TITLE: Write TEST Scenario for Gmail Login Page

AIM: The students will understand and enhance their learning of test scenario.

OBJECTIVES:

1. To learn the test scenarios.

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. In such cases, Test Scenarios must be already heavily documented in the previous test cycles. This is especially true for Maintenance projects.

CONCLUSION: Thus, we have created test scenarios for Gmail login page.

TITLE: Write TEST Scenario for Gmail inbox Page

AIM: The students will understand and enhance their learning of test scenario.

OBJECTIVES:

1. To learn the test scenarios.

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.

How to Write Test Scenarios

As a tester, you can follow these five steps to create Test Scenarios-

- **Step 1**: Read the Requirement Documents like BRS, SRS, FRS, of the System Under Test (SUT). You could also refer uses cases, books, manuals, etc. of the application to be tested.
- **Step 2**: For each requirement, figure out possible users' actions and objectives. Determine the technical aspects of the requirement. Ascertain possible scenarios of system abuse and evaluate users with hacker's mindset.
- **Step 3:** After reading the Requirements Document and doing your due Analysis, list out different test scenarios that verify each feature of the software.

- **Step 4:** Once you have listed all possible Test Scenarios, a Traceability Matrix is created to verify that each & every requirement has a corresponding Test Scenario
- **Step 5:** The scenarios created are reviewed by your supervisor. Later, they are also reviewed by other Stakeholders in the project.

Tips to Create Test Scenarios

- Each Test Scenario should be tied to a minimum of one Requirement or User Story as per the Project Methodology.
- Before creating a Test Scenario that verifies multiple Requirements at once, ensure you have a Test Scenario that checks that requirement in isolation.
- Avoid creating overly complicated Test Scenarios spanning multiple Requirements.
- The number of scenarios may be large, and it is expensive to run them all. Based on customer priorities only run selected Test Scenarios

CONCLUSION: Thus we have created test scenarios for Gmail inbox page.

TITLE: Write Test cases in excel sheet for Social Media application or website

AIM: The students will understand and enhance their learning of test cases.

OBJECTIVES:

1. To learn the test cases.

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.

Best Practice for writing good Test Case.

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.

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 behaviour.

- **State Transition Technique**: This method is used when software behaviour 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.

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 Centre and JIRA

CONCLUSION: Thus we have created Test cases in excel sheet for Social Media application or website.

TITLE: Installation of Selenium grid and selenium Web driver java eclipse (automation tools).

AIM: The students will learn installation of automation tools(selenium).

OBJECTIVES:

1. To learn installation of selenium.

The process of installing Selenium involves 3 steps, namely:

1. Install Java
2. Install Eclipse IDE
3. Install Selenium WebDriver

What is Selenium?

Selenium refers to a suite of tools that are widely used in the testing community when it comes to cross-browser testing. Selenium cannot automate desktop applications; it can only be used in browsers. It is considered to be one of the most preferred tool suites for automation testing of web applications as it provides support for popular web browsers which makes it very powerful.

It supports a number of browsers (Google Chrome 12+, Internet Explorer 7,8,9,10, Safari 5.1+, Opera 11.5, Firefox 3+) and operating systems (Windows, Mac, Linux/Unix).

Selenium also provides compatibility with different programming languages – C#, Java, JavaScript, Ruby, Python, PHP. Testers can choose which language to design test cases in, thus making Selenium highly favourable for its flexibility.

Selenium Components

The Selenium test suite comprises four main components:-

- **Selenium IDE**

  Selenium IDE (Integrated Development Environment) is primarily a record/run tool. It is an Add-on or an extension available for both Firefox and Chrome that generates tests quickly through its functionality of record and playback. You don't need to learn any test scripting language for authoring any functional tests.

- **Selenium RC**

  In the case of working with Selenium RC (Remote Control), one must have good knowledge of at least one programming language. This tool allows you to develop responsive design tests in any scripting language of your choice. Server and client libraries are the two main components of Selenium RC. Its architecture is complex and it has its limitations.

- **Selenium Web driver**

  Selenium WebDriver is an enhanced version of Selenium RC. It was introduced in the market to overcome the limitation faced in Selenium RC. Though it is an advanced version of RC, its architecture is completely different from that of RC. Just like Selenium RC, Selenium WebDriver too supports multiple programming platforms to provide wider flexibility and requires knowing any one programming language.

- **Selenium Grid**

  Selenium Grid is a tool that is used for concurrent execution of test cases on different browsers, machines, and operating systems simultaneously. This tool makes Cross-browser compatibility testing very easy. There are two versions of the Selenium Grid – the older version is known as Grid 1 and the recent version is known as Grid 2.

CONCLUSION: Thus we completed installation of selenium web driver and selenium grid.

TITLE: Prepare Software requirement specification for any project or problem statement

AIM: The students will learn Software requirement specification.

OBJECTIVES:

1. To learn SRS preparation.

What is Software Requirement Specification - [SRS]?

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Types of Requirements:

The below diagram depicts the various types of requirements that are captured during SRS.

- **Software Requirement Specification (SRS) Format** as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system.
- These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because it's necessary to fully understand needs of customers.
- Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

1. **Introduction**
2. **General description**
3. **Functional Requirements**
4. **Interface Requirements**
5. **Performance Requirements**
6. **Design Constraints**
7. **Non-Functional Attributes**
8. **Preliminary Schedule and Budget**
9. **Appendices**

CONCLUSION: Thus we have completed SRS for project topic.