Test case

Tasniya Ahmed
Lecturer
IIT, NSTU

What is a Test Case?

- Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end user.
- Learning how to write test cases requires basic writing skills, an attention to detail, and a good understanding of the application under test (AUT).
- **A well-written test case should allow any tester to understand and execute the test**

How to write test cases for software:

1.Use a Strong Title

- A good test case starts with a strong title. As a best practice, it’s good to name the test case along the same lines as the module that you are testing. For example, if you’re testing the login page, include “Login Page” in the title of the test case. In some cases, if the tool you’re using doesn’t already do this, it might makes sense to include a unique identifier in the title of the test case as well, so the identifier can be referenced instead of a long title

2. Include a Strong Description

- The description should tell the tester what they're going to test. Sometimes this section might also include other pertinent information such as the test environment, test data, and preconditions/assumptions. A description should be easy to read and immediately communicate the high-level goal of the test.

3. Include Assumptions and Preconditions

- You should include any assumptions that apply to the test and any preconditions that must be met prior to the test being executed. This information can include which page the user should start the test on, dependencies on the test environment, and any special setup requirements that must be done before running the test. This information also helps keep the test steps short and concise.

4. Keep the Test Steps Clear and Concise

- Test cases should be simple. Keep in mind, the person who wrote the test case might not be the same person who executes the test itself. The test steps should include the necessary data and information on how to execute the test. This is perhaps the most important part of a test case. Keep this section clear and concise, but don't leave out any necessary details. Write the test case so that anyone can go in and perform the test.

5. Include the Expected result

- The expected result tells the tester what they should experience as a result of the test steps. This is how the tester determines if the test case is a “pass” or “fail”.

6. Make it Reusable

- A good test case is reusable and provides long-term value to the software testing team. When writing a test case, keep this in mind. You can save time down the road by re-using the test case instead of re-writing it.

Sample of a Test Case

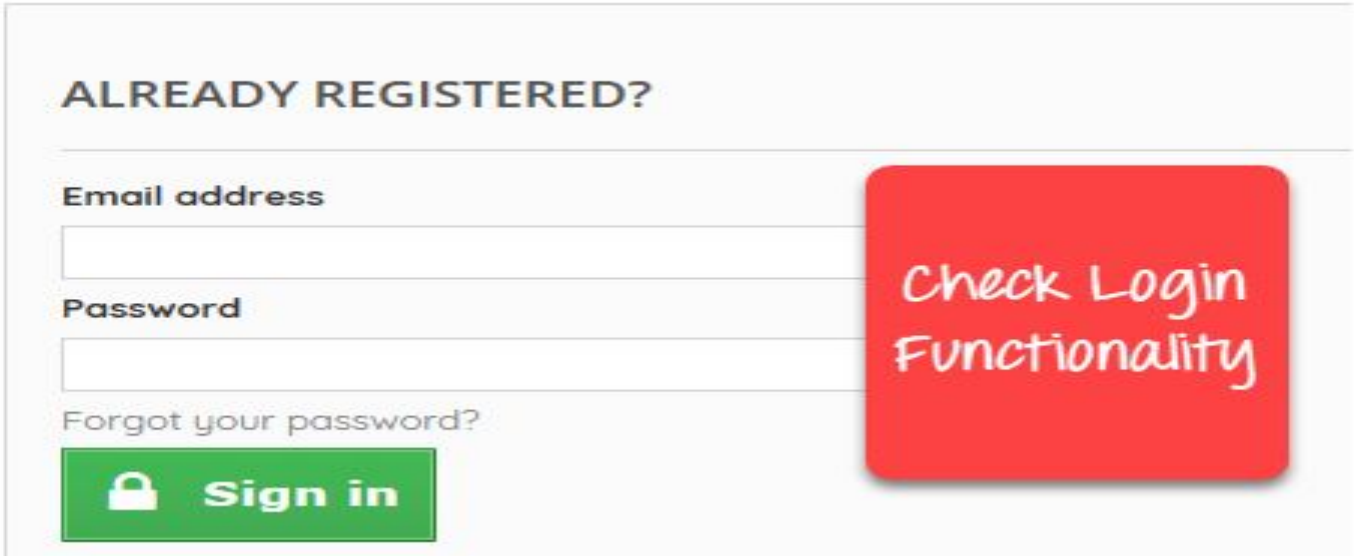
- Here is an example of a test case:
- **Title**: Login Page – Authenticate Successfully on gmail.com
- **Description:** A registered user should be able to successfully login at gmail.com.
- **Precondition:** the user must already be registered with an email address and password.
- **Assumption:** a supported browser is being used.

- **Test Steps:**
- Navigate to gmail.com
- In the ’email’ field, enter the email address of the registered user.
- Click the ‘Next’ button.
- Enter the password of the registered user
- Click ‘Sign In’
- **Expected Result:** A page displaying the gmail user’s inbox should load, showing any new message at the top of the page.

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)** 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: guru99@email.com Password: lNf9^Oti7^2h

- **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: guru99@email.com Password: INf9^Oti7^2h

- **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: 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	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 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


Simple
Keep End user in Mind
Avoid Repetition
Do not Assume
Ensure 100% Coverage
Identifiable
Use Testing Techniques
Self-cleaning
Repeatable
Peer Review

The format of Standard Test Cases

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.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