International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.4 (July-2020)

http://dx.doi.org/10.12785/ijcds/090413

Testing Approaches for Web and Mobile Applications: An Overview

Zahra AbdulKarim Hamza[1] and Mustafa Hammad[1]

[1] *Department of Computer Science, University of Bahrain, Sakhir, Kingdom of Bahrain*

Received 30 Sep. 2019, Revised 15 Jun. 2020, Accepted 24 Jun. 2020, Published 1 Jul. 2020

Abstract: Software testing is one of the main phases in the software development lifecycle. Each software is being tested to ensure its conformance with the software requirements. Web and mobile applications are considered among the software that should be tested carefully. Such applications are heavily used by different people for different purposes. There are many research endeavors in the field of software testing. Many approaches are proposed for the three types of testing, black, grey, and white box. It is important to survey the literature of the testing approach to help the software engineers/ developers/ testers in choosing the right testing approach and methodology based on the software scenarios and needs. In this paper, we surveyed some of the black, grey, and white box testing approaches along with some tools. This survey helps the software engineers to choose the appropriate approaches for the different web and mobile applications.

Keywords: Software Testing, Black-Box, White-Box, Grey Box, Web, Mobile

1. INTRODUCTION

Web and mobile applications are used worldwide for different purposes. Such applications became a need to ease the life of people in several aspects. Web/mobile applications substitute many activities that were done by humans, such as activities related to shopping, healthcare, controlling smart houses/ cities/ farms, and others.

As for all types of software, testing is considered as a main phase of the development process. In such a phase, the faults can be detected and the software is verified and validated. Moreover, software testing is used to make sure of the software quality in general. Testing importance led IT testing specialists to work on finding methodologies and approaches that assist in the testing process. Many research problems have been studied to make the testing process more efficient and effective.

Black and white box approaches are two of the software testing types. Both of them are used in the different software testing. Black box testing does not consider the internal structure and implementation of the block that is under the testing process, but it does focus on the input and output. The black-box approach is more about observing the software behavior according to a certain input. However, the white box approach focus is on the internal design and implementation of the block that is under testing. The white box approach is about tracing and finding out the journey of certain input in the software that is going through different and multiple paths.

In this paper, the literature of testing mobile and web applications, and the white and black box approaches, is reviewed. Furthermore, a comparison of the current approaches is done using specific test-key factors, which are important and considered in the testing process.

Such a survey is important for software testers from many points of view. This survey helps to choose one or more testing approach to test the web/mobile applications. Moreover, the survey will assist the software engineers in preparing the requirements and take the needed pre-processing steps.

2. BACKGROUND

This section covers overview information about the software testing and black/ white box testing processes. Besides, the web and mobile application structures are comparatively discussed.

A. Web and mobile applications

Web applications are used to perform tasks over the internet. A browser is needed as an environment to use the web applications. There are also a set of protocols that are used to send and receive requests from/to the web servers to perform a certain task.

Email: 20112417@stu.uob.edu.bh, mhammad@uob.edu.bh

For mobile applications, the features are the same as the web applications, except certain differences. Some of the mobile applications are web applications, but they are developed as mobile-friendly applications. Moreover, mobile applications can run offline without an internet connection. There are three types of mobile applications, native application, hybrid application and web applications [1]. The hybrid and web applications are sharing the same components of the real web applications, but with additional features that give them the ability to work in the mobile environment.

B. *Software testing*

Software testing is one of the main phases in all Software Process Models (SPM). Software is created to fulfill certain requirements and needs. So, software testing comes to ensure that all requirements and specifications have been considered while developing the software. Furthermore, testing is used to detect any faults that make the software in the error state, which can lead to failures.

Figure 1 shows the main and common components of the web/mobile applications. Those components contain multiple sets of blocks, which represent the units of the application, and should be covered in the testing process with consideration of the specified layers and interfaces.

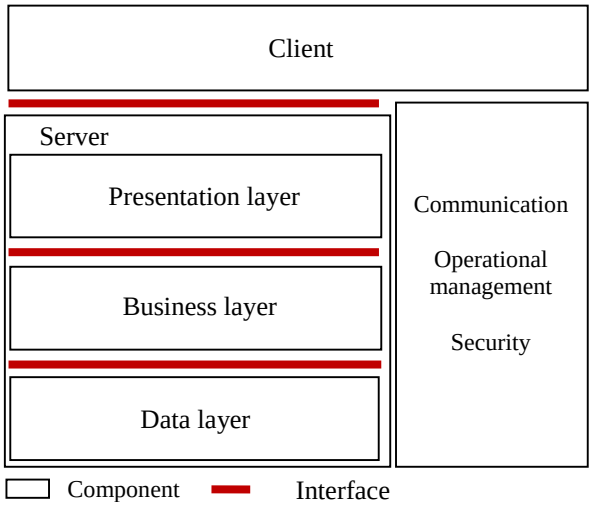

Figure 1. Web/ Mobile applications' layers

In the coming sub-section, black, grey, and white testing approaches are discussed. The properties of each testing approach are also mentioned to differentiate between the three types.

C. *Black, white and grey box testing*

Black-box is one of the software testing types. It is used to make sure of the behavioral issues of the software. Such a testing type is called also interface-level testing because the testers are using the user interfaces usually to perform the testing task. Moreover, it is called functional testing due to the tester type, which could be the users of the software, who do not know the software, but need to test its functionality.

In black-box testing, testers' concerns are only about the software specifications. Although the software is under testing, the internal artifacts of it are excluded in the black box testing. The tester has zero knowledge about the software code or the internal structure of it. Such testing is used to examine how the software is reacting regarding certain input, which is prepared intentionally to see the possible outputs (results) of the software for each prepared input.

There are many benefits to the black box testing type. One of the most important ones is to test the software as an attacker. The attacker does not know the software, which makes him/ her tries the possible inputs until certain outputs are found. The other benefit of the black box testing is represented in the least time-consuming testing type, where the testers should only concern about the input and output. Such a testing approach is used for acceptance testing and entire system testing.

White-box testing is another type of software testing. In such a testing process, software input, software artifacts, and software output are included in the testing. White box testing process is also called code level testing. Each statement of the code in the software module, which is under testing, should be executed.

White box testing has many properties. Such a testing type is about addressing all the internal problems of the software. Since the testers are looking into the software code, the programming and implementation knowledge is required to perform the testing tasks. Such a testing approach is used for unit testing and integration testing.

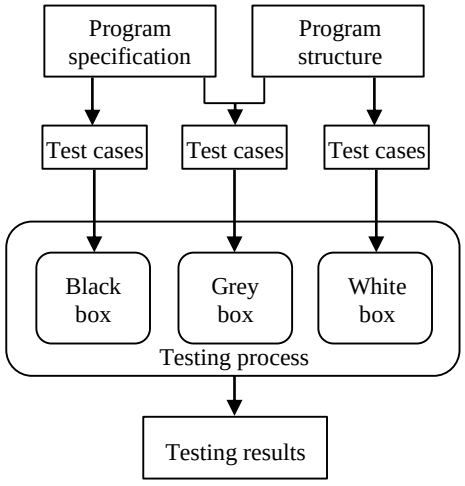

Figure 2. Testing processes

The other type of testing approaches is the grey box testing. In grey box testing, the testers' concerns are with both the specifications and the internal structure of the

program. Grey box testing is done by the developers to debug the software.

The test cases are generated in a different way for each of the testing types. As shown in Figure 2, the program specifications are used to generate the test cases for the black box testing approach. However, for the white box testing, the focus is on the internal program structure to generate the test cases. For the grey box testing, both of the program specifications and program structure is used to generate the test cases.

D. Differences between web testing and mobile testing

Web applications' testing, nowadays, is more complex than before, due to the new technologies, business growth, and users' requirements [2]. Many approaches are proposed to define testing models and methods for web application testing, such as the work in [3, 4].

Mobile applications are applications that are developed for mobile devices. "Mobile devices" as a term is usually used for the smartphones as they are the well-known mobile devices. There are many endeavors in the field of mobile application testing. Testing frameworks and approaches have been proposed to ease the testing process of the mobile applications, such as the proposed approach in [5] and the testing framework proposed in [6].

There is a set of differences between web application testing and mobile application testing. Those differences are not in the structure itself, but in the content of structures' components.

Graphical User Interface is one of the major differences that can affect the testing process and makes it different between web and mobile. In the web applications, the area to display the content for the user to interact with is wider than the mobile applications, due to the different screens' sizes [7]. Also, the view of the interface should be tested in "Portrait" and "Landscape" modes in the mobile applications. However, in web applications, there is only one orientation.

The other point that is considered as a difference in web and mobile application testing is the power management [8]. The power in the case of web applications in not being cut-off frequently. Although the web applications can be used in the browsers of mobile devices, it is usually accessed using desktop computers or laptops. However, in the case of mobile applications, the power should be managed through the software, because it is depending on the battery of the device.

Furthermore, the lifecycle of the application is also should be considered in the testing. In the mobile applications, the lifecycle care is a responsibility of the application, because of the limited resources [7]. However, in web applications, the operating system is taking care of the "applications lifecycle".

3. REQUIREMENTS OF SOFTWARE TESTING

Testing tasks require certain preparation to get started. Each of the testing types (black box, white box, and grey box in our case) has a set of requirements and some of them are common.

One of the requirements for software testing is test sequences. Test sequences can be driven from the software requirement after the analysis phase. In the software requirements analysis phase, the software specifications are generated to be used for implementing the software by the developers. The software specifications, which can be represented in UML diagrams, are the most important part that the test sequences can be generated from. The work in [9, 10, 11, 12, and 13] proposed approaches to generate test sequences from the UML models. Such test sequences are used also in taking the decisions in generating test cases as input for the software in the testing process.

There are many ways of generating test cases, which is one of the testing preparation steps. Using UML is considered as an effective way of generating test cases. Many approaches have been proposed, such as in [14 and 15], are using UML activity diagrams to generate the test cases. Moreover, other approaches are focusing on sequence diagrams in the process of generating test cases. For example, the approaches proposed in [16 and 17].

Using test sequences and test cases in black and white box testing can be seen from different aspects. In white-box testing, test sequences generation assists the developers in listing the artifacts that should be tested. Moreover, in the test sequences, the order of the software blocks is considered to be tested sequentially. However, the test cases are used to prepare the sets of data as software testing input, and also to state the expected output regarding each input.

As mentioned previously in this paper, black box, white box, and grey-box testing are testing approaches that are used to test web and mobile applications. Such applications are consisting of several parts, which require specific preparation of test sequences and test cases. Since there are common layers between the web architecture and mobile architecture, there might be some similarities in the ways of preparing the requirements of software testing. However, the type of the data in the test cases and test sequences are different, because of the environmental differences between the web and mobile applications.

Testing approaches have been improved in terms of different aspects, such as automation, security, intelligence of using them… etc. Part of the literature of the use of black and white box testing for web and mobile applications is surveyed to highlight the most important points to the software engineers, software developers, and software testers.

4. CURRENT APPROACHES

There are many proposed approaches for mobile and web application testing. The approaches differ from one to another according to the used methodology.

We are proposing four test-key factors to survey the current approaches. Such factors are important to software testers and developers to choose the right testing approach/tool. The test-key factors are shown in Table 1.

TABLE I. PROPOSED TEST-KEY FACTORS

F#	Factor	Description
F1	Machine learning	Using machine learning as an artificial intelligence technique to develop test plans, test sequences and test cases
F2	Security	Applying security checking procedure and rules
F3	Automation	Automating the testing process by preforming sequentially the procedures starting from the test cases' generation
F4	Heuristic search	Applying some heuristic concepts in the testing processes to find an acceptable solution for certain cases

There are many tools used for automating the process of web applications' testing. As an example, the work in [18] presented three algorithms and one tool to automate the testing of web applications. The tool and the related algorithms are built using a method of Search-Based Software Testing (SBST). The results are represented in reducing the test efforts by 30%.

As the web applications consist of different units, there are some built tools for automating only the SQL injection testing. Sania [19] is a new technique, which is developed to discover the vulnerabilities caused by SQL injections in the web applications. This technique is used in the development phase to debug the software before reaching the delivery phase. Sania has been evaluated using real web applications. The evaluation results show that it is efficient when compared with well-known vulnerabilities scanners. Gregory et al. [27] proposed another automatic prevention technique for SQL injection attacks based on the parse tree method, which is the same as the proposed approach in [20]. The technique has been implemented and experimentally evaluated using a case study. The results were promising as it shows the effectiveness of this technique in preventing such SQL injection attacks. Moreover, the work in [21] suggested an evaluation technique to select a suitable tool as a vulnerabilities scanner for SQL injection in web applications. This tool provides help to the testers to choose the right tool to test the SQL injection in web applications automatically.

Testing using session-data is an important type of testing for web applications. The work in [22] and [23] studied the possibility of using the session-data as part of the test cases in the web application testing. The results of the studies show that the session-data could improve the testing process and assist in finding bugs in the web applications. Furthermore, an approach has been proposed in [24] includes the session-data in the automated web application testing. The evaluation results show that the approach is working efficiently in the web application testing with the session data by finding more issues to solve.

Furthermore, there are some proposed approaches in the field of formal verification. The approaches of [25] and [26] are proposed based on UPPAAL for formal verification. The work in [25] provided an approach to verify the wireless sensors and used UPPAAL to verify the functional and non-functional properties. However, the work in [26] is mainly for cloud-based applications, which are web applications.

A. Black box testing approaches

Black-box testing has been frequently used as part of the mobile application testing process. Mobile-Test [27] is a tool, which is built to automate the black-box testing for mobile applications. The proposed tool is also supporting the test case generation, which is used in the testing process. The results show that the tool reduces the efforts and eases the process of testing by automating it. Another tool called B-Box-Tester [28] is created to support the black-box testing of Android mobile applications. This tool provides detailed reports about the status of the software in terms of security. The efficiency and the effectiveness of the B-Box-Tester has been evaluated through mobile applications. The assessment results show that B-Box-Tester can be used as the main tool in the testing process due to its high accuracy. Another work [29] presented the state-of-the-art of the black-box testing automation for web applications. The studying results were promising regarding the effectiveness of the black-box testing using the automation tools.

Automating black-box testing process became necessary due to the required efforts. Auto-black-test [30] is a tool that has been implemented to automate the black-box testing. The tool automates first the test case generation and then proceed with the testing process. Meinke [31] suggested an approach based on polynomial functions. The approach has been evaluated using a case study to show how automated black-box testing can be represented mathematically/ logically.

Black-box testing can be used also along with the genetic algorithms. For example, the work in [32] proposed a genetic algorithm to generate the test cases for black-box testing. The algorithm is based on artificial intelligence methods and techniques. The algorithm is tested using several case studies by generating test cases.

B. White box testing approaches

White-box testing is used also in the mobile application testing process. The work in [33] proposed an algorithm to use the white-box testing in Android applications. The algorithm has been implemented and tested using several applications. The implemented tool discovered more than 30 new bugs in the applications in the experimental evaluation. Another approach [34] is suggested to automate security testing using a white-box methodology for mobile applications. Generating the test cases automatically is the starting point for the approach of automating the white-box test.

Modeling approaches have been frequently used in improving the web application testing. The work in [36] and [35] proposed models that help in web application testing. The models are based on the object-oriented methodology. Each of the models has been evaluated using real-world web applications. The results show that the models improve the testing process quality, especially for the white-box testing.

C. Grey box testing approaches

Many testing approaches have been proposed based on the grey box testing. Chen et.al. [37] suggested an approach that is based on grey box testing. The approach is proposed to perform penetration testing for Internet of Things (IoT) applications, which is web or mobile. Furthermore, the work in [38 and 39] described how the grey box is used for the software testing, especially the software correctness testing. However, the work in [39] focused mainly on web services. Moreover, for the web service, the proposed work in [40 and 41] is a service-oriented architecture testing approach that is based on a grey box concept. Xu et. al. [42] discussed some methods for web application testing using a grey box, due to its effectiveness in finding the errors. Another work in the field of web service testing is [43], which proposed a testing approach based on the grey box. In this work, the event exposure for web services is the main focus of the proposed approach.

5. TESTING APPROACHES AND TOOLS EVALUATION

Table 2 shows some of the current approaches with the corresponding used method, which is represented as test-key factors. Where F1 is Artificial Intelligence, F2 is Security focused, F3 is Fully automated and F4 is Heuristic search.

The black box approaches, which are shown in Table 2, differ from each other due to the used methodology. The Mobile-test tool [27] is provided fully automating the testing process. Such an approach is suitable for mobile applications that are not complex and not heavy in terms of code. As the automation feature helps in completing the testing phase faster than the manual testing, it gives more accurate results in the simple applications. To use such automation tools in the complex applications, testers along with the developers may divide the application into smaller parts. Auto-black-test [30] is another tool that automates the testing process. This tool is used to test the interactive applications, despite their type. Auto-black-test uses the applications' GUI to generate scenarios and perform the testing. For Android applications, B-Box-tester [28] is built to accomplish the testing phase in the development process. B-Box-tester considers the security property while testing the Android applications. Security is an important quality attribute that users, in general, worry about. So, testers can use B-Box-tester to make sure of the application's security. Last et al. [32] genetic algorithm is suitable to be used with the various types of applications. Using intelligence, the algorithm is developed to generate test cases with consideration of having a small number that covers all the paths and scenarios of the application.

TABLE II. SOME OF THE CURRENT BLACK/WHITE TESTING APPROACHES WITH THE TEST-KEY FACTORS

Testing Approach	Black box	White box	Grey box	F1	F2	F3	F4
Bo. et al. [27]	✓					✓	
Zhauniarovich et al. [28]	✓				✓		
Mariani et al. [30]	✓					✓	
Last et al. [32]	✓			✓			✓
Godefroid et al. [33]		✓				✓	
Mahmood et al. [34]		✓			✓		
Chen et.al. [37]			✓		✓		

The white-box based approaches that shown in the table are suitable for tracing certain input in the application. The proposed algorithm in [33] has been built the base of the white-box approach. The algorithm is designed to automate the testing of Android applications. The automation feature makes this approach suitable to be used by software testers in testing the mobile application. Mahmood et al. [34] approach focused on the security part more than other quality attributes, which makes it the choice for testers that are willing to test the security using the white-box method.

For the grey box testing, there is only one reviewed approach that can be differentiated using the test-key factors. Chen et al. [37] proposed an approach for penetration testing. The penetration testing is mainly performed to identify any vulnerability or threats in the software.

TABLE III. AVAILABLE TESTING TOOLS

Testing tool	Web	Mobile	Description
Selenium [44]	✓	✓	Open-source tool with easy interface and can be used in several operating systems
Watir [45]	✓		Based on Ruby and provide the feature of automating the test cases. It is used mainly for web services.
JMeter [46]	✓		Open source and it can test the performance through some protocols. It is used for load testing
SoapUI [47]	✓		An open-source tool for web services testing and mainly for the service-oriented architecture
Fortify [48]	✓		Testing tool focused on security. It scans the web application code to determine any vulnerability.
Appium [49]		✓	Open-source testing tool for Android and IOS applications. It is for the different types of mobile applications (Native, Hybrid, and mobile web). It can automate the tests.
Robotium [50]		✓	Open-source testing tool for Android, which is written in JAVA. It allows the users to develop black-box test cases to be automated tests.
MonkeyRunner [51]		✓	Testing tool written in Python to test Android applications only.

Table 3 shows the top testing tools for web and mobile testing. Most of the testing tools for web and mobile are open source. Some of the testing tools, as shown in Table 3 are having the automation feature, which eases the testing process for the testers. Some of the available testing tools are developed for specific testing, such as Fortify [47] that is built for security testing. Each of the testing tools has a feature that differentiates it from the others. For example, JMeter [45] is used to test the performance and some protocols. Moreover, Watir [44] have an automation feature that allows the test cases to be used automatically in the test process.

Such evaluation can assist the software engineers to complete automating the software development life-cycle processes. When the testing tools are categorized and analyzed, the testing tools' selection can be done automatically. The work in [52] proposed an approach to automate the UML use case generation from the requirements, and the work in [9] generates the test sequences from the UML use case. However, the work in [53] surveyed some of the testing tools. This sequential generation process can lead to the automation of the testing tools' selection as well.

6. CONCLUSION

In this paper, a set of current testing approaches and available tools, which are based on the black, white, and grey box, have been surveyed using four test-key factors. Such a survey assists the software engineers/ developers/ testers in deciding on the needed testing approach/tool. It also highlights the possible ways of preparing test sequences and test cases to be used for the black box white box and grey box testing.

The points that limit the survey are related to the test-key factors. There are other methods, which should be considered as test-key factors, such as object-oriented models. Moreover, more approaches are to be surveyed along with the ones that are in this research paper to study the majority of the current approaches.

REFERENCES

[1] Khoshgoftaar and Taghi, Software engineering with computational intelligence, Boston: Springer Science & Business Media, 2012

[2] Ricca and Tonella, "Analysis and testing of web applications," in *International conference on software engineering. ICSE 2001*, 2001.

[3] Liu, Kung, Hsia and Hsu, "Structural testing of web applications," in *In Proceedings 11th International Symposium on Software Reliability Engineering. ISSRE 2000 - IEEE*, 2000.

[4] Zhongsheng, "Test case generation and optimization for user session-based web application testing," *Journal of Computers*, vol. 5, no. 11, pp. 1655-1662, 2010.

[5] Baride and Dutta, "A cloud based software testing paradigm for mobile applications," *ACM SIGSOFT Software Engineering Notes*, vol. 36, no. 3, pp. 1-4, 2011.

[6] Ping, Sharbini, Lin and Julaihi, "Designing a mobile application testing model," in *In The International Conference on Computing, Networking and Digital Technologies (ICCNDT2012)*, 2012.

[7] Ahmed and Ibrahim, "A comparative study of web application testing and mobile application testing," in *In Advanced Computer and Communication Engineering Technology - Springer*, 2015.

[8] Zhang, Tiwana, Qian, Wang, Dick, Mao and Yang, "Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., MAccurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/ software codesign and system synthesis*, 2010.

[9] Abdulkarim Hamza Z. and Hammad M., "Generating test sequences from the UML use-case diagram". In *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1-6. IEEE, 2019.

[10] L. C. Briand and Y. Labiche. A uml-based approach to system testing. In Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools, UML'01, pages 194–208, London, UK, Springer-Verlag, 2001

[11] P. Fr¨ohlich and J. Link. Automated test case generation from dynamic models. In Proceedings of the 14th European Conference on Object-Oriented Programming, ECOOP '00, pages 472–492, London, UK, Springer-Verlag, 2000

[12] J. Ryser and M. Glinz. A scenario-based approach to validating and testing software systems using statecharts. In In 12th International Conference on Software and Systems Engineering and their Applications (ICSSEA'99), page 7, 1999

[13] J. Hartmann, M. Vieira, H. Foster, and A. Ruder. A uml-based approach to system testing. Innovations in Systems and Software Engineering, 1(1):12–24, 2005

[14] Hettab, Chaoui and Aldahoud, "Automatic test cases generation from UML activity diagrams using graph transformation," in *6th ICIT*, 2013

[15] Kim, Hyungchoul, S. Kang, J. Baik and Inyoung-Ko, "Test Cases Generation from UML Activity Diagrams," in *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007) - IEEE*, 2007

[16] Dhineshkumar, "An approach to generate test cases from sequence diagram," in *International Conference on Intelligent Computing Applications - IEEE*, 2014

[17] Nayak and Samanta, "Automatic test data synthesis using uml sequence diagrams," *journal of Object Technology*, vol. 9, no. 2, pp. 75-104, 2010

[18] Alshahwan and Harman, "Automated Web Application Testing Using Search Based Software Engineering," in *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, 2011.

[19] osuga, Kono, Hanaoka, Hishiyama and Takahama, "Sania: Syntactic and semantic analysis for automated testing against sql injection," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)-IEEE*, 2007

[20] Buehrer, Weide and Sivilotti, "Using Parse Tree Validation to Prevent SQL Injection Attacks," in *Proceedings of the 5th international workshop on Software engineering and middleware-ACM*, 2005

[21] Fonseca, Vieira and Madeira, "Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks," in *13th Pacific Rim international symposium on dependable computing (PRDC 2007) - IEEE*, 2007

[22] Elbaum, Sebastian, Karre and Rothermel., "Improving web application testing with user session data," in *Proceedings of the 25th International Conference on Software Engineering 2003 - IEEE*, 2003.

[23] Elbaum, Rothermel, Karre and Fisher, "Leveraging User-Session Data to Support Web Application Testing," in *IEEE Transactions on Software Engineering*, 2005

[24] Harman and Alshahwan, "Automated Session Data Repair for Web Application Regression Testing," in *2008 1st International Conference on Software Testing, Verification, and Validation - IEEE*, 2008

[25] Chaudhry, Y. A. K., & Hammed, M, "Formal Verification of Cloud based Distributed System using UPPAAL," *in 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT*, 2019.

[26] Hammad, M., & Cook, J, "Compositional verification of sensor software using UPPAAL," In 2012 IEEE 23rd International Symposium on Software Reliability Engineering, 2012.

[27] Bo, Xiang and Xiaopeng, "MobileTest: A tool supporting automatic black box test for software on smart mobile devices," in *Proceedings of the Second International Workshop on Automation of Software Test 2007 - IEEE*, 2007

[28] Zhauniarovich, Philippov, Gadyatskaya, Crispo and Massacci, "Towards Black Box Testing of Android Apps," in *2015 10th International Conference on Availability, Reliability and Security* 2015 - IEEE, 2015

[29] B. Jason, E. Bursztein, D. Gupta and J. Mitchell, "State of the art: Automated black-box web application vulnerability testing," in *2010 IEEE Symposium on Security and Privacy*, IEEE, 2010

[30] Mariani, Pezzè, Riganelli and Santoro, "AutoBlackTest: A Tool for Automatic Black-Box Testing," in *33rd International Conference on Software Engineering (ICSE) 2011 - IEEE*, 2011

[31] Meinke, "Automated Black-Box Testing of Functional Correctness using Function Approximation," in *ACM SIGSOFT Software Engineering Notes. - ACM*, 2004

[32] Last, Eyal and Kandel, "Effective black-box testing with genetic algorithms," in *Haifa Verification Conference*, 2005

[33] Godefroid, Levin and Molnar, "Automated Whitebox Fuzz Testing," *NDSS 2008*, vol. 8, pp. 151-166, 2008

[34] Mahmood, Esfahani, Kacem, Mirzaei, Malek and Stavrou, "A whitebox approach for automated security testing of Android applications on the cloud," in *Proceedings of the 7th International Workshop on Automation of Software Test 2012 - IEEE*, 2012

[35] Ricca and Tonella, "Analysis and Testing of Web Applications," in *Proceedings of the 23rd international conference on Software engineering 2001* - IEEE, 2001

[36] Kung, Liu and Hsia, "An Object-Oriented Web Test Model for Testing Web Applications," in Proceedings First Asia-Pacific Conference on Quality Software 2000 - IEEE, 2000

[37] Chen, C. K., Zhang, Z. K., Lee, S. H., & Shieh, S., "Penetration testing in the iot age," *In computer*, Vol. 51, IEEE, 2018

[38] Khan, Mohd Ehmer. "Different forms of software testing techniques for finding errors." International Journal of Computer Science Issues (IJCSI), vol. 7. 2010

[39] Nikfard, P., Zadeh, M.H.A. and Ibrahim, S.B. "A comparative evaluation of approaches for web application testing," *In International Conference on Soft Computing and Software Engineering* 2013 (SCSE'13), 2013

[40] Jehan, Seema, Ingo Pill, and Franz Wotawa. "SOA Grey Box Testing--A Constraint-Based Approach." In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*, pp. 232-237. IEEE, 2013

[41] Endo, André Takeshi, Michael Linschulte, Adenilso da Silva Simão, and Simone do Rocio Senger de Souza. "Event-and coverage-based testing of web services." In *2010 Fourth International Conference on Secure Software Integration and Reliability Improvement Companion*, pp. 62-69. IEEE, 2010

[42] Xu, Lei, Baowen Xu, and Jixiang Jiang. "Testing web applications focusing on their specialties." *ACM SIGSOFT Software Engineering Notes* 2005

[43] Ye, Chunyang, and Hans-Arno Jacobsen. "Event exposure for web services: a grey-box approach to compose and evolve web services." In *The smart internet*, pp. 197-215. Springer, Berlin, Heidelberg, 2010.

[44] Hafsah Mahmood and Mehreen Sirshar. "A Case Study of Web Based Application by Analyzing Performance of a Testing tool". *In: IJ Education and Management Engineering*. 2017

[45] WATIR. R etrieved on J uly 5, 2019 fro m http://watir.com/

[46] The Apache Software Foundation. Apache JMeter. R etrieved on J uly 5, 2019 fro m https://jmeter.apache.org/.

[47] LoadUI. Last access: July 5, 2019. https://smartbear.com/product/ready-api/loadui/overview/.

[48] Fortify. Last access: July 5, 2019. https://www.microfocus.com/enus/products/static-code-analysis-sast/overview

[49] Appium. Last access: Oct 16, 2019. http://appium.io/

[50] Robotium. Last access: Oct 16, 2019. https://github.com/robotiumtech/robotium

[51] MonkeyRunner. Last access: Oct 16, 2019. https://developer.android.com/studio/test/monkeyrunner/index.html

[52] Hamza, Z. A., and Hammad, M. "Generating UML Use Case Models from Software Requirements Using Natural Language Processing." In *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, pp. 1-6. IEEE, 2019.

[53] Hamza, Z. A., & Hammad, M. Web and Mobile Applications' Testing using Black and White Box approaches - IET. 2019.

Zahra Abdulkarim Hamza is currently studying M.Sc. Software engineering at University of Bahrain. She is Graduated from University of Bahrain, B.Sc. Computer Science and got the best senior projects award - 3rd place, 2017, for project titled: "Automatic Diacritization for Arabic Text using Voice Recognition Technique". She Worked at Batelco for two years in Business Intelligence, Corp. Apps and Enterprise Data Warehouse.

Mustafa Hammad is an Associate Professor in the Department of Computer Science at the University of Bahrain and Mutah University. He received his Ph.D. in Computer Science from New Mexico State University, USA in 2010. He received his M.Sc. degree in Computer Science from Al-Balqa Applied University, Jordan in 2005 and his B.Sc. in Computer Science from The Hashemite University, Jordan in 2002. His research interests include machine learning, software engineering with focus on software analysis and evolution.