



COMPARATIVE ANALYSIS FOR SOFTWARE TESTING: MOBILE APPLICATIONS VERSUS WEB APPLICATIONS

Zainab Hassan Muhamad and Rosziati Ibrahim

Department of Software Engineering, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia

E-Mail: zainab_hssn@yahoo.com

ABSTRACT

Software testing has an important role in software engineering, and is fundamental to Software Quality Assurance (SQA). Besides the popularity of web applications, mobile applications have gained paralleled advancement despite increasing complexity. On one hand, this issue reflects the rising concerns for ensuring performance both of web and mobile applications. On the other hand, a comparative analysis of software testing issues between web and mobile applications has not been completed. Thus, this study aims to employ an effective testing approach that is able to adapt both of web and mobile application testing to detect possible failures. To achieve this, UML activity diagrams were developed from four case studies for web and mobile applications to describe the behaviour of those applications. Test cases were then generated by using the MBT technique from the developed UML activity diagrams. Performance measurements Hits per Second, Throughput and Memory Utilization for each case study were evaluated by execution of test cases that were generated by using HP LoadRunner 12.02 tool. Finally, the Mean Square Error (MSE) of performance measurements was compared and analysed among the four case studies. The experimental results showed that the disparity between the mobile applications and web applications was obvious. Based on the comparison analysis for software testing of mobile applications versus web applications that was the web applications were lesser than mobile applications for software testing of four case studies in terms each of the Hits per Second, Throughput and Memory Utilization. Consequently, mobile applications need more attention in the testing process.

Keywords: software testing, mobile application testing, web application testing, model-based testing, unified modeling language.

INTRODUCTION

A Mobile application, also known as mobile apps, is a software application that can be installed on handheld devices, such as mobile phone, tablet, e-reader, or other portable device. It is supported by operating systems and is able to connect to wireless networks (Gahran, 2011). While, a web application is an application that is invoked by a client web browser over the Internet or an Intranet. A web application allows the information processing functions to be initiated remotely from a client and executed partly on a web server, application server, and/or database server. These applications are specifically designed to be executed in a web-based environment. Web applications are playing a very important role in many business domains like retail, finance, sales, marketing and management (Imran and Roopa, 2012).

Software testing has an important role in software engineering, and is fundamental to Software Quality Assurance (SQA). The objective of software testing is to show the differences between the expected and actual behaviors of the System under Test (SUT). Testing is essential to the Software Development Life Cycle (SDLC) that impacts the popularity of software and hardware (Ang *et al.*, 2014). The goal of software testing is to detect whether the behavior of the system implemented has visible differences from the expected behavior stated in the specification (Sumit and Narendra, 2014). As in software engineering, performance testing is performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure and validate quality attributes of the system, such as scalability, reliability and

resource usage (Shilpa and Meenakshi, 2014). It is concerned with achieving hits per second, throughput, and resource utilization levels that meet the performance objectives for the system. There are many tools that can be used to simulate the load in terms of users, connections and capture data related to hits per second, throughput, and resource utilization. Among the most important tools is HP LoadRunner (Sheetal and Joshi, 2012). There are various parameters based on which performance of the system is measured. They are known as performance measurements. One is resource utilization, which is the number of resources used to serve the user request. These resources can be memory, processors, disk I/O and network utilizations (Kalpan and Ramakanth, 2012). This study is focused solely on resource utilization of memory. Throughput is an important indicator for measuring server performance, which represents the throughput capacity of the server at any time. Generally, the higher the throughput, the better the server performance will be. Hits per second are the number of HTTP requests per second that virtual users submit to the Web servers. It can reflect whether or not the system is stable, when the number of user's increases, the hits per second will increase accordingly (Weibiao *et al.*, 2014).

The testing process is a very costly and time consuming. In order to cut down on costs, save time, and increase reliability, Model-Based Testing (MBT) approach was used in this study. MBT is a process of generating test cases and evaluating test results based on the design and analysis of models. Recently, MBT has gained attention with the popularization of modeling in software development. The Unified Modeling Language (UML)



modeling based testing approach intends to solve this problem (Sandeep, Sangeeta and Gupta, 2012; Ang *et al.*, 2014). MBT is one of the most of significant techniques that has been applied to generate test cases by using UML diagrams for web applications by (Zhang, Rong and Zhang, 2007; Ke, Xiao-Hong and Zhi-Yong, 2010; Prachet and Abhishek, 2013). Moreover, MBT has been applied to mobile applications by (Chouhan, Shrivastava and Parminder, 2012; Tobias and Volker, 2014; Ang *et al.*, 2014). The UML activity diagrams support GUI modeling, automated test case generation and error diagnosis. This approach can reduce the overall test time, and can effectively detect fatal faults in mobile applications (Ang *et al.*, 2014). In addition, the UML activity diagram is one of the most important diagrams among the thirteen diagrams. It is characterized by the high level of abstraction compared to other diagrams like sequence diagrams, class diagrams, etc. Furthermore, it is able to represent loops and concurrent activities. UML activity diagrams capture the key system behavior. The main advantage of this model is its simplicity and ease of understanding the flow of logic of the system as well. For all these reasons, activity diagrams are well suited for treating system level testing of web applications (Aye and Myat, 2014).

Currently, mobile applications have parallel advancement with web applications. This issue reflects the rising concerns for ensuring performance both of web and mobile applications (Maryam and Rosziati, 2014). As the growth both of web and mobile applications is rapid, this issue was interesting to some researchers (Vikas and Rajesh, 2014; Prachet and Abhishek, 2013), and they have taken into consideration web application testing. On the other hand, other researchers (Tobias and Volker, 2014; Ang *et al.*, 2014) were focused on mobile application testing. But, the comparative analysis between web application testing and mobile application testing is an issue that has not yet been resolved. Thus, the motivation of this study is to employ an effective testing approach, which is able to adapt with both web and mobile application testing to discover failures in the required performance. Therefore, the UML activity diagrams developed from four case studies for web and mobile applications to describe the behaviour of those applications. Test cases then generated using the MBT technique based on Test Case Generation based on Activity Diagram (TCBAD) model from the developed UML activity diagrams. In addition, performance measurements Hits per Second, Throughput and Memory Utilization for each case study were evaluated by execution of test cases that were generated by using HP LoadRunner tool. Finally, the performance measurements Hits per Second, Throughput and Memory Utilization compared and analysed among the case studies.

RELATED WORKS

Due to the lack of research on comparisons between web applications testing and mobile application testing, some of the research related to automated testing and techniques will be reviewed in this section to generate

test cases for web and mobile applications. In addition to the background of the work requirements for this study, other related works that consist of similar efforts to demonstrate the state-of-the-art in the test case generation will also be presented. Most researchers were used the MBT technique based on the UML activity diagrams, such as Tobias and Volker (2014), Ang *et al.* (2014), Chouhan *et al.* (2012) and Pakinam *et al.* (2011). While, the other researchers, such as Vikas and Rajesh (2014) were used sequence diagram and web diagram, and Prachet and Abhishek (2013) were used use case diagram and activity diagram.

Tobias and Volker (2014) proposed an approach to test case generation and automated execution uses MBT supported by UML activity diagrams to improve the testing of context-aware mobile applications by reducing test cases from design-time system models. Likewise, Ang *et al.* (2014) proposed the AD Automation framework to enable automated Graphical User Interface (GUI) testing of smartphone applications based on UML activity diagrams, which supports user behavior modeling and automated GUI test case generation. This approach can reduce the overall test time, testing efforts as well as improving test adequacy. Moreover, this approach effectively detects fatal faults in complex GUI implementations and improves the quality of designs. Also, Pakinam *et al.* (2011) proposed an automated approach for generating test cases from UML Activity diagram. The activity diagram is used to generate table called Activity Dependency table (ADT) and convert it into a directed graph called Activity Dependency Graph (ADG). The algorithm proposed applies to the graph for obtaining all the possible test paths. All the details are added to each test path using the ADT to produce the final test cases. Validating the generated test cases was achieved by Cyclomatic complexity. The proposed model was applied to three different case studies of web applications. Chouhan *et al.* (2012) proposed Test Case Generation based on Activity Diagram (TCBAD) using MBT for mobile application. This approach was extended from previous study by Pakinam *et al.* (2011) for web application. TCBAD model proposed uses MBT for mobile application based on activity diagram, that were used in representing the workflows of stepwise activity and actions with support for choice, iteration and concurrently, the complexity was calculated using Cyclomatic Complexity. The proposed model automatically creates ADT from activity diagram, and then uses it to create ADG. TCBAD algorithm is introduced to generate test paths from ADG. Finally the test paths with the ADT are used to generate the final test cases. The proposed model saves time and effort and also increases the quality of generated test cases.

Vikas and Rajesh (2014) proposed an approach uses MBT supporting UML diagrams, namely sequence and web diagrams, for test case generation for web applications. Web diagrams provide the functional requirements and sequence diagrams provided the navigation web application in terms dynamic behavior of the application under test. The proposed approach



automatically generates test cases, overcomes problems related to omitted data, captures the dynamic behavior of web applications and improves test case information. While, Prachet and Abhishek (2013) proposed an approach uses MBT supporting UML diagrams, namely use case diagrams and activity diagrams, to generate and prioritize test cases for regression testing of web application. This approach combined activity diagrams to

get a clearer picture of test case generation and effective coverage path of those test cases.

The overall summary is described in tabular form for a quick review and comparison of the testing techniques. Table-1 shows a comparative summary of the related works for mobile application testing. Table-2 shows a comparative summary of the related works for web application testing.

Table-1. Review of the comparison of testing techniques of the related works regarding to mobile applications.

Author (Year)	Technique	UML diagram	Conclusion
Tobias and Volker (2014)	Model-Based Testing	Activity Diagram	The proposed approach can reduce test cases from design-time system models.
Ang <i>et al.</i> (2014)	Model-Based Testing	Activity Diagram	The proposed approach can reduce the overall test time, testing efforts as well as improving test adequacy.
Chouhan <i>et al.</i> (2012)	Model-Based Testing	Activity Diagram	The proposed approach model saves time and effort and also increases the quality of generated test cases.

Table-2. Review of the comparison of testing techniques of the related works regarding to web applications.

Author (Year)	Technique	UML diagram	Conclusion
Vikas and Rajesh (2014)	Model-Based Testing	Sequence Diagram and Web Diagram	The proposed approach captures the dynamic and improves test case information.
Prachet and Abhishek (2013)	Model-Based Testing	Use Case Diagram and Activity Diagram	The proposed approach can get a clearer picture of test case generation and effective coverage path of those test cases.
Pakinam <i>et al.</i> (2011)	Model-Based Testing	Activity Diagram	The proposed approach can save time, effort and increasing the overall testing process performance.

Table-1 and Table-2 show the techniques and various UML diagrams that were used for mobile application testing and web application testing. Most researchers used MBT technique and UML activity diagrams for mobile application testing and web application testing. This approach can reduce the overall test time, testing efforts as well as improving test adequacy. UML activity diagrams are one of the important UML models used in representing the workflows of stepwise activities and actions with support for choice, iteration and concurrency. Moreover, UML activity diagrams can be utilized to describe the business and operational step-by-step workflows of components in a system, as it has all the characteristics that can improve the quality of the automatically generated test cases (Pakinam *et al.*, 2011). Therefore, based on related works, TCBAD model to generate test cases based on UML activity diagram was used in this study as proposed by Chouhan *et al.* (2012).

METHODOLOGY

In this study four phases are undertaken, as shown in Figure-1. These phases are detailed and discussed in the following:

Phase 1 (Developing UML activity diagrams): The flexibility in describing various control flows makes UML activity diagram a promising candidate for application behavior modeling, reducing overall testing efforts as well as improving test adequacy. The application behavior derivation and its depiction using UML activity diagrams to describe either sequential or concurrent workflows of stepwise activities and actions (Aye and Myat, 2014). This study used the TCBAD model (Chouhan *et al.*, 2012) of four case studies of the mobile and web applications. The TCBAD model includes the development of the UML activity diagram to describe the application behaviour, which transforms the UI to the UML activity diagram. The overall activity diagram describes the business and operational activities step-by-step for all of the main



functions in the application interface. The case studies are eBook store, namely Inktera Books and Kobo Books for mobile applications, which are available at Google play apps. While, Inktera and Kobo for web applications, which are available at (<https://www.inktera.com>, <http://www.kobobooks.com>). All case studies have same main functions, namely Sign in, Search, Browse and Library.

Phase 2 (Generating test cases using mbt technique based on activity diagram): The test cases were generated from four case studies for web and mobile applications by using the MBT technique based on TCBAD model proposed by Chouhan *et al.* (2012). This model used to generate test cases based on activity diagrams developed. TCBAD model includes four steps after application behavior modeling by using UML activity diagrams. Its steps are generation of ADT (The ADT has six columns as shown in Table-3), generation of ADG (The ADG is generated automatically from the ADT constructed), generation of test paths based on the TCBAD algorithm as shown in Figure-2 and validation number of test paths based on Equation (1) of the Cyclomatic Complexity (CC) (Agarwal *et al.*, 2010), and finally the test paths with the ADT are used to generate the final test cases.

Phase 3 (Evaluation of performance measurements): HP LoadRunner 12.02 tool (HP Loadrunner, 2015) is able to monitor and assess the real-time performance for both of mobile applications and web applications. The usage of HP LoadRunner 12.02 tool can shorten test time in maximum limit and optimize performance (Prakash and Gopala, 2012). The HP LoadRunner 12.02 tool was used to evaluate performance measurements Hits per second, Throughput and Memory utilization by execution of test cases generated and to achieve the results for each case study.

Phase 4 (Comparative analysis): The results were compared and analyzed after they obtained values of the each Hits per second, Throughput and Memory utilization by execution of the test cases generated of four case studies by the HP LoadRunner 12.02 tool. The Mean Square Error (MSE) is defined as (Huang and Kuo, 2002) in this quantitative comparison; MSE was used because it is easier to understand. A smaller MSE indicates a smaller fitting error, and better overall performance. The MSE values of the each Hits per second, Throughput and Memory utilization were considered as comparison criteria the differences and similarities among the case studies of web and mobile applications based on Equation (2) (Salkind, 2010).

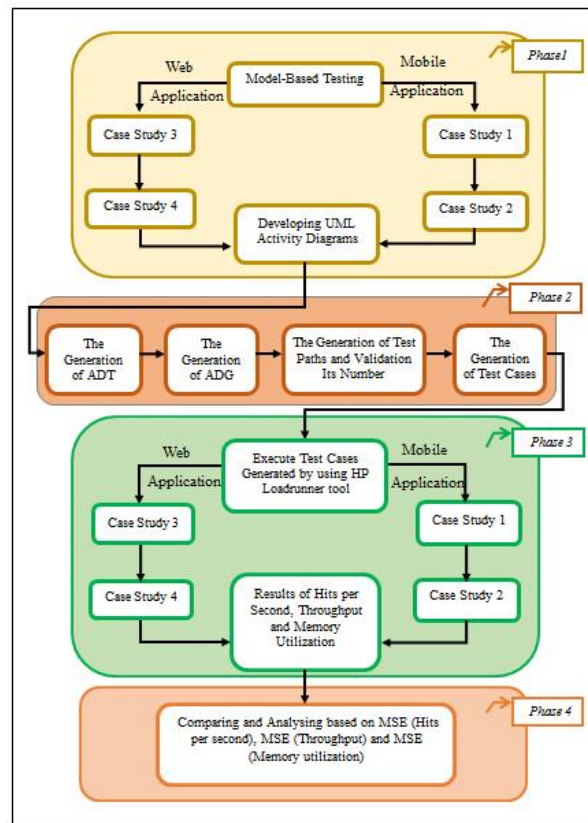


Figure-1. Research framework.

Table-3. Activity dependency table. (Chouhan *et al.*, 2012).

NO	Vertex	Activity Name	Dependency nodes	In degree value	Dependent nodes	Out degree values
----	--------	---------------	------------------	-----------------	-----------------	-------------------

Algorithm for TCBAD (Test case generation based on Activity Diagram.)
Input: ADG.
Output: Test Paths // This algorithm is used to generate automatic test paths using ADG. Root node is a node whose in degree value is null and End node shows out degree value is null.
STEP 1: Select Root node and End node. // Root node and End node decided on the Basis of in order and out order values.
STEP 2: Mark Root node as visited, and push this node onto stack.
STEP 3: Scan the whole graph and push every adjacent node of root node into the stack.
STEP 4: Repeat STEP 1 to STEP 3 till all paths are covered.

Figure-2. The TCBAD algorithm for generating the test paths. (Chouhan *et al.*, 2012).



$$CC(G) = E - N + 2 \quad (1)$$

E = Number of Edges,
 N = Number of Nodes.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (2)$$

N = Number of Values,
 Y = Actual Value,
 \hat{Y} = Estimated Value.

COMPARATIVE ANALYSIS

According to the results obtained by applying MBT technique based on the TCBA model using UML activity diagrams on four case studies of mobile and web applications in order to performance testing. The testing process was dependent on the execution of the test cases generated by using the HP Loadrunner tool. The MSE for each measurement is considered as comparison criteria in this study. Table-4 shows the comparative analysis of the MSE value of the Hits per second, Throughput and Memory utilization for each case study of mobile and web applications.

Table-4.Comparative analysis of case studies of mobile and web applications.

Application Type		MSE (Hits per Second)	MSE (Throughput)	MSE (Memory Utilization)
Mobile Applications	Case study 1	0.3795	0.00002	16.06727
	Case study 2	0.3237	0.00002	15.42017
Web Applications	Case study 3	0.23955	0.00002	14.21717
	Case study 4	0.24762	0.00001	12.26179

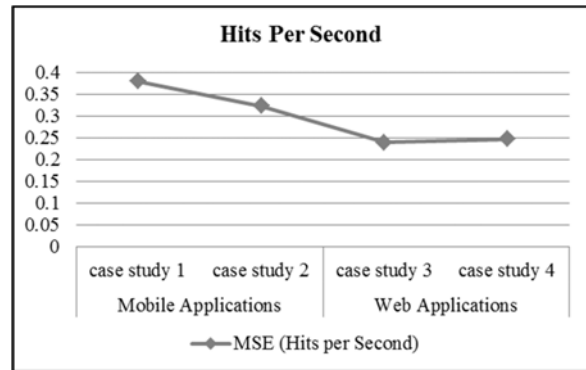


Figure-3. Comparative analysis diagram for case studies of mobile and web applications of the MSE value of the Hits per second.

Figure-3 shows the comparative analysis of the MSE value based on the Hits per Second for mobile and web applications. Where the MSE of the Hits per Second of Case Study 1 of mobile application is 0.3795 and Case Study 3 of web application is 0.23955, while Case Study 2 of mobile application is 0.3237 and Case Study 4 of web application is 0.24762. This implies that there is a difference in performance between mobile applications and web applications in terms of measurement of the Hits per Second. The Case Study 3 has the smallest MSE of the Hits per Second. This indicates that Case Study 3 of the web application has the smallest fitting error, and better performance in terms the number of HTTP requests sent to the web server during a specific time period of the performance test.

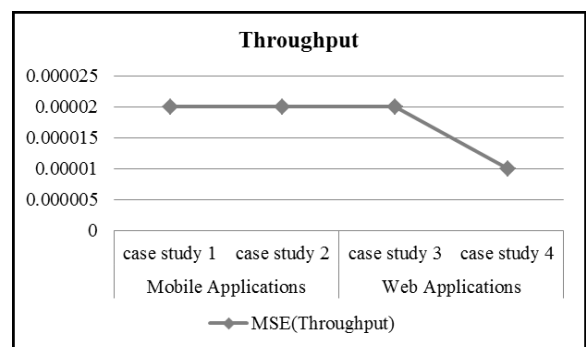


Figure-4. Comparative analysis diagram for case studies of mobile and web applications of the MSE value of the throughput.

Figure-4 shows the comparative analysis based on the MSE value of the Throughput for mobile and web application. The MSE of the Throughput for Case study 1 of mobile application is 0.00002. It is similar to Case Study 3 of web application. While the MSE of the Throughput for Case Study 2 of mobile application is 0.00002 and Case Study 4 of web application is 0.00001. This implies that there is slightly different in performance between mobile applications and web applications in terms of measurement of the Throughput. The Case Study 4 has



the smallest MSE of the Throughput. This indicates that Case Study 4 of the web application has the smallest fitting error and better performance in terms the server response during a specific time period of the performance test.

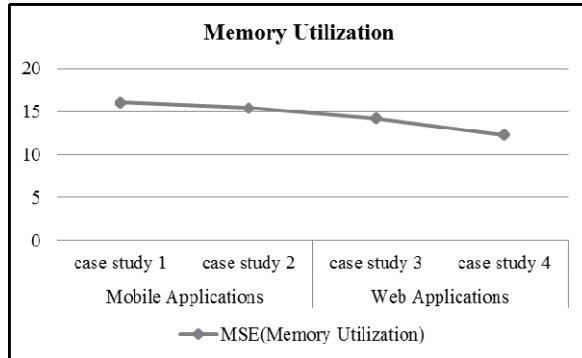


Figure-5. Comparative analysis diagram for case studies of mobile and web applications of the MSE value of the memory utilization.

Figure-5 shows the comparative analysis based on the MSE value of the Memory utilization for mobile and web application. The MSE of the Memory utilization of Case Study 1 of mobile application is 16.06727 and Case Study 3 of web application is 14.21717, while Case Study 2 of mobile application is 15.42017 and Case Study 4 of web application is 12.26179. This implies that there is a difference in performance between mobile applications and web applications in terms of measurement of the Memory utilization. The Case study 4 has the smallest MSE of the Memory utilization. This indicates that Case Study 4 of the web application has the smallest fitting error, and better performance in terms the percentage of memory used during the scenario execution of the performance test.

In terms of mobile and web applications, the comparison has been made for Case Study 1 (of mobile application) with Case Study 3 (of web application) based on number of test paths, number of test cases and MSE of performance measurements: Hits per Second, Throughput and Memory Utilization, as shown in Table-5. Likewise, the comparison has been made for Case Study 2 (of mobile application) with Case Study 4 (of web application), as shown in Table-6.

Overall, the comparative analysis indicates that the variation was clearly between the case studies of both mobile and web applications in terms of the number of test paths, number of test cases, MSE of Hits per Second, MSE of Throughput and MSE of Memory Utilization. This implies, the mobile applications, in terms of the number of test paths and number of test cases lesser than web applications. On the other hand, the mobile applications, in terms of the MSE criterion have high Hits per Second, high Throughput and high Memory Utilization, which means that the mobile applications have poor performance compared to web applications. It can be concluded, taking into consideration more emphasis on the mobile testing.

Table-5. The comparative analysis between Case Study 1 (of mobile application) with Case Study 3 (of web application).

The comparison criteria		Case study 1 (mobile app)	Case study 3 (web app)
Number of Test Paths		19	23
Number of Test cases		19	23
MSE of Performance Measurements	Hits per Second	0.3795	0.23955
	Throughput	0.00002	0.00002
	Memory Utilization	16.06727	14.21717

Table-6. The comparative analysis between case study 2 (of mobile application) with case study 4 (of web application).

The Comparison Criteria		Case Study 2 (mobile app)	Case Study 4 (web app)
Number of Test Paths		21	22
Number of Test cases		21	22
MSE of Performance Measurements	Hits per Second	0.3237	0.24762
	Throughput	0.00002	0.00001
	Memory Utilization	15.42017	12.26179

CONCLUSIONS

The number of mobile and web applications is growing rapidly, which creates an impetus for researchers and developers to come up with adaptation testing techniques for both of these kinds of applications to ensure their reliability. The MBT approach can generate highly efficient test cases with the minimum number of steps, saving time, effort and increasing the quality of generated test cases thus improving the overall testing process performance. Moreover, it includes validation of the number of generating test paths during the generation process to ensure their coverage. In addition, the UML activity diagram is considered a de-facto standard for development models and its usage is widespread in the testing process according to related works. During the practical experience in this study by using activity diagram for each of mobile applications and web applications, which is showing that there are some difficulties when transforming the UI to the activity diagram in the applications of mobile compared to the web application, because, there were a fundamental differences between mobile applications and web applications in terms of navigation in the UI. However, this study has been achieved its objectives.

The experimental results showed that the disparity between the mobile applications and web applications was obvious. Based on the comparison analysis for software testing of mobile applications versus web applications that was the web applications were lesser than mobile applications for software testing of four case



studies in terms each of the Hits per second, Throughput and Memory utilization. Consequently, mobile applications need more attention in the testing process.

In future work, the authors may aim to expand work to developing more specific software testing tool for mobile applications.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Rosziati Ibrahim, for the continuous support, motivation, and immense knowledge. Also, I would like to express my deepest gratitude to Universiti Tun Hussein Onn Malaysia (UTHM), and especially the Faculty of Computer Science and Information Technology / Department of Software Engineering for giving me the opportunity to further my Master's degree.

REFERENCES

- Agarwal, B., Tayal, S. and Gupta, M. 2010. Software Engineering and testing An Introduction. Hingham, Toronto: Infinity Science Press, Jones and Bartlett's books. ISBN: 978-1-934015-55-1.
- Ang, L., Zishan, Q., Mingsong, C. and Jing, L. 2014. ADAutomation: An Activity Diagram Based Automated GUI Testing Framework for Smartphone Applications. Proceedings of the Eighth IEEE International Conference on Software Security and Reliability (SERE), June 30, 2014-July 2, 2014. San Francisco, CA: IEEE. pp. 68-77.
- Aye, K. and Myat, M. 2014. An Efficient Approach for Model Based Test Path Generation. International Journal of Information and Education Technology, 5(10), pp. 763-767. doi: 10.7763/IJiet.2015.V5.607.
- Chouhan, C., Shrivastava, V. and Parminder, S. 2012. Test Case Generation based on Activity Diagram for Mobile Application. International Journal of Computer Applications, 57(23), pp. 4-9. doi: 10.5120/9436-3563.
- Gahran, A. 2011. What'S A Mobile App?. Retrieved on October 02, 2014, from <http://www.contentious.com/2011/03/02/whats-a-mobile-app/>.
- HP Loadrunner. 2015. HP Loadrunner 12.02, Retrieved on March 01, 2015 from <http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/index.html>
- Huang, C. and Kuo, S. 2002. Analysis and assessment of incorporating logistic testing effort function into software reliability modelling. IEEE Trans. Rel., 51(3), pp. 261-270.
- Imran, A. and Roopa, S. 2012. Quality Assurance And Integration Testing Aspects In Web Based Applications. International Journal of Computer Science Engineering and Applications (IJCSEA), 2(3), 109-116. doi: 10.5121/ijcsea.2012.2310.
- Kalpan, B. and Ramakanth, K. 2012. A Survey on Performance Testing Approaches of Web Application and Importance of WAN Simulation in Performance Testing. International Journal on Computer Science and Engineering (IJCSE), 4(5).pp. 883-885. ISSN: 0975-3397.
- Ke, H., Xiao-Hong, L. and Zhi-Yong, F. 2010. Approach to activity diagram model driven testing for Web applications. Journal of Computer Applications, 30(9), pp. 2365-2369.
- Maryam, A. and Rosziati, I. 2014. A Comparative Study of Web Application Testing and Mobile Application Testing. Advanced Computer and Communication Engineering Technology, Lecture Notes in Electrical Engineering. Springer International Publishing, 315, pp. 491-500. doi: 10.1007/978-3-319-07674-4_48.
- Pakinam, N., Nagwa, L., Mohamed H. and Mohamed F. 2011. A Proposed Test Case Generation Technique Based on Activity Diagrams. International Journal of Engineering and Technology, 11(3), pp. 35-52.
- Prachet, B. and Abhishek, K. 2013. Model Based Regression Testing Approach of Service oriented Architecture (SOA) Based Application: A Case Study. International Journal of Computer Science and Informatics, 3(2), pp. 11-16.
- Prakash, V. and Gopala, S. 2012. Cloud Computing Solution- Benefits and Testing Challenges. Journal of Theoretical and Applied Information Technology, 39(2), pp. 114-118. ISSN: 1992-8645.
- Salkind, N. 2010. Encyclopedia of Research Design. SAGE. ISBN: 9781412961271.
- Sandeep, K., Sangeeta, S. and Gupta, J. 2012. A Novel Approach for Deriving Test Scenarios and Test Cases from Events. Journal of Information Processing Systems, 8(2), pp. 213-240. doi: 10.3745/JIPS.2012.8.2.213.
- Sheetal, S. and Joshi, S. 2012. Identification of Performance Improving Factors for Web Application by Performance Testing. International Journal of Emerging Technology and Advanced Engineering, 2(8), pp.433-436. ISSN 2250-2459.
- Sumit, M. and Narendra, K. 2014. Model Based Testing Of Website. International Journal on Computational Sciences and Applications (IJCSA), 4(1), pp. 143-152. doi:10.5121/ijcsa.2014.4114.
- Tobias, G. and Volker, G. 2014. A Model-Based Approach to Test Automation for Context-Aware Mobile Applications. Proceedings of the ACM 29th Annual



Symposium on Applied Computing. New York, NY, USA: ACM, pp. 420-427. doi:10.1145/2554850.2554942.

Vikas, S. and Rajesh, B. 2014. Model based Test Cases Generation for Web Applications. International Journal of Computer Applications (0975-8887), 92(3), pp. 23-31.

Weibiao, Z., Fang, H., Shuanshuan, B., Lei, D., Jie, F., Ling, J. 2014. Performance Evaluation of Concurrent-access-intensive WebGIS Applications Based on CloudStack. International Journal of Advancements in Computing Technology (IJACT), 6(5). pp. 41-52.

Zhang, G., Rong, M. and Zhang, J. 2007. A Business Process of Web Services Testing Method Based on UML2.0 Activity Diagram. IEEE Workshop on Intelligent Information Technology Application. Zhang Jiajie: IEEE, pp. 59- 65. doi:10.1109/IITA.2007.83.