

# IMPLEMENTATION OF VARIOUS SOFTWARE TESTING TECHNIQUES ON MERIT BASED MANAGEMENT SYSTEM FOR BEHAVIORAL AUTISM SPECTRUM DISORDER (MBMSB-ASD)

<sup>1</sup>ABD RASID MAMAT, <sup>2</sup>ROHANA ISMAIL, <sup>3</sup>MOHAMAD AFENDEE MOHAMED, <sup>4</sup>MOHD FADZIL ABD KADIR, <sup>5</sup>ANIS NURSYAFIQAH AB WAHAB, <sup>6</sup>BASHIR MUZAKKARI

<sup>1,2,3,4,5</sup>Computer Science Department, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, 22200 Terengganu, Malaysia

<sup>6</sup>Department of Computer Science, Northwest University Kano, Nigeria

E-mail: <sup>1</sup>arm@unisza.edu.my, <sup>2</sup>rohana@unisza.edu.my, <sup>3</sup>mafende@unisza.edu.my, <sup>4</sup>fadzil@unisza.edu.my, <sup>5</sup>anissyah79@gmail.com, <sup>6</sup>muzakkar@gmail.com.

## ABSTRACT

Software testing is an important process in software development. If proper testing is not done on the developed software, this can cause major problems for the software. Among the purposes of system testing is to test compliance with software requirements, detect bugs, errors and so on. In this paper various testing techniques are discussed and then implemented into a web-based software that has been developed known as Merit Based Management System for Behavioral of Autism Spectrum Disorder (MBMSB-ASD). The advantages and disadvantages of each technique are discussed while testing the software above. As a conclusion, which technique or combination of techniques is more appropriate to use depends on attributes such as cost, time and people allocated.

**Keywords:** *Testing, Error, Bug, Requirements, Software*

## 1. INTRODUCTION

Testing is defined as an evaluation process whether a system meets the original requirements agreed earlier or vice versa. It is primarily a process that includes verification and validation stages that test the developed system meets user determined needs. To check whether software meets user needs and produces functional applications, software testing becomes a critical process [1-4]. Testing must be done before the software release and so as to discover errors in a timely manner and ensure that the functionality of the software works as needed. It also minimizes the defects found after the product has been used [5].

## 2. LITERATURE REVIEW

Software testing is an essential and mandatory phase of software development life cycle [6]. Software testing is the process of testing bugs in lines

of code of a computer program that can be performed by manual or automation testing.

The theory of software testing involves problem definitions of testing such as test team, failure after testing, manual testing, uncertainty principle, participation, and incorrect test case selection [7]. A research by [8] discusses various system testing techniques. Among the techniques discussed are unit testing, integration testing and system testing while the methodology is white box and black-box testing and finally software testing tools including Ranorex, RFT, Janova. In conclusion the authors concluded, software testing research is the driving for development of an application.

According to [9], they explained that a unit test is a piece of code written by a developer with the purpose of running/testing a specific function on a small area of code. The importance of unit testing in software development is to detect problematic areas and reduce bugs in source code [10].

In [11] the authors apply usability testing for user satisfaction on the prototype system. The test investigates the capability of the software to be understood, learned, used and attractive to the user when used under specified conditions and this leads to user satisfaction and acceptability in using the software.

In addition to manual testing, various tools on automation tests that can be used on different platforms, and this was discussed by [12]. Automatic testing tools are important because they make it easier for testers to test the entire system much easier.

In addition, Artificial Intelligence (AI) has also been used in software testing. The authors in [27] have used AI in their study which aims to recognize the impact of AI technologies on various software testing activities in the software testing life cycle (STLC). The authors identify seven software testing activities that are most improved by AI techniques such as GA, Reinforcement Learning and ANN. Problems and challenges using AI techniques have also been identified.

In [28], the authors study large scale empirical testing of open-source Android (Apps). The four elements that are evaluated firstly how well the application is actually tested, secondly how good is the existing test design, thirdly what is its effectiveness, and finally how much manual testing can reduce the risk of having defects in the production code. The author also created a focus group (5 experts in android testing) to discuss the findings and future direction of findings.

Based on finite automata theory, the authors in [30] discusses the features of the implementation of algorithms for the functioning of the microcontroller for software testing system. This theory is used in an enlarged block diagram of a testing system at the software and hardware levels and its results provide invaluable help to developers in testing the system.

### 3. SOFTWARE TESTING TECHNIQUES

Software testing is the process of exercising a software system using a variety of input with the aim of confirming its behavior and discovering errors or faults [13-14] [29]. The bugs or defects are known as faults and it can cause failures in their software systems. According to [15], the author categorized testing in three stages namely development testing, release testing and user testing.

#### 3.1 Development testing

This testing is performed during the system development process [16 -18]. Development testing includes all testing activities that are carried out by the team developing the system. The person involved for this testing normally are the system designers and programmers. There are three stages of development testing namely unit testing, component testing dan system testing.

The aims for unit testing are to focus on testing of the functionality of objects or methods, meanwhile component testing is used to test the component interfaces that provides access to the component functions. Lastly, system testing is used to test the system as a whole. System testing during development involves integrating components to create a version of the system and then testing the integrated system. System testing checks if the components are compatible, interact correctly, and transfer the right data at the right time across their interfaces.

#### 3.2 Requirement based testing

Requirements-based testing is a testing approach in which test cases, conditions and data are derived from requirements. It includes functional tests and also non-functional attributes such as performance, reliability or usability. The requirements-based testing process addresses two main issues, the first aims to verify that the requirements are correct, complete, unambiguous and logically consistent. Second, to design a necessary and sufficient set of test cases (from a black box point of view) from the requirements to ensure that the design and the code fully meet the requirements [19].

This test can also be used to compare the objective requirements and why this project needs to be implemented (early stage) with the output system that has been completed. Adherence or not to these objective requirements indicates the success of a system or otherwise.

#### 3.3 Release testing

This testing is usually a black box testing process where tests are only derived from the system specification. Release testing is the process of testing a particular release of a system that is intended for use outside of the development team [20-21].

The primary goal of the release testing process is to convince the customer that the system is good enough for use. Release testing, therefore, has to show that the system delivers its specified

functionality, performance and dependability, and that it does not fail during normal use.

The different between release testing with the system testing where system testing is done by the development team and should focus on discovering bugs in the system (defect testing). While the objective of release testing is to check that the system meets its requirements and is good enough for external use (validation testing) [13].

### 3.4 Scenario testing

Scenario testing is a statement that describes the application functionality of a system to be tested. It is used for end-to-end testing of a feature or function and is usually derived from use cases. According to [22], scenario testing is best suited for complex transactions or events, to study end-to-end delivery of the benefits of the program, to explore how the program will work in the hands of an experienced user, and to develop more persuasive variations of bugs found using other approaches.

### 3.5 Performance testing

Performance testing refers to that phase of testing where an application is tested specifically with performance requirements in view [14],[23]. The most common types of performance testing are volume testing, stress testing, spike testing, scalability, load testing and endurance testing, and stress testing.

For example, a compiler might be tested to check if it meets the performance requirements stated in terms of number of lines of code compiled per second. Others example of performance requirements are stated with respect to a hardware and software configuration such as an application might be required to process a number of billing (eg. 2000) transactions per minute on a specific Intel processor-based machine running a specific operating system.

### 3.6 Test Driven Development

Test-driven development (TDD) is an approach to program development in which you interleave testing and code development [24-25].

The TDD is a software engineering practice that requires unit tests to be written before the code they are supposed to validate. TDD is proposed to be a suitable methodology for agile development, as it encourages thinking on the functionality ahead of writing the code, refactoring the code frequently and continuously maintaining high code quality, and

increasing the maintainability and reliability of the code through running automated regression test cases [26].

### 3.7 User testing

User or customer testing is a stage in the testing process in which users or customers provide input and advice on system testing. There are three different types of user testing, namely Alpha testing, Beta testing and Acceptance testing [15]. Alpha testing is a type of user acceptance testing and aims to identify all possible issues and errors that the system may produce and resolve all those issues before the final product is delivered to the end user or to the public. This testing is done before beta testing and at an early stage which is near the end of the software development life cycle.

Beta testing refers to software being made available to a larger group of users to allow them to try and troubleshoot problems they encounter with system developers. As it is the final stage of testing, it is done in production or real environment. While acceptance testing is where customers test a system to decide whether or not it is ready to be accepted from the system developers and deployed in the customer environment. It is also testing used to determine if a system meets the determined requirements prior to its delivery.

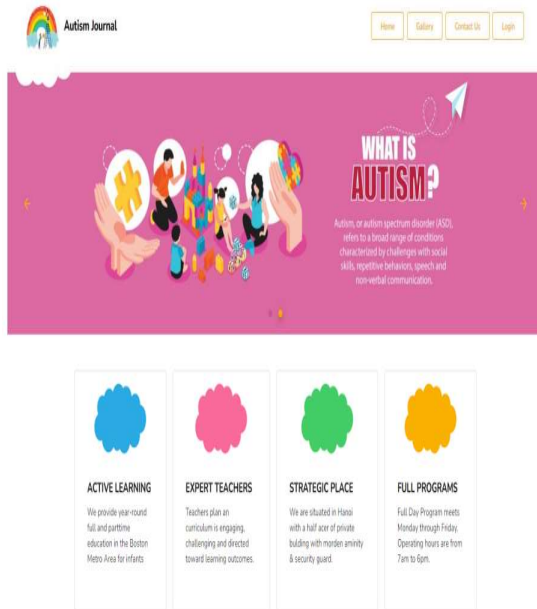
## 4. CASE STUDY- SOFTWARE TESTING TECHNIQUES IN MERIT BASED MANAGEMENT SYSTEM FOR BEHAVIORAL OF AUTISM SPECTRUM DISORDER (MBMSB-ASD)

### 4.1 Introduction of Merit Based Management System for Behavioral of Autism Spectrum Disorder (MBMS-ASD)

Autism, or autism spectrum disorder (ASD) refers to a broad range of conditions characterized by challenges with social skills, repetitive behaviours, speech and non-verbal communication. Taking care of autistic child is a huge responsibility, especially if you are opening an autistic centre which requires you to take care of a group of autistic children. To understand each and every one of them are not easy, records are needed to help the parents or caregivers to note about the children's conditions. Therefore, to achieve the purposes, we have conducted research and develop the MBMSB-ASD system that we believed can help to solve or reduce the problem.

The aims of this system are divided into three, firstly to study the behaviour of the autistic children to record their progress and development, secondly to design and develop a system and finally to test and evaluate the functionalities of the system in executing an accurate result. To ensure the system is free from errors or bugs, software testing techniques have been used. The next paragraph describes the testing techniques that were applied to MBMSB-AB. Figure 1 shows the main page of the MBMS-ASD system.

Figure 1: The Main Page Of The MBMS-ASD System



4.2 Development testing

This sub section explains an example of the test case for the development testing strategies. Table 1 shows an example unit testing test case for login module of MBMSB-AB.

An example of component testing test case is Kid Information module (MBMSB-AB) and shown in Table 2. Finally, for an example of system testing test case of MBMSB-AB is summarized in Table 3. According to the conduct of development testing, some advantages of disadvantages are shown in Table 4.

4.3 Release testing

This subsection shows an example of the release testing conducted in MBMS-ASD system as in Table 5. Test cases can be implemented similar to that of Table 3, but they are executed by customers or users and not by the development team (system testing).

Table 1: An Example Unit Testing Test Case For Login Module

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-UT-01 (Sign in user)	Device must connect to a good range of internet connection.	1. Email & Password incorrect. 2. Email correct & Password incorrect. 3. Both Email & Password incorrect. 4. Both Email & Password correct.	1. Click Login button 2. Enter test input 3. Click submit	For test input no.1, no.2 & no.3 -system should not access to the next page and test input no.4 the result is opposite.

Table 2: An Example Component Testing Test Case For Kid Information Module

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-CT-01 (Kid Info Module)	Pass TEST ID: TC-U-01	1.Kid Info: My Kids No, Nama, DOB, Religion. 2.Info Parent: IC No, Name, DOB, Religion, Address, Tel. No	1. Click Kid Profile button. 2. Enter test input 3. Click save button 4.Click retrieve button. 5. Enter test input-My Kid No. 6. Click button search	1.For Test Procedure No. 1 to 3, the data save in database. 2. For Test Procedure no. 4 to 6: If My Kid No not exist in database -display message. If My Kid No exist- display test input.

Table 3: An Example System Testing Test Case For All System Functions Including All Process, Buttons, Sub Menus And Others

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-ST-01 (As a whole system)	Device must connect to a good range of internet connection.	All system functions including the of all process, buttons, sub menus and others (as a whole system).	Execute all system functions, buttons, sub menus, reports and related to the system (as a whole system).	All system functions, buttons, sub menus, reports and related to the system work as expected

Table 4: Advantages And Disadvantages Of Development Testing

Types of development testing	Example part of system/whole system	Advantages	Disadvantages
Unit testing (white box)	TC-UT-01 - MBMSB-AB	<ul style="list-style-type: none"> <li>Improve coding and find the bug earlier</li> </ul>	<ul style="list-style-type: none"> <li>Used a lot of time to write test code and maintenance (find an error and correct its) and also time consuming and labour intensive (all the unit testing).</li> </ul>
Component testing	TC-CT-01	<ul style="list-style-type: none"> <li>It can detect defects in the module and verify the functionality of the software</li> </ul>	<ul style="list-style-type: none"> <li>There is a need to compromise with the requirements (either fulfil or not)</li> </ul>
System testing	TC-ST-01	<ul style="list-style-type: none"> <li>The first testing level in which the whole system is under test from end finish.</li> <li>To verifies the system against the business, functional and technical requirements of the stakeholder and also helps in finding maximum bugs before acceptance testing.</li> </ul>	<ul style="list-style-type: none"> <li>To test the system, requires a lot of time.</li> <li>Cost may also increase because relying upon the business necessities and application architecture.</li> </ul>

#### 4.4 Requirement Based testing (RBT)

This sub module includes examples of RBT used in the MBMS-ASD system and also the pros and cons of this test are described in this sub section. Example of RBT is shown in Table 7.

Table 8 shows the advantages and disadvantages of requirement-based testing.

#### 4.5 Scenario testing

The description of an example scenario testing in the MBMS-ASD system and the pros and cons of

this testing are described in this sub module. Table 9 shows an example of scenario test case for login module.

Based on the tests that have been done, this test has several advantages and disadvantages, and they are shown in Table 10.

**4.6 Scenario testing**

This sub module explains an example of the performance testing. One example of the performance testing’s is load testing. Load testing aims to checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live. For example, 20 user’s logins simultaneously into the MBMS-ASD system, as shown in Table 11 while and Table 12 portray an advantages and disadvantages of performance testing (example load testing).

**4.7 User testing**

This sub module explains briefly user testing for MBMS-ASD system. This testing is divided into three, namely alpha, beta and acceptance testing.

**4.7.1 Alpha testing**

Table 13 shows an example of alpha testing test case.

**4.7.2 Beta testing**

Table 14 shows an example of beta testing test case.

**4.7.3 Acceptance testing**

Table 15 present an example of acceptance test case.

*Table 5: An Example Releases Testing Test Case For All System Functions Including The Functions Of All Process, Buttons, Sub Menus And Others (As Whole A System).*

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-RT-01 (As a whole system)	Device must connect to a good range of internet connection	All system functions including the functions of all process, buttons, sub menus and others	Execute all system functions, buttons, sub menus, reports and related to the system	All system functions, buttons, sub menus, reports and related to the system work as expected

*Table 6: Advantages And Disadvantages Of Release Testing.*

Types of testing	Example part of system/whole system	Advantages	Disadvantages
Release testing	TC-RT-01	<ul style="list-style-type: none"> <li>System releases are for customers and users, they can easily identify and suggest necessary modifications based on users’ perspective.</li> </ul>	<ul style="list-style-type: none"> <li>May not get better feedback due to varying user views and ideas.</li> <li>May increase the effort of testing as the users may not have clear view of the system.</li> <li>Testing perspective may sometimes be deviated to some other view.</li> </ul>

Table 7: An Example Requirements-Based Testing Test Case For Caregiver.

Test ID	Pre-conditions	Test Input – Examples validate with the objectives of the system.	Test Procedure	Expected Result
TC-RBT-01 (Caregiver requirements)	Device must connect to a good range of internet connection	Example of the caregiver requirements: <ul style="list-style-type: none"> <li>The system allows the caregiver to log in to record all activities of the autistic child, add and delete photo of the children into the gallery.</li> <li>The caregiver can also add, edit and delete a child's profile and view the child's progress report.</li> </ul>	Execute all modules involving caregivers	The system expected to fulfil of the objectives of caregiver requirements

Table 8: Advantages And Disadvantages Of Requirements-Based Testing.

Types of testing	Example part of system/ whole system	Advantages	Disadvantages
requirements-based testing	TC-RBT-01	<ul style="list-style-type: none"> <li>Reduces the time to deliver by allowing testing to be performed concurrently with the rest of the development activities.</li> <li>To discover and fix the low-quality requirements thereby producing valid inputs that contribute greatly in defining a clear project scope.</li> <li>To find requirements errors before it becomes too expensive to fix and manage the changes.</li> </ul>	<ul style="list-style-type: none"> <li>It is possible to easily overlook important aspects of software quality and this may be because requirements too brief and may have trouble capturing important elements of software design</li> </ul>

Table 9: An Example Scenario Test Case For Login Module

Test ID	Scenario	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-SC-01	Verify that the user is not able to login with incorrect credentials.	Device must connect to a good range of internet connection.	1. Email & Password incorrect. 2. Email correct & Password incorrect. 3. Both Email & Password incorrect.	1. Click Login button 2. Enter test input 4. Click submit	For test input no.1, no.2 & no.3 -system should not access to the next page.

Table 10: Advantages And Disadvantages Of Scenario Testing

Types of testing	Example part of system	Advantages	Disadvantages
Scenario testing	TC-SC-01	<ul style="list-style-type: none"> <li>To conduct this testing relatively faster than testing using test cases, good test coverage since the test scenarios are derived from user stories(functionality) and saves a lot of time (suitable for projects having time constraints).</li> </ul>	<ul style="list-style-type: none"> <li>The practice of scenario is very time-consuming. Therefore, there could be a wish to condense scenario building to a half-day or one day activity. However, this may not give participants enough time.</li> <li>A more qualitative approach has to put a strong emphasis on the selection of suitable participants/ experts, and in practice this could not be an easy task to fulfil.</li> </ul>

Table 11: An Example Performance Testing Test Case For Login Module

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-PT-01	Device must connect to a good range of internet connection	30 users input logins and passwords simultaneously	<ol style="list-style-type: none"> <li>Click Login button</li> <li>Enter test input</li> <li>Click submit</li> </ol>	There should be no time delay for users to login to the system

Table 12: Advantages And Disadvantages Of Performance Testing (Example Load Testing)

Types of development testing	Example part of system	Advantages	Disadvantages
Load testing	TC-PT-01	<ul style="list-style-type: none"> <li>To identifying performance bottlenecks before users use them and provides a sense of confidence and reliability in the application's performance.</li> <li>To helps in configuring the most optimal infrastructure in preparation for later use. This can save infrastructure costs (without buying unnecessary equipment).</li> </ul>	<ul style="list-style-type: none"> <li>Increase cost because many of the load testing tools are licensed and charge a good amount of money for the license.</li> <li>Although there are free and open-source tools like JMeter, the required environment setup should be as close as possible to the production environment setup. This again leads to additional resources and costs.</li> <li>Incorrectly configured or scripted load test plan/script can lead to false performance issues which take a considerable amount of time and resources.</li> </ul>



Table 13: An Example Alpha Testing Test Case For All System Functioning (As A Whole System) And Conduct By users

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-AT-01	Device must connect to a good range of internet connection.	All system functions including the functions of all process, buttons, sub menus and others (as a whole system)	Execute all system functions, buttons, sub menus, reports and related to the system	All system functions, buttons, sub menus, reports and related to the system work as expected which is no errors and bugs

Table 14: An Example Beta Testing Test Case And Conduct By End Users (Larger Group).

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-BT-01	Device must connect to a good range of internet connection.	All the functions that make up the system (eg, requirements functions, buttons, menus and so on as a whole system)	Beta testing checks for reliability, robustness and security of software and performed in the end-user's location and outside the company	The system should be reliability, robustness and secure

Table 15: An Example Acceptance Testing Test Case For Complete MBMS-ASD System

Test ID	Pre-conditions	Test Input	Test Procedure	Expected Result
TC-AT-01	Device must connect to a good range of internet connection.	The complete system	Test all functions and performance of a system and the meets the needs of stakeholders (requirement)	The system successfully meets all the tests in the test procedure

Table 16: Advantages And Disadvantages Of User Testing

Types of user testing	Advantages	Disadvantages
Alfa testing (Using Black box testing and white box testing)	<ul style="list-style-type: none"> <li>• The black box testing will test the system's input and output functionality at a high level. The white box testing to test the system's design and internal structure. This is important to verify the product's input and output flows, for all required and possible scenarios.</li> <li>• Improved software quality where the system is tested in a simulated environment similar to the environment that will be used.</li> <li>• Various insights into usability and reliability are available. It gives room to</li> </ul>	<ul style="list-style-type: none"> <li>• In this testing the complete product will be tested at a high level and in-depth, using the different black box and white box techniques. This means the test execution cycle takes a longer time to complete.</li> <li>• The virtual environment creates limitations for non-functional requirement testing. For example, aspects like maintainability, in-depth security and reliability are tricky to test,</li> </ul>

	understand how the system will behave when it is released to end users.	simply because the alpha test takes place in a virtual environment.
Beta testing	<ul style="list-style-type: none"> <li>• It provides an additional level of testing/product validation to the development lifecycle of the product and helps in greatly increasing the scope of testing (because used a large group of users).</li> <li>• In general, it is cost-effective as compared to alpha testing.</li> </ul>	<ul style="list-style-type: none"> <li>• The bug reporting of the identified bugs is not proper and systematic and a lot of duplicate bugs.</li> <li>• The testing and test environment is not under the control of the development team. It is often hard to reproduce the bugs because the testing environment differs from user to user</li> </ul>
Acceptance testing	<ul style="list-style-type: none"> <li>• It improves and understanding the requirements, increase the satisfaction of clients (client tests requirements definition according to needs) and help to deliver the system without bug.</li> </ul>	<ul style="list-style-type: none"> <li>• Sometimes the users cannot comprehend the test cases defined by a software tester. Hence a software tester must assist the users by translating them to carry out the testing.</li> </ul>

## 5. CONCLUSION AND DISCUSSION

In conclusion, various types of testing techniques that have been performed on the MBMS-ASD system have successfully reduced unit errors, system errors and thus met the needs of users. However, such testing involves more costs in terms of time, money and energy required.

To reduce this, without affecting performance, one of the steps that can be taken is the integration of the test process. For example, in development testing there are three stages of development testing namely unit testing, component testing and system testing may be combined. The integration of this test is effective if it includes the three states above. Next, there are tests that overlap each other and the use of only one of the tests. The selection of overlapping tests and MBMS-ASD modules must be accurate and suitable are important. Finally perform testing techniques simultaneously, for example requirement-based testing and scenario-based testing. Same as above, performance is also affected by the selection of the right technique and MBMS-ASD modules.

In addition, in the future, performance-based testing such as stress, endurance and usability should also be performed.

## ACKNOWLEDGMENT

This project is partially funded by the Center for Research Excellence, Incubation Management Center, Universiti Sultan Zainal Abidin, Malaysia.

## REFERENCES:

- [1] Jupudy, I., & Manjula, R. (2016). Comparative analysis of model based and formal based software testing methods. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3), 66-70.
- [2] Fachrurrazi, "The End User Requirement for Project Management Software Accurcay," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 2, pp. 1112-1121, April 2018.
- [3] S. K. Pandey and K. Mustafa, "A Comparative Study of Risk Assessment Methodologies for Information Systems," *Bulletin of Electrical Engineering and Informatics*, vol. 1, no. 2, pp. 111-122, June 2012.
- [4] Mamat, A. R., Mohamed, M. A., Azhar, A. F., Saany, S. I., Rawi, N. A., Amin, M. M., ... & Nor, M. A. M. (2019). Modelling decision support system for selection maahad tafiz center using analytical hierarchal analysis. *Indones. J. Electr. Eng. Comput. Sci*, 13(1), 35-40.
- [5] Okezie, F., Odun-Ayo, I., & Bogle, S. (2019, December). A critical analysis of software testing tools. In *Journal of Physics: Conference Series* (Vol. 1378, No. 4, p. 042030). IOP Publishing.
- [6] Harrold, M. J. (2000, May). Testing: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 61-72).
- [7] The Theory of Software Testing Adtha Lawanna (2022) *AU J.T.* 16(1): 35-40 (Jul. 2012)
- [8] Chauhan, R. K., & Singh, I. (2014). Latest research and development on software testing techniques and tools. *International Journal of*

- Current Engineering and Technology, 4(4), 2368-2372
- [9] Hunt, A., & Thomas, D. (2003). Pragmatic unit testing in Java with JUnit. The Pragmatic Bookshelf.
- [10] Aniche, M. F., Oliva, G. A., & Gerosa, M. A. (2013, March). What do the asserts in a unit test tell us about code quality? a study on open source and industrial projects. In 2013 17th European Conference on Software Maintenance and Reengineering (pp. 111-120). IEEE.
- [11] Mamat, A. R., Rasid, N. H. A., Ahmad, F., Abd Rawi, N., Mohamed, M. A., Awang, W. S. W., & Ghazali, F. (2017, October). Pre school selection system using Analytic Hierarchy Process (AHP): Implementation and usability study. In 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 7-12). IEEE.
- [12] Bhateja, N. (2015). A study on various software automation testing tools. International Journal of Advanced Research in Computer Science and Software Engineering, 5(6), 1250-1252.
- [13] Fraser, G., & Rojas, J. M. (2019). Software testing. In Handbook of Software Engineering (pp. 123-192). Springer, Cham.
- [14] Graham, D., Black, R., & Van Veenendaal, E. (2021). Foundations of software testing ISTQB certification. Cengage Learning.
- [15] Sommerville, I. (2015). Software engineering 10th Edition. ISBN-10, 137035152, 18.
- [16] Darlow, S., & Wen, K. Y. (2016). Development testing of mobile health interventions for cancer patient self-management: a review. Health informatics journal, 22(3), 633-650.
- [17] Jamil, M. A., Arif, M., Abubakar, N. S. A., & Ahmad, A. (2016, November). Software testing techniques: A literature review. In 2016 6th international conference on information and communication technology for the Muslim world (ICT4M) (pp. 177-182). IEEE.
- [18] Amin, Maizan Mat, Jannifer Yep Ai Lan, Mokhairi Makhtar, and Abd Rasid Mamat. "A decision tree based recommender system for backpackers accommodations." *International Journal of Engineering & Technology* 7, no. 2.15 (2018): 45-48.
- [19] Skoković, P., & Rakić-Skoković, M. (2010). Requirements-based testing process in practice. International Journal of Industrial Engineering and Management (IJIEM), 1(4), 155-161.
- [20] Hamdy, Abeer, Osman Ibrahim, and Ahmed Hazem. "A web based framework for pre-release testing of mobile applications." In MATEC Web of Conferences, vol. 76, p. 04041. EDP Sciences, 2016.
- [21] Kapur, P. K., Shrivastava, A. K., & Singh, O. (2017). When to release and stop testing of a software. Journal of the Indian Society for Probability and Statistics, 18(1), 19-37.
- [22] Cem Kaner, J. D. (2013). An introduction to scenario testing. *Florida Institute of Technology, Melbourne*, 1-13.
- [23] Mathur, A. P. (2013). Foundations of software testing, 2/e. Pearson Education India.
- [24] Tosun, A., Ahmed, M., Turhan, B., & Juristo, N. (2018, May). On the effectiveness of unit tests in test-driven development. In Proceedings of the 2018 International Conference on Software and System Process (pp. 113-122).
- [25] Hsieh, C. H., & Wang, Z. (2022). A Two-stage Method of Synchronization Prediction Framework in TDD. Arabian Journal for Science and Engineering, 47(2), 2345-2357.
- [26] Huang, L., & Holcombe, M. (2009). Empirical investigation towards the effectiveness of Test First programming. Information and Software Technology, 51(1), 182-194.
- [27] Khaliq, Z., Farooq, S. U., & Khan, D. A. (2022). Artificial Intelligence in Software Testing: Impact, Problems, Challenges and Prospect. *arXiv preprint arXiv:2201.05371*.
- [28] Pecorelli, F., Catolino, G., Ferrucci, F., De Lucia, A., & Palomba, F. (2022). Software testing and Android applications: a large-scale empirical study. Empirical Software Engineering, 27(2), 1-41.
- [29] Aniche, M. (2022). *Effective Software Testing: A Developer's Guide*. Simon and Schuster.
- [30] Zozulya, M. M., Kravets, O. J., Atlasov, I. V., Aksenov, I. A., Bozhko, L. M., & Rahman, P. A. (2022). ALGORITHMIZATION OF THE SOFTWARE TESTING SYSTEM BASED ON FINITE AUTOMATA. International Journal on Information Technologies & Security, 14(1).