Unit- II
Test Case Development

- Dr. Shivani Budhkar

What is a Test Case?

- A **TEST CASE** is a set of actions executed to verify a particular feature or functionality of your software application.
- A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.
- The process of developing test cases can also help find problems in the requirements or design of an application.
- A Test Case contains test steps, test data, precondition, post condition developed for specific test scenario to verify any requirement.
- The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

Test Case

- Test scenarios are rather vague and cover a wide range of possibilities. Testing is all about being very specific.
- For a Test Scenario: Check Login Functionality there many possible test cases are:
- **Test Case 1:** Check results on entering valid User Id & Password
- **Test Case 2:** Check results on entering Invalid User ID & Password
- **Test Case 3:** Check response when a User ID is Empty & Login Button is pressed, and many more

Test Case Template

- A test case can have the following elements.
- However, that a test management tool is normally used by companies and the format is determined by the tool used.

Test Suite ID	The ID of the test suite to which this test case belongs.
Test Case ID	The ID of the test case.
Test Case Summary	The summary / objective of the test case.
Related Requirement	The ID of the requirement this test case relates/traces to.
Prerequisites	Any prerequisites or preconditions that must be fulfilled prior to executing the test.
Test Procedure	Step-by-step procedure to execute the test.
Test Data	The test data, or links to the test data, that are to be used while conducting the test.
Expected Result	The expected result of the test.
Actual Result	The actual result of the test; to be filled after executing the test.
Status	Pass or Fail. Other statuses can be 'Not Executed' if testing is not performed and 'Blocked' if testing is blocked.
Remarks	Any comments on the test case or test execution.
Created By	The name of the author of the test case.
Date of Creation	The date of creation of the test case.
Executed By	The name of the person who executed the test.
Date of Execution	The date of execution of the test.
Test Environment	The environment (Hardware/Software/Network) in which the test was executed.

Test Case Example / Test Case Sample

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	To verify that clicking the Generate Coin button generates coins.
Related Requirement	RS001
Prerequisites	User is authorized. Coin balance is available.
Test Procedure	Select the coin denomination in the Denomination field. Enter the number of coins in the Quantity field. Click Generate Coin.
Test Data	Denominations: 0.05, 0.10, 0.25, 0.50, 1, 2, 5 Quantities: 0, 1, 5, 10, 20
Expected Result	Coin of the specified denomination should be produced if the specified Quantity is valid (1, 5) A message 'Please enter a valid quantity between 1 and 10' should be displayed if the specified quantity is invalid.
Actual Result	If the specified quantity is valid, the result is as expected. If the specified quantity is invalid, nothing happens; the expected message is not displayed

Test Case Example / Test Case Sample - continued

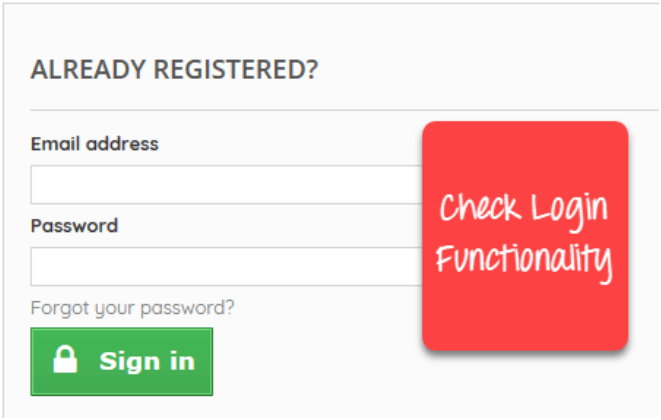
Status	Fail
Remarks	This is a sample test case.
Created By	John Doe
Date of Creation	01/14/2020
Executed By	Jane Roe
Date of Execution	02/16/2020
Test Environment	○ OS: Windows Y ○ Browser: Chrome N

Writing Good Test Cases

- As far as possible, write test cases in such a way that you test only one thing at a time. Do not overlap or complicate test cases. Attempt to make your test cases 'atomic'.
- Ensure that all positive scenarios AND negative scenarios are covered.
- Language:
  - Write in simple and easy-to-understand language.
  - Use active voice instead of passive voice: Do this, do that.
  - Use exact and consistent names (of forms, fields, etc).
- Characteristics of a good test case:
  - *Accurate*: Exacts the purpose.
  - *Economical*: No unnecessary steps or words.
  - *Traceable*: Capable of being traced to requirements.
  - *Repeatable*: Can be used to perform the test over and over.
  - *Reusable*: Can be reused if necessary.

How to Write Test Cases in Manual Testing

- Let's create a Test Case for the scenario: Check Login Functionality



Steps

- **Step 1)** A simple test case to explain the scenario would be

Test Case #	Test Case Description
1	Check response when valid email and password is entered

- **Step 2)** In order to execute the test case, you would need Test Data. Adding it below

Test Case #	Test Case Description	Test Data
1	Check response when valid email and password is entered	Email: sab12@gmail.com Password: lNf9^Oti7^2h

•Identifying test data can be time-consuming and may sometimes require creating test data afresh. The reason it needs to be documented.

Steps

- **Step 3)** In order to execute a test case, a tester needs to perform a specific set of actions on the AUT. This is documented as below:

Test Case #	Test Case Description	Test Steps	Test Data
1	Check response when valid email and password is entered	1) Enter Email Address 2) Enter Password 3) Click Sign in	Email: sab12@gmail.com Password: lNf9^Oti7^2h

- Many times the Test Steps are not simple as above, hence they need documentation. Also, the author of the test case may leave the organization or go on a vacation or is sick and off duty or is very busy with other critical tasks. A recently hire may be asked to execute the test case. Documented steps will help him and also facilitate reviews by other stakeholders.
- **Step 4)** The goal of test cases in software testing is to check behavior of the AUT for an expected result. This needs to be documented as below

Test Case #	Test Case Description	Test Data	Expected Result
1	Check response when valid email and password is entered	Email: sab12@gmail.com Password: lNf9^Oti7^2h	Login should be successful

- During test execution time, the tester will check expected results against actual results and assign a pass or fail status

Steps

Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid email and password is entered	Email: sab12@gmail.com Password: lNf9^Oti7^2h	Login should be successful	Login was successful	Pass

Step 5) That apart your test case -may have a field like, Pre - Condition which specifies things that must in place before the test can run. For our test case, a pre-condition would be to have a browser installed to have access to the site under test. A test case may also include Post - Conditions which specifies anything that applies after the test case completes. For our test case, a post condition would be time & date of login is stored in the database

The format of Standard Test Cases

- Below is a format of a standard login Test cases example.

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	1. Go to site http://demo.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = sab12 Password = pass99	User should Login into an application	As Expected	Pass
TU02	Check Customer Login with invalid Data	1. Go to site http://demo.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = sab12 Password = glass99	User should not Login into an application	As Expected	Pass

- This entire table may be created in Word, Excel or any other Test management tool.

Best Practice for writing good Test Case Example.

- **1. Test Cases need to be simple and transparent:**
- Create test cases that are as simple as possible. They must be clear and concise as the author of the test case may not execute them.
- Use assertive language like go to the home page, enter data, click on this and so on. This makes the understanding the test steps easy and tests execution faster.
- **2. Create Test Case with End User in Mind**
- The ultimate goal of any software project is to create test cases that meet customer requirements and is easy to use and operate. A tester must create test cases keeping in mind the end user perspective
- **3. Avoid test case repetition.**
- Do not repeat test cases. If a test case is needed for executing some other test case, call the test case by its test case id in the pre-condition column
- **4. Do not Assume**
- Do not assume functionality and features of your software application while preparing test case. Stick to the Specification Documents.
- **5. Ensure 100% Coverage**
- Make sure you write test cases to check all software requirements mentioned in the specification document. Use Traceability Matrix to ensure no functions/conditions is left untested.

Best Practice for writing good Test Case Example.

- **6. Test Cases must be identifiable.**
- Name the test case id such that they are identified easily while tracking defects or identifying a software requirement at a later stage.
- **7. Implement Testing Techniques**
- It's not possible to check every possible condition in your software application. Software Testing techniques help you select a few test cases with the maximum possibility of finding a defect.
- **Boundary Value Analysis (BVA):** As the name suggests it's the technique that defines the testing of boundaries for a specified range of values.
- **Equivalence Partition (EP):** This technique partitions the range into equal parts/groups that tend to have the same behavior.
- **State Transition Technique**: This method is used when software behavior changes from one state to another following particular action.
- **Error Guessing Technique:** This is guessing/anticipating the error that may arise while doing manual testing. This is not a formal method and takes advantages of a tester's experience with the application
- 8. **Self-cleaning**
- The test case you create must return the Test Environment to the pre-test state and should not render the test environment unusable. This is especially true for configuration testing.
- 9. **Repeatable and self-standing**
- The test case should generate the same results every time no matter who tests it
- **10. Peer Review.**
- After creating test cases, get them reviewed by your colleagues. Your peers can uncover defects in your test case design, which you may easily miss.