**GUIDELINES DOCUMENT**

**FOR DISPLAY BOOKSHELVES**

[**https://www.urbanladder.com**](https://www.urbanladder.com)

**Submitted By**

KINESTHETICS

SRAVYA KATUKURI

SAILAJA KANDLURI

NACHIKET KISAN SHEJWAL

MOHAMMAD KHALEED DUDEKULA

912790

912797

912649

912786

CONTENT

1. INTRODUCTION
2. SCOPE
3. TEST APPROACH
4. TEST TECHNIQUES
   1. Decision-table Testing
   2. State Transition Testing
5. TYPES OF TESTING
   1. Functional Testing
   2. Smoke Testing
   3. Regression Testing
6. CODING STANDARDS
   1. Naming Conventions
   2. Error Handling
   3. General Guidelines
7. TESTING STANDARDS
   1. Testing Strategies
   2. Inspecting Results
   3. General Guidelines
8. DEVELOPING FRAMEWORK
9. TEMPLATES
   1. Scenarios
   2. Test Cases
   3. RTM
10. CONCLUSION

3

3

3

4

5

6

7

7

8

9

1. **INTRODUCTION**

A Guidelines Document helps to ensure the consistency of the project which later results in the efficiency of the developed applications. These guidelines in the document must be treated as reference and must be adopted accordingly.

1. **SCOPE**

This Document is going to give the outline of the requirements for the project. This includes a collection of standards, conventions and guidelines for designing and developing framework for java code in selenium automation and testing guidelines.

**Intended Audience:**

* Business Unit (BU) Person
* Academy Coach
* Technical Trainer
* Project Team

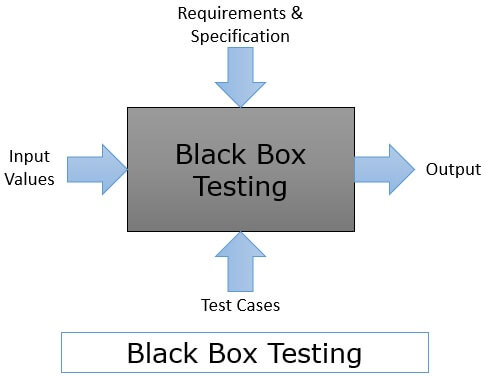
1. **TEST APPROACH**

BLACK BOX TESTING

Black box testing is a functional testing type where the input is given and output is obtained without having knowledge about the internal structure of the application. The obtained output is verified with the expected result. Below added picture best describes the Black Box testing approach. Test cases will be given and output will be asserted.

Characteristics:

* Functionality.
* Requirements, usage standards
* Correctness.
* Business forms, Documents.
* Does system meet business requirements.



1. **TEST TECHNIQUES**
   1. DECISION TABLE TESTING

* This technique comes into picture when there are complex business rules dictating the processing of the input. In this method, a table is created which shows all the inputs, business rules, and corresponding outputs.
* For instance, to understand the decision table, consider the example of a simple login page which has 2 inputs user id and password and depending on these inputs output or error is shown:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Decision table** | **Rule1** | **Rule2** | **Rule3** | **Rule4** |
| **Input** |  |  |  |  |
| User id | N | N | Y | Y |
| Password | N | Y | N | Y |
| **Output** |  |  |  |  |
| Success |  |  |  | X |
| Error Message | X | X | X |  |

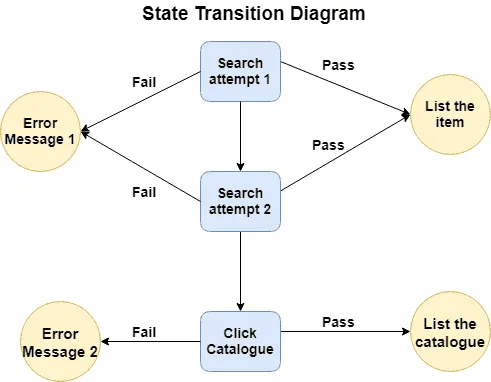
*N: invalid input*

*Y: valid input*

*X: identifies the output*

* 1. STATE TRANSITION TESTING
* State transitioning testing method traces the changes in the state of the application under test based on the inputs and conditions. The state transition testing incorporates testing of transition in the state and state sequences of the application as per the events. It is suitable for smaller applications with finite inputs and states.
* For instance, consider search item functionality of an online e-commerce website.

1. User searches for an item. If found system lists it. Else system shows an error.
2. User searches the item again and if found system lists it. Otherwise shows an error.
3. User clicks on Catalogue and hence the system shows Catalogue menu. Or else, in case of issues shows an error message.



1. **TYPES OF TESTING**
   1. FUNCTIONAL TESTING

* The purpose of functional testing is to assess whether the application meets the functional requirements.
* Also, it discovers the non-conformance of the application with the end user requirements.
  1. SMOKE TESTING
* Smoke testing is done in the initial stage to check the stability of the system.
* It verifies the major functionalities of the system in the surface level.
  1. REGRESSION TESTING
* Regression testing is a form of re-testing done to ensure that there is no impact on the system after a modification is done.
* Sanity testing is a part of regression testing which validates the functionality of the change performed.
* Also, in a deep level it checks if the functionality of the overall system is increasing and if the system is stable.

1. **CODING STANDARDS**
   1. NAMING CONVENTIONS

* A naming convention is a set of rules for choosing the character sequence to be used for identifiers which denote variables, types, functions, and other entities in source code and documentation.
* It helps other programmers to quickly pick up the code when they have to, therefore leveraging overall readability of the code.
  1. WARNINGS

Compiler warnings shall be treated as errors and fixed. Even though the program will continue to compile in the presence of warnings, they often indicate problems which may affect the behaviour, reliability, and portability of the code.

* 1. GENERAL GUIDELINES
* Keep code concise and easy to understand.
* Build reusable methods.
* Add comments for better understanding.
* Add indentation spaces for better understandability of the code.
* Handle possible exceptions.

1. **TESTING STANDARDS**
   1. TEST STRATEGIES
   * Testing should be done with correct data and test cases as well as with flawed test cases to make sure the system is leak proof. Test cases must be well documented to ensure future reuse for testing at later stages.
   1. INSPECTING RESULTS

Most of the test cases yield one of the 3 results

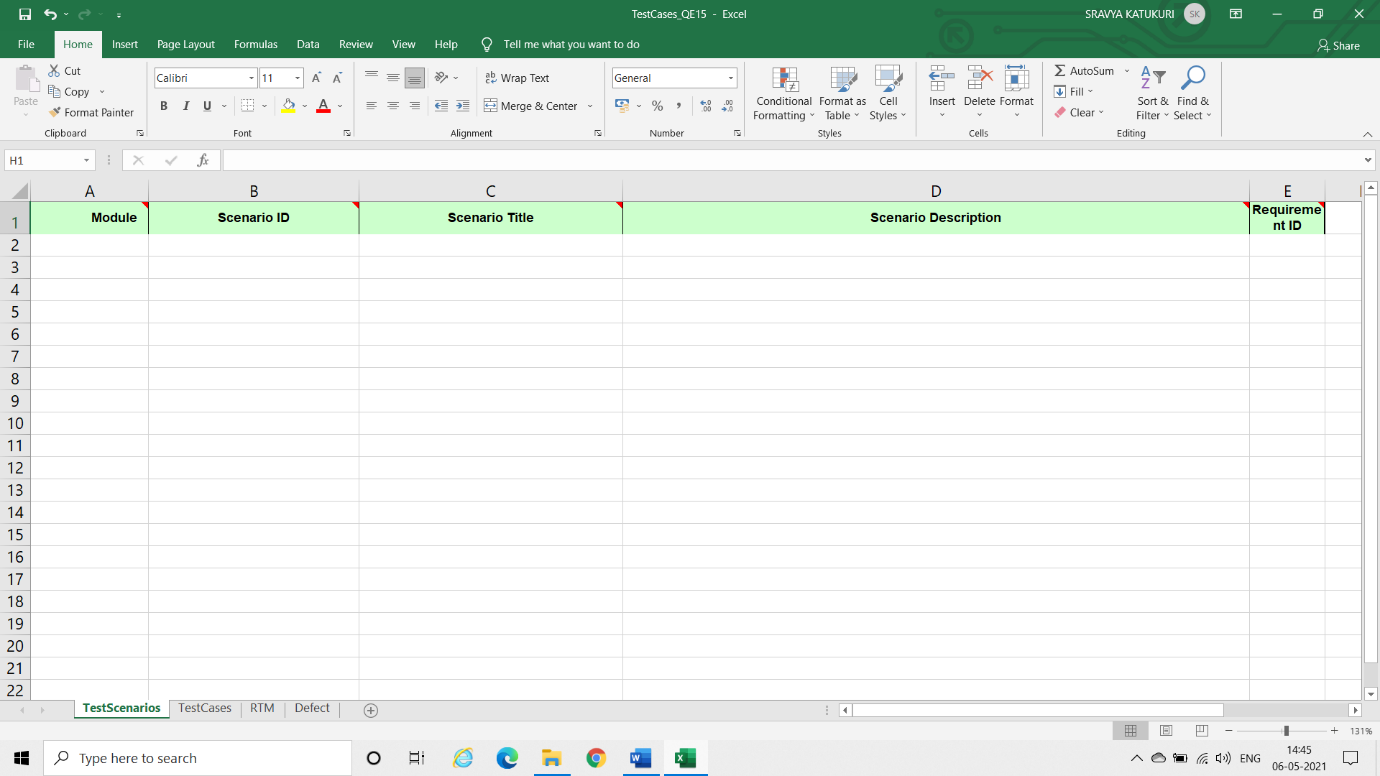
* PASS: A test case is declared pass when the actual output matches to expected output.
* FAIL: A test case is declared fail when the actual output is different from the expected output.
* BLOCKED: A test case is declared blocked when it is depending on a previous case which is failed and reported as bug.
  1. GENERAL GUIDELINES
* Assumptions must be validated.
* Each method, module, class mut be tested individually.
* Waits must be used.
* When required screenshots must be taken to report a bug.

1. **DEVELOPING FRAMEWORK**

* **Eclipse-IDE for Java development** will be used for running and holding the complete project. It will work as an engine for the testing environment. Eclipse Neon 3 is expected to be used.
* **Selenium** Open-Source library to automate the browser for performing automation testing. Selenium 3.141.59 version is expected to be used.
* **TestNG** is an automation testing framework in which NG stands for "Next Generation". This will be used to manage the test cases and the test scenarios.
* **Maven** Build automation tool will be used to manage the project flow and dependencies. It makes the build consistent with other projects.
* **Apache POI** Apache POI library will be used for the data driven Selenium tests. POI is a set of library files under API which is capable enough to read and write both XLS and XLSX file format of Excel. Apache POI 5.0.0 is expected to be used.
* **Page Object Model,** also known as POM is a design pattern in Selenium that creates an object repository for storing all web elements. It is used for reducing code duplication and improves test case maintenance.

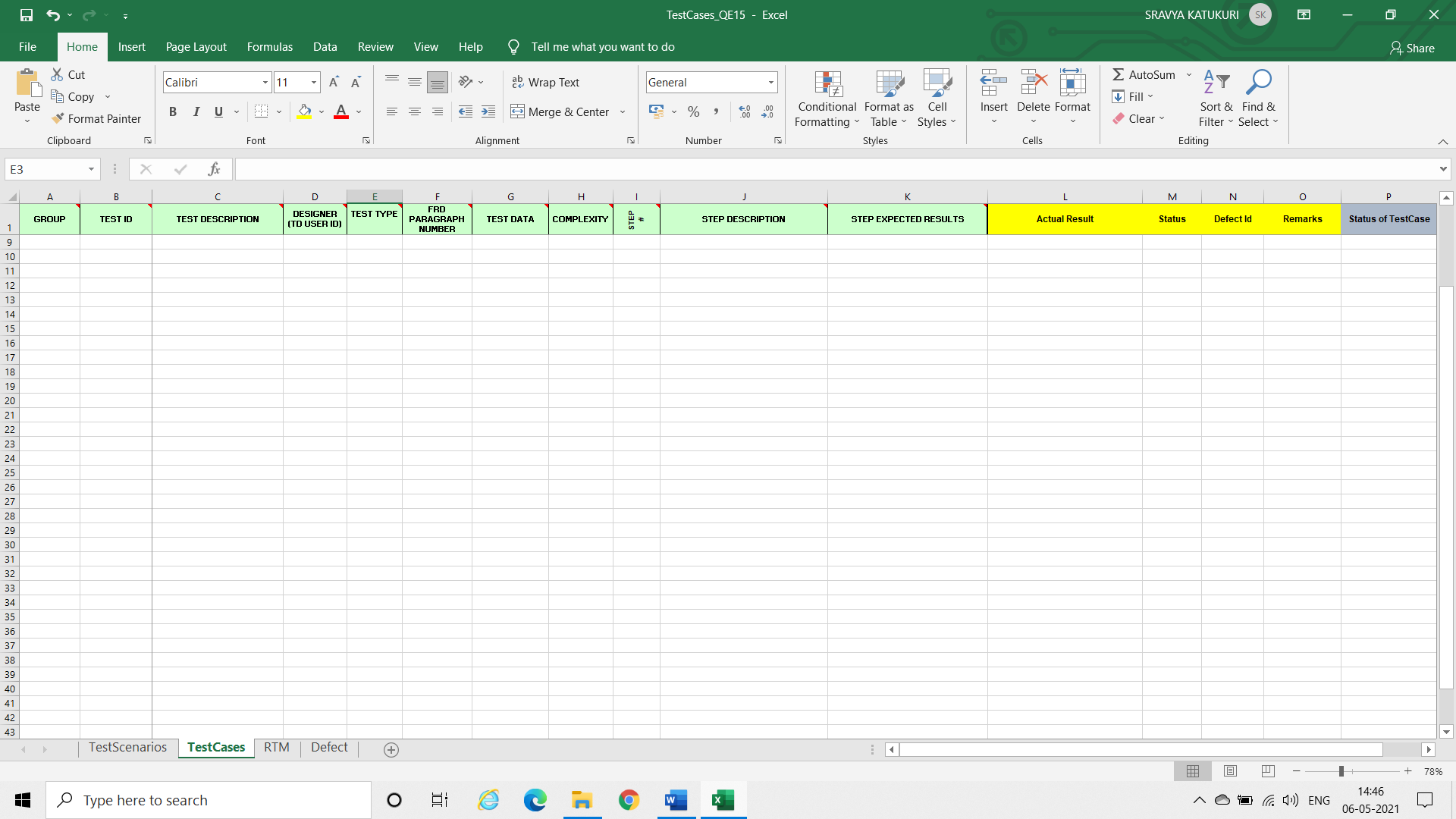
1. **TEMPLATES**
   1. TEST SCENARIOS

* Test scenario document is a typical breakdown of business requirements into high level testable requirements.
* A well written test scenario is easy to read and understand.



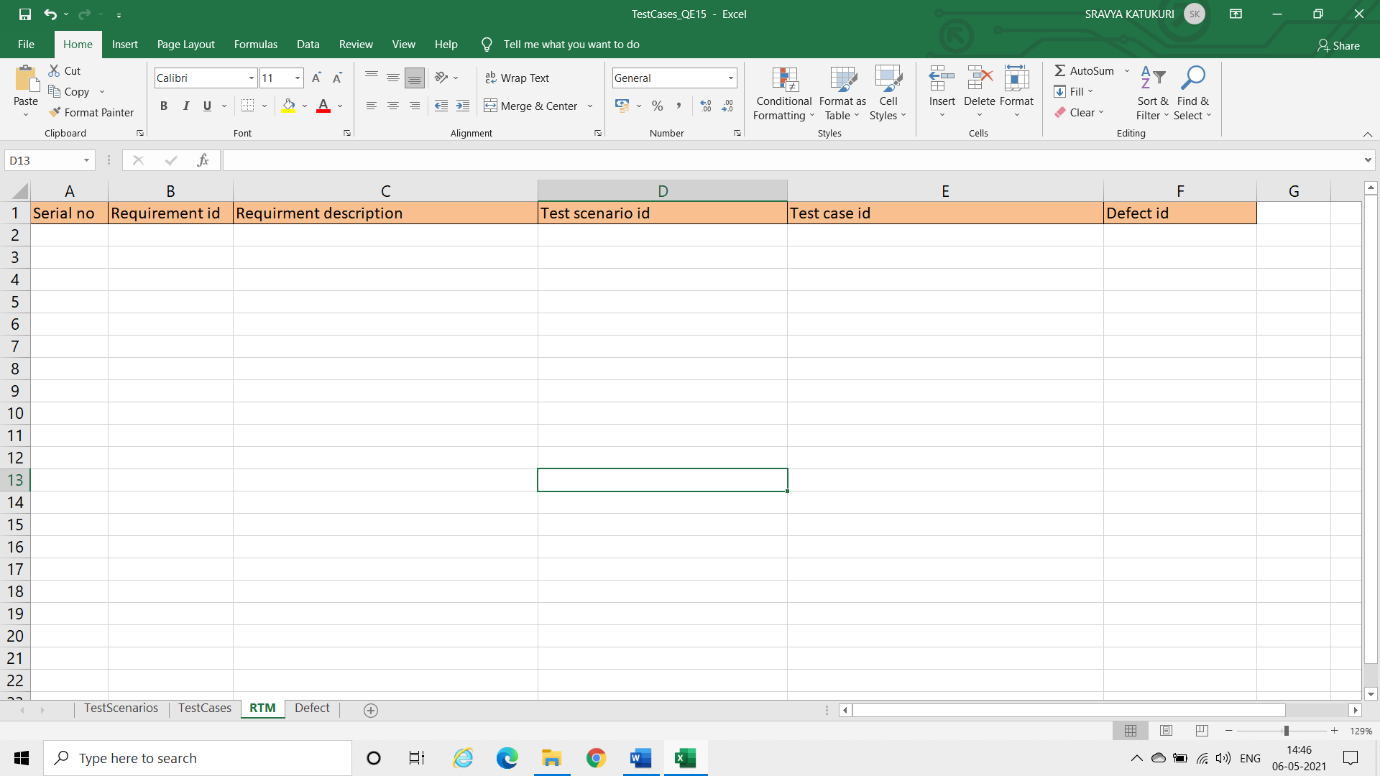
Template of Test Scenario document

* 1. TEST CASES
* Test cases are a breakdown of test scenarios.
* They contain set of inputs, execution conditions, expected and actual results.
* Test cases must have reasonable probability of catching an error.
* Test cases must cover both positive and negative conditions.



Template of Test case document

* 1. RTM
* RTM stands for Requirement Traceability Matrix.
* This document validates and links the functional requirements against testing process.



Template of RTM

1. **CONCLUSION**

This document outlined all the work required to deliver a project. This document described a collection of standards, conventions and guidelines for designing and developing framework for java code in selenium automation and testing guidelines.