

Novel Framework for Automation Testing of Mobile Applications using Appium

Ashwaq A. Alotaibi, Rizwan J. Qureshi

Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Email: Ashwaqawad@yahoo.com, rmuhammad@kau.edu.sa

Abstract—Testing is an important phase in system development to improve the quality, reliability, and performance of software systems. As more and more mobile applications are developed, testing and quality assurance are vital to developing successful products. Users are widely opening browsers on mobiles to interact with web pages. It is important to ensure that our web application will be compatible with mobile browsers and applications. There is a need for a framework that can be applied to test different mobile browsers and applications. This paper proposes a novel framework for automation testing of mobile applications using Appium tool. The novel framework uses automation testing to reduce effort and speed up the process of automated testing on a mobile device. Survey is conducted to validate the proposed framework.

Index Terms—Automated testing, mobile applications, iOS, Android, Appium tool.

I. INTRODUCTION

Mobile technology now takes important part in everyday life. There are more mobile applications produced in a short time. Most people have started using browsers on mobiles to open web page/s rather than using computers. It is important to ensure that our web application will be compatible with mobile browsers and applications [1]. The application developer needs to deliver best Applications across different platforms within a quick time. Testing is an important phase in the software development life cycle to improve the quality, reliability and performance of the system. Testing of mobile applications (could be done by manual execution that) used visual verification to verify the results. However, it is very time-consuming. The major factor in reducing the testing effort is automation. Automation testing uses software to execute tests and compare actual and expected results. Automated testing helps to develop a high-quality, strong and reliable software. It provides several benefits to a software development company such as improvement in testing efficiency, effectiveness and delivery. The mobile automation testing is an extremely efficient approach to test such applications.

Automation of software testing is the new trend to ensure a high performance application in a short period. Appium tool helps to deliver powerful features. It allows

performing tests for different applications directly on the mobile device [2].

Appium tool helps to make automated testing for native, hybrid and web mobile applications.

So, there is a need of a framework that can be applied to test different mobile browsers and applications.

This paper proposes a novel automated testing framework for mobile Applications. The novel framework will enable the tester to perform automated testing of different browsers and applications on mobile devices using Appium tool.

The remaining parts of the paper are arranged as follows: Section II covers related work. Section III describes the research problem. Section IV proposes a novel framework for automation testing of mobile applications using Appium. Section V presents the validation of the proposed framework.

II. RELATED WORK

Rafi et al. [3] discuss the benefits and restrictions of automated software testing. The survey tries to close gap between academia and industry. It finds the benefits comes from stronger sources of evidence while restrictions often comes from experience reports, it also shows the benefits are related to test reusability and save effort in execution phase while the restrictions are found in the automated setup process, choose the tool and lack of experience. The survey does not include the aim of automation testing (test execution, analysis, levels and technology used). In the future work, it should conduct survey using these different dimensions.

A study shows that automated scripts for testing are more efficient, accurate and cost effective over the manual testing [4]. It shows manual testing is error prone compared to automated testing. The mistakes committed during manual testing process are minimized as academic skill sets augments. It recommends concerned stakeholder to consider regression for risk in automated scripts to get better efficiency and accuracy. It is limited to a kind of custom made scripts at the top of QTP v11.5 using VB scripting as a scripting language. Therefore, the finding might not apply for all types of testing tools.

A study performs an investigation on test automation frameworks [5]. It indicates that a large part of the impediments are linked to centralized the environment, how to use a framework and its components and also

related to using of software development tools. On the other side, it finds most of help requests about how to design scripts. It proposes to make further researches for the advantages of centralization environments and tools, organized deployment and use of test automation in an efficient way. It determines the challenges that faced users and developers in automatic software testing without solving it. In the future works, it should make investigations into how to prevent these problems.

A study compares Selenium, Testcomplete and Quick Test Professional (QTP) automation testing tools [6]. It shows that QTP is a versatile tool for a critical and more risky application under test. Selenium can also be used when don't want to spend on testing tool. QTP is the best tool among the three. Evaluate and compare done only between Selenium, QTP and Testcomplete. There are many software testing tools to make a study on it.

Hanna et al. [7] make comparisons between various scripting techniques that used in automated testing. The study presents an overview of different scripting techniques. It shows using these techniques will reduce the costs, speed up the process, and the products will take a short time to be delivered. It is briefly compared between the main features of various scripting techniques. Comparison between different scripting techniques in details should be conducted in the future work.

Rao et al. [8] analyze software automation testing, definition, characteristics and functions. The study proposed method to improve the overall testing process.

Automated testing is needed for an experience and the testers cannot be able to automate all the tests in a short period. It shows automated testing to be success requires developing suitable test program. It suggests taking into account the risk of implementation, resources and objectives to make good strategies. It should define the feasibility of automated testing, make risk assessment of the testing process and carefully develop test plan. Combination of manual and automated testing can improve the automated testing process. It generally discusses characteristics and functions of software automation. Also, it proposes some points to improve the automated testing process.

Jain and Sharma [9] make a keyword driven test automation framework study with design and develop keywords. It shows that by using keywords it easily and quickly to create test cases. The keyword driven framework will execute these keywords on some simple web applications. The results can be used by test engineers and users for their reporting or other purpose. It creates some keywords for web applications and proves that using these keywords will decrease the execution time. For future work, develop repository contains keywords for a specific domain and requirements.

Shaikat et al. [10] make taxonomy and a comparative study of different testing tools that available in the market to ensure the quality of software products. The results are generated from the data received from different companies. Interviews and test to verify the performance of tools not done and also experiments results are not presented in this research. For future work,

interviews, test performance and experiments results should conduct.

Singh et al. [11] describes a methodology that provides immediate test feedback to the software developer which allows achieving continuous testing during the development process. The Test Orchestration System useful for a pipeline of test phases. Continuous delivery of software in an industry requires periodic testing of various types. It is a new approach to detect problems early that speedup testing process. Test Orchestration is an attempt to improve test automation and it uses the static machine for test execution. The future plan is making a Test Orchestration system cloud based, so the resources can be used in the best way.

Kumar et al. [12] compared between Load Runner and QTP based on the efforts in creating scripts, capability to run the scripts, cost, speed and result reports. The study analyzes the features that help to minimize the used resources for script maintenance and features that the increasing probability of reusing the scripts. It shows Load Runner tool will be a good choice for applications that need less security and QTP is suitable for applications when data security is important. Comparison is done only between QTP and Load Runner tools. Comparative study using different automated testing tools should be done.

A study shows a comparison between QTP and Ranorex automation tools [13]. In case, if a project is already implementing QTP and there is no plan to add new modules then switching to Ranorex would be unfruitful. But, if the project is to be expanded and new licenses of tools are to be procured for the Project, switching to Ranorex is highly advisable. Comparison of automated testing tools is limited on Ranorex and QTP. There is a need to compare between different automation testing tools.

Dubey and Shiwani [14] tried to solve and adds s new ideas to support testing by Ranorex and TestComplete. They provided a comparative study between Ranorex and Test Complete automated tools depend on concepts and features like creating test scripts, capacity to reuse the scripts and result reports. The main objective of this paper is to examine the features supported by these two testing tools that help in decreasing the used resources in script maintenance and efficiently reuse script. Ranorex is the best tool for web based applications. Comparison is done only between Ranorex and TestComplete. In the future works, Comparative study using different automated testing tools should be done.

Shah et al. [2] talk about Appium testing tool. Automation of software testing is the new trend to ensure a high performance application in a short period. Appium seems to be much more promising as it delivers powerful features that can save a lot of time, labor and cost of the project. Appium tool provides a complete new revolution in automated testing, which promises efficient, bug-free and quality-rich applications. Talks about other latest automated software testing technologies in future researches should be done.

Table 1. The Limitations in the Related Work

Title of Paper	Limitations
Novel Framework for Browser Compatibility Testing of a Web Application using Selenium [1].	Automation testing only for a web application on different web browsers using Selenium WebDriver.
Software Testing Automation using Appium [2].	Only talks about Appium tool.
Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey [3].	The survey does not include the aim of automation testing (test execution, analysis, levels and technology used)
Benefits of Automated Testing Over Manual Testing [4].	The study is limited to a kind of custom made scripts. Therefore, the findings of this research might not apply for all types of testing tools.
Impediments for Automated Testing - An Empirical Analysis of a User Support Discussion Board [5].	Determine the problems that faced users and developers of automation testing software without solving it.
Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete [6].	Evaluate and compare done only between Selenium, Quick Test Professional and Testcomplete.
A Review of Scripting Techniques Used in Automated Software Testing [7].	Briefly compares between the main features of various scripting techniques.
Quality Benefits Analysis of Software Automation Test Protocol [8].	Generally, discuss of software automation testing and propose some points to improve software automated testing process.
Efficient Keywords Driven Test Automation Framework For Web Application [9].	Create and develop keywords in keyword driven test automation framework
Taxonomy of Automated Software Testing Tools [10].	Represent the taxonomy of different testing tools and performing a comparative study of these tools. But interviews, test performance of tools and experiments results not conducted.
Automated Software Testing using Test Orchestration System Based on Engineering Pipeline [11]	The Test Orchestration system uses static machine for test execution.
Comparative Study of Automated Testing Tools: Quick Test Pro and Load Runner [12].	Comparison is done only between QTP and Load Runner tools.
A Comparison of Ranorex and QTP Automated Testing Tools and their impact on Software Testing [13].	Comparison of automated testing tools is limited on Ranorex and QTP.
Studying and Comparing Automated Testing Tools; Ranorex and TestComplete [14].	Comparison is done only between Ranorex and TestComplete.
Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir [15].	Comparison is done only between Selenium, Sikuli and Watir.

The study presents the comparative analysis of Selenium, Sikuli and Watir tools using their recording capabilities, data driven testing, efficiency, language support, testing and code reusability [15]. It shows that Selenium is the best tool due to the enhanced recording features, data driven testing, and ease of learning, support for 3rd party application integration. A comparison using different automation testing tools should be done.

Motwani et al. [1] develop a prototype framework for browser compatibility testing. It deals with automation testing for a web application on different web browsers using Selenium WebDriver. This framework is not able to test the web application functionality on mobile browsers.

III. PROBLEM STATEMENT

The major factor in reducing the testing effort is Automation. There are many researches talking about automation testing for a web application. Recently, most people have started using browsers on mobiles to open web pages. Therefore, it is important to ensure that our web application will be compatible with mobile browsers and applications [1]. There is a need of a framework that can be applied to test different mobile browsers and

applications. This paper is proposing a novel framework to serve the purpose.

IV. THE PROPOSED SOLUTION

The current demand of the customer is to develop applications that can operate on both computer and mobile. Testing is a most important step, especially when an application is developed to use in critical areas of the market where a small error can lead to a huge failure.

This paper proposes a novel framework that can be applied to test different mobile browsers and applications using Appium tool. The novel framework works for all: native, hybrid and mobile-web applications for iOS and Android.

It is a data driven test automation framework that using an Appium test library and server to automatically test mobile applications. It allows to directly automating testing on mobile devices. Fig.1 shows the main components of the proposed framework and interaction between them. The main components of the proposed framework are Repository, Test Driver, Report, Exporter, Appium Library and Appium Server.

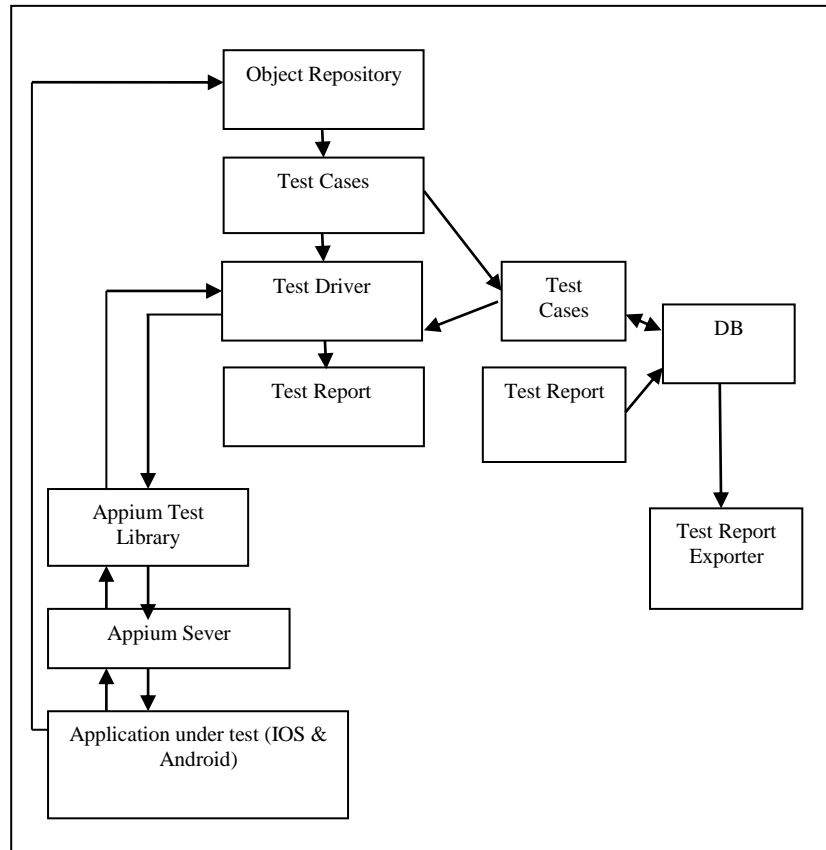


Fig.1. The Proposed Novel Framework

The Repository contains the spy which recognizes the objects available in the user interface of the application and recovers its comparing properties. After the spy does these works, the subtle elements are delivered to the repository which will view all things in the application that has been under test. The Driver runs test data that were created and stored in a database by using test script that already stored in the database. It compares between test data and test script then, executed the test and performed different actions on the user interface of applications using Appium test library.

The Report generator uses execution details and creates a report which will be stored in the database. The Exporter sends out the report of the result to an exterior file which will be displayed on the mobile. Appium Library used for user interface based automation for mobile applications. Appium Server is open source engine and interpreter that performs actions on the user interface of applications using Appium library commands.

The novel framework uses SerME database that allows performing different manipulations on data presented in the database. SerME is a most suitable database for mobiles. The framework is implemented using Java packages where every package contains classes. These classes allow communication between framework's components. For example, TestCasesRecorder contains create testcases class, TestReportGenerator contains generate reports class and there is database package that

consists of classes that responsible for store data in the database.

The novel framework uses client/server architecture. The server gets associations from test driver, listens to orders, executes those commands on a mobile device, and reacts with a HTTP response which the result of execution that commands. Test code can be written in any language that has Appium library.

Automation is performed in the context of a session. The client creates a session with a server and it sends an object which called desired capabilities'. At this time the server will begins the automation session and reply with a session id. The session id is used for sending more commands. Desired capabilities are a set of keys and values to tell the Appium server what the kind of session that wants to begin. By using the requirements different capabilities can be defined.

According to These requirements the behavior of the server can be changed. For example, in order to notify the Appium server that an IOS session used, set the 'platformName' capability to 'IOS'. The Appium server can be installed from Internet sources or directly from NPM. Appium server handles the execution of the script and linking it with the simulator/emulator. Appium.app is GUI wrappers around the Appium server exist and can be downloaded. These contain everything required to run the Appium server. Appium client libraries are available in Java, PHP, Python, Ruby, and JavaScript and C#. The server and framework libraries are what ultimately work together with the mobile application.

The Repository recovers the details of objects from the application that has been under test and shows these details to the client. The framework includes a Test Case Recorder which permits a user to make test scripts that will be used in testing an application. This data will be stored in a database as a test script. The test scripts are created and stored in the database, then calling Test Driver to run the test on mobile application by sending the result to Appium server through Appium Library. After test execution is completed, a report is generated depending on the process details received by the Test Driver.

The Reports can be sent out to an external file by using Exporter. The Exporter recovers the report that saved in the database and shows it as a data file which will be displayed on the mobile device.

There are many advantages of the novel framework. It allows testing different mobile browsers and applications using Appium tool, save time and faster to detect errors in applications. It is useful for company to speed up the process of testing and release mobile applications in short period of time.

V. VALIDATION

Validation for proposed framework was conducted using survey. The questionnaire contained fifteen questions. Data gathering was designed to achieve three goals from forty professionals.

- Goal. Improve the testing process for mobile applications (Performs test automation directly on the mobile device by using Appium tool).
- Goal 2. Save time and faster to detect errors in application.
- Goal 3. Speed up the process of testing and release any mobile application in a short period.

A. Cumulative Statistical Analysis of Goal 1

The results are shown in Table 2. Table 2 displayed that 47% and 31% of the respondents are high and very high supported by the goal 1. 2% of the respondents are agreed with very low and 5% of the sample agreed with low. 16% of the sample chose to be neutral. Fig. 2 shows the cumulative evaluation of goal 1.

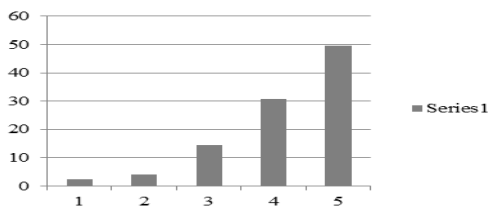


Fig.2. Cumulative Evaluation of Goal 1.

B. Cumulative Statistical Analysis of Goal 2

Cumulative static analysis of goal 2 is shown in Table 3. It is cleared from the Table 3 that 51% and 31% of the sample support that the novel framework will save time, less effort, accurate and faster to detect errors in applications. 7% of the respondents are disagreeing. 12% of the professionals are remaining neutral. Fig. 3 shows the cumulative evaluation of goal 2.

Table 2. Cumulative Statistical Analysis of goal 1

Q. No	Very low	Low	Neutral	High	Very high
1	2	3	6	10	19
2	1	2	8	11	18
3	0	1	5	14	20
4	1	2	5	11	21
5	0	2	7	15	16
Total	4	10	31	61	94
Avg.	0.8	2	6.2	12.2	18.8
Percentage	2%	5%	16%	31%	47%

Table 3. Cumulative Statistical Analysis of goal 2

Q. No	Very low	Low	Neutral	High	Very high
1	2	1	3	11	23
2	1	2	3	13	21
3	1	0	6	15	18
4	2	1	8	11	18
5	1	2	4	12	21
Total	7	6	24	62	101
Avg.	1.4	1.2	4.8	12.4	20.2
Percentage	4%	3%	12%	31%	51%

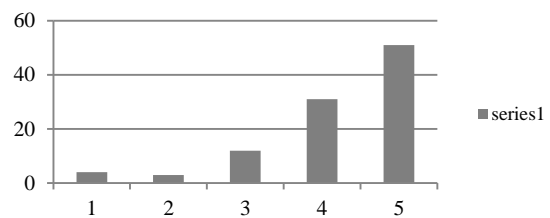


Fig.3. Cumulative Evaluation of Goal 2

C. Cumulative Statistical Analysis of Goal 3

Table 4. Cumulative Analysis of Goal 3

Q. No.	Very low	Low	Neut ral	High	Very high
1	0	2	4	10	24
2	0	2	6	13	19
3	1	0	8	9	22
4	0	2	5	14	19
5	1	2	6	14	17
Total	2	8	29	60	101
Avg.	0.4	1.6	5.8	12	20.2
Percentage	1%	4%	15%	30%	51%

Cumulative analysis of goal 3 is shown in Table 4. Table 4 shows that 51% of the respondents are very highly agreed with goal 3 and 30% of the professionals

are agreed with the goal 3. 15% of the participants are remaining neutral. Fig. 4 shows the cumulative evaluation of goal 3.

D. Final Cumulative Evaluation of all goals

The final cumulative statistical analyses of the 3 goals are shown in Table 5. It is shown that 81% (31% and 50%) of respondents agreed with the proposed framework. 14% of the respondents remained neutral. 6% of the respondents disagreed with the proposed framework. Fig. 5 shows the final cumulative analysis of three goals.

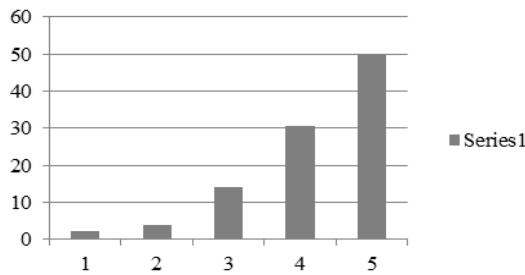


Fig.4. Cumulative Evaluation of Goal 3.

Table 5. Final Cumulative Evaluation of Three Goals

Goal No.	Very low	Low	Neutral	High	Very high
1	2	5	16	31	47
2	4	3	12	31	51
3	1	4	15	30	51
Total	7	12	43	92	149
Avg.	2.33%	4%	14.3%	30.6%	49.6%

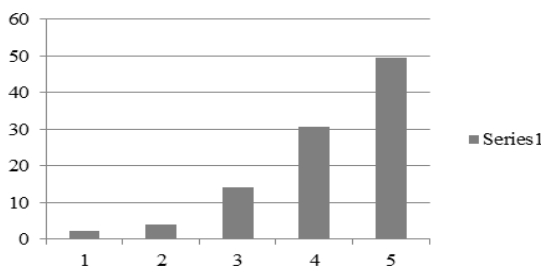


Fig.5. Cumulative Evaluation of 3 Goals.

VI. CONCLUSION

Testing and quality assurance is always important for the development of successful web/mobile application. Automation of software testing is a new trend that is taken up by the developers to deliver high performance applications in short periods of time. It is important to ensure that our web application will be compatible with mobile browsers and applications. This paper proposes a novel framework that allows automation of hybrid, native and web mobile applications for Android and iOS by using Appium tool. It provides efficient, bug-free and quality-rich applications. Validation of the novel

framework is performed using a survey. It is concluded that 81% of respondents believe that the novel framework will improve the testing process for mobile applications, save time and speed up the process of testing and release any mobile application in a short period of time.

ACKNOWLEDGEMENT

First of all, I am thankful to Allah for allowing me to complete this research. I am also thankful to my family for supporting me. Special regards and thanks to my supervisor, Dr. M. Rizwan Jameel Qureshi. He guided and helped me to complete this paper.

REFERENCES

- [1] A. Motwani, A. Agrawal, N. Singh and A. Shrivastava, "Novel Framework for Browser Compatibility Testing of a Web Application using Selenium", *International Journal of Computer Science and Information Technologies*, vol. 6, no. 6, pp. 5159-5162, 2015.
- [2] G. Shah, P. Shah and R. Muchhala, "Software Testing Automation using Appium", *International Journal of Current Engineering and Technology*, vol. 4, no. 5, pp. 3528-3531, 2014.
- [3] D. Rafi, K. Moses and K. Petersen, "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey", *IEEE*, pp. 36-42, 2012.
- [4] Asfaw, D. 2015. Benefits of Automated Testing Over Manual Testing. *International Journal of Innovative Research in Information Security*, 2(1): 5-13.
- [5] K. Wiklund, D. Sundmark, S. Eldh and K. Lundqvist, "Impediments for Automated Testing - An Empirical Analysis of a User Support Discussion Board", *IEEE International Conference on Software Testing, Verification, and Validation*, pp. 113 - 122, 2014.
- [6] H. Kaur and D. Gupta, "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete", *Int. Journal of Engineering Research and Applications*, vol. 3, no. 5, pp. 1739-1743, 2013.
- [7] M. Hanna, N. El-Haggag and M. Sami, "A Review of Scripting Techniques Used in Automated Software Testing", *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 194-202, 2014.
- [8] Rao, A. Kumar, S. Prasad and E. Rao, "Quality Benefit Analysis of Software Automation Test Protocol", *International Journal of Modern Engineering Research*, vol. 2, no. 5, pp. 3930-3933, 2012.
- [9] J. A. Jain and S. Sharma, "AN Efficient Keyword Driven Test Automation Framework For Web ", *International Journal Of Engineering Science & Advanced Technology*, vol. 2, no. 3, pp.
- [10] K. Shaukat, U. Shaukat, F. Feroz, S. Kayani and A. Akbar, "Taxonomy of Automated Software Testing Tools", *International Journal of Computer Science and Innovation*, vol. 1, pp. 7-18, 2015.
- [11] R. Singh, P. Sonavane and S. Inamdar, "Automated Software Testing using Test Orchestration System Based on Engineering Pipeline", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 5, pp. 1619-1624, 2015.
- [12] S. kumar, "Comparative Study of Automated Testing Tools: Quick Test Pro and Load Runner", *International Journal of Computer Science and Information Technologies*, vol. 3, no. 4, pp. 4562 - 4567, 2012.

- [13] A. Jain, M. Jain and S. Dhankar, "A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing", *International Journal of Engineering, Management & Sciences*, vol. 1, no. 1, pp. 8-12, 2014.
- [14] N. Dubey and S. Shiwani, "Studying and Comparing Automated Testing Tools; Ranorex and TestComplete", *International Journal Of Engineering And Computer Science*, vol. 3, no. 5, pp. 5916-5923, 2014.
- [15] I. Singh and B. Tarika, "Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir", *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1507-1518, 2014.



Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honored as the best researcher from the Department of Computer Science, COMSATS Institute of Information Technology, Pakistan in 2008.

Authors' Profiles

Ashwaq Awad Alotaibi received her BS degree in Information Technology from Faculty of Computing and Information Technology from King Abdulaziz University. She is pursuing her MS degree in Information Technology in King Abdulaziz University.

How to cite this paper: Ashwaq A. Alotaibi, Rizwan J. Qureshi, "Novel Framework for Automation Testing of Mobile Applications using Appium", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.9, No.2, pp.34-40, 2017.DOI: 10.5815/ijmecs.2017.02.04