Global Scientific JOURNALS

GSJ: Volume 10, Issue 1, January 2022, Online: ISSN 2320-9186

www.globalscientificjournal.com

VERIFICATION AND VALIDATION OF SCHOOL MANAGEMENT INFORMATION SYSTEM (SMIS) USING BLACKBOX TESTING APPROACH (A CASE STUDY OF MAI IDRIS ALOOMA POLYTECHNIC)

[1]Hamisu Alhaji Ali, [2]Mohammed Shettima, [3]Ibrahim Abba

[123]Department of Computer science Mai Idris Alooma Polytechnic Geidam, Yobe State. Nigeria.

ABSTRACT

Software testing is a critical element of software quality assurance and represents the ultimate review of system requirement specification document, system design and programming aspects. The need to test a software is a necessity in any software development paradigm which is conducted in accordance and conformity to software quality assurance principles and characteristics set by scholars and governing bodies in the field of software engineering and Computing in general. The School Management Information System (SMIS) is a desktop/wed-based system designed using PHP/MySQL for front and backend accessibility coupled using CodeIgniter, CSS and Java Script. The SMIS is mostly administrative based which is designed with a detailed analysis and report of individual records and entities. Black box method of validation will be used to test key functional parts of the SMIS assessing inputs and outputs using test cases. The test plan and cases are tabulated with their expected results/ actual results which are displayed in a form of screen shot. Test log was provided which consist of an analysis of the success rates of each and every test case. The results show all the tested inputs/output of the SMIS are successful and the results are evaluated based of software system design guidelines and principles.

Keywords: MIS, School, Testing, Black-Box, Management, Information, System, Test Case.

INTRODUCTION

School Management System (SMIS) is software application for the education that can be used to managed student information and data. This system is designed for primary school to help administrators and teachers in the management of students. Students Management System consists of four modules including Teachers, Students and Parents Profile, Student Exam Results, Comment Reply and SMIS notification. Teachers can use this system to register information of student and managing parents and student profile. Teachers can view results of the students in graph format to know their performance. Besides that, teachers also can send the student results to parents through Short Message Service (SMIS). For the parents, they can login to system and view their children examination results and also graph for know their performance. This system also provides a communication platform for parents and teachers. Thus, by using this system can benefit to administrators, teachers and parents. Accordingly, testing the SMIS is a very crucial and essential aspect of any software development so as analyze and evaluate the strengths of the system by validation and verifying key functions of the system based on inputs and outputs.

TESTING

Testing is a technique for verification and validation of a system that is anticipated to ensuring that software adapts to its specification and meets its requirements. Testing includes checking and reviewing processes and system testing, it also involves executing the system with test cases that are derived from the specification. Currently the dominant technique used for verification is testing. And testing typically consumes an enormous proportion (sometimes as much as 50%) of the effort of developing a system. The objectives of testing a software is executing a program with the intention of discovering some errors, a good test case is one that has a high probability of finding errors and a successful test is one that uncovers an as-yet-undiscovered error.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing cannot show the absence of defects, it can only show that software defects are present.

"Tests are conducted and all results are evaluated, i.e. test results are compared with the expected results. When erroneous data is encountered, debugging commences. As test results are gathered and evaluated, a qualitative indication of software quality and reliability begins to surface. Two possible situations can occur here: if there are severe errors that require design modification are encountered regularly, software quality and reliability are suspect, and further tests are indicated. If, on the other hand, software functions appear to be working properly and the errors encountered are easily correctable, either:

Software quality and reliability are acceptable or Tests are inadequate to uncover severe errors."

RELATED LITERATURE

According to ANSI/IEEE-1059 standard, testing is the technique or method of examining a system to trace the variances among existing and necessary conditions which constitutes of errors or bugs and to evaluate the functions of the software through its codes based on input, output and procedures while manipulating the system

(Hutchison, 2003). As such, (Kumar and Trivedi 2012) outlined that, one key reason of testing a software is for verification, confirmation (validation) and fault finding in order to detect faults and mistakes and the intention of ruling those problems is to get them fixed. Also, (Beizer, 2009) explained that, testing is done solely to achieve a high-quality software or system. Testing is ready to spot presence of bugs, which cause software to malfunction, halt or crash. Nonetheless, testing is time consuming, expensive and also requires a lot of effort. (Sharma, Sabharwal and Sibal 2013). Moreover, Nidhira and Dondeti, 2012 lamented that, software testing is measured to be labor intensive and expensive, which accounts for > 50 % of the total cost of software development. Software testing is one the most vital stages of the system development life cycle (SDLC) its boost developer confidence, as such some stages of SDLC require testing at every stage in other limit the occurrence of bugs in the system though it will be a time-consuming effort.

Software testing is an action which is envisioned for estimating features or an ability of a software and guarantee that it meets the necessary effect. Software developed will never be bug-free without being tested and accepted. These warrants for the decision to assess and choose a technique for testing concealed bugs at different stages. Testing involves techniques for spotting hidden bugs, errors in coding or any unused line of code. Presently, there are tons of techniques being used for testing faults in software; among them, most proficient techniques for testing software are black-box testing, white-box testing, and gray-box testing. As suggested by (Kumar and Trivedi 2012), it is a tedious and time-consuming activity yet compulsory. As such, it is essential to choose an appropriate technique for a testing activity after a thorough evaluation and justification of the techniques to be implemented. Many of the empirical study had performed to estimate software testing techniques in terms of accuracy and efficiency (Tyagi and Malhotra, 2014). As such, software testing is the process whose aim is to find errors, to evaluate the capabilities and attributes of the software unit and to check whether it satisfies the requirements of the users or not. In this process, system components and system requirements are evaluated and exercised manually or by automated tools to check whether specified requirements are satisfied and difference between expected and actual results are analyzed (Hooda and Rajinda, 2015).

TESTING TECHNIQUES

There are several techniques for testing but the two key techniques will be evaluated which comprises of white-box and black-box testing.

BLACK BOX TESTING

While performing black box testing, a tester must know the system design and will not have access to the source code. Black box testing also referred to as functional or behavioral testing, it is used to test the functionalities of the system focusing on its functional requirements. It derives some inputs that will exercise the functional requirements of the system. This type of testing is termed black box testing because no knowledge of the workings of the program is used as part of the testing, we only consider inputs and outputs (Shettima, Sani, Abubakar, Tijjani, Mohammed, 2018). In black box testing, the key items that require testing includes errors in data structures, software performance errors, code initialization and termination errors, missing function, incorrect procedures, interface errors and also database accessibility errors.

Black box testing involves creation of test cases for inputs, outputs and processes with the software developed. It works hand in hand with the system specifications during the analysis stage where all the input and output procedures of the system are specified. One sole reason for deciding the test cases are requirements and specifications of the module or program (Dhawan and Heena, 2018). The system is treated as a black box so explicit usage of knowledge of internal code or structure is not used. So, it can be said that black box testing lay emphasis on functional requirements of the software. The test cases are designed to check for the errors which are related to missing or incorrect functions, data structures, interfaces, initialization and termination errors. (Pressman, 5th Ed.).

By applying black box testing techniques, a set of test cases can be derived to satisfy the following criteria:

- Test cases that reduce, by a count that is greater than 1, the number of additional test cases that must be designed to achieve reasonable testing.
- Tests cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specific test at hand.

The procedure for black box testing is shown in the following diagram.

- Test data and desired results are derived from the specification. The test data is fed to the implementation, and the outputs from execution are compared with the desired results. If they match, testing proceeds. If not, the implementation must be debugged.

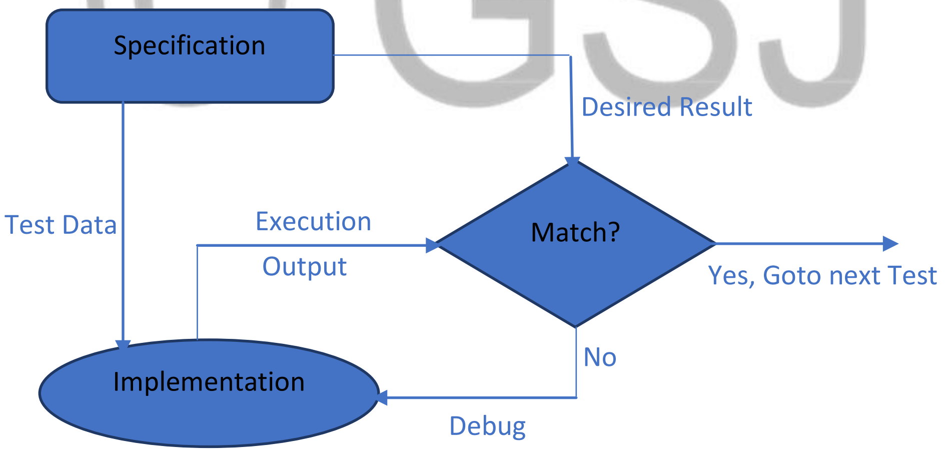


Fig. 1: Black Box Testing Approach

WHITE BOX TESTING

White box testing also referred to as **structural or glass box testing** is used to test the structure of a program, it is a technique for test case design that uses the control structure of the procedural design to originate test cases, it ensures that:

1. All independent paths within a process should be exercised at least once
2. Exercise all logical decisions on their true and false sides

3. Execute all loops at their boundaries and within their operational bounds
4. Exercise internal data structures to ensure their validity.

"Tests are derived from an examination of the source code for the modules of the program. These are fed as input to the implementation, and the execution traces are used to determine if there is sufficient coverage of the program source code."

White box, or structural testing is derived directly from the implementation of a module. Hence, it is able to test all the implemented code, but it cannot test that any functionality is missing from the implementation. In this sense, it is complementary to black box testing (Grover, 2016). The process of white box testing is shown in the following diagram.

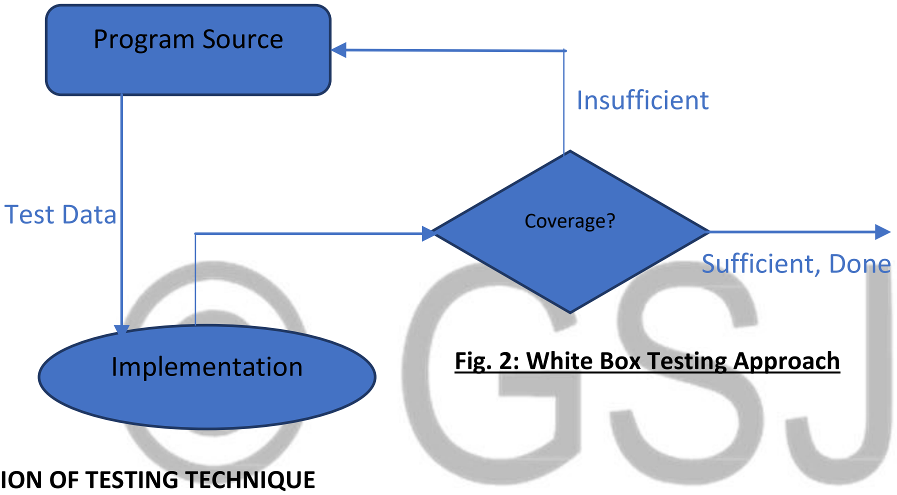


Fig. 2: White Box Testing Approach

JUSTIFICATION OF TESTING TECHNIQUE

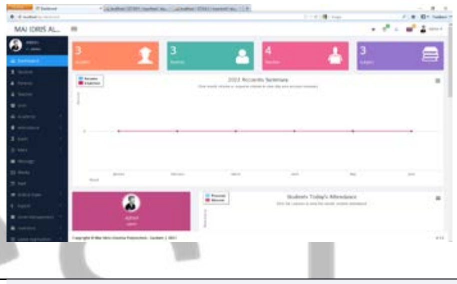
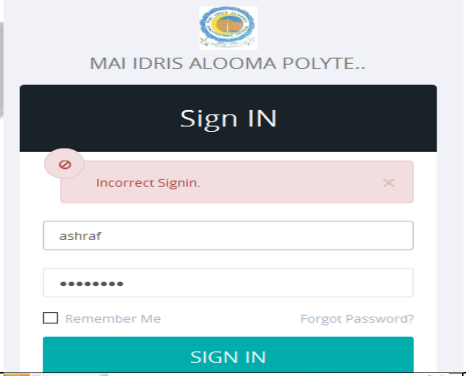
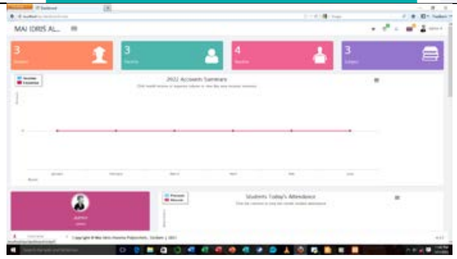
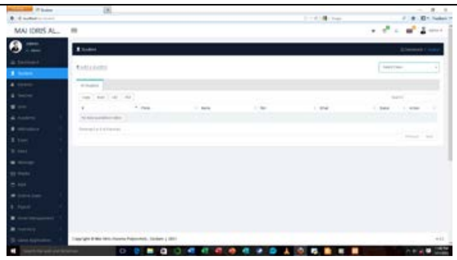
The technique chosen for testing the SMIS artefact is the black box testing because:

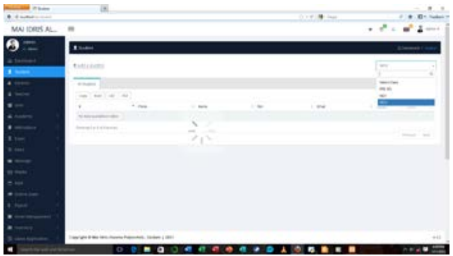
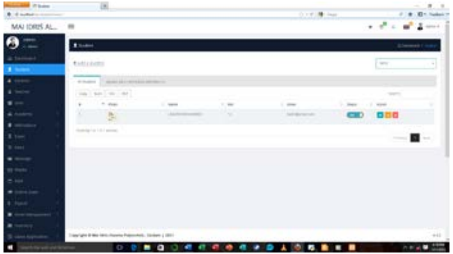
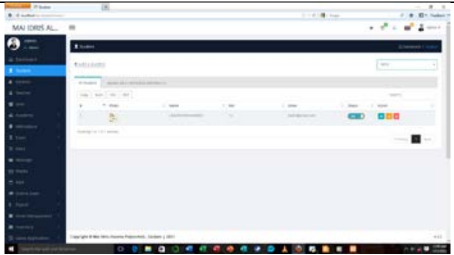
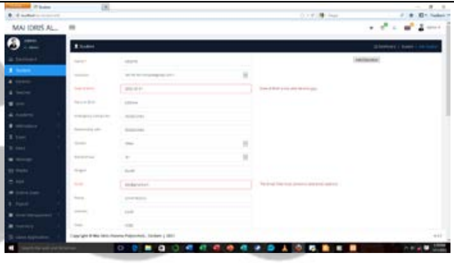
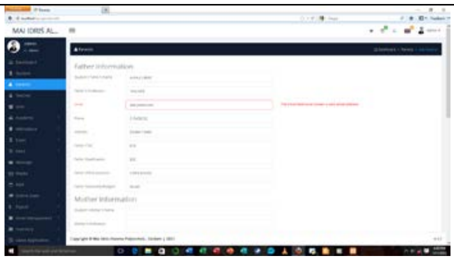
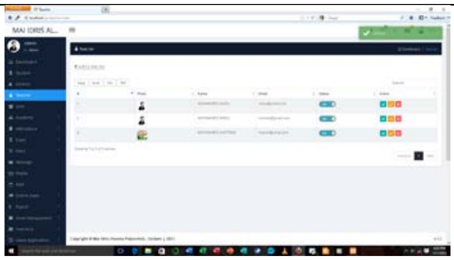
- The software developed has a collaborative interface and all the bugs detected by the compiler within the source program are debugged and thorough compilation process has been conducted. As such, **White Box Testing is Irrelevant.**
- Black box testing's main efforts are to do the following key activities:
    - Find errors in an incorrect or missing function
    - Interface errors
    - Errors in data structures or external database access
    - Behaviour or performance errors
    - Initialization and termination errors.

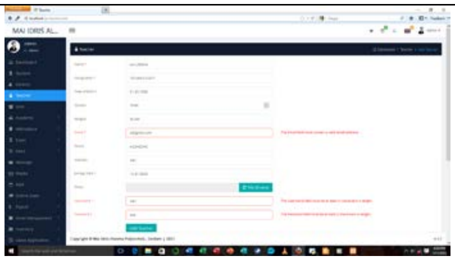
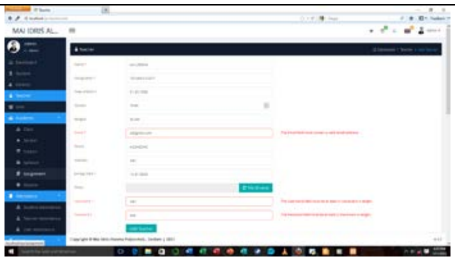
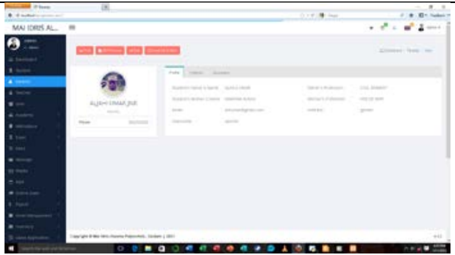
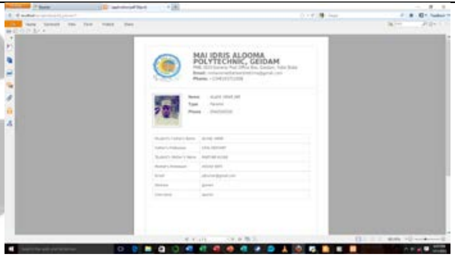
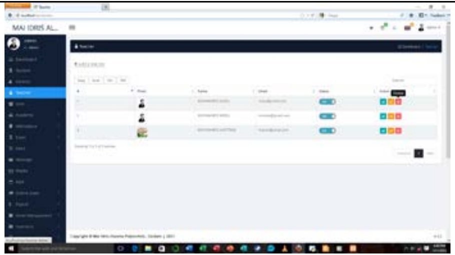
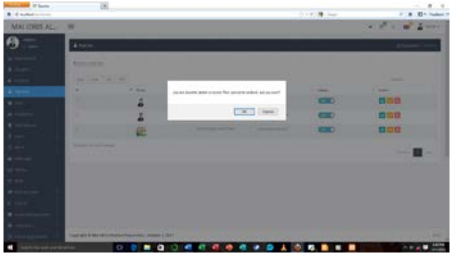
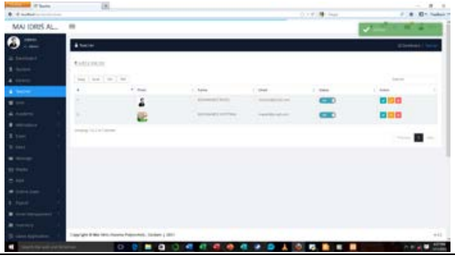
White Box testing is conducted immediately after coding which is a bit earlier than Black Box testing. Moreover, Black Box testing is conducted after complete debugging process. As such, Black Box can be done by someone with little or no experience in programming. Lastly, test cases can be designed as soon as the specifications are complete and tests are done from a user's point of view. As concluded by (Jat and Sharma, 2017) Black-Box testing takes least time and effort to test a test-case as compare to other techniques.

TESTING PROCEDURE

The process of testing the key functions of the SMIS comprises of the test plan which comprises of test cases and test log to determine the success of the testing conducted. It includes the results expected, the actual result, test data and conclusions of each functional testing. The major functions of the system will be tested and evaluated in this section. The testing will be run for both the wrong input and correct one. The expected results for the correct input are to allow the required transaction to held, and disallow the transaction when wrong input is type or mandatory fields are skipped. Below is the testing case for the functional requirement.

S/N	Function (Test Case)	Input (Test Data)	Output (Expected result)	Screen short (Actual Result)	(Test log)
1.	Login using valid password	Type username Type password Press the Sign in button	Login successful, proceed to Dashboard		Successful
2.	Login using invalid password	Wrong email Wrong password	Incorrect Signin	 MAI IDRIS ALOOMA POLYTE.. Sign IN Incorrect Signin. ashraf •••••••• □ Remember Me Forgot Password? SIGN IN	Successful
3.	Hiding Task Pane	Click on task pane button on the left side of Dashboard	Task pane Hidden		Successful
4.	Student Module Link	Click on Student in the Task Pane	Student Page Opens		Successful

5.	Student List Based on Class	Click on the Select Class Arrow to Display List of Class	List of Classes is shown	 	Successful
6.	Student Class List	Click Class you want to open	Class Opened with List of Students		Successful
7.	Validation of add student form	Input Incorrect Student Details to Check if field are Validated	Forms Shows Error message for Incorrect input		Successful
8.	Validation of Add Parents Form	Input Incorrect Parents Details to Check if field are Validated	Form Shows error message Incorrect Input		Successful
9.	Validation of Add Teacher Form	Input CORRECT Teacher Details	Record Added Successfully		Successful

10.	Validation of Add Teacher Form	Input Incorrect Teachers Details to Check if field are Validated	Form Shows error message Incorrect Input	 MAI IDRIS AL...	Successful
11.	All links on The Dashboard Task pane	Click on All the Links in the Dashboard pane to see if they work	All Links are working Successfully	 MAI IDRIS AL...	Successful
12.	View Parent Details	Click on View Parent Link/ icon	Parent Details Viewed Successfully	 MAI IDRIS AL...	Successful
13.	Convert Parents Details to PDF	Click on PDF on the Parent information	Parent Details is Converted for PDF and ready for printing	 MAI IDRIS ALOOMA POLYTECHNIC, GEIDAM	Successful
14.	To delete Teachers, Record	Click on Delete Icon/ Button	Teachers Record Deleted	 MAI IDRIS AL...  MAI IDRIS AL...  MAI IDRIS AL...	Successful

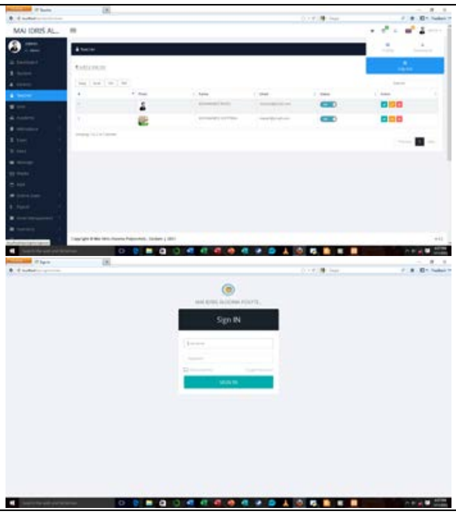
15.	To Log Out of the System	Click on Log Out on the Admin Log Out Bar	Logged Out Successfully		Successful

Table 1: Black Box Testing Information for SMIS with test case, result and log.

DISCUSSION

Black box functional testing is conducted on some random key function of the SMIS in order to evaluate the performances and user interaction of the software artefact.

It is learnt that from the black box testing that

- The system lives up to its promise and delivered the entire promised requirement and design. And only one error detected in the functional requirement which needed to be debugged.
- Since some aspects of White Box testing has already been conducted during the coding stages of the development life cycle and also, corrections being applied, the testing was very successful, with no errors being found. The website anticipated all potential mistakes, providing ultimate validation and verification process. Moreover, as the saying goes, “No software is perfect” the SMIS working as expected even though there are some difficulties or errors encountered and will continue to occur, it will not hinder any bottlenecks and the system will function as it is intended to.

CONCLUSION

Software testing can be viewed as a process of examining and analyzing the software unit in order to find errors and to check the differences between expected and actual behavior. Two testing approaches and techniques have been proposed in the literature. Both the white box and black box software testing techniques have their own merits and demerits and are complementary approaches to each other. It cannot be said that one is alternative for another as both forms of testing techniques uncover different forms of errors. The need to test a software is a necessity in any software development paradigm which is conducted in accordance and conformity to software quality assurance principles and characteristics set by scholars and governing bodies in the field of software engineering and Computing in general. The School Management Information System (SMIS) is a desktop/wed-based system designed using PHP/MySQL for front and backend accessibility coupled using CodeIgniter, CSS and Java Script. The SMIS is mostly administrative based which is designed with a detailed analysis and report of individual records and entities. The SMIS functional requirements has been tested to make sure appropriate input and output of data is being is been processed by the system for verification and validation purpose. All the functions tested has been documented and the results yielded are all positive. Key functions were tested because as it implies, black box testing sometimes seems to be time consuming because of the capacity of source codes written and the usage or implementation of several functions in a single page can cause delay and make the whole testing process hectic.

RECOMMENDATION

Prior to the testing conducted, there is a need to evaluate the whole system in order to assess its functionalities at both the system and user point of view. Since any software developments is all about meeting users' needs and expectation, vividly focusing on usability guidelines and issues. As such, "**Think aloud Usability Testing/ Evaluation Approach"** can be used to test and evaluate user experience, to assess how the user will interact with the SMIS taking into consideration all User Centered Design Principles and Guidelines set by Human Computer Interaction Design (HCID) experts.

Acknowledgment

The authors wish to thank their able Rector, Chief Librarian and above all the reviewer. This work was supported in part by a grant from the Tertiary Education Trust Fund (TETfund) Nigeria.

REFERENCES

[1]. A. Anitha, "A Brief Overview of Software Testing Techniques and Metrics", International Journal of Advanced Research in Computer and Communication Engineering, Volume-2, Issue-12, pp.4655-4659, 2013.

[2]. B. Beizer, "Software Testing Techniques" ,DreamTech Publisher , Edition-II ,India, pp. 13 247,2009

[3]. C. Sharma, S. Sabharwal, R. Sibal , "A Surveyon Software Testing Techniques using Genetic Algorithm", International Journal of Computer Science,Voume-10,Issue-1,pp.381-393,2013.

[4]. Gaur J.; Goyal A.; Choudary T. and Sabitha S. (2016), "A Walk Through of Software Testing Techniques", *International Conference on System Modeling and Advancement in Research Trends*, November, pp-103-08.

[5]. Grover, D. K. (2016), "Comparison Study of Software Testing Methods and Levels-A Review", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 4, No. 8, August, pp.15011-16.

[6]. Hooda, I. and Rajinder, C.S. (2015), "Software Testing Process, Testing Types and Techniques", *International Journal of Computer Applications,* Vol. 111 No 13, Feb, pp. 10-14.

[7]. M.L. hutcheson, "Software Testing Fundamentals Methods and Metrics", Wiley Publication Inc. ,USA, pp.3-335,2003.

[8]. M. Tyagi, S. Malhotra, "A Review of Empirical Evaluation of Software Testing Techniques with Subjects", International Journal of Advanced Research in Computer Engineering & Technology, Volume-3, Issue 2,pp.519-523, 2014.

[9]. Roger Pressman,"Software Engineering, A Practitioner's Approach", Fifth Edition, ISBN-0-07-365578-3.

[10]. Shettima, M., Sani, A, J., Mohammed, A., Tijjani, A, U., Mohammed, I. " Online Student Room Availability System", Global Scientific Journal, GSJ: Volume 6, Issue 12, December 2018 ISSN 2320-9186.

[11]. S. Kumar, H. Trivedi, "Software Testing Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume-2, pp. 433-438, Oct 2012

[12]. W3schools (2014) *Educate yourself: Beginners ang experts*. Available at: http://www.w3schools.com/default.asp (Accessed: 15, June 2021).

[13]. Wampserver (2014) *Wampserver: A windows web development environment.* Available at: http://www.wampserver.com/en/ (Accessed: 01, August 2021).