**Experiment No.09**

PART A

(PART A: TO BE REFFERED BY STUDENTS)

**Experiment 9**

**A.1 Aim:** Design Test Cases for any two functionalities of your project

**A.2** **Prerequisite:**

Unit testing, Black box and white box testing

**A.3** **Outcome:**

Test cases

**A.4** **Theory:**

* Testing is the process of analyzing a system or system component to detect the differences between specified (required) and observed (existing) behavior.
* Activities involved in testing are:
  + Establish the test objectives
  + Design the test cases
  + Write the test cases
  + Test the test cases
  + Execute the tests
  + Evaluate the test results
  + Change the system

1. Select what has to be tested
   * Analysis: Completeness of requirements
   * Design: Cohesion
   * Implementation: Source code
2. Decide how the testing is done
   * Review or code inspection
   * Proofs (Design by Contract)
   * Black-box, white box,
   * Select integration testing strategy (big bang, bottom up, top down, sandwich)
3. Develop test cases
   * A test case is a set of test data or situations that will be used to exercise the unit (class, subsystem, system) being tested or about the attribute being measured

**Important testing documents are**

* Test plan
  + Focuses on managerial aspects of testing
  + Documents the scope, approach, resources and schedule of testing activities
  + Requirements and the components to be tested are identified in this document
* Test case specification
  + Writing **effective test cases** is a skill and that can be achieved by some experience and in-depth study of the application on which test cases are being written
* Test Incident Report
  + Each execution of each test is documented by test incident report
  + Actual results of the tests and differences from the expected output are recorded
* Test summary reports
  + It lists all the failures discovered during the tests that need to be investigated
  + Developers analyze and prioritize each failure
  + And plan for changes in the system.
  + These changes in turn can trigger new test cases and new test executions

Test Case Specification Template example:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Priority** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected results** | **Actual results** | **Pass/fail** | **Comments** |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username,  correct password | 1)Enter input(correct )username and password on the respective fields 2)click submit/login | User must successfully login to the web page | (note down the results you have observed) |  |  |
| 2 | Test if unregistered users is not able to login to the site | A |  | incorrect username,incorrect password | 1)Enter input(incorrect )username and password on the respective fields 2)click submit/login | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) |  |  |

**A.4 Task:**

**Write appropriate test cases for your project (Consider any two functionalities)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Priority** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected results** | **Actual results** | **Pass/fail** | **Comments** |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username,  correct password | 1)Enter input(correct )username and password on the respective fields 2)click submit/login | User must successfully login to the web page | (note down the results you have observed) |  |  |
| 2 | Test if unregistered users is not able to login to the site | A |  | incorrect username,incorrect password | 1)Enter input(incorrect )username and password on the respective fields 2)click submit/login | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) |  |  |

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

**(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)**

|  |  |
| --- | --- |
| Roll No. B228 | Name: Pranav Kolhe |
| Program: Btech CS | Division: A |
| Batch: A | Date of Experiment: |
| Date of Submission: | Grade: |

**B.1 Tasks given in PART A to be completed here**

*(****Students must write the answers of the task(s) given in the PART A)***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test cases | Priority | Preconditions | Input test data | Steps to be executed | Expected results | Actual results | Pass/fail | Comments |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username,  correct password | 1)Enter input(correct )username and password on the respective fields 2)click submit/login | User must successfully login to the web page | (note down the results you have observed) | Pass | Works |
| 2 | Test if unregistered users is not able to login to the site | A |  | incorrect username,incorrect password | 1)Enter input(incorrect )username and password on the respective fields 2)click submit/login | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) | Fail | Does not work |
| 3 | Test if the database is connected successfully | A | MySQL server should be configured, python connector should be installed and should be of the correct version | - | Open application | No errors should be displayed in command prompt | No errors displayed | Pass | Works |
| 4 | Test if CreditCard number is right. | A | CreditCard can accept number which is of 12 digits | 12-digit CreditCard number | Enter creditcard number in the creditcard page | No error should be shown | No errors displayed | Pass | Works |
| 5 | Test if CreditCard number is right. | A | CreditCard can accept number which is of 12 digits | 10-digit CreditCard number | Enter creditcard number in the creditcard page | Error is displayed | Error is displayed – invalid creditCard number | Fail | Does not work |
| 6 | Test if CreditCard CVV is right. | A | CreditCard can accept CVV number which is of 3 digits | 3-digit CreditCard number | Enter creditcard CVV number in the creditcard page | No error should be shown | No errors displayed | Pass | Works |
| 7 | Test if CreditCard CVV is right. | A | CreditCard can accept CVV number which is of 3 digits | 5-digit CreditCard number | Enter creditcard CVV number in the creditcard page | Error is displayed | Error is displayed – invalid creditCard CVV number | Fail | Does not work |

**B.2 Observations and Learning:**

*(****Students must write the observations and learning based on their understanding built about the subject matter and inferences drawn)***

**Validations were checked.**

**B.3 Conclusion:**

*(****Students must write the conclusive statements as per the attainment of individual outcomes listed above and learning/observation noted in section B.2)***

The experiment was executed successfully.

**B.4 Question of curiosity:**

1. What is black box and white box testing? Differentiate between them.

In Black-box testing, a tester doesn’t have any information about the internal working of the software system. Black box testing is a high level of testing that focuses on the behavior of the software. It involves testing from an external or end-user perspective. Black box testing can be applied to virtually every level of software testing: unit, integration, system, and acceptance.

White-box testing is a testing technique which checks the internal functioning of the system. In this method, testing is based on coverage of code statements, branches, paths or conditions. White-Box testing is considered as low-level testing. It is also called glass box, transparent box, clear box or code base testing. The white-box Testing method assumes that the path of the logic in a unit or program is known.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Black Box testing** | **White Box testing** |
| **Definition** | It is a testing approach which is used to test the software without the knowledge of the internal structure of program or application. | It is a testing approach in which internal structure is known to the tester. |
| **Alias** | It also knowns as data-driven, box testing, data-, and functional testing. | It is also called structural testing, clear box testing, code-based testing, or glass box testing. |
| **Base of Testing** | Testing is based on external expectations; internal behavior of the application is unknown. | Internal working is known, and the tester can test accordingly. |
| **Usage** | This type of testing is ideal for higher levels of testing like System Testing, Acceptance testing. | Testing is best suited for a lower level of testing like Unit Testing, Integration testing. |
| **Programming knowledge** | Programming knowledge is not needed to perform Black Box testing. | Programming knowledge is required to perform White Box testing. |
| **Implementation knowledge** | Implementation knowledge is not requiring doing Black Box testing. | Complete understanding needs to implement WhiteBox testing. |
| **Automation** | Test and programmer are dependent on each other, so it is tough to automate. | White Box testing is easy to automate. |
| **Objective** | The main objective of this testing is to check what functionality of the system under test. | The main objective of White Box testing is done to check the quality of the code. |
| **Basis for test cases** | Testing can start after preparing requirement specification document. | Testing can start after preparing for Detail design document. |
| **Tested by** | Performed by the end user, developer, and tester. | Usually done by tester and developers. |
| **Granularity** | Granularity is low. | Granularity is high. |
| **Testing method** | It is based on trial and error method. | Data domain and internal boundaries can be tested. |
| **Time** | It is less exhaustive and time-consuming. | Exhaustive and time-consuming method. |
| **Algorithm test** | Not the best method for algorithm testing. | Best suited for algorithm testing. |
| **Code Access** | Code access is not required for Black Box Testing. | White box testing requires code access. Thereby, the code could be stolen if testing is outsourced. |
| **Benefit** | Well suited and efficient for large code segments. | It allows removing the extra lines of code, which can bring in hidden defects. |
| **Skill level** | Low skilled testers can test the application with no knowledge of the implementation of programming language or operating system. | Need an expert tester with vast experience to perform white box testing. |
| **Techniques** | Equivalence partitioning is Black box testing technique is used for Blackbox testing.  Equivalence partitioning divides input values into valid and invalid partitions and selecting corresponding values from each partition of the test data.  Boundary value analysis  checks boundaries for input values. | Statement Coverage, Branch coverage, and Path coverage are White Box testing technique.  Statement Coverage validates whether every line of the code is executed at least once.  Branch coverage validates whether each branch is executed at least once  Path coverage method tests all the paths of the program. |
| **Drawbacks** | Update to automation test script is essential if you to modify application frequently. | Automated test cases can become useless if the code base is rapidly changing. |

2. What are stubs and drivers? Differentiate between them.

Stubs:

Stubs are developed by software developers to use them in place of modules, if the respective modules aren’t developed, missing in developing stage, or are unavailable currently while Top-down testing of modules. A Stub simulates module which has all the capabilities of the unavailable module. Stubs are used when the lower-level modules are needed but are unavailable currently.

Drivers:

Drivers serve the same purpose as stubs, but drivers are used in Bottom-up integration testing and are also more complex than stubs. Drivers are also used when some modules are missing and unavailable at time of testing of a specific module because of some unavoidable reasons, to act in absence of required module. Drivers are used when high-level modules are missing and can also be used when lower-level modules are missing.

|  |  |  |
| --- | --- | --- |
| S.No. | Stubs | Drivers |
| 1. | Stubs are used in Top-Down Integration Testing. | Drivers are used in Bottom-Up Integration Testing. |
| 2. | Stubs are basically known as a “called programs” and are used in the Top-down integration testing. | While, drivers are the “calling program” and are used in bottom-up integration testing. |
| 3. | Stubs are similar to the modules of the software, that are under development process. | While drivers are used to invoking the component that needs to be tested. |
| 4. | Stubs are basically used in the unavailability of low-level modules. | While drivers are mainly used in place of high-level modules and in some situation as well as for low-level modules. |
| 5. | Stubs are taken into use to test the feature and functionality of the modules. | Whereas the drivers are used if the main module of the software isn’t developed for testing. |
| 6. | The stubs are taken into concern if testing of upper-levels of the modules are done and the lower-levels of the modules are under developing process. | The drivers are taken into concern if testing of lower-levels of the modules are done and the upper-levels of the modules are under developing process. |
| 7. | Stubs are used when lower-level of modules are missing or in a partially developed phase, and we want to test the main module. | Drivers are used when higher-level of modules are missing or in a partially developed phase, and we want to test the lower(sub)- module. |

3. What are the advantages and disadvantages of top down over bottom up approach? Explain in detail.

|  |  |
| --- | --- |
| **Bottom-up approach** | **Top-down approach** |
| Summary | |
| * High deployment coverage in early phases * Earlier return on investment * High visibility of organizational changes * Higher impact to organization | * Tactical, limited coverage * Delayed return on investment * Lower impact to overall organization * Higher deployment costs |
| Advantages | |
| * User and business awareness of the product. Benefits are realized in the early phases. * You can replace many manual processes with early automation. * You can implement password management for a large number of users. * You do not have to develop custom adapters in the early phases. * Your organization broadens identity management skills and understanding during the first phase. * Tivoli Identity Manager is introduced to your business with less intrusion to your operations. | * Your organization realizes a focused use of resources from the individual managed application. * The first implementation becomes a showcase for the identity management solution. * When the phases are completed for the managed application, you have implemented a deeper, more mature implementation of the identity management solution. * Operation and maintenance resources are not initially impacted as severely as with the bottom-up approach. |
| Disadvantages | |
| * The organizational structure you establish might have to be changed in a later roll-out phase. * Because of the immediate changes to repository owners and the user population, the roll-out will have a higher impact earlier and require greater cooperation. * This strategy is driven by the existing infrastructure instead of the business processes. | * The solution provides limited coverage in the first phases. * A minimal percentage of user accounts are managed in the first phases. * You might have to develop custom adapters at an early stage. * The support and overall business will not realize the benefit of the solution as rapidly. * The implementation cost is likely to be higher. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*