Bridge) is used for Android emulator. It is a command line client server tool.
2. 'Wake Lock Detector Lite' is the tool which is used for checking various types of wake locks. Different types of wake locks detected using this tool are – Partial Wake Lock, Screen Dim Wake Lock, Screen Bright Wake Lock & Full Wake Lock.
3. 'Trepn Profiler 6x' (From Qualcomm) is used as a Profiling tool for checking the performance of the mobile web application. This tool helps in identifying the applications that hog the CPU, consumes excess data and which drains the battery. It gives real-time view of individual CPU cores.
4. 'Micro Focus Network Capture Express 3x', is used as a network monitoring tool for mobile web applications. This tool measures and records the network latency, packet loss and downstream bandwidth.
5. 'AndroSensor 1.9.4' (By Fiv Asim) and 'Sensor Box 6.3' (By iMobLife) are used as Power usage monitoring tools.
6. 'Dumpsys' in Android is used for diagnostics purposes.
7. 'Android Device Monitor' is used as an Android monitoring tool. It is used for monitoring the CPU usage, Memory usage, GPU usage, and Network traffic and log messages.
8. 'ARO' (Application Resource Optimizer) by AT&T is used as a mobile application optimizing tool. This tool is used to find the wasteful data and power drain sources. It provides suggestions on fixing persistent performance problems and collects traces from the test devices and Emulators.

The Mobile application Performance tuning areas under this framework includes the following-

(a) Radio Resource Control
(b) HTTP Pipelining
(c) JavaScript Execution
(d) Browser Cache
(e) CPU (Weaker)
(f) Storage Space
(g) Network issues
(h) Power and
(i) Variety of Sensors like – Ambient Temperature, battery Temperature, Barometer, Magnetic Field, Light Sensor, Gyroscope, Accelerometer, Cameras, Microphone, GPS, Touch, NFC, Heart rate monitor, Bluetooth, WiFi.

IV. CASE STUDY

Android Studio with emulators is used in the framework for customizing and designing the scripts and for pulling the required statistics.

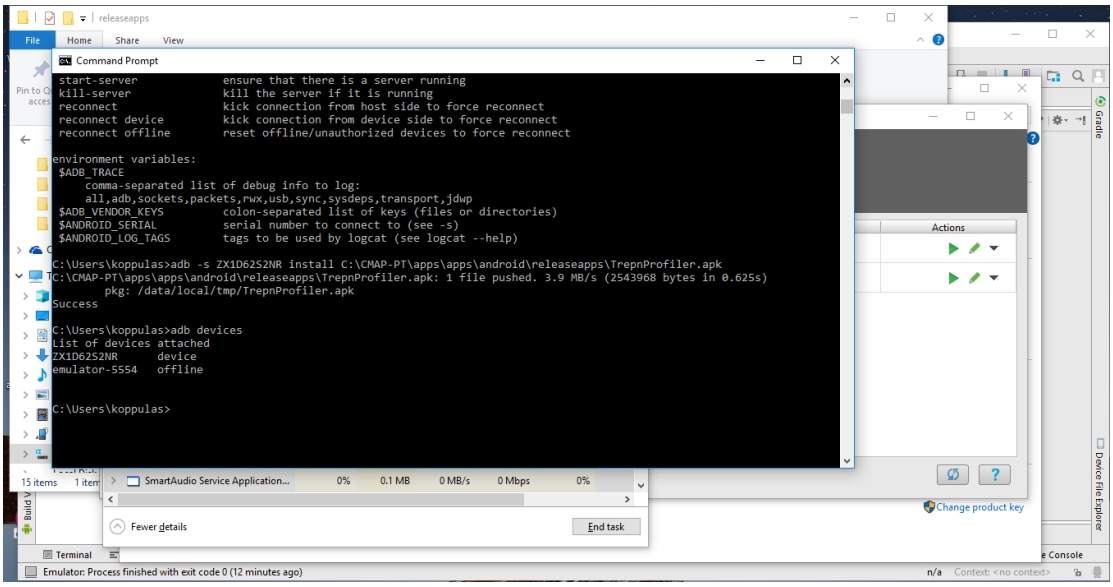

Fig.4. Screenshot showing the device connectivity.

Fig.4. shows the device connectivity for testing. The device connectivity is checked after starting the Android studio. The CPU statistics for the resource are also pulled up for the mobile device under testing (Fig.5.).

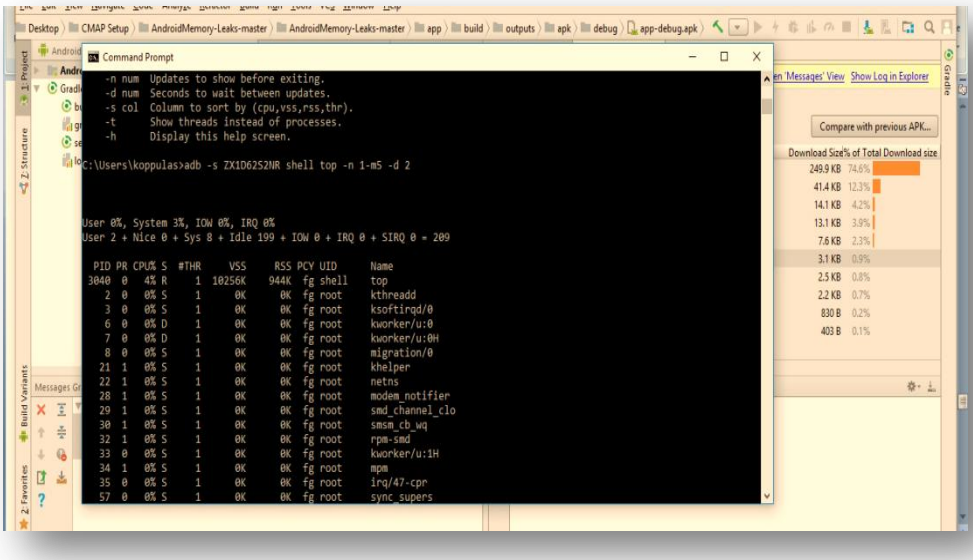

Fig.5. CPU Statistics for the resource.

The Wake lock detection is tested using the Wake Lock Detector Lite tool. The tool gives the Total awake time, Deep sleep time Screen-on time (Fig.6.).

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

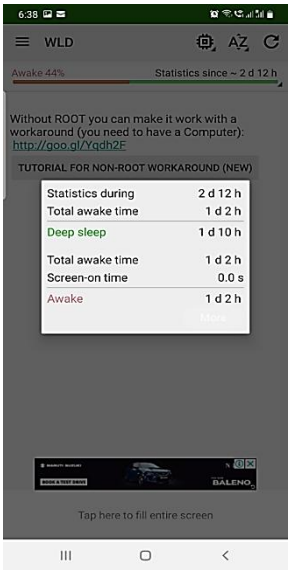

Fig.6. Wake Lock Detection at Wake Lock Detector Lite tool.

The Network related statistics are obtained through the Micro Focus Network Express Capture app (Fig.7.). The packet loss and latency issues can easily be identified through this tool for the mobile application under test.

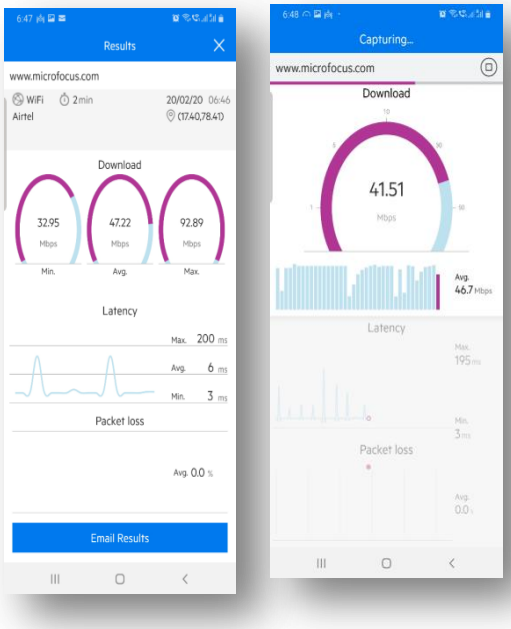

Fig.7. Network statistics being captured at Micro Focus Network Express Capture App.

Web page related statistics with the component breakdown for the software mobile applications captured during the testing using this framework are shown at Fig.8.

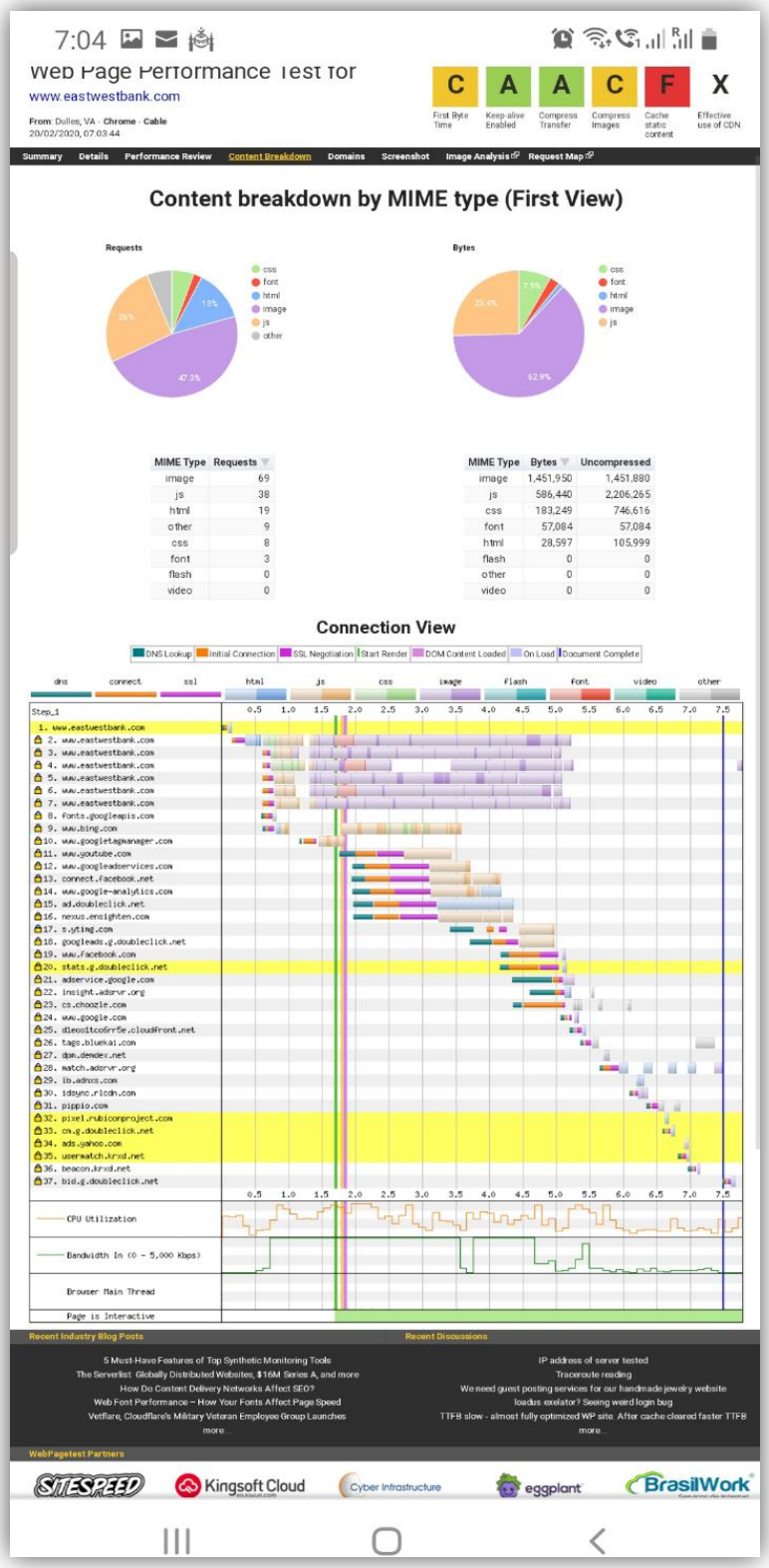

Fig.8. Web page statistics with component breakdown for Mobile application under test.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 7, July 2020||

V. CONCLUSION

With the advancement of technology every day in the Mobile industry, it is very much needed to focus on the quality of the software mobile applications with respect to its performance. The mobile applications have to be tested End-to-End for the Performance bottlenecks to achieve a seamless performance which is generally expected by the end users. The proposed framework caters the performance testing needs of these software mobile applications. This framework can be applied within the general mobile performance testing cycle and can be customized according to the need and testing requirement. Various modules of Performance testing like the load generation module, diagnostics, profiling, monitoring, analysis and reporting modules are also a part of this framework which is facilitating a lot in covering the End-to-End performance testing activity. This framework has also a lot of open source tools, This way it is also cost effective and can a part at the PTCoE (Performance Test Centre of Excellence). With the advancement of the technology, new tools can also be added to this framework to their respective modules with a little bit of tool analysis.

The proposed framework is compatible and customizable for most of the tools and platforms. However there are some limitations with respect to the usage of emulators, simulators. They are - The Hardware features like accelerometer, gyroscope, camera, microphone, proximity sensors and external accessories may not be simulated and also it is not possible to simulate or emulate battery issues using this framework.

REFERENCES

[1] Shiping C, David M, Surya N and John Z, “Yet Another Performance Testing Framework”, 19th Australian Conference on Software Engineering, April 2008.
[2] Peeyush P, Rajneesh C and Harshita B, “A Comparative Study of Mobile Application Testing Frameworks”, 10th Biyani International Conference (BICON-15), Sep 2015.
[3] Bakhtiar A, Sardasht M and Joun L, “Mobile Application Testing Matrix and Challenges”, ifth International Conference on Computer Science and Information Technology, Vol. 04, Apr 2015.
[4] Vikrant N, Vijay E and Vivek S, “Appstrument - A Unified App Instrumentation and Automated Playback Framework for Testing Mobile Applications”, 10th International Conference, MOBIQUITOUS, Dec-2013.
[5] A.A. Menegassi and A.T. Endo, “Automated Tests for Cross-Platform Mobile Apps in Multiple Configurations”, IET Software, Sep 2019.
[6] Vijaya Shetty S and Sarojadevi H, “Performance Evaluation of Cloud and Mobile Application”, ‘International Research Journal of Engineering and Technology, Vol. 05, Issue 05, pp. 2974-2979, May 2018.
[7] Shengqian Y, Dacong Y and Atanas R, “Testing for Poor Responsiveness in Android Applications”, 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS), pp. 1-6, May 2013.
[8] Dong-Han H, Jeongyun H, Peter F, Willian W, Sanghyun P, Chiwon S and Mike B, “Conceptual Framework and Models for Identifying and Organizing Usability Impact Factors of Mobile Phones”, Interacting with Computers, Aug 2009.
[9] Anita A and Kire J, “Performance Evaluation of Mobile Applications”, XIV International Conference – ETAI, Sep 2018.
[10] Sadiq M, Shahid Iqbal M, Amizah M and Wan Ainun M O, “A Survey of Most Common Referred Automated Performance Testing Tools”, ARPN Journal of Science and Technology, Vol. 05. No. 11, Nov 2015.
[11] Mendez-Porras A, Quesada-Lopez C and Jenkins M, “Automated Testing of Mobile Applications: A Systematic Map and Review”, XVIII Ibero-American Conference on Software Engineering, Apr 2015.
[12] Jozef Goetz and Michael Ruvalcaba, “Mobile Application Performance for Different Platforms”, International Research Conference on Engineering and Technology – IRCET, Jun 2017.