**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 contains test steps, test data, precondition, postcondition 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 Scenario Vs 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

This is nothing but a Test Case.

**The format of Standard Test Cases**

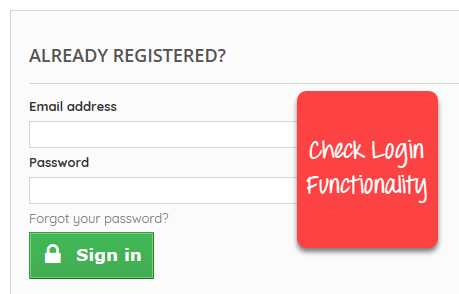
Below is a format of a standard login Test cases example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check Customer Login with valid Data | 1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit | Userid = guru99 Password = pass99 | User should Login into an application | As Expected | Pass |
| TU02 | Check Customer Login with invalid Data | 1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit | Userid = guru99 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](https://www.guru99.com/top-20-test-management-tools.html). That’s all to Test Case Design

**How to Write Test Cases in Manual Testing**

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



**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)** Test the Data.  
In order to execute the test case, you would need [Test Data](https://www.guru99.com/software-testing-test-data.html). Adding it below

|  |  |  |
| --- | --- | --- |
| **Test Case #** | **Test Case Description** | **Test Data** |
| 1 | Check response when valid email and password is entered | Email: guru99@email.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.

**Step 3)** Perform actions.  
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: guru99@email.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)** Check behavior of the AUT.  
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

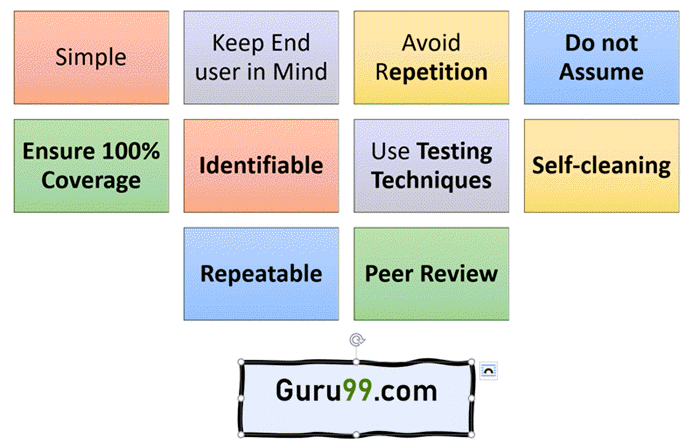
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case #** | **Test Case Description** | **Test Data** | **Expected Result** |
| 1 | Check response when valid email and password is entered | Email: guru99@email.com Password: lNf9^Oti7^2h | Login should be successful |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Description** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Check response when valid email and password is entered | Email: guru99@email.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 be 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 postcondition would be time & date of login is stored in the database

**Best Practice for writing good Test Case.**



**Test Case Best Practice**

**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](https://www.guru99.com/traceability-matrix.html)to ensure no functions/conditions is left untested.

**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](https://www.guru99.com/test-environment-software-testing.html)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.

**While drafting a test case to include the following information**

* The description of what requirement is being tested
* The explanation of how the system will be tested
* The test setup like a version of an application under test, software, data files, operating system, hardware, security access, physical or logical date, time of day, prerequisites such as other tests and any other setup information pertinent to the requirements being tested
* Inputs and outputs or actions and expected results
* Any proofs or attachments
* Use active case language
* Test Case should not be more than 15 steps
* An automated test script is commented with inputs, purpose and expected results
* The setup offers an alternative to pre-requisite tests
* With other tests, it should be an incorrect business scenario order

**Test Case Management Tools**

Test management tools are the automation tools that help to manage and maintain the Test Cases. Main Features of a test case management tool are

1. **For documenting Test Cases:**With tools, you can expedite Test Case creation with use of templates
2. **Execute the Test Case and Record the results:**Test Case can be executed through the tools and results obtained can be easily recorded.
3. **Automate the Defect Tracking:**Failed tests are automatically linked to the bug tracker, which in turn can be assigned to the developers and can be tracked by email notifications.
4. **Traceability:**Requirements, Test cases, Execution of Test cases are all interlinked through the tools, and each case can be traced to each other to check test coverage.
5. **Protecting Test Cases:** Test cases should be reusable and should be protected from being lost or corrupted due to poor version control. Test Case Management Tools offer features like

* Naming and numbering conventions
* Versioning
* Read-only storage
* Controlled access
* Off-site backup

Popular Test Management tools are: [Quality Center](https://www.guru99.com/hp-alm-free-tutorial.html)and [JIRA](https://www.guru99.com/jira-tutorial-a-complete-guide-for-beginners.html)

## What is a Test Scenario?

A **Test Scenario** is defined as any functionality that can be tested. It is also called *Test Condition* or *Test Possibility*. As a tester, you should put yourself in the end user’s shoes and figure out the real-world scenarios and use cases of the Application Under Test.

**Why create Test Scenarios?**

Test Scenarios are created for the following reasons,

* Creating Test Scenarios ensures complete Test Coverage
* Test Scenarios can be approved by various stakeholders like Business Analyst, Developers, Customers to ensure the Application Under Test is thoroughly tested. It ensures that the software is working for the most common use cases.
* They serve as a quick tool to determine the testing work effort and accordingly create a proposal for the client or organize the workforce.
* They help determine the most important end-to-end transactions or the real use of the software applications.
* For studying the end-to-end functioning of the program, Test Scenario is critical.

**When not create Test Scenario?**

Test Scenarios may not be created when

* The Application Under Test is complicated, unstable and there is a time crunch in the project.
* Projects that follow Agile Methodology like Scrum, Kanban may not create Test Scenarios.
* Test Scenario may not be created for a new bug fix or [Regression Testing](https://www.guru99.com/regression-testing.html). In such cases, Test Scenarios must be already heavily documented in the previous test cycles. This is especially true for Maintenance projects.