<Project/Sub-project Title>

**Test Case**

<Testing Phase>

<Office/Group>

TABLE OF CONTENTS

[1. Introduction: 4](#_Toc10926846)

[1.1 Purpose of the Test Case Document 4](#_Toc10926847)

[2. Test Case Specification: 4](#_Toc10926848)

[2.1 Test Case Description: 4](#_Toc10926849)

[2.2 Resources 4](#_Toc10926850)

[2.3 Postconditions 5](#_Toc10926851)

[2.4 Inclusion/Exclusion Points: 5](#_Toc10926852)

[2.5 Special Requirements: 5](#_Toc10926853)

[2.5.1 Testing Techniques: 5](#_Toc10926854)

[3. Test Case Content: 6](#_Toc10926855)

[3.1 Test Case ID 6](#_Toc10926856)

[3.2 Test Scenario 6](#_Toc10926857)

[3.3 Test Steps 6](#_Toc10926858)

[3.4 Test Data 6](#_Toc10926859)

[3.5 Expected Results 6](#_Toc10926860)

[3.6 Actual Result 6](#_Toc10926861)

[3.7 Status 7](#_Toc10926862)

[4. Test Case sample 8](#_Toc10926863)

**Document Control**

**Document Version History**

This table shows a record of significant changes to the document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author (Name/ Position)** | **Description of Change**  **(Including Section of Script)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Review and Approval History**

| Reviewer Name | Position | Date Reviewed | Document Review Status (Approved/Not Approved) | Comments Incorporated (Yes/No) |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Document References**

| Name | Document Location | Type of Reference |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# Introduction:

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies the requirements or works correctly. In order to get the best out of a product at the end of its production phase, you will need to make sure that it is bug-free. A test case, may in the end, be a scientific method not only for detecting bugs but for creating a hypothesis on how to remove them.

## Purpose of the Document

[Provide the purpose of the Test Case Document. This document should be modified to fit a project’s needs.]

The Test Case documents the functional requirements of the *<test case title>* test case. The predetermined audience is the project manager, project team, and testing team. Some segments of this document may be shared with the client/user and other stakeholder when required, whose input/approval into the testing process is needed.

# Test Case Specification:

[Describe the test case to be performed]

## Test Case Description:

[Describe the test case and the person involved in the testing]

|  |  |
| --- | --- |
| **Test Case ID** |  |
| **Test Priority (Low/Medium/High** |  |
| **Module Name** |  |
| **Test Title** |  |
| **Description** |  |
| **Test Designed by** |  |
| **Test Designed date** |  |
| **Test Executed by** |  |
| **Test Execution date** |  |

## Resources

[Describe who are involved in the testing, their responsibilities, and their contribution towards the test case.]

|  |  |  |
| --- | --- | --- |
| **Responsibilities** | **Team Member** | **Role** |
| **Test Case Creation** |  |  |
| **Test Case Review** |  |  |
| **Test Case Execution** |  |  |
| **Test Data Setup** |  |  |

Prerequisites

[Describe the prerequisites for the test case. A prerequisite is the state of the system that must be fulfilled before a test case can be performed.]

|  |  |
| --- | --- |
| **Serial #** | **Prerequisites** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

## Postconditions

[Describe the post conditions for the use case. A post condition is a list of possible states the system can be in just after finishing a test case.]

|  |  |
| --- | --- |
| **Serial #** | **Postconditions** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

## Inclusion/Exclusion Points:

[Describe other inclusions and exclusions points in the act of executing this test case.]

|  |  |  |
| --- | --- | --- |
| **Serial #** | **Inclusions** | **Exclusions** |
|  |  |  |
|  |  |  |
|  |  |  |

## Special Requirements:

[Describe any special requirements necessary to perform the test case.]

### Testing Techniques:

Testing a system or a program is hard, not only that the testing needs to be taken care of, you also need to think of all the possible bugs that can occur, as well as the conditions that need to be made. With the help of a testing technique, you can get a few test cases that have the highest possibility of finding a bug in your software.

* **Equivalence Partition:** This will partition the range into groups that have similar behaviour
* **Error Guessing Technique:** This method will guess the most probable error that may occur during testing, it relies on the experience of the tester
* [**Boundary Value Analysis**](http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf)**:** This technique will help you define boundary testing for a specified range of values

# Test Case Content:

## Test Case ID

A test case ID is important because you will be able to document your results with ease. It’s helps to avoid testing the same things repeatedly. An ID can be marked however you want, if you’re doing multiple systems you can start with the name with a short acronym of the programs name or even with the full name.

## Test Scenario

The test scenario field will include all possible information about the testing. It won’t contain any sequences or steps of the testing phase, but it will contain a specific objective that the tester should keep in mind. This field could contain information like “Check admin login page with valid data”.

## Test Steps

It is plain and simple and contains all the info needed in concise short steps. Depending on how complicated the objective is, you may need to give the tester a few more steps, but always keep them short and specific. Here are few steps:

1. Start up the software
2. Head over to the registration area
3. Insert data in all fields
4. Submit the data by clicking register

## Test Data

It’s for an objective where the tester needs to test its functionalities by placing the data and trying to do his testing, For example login to a page, he need to place the username and password data, he will need that information in order to fulfil his objective.

## Expected Results

This is the most important part because this field will explain the tester on what results he should expect, given that the system is bug-free. If the tester is trying to log into an account on the given software, the expected results could for an example say that he should end up logging into his account.

This could take a twist, perhaps the tester is given invalid data for the login fields, then naturally he should have expected that the website or software does not let him log into his account. Make sure that the expected results field should be precise as well and that it includes everything that the software is intended to do in that step.

## Actual Result

Once the tester fulfils all the test steps and inserts all the test data, he should now get the results of his testing. The results can be both positive and negative, they can either confirm that the program is working as intended or that it gives an entirely different result.

Although it is more preferred to have positive results, negative results will help you a lot by detecting that there is, in fact, a bug hidden in the code. For an example, if clicking a picture of a red button should convert its colour into blue, then the results are as expected if that’s what the system is supposed to do. But if the colour of the button stays the same or if it changes into yellow, then there is a problem that should be fixed.

## Status

This field simply describe the status of a test case which can Pass/Fail/Not Executed/Suspended.

# Test Case sample

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Case Steps** | **Expected Result** | **Actual Result** | **Test Date** | **Environment** | **Result (Pass/ Fail)** | **Remarks** |
|
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |