

ISO 9001:2008 Certified Institute

**JAVA INSTITUTE FOR ADVANCED TECHNOLOGY**

Department of Examinations



|  |  |
| --- | --- |
| COURSE(S) – (LEADING TO) | PROFESSIONAL HIGHER DIPLOMA IN SOFTWARE ENGINEERING |
| ASSIGNMENT STARTING DATE | 05th May 2020 |
| ASSIGNMENT CLOSING DATE | 12th May 2020 |
| UNIT NAME | SOFTWARE ENGINEERING II (SOFTWARE TESTING, QUALITY ASSURANCE AND MAINTENANCE) |
| UNIT ID | HF2W 04 |
| ASSIGNMENT ID | HF2W 04/AS/05 |
| DESCRIPTION | Test case (Individual Assignment) |
| DURATION | 1 WEEKS |

**GUIDE LINES FOR CANDIDATES**

Student should describe Testing is a process rather than a single activity. This process starts from test planning then designing test cases, preparing for execution and evaluating status till the test closure through the assignment.

|  |  |
| --- | --- |
| NAME : | KANDAGE DON ISHAN VIHANGA VIMUKTHI  .................................................................................................................................... (BLOCK CAPITALS) |
| NIC : | 960263812v  .................................................................................................................................... |
| SCN NO : | 177646148  .................................................................................................................................... |

Java Institute for Advanced Technology Sri Lanka

Q1: Design test case for: ERP (Enterprise resource planning***)***, Traffic controller and university management system.

The main objective of the test is to find defects in the application or system. To achieve this test cases should be written well and should have the below details:

1. Test case number
2. Test case name
3. Test case description
4. Pre-conditions
5. Test data/Input data
6. Step name
7. Step description/action
8. Expected result 54 marks (18 \*3)

Test Case For ERP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Name | Test Case Description | Step Name | Step Description/  Action | Expected Result |
| 1.Approve the  Document by Manager | Document is created by an employee and is sent to manager for approval, manager approves the document  Pre-condition : Document is created by employee and sent for manager approval  Test Data : Employee and Manager login ID, Password, Document to be approved | Step 1 | Login to the system as manager | Manager login should be successful |
|  |  | Step 2 | Navigate to the document approval screen | Manager document approval screen should display |
|  |  | Step 3 | Select the document from the user | Document should be selected for the approval |
|  |  | Step 4 | Approve the document | Document should be approved successfully |
|  |  | Step 5 | Verify that document status is updated as approved | Document status should be approved |
|  |  | Step 6 | Login the system as employee | Employee login should successful |
|  |  | Step 7 | Navigate to the employee document link | Employee document link should display |
|  |  | Step 8 | Verify that document is approved and status is updated | Document status should be approved |

Test case for Traffic Controller

1. There should be three signal light (Red,Yellow,Green) also there should be provision to have a left and right arrow signal.
2. At a time only one stright and aadjacent left direction should have green signal on. 3- There should be duration of 5 second after completation of Green signal.
3. At the time of Green signal the Zebra cross signal should be red.
4. There should a provision when all the signal should be closed and only zebra crossing is allowed. 6- To make sure the visibility of the signal the height of the signal pole should be grater than 10 feet.

7- There should be provision to allow maual control if it is not a digitally controled. 8- Each and every signal should have time meter in the decreasing order.

1. There should be power backup provided to the signal.
2. The intensity of the signal light should be brighter and should not be dull.

Test case for university management system

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name** | **Test Case Description** | **Step Name** | **Step Description** | **Expected Result** |
| **1. Adding**  **Student** | Student takes  admission and  register with  his/her roll number | Step 1 | Login to the system as new student | A new  registration forms appears |
|  |  | Step 2 | Student provides his/her roll  number | A valid roll number must be  entered |
|  |  | Step 3 | Student provides his personal details like  mobile, email, address etc | Details must be provided in correct format |
|  |  | Step 4 | Student clicks on register button | If all details are valid and in correct format student registration is successful and student Is  redirected to student-  homepage |

Q2. Explain various types of testing in detail. 46 marks (2\*23) (You should provide brief explain about each types and It should contain at least 23 types)

**Ad-hoc testing** : This type of software testing is very informal and unstructured and can be performed by any stakeholder with no reference to any test case or test design documents. The person performing Ad-hoc testing has a good understanding of the domain and workflows of the application to try to find defects and break the software. Ad-hoc testing is intended to find defects that were not found by existing test cases.

**Acceptance Testing** : Acceptance testing is a formal type of software testing that is performed by end user when the features have been delivered by developers.

**The aim of this testing** is to check if the software confirms to their business needs and to the requirements provided earlier. Acceptance tests are normally documented at the beginning of the sprint (in agile) and is a means for testers and developers to work towards a common understanding and shared business domain knowledge.

**Accessibility Testing** : When doing accessibility testing, the aim of the testing is to determine if the contents of the website can be easily accessed by disable people. Various checks such as color and contrast (for color blind people), font size for visually impaired, clear and concise text that is easy to read and understand.

**Agile Testing** : Agile Testing is a type of software testing that accommodates agile software development approach and practices. In an Agile development environment,testing is an integral part of software development and is done along with coding. Agile testing allows incremental and iterative coding and testing.

**API Testing** : API testing is a type of testing that is similar to unit testing. Each of the Software APIs are tested as per API specification. API testing is mostly done by testing team unless APIs to be tested or complex and needs extensive coding. API testing requires understanding both API functionality and possessing good coding skills.

**Automated testing** : This is a testing approach that makes use of testing tools and/or programming to run the test cases using software or custom developed test utilities. Most of the automated tools provided capture and playback facility, however, there are tools that require writing extensive scripting or programming to automate test cases.

**All Pairs testing** : Also known as Pair-wise testing, is a black box testing approach and a testing method where for each input is tested in pairs of inputs, which helps to test software works as expected with all possible input combinations.

**Beta Testing** : This is a formal type of software testing that is carried out by end customers before releasing or handing over software to end users. Successful completion of Beta testing means customer acceptance of the software.

**Black Box testing** : Black box testing is a software testing method where testers are not required to know coding or internal structure of the software. Black box testing method relies on testing software with various inputs and validating results against expected output.

**Backward Compatibility Testing** Type of software testing performed to check that the newer version of the software can work successfully on top of the previous version of the software and that the newer version of the software works as fine with table structure, data structures and files that were created by the previous version of the software.

Boundary Value Testing (BVT) : Boundary Value Testing is a testing technique that is based on concept “error aggregates at boundaries”. In this testing technique, testing is done extensively to check for defects at boundary conditions. If a field accepts value 1 to 100 then testing is done for values 0, 1, 2, 99, 100 and 101.

**Big Bang Integration testing** : This is one of the integration testing approaches, in Big Bang integration testing all or all most all of the modules are developed and then coupled together.

**Bottom up Integration testing** Bottom-up integration testing is an integration testing approach where testing starts with smaller pieces or sub-systems of the software till all the way up covering entire software system. Bottom-up integration testing begins with small portions of the software and eventually scale up in terms of size, complexity, and completeness.

**Branch Testing** Is a white box testing method for designing test cases to test code for every branching condition. Branch testing method is applied during unit testing.

**Browser compatibility Testing** Its one of the sub-types of testing of compatibility testing performed by the testing team. Browser compatibility testing is performed for web applications with combinations of different browsers and operating systems.

**Compatibility testing** : Compatibility testing is one of the test types performed by the testing team. Compatibility testing checks if the software can be run on different hardware, operating system, bandwidth, databases, web servers, application servers, hardware peripherals, emulators, different configuration, processor, different browsers and different versions of the browsers etc.,

**Component Testing** : This type of software testing is performed by developers. Component testing is carried out after completing unit testing. Component testing involves testing a group of units as code together as a whole rather than testing individual functions, methods.

**Condition Coverage Testing** : Condition coverage testing is a testing technique used during unit testing, where developer tests for all the condition statements like if, if-else, case etc., in the code being unit tested.

**Dynamic Testing** : Testing can be performed as Static Testing and Dynamic testing, Dynamic testing is a testing approach where testing can be done only by executing code or software are classified as Dynamic Testing. Unit testing, Functional testing, regression testing, performance testing etc.,

Decision Coverage Testing : Is a testing technique that is used in Unit testing. The objective of decision coverage testing is to exercise and validate each decision block in the code e.g. if, if-else, case statements.

**End-to-end Testing** : End to end testing is performed by the testing team and the focus is to test end to end flows e.g. right from order creation till reporting or order creation till item return etc and checking. End to end testing is usually focused on mimicking real life scenarios and usage. End to end testing involves testing information flow across applications.

**Exploratory Testing** : Exploratory testing is an informal type of testing conducted to learn the software at the same time looking for errors or application behavior that seems non-obvious. Exploratory testing is usually done by testers but can be done by other stakeholders as well like Business Analysts, developers, end users etc. who are interested in learning functions of the software and at the same time looking for errors or behavior is seems non-obvious.

**Equivalence Partitioning** : Equivalence partitioning is also known as Equivalence Class Partitioning is a software testing technique and not a type of testing by itself. Equivalence partitioning technique is used in black box and gray box testing types. Equivalence partitioning classifies test data into Equivalence classes as positive Equivalence classes and negative Equivalence classes, such classification ensures both positive and negative conditions are tested.

**Functional testing** is a formal type of testing performed by testers. Functional testing focuses on testing software against design document, Use cases, and requirements document. Functional testing is a type of testing and does not require internal working of the software, unlike white box testing.

Fuzz Testing : Fuzz testing or fuzzing is a software testing technique that involves testing with unexpected or random inputs. The Software is monitored for failures or error messages that are presented due to the input errors.

**GUI (Graphical User Interface) testing** : This type of software testing is aimed at testing the software GUI (Graphical User Interface) of the software meets the requirements as mentioned in the GUI mockups and Detailed designed documents. For e.g. checking the length and capacity of the input fields provided on the form, type of input field provided, e.g. some of the form fields can be displayed as drop-down box or a set of radio buttons. So GUI testing ensures GUI elements of the software are as per approved GUI mockups, detailed design documents, and functional requirements. Most of the functional test automation tools work on GUI capture and playback capabilities. This makes script recording faster at the same time increases the effort on script maintenance.

**Glass box testing** is another name for White box testing. Glass box testing is a testing method that involves testing individual statements, functions etc., Unit testing is one of the Glass box testing methods.

**Gorilla Testing** : This type of software testing is done by software testing team, has a scary name though ?. Objective of Gorilla Testing is to exercise one or few functionality thoroughly or exhaustively by having multiple people test the same functionality.

**Happy path testing** : Also known as Golden path testing, this type of testing focuses on successful execution of tests that do not exercise the software for negative or error conditions.

**Integration testing** is one of the most common and important types of software testing. Once the individual units or components are tested by developers as working then testing team will run tests that will test the connectivity among these units/component or multiple units/components. There are different approaches for Integration testing namely, Top-down integration testing, Bottom-up integration testing and a combination of these two known as Sand witch testing.

**Interface Testing Interface** : Testing is needed when a software provides support for one or more interfaces like “Graphical user interface”, “Command Line Interface” or “Application programming interface” to interact with its users or other software. Interfaces serve as the medium for software to accept input from a user and provide an output to the user. Approach for interface testing depends on the type of the interface being testing like GUI or API or CLI.

**Internationalization testing** is a type of testing that is performed by software testing team to check the extent to which software can support Internationalization i.e., usage of different languages, different character sets, double byte characters etc., For e.g.: Gmail, is a web application that is used by people all over work with different languages, single by or multi-byte character sets.

**Keyword-driven Testing** : Keyword driven testing is more of an automated software testing approach than a type of testing itself. Keyword driven testing is known as action-driven testing or table-driven testing.

**Load Testing** is a type of non-functional testing; load testing is done to check the behavior of the software under normal and over peak load conditions. Load testing is usually performed using automated testing tools. Load testing intends to find bottlenecks or issues that prevent software from performing as intended at its peak workloads.

**Localization Testing** a type of software testing performed by software testers, in this type of testing, software is expected to adapt to a particular locale, it should support a particular locale/language in terms of display, accepting input in that particular locale, display, font, date time, currency etc., related to a particular locale. For e.g. many web applications allow choice of locale like English, French, German or Japanese. So once locale is defined or set in the configuration of software, software is expected to work as expected with a set language/locale.

**Negative Testing** : This type of software testing approach, which calls out the “attitude to break”, these are functional and non-functional tests that are intended to break the software by entering incorrect data like incorrect date, time or string or upload binary file when text files supposed to be upload or enter huge text string for input fields etc. It is also a positive test for an error condition.

**Non-functional testing** : Most Softwares are built to fulfill functional and non-functional requirements, non-functional requirements like performance, usability, localization etc., There are many types of testing like compatibility testing, compliance testing, localization testing, usability testing, volume testing etc., that are carried out for checking non-functional requirements.

**Pair Testing** is a software testing technique that can be done by software testers, developers or Business analysts. As the name suggests, two people are paired together, one to test and other to monitor and record test results. Pair testing can also be performed in combination of tester-developer, tester-business analyst or developer-business analyst combination. Combining testers and developers in pair testing helps to detect defects faster, identify the root cause, fix and test the fix.

**Performance Testing** is a type of software testing and part of performance engineering that is performed to check some of the quality attributes of software like Stability, reliability, availability. Performance testing is carried out by performance engineering team. Unlike Functional testing, Performance testing is done to check non-functional requirements. Performance testing checks how well the software works in anticipated and peak workloads. There are different variations or sub types of performance like load testing, stress testing, volume testing, soak testing and configuration testing.

**Penetration Testing** is a type of security testing. Penetration testing is done to tests how secure software and its environments (Hardware, Operating system, and network) are when subject to attack by an external or internal intruder. An intruder can be a human/hacker or malicious programs. Pentest uses methods to forcibly intrude (by brute force attack) or by using a weakness (vulnerability) to gain access to a software or data or hardware with an intent to expose ways to steal, manipulate or corrupt data, software files or configuration. Penetration Testing is a way of ethical hacking, an experienced Penetration tester will use the same methods and tools that a hacker would use but the intention of Penetration tester is to identify vulnerability and get them fixed before a real hacker or malicious program exploits it.

**Regression Testing** is a type of software testing that is carried out by software testers as functional regression tests and developers as Unit regression tests. The objective of regression tests is to find defects that got introduced to defect fix(es) or introduction of new feature(s). Regression tests are ideal candidates for automation.

**Retesting** is a type of retesting that is carried out by software testers as a part of defect fix verification. For e.g. a tester is verifying a defect fix and let us say that there are 3 test cases failed due to this defect. Once a tester verifies the defect fix as resolved, the tester will then retest or test the same functionality again by executing the test cases that were failed earlier.

**Risk-based Testing** is a type of software testing and a different approach towards testing a software. In Risk-based testing, requirements and functionality of a software to be tested are prioritized as Critical, High, Medium and low. In this approach, all critical and High priority tests are tested and them followed by Medium. Low priority or low-risk functionality are tested at the end or may not be tested at all, depending on timescales.

**Smoke testing** is a type of testing that is carried out by software testers to check if the new build provided by the development team is stable enough i.e., major functionality is working as expected in order to carry out further or detailed testing. Smoke testing is intended to find “show stopper” defects that can prevent testers from testing the application in detail. Smoke testing carried out for a build is also known as build verification test.

**Security Testing** is a type of software testing carried out by a specialized team of software testers. The objective of security testing is to secure the software is to external or internal threats from humans and malicious programs. Security testing basically checks, how good is software’s authorization mechanism, how strong is authentication, how software maintains confidentiality of the data, how does the software maintain integrity of the data, what is the availability of the software in an event of an attack on the software by hackers and malicious programs is for Security testing requires good knowledge of application, technology, networking, security testing tools. With increased number of web applications, security testing has become more important than ever.

**Sanity Testing** is a type of testing that is carried out mostly by testers and in some projects by developers as well. Sanity testing is a quick evaluation of the software, environment, network, external systems are up & running, software environment as a whole is stable enough to proceed with extensive testing. Sanity tests are narrow and most of the time sanity tests are not documented.

**Scalability Testing** is a non-functional test intended to test one of the software quality attributes i.e. “Scalability”. Scalability test is not focused on just one or few functionality of the software instead performance of software as a whole. Scalability testing is usually done by performance engineering team. Objective of scalability testing is to test the ability of the software to scale up with increased users, increased transactions, increase in database size etc., It is not necessary that software’s performance increases with increase in hardware configuration, scalability tests helps to find out how much more workload the software can support with expanding user base, transactions, data storage etc.,

**Stability Testing** is a non-functional test intended to test one of the software quality attributes i.e. “Stability”. Stability testing focuses on testing how stable software is when it is subject to loads at acceptable levels, peak loads, loads generated in spikes, with more volumes of data to be processed. Scalability testing will involve performing different types of performance tests like load testing, stress testing, spike testing, soak testing, spike testing etc.

**Static Testing** is a form of testing where in approaches like reviews, walkthroughs are employed to evaluate the correctness of the deliverable. In static testing software code is not executed instead it is reviewed for syntax, commenting, naming convention, size of the functions/methods etc. Static testing usually has check lists against which deliverables are evaluated. Static testing can be applied for requirements, designs, test cases by using approaches like reviews or walkthroughs.

**Stress Testing** is a type of performance testing, in which software is subjected to peak loads and even to a break point to observe how the software would behave at breakpoint. Stress testing also tests the behavior of the software with insufficient resources like CPU, Memory, Network bandwidth, Disk space etc. Stress testing enables to check some of the quality attributes like robustness and reliability.

**System Testing** this includes multiple software testing types that will enable to validate the software as a whole (software, hardware, and network) against the requirements for which it was built. Different types of tests (GUI testing, Functional testing, Regression testing, Smoke testing, load testing, stress testing, security testing, stress testing, ad-hoc testing etc.,) are carried out to complete system testing.

**Soak Testing** : Soak Testing is a type of performance testing, where in software is subjected to load over a significant duration of time, soak testing may go on for few days or even for few weeks. Soak testing is a type of testing that is conducted to find errors that result in degeneration of software performance with continued usage. Soak testing is extensively done for electronic devices, which are expected to run continuously for days or months or years without restarting or rebooting. With growing web applications soak testing has gained significant importance as web application availability is critical for sustaining and success of business.

**System Integration Testing** known as SIT (in short) is a type of testing conducted by software testing team. As the name suggests, focus of System integration testing is to test for errors related to integration among different applications, services, third party vendor applications etc., As part of SIT, end-to-end scenarios are tested that would require software to interact (send or receive data) with other upstream or downstream applications, services, third party application calls etc.

**Unit testing** is a type of testing that is performed by software developers. Unit testing follows white box testing approach where a developer will test units of source code like statements, branches, functions, methods, interface in OOP (object oriented programming). Unit testing usually involves in developing stubs and drivers. Unit tests are ideal candidates for automation. Automated tests can run as Unit regression tests on new builds or new versions of the software. There are many useful unit testing frames works like Junit, Nunit etc., available that can make unit testing more effective.

**Usability testing** is a type of software testing that is performed to understand how user-friendly the software is. Objective of usability testing is to allow end users to use the software, observe their behavior, their emotional response (whether users liked using software or were they stressed using it? etc.,) and collect their feedback on how the software can be made more useable or user friendly and incorporate the changes that make the software easier to use.

**User Acceptance testing (UAT )** User Acceptance testing is a must for any project; it is performed by clients/end users of the software. User Acceptance testing allows SMEs (Subject matter experts) from client to test the software with their actual business or real-world scenarios and to check if the software meets their business requirements.

**Volume testing** is a non-functional type of testing carried out by performance engineering team. Volume testing is one of the types of performance testing. Volume testing is carried out to find the response of the software with different sizes of the data being received or to be processed by the software. For e.g. If you were to be testing Microsoft word, volume testing would be to see if MS word can open, save and work on files of different sizes (10 to 100 MB).

**Vulnerability Testin**g involves identifying, exposing the software, hardware or network Vulnerabilities that can be exploited by hackers and other malicious programs likes viruses or worms. Vulnerability Testing is key to software security and availability. With the increase of hackers and malicious programs, Vulnerability Testing is critical for the success of a Business.

**White box Testing** White box testing is also known as clear box testing, transparent box testing and glass box testing. White box testing is a software testing approach, which intends to test software with knowledge of internal working of the software. White box testing approach is used in Unit testing which is usually performed by software developers. White box testing intends to execute code and test statements, branches, path, decisions and data flow within the program being tested. White box testing and Black box testing complement each other as each of the testing approaches have the potential to uncover specific category of errors.