



Republic of the Philippines
City of Taguig
Gen. Santos Avenue, Central Bicutan, Taguig City



Taguig City University

Web Testing Startup Kit – Automated Test Case Testing Management Tool

A Thesis Proposal Presented to the Faculty of the Computer Science

Taguig City University, General Santos Avenue

Central Bicutan, Taguig City

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science

By:

Rasdi I. Kasim

Mark Jervin G. Roldan

Hiedy G. Uhuad

Mark V. Valles

July 2019



Approval Sheet

This thesis entitled “**Web Testing Startup Kit – Automated Test Case Testing Management Tool**” prepared and submitted by Rasdi I. Kasim, Mark Jervin Roldan, Hiedy G. Uhuad and Mark V. Valles. In partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science**, examined and recommended for oral examination.

Prof. Reynaldo G. Alvez, MSIT

Thesis Adviser

Approved by the PANEL OF EXAMINERS ON ORAL EXAMINATION.

Prof. Wilber B. Sabado, MSIT

Faculty, Computer Science
Department

Mr. Charlemagne P. Valdez

Management Specialist,
PCIEERD - DOST

Date: _____

Date: _____

Accepted in partial fulfillment of the requirements for the degree Bachelor of
Science in Computer Science.

Prof. Reynaldo G. Alvez, MSIT

Dean, CICT

Date: _____



Certification of Originality

This is to certify that the research work presented in this thesis entitled **“Web Testing Startup Kit – Automated Test Case Testing Management Tool”** for the degree of Bachelor of Science at Taguig City University embodies the result of originality and scholarly work carried out by the undersigned. This thesis does not contain words or ideas taken from published sources or written works that have been accepted as basis for the award of a degree from any higher education institution, except where proper referencing and acknowledgements were made.

Rasdi I. Kasim

Mark Jervin G. Roldan

Hiedy G. Uhuad

Mark V. Valles

Researchers

July 2019



Acknowledgement

The researchers would like to express their deepest appreciation to their adviser, Professor Reynaldo Alvez for his help and support. He continually and convincingly conveyed a spirit of adventure with regards to the researcher's study. Without his guidance and persistent help, the thesis would have not been possible.

The researchers would like to extend their sincere thanks to all who helped them in making this thesis a success. Lastly, the researchers would like to thank our families for being understanding during the process of forming this thesis. Our deepest gratitude is for them.



Abstract

Web system applications are dynamic and interactive as compared to traditional applications. Therefore, traditional testing techniques are not sufficient for web system application testing. Basically, software testing is divided into two types which is the automated and manual testing. Manual testing is a process of finding out the defects or bugs in a program. However, this traditional way consumes a lot of time and test results are not accurate.

Effective software testing can save money and effort by catching problems before they make it very far through the software development process. It is known that the longer a defect remains undetected, the more expensive it is to fix. Software testing is an integral part of software development process. Software testing is analyzing a system or a component by providing defined inputs and comparing them with the desired outputs to check the discrepancies between the desired and actual outputs and correct them.

Automating functional software testing has the potential to reduce the amount of time spent on testing and open up opportunities to reuse test scripts in other areas of software testing, such as load testing, regression testing and performance testing. The first problem posed in this research, is to reduce the amount of time spent performing software testing. However, there is a need that



when performing a standard software testing, testers must learn on how to write test cases.

When developing a web system software, the ultimate goal of the tester or developer is to ensure that the application is tested often and thoroughly. Nowadays with the used of automated testing tools, IT companies became more confident to deliver their software on time when it comes to project deadlines. The development for the proposed system was done from different phases such as requirement analysis, UI & functional design, evaluation, system verification and lastly the software maintenance.

The researchers conducted a survey to know on how important the study is and the general result came out with a highly acceptable interpretation. The general ratings from the survey thought us that Web Testing Startup Kit – Automated Test Case Testing Tool Management was appreciated by the beloved respondents of the researchers. The questions from the recent survey where categorized from its functionality, efficiency, usability, portability and the security.



Table of Contents

Title Page.....	i
Approval Sheet.....	ii
Certification of Originality.....	iii
Acknowledgement.....	iv
Abstract.....	v
Table of Contents.....	vii

CHAPTER I: Introduction

Project Context.....	2
Purpose and Description.....	4
Objectives.....	5
Scope and Limitation.....	6

CHAPTER II: Review of Related Literature and Studies

Foreign Literature.....	8
Local Literature.....	14
Foreign Study.....	19
Local Study.....	24

CHAPTER III: Technical Background

System Architecture.....	29
Software.....	30
Hardware.....	34



CHAPTER IV: Design and Methodology

Developmental Research.....	36
Population, Sample Size, Sampling Techniques.....	37
Data Gathering Procedures.....	39
Pre-survey Result.....	40
Waterfall Model.....	42
Sample Test Parameters.....	44
Flowcharts.....	47
Data Flow Diagrams.....	55

CHAPTER V: Results and Discussion

Post Survey Result.....	63
Summary.....	66
Conclusion.....	67
Recommendation.....	68
References.....	69

Appendices.....70

Appendix A: User Manual.....	71
Appendix B: System Codes.....	83
Appendix C: Pictures During Survey.....	135
Appendix D: Grammarian Certificate.....	139
Appendix E: Plagiarism Screenshot.....	140
Appendix F: Pre-survey Questionnaire.....	141



List of Figures

Figure 1 – System Architecture.....	29
Figure 2 – Notepad++.....	30
Figure 3 – MySQL.....	31
Figure 4 - XAMPP.....	31
Figure 5 - Google Chrome & Mozilla Firefox.....	32
Figure 6 – Bootstrap & jQuery.....	32
Figure 7 – JDK.....	33
Figure 8 – Desktop Computer & Laptop.....	34
Figure 9 – Server Unit.....	35
Figure 10 - Waterfall Software Development Cycle.....	42
Figure 11 – Maintenance Plan Chart.....	46
Figure 12 – Login Flowchart.....	47
Figure 13 – Forgot Password Flowchart.....	48
Figure 14 – Add New Test Project Flowchart.....	49
Figure 15 – Edit Test Project Flowchart.....	50
Figure 16 – Delete Test Project Flowchart.....	51
Figure 17 – Add Test Script Flowchart	52
Figure 18 – Start Test Script Flowchart.....	53
Figure 19 – Add User Account Flowchart.....	54
Figure 20 – Data Flow Diagram Level 0.....	55
Figure 21 – Data Flow Diagram Level 1.....	56
Figure 22 – Registration Data Flow Diagram Level 2.....	58
Figure 23 – Login Data Flow Diagram Level 2.....	59
Figure 24 – Add Test Project Data Flow Diagram Level 2.....	60



Figure 25 - View Test Logs Data Flow Diagram Level 2.....61

Figure 26 - Add Test Script Data Flow Diagram Level 2.....62

List of Tables

Table 1 – Population of the Respondents.....38

Table 2 – Lickert Scale.....39

Table 3 – Sample Test Parameters.....44

Table 4 – Post Survey Result.....63



CHAPTER I

Introduction

The introduction of computers has given business an interesting upper hand by supplying companies with endlessly amounts of information. In today's competitive business environment, speed is the key for all the business processes, there is a need for many automated systems that are capable of replacing or reducing human effort in their daily activities and jobs. An automated software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects, to ensure that the product is defect free in order to produce the quality product. Manual testing is a process of finding out the defects or bugs in a software program. In this way, tester will simulate the role of end user to go through all the features throughout the application to make sure that they are working as they are supposed to work. However, manual testing consumes more time and low accuracy result, therefore manual testing is not reliable. Nowadays, IT firms encounters many difficulties in terms of manual testing, one of the problems they been suffering is time-consuming due to running test cases as everything is done manually. As researcher proposed a study that will solve the current issue of manual system application testing entitled the Web Testing Startup Kit – Automated Test Case Testing Management Tool.



Project Context

Whether the company is small or a medium to large-scale business, their product must be quality and defective less. Nowadays, every software development group tests its products, yet delivered software always has defects. Test engineers strive to catch them before the product is released but they always creep in and they often reappear, even with the best manual testing processes. When a software tester manually checks a system, he cannot avoid to make a mistake, especially when an application contains hundreds or more of lines of code. Manual software testing is performed by a human sitting in front of a computer carefully going through application screens, trying various usage and input combinations, comparing the results to the expected behavior and recording their observations. Manual tests are repeated often during development cycles for source code changes and other situations like multiple operating environments and hardware configurations. Automation helps the system developers avoid these application testing mistakes and executes checks in a faster timeframe than a person. Test Automation software is the best way to increase the effectiveness, efficiency and coverage of your software testing. Businesses must make sure that embracing of test automation does not equal or exceed the benefits that will come from it. Automating software testing is widely accepted as a way for businesses to reduce the use of resources, save time and eliminate scheduling issues. The said disadvantage's in manual system testing is a proof that Web Testing Startup Kit –



Automated Test Case Testing Management Tool is one of the solutions to help the developers out there.



Purpose and Description

The main purpose of Web Testing Startup Kit – Automated Test Case Testing Management Tool is to provide an open source web system application testing for the developers who manually evaluates their developed system which encounters many difficulties. Today one of the top software tester tools is Ranorex Studio. Ranorex Studio is one of the most comprehensive automation tools on the market. However, this software tester tool is too expensive and supports only C# and VB.NET programming language that can't be launched on macOS. Unlike other expensive software testers like Ranorex Studio, Web Testing Startup Kit - Automated Test Case Testing Management Tool is a multi-platform web system application tester that provides faster testing result. Manual system test takes much time in order to ensure that all of the functions in the developed system are working. Automated testing is a process to validate the system application functions appropriately and meets requirements before it is released into production. This aim of this study is not to add new knowledge, but to lessen and prevent the impact of an ineffective manual system application testing. It is aimed to explore and present the facilitations that are offered in the daily work tasks of individuals. It is tried to present new opportunities to the IT business organization sectors by interpreting the experiences of developers that have taken advantage of such system.



Objectives

This study aims to develop a user-friendly web system application tester, for the programmers and non-programmers who are experiencing difficulties when they're testing their developed web system application. With this system application testing tool, test cases are automatically recorded in your computer screen and displays real-time form by form test results after the tester enters the data in a web application screen. The application tool will also have other features such as user information, user level of access, password retrieval and automatically log out when it's standby for 30 minutes. Generally, the main objective of this study is to develop an open-source web system application tester that will uncover bugs and errors as many as possible in a certain system application product.



Scope and Limitation

The project focused solely in developing a web system application tester named Web Testing Startup Kit - Automated Test Case Testing Management Tool. This system provides the following features and limitation.

Scope

1. The system will count and notify the user about the remaining scripts that has not been tested in a certain test project.
2. The system will open a browser to display real-time testing activity depending on the web driver selected, either Mozilla or Google Chrome.
3. The developed system provides reports such as test cases testing logs, project reports containing usability rate, test progress, script counts and remarks.
4. The system is capable to run test cases via network.
5. The system disallows multiple log in with same account credentials on another computer.
6. The system can set the repetition of execution in a certain test script.
7. The system can set the time interval of test script execution.



Limitation

1. The developed system can only test web-based applications made from HTML, HTML5 with backend PHP, Python and ASP.NET.
2. The developed system requires specific browsers like Google Chrome version 72.0.3626.121 onwards and Mozilla Firefox with atleast 65.0.2 version.
3. The system is not applicable on mobile devices.
4. The system is not capable to identify the expected result of the tester whether the output is right or wrong.



CHAPTER II

Foreign Literature

According to Rashmi (2016) in her study about A Keyword Driven Framework for Testing Web Applications. Testing is an integral part of the software development. The goal of software testing is to find faults from developed software and to make sure they get fixed. It is important to find the faults as early as possible because fixing them is more expensive in the later phases of the development. The purpose of testing is also to provide information about the current state of the developed software from the quality perspective. On a high level, software testing can be divided into dynamic and static testing. The division to these two categories can be done based on whether the software is executed or not. Static testing means testing without executing the code. This can be done with different kinds of reviews. Reviewed items can be documents or code. Other static testing methods are static code analysis methods for example syntax correctness and code complexity analysis. The purpose of functional testing is to verify that software corresponds to the requirements defined for the system. The focus on functional testing is to enter inputs to the system under test and verify the proper output and state. The non-functional testing means testing quality aspects of software. Benefits of non-functional testing are performance, security, usability, portability, reliability, and memory management testing.



Based on the study of Rashmi, the test automation is considered to be integral part of the software development. Software testing is the activity to find or verify errors and bugs in a software. As developers of Web Testing Startup Kit discovers that software testing can be divided into dynamic and static testing. Static test is done without executing the code, while in dynamic it is executed with the source code.

According to Anika (2016) in her study the Investigation of the use Test automation in Software Quality Assurance. The focus of this thesis has been about automated testing in practice. Using testing tools is quite common phenomenon in software companies. A questionnaire about the use of testing tools in software quality assurance to investigate Norwegian software companies has been made. The questionnaire has been sent out to a number of Norwegian IT/software and non-IT/software companies to get feedback on their use of testing tools for projects and product development. Four interviews have been conducted to find out more about the real-life practice of using testing tools in different companies. Quantitative Analysis has been used on the responses from the questionnaire and each response from the questionnaire have been statistically analysed. The statistical analysis of the responses shows the percentage of how many people are using testing tools and why, who are still solely dependent on manual testing and why and how the software quality assurance testing's is done. The main objective is to find out facts related to the research questions; the less practice of



using testing tools in real life despite of its popularity in literature reviews and how the companies are following agile method and debugging at the same time.

Based on study of Anika about the investigation of the use of test automation in software quality assurance helps us to know about the test automation what is. Testing is used to ensure software quality in verification and validation and reliability estimation of software products. It is the most widely used approach to ensure software quality since software quality assurance occupies most concern in the software industry.

According to Jansing (2015) in his study of Enhancing the Effectiveness of Software Test Automation. Effective software testing can save money and effort by catching problems before they make it very far through the software development process. It is known that the longer a defect remains undetected, the more expensive it is to fix. Testing is, therefore a critical part of the development process. It can also be expensive and labor intensive, particularly when done by hand. It is estimated that the total effort testing software consumes at least half of a project's overall labor. Automation can make much of the testing an organization does more accurate and cheaper than merely putting several people in a room and having them run tests from a paper script. It also frees the testing staff to do more specific and in-depth testing than would otherwise be possible. This paper focuses mainly on software test automation techniques and how automation can enhance the efficiency of a software team as well as the quality of the final product.



Based on the study of Jansing, it shows how expensive and hassle when companies needs to test their developed system before deploying. This shows that our study will be useful in the future for the IT firms before they will deploy their developed system.

According to Kanewala & Bieman (2014) in their study about Testing Scientific Software at the Colorado State University. Context: Scientific software plays an important role in critical decision making, for example making weather predictions based on climate models, and computation of evidence for research publications. Recently, scientists have had to retract publications due to errors caused by software faults. Systematic testing can identify such faults in code. This study aims to identify specific challenges, proposed solutions, and unsolved problems faced when testing scientific software. (1) testing challenges that occur due to characteristics of scientific software such as oracle problems and (2) testing challenges that occur due to cultural differences between scientists and the software engineering community such as viewing the code and the model that it implements as inseparable entities. In addition, the researcher identified the methods to overcome these challenges and their limitations. Finally, the researcher describes unsolved challenges and how software engineering researches and practitioners can help to overcome them. Scientific software presents special challenges for testing. Specifically, cultural differences between scientist developers and software engineers, along with the characteristics of the



scientific software make testing more difficult. Existing techniques such as code clone detection can help to improve the testing process. Software engineers should consider special challenges posed by scientific software such as oracle problems when developing testing techniques.

Based on the study of Kanewala & Bieman, automation on software testing is a critical method and need to conduct a survey. the researchers conducted a systematic literature survey to identify and analyze relevant literature. We identified 9 studies that provided relevant information about testing scientific software.

According to Garcia (2014) in his study on Software Product Quality at Switzerland. Software products quality is strongly influenced by the quality of the process that generated them; particularly, the testing process contributes to product quality and represents a significant effort in software development projects. In this context, this study aims to find which test process models has been defined, adapted or extended in software industry from 1990 to the current date. For this purpose, a systematic literature review has been performed according to relevant guidelines. This study has identified 23 test process models, many of them adapted or extended from TMMi and TPI, which have different architectures and the new ISO/IEC 29119 with an architectural approach aligned to other ISO/IEC software process models.

Based on the study of Garcia about the Software Product Quality gives us the idea and motivation to create a Web Tester Startup Kit to help the developers



Taguig City University



13

to check the bugs and error and contributes for the product quality and the participation of International Organization for Standardization (ISO) and International Electronical Commission.



Local Literature

According to Salazar (2015) in his study about the Software Quality Testing. The testing of software is important to determine its quality. Testing typically consumes 40-50% of development efforts. It is an important part of the software development lifecycle. Testing is a process of analysing software to find the differences between existing and required conditions i.e. defect sand to evaluate features of that software. Testing is evaluation of a system or its component(s) to find whether it satisfies the specified requirements of system or not. Testing is executing a system identify gaps, errors according to the actual requirements. There are lots of testing techniques and tools are available to complete the task. In this paper, various testing techniques and tools are described.

Based on the study of Salazar about the Software Quality Testing, the importance of a testing tool in the field and it's consumes 40%-50% of the development efforts. The researchers of Web Testing Start-Up Kit got a boost and the determination for the developers who's doing a testing and checking for manual that consumes their lot of time.

According to Martin (2015) in his study about the Test Automation Framework Design and Implementation. Test automation framework design and implementation is an inexact science begging for a reusable set of standards that can only be derived from a growing body of precedence; this book helps to establish such precedence. Much like predecessor court cases are cited to support



subsequent legal decisions in a judicial system, the diverse case studies in this book may be used for making contemporary decisions regarding engagement in, support of, and educating others on software test automation framework design and implementation.

Based on the study of Martin, he stated that test automation framework design and implementation is an inexact science begging for a reusable set of standards that can only be derived from a growing body of precedence, this study was truly useful in terms of decision making.

According to Lopez (2016) in his study about Cloud Computing Testing Service. Cloud computing is causing a paradigm shift in the provision and use of computing services; away from the traditional desktop form to online services. This implies that the manner in which these computing services are tested should also change. This paper discusses the research issues that cloud computing imposes on software testing. These issues were gathered from interviews with industry practitioners from eleven software organizations. The interviews were analysed using qualitative grounded theory method. Findings of the study were compared with existing literature. The research issues were categorized according to application, management, legal and financial issues. The issues are discussed with the intention of soliciting academic research on software testing in the cloud. By addressing these issues, researchers can offer reliable recommendation for testing vendors and customers.



Based on the study of Lopez, he said that cloud computing is causing a paradigm shift in the provision and use of computing services. Nowadays, system applications are being used by many companies in web-based. The users of this web-based systems were encountering bugs and errors due to testing conflicts.

According to Dela Cruz (2015) in his study about Automated Software Testing. In the recent years the complexity of software products has increased tremendously. Many software companies strive to achieve higher quality software with shorter delivery time. Today software testing is used by the companies to increase the quality of their products and to ensure that the client requirements are met. In the industrial practice, verification and validation activities are often performed with pricey, laborious and error-prone manual test practices. Developers are usually under the pressure to release faster, thus, quality often becomes hindered. A possible solution to this problem is test automation, but the available tools and techniques bear a lack of generic applicability. One way to mitigate these problems is to unify various test methods and practices. This thesis presents a design science research, which explores the challenges of unification. Additionally, it tries to find a solution to the problems that these challenges present by developing a system for semi-automated testing (SeAT). The thesis also analyzes the effects of SeAT on the testing processes at a case company.

Based on the study of Dela Cruz about the Automated Software Testing is that the recent years the complexity of software products are having a difficulty in



checking or finding the bugs and error. But today's time some of the companies are using a testing tool for it. So, we the researcher are doing a testing tool for everyone because the Web Start-Up Kit will be an open-source software system. And everyone can easily check their developed software.

According to Garcia (2014) in his study about Test Automation Software System. Test automation can decrease release cycle time for software systems compared to manual test execution. Manual test execution is also considered inefficient and error-prone. However, few companies have gotten far within the field of test automation. This thesis investigates how testing and test automation is conducted in a test consulting setting. It has been recognized that low test process maturity is common in customer projects and this has led to equally low system testability and stability. The study started with a literature survey which summarized the current state within the field of automated testing. This was followed by a consulting case study. In the case study it was investigated how the identified test process maturity problems affect the test consulting services. The Consulting Automated Testing Strategy (CATS) been developed to meet the current identified challenges in the domain. Customer guidelines which aim to increase the test process maturity in the customer organization have also been developed as a support to the strategy. Furthermore, the study has included both industrial and academic validation which has been conducted through interviews with consultant practitioners and researchers.



Based on the study of Garcia about the Test automation Software System is just like of Dela Cruz that are having more time consuming in term of finding the bugs or error for manually checking of their developed system. And sometimes the manual checking is not purified and still they are encountering bugs and error. So, Garcia made an investigation about testing and test automation.



Foreign Study

According to Bertolino (2012) in his study about Software Testing Research: Achievements, Challenges, Dreams. The paper attempts to provide a comprehensive view of the field of software testing. The objective is to put all the relevant issues into a unified context, although admittedly the overview is biased towards my own research and expertise. In view of the vastness of the field, for each topic problems and approaches are only briefly tackled, with appropriate references provided to dive into them. I do not mean to give her a complete survey of software testing. Rather I intend to show how an unwieldy mix of theoretical and technical problems challenge software testers, and that a large gap exists between the state of the art and of the practice.

Based on the study of Bertolino, he said that software testers are experiencing technical problems when they test different systems. Software testing is considered as a critical job nowadays, once they deploy developed system, they're not totally sure that the system is bug-free.

According to Sriraman (2014) in his study about Software Testing in India. Software testing is the process of evaluation of a software item to detect differences between given input and the output which is expected. Also, to assess the feature of a software item, testing assess the quality of the product. Software testing is a process which is to be done during the development process. In other words, software testing is a validation and verification process. Executing a system



in order to establish gaps, errors or missing requirements in contrary to the actual desire or requirements is testing. White Box (or glass box) testing is the process of giving input to the system and checking how the system processes input to generate the output. Black Box testing is the process of giving input to the system and checking the output without bothering how the output is generated. White Box and Glass Box Testing combined to produce the Gray Box Testing. In this, the tester has little knowledge about the internal working of the software. So, the testing of the output as well as the process carried out to generate the output. Software testing is a method of assessing the functionality of a software program. And Software testing has been classified into two main categories, dynamic testing and static testing. In this survey paper we discussed what is Testing, Terminology used in Testing, Levels in Testing and Comparison of Manual and Automation Testing, Comparison of Selenium and QTP tool and Comparison of Black, White and Gray box Testing.

Based on the study of Sriraman about the Comparative Study on Software Testing Techniques gives us the idea about the security, correctness, completeness and quality of the developed computer software.

According to Sundmark (2013) in his study about Automated Software Testing. Automated software testing is a critical enabler for modern software development, where rapid feedback on the product quality is expected. To make the testing work well, it is of high importance that impediments related to test



automation are prevented and removed quickly. An enabling factor for all types of improvement is to understand the nature of what is to be improved. We have performed a systematic literature review of reported impediments related to software test automation to contribute to this understanding. In this paper, we present the results from the systematic literature review: The list of identified publications, a categorization of identified impediments, and a qualitative discussion of the impediments proposing a socio-technical system model of the use and implementation of test automation.

Based on the study of Sundmark, the automated software testing is critical in the part of modern software development because of the feedback that must be a positive outcome, so that the web testing tool is very important in so many various moments of time. So, the developers of Web Testing Start-Up Kit will be a tool for different developed software.

According to Hengliang (2012) in his Research about Distributed Software Testing Platform Based on Cloud Resource. Computer School of Henan University of Science & Technology. In order to solve the low efficiency problem of large-scale distributed software testing, CBDSTP (Cloud-Based Distributed Software Testing Platform) is put forward. This platform can provide continuous integration and automation of testing for large software systems, which can make full use of resources on the cloud clients, achieving testing results in the real environment and reasonable allocating testing jobs, to resolve the Web application software



configuration test, compatibility test and distributed test problems, to reduce costs, improve efficiency. Through making MySQL testing on this prototype system, the verification is made for platform architecture and job allocation effectiveness.

Based on the study of Hengliang, web-based software testing will help the IT firms by reducing their resources when it comes to testing. This platform will bring efficiency and effectiveness of all products being develop by the companies.

According to Tamilarasi (2013) in his study about the Development on Software Testing Techniques and Tools. India Software Testing is the pre-eminent part of the software development life cycle process. It is the process of evaluating system or its components with the specified requirements or not. It is important process by means of accessing the quality of software, reusability and traceability of the requirements specified. There are many numbers of testing techniques and tools available for this task. Software testing is an essential part of research and lot of development has been made in this field. In this chapter, testing techniques and tools including test design tools, load and performance testing tools, test management tools, test implementation tools, test evaluation tools, static analysis tools, Basic path testing, Loop testing, Control structure testing, code-based techniques, Path testing, data flow testing, syntax testing has been described. Some distinctive latest research and development in testing strategy have been summarized.



Based on the study of Tamilarasi, the researchers knew the techniques and testing tools like the test management, test implementation and other. The group also got idea on the Software Development Life Cycle in the testing that finding the defects and bugs. The Web Testing Start-Up Kit will be very useful tool for the testing of developed web system.



Local Study

According to Vergel (2015) in his research about The Effectiveness of Test Automation. This research examines the effect of test automation on software testers. Over time testing has carved out a separate discipline distinct from software development. Software has become increasingly more complex and distributed. The pressure to release into production multiple times a day has raised the strategic importance of test automation. The history and background of testing are considered in light of a critical analysis of the current body of research. Using both qualitative and quantitative methods, a picture emerges that is in contrast to the development strategy being promoted by the Cloud Services Company where the semi-structured interviews took place. After analysis of the data, the fact that a lot of testing carried out is still manual was unexpected and means that manual testers are still required. However, the problems that are preventing automation being more successful and widely implemented are solvable and can be addressed in the next five years or less. The conclusion to this research suggests that people currently in manual software testing are in danger of being replaced. To survive in the software industry, they will need to retrain and learn to code. Otherwise, their positions will become obsolete.



Based on the study of Vergel, test automation methods for testing are truly incomparable. However, this will bring dark days for the personnel's that their job is to test system manually. Companies will embrace innovation to lessen their problem and to compete with other companies.

According to De Leon (2013) in his study about the Impacts of Test Automation on Software's Cost, Quality and Time to Market. In spite of the availability of most proficient quality assurance teams and tools, software testing has always been a time-consuming task. Thus, test automation is being profoundly practiced in most of the software industries to leverage the total development time. Although the test automation has its own advantages and disadvantages and it influences various other development phases, the higher management is particularly interested in reckoning its effects on total software's cost, quality and time. In this paper, we have tried to ascertain some of the critical factors related to test automation and cost/return of/from automation. As automation is itself a pricy activity, it requires development effort and significant time, we have attempted to enumerate test automation's impacts on software's cost, time and quality on three different software's. The results of our experiments clearly show the positive effects of test automation on cost, quality and time to market of the software.



Based on the study of De Leon, test automation is being profoundly practiced in most of the software industries to leverage the total development time. Automation in software testing lessens time and effort. As researchers, automation brings efficiency in terms of software quality and minimizes the time consumption in testing.

According to Evangelista (2012) in his study about the Comparative Study of Manual and Automated Testing for Industrial Control Software. Testing is an important activity in engineering of industrial control software. To support developers in testing their software, researchers have proposed different approaches for producing good test cases. In the last couple of years, a wide range of techniques for automated test generation have been explored with the goal of complementing manual testing. Even though there is some evidence suggesting that automatically generated test suites may even cover more code than those manually written by developers, this does not necessarily mean that these tests are effective in terms of detecting faults. As manual testing and automated code coverage-directed test generation are fundamentally different and each strategy holds its own inherent limitations, their respective merits or demerits should be analyzed more extensively in comparative studies.

Based on the study of Evangelista, automation in software testing may even cover more code than those manually written by developers. As developers of Web Testing Startup Kit, the develop project aims to be a code-less web system



application tester, enable to test a system without coding just like similar to a manual testing.

According to Velasquez (2014) In his study about Manual to Automated Testing. In today's business environment, project teams are expected to do more and deliver higher quality systems in less time with fewer resources. And when companies tighten their budget, software testing is often one of the first systems-development items to be done away with. IT systems that don't solve real business problems or don't perform as promised, impose a similar economic toll on business costs and results. Most software projects that fail to meet objectives or suffer significant schedule and budget slippage because defects are discovered too late. As a result, late product can lose revenue, customer, and market share. Manual testing is still being used at both participated organizations and some concerns have been raised at a managerial level. These concerns include timing, coverage, reusability and tester's motivation and efficiency, searching for a long-term and cost-effective solution and achieving coverage against requirements.

Based on the study of Velasquez, manual to automated testing solves big problems in today's businesses by minimizing the time consumption in testing their software product, so they can meet the objective or significant schedule in deploying their software. Automation on software testing brings not just time consumption, but it is also cost-effective.



According to Buenaventura (2015) in his study about Automated Software Testing Practices. automated software testing means nothing more than automating any part of the testing of software at any stage in the software development process. Any software testing that can be automated can be tested manually. The goal of Automated Software Testing (AST) is the same as the goal of automation in a production line, to optimize for throughput and quality by improving the speed of each stage and the repeatability of each process. In the world of software testing, automation can reduce the time it takes to uncover design flaws or trace bugs and it can improve software quality by reducing uncertainty through increased coverage by checking a greater percentage of the software or system under test (SUT) for errors, especially by way of negative testing. In many cases, AST frees up human testers to focus on manual exploratory (and context-specific) testing, which can be better suited to finding faults. This document seeks to provide information and insight into the planning, architectures or implementations, and test design strategies for AST, and to describe how AST folds into the larger issue of software economics.

Based on the study of Buenaventura, the main objective on automated testing is the same as the goal of automation in a production line, to optimize for quality by improving the speed of each stage and the repeatability of each process. Nowadays, automation in every business process is widely seen as one of the best solutions.



CHAPTER III

Technical Background

System Architecture

The following Diagram shows the relationship between different components. Usually they are created for systems which include hardware and software and this are represented in the diagram to show the interaction between them.

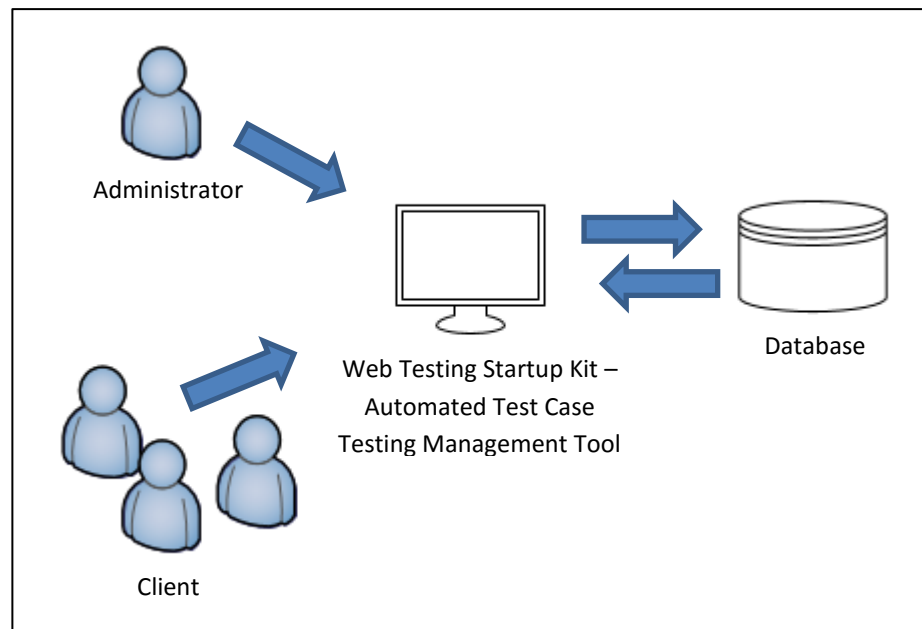


Figure 1: System Architecture



Software

Notepad++ - a text editor similar to Notepad but provides additional features, such as advanced source code editing tools. The text editor is made up of a tabbed document interface and split screen editing, while providing drag-and-drop functionality. The system will be develop using this text editor version 6.2.3. The coding part of the development will be written in this software.



Figure 2: Notepad++



MySQL – MySQL is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. The system application will be using MySQL database. All of the data and information will be stored in this database.



Figure 3: MySQL

XAMPP – XAMPP is a free and open source cross-platform web server solution stack package written in the PHP and Perl programming languages. The system will be develop using this software having a version of 5.6.36 or higher. XAMMP will served as a local web server for testing and deployment purposes.



Figure 4: XAMPP



Google Chrome & Mozilla Firefox – web browser is a software program that allows a user to locate, access, and display web pages. In common usage, a web browser is usually shortened to "browser." The development for this system application requires installers of Google Chrome version 72.0.3626.121 onwards and Mozilla Firefox version 65.0.2 onwards. This software will serve as the User Interface (UI) of the system.



Figure 5: Google Chrome & Mozilla Firefox

Bootstrap & jQuery – Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The developers have used Bootstrap version 4.0.0 and jQuery version 4.2.1. The role of these framework is to enhanced the design of the system interface.



Figure 6: Bootstrap & jQuery



JDK - The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. The development for this project requires JDK version 8.1 onwards. The role of this software is to retrieve methods and be able to communicate with browser driver in a form of json wire protocol.



Figure 7: JDK



Hardware

Computer - The developers are freely to use desktop computer or a laptop for the programming stage and for the testing of the system. The specifications of the computer on this development must have a minimum of i3 processor or higher, 2 giga bytes of RAM, and lastly the operating system should be windows 7 onwards.



Figure 8: Desktop Computer & Laptop



Server Unit - A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data center, the physical computer that a server program runs on is also frequently referred to as a server. This hardware is the best component that will reduce the time execution in performing test script testing. Dell Power edge T600 server model are one of the advisable servers for this project.

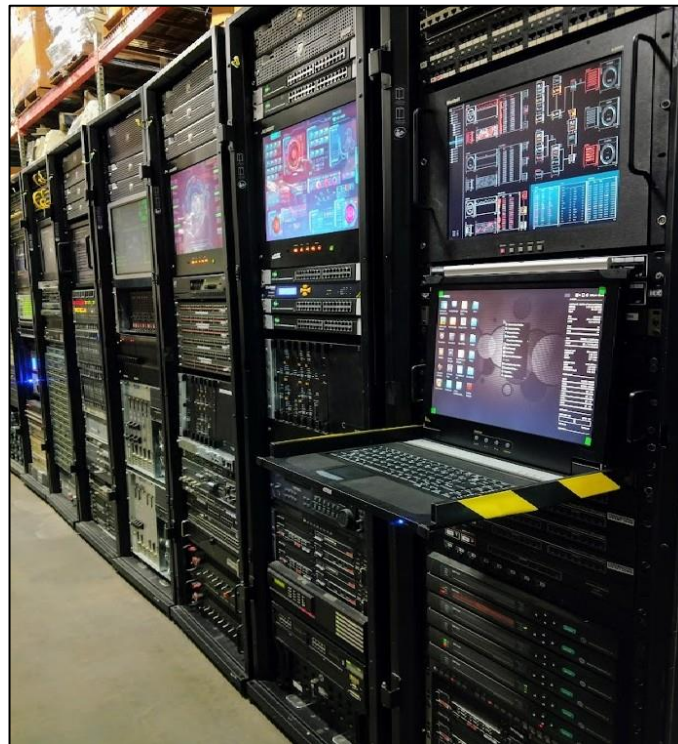


Figure 9: Server Unit



CHAPTER IV

Design and Methodology

Research Method

Developmental Research

Development, in its most generic sense, implies gradual growth evolution and change. This concept has been applied to diverse areas of study and practice. The purpose of developmental research is to assess changes over an extended period of time. Developmental research would be an ideal choice to assess the differences in the process of accomplishing web system application testing.

Developmental research is defined as the system in study of developing, evaluating, processing programs, designing that must meet the criteria of internal effectiveness. This research is particularly important in the field of instructional technology. It is based on either situation-specific problem solving or generalized inquiry procedures. The most common types of developmental research involve situations in which the development is analyzed and described. An important distinction should be made between reports of actual developmental research and descriptions of design and development models.



Population, Sample Size, Sampling Techniques

The researchers will develop a system entitled Web Testing Startup Kit – Automated Test Case Testing Management Tool, that can help all of the programmers and non-programmers to lessen the time exerted in testing and provides accurate test results for their developed system applications. The population in which constituted sample of the respondents of the study consist the students of the Computer Science course of Taguig City University and representative of each department.

All of the participants were selected through the use of selected sampling. This method was conducted where each member of a population has an equal opportunity to become research respondents. This technique is the most efficient sampling procedure. Then, the researchers must define the population first and list down all the members of the population. The researchers used Slovin's formula to get the exact sampling quantity of the population.

$$n = \frac{N}{1 + Ne^2}$$



Where:

n = sample size

N = total no. of population

e = margin of error (0.05)

Table 1: Shows the population of the respondents according to their positions.

Slovin's formula was used in determining the sample size.

Positions	Population	Sample Size	Percentage (%)
Computer Science Students Taking Advance Web Management Subject	74	53	97.76%
IT Professionals	5	2	2.24%
Total	79	55	100%



Likert Scale Scoring System

Table 2: Likert Scale

Likert	Numerical Rating	Verbal Interpretation
1	4.00 – 3.25	Strongly Agree
2	3.24 – 2.25	Agree
3	2.24 – 1.25	Disagree
4	1.24 – 0	Strongly Disagree

Table 2: Demonstrate the scoring system that was used in calculating the level of acceptance of the developed system. The point that has 4.00– 3.25 was the highest score and 1.24 – 0 was the lowest score.

Data Gathering Procedures

The data gathered for this research was collected through a survey questionnaire. The survey was created using suitable questions from related research and individual question formed by the researchers. Respondents will evaluate the system, Web Testing Startup Kit – Automated Test Case Testing Management Tool.



Pre-Survey Result

Questions	Weighted Mean	Verbal Interpretation
1. Automated software testing is faster in execution rather than manual testing	3.54	Highly Acceptable
2. Automated software testing is cheaper compared to manual testing in a long run	3.61	Highly Acceptable
3. Automated software testing is more reliable compared to the traditional testing	3.67	Highly Acceptable
4. Automated software testing is more detailed in terms of producing reports.	3.64	Highly Acceptable
5. Automated software testing can handle load & performance testing	3.78	Highly Acceptable
6. Automated software testing is more convenient rather than traditional test case testing.	3.64	Highly Acceptable
7. Automated software testing can eliminate human error	3.70	Highly Acceptable
8. Automated software testing enables us to deliver high-quality software	3.71	Highly Acceptable
9. Automated software testing is more productive compared to manual testing	3.66	Highly Acceptable
10. Automated software testing can overcome the limitations of manual testing	3.81	Highly Acceptable



The evaluation procedures are the following:

1. Explaining the problems mostly encountered.
2. Explaining the step by step procedures, main functions of developed system.
3. The researchers present the survey questionnaire to the respondents to answer it.
4. When the respondents are done answering the survey questionnaire, the researchers must collect all the questionnaire, and interpret the data according to the appropriate statistical treatment.

Internet research:

The researchers use the internet to find related information about the developed system. This research method can provide an intermediate, quick and worldwide access to information, although results may be affected by unrecognized bias, difficulties in verifying a writer's credentials, and whether the searcher has sufficient skill to draw results from abundance of materials available.

Waterfall Model

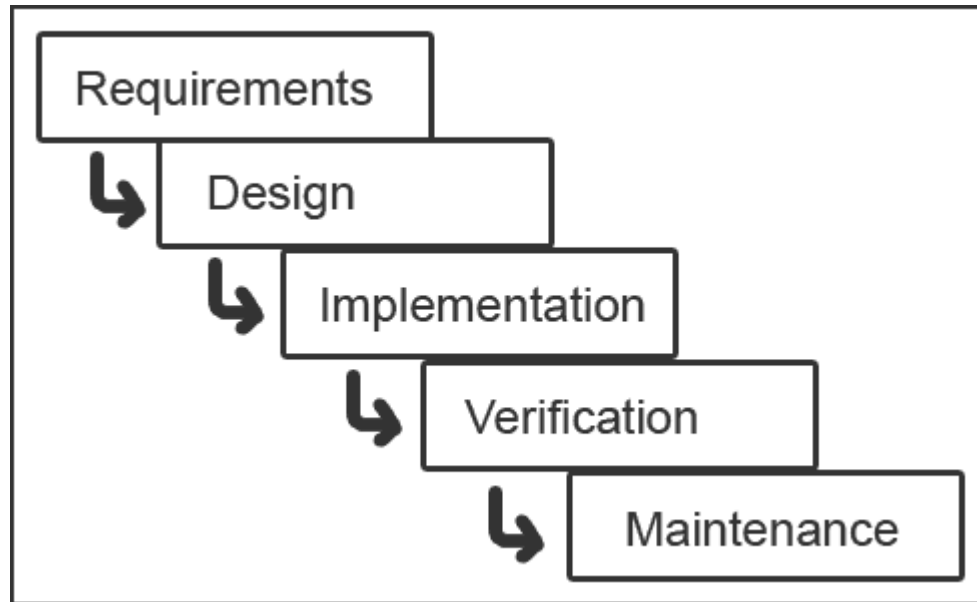


Figure 10: Waterfall Software Development Cycle

The researchers have chosen the Waterfall Software Development Methodology because the researchers project development progress flows steadily towards the conclusion.



Requirements

The requirements/analysis phase happened when the researchers were thinking about the software and hardware requirements to be used for the development of Web Testing Startup Kit – Automated Test Case Testing Management Tool.

Design

The designing phase was done with the help of the internet. The researchers look on related automated software testing systems that were posted from different forums.

Implementation

Implementing the “Web Testing Startup Kit – Automated Test Case Testing Management Tool” by uploading from different open source websites like GitHub.

Verification

The verification phase has done by the respondents who witnessed the complete scope and functionalities of the system. Data gathering is involved in this phase in which the researchers have used questionnaires in order to know if the system is acceptable or not. The user’s feedback will serve as a guide to the researchers what should be the changes to upgrade more the system.



Sample Test Parameters

Researchers made a questionnaire on Web Testing Startup Kit

Name (optional): _____ **Date:** _____

Course: IT/CS ☐ Non – IT/CS ☐

Gender: Male ☐ Female ☐

Age: (15 – 20) ☐ (21 – 25) ☐ (26 – Above) ☐

Please complete the following questionnaire with specific regard to the above inquiry, by placing a CHECK ☒ in the appropriate box

Functionality	4	3	2	1
1. The system easily stores test projects.				
2. The system can test web-based system efficiently.				
3. The system generates detailed test information.				
4. The system navigation buttons function properly.				
Reliability				
1. The system provides accurate test result.				
2. The system validates the user's information accurately.				
3. The system can handle consecutive processes.				
4. The system minimizes software test timeframe.				
Usability				
1. The system is user-friendly.				



2. The system performs zero tolerance of error.				
3. The system is useful for the software testers.				
4. The system makes the software testers life easier.				
Efficiency				
1. The system provides faster repeatable testing process.				
2. The system minimizes the time exerted in testing a certain web-based system.				
3. The system is more effective than manual setting.				
4. The system works in a long period of time as needed.				
Portability				
1. The system has a low software requirement.				
2. The system has a low hardware requirement.				
3. The system is easy to install and launch.				
4. Overall, I 'am satisfied with how easy it is to use this system.				
Security				
1. The system account credential uses hashing algorithms.				
2. The system requires user login.				
3. The system has user level of access.				
4. The system will log out when the system is standby for 30 minutes.				



Maintenance

The maintenance phase is the stage of development where the developer will continue to monitor and support the system to ensure it continues to meet the benefit of the users.

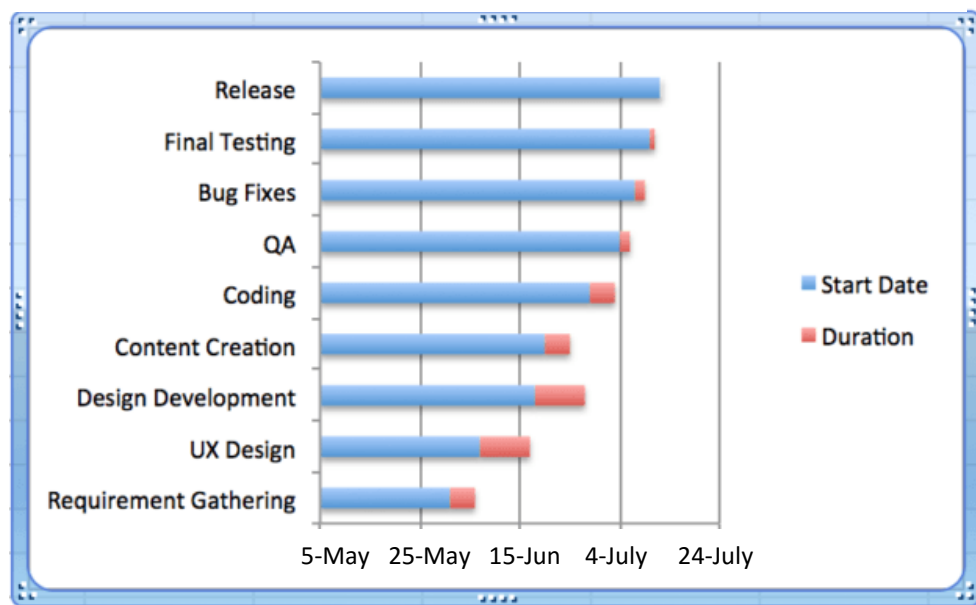


Figure 11: Maintenance Plan Chart



Proposed Login Flowchart

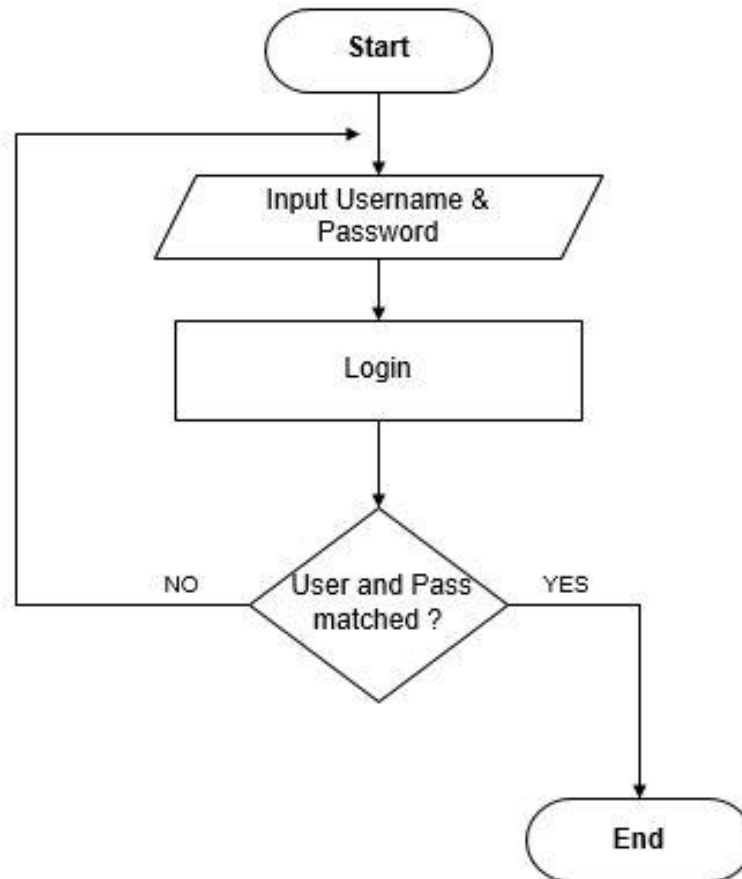


Figure 12: Login Flowchart

- The system has two (2) user level of access and each level has its own designation in the system.



Forgot Password Flowchart

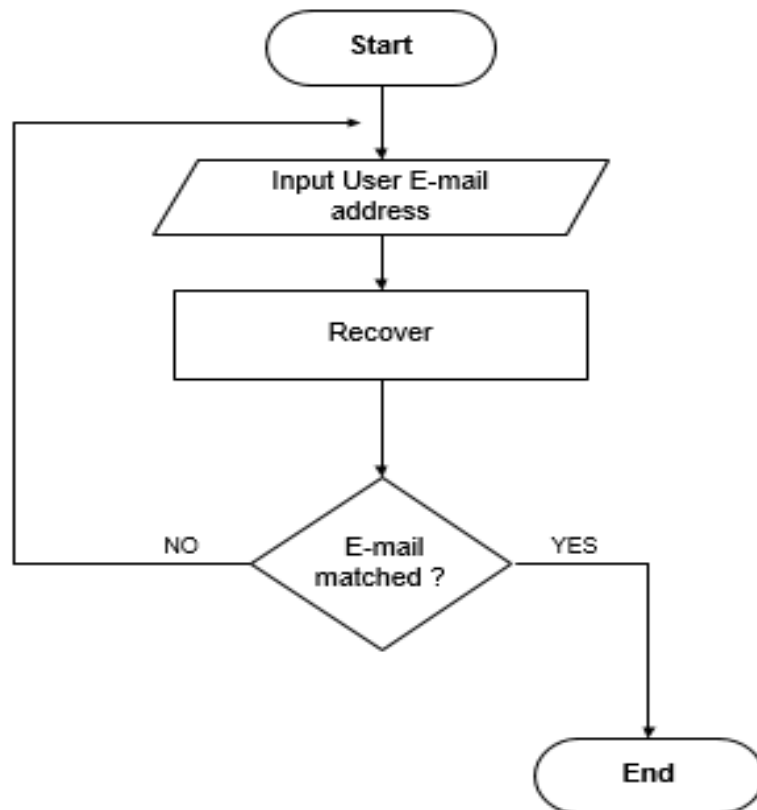


Figure 13: Forgot Password Flowchart

- The system has a password retrieval process which requires valid E-mail address of the user to send his/her password via E-mail.

Add New Test Project Flowchart

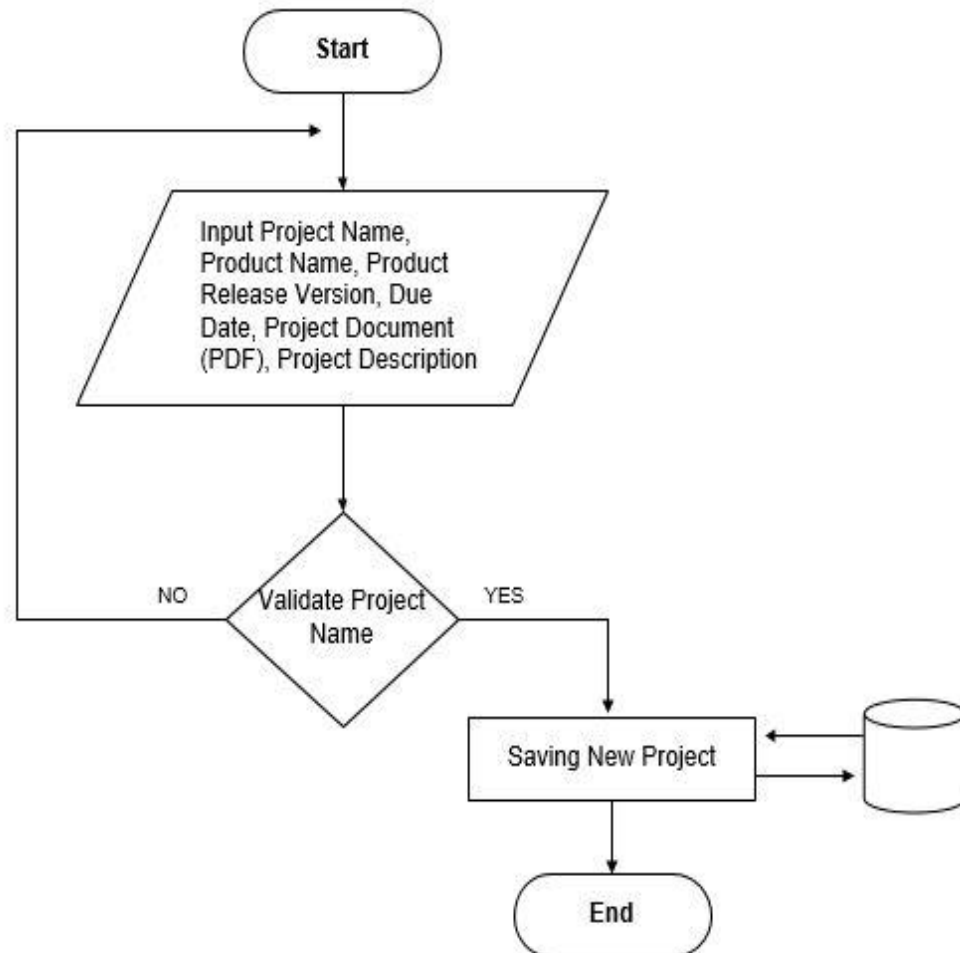


Figure 14: Add New Test Project Flowchart

- The system will store the new test project to the database if the project name is not same from the existing.

Edit Test Project Flowchart

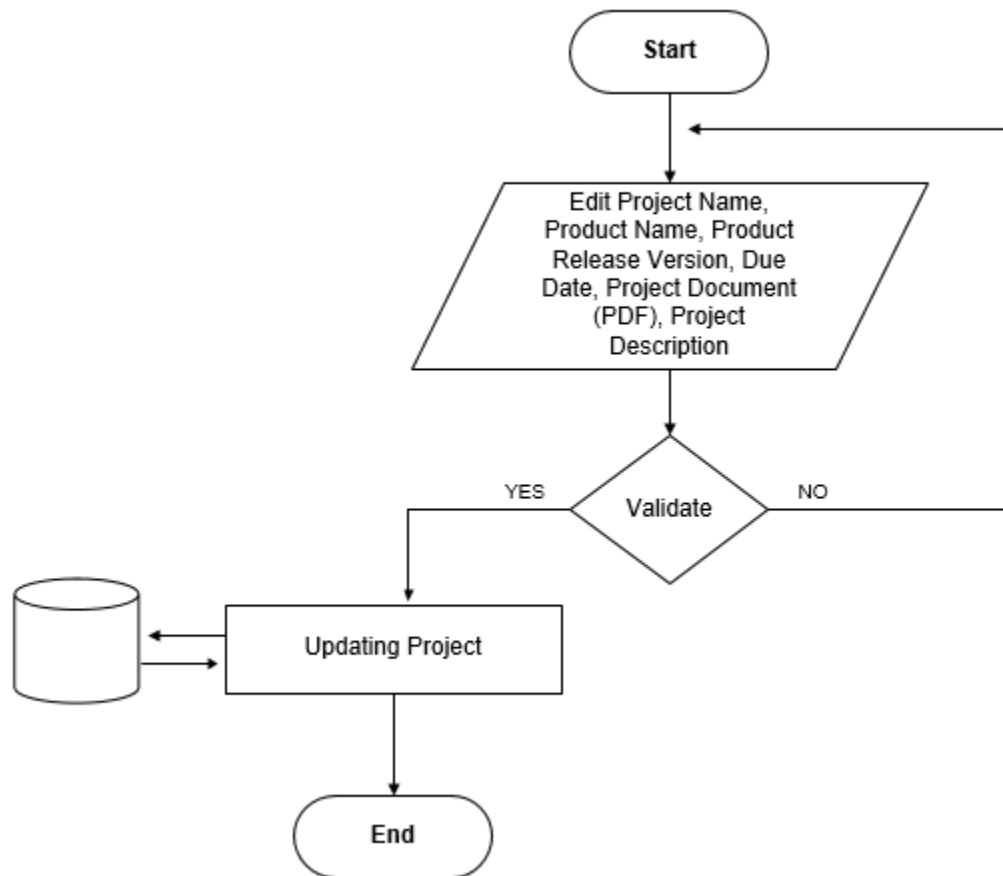


Figure 15: Update Test Project Flowchart

- The system will update the project details from the database if the inputted data are valid.

Delete Test Project Flowchart

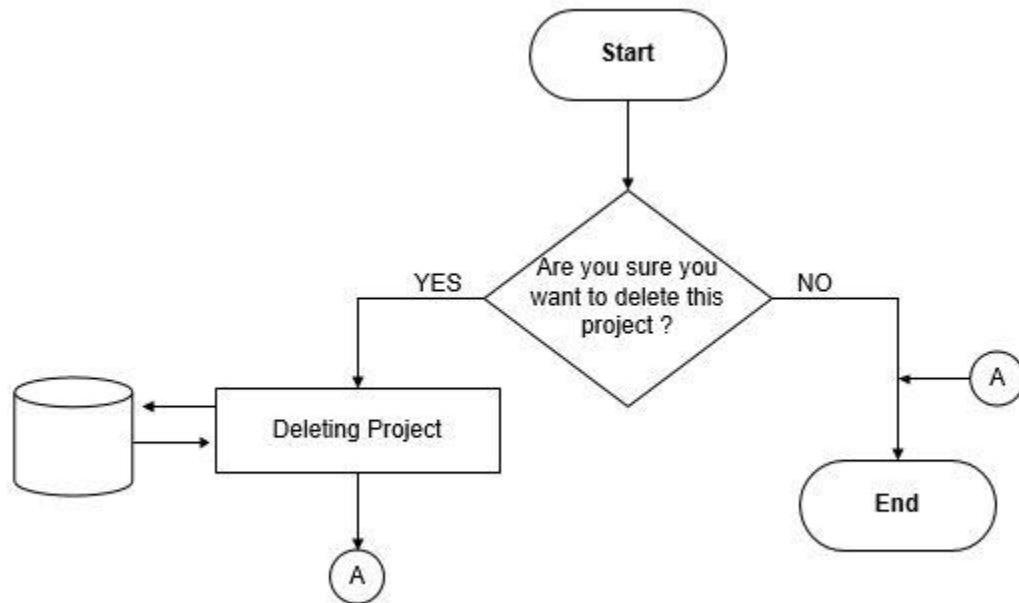


Figure 16: Delete Test Project Flowchart

- The system will delete the users selected test project after confirming if he/she wants to continue.

Add New Test Script Flowchart

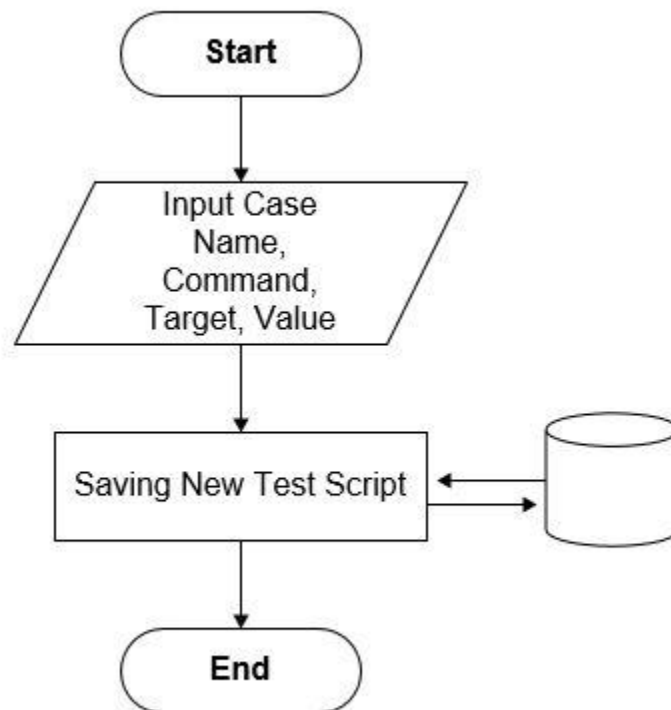


Figure 17: Add Test Script Flowchart

- The system will store the data to the database from the user who inputted all of the data.

Start Test Script Flowchart

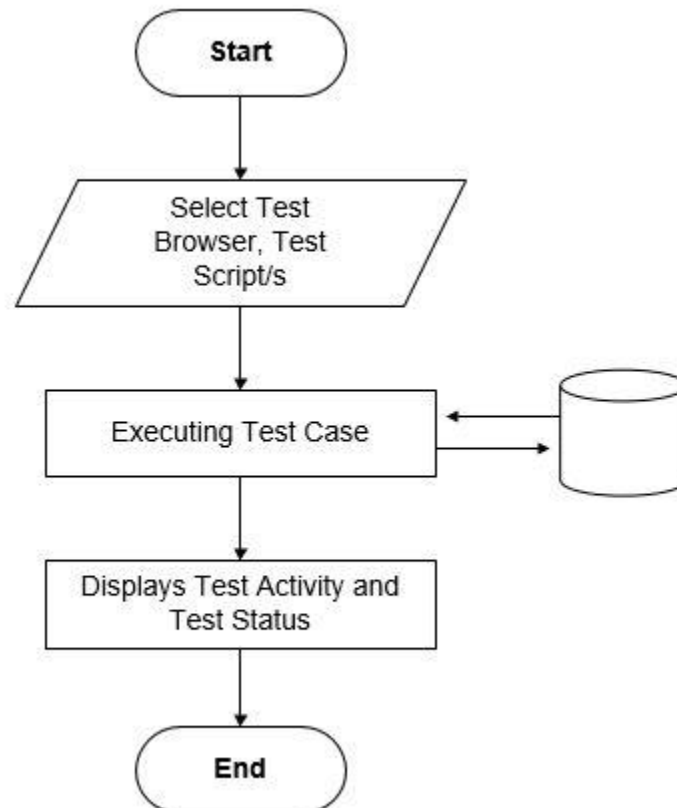


Figure 18: Start Test Script Flowchart

- The system will display the testing activity or follows the instruction based on the test script.

Add User Account Flowchart

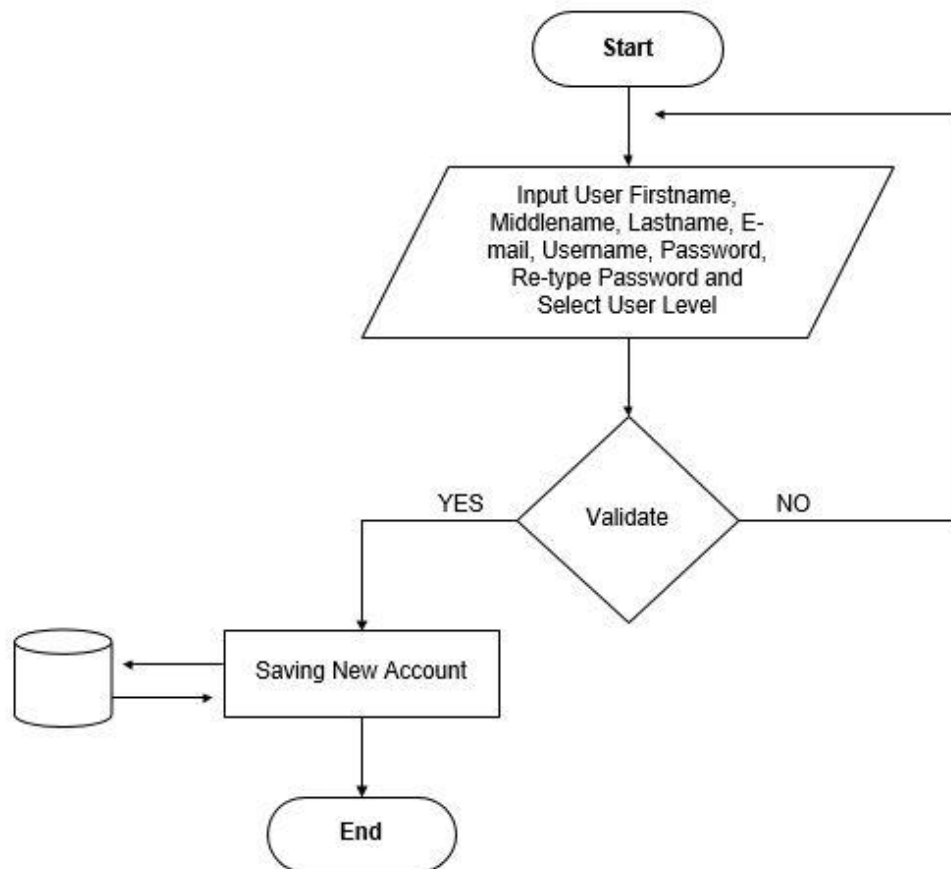


Figure 19: Add User Account Flowchart

- The system has a certain user level which is the administrator who only has the access of adding a user account.



Data Flow Diagram Level 0

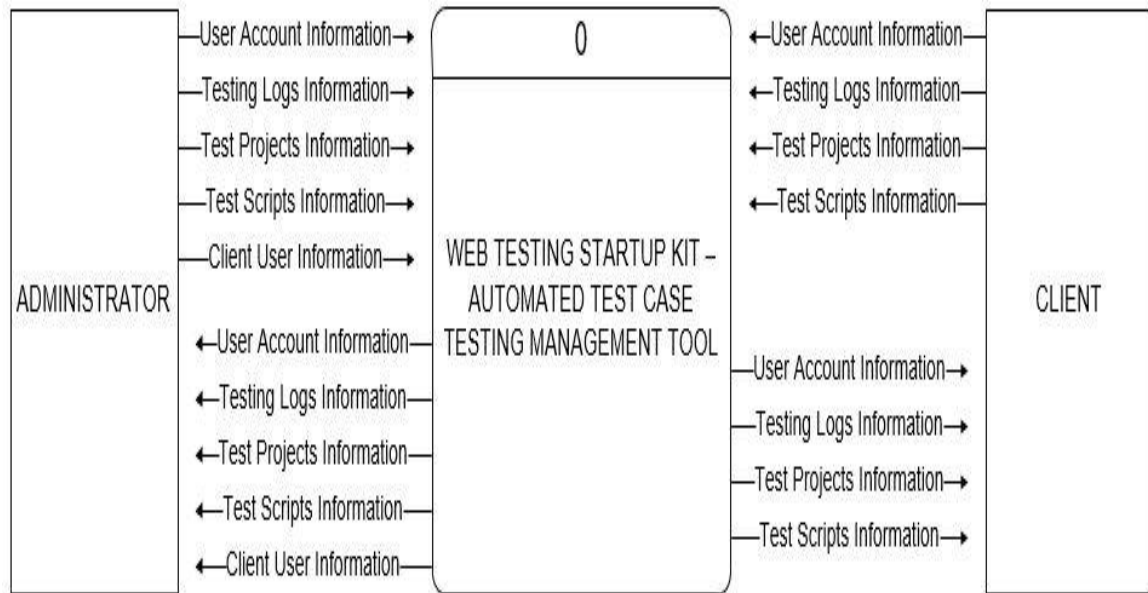


Figure 20: Data Flow Diagram Level 0

- This is the overview of all information process in the Web Testing Tool.

Data Flow Diagram Level 1

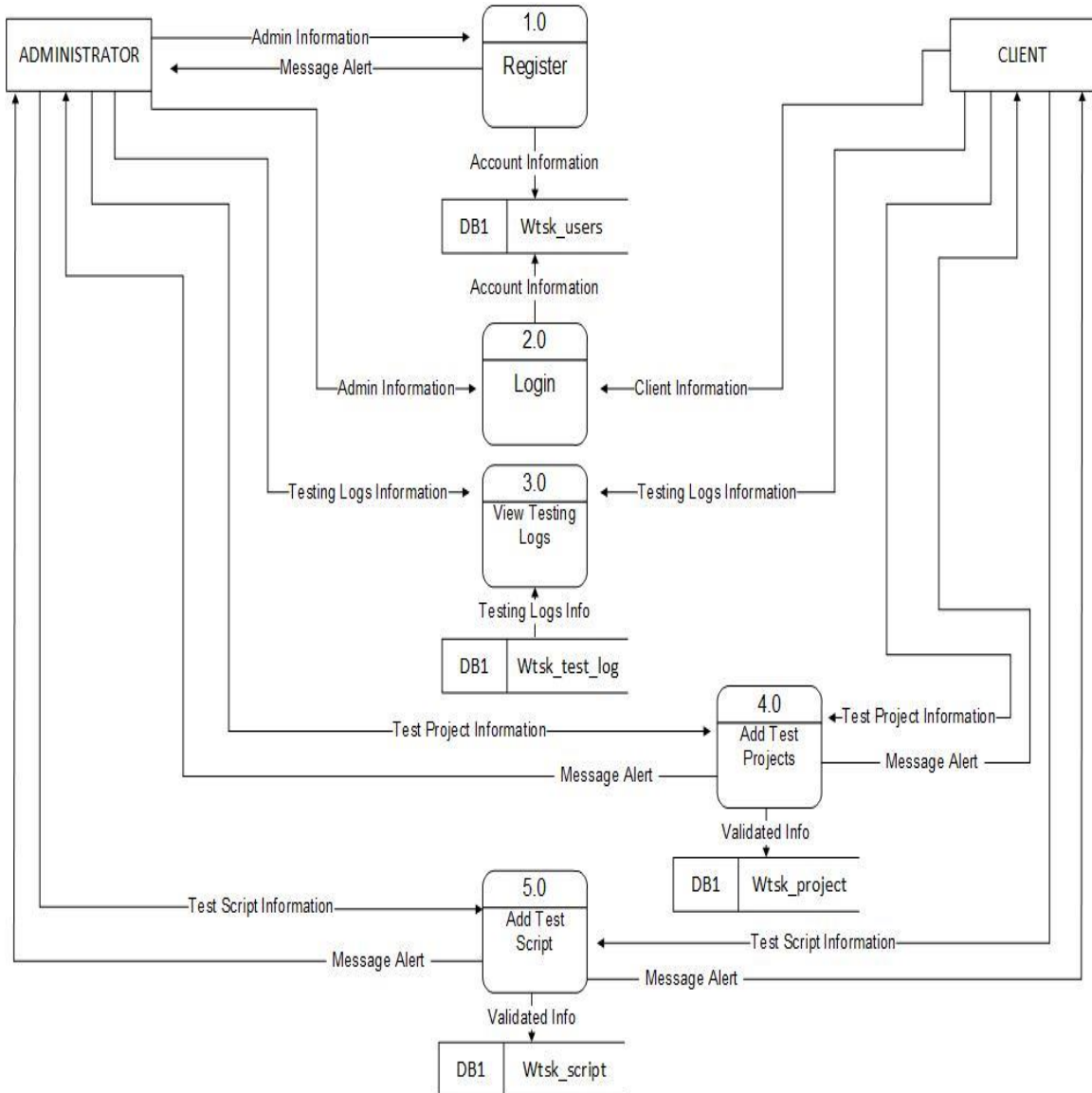


Figure 21: Data Flow Diagram Level 1

- In process 1.0, all of the user account information will be gathered and stored in a database.



- In process 2.0, the users are required to input their account information to gain the access from the system.
- In process 3.0, the users can view their previous testing logs via report.
- In process 4.0, all of the project details are gathered and stored in a database.
- In process 5.0, all of the test script information will be gathered and stored in a database.



Data Flow Diagram Level 2

Registration

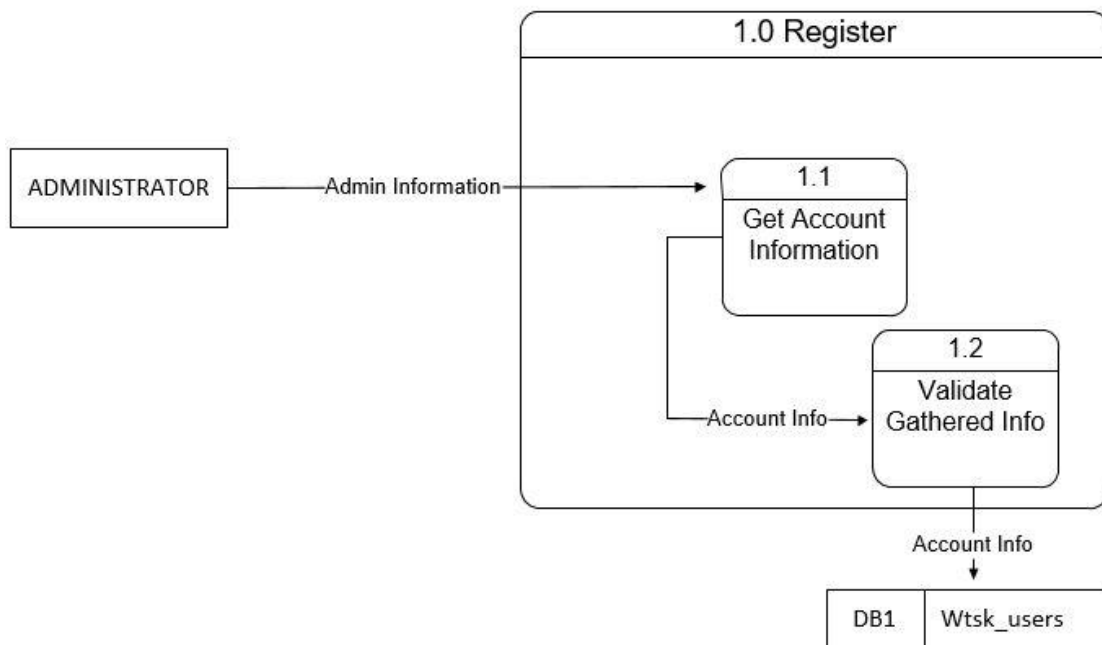


Figure 22: Registration Data Flow Diagram Level 2

- In process 1.0, the administrator will be storing new user account.
- In process 1.1, the administrator will be inputting the user accounts information.



- In process 1.2, all gathered information will be validated before storing to the database.

Login

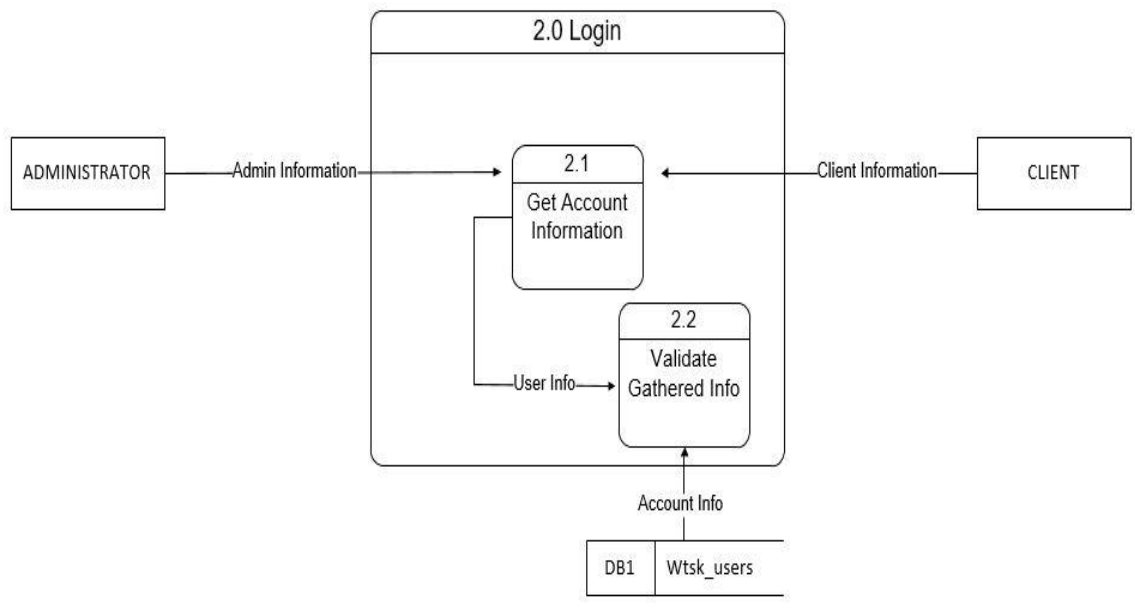


Figure 23: Login Data Flow Diagram Level 2

- In process 2.0, the users will log on to the system regardless to their designations.
- In process 2.1, the users will be inputting their account information.
- In process 2.2, all gathered information will be validated from the database.

Add Test Project

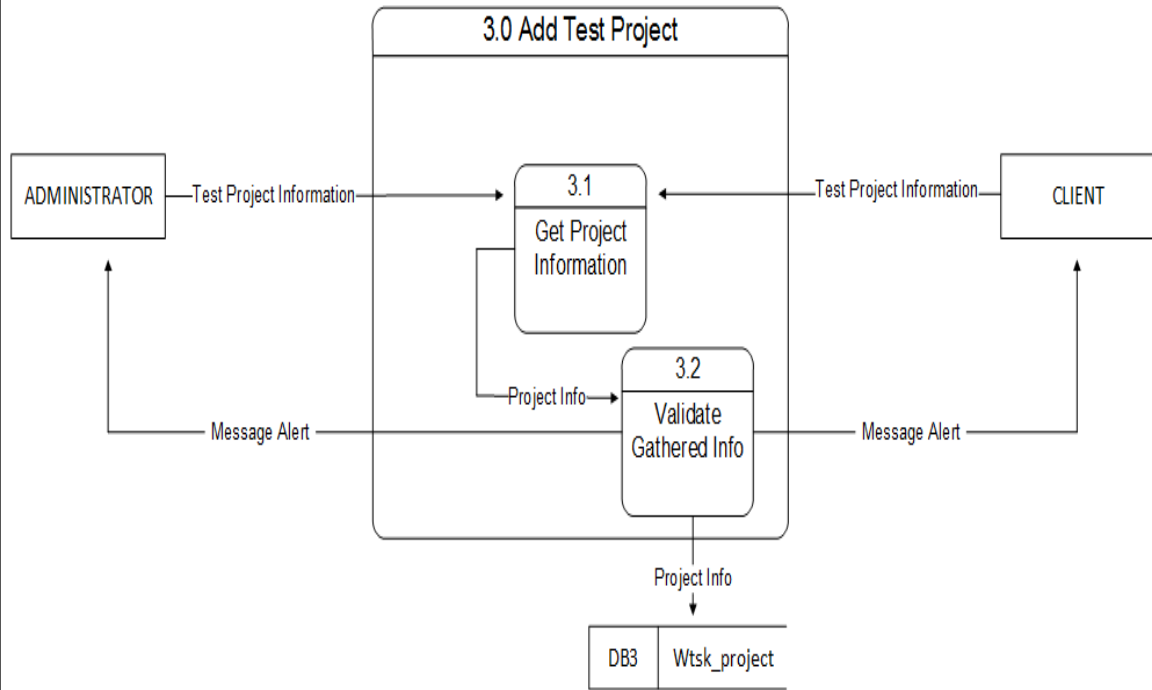


Figure 24: Add Test Project Data Flow Diagram Level 2

- In process 3.0, the users will be storing a new test project.
- In process 3.1, the users will be inputting their test project information.
- In process 3.2, all gathered information will be validated before storing to the database and notifies the user.

View Test Logs

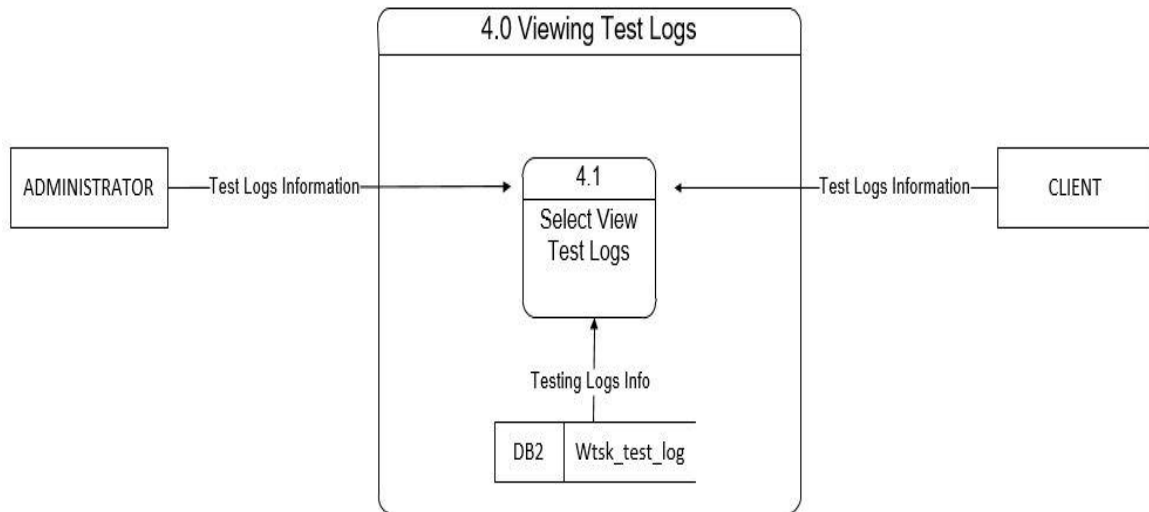


Figure 25: View Test Logs Data Flow Diagram Level 2

- In process 4.0, the users will view a recent test log.
- In process 4.1, the selected test log will be gathered from the database and displays in the system.

Add Test Script

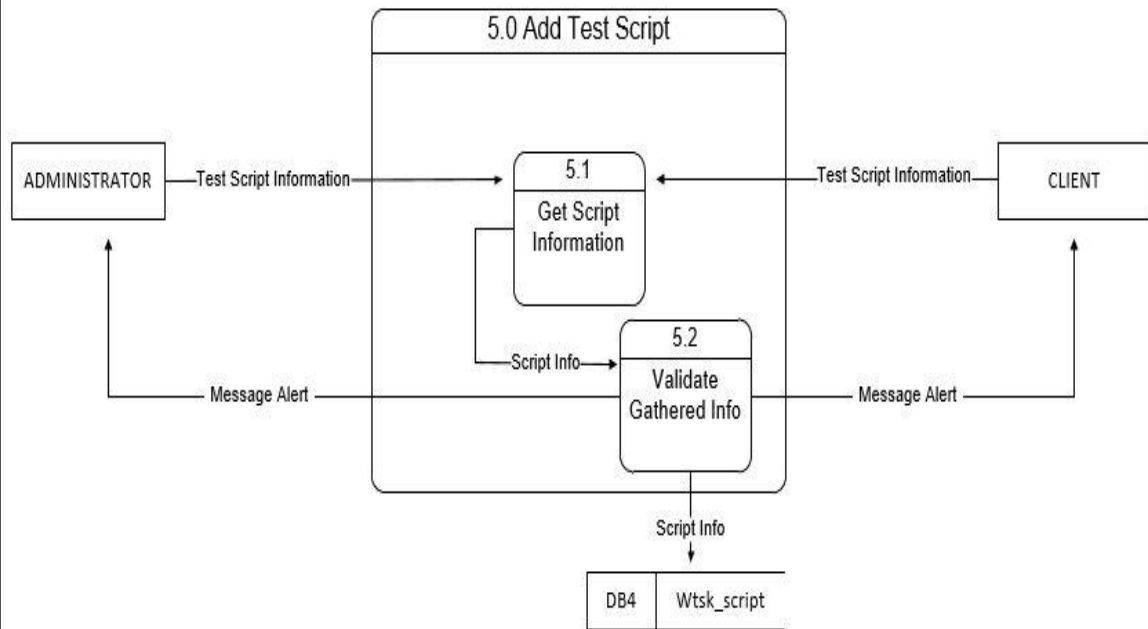


Figure 26: Add Test Script Data Flow Diagram Level 2

- In process 5.0, the users will be storing a test script to the database.
- In process 5.1, the users will be inputting their test script information.
- In process 5.2, all gathered information will be validated before storing to the database and notifies the user.



CHAPTER V

Results and Discussions

Post Survey Result

Functionality	Weighted Mean	Verbal Interpretation
1. The system easily stores test projects.	3.76	Highly Acceptable
2. The system can test web-based system efficiently.	3.60	Highly Acceptable
3. The system generates detailed test information.	3.60	Highly Acceptable
4. The system navigation buttons function properly.	3.59	Highly Acceptable
Overall Mean	3.64	Highly Acceptable

The researchers overview the functionality of the system with the average of 3.64, with verbal interpretation of highly acceptable means that they agree that the functionality will be experienced only when they interact with the system.

Reliability	Weighted Mean	Verbal Interpretation
1. The system provides accurate test result.	3.65	Highly Acceptable
2. The system validates the user's information accurately.	3.71	Highly Acceptable
3. The system can handle consecutive processes.	3.59	Highly Acceptable
4. The system minimizes software test timeframe.	3.63	Highly Acceptable
Overall Mean	3.65	Highly Acceptable



The researchers accept the system's reliability that averages 3.65 which indicates it is acceptable, it will meet its goal because they were able to witness the system reliability in terms of test results.

Usability	Weighted Mean	Verbal Interpretation
1. The system is user-friendly.	3.82	Highly Acceptable
2. The system performs zero tolerance of error.	3.45	Highly Acceptable
3. The system is useful for the software testers.	3.72	Highly Acceptable
4. The system makes the software testers life easier.	3.47	Highly Acceptable
Overall Mean	3.62	Highly Acceptable

The researchers recognize the system's usability with the average 3.62, which says acceptable, it is really important to let the users see the capabilities of the system, if given the right amount of time to the students.

Efficiency	Weighted Mean	Verbal Interpretation
1. The system provides faster repeatable testing process.	3.63	Highly Acceptable
2. The system minimizes the time exerted in testing a certain web-based system.	3.54	Highly Acceptable
3. The system is more effective than manual setting.	3.59	Highly Acceptable
4. The system works in a long period of time as needed.	3.81	Highly Acceptable
Overall Mean	3.64	Highly Acceptable



The researchers recognize the system's efficiency that averages 3.64 which say acceptable. The efficiency of the system would help increase the productivity and quality of the program.

Portability	Weighted Mean	Verbal Interpretation
1. The system has a low software requirement.	3.62	Highly Acceptable
2. The system has a low hardware requirement.	3.65	Highly Acceptable
3. The system is easy to install and launch.	3.63	Highly Acceptable
4. Overall, I 'am satisfied with how easy it is to use this system.	3.75	Highly Acceptable
Overall Mean	3.66	Highly Acceptable

The researchers saw the portability averages at 3.66 which says the system is launchable in different environment.

Security	Weighted Mean	Verbal Interpretation
1. The system account credential uses hashing algorithms.	3.68	Highly Acceptable
2. The system requires user login.	3.73	Highly Acceptable
3. The system has user level of access.	3.89	Highly Acceptable
4. The system will log out when the system is standby for 30 minutes.	3.55	Highly Acceptable
Overall Mean	3.71	Highly Acceptable

The researchers saw the security averages at 3.71 which says the tool is secured from possible attacks like SQL injection.



Summary

The automated software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not hence, to identify the defects, to ensure that the product is defect free in order to produce the quality product. The introduction of computers has given business an interesting upper hand by supplying companies with endlessly amounts of information. In today's competitive business environment, speed is the key for all the business processes, there is a need for many automated systems that are capable of replacing or reducing human effort in their daily activities and jobs.

Web Testing Startup Kit – Automated Test Case Testing Management Tool aims to provide an open source web system application testing for the developers who manually evaluates their developed system which encounters many difficulties. Automating software testing is widely accepted as a way for businesses to reduce the use of resources, save time and eliminate scheduling issues. The said disadvantage's in manual system testing is a proof that Web Testing Startup Kit – Automated Test Case Testing Management Tool is one of the solutions to help the developers out there.



Conclusion

The automation of some kinds of software testing can dramatically increase the productivity and effectiveness of software development projects, rarely it is found that majority of the requirements can be covered by automated tests. Some testing is still best done manually because human testers are far more flexible and creative than any computer program but, this traditional way of testing has its limitation. The challenges of an automated testing model include maintenance of tests scripts. As user interface change, new scripts require development and existing scripts updating. Unless test scripts are going to be reused and maintained then the initial investment in time required writing the scripts may prove too onerous. In addition, some scripts still require human intervention to verify the test was successful, for example checking an expected output.



Recommendation

These are the following recommendations made by the developers of the system:

1. It is recommended that the users must have a knowledge from different types of software testing.
2. Users must start learning about the basics in writing a test case.



References

Diangson, Louie V. (2015) *Why Automated testing*. Retrieved from

<https://www.google.com/search?why-automated-testing>

Farouk, Shabib M. (2015) *Automation-testing*. Retrieved from

<https://www.guru99.com/automation-testing.html>

Gee, Garret C. (2016) *Selenium pros and cons*. Retrieved from

<https://wiki.saucelabs.com/display/DOCS/pros&812cons>

Green, Samantha O. (2016) *Selenium Compatibility with other browsers*. Retrieved from

<http://plmun.edu.ph/research.php?researchid=12>

Dr. Gurban, Ataizi P. (2018) *Selenium introduction*. Retrieved from

<https://www.edureka.co/blog/what-is-selenium/>

Herand, Deniz G. (2016) *How web automated testing works*. Retrieved from

<http://article.sapub.org/10.5923.s.mm.201401.04.html>

Murray, Kathleen P. (2016) *Benefits of Automated Software Testing*. Retrieved from

<https://www.yugatech.com/feature/automated+software+testing>

Smith, Rachel L. (2018) *Automated Testing*. Retrieved from

<https://www.google.com/search?firefox-b-bd&q=automated+testing>

Sunil, Jadhav C. (2017) *Automated vs Manual Testing*. Retrieved from

<https://www.guru99.com/automated-vs-manual-testing.html>



Appendices



Appendix A:

User Manual

1.0 General Information

General information explains the general terms the system and the purpose for which it is intended.

1.1 System Overview

Web Testing Startup Kit – Automated Test Case Testing Management Tool is a web testing tool with the use of database. This software operates on a desktop computers and laptops in Windows operating system.

1.2 Organization of the Manual

The user's manual consists of five sections: General Information, System Summary, Getting Started, Using the System and Reporting.

General Information section explains the general system and the purpose for which it is intended.

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access level, and the systems behavior in case of any contingencies.



Getting Started section explains how to get TCU Student Clearance and install in a computer. The section presents briefly how to configure the application to run it.

Using the System section provides a detailed description of system functions.

Reporting section describes the way the information collected by the application are presented, and the process to access such information.

2.0 System Summary

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system configuration, user access level and systems behavior in case of any contingencies.

2.1 System Configuration

Web Testing Startup Kit – Automated Test Case Testing Management Tool operates in both 64 bit and 32-bit Windows OS.

Hardware Requirements

1. Desktop Computer or Laptop



Minimum System Requirements

1. i3 processor
2. 2gb RAM
3. Windows 7 OS

Recommended System Requirements

1. i7 processor
2. 8gb RAM
3. Windows 10 OS

Software Requirements

1. Notepad++ version 6.2.3 onwards
2. XAMPP version 5.6.36 onwards
3. Google Chrome version 71 onwards
4. Mozilla Firefox version 65.0.2 onwards
5. JDK version 8.1 onwards
6. Bootstrap version 2017 onwards
7. jQuery version 2017 onwards

2.2 User Access Levels

There are Administrator and Client users in the system.

Administrator

1. Handles test project information
2. Handles account information
3. Handles test logs information
4. Handles test scripts information
5. Handles client's information



Client

1. Handles test project information
2. Handles account information
3. Handles test logs information
4. Handles test scripts information

2.3 Contingency

This section will explain and give procedures to any kind of errors you may encounter throughout the application.

Connection Error

1. Server error - The target server is down and cannot be found.
Troubleshoot: Checking the Apache Server if it is running. Checking the Connection url if it is correct, example (localhost/wtsk)

Browser Compatibility Error

2. Browser Version – The system is not compatible with the older versions from 71 below of Google Chrome and a version 64 and below, if the user is using Mozilla Firefox.



2.0 Getting Started

Getting Started section explains how to get Web Testing Startup Kit and install it on a computer. The section presents briefly how to configure the application to run it.

3.0 Installation

You need to install all the software required for the application to run.

XAMMP – Is an application that runs the apache server and MySQL.

Download Link:

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwixw4DA7LHhAhUVfXAKHQOfAmsQFjAAegQIBRAB&url=https%3A%2F%2Fwww.apachefriends.org%2Fdownload.html&usq=AOvVaw0FJaxz5J9FUgyQldXelZ6c>

JDK – Is a java application library that reads and execute java codes in a java program.

Download Link:

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>



4.0 System Login

You need to install all the software required for our application to run.

The image shows a login screen for a 'Web Testing Startup Kit'. The title 'Web Testing Startup Kit' is at the top in a large, bold, white font. Below it, the subtitle 'Automated Testcase Testing management tool' is in a smaller white font. There are two input fields: 'Username' and 'Password'. The 'Username' field has a green icon of a person on the left. The 'Password' field has an orange icon of a padlock on the left. Below these fields is a green 'Login' button. Three orange arrows point to the fields and button: arrow 1 points to the Username field, arrow 2 points to the Password field, and arrow 3 points to the Login button.

1. **Username** - Input here the username that you created.
2. **Password** - Input here the password that you created.
3. **Login** – This button will let you proceed to home page.



4.1 Using the System

Using the System section provides a detailed description of system functions.

4.1.1 Home Page

The screenshot displays the system's home page. At the top, a blue 'ADD PROJECT' button is labeled with a '2'. Below it, a search bar is labeled with a '3'. A table lists project details with columns: Project Name, Product Name, Start & Due date, and Date Created. A row shows 'sad2 project testing', 'ROAS', '2019-06-30', and '2019-03-29 06:56:04'. To the right of the table, two progress bars are shown: 'Usability rate: 100%' and 'Overall Test Progress: 100%', with arrows '8' and '9' pointing to them respectively. Below the progress bars are four buttons: 'Open Project' (labeled '4'), 'Edit' (labeled '5'), 'Delete' (labeled '6'), and 'VIEW DOCUMENTS' (labeled '7'). A bracket labeled '1' encompasses the entire table and button area.

1. **Project Information** – It shows the complete project detail.
2. **Add Project** – It redirects to add project page.
3. **Search Field** – This field filters the data from the project table.
4. **Open Project** – This button will display the project documents.
5. **Edit** – This button can update the project information.
6. **Delete** – This button can delete the project.
7. **View Documents** – This button shows the uploaded project document.
8. **Usability Rate** – This bar shows how usable the project.
9. **Overall Test Progress** – This bar shows the overall progress of a project.



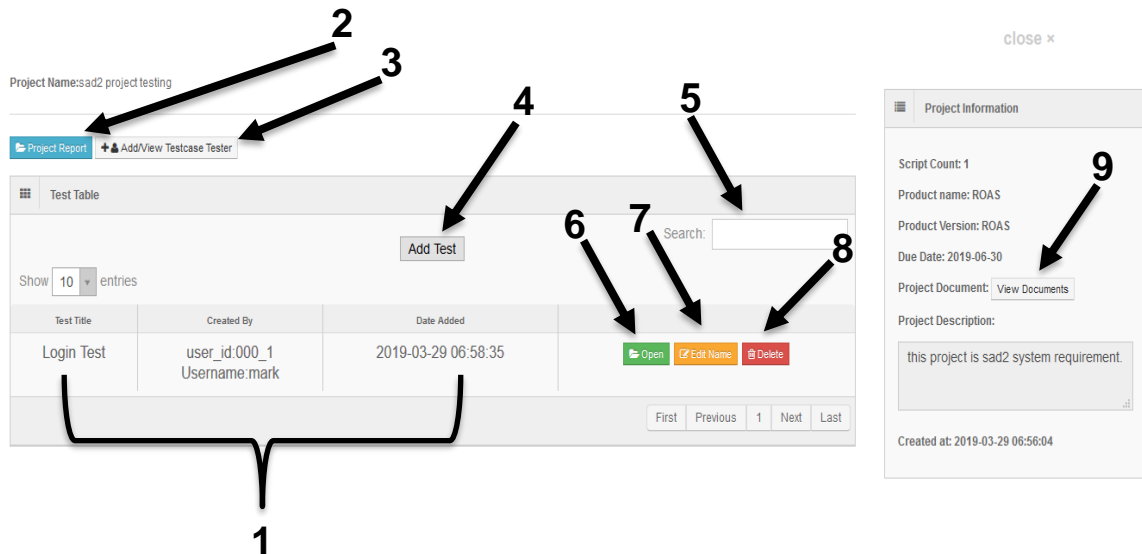
4.1.2 Add Project Form

The screenshot shows a web form titled "Project Information" with a "close x" button in the top right corner. The form contains the following fields and elements:

- Project Name**: A text input field, indicated by arrow 1.
- Product Name**: A text input field, indicated by arrow 2.
- Product Release Version**: A text input field, indicated by arrow 3.
- Due Date**: A date input field with a placeholder "mm / dd / yyyy", indicated by arrow 4.
- Project documents (PDF)**: A section with a "No file selected" button and a "Choose File" button, indicated by arrow 5.
- Project Brief Description**: A text area with a placeholder "Type here...", indicated by arrow 6.
- Submit**: A green button at the bottom right, indicated by arrow 7.

1. **Project Name** – Input here your project name.
2. **Product Name** – Input here your projects product name.
3. **Product Release Version** – Input here your products version.
4. **Due Date** – Input here the estimated date that your project testing will be done.
5. **Project Documents** – Upload here the project document ex. DFD, Flowchart, etc...
6. **Project Brief Description** – Input here your project description.
7. **Submit** – This button will add your project information.

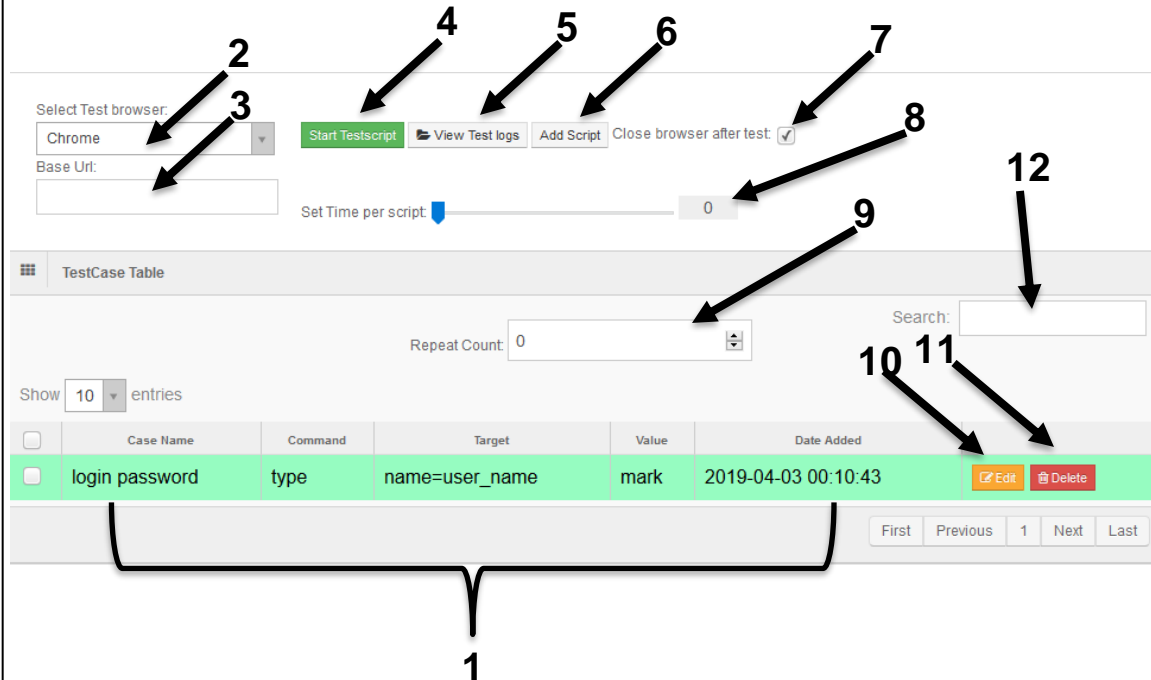
4.1.3 Open Project



The screenshot displays a web application interface for managing projects. At the top, the project name is 'sad2 project testing'. Below this, there are two tabs: 'Project Report' and 'Add/View Testcase Tester'. The 'Project Report' tab is active, showing a 'Test Table' with columns for 'Test Title', 'Created By', and 'Date Added'. A table entry for 'Login Test' is visible, created by 'user_id:000_1' with the username 'mark'. To the right of the table is an 'Add Test' button. Below the table, there are three buttons: 'Open' (green), 'Edit Name' (orange), and 'Delete' (red). A search field is located above these buttons. To the right of the main interface is a 'Project Information' sidebar containing details like 'Script Count: 1', 'Product name: ROAS', 'Product Version: ROAS', 'Due Date: 2019-06-30', and a 'View Documents' button. The sidebar also shows the project description: 'this project is sad2 system requirement.' and the creation date: 'Created at: 2019-03-29 06:56:04'. Numbered callouts 1 through 9 point to specific elements: 1 points to the project description in the sidebar; 2 points to the 'Project Report' tab; 3 points to the 'Add/View Testcase Tester' button; 4 points to the 'Add Test' button; 5 points to the search field; 6 points to the 'Open' button; 7 points to the 'Edit Name' button; 8 points to the 'Delete' button; and 9 points to the 'View Documents' button.

1. **Project Description** – It shows the project information.
2. **Project Report** – It shows the overall progress of the tested system.
3. **Add/View Testcase Tester** – This button can add a tester to perform collaboration with the admin and view the client system.
4. **Add Test** – This button can add project.
5. **Search Field** – This field filters the data from the project table.
6. **Open Project** – This button will display the project documents.
7. **Edit Name** – This button can update the title of the project.
8. **Delete** – This button can delete the project.
9. **View Documents** – This button will display the uploaded project document.

4.1.4 Test Script Project



The screenshot shows the Test Script Project interface. At the top, there are controls for selecting a test browser (Chrome), a base URL, and buttons for 'Start Testscript', 'View Test logs', and 'Add Script'. A checkbox for 'Close browser after test' is checked. Below these are a 'Set Time per script' slider set to 0 and a 'Repeat Count' dropdown set to 0. A search bar is located on the right. The main section is a 'TestCase Table' showing a list of test cases. The first row is highlighted in green and contains the following data: Case Name: login password, Command: type, Target: name=user_name, Value: mark, Date Added: 2019-04-03 00:10:43. To the right of this row are 'Edit' and 'Delete' buttons. At the bottom of the table are pagination controls: First, Previous, 1, Next, Last. A bracket labeled '1' points to the entire table area.

Case Name	Command	Target	Value	Date Added	
login password	type	name=user_name	mark	2019-04-03 00:10:43	Edit Delete

1. **Test Script Description** – It shows the test script information.
2. **Select Test Browser** – This drop-down list shows what browser you will use.
3. **Base URL** - In this field you will input the link of the project that will test.
4. **Start Testscript** – This button is about to start the testing of the script.
5. **View Test Logs** – This button will display what happened to script tested.
6. **Add Script** – This button can add script.
7. **Close Browser After Test** – This checkbox can close automatically the browser after testing.
8. **Set Time Per Script** – It can set the execution time of testing script.
9. **Repeat Count** – It can set how many times you test the script.



10. **Edit**– This button can update the test script.

11. **Delete** – This button can delete the test script.

12. **Search Field** – This field filters the data from the script table.

4.1.5 Adding Test Script

The screenshot shows a web form titled 'Add Script'. It contains the following fields and buttons:

- Case Name :** A text input field (indicated by arrow 1).
- Command :** A dropdown menu with 'type' selected (indicated by arrow 2).
- Target:** A dropdown menu with 'NAME' selected (indicated by arrow 3).
- Value :** A text input field (indicated by arrow 4).
- Save** button (green, indicated by arrow 5).
- BACK** button (red, indicated by arrow 6).

1. **Case Name** – Input here your case name.

2. **Command** – This list has a different command which you can select to perform for your test.

3. **Target** – This list is to identify the type of HTML element that you want to interact.

4. **Value** – Input here your preferred value for the field.

5. **Save** – This button will save your test script in the database.

6. **BACK** – This button will redirect you to the test script project form.



4.1.6 New User Account

The screenshot shows a web form titled "Add User" with a close button (X) in the top right corner. The form contains the following fields and controls, each indicated by a numbered arrow:

- 1. First Name (text input)
- 2. Middle Name (text input)
- 3. Last Name (text input)
- 4. Email (text input)
- 5. User Name (text input)
- 6. Password (text input)
- 7. Re-type Password (text input)
- 8. Select User Level (dropdown menu, currently showing "Admin")
- 9. Save (green button)

1. **First Name** – Input here the user's firstname.
2. **Middle Name** – Input here the user's Middlename.
3. **Last Name** – Input here the user's Lastname.
4. **Email** – Input here the user's E-mail.
5. **Username** – Input here the accounts username.
6. **Password** – Input here the accounts password.
7. **Re-type** – Confirm your previous inputted password.
8. **Userlevel** – Select the account user level.



Appendix B:

System Codes

Login

<?php

```
class login_process{
    public function login($conn){
        if(isset($_POST['btn_submit']))
        {

            if(isset($_POST['user'],$_POST['pass']))
            {

                $user_=mysqli_real_escape_string($conn,$_POST['user']);

                $pass_=mysqli_real_escape_string($conn,hash('sha256',$_POST['pass']))
;

                $stmt = $conn->query("SELECT * FROM wtsk_users
WHERE username='$user_' AND password='$pass_' AND acc_archive_flag=1");

                $get_user = mysqli_fetch_assoc($stmt);
```



```
if($get_user['password'] === $pass_ AND
$get_user['username'] === $user_){

    $_SESSION['user_id'] =
base64_encode($get_user['user_id']);

    if($get_user['user_level'] == 1)
    {
        echo '<script>window.location.replace("admin'.
warp . '" );</script>';
    }

    if($get_user['user_level'] == "client")
    {
        echo '<script>window.location.replace("staff'.
warp . '" );</script>';
    }

    $_SESSION['start'] = time(); // Taking
now logged in time.

    // Ending a session in 30 minutes from
the starting time.

    $_SESSION['expire'] =
$_SESSION['start'] + (30 * 60);
}
else{
```



```
        echo '<script>alert("LOGIN FAILED!");</script>';
    }
}

}

}
}

$login = new login_process;
$login_access = $login-> login($conn);
?>

Connection

<?php
@$conn = new mysqli("localhost","root","","wtsk");
session_start();
if($conn->connect_error){
    echo "Connection Error";
    die;
}

date_default_timezone_set('Asia/manila');
```



?>

<title>WTSK - AUTOMATED TESTCASE TESTING MANAGEMENT
TOOL</title>

<?php

\$uri = \$_SERVER['REQUEST_URI'];

if(\$uri!="wtsk/index.php" AND \$uri!="wtsk/login.php"){

\$now = time(); // Checking the time now when home page starts.

if (\$now > @\$_SESSION['expire']) {

session_destroy();

echo '<center>

</br>

</br>

</br>

</br>

<h1 style="color:#000;font-family:tahoma;">YOUR SESSION HAS
EXPIRED PLEASE RE-LOGIN</h1>

</br>

<button
style="width:100px;height:40px;background-
color:gold;border:none;cursor:pointer;" class="btn btn-large btn-warning">CLICK
HERE</button>

</center>;



```
die;

}

if(isset($_SESSION['start']) AND isset($_SESSION['expire'])){

    $_SESSION['start'] = time(); // Taking now logged in time.

    // Ending a session in 30 minutes from the starting time.

    $_SESSION['expire'] = $_SESSION['start'] + (30 * 60);

}

}

?>
```

Index

```
<?php
```

```
require 'assets/defines.php';
```

```
require divide . 'assets' . warp . 'get_user.php';
```

```
if(!isset($_SESSION['user_id'])){

    header('location:login.php');

}
```

```
if(isset($_SESSION['user_id'])){

    if($select_user['user_level'] == 1){

        echo'<script>window.location.replace("admin/home.php");</script>';

    }

}
```



```
if($select_user['user_level'] == 0){  
    echo'<script>window.location.replace("client/home.php");</script>';  
}  
}
```

```
echo "THERE IS AN ERROR";  
die;  
?>
```

Home

```
<?php  
require '../assets/defines.php';  
require root . 'assets' . 'warp' . 'get_user.php';  
  
if(!isset($_SESSION['user_id'])){  
    echo'<script>window.location.replace("../");</script>';  
}  
  
?>  
<!DOCTYPE html>  
<html lang="en">
```



```
<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<?php
if(isset($_GET['edit'])){
    $edit_id_get = mysqli_real_escape_string($conn,$_GET['edit']);
    echo '<style>body{overflow:hidden;scroll:none;}</style>';
}

require root . 'css_ext.php';

?>

</head>

<body>

<?php
if(isset($_GET['edit'])){
    ?>

        <div id="displayAlerts">

            <hr></hr>

            <a href="home.php"><h1 style="float:right;margin-
right:50px;"><i class="icon icon-remove"></i></h1></a>

            <?php

                @$php_select_project = $conn->query("Select * from
wtsk_project WHERE project_id='$edit_id_get'");
```



```
@ $value_fetch =
mysqli_fetch_assoc($php_select_project);

?>

<div class="widget-content">

    <form class="form-horizontal" method="post" action=""
name="project_form" id="project_form" enctype="multipart/form-data">

        <div class="row-fluid">

            <div class="span2"></div>

                <div class="span8">

                    <div class="span6">

                        <div class="control-group">

                            <label class="control-label">Project
Name</label>

                                <div class="controls">

                                    <input type="text" class="form-control"
name="project_name" id="project_name" value="<?php echo
$value_fetch['project_name'];?>" maxlength="255">

                                        </div>

                                    </div>

                                <div class="control-group">

                                    <label class="control-label">Product
Name</label>

                                        <div class="controls">

                                            <input type="text" name="product_name"
id="product_name" value="<?php echo $value_fetch['product_name'];?>"
maxlength="255">

                                                </div>

                                            </div>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </form>

</div>
```




```
<li> <a href=""><i class="icon icon-flag"></i> <span> Activity Logs</span></a>
</li>

<li> <a href="acc.php"><i class="icon icon-user"></i>
<span>Accounts</span></a> </li>

</ul>
</div>

<!--sidebar-menu-->

<!--main-container-part-->
<div id="content">
<!--breadcrumbs-->
<div id="content-header">

<div id="breadcrumb"> <a href="index.html" title="Go to Home" class="tip-
bottom"><i class="icon-home"></i>Home</a></div>

</div>
<!--End-breadcrumbs-->

<!--Action boxes-->

    <?php
        if(!isset($_GET['add_project'])){
            ?>

<div class="container-fluid">
```




```
$archive1 = $conn->query("UPDATE wtsk_project SET
project_archive_flag='0' WHERE project_id='$_project_id'");

$archive2 = $conn->query("UPDATE wtsk_scripts SET
script_archive_flag='0' WHERE project_id='$_project_id'");

$archive3 = $conn->query("UPDATE wtsk_script_test_logs SET
script_test_log_archive_flag='0' WHERE project_id='$_project_id'");

echo '<script>alert("Project Archive Successfully!");</script>';

echo
'<script>window.location.replace("home.php");</script>';

}

$archive_id_get = mysqli_real_escape_string($conn,$_GET['archive']);

$select_project_archive = $conn->query("SELECT * FROM wtsk_project
WHERE project_id='$archive_id_get' AND project_archive_flag=1")-
>fetch_assoc();

//echo '<style>body{overflow:hidden;scroll:none;}</style>';

echo '<center><div class="alert alert-success"><h3>ARCHIVE THIS PROJECT
['.$select_project_archive['project_name'].']? All of components will be
deleted.</h3></div>

<form action="" method="POST">

<input type="hidden"
value="'.mysqli_real_escape_string($conn,$_GET['archive'])."'
name="archive_id_post"/>

<input type="submit" Value="YES" class="btn btn-large btn-danger"
name="btn_archive_id_post"/>

<a href="home.php" Value="Close" class="btn btn-large btn-default">Close</a>
```



```
</form>

</center>

<br/>

<br/>

<hr/>

<br/>

';

}

?>

        <center><a href="home.php?add_project"><button class="btn btn-large
btn-info">ADD PROJECT</button></center></a>

        <div class="widget-box">

            <div class="widget-title"> <span class="icon"><i class="icon-
th"></i></span>

                <h5></h5>

            </div>

            <div class="widget-content nopadding" style="overflow-x:scroll;">

                <table class="table table-bordered data-table" >

                    <thead>

                        <tr>

                            <th>Project Name</th>

                            <th>Product Name</th>

                            <th>Start & Due date</th>
```



```
<th>Date Created</th>

<th>Form Tested Progress</th>

</tr>

</thead>

<tbody>

        <?php

                $select_project = $conn->query("SELECT * FROM
wtsk_project WHERE project_archive_flag=1");

                While($row = mysqli_fetch_assoc($select_project)){

                        //all tested pass or failed

                        $select_all_project_script_tested = $conn-
>query("SELECT * FROM wtsk_scripts INNER JOIN wtsk_project ON
wtsk_scripts.project_id=wtsk_project.project_id WHERE
wtsk_scripts.script_test_flag='1' AND wtsk_scripts.script_archive_flag='1' AND
wtsk_project.project_id='".$row['project_id']."'");

                        //fetch all project query

                        $select_all_project_script = $conn->query("SELECT *
FROM wtsk_scripts INNER JOIN wtsk_project ON
wtsk_scripts.project_id=wtsk_project.project_id WHERE
wtsk_scripts.script_archive_flag='1' AND
wtsk_project.project_id='".$row['project_id']."'");

                        //fetch all success test

                        $select_project_script_complete = $conn-
>query("SELECT * FROM wtsk_scripts INNER JOIN wtsk_project ON
wtsk_scripts.project_id=wtsk_project.project_id WHERE
```



```
wtsk_scripts.script_status='1' AND wtsk_scripts.script_archive_flag='1' AND  
wtsk_project.project_id='".$row['project_id']."'");
```

```
        $this_project_scripts =  
mysql_num_rows($select_all_project_script);
```

```
        $this_project_done_script =  
mysql_num_rows($select_project_script_complete);
```

```
        $this_noexcepton_tested_fetch =  
mysql_num_rows($select_all_project_script_tested);
```

```
        @$formula =  
$this_project_done_script/$this_project_scripts*100;
```

```
        @$formula2 =  
$this_noexcepton_tested_fetch/$this_project_scripts*100;
```

?>

```
<tr class="gradeX" style="text-align:center;">
```

```
        <td><center><br></br></br></br><?php echo  
$row['project_name'];?></center></td>
```

```
        <td><center><br></br></br></br><?php echo  
$row['product_name'];?></center></td>
```

```
        <td><center><br></br></br></br><?php echo  
$row['due_date'];?><center></td>
```



```
<td><center></br></br></br></br><?php echo
$row['created_at'];?></center></td>

<td class="center">

<?php
//failed formula
$failed_scripts = $this_project_scripts -
$this_project_done_script;

?>

<p style="font-size:12px;">Total Script Count:<?php
echo "[".$this_project_scripts.""];?>, Successful Script: <?php echo
 "[".$this_project_done_script.""];?>, Failed Script:<?php echo
 $failed_scripts;?></p>

Usability rate:

<div class="progress progress-mini active
progress-info">

<div style="width:

<?php
if($this_project_scripts==0){
    echo "0%";
}
else{
    echo $formula."%";
}

?>;color:#333;" class="bar"><span
class="percent">
```




```
<?php
    if($this_project_scripts==0){
        echo "0%";
    }
    else{
        echo round($formula,2)."%";
    }
?></span></div>
</div>

Test Progress:
<div class="progress progress-mini active
progress-info">

    <div style="width:
    <?php
        if($this_project_scripts==0){
            echo "0%";
        }
        else{
            echo $formula2."%";
        }
    ?>;color:#333;" class="bar"><span
class="percent">

    <?php
```



```
        if($this_project_scripts==0){
            echo "0%";
        }
        else{
            echo round($formula2,2)."%";
        }
        ?></span></div>

</div>

<center><a
href="project.php?project_id=<?php echo $row['project_id']; ?>"><button
class="btn btn-mini btn-success"><i class="icon icon-folder-open"></i> Open
Project</button></a>

        <a href="?edit=<?php echo
$row['project_id']; ?>"><button class="btn btn-mini btn-warning" value="<?php
echo $row['project_id']; ?>" id="Edit_btn"><i class="icon icon-edit"></i>
Edit</button></a>

        <a href="?archive=<?php echo
$row['project_id']; ?>"><button class="btn btn-mini btn-danger" value="<?php
echo $row['project_id']; ?>" onClick="archive_id_tigger(this.value)"><i class="icon
icon-trash"></i> Delete</button></a>

        <a
href=" ../documents/viewpdf_.php?file=<?php echo
$row['project_document']; ?>"><button class="btn btn-mini btn-default">VIEW
DOCUMENTS</button></a>

</center>

</td>
```



```
</tr>

                                <?php }?>

</tbody>
</table>
<style>

#displayAlerts{
    opacity:1;
    display:none;
    width:101vw;
    height:100vh;
    position:fixed;
    background-color:#fff;
    top:0px;
    left:0px;
    z-index:555;
    overflow-y:scroll;
}
</style>

</div>
</div>
```




```
<script type="text/javascript">

// This function is called from the pop-up menus to transfer to
// a different page. Ignore if the value returned is a null string:
function goPage (newURL) {

    // if url is empty, skip the menu dividers and reset the menu selection to
    default

    if (newURL != "") {

        // if url is "-", it is this page -- reset the menu:
        if (newURL == "-" ) {
            resetMenu();
        }

        // else, send page to designated URL
        else {
            document.location.href = newURL;
        }
    }
}

// resets the menu selection upon entry to this page:
function resetMenu() {
    document.gomenu.selector.selectedIndex = 2;
```



```
}  
</script>  
  
    <script>  
$("#displayAlerts").slideDown(1000);  
  
        function archive_id_tigger(click_id){  
            //var the_id = click_id;  
  
            //document.getElementById("archive_hidden_id").value=the_id;  
            //alert("sample");  
        }  
  
    <?php  
  
    $project_id_key=mysqli_real_escape_string($conn,$_GET['edit']);  
        if(isset($project_id_key)){  
            try{  
                if($project_id_key<1){  
                    echo  
'<script>window.location.replace("index.php");</script>';  
                }  
            }catch(Exception $e){  
                echo  
'<script>window.location.replace("index.php");</script>';  
            }  
        }  
    ?>
```



```
<?php }?>
```

```
</script>
```

```
</body>
```

```
</html>
```

Project

```
<?php
```

```
require '../assets/defines.php';
```

```
require root . 'assets' . 'warp' . 'get_user.php';
```

```
$secure_project_id = mysqli_real_escape_string($conn,$_GET['project_id']);
```

```
if(isset($_POST['btn_save_script'])){
```

```
    //$baseUrl_ = mysqli_real_escape_string($conn,$_POST['baseUrl_']);
```

```
    $caseNAME_ = mysqli_real_escape_string($conn,$_POST['caseName']);
```

```
    $command_ = mysqli_real_escape_string($conn,$_POST['command']);
```

```
    $target_ = mysqli_real_escape_string($conn,$_POST['target']);
```

```
    $value_ = mysqli_real_escape_string($conn,$_POST['value']);
```

```
    $save_script=$conn->query("INSERT INTO  
wtsk_scripts(project_id,case_name,command,target,value) VALUES  
('$secure_project_id','$caseNAME_','$command_','$target_','$value_')") or  
die(mysqli_error($conn));
```

```
    if($save_script==true){
```

```
        echo'<script>alert("Successfully Added");</script>';
```



```
}

if($save_script==false){
    echo'<script>alert("Error!");</script>';
}
}

if(!isset($_SESSION['user_id'])){
    echo'<script>window.location.replace("../");</script>';
}

if(isset($_GET['project_id'])){
    $select_project = $conn->query("SELECT * FROM wtsk_project WHERE
project_id='$secure_project_id'");
    $select_project_row=mysqli_fetch_assoc($select_project);
}

if(!isset($_GET['project_id'])){
    echo 'Error!';
    die;
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
```




```
<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<?php

require root . 'css_ext.php';

?>

<?php
require root . 'js_ext.php';

?>

<style>
#viewScriptLogs{
    opacity:1;
    display:none;
    width:100vw;
    height:100vh;
    position:fixed;
    background-color:rgba(0,0,0,0.5);
    top:0px;
    left:0px;
    z-index:1;
    overflow-y:scroll;
}

</style>
```



```
</head>

<body style="">

  <!-- script test log table -->

  <div id="viewScriptLogs">

    <div class="modal-dialog" role="document">

      <div class="modal-content">

        <div class="container-fluid" style="background-color:;>

          <div class="row-fluid">

            <div class="span3"></div>

            <div class="span6">

              <div class="widget-box">

                <div class="widget-title"> <span class="icon"><i class="icon-th"></i></span>

                <h5>TestCase Table</h5>

                <button type="button" id="testlogCLose" class="close" data-dismiss="modal" aria-label="Close">

                  <span aria-hidden="true" style="margin-right:10px;">&times;</span>

                </button>

              </div>

              <div class="widget-content nopadding">

                <table class="table table-bordered test_script_logs_tbl">

                  <thead>

                    <tr>
```



```

<th>Executed by</th>

<th>Date</th>

<th></th>

</tr>

</thead>

<tbody>

<?php
    $select_script = $conn->query("SELECT *
FROM wtsk_scripts_test_logs INNER JOIN wtsk_users ON wtsk_users.user_id =
wtsk_scripts_test_logs.executed_by WHERE
wtsk_scripts_test_logs.project_id='$secure_project_id' AND
wtsk_scripts_test_logs.script_test_log_archive_flag=1");

    while($fetch_script =
mysqli_fetch_assoc($select_script)){

        ?>

        <tr class="gradeX" style="text-
align:center;<?php

            if($fetch_script['script_status']==1 AND
$fetch_script['script_test_flag']=1){

                echo"background-
color:#96fcc1;color:#000;";

            }

            if($fetch_script['script_status']==11 AND
$fetch_script['script_test_flag']=1){

                echo"background-
color:#f29760;color:#fff;";

            }

```



```
?>">

        <td><center><?php echo
$fetch_script['lname']. " ".$fetch_script['mname'].
".$fetch_script['fname'];?></center></td>

        <td><center><?php echo
$fetch_script['created_at'];?></center></td>

        <td style="text-align:center;"><a
href="view_script_log.php?script_log_id=<?php echo
$fetch_script['log_id'];?>&project_id=<?php echo
$secure_project_id;?>&access_file"><button class="btn btn-min btn-primary"><i
class="icon icon-folder-open"></i> OPEN</button></a></td>

        </tr>

        <?php }?>

</tbody>
</table>

</div>

</div>

</div>

<div class="span3"></div>

</div>

</div>

</div>

</div>

</div>

</div>
```



```
<div class="container-fluid" style="background-color:#fff;">
    <div class="span12" style="padding:1%;" style="padding-
left:10%;">
        <a href="home.php"><button type="button" class="close"
style="padding-left:10%;">
            <span aria-hidden="true"><h4>close
&times;</h4></span>
        </button></a>
    </div>
    <div class="row-fluid">
        <div class="span12">
            <div class="widget-box">
                <div class="widget-title"> <span class="icon"> <i class="icon-file"></i>
</span>
                <h5>Log</h5>
            </div>
            <div class="widget-content">
                <div id="" style="background-
color:#000;width:98%;height:auto;padding:1%;color:#fff;overflow-y:scroll;">
                    <?php
                        require 'script_exec.php';
                    ?>
                </div>
            </div>
        </div>
    </div>
```




```
<input type="text" style="border:none;border-color:#fff;"
id="range_display" value="0"></input>

<br>

<br>

<div class="row-fluid" style="text-align:center;">
<div class="widget-box" id="widget-box_table">
  <div class="widget-title"> <span class="icon"><i class="icon-
th"></i></span>
    <h5>TestCase Table</h5>
  </div>
  <div class="widget-content nopadding">

    <table class="table table-bordered data-table">

      <thead>
        <tr>
          <th style=""><center><input type="checkbox"
id="allcheck" name="allcheck"/></center></th>

          <th>Case Name</th>
          <th>Command</th>
          <th>Target</th>
          <th>Value</th>
          <th>Date Added</th>
          <th></th>
```




```
</tr>

</thead>

<tbody>

        <?php

                $select_script = $conn->query("SELECT *
FROM wtsk_scripts WHERE project_id='$secure_project_id' AND
script_archive_flag='1' ");

                while($fetch_script =
mysqli_fetch_assoc($select_script)){

                        ?>

                        <tr class="gradeX" style="text-
align:center;<?php

                                if($fetch_script['script_status']==1){

                                        echo"background-
color:#96fcc1;color:#000;";

                                }

                                if($fetch_script['script_status']==11){

                                        echo"background-
color:#f29760;color:#fff;";

                                }

                        ?>">

                        <td><center><input type="checkbox"
id="testcase[]" name="testcase[]" value="<?php echo $fetch_script['script_id'];?>"
/></center></td>
```



```
<td><?php echo  
$fetch_script['case_name'];?></td>  
  
<td><?php echo  
$fetch_script['command'];?></td>  
  
<td><?php echo $fetch_script['target'];?></td>  
  
<td><?php echo $fetch_script['value'];?></td>  
  
<td><?php echo  
$fetch_script['created_at'];?></td>  
  
<td>  
  
<button class="btn btn-mini btn-danger"><i  
class="icon icon-trash"></i> Delete</button>  
  
<button class="btn btn-mini btn-warning"><i  
class="icon icon-edit"></i> Edit</button>  
  
</td>  
  
</tr>  
  
<?php }?>  
  
</tbody>  
  
</table>  
  
</div>  
  
</div>  
  
</div>  
  
</form>  
  
<div class="span3">
```



```
<div class="widget-box">

    <div class="widget-title"> <span class="icon"> <i class="icon-list"></i>
</span>

    <h5>Project Information</h5>

</div>

<div class="widget-content">

        <h6>Script Count: <?php echo
mysqli_num_rows($select_script);?></h6>

        <h6>Product name: <?php echo
$select_project_row['product_name'];?></h6>

        <h6>Product Version: <?php echo
$select_project_row['product_name'];?></h6>

        <h6>Due Date: <?php echo
$select_project_row['due_date'];?></h6>

        <h6>Project Document: <a
href="../../documents/viewpdf_.php?file=<?php echo
$select_project_row['project_document'];?>"><button class="btn btn-mini btn-
default">View Documents</button></a>

        <h6>Project Description:</h6><textarea
class="span12" readonly> <?php echo
$select_project_row['project_desc'];?></textarea>

        <h6>Created at: <?php echo
$select_project_row['created_at'];?></h6>

        <h6>

                <button class="btn btn-mini btn-default"><i
class="icon icon-edit"></i> Edit</button>

                <button class="btn btn-mini btn-default"><i
class="icon icon-trash"></i> Delete</button>
```



</div>

</div>

</div>

<!--end-Footer-part-->

```
<script type="text/javascript">
```

```
// This function is called from the pop-up menus to transfer to
```

```
// a different page. Ignore if the value returned is a null string:
```

```
function goPage (newURL) {
```

```
// if url is empty, skip the menu dividers and reset the menu selection to
default
```

```
if (newURL != "") {
```



```
// if url is "-", it is this page -- reset the menu:
if (newURL == "-" ) {
    resetMenu();
}

// else, send page to designated URL
else {
    document.location.href = newURL;
}
}
}

// resets the menu selection upon entry to this page:
function resetMenu() {
    document.gomenu.selector.selectedIndex = 2;
}
</script>

<script>

    var test_submit=
document.getElementById('allcheck');

    var testcase = document.getElementById('testcase[]');

    if(test_submit.checked){
        testcase.checked;
```



```
    }

    //$.ajax({
    //    url:,
    //    method:,
    //    dataType:,
    //    success:
    //});

    $("#viewtestLogs").click(function(){
        $("#viewScriptLogs").fadeIn();
    });

    $("#testlogCloses").click(function(){
        $("#viewScriptLogs").fadeOut();
    });

    $("#click_test_res").click(function(){
        window.open("https://www.w3schools.com");
    });

    setInterval(function(){
        $('#tr_ajax').load('project.php #tr_ajax').unwrap();
    },1000);

    function rangevalue(value_of_range){

    document.getElementById('range_display').value=value_of_range;
```



```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Script

```
<?php
```

```
require_once "../selenium/phpwebdriver/WebDriver.php";
```

```
$webdriver = new WebDriver("localhost", "4444");
```

```
$webdriver->connect("chrome");
```

```
$webdriver->get("http://localhost/wais");
```

```
$selement1 = $webdriver->findElementBy(LocatorStrategy::name,"user_name")-  
>setImplicitWaitTimeout(10,1000)->until(
```



```
WebDriverExpectedCondition::visibilityOfElementLocated($element1-
>findElement(WebDriverBy::name('user_name')))
);
// $element2 = $webdriver->findElementBy(LocatorStrategy::name,"pass_word");

$element1->getKeyboard()->sendKeys('James');

//if($element4!=0){
//    $element_ = $webdriver-
>findElementBy(LocatorStrategy::id,"messagesDropdown");
//    $element_->click();
//}
//echo "found";

//if(sizeof($element) != 0 ){
//    if ($element) {
//        $element = $webdriver-
>findElementBy(LocatorStrategy::name,"user_name");
```




```
//      $element->sendKeys(array("php webdriver"));
//      echo "found";
//      }
//}
//
//
//if(sizeof($element) == 0 ){
//      echo "not found";
//      die;
//}

//if ($element){
//      $element->sendKeys(array("php webdriver" ) );
//      // $element->submit();
//}
//$webdriver->close();

?>
```

Add Script

```
<?php
```

```
require '../assets/defines.php';
```



```
require root . 'assets' . warp . 'get_user.php';

?>

<!DOCTYPE HTML>

<html>

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<?php
require root . 'css_ext.php';
?>
<style>
td{
    text-align:center;
    border:solid 1px #ffffb2;
}
tr{
    font-weight:bolder;
    text-align:center;
    border:solid 1px #ffffb2;
}
</style>
```



</head>

<body>

```
<div class="container-fluid" style="background-color:">
```

```
<div class="row-fluid">
```

```
<div class="span3"></div>
```

```
<div class="span6">
```

```
<div class="widget-box">
```

```
<div class="widget-title"> <span class="icon"><i  
class="icon-plus"></i> Add Script</span>
```

```
</div>
```

```
<div class="widget-content nopadding">
```

```
<form action="project.php?project_id=<?php echo  
$_GET['project_id'];?>" method="POST" class="form-horizontal" id="add_script">
```

```
<div class="control-group">
```

```
<label class="control-label">Case Name :</label>
```

```
<div class="controls">
```

```
<input type="text" class="span11" name="caseName" id="caseName"
```

```
/>
```

```
</div>
```

```
</div>
```




</body>

</html>

Script Execution

<?php

```
if(isset($_POST['submit_testscript']))
```

```
{
```

```
    if(!isset($_POST['testcase'])){
```

```
        echo '<script>alert("Select Script!");</script>';
```

```
        echo
```

```
'<script>window.location.replace("project.php?project_id=".$secure_project_id.");</script>';
```

```
        die;
```

```
    }
```

```
    function openSeleniumServer($cmd){
```

```
        popen($cmd,"r");
```

```
    }
```

```
    openSeleniumServer('start cmd.exe @cmd /k"java -jar selenium-server-standalone-3.4.0.jar"');
```

```
    //secured script_id
```



```
$consoleMsg = array();

$BASEURL_POSTED=mysqli_real_escape_string($conn,$_POST['baseU
rl_']);

$POSTWEBBROWSER =
mysqli_real_escape_string($conn,$_POST['select_browser']);

require_once "../selenium/phpwebdriver/WebDriver.php";

$webdriver = new WebDriver("localhost", "4444");

$webdriver->connect($POSTWEBBROWSER);

$webdriver->get($BASEURL_POSTED);

array_push($consoleMsg,"Speed
Set:".$_POST['script_exec_time'].".</br>");

//$webdriver->setSpeed(12000000000000000);

//$webdriver->setSpeed(3000);


function
AddTestlog($script_id,$conn,$console,$secure_project_id,$id){

    $user_id__=mysqli_real_escape_string($conn,$id);

    $the_id = mysqli_real_escape_string($conn,$script_id);

    $sql_insert_script_logs = $conn->query("INSERT INTO
wtsk_scripts_test_logs(
```



```
script_id,  
project_id,  
result,  
executed_by  
)  
VALUES(  
'$the_id',  
'$secure_project_id',  
'$console',  
'$user_id__'  
)");  
  
}  
  
function updateStatus($script_id,$conn,$status_input){  
    $the_id = mysqli_real_escape_string($conn,$script_id);  
    $sql_update_status = $conn->query("UPDATE wtsk_scripts  
SET script_status='$status_input',script_test_flag='1' WHERE  
script_id='$the_id'");  
}  
  
foreach($_POST['testcase'] as $selected_id)  
{
```




```
$scripID_=mysqli_real_escape_string($conn,$selected_id);

$select_script = $conn->query("SELECT * from wtsk_scripts
WHERE script_id='$scripID_'");

$row_select = mysqli_fetch_assoc($select_script);

$EXPLODEDtarget = explode('=', $row_select['target']);

//TARGET VALIDITY

if($EXPLODEDtarget[0]!="name" AND $EXPLODEDtarget[0]!="id"
AND $EXPLODEDtarget[0]!="xpath")

{

    updateStatus($row_select['script_id'],$conn,$status_input=11);

    array_push($consoleMsg,"Invalid Target
name:". $row_select['target']. "<br>");

}

require 'locator.php';

}

$count = 1;

$console="";

foreach($consoleMsg as $msg){

    @$console .= "INFO 000".$count++." :: ".$msg."";

}

echo $console;

//adding test log
```



```
AddTestlog($row_select['script_id'],$conn,$console,$secure_project_id,$i
d);

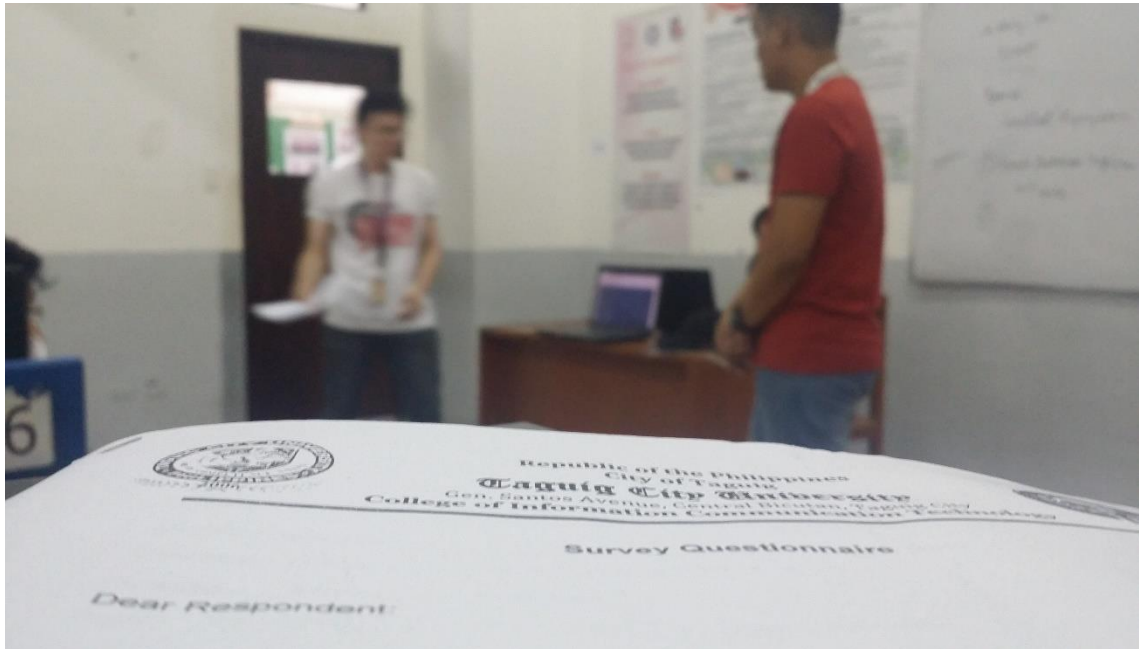
if(isset($_POST['close_window'])){
    $webdriver->close();
}
echo "Test Done";
}
?>

<?php
    if(isset($_POST['close_window_cmd'])){
        shell_exec('taskkill /IM cmd.exe');
        shell_exec('taskkill /IM conhost.exe');
        echo "value:".$_POST['close_window_cmd'];
    }
?>
```



Appendix C:

Picture taken during the survey







Appendix D:

Grammarian Certificate

This is to certify that the undersigned grammarian has reviewed and went through all the pages of the proposed research entitled “**Web Testing Startup Kit – Automated Test Case Testing Management Tool**” developed by Kasim Rasdi I., Roldan, Mark Jervin G., Uhuad, Hiedy G. and Valles, Mark V. is aligned with the set of structural rules that govern the composition of sentences, phrases, and words in the English language. Also, all corrections and recommendations made have been done and/or incorporated in the final document.

Signed By:

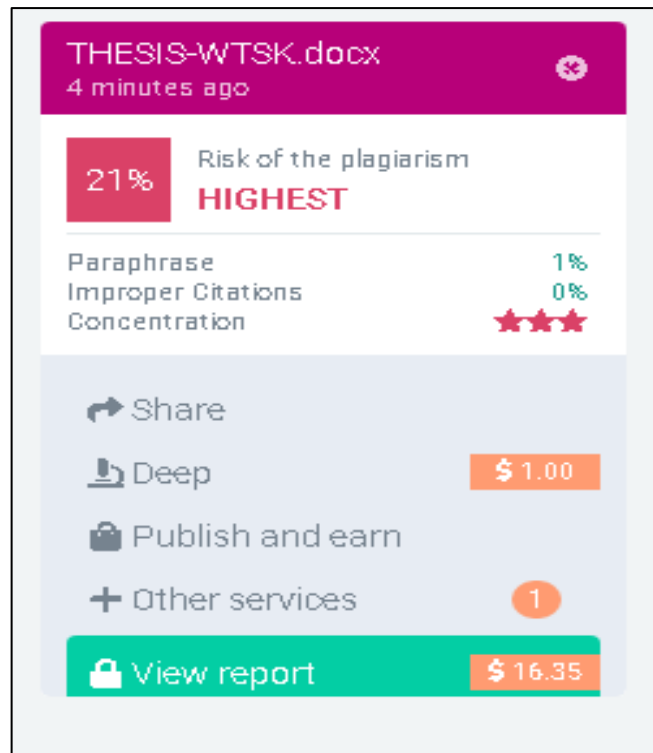
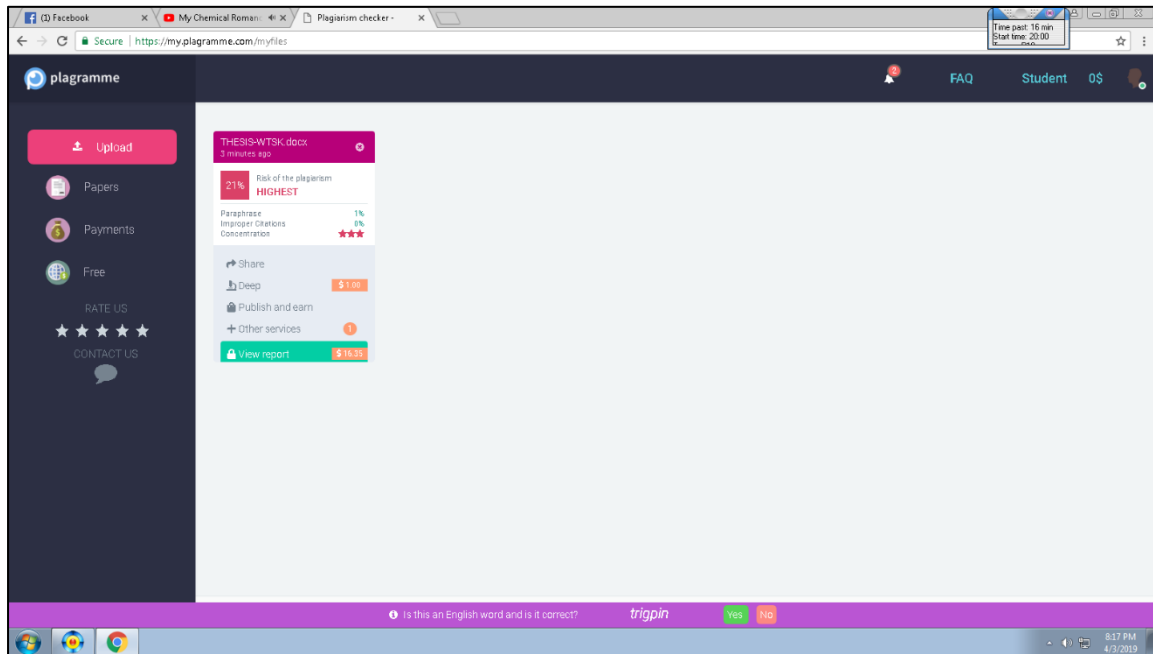
Prof. Marilou C. Mendoza, MA-ENG

Thesis Grammarian



Appendix E:

Plagiarism Screenshot





Appendix F:

Pre-survey Questionnaire

Name (optional): _____ **Date:** _____

Course: IT/CS ☐ Non – IT/CS ☐

Gender: Male ☐ Female ☐

Age: (15 – 20) ☐ (21 – 25) ☐ (26 – Above) ☐

Please complete the following questionnaire with specific regard to the above inquiry, by placing a CHECK ☒ in the appropriate box

| How much do you agree with the following statements | strongly agree | agree | uncertain/
not applicable | disagree | strongly disagree |
|---|--------------------------|--------------------------|------------------------------|--------------------------|--------------------------|
| 1. Automated software testing is faster in execution rather than manual testing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Automated software testing is cheaper compared to manual testing in a long run | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Automated software testing is more reliable compared to the traditional testing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Automated software testing is more detailed in terms of producing reports. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Automated software testing can handle load & performance testing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. Automated software testing is more convenient rather than traditional test case testing. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. Automated software testing can eliminate human error | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. Automated software testing enables us to deliver high-quality software | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. Automated software testing is more productive compared to manual testing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. Automated software testing can overcome the limitations of manual testing | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |