**GUIDELINES DOCUMENTATION FOR “DISPLAY BOOKSHELVES”**

Submitted to:

Cognizant Technology Solutions (CTS)

***Submitted by***

|  |  |
| --- | --- |
| **Name** | **Employee ID** |
| Gyanraj Hansdah | 2361781 |

For more information, please visit on website:

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

**TABLE OF CONTENTS**

**Contents**

[**1.INTRODUCTION 2**](#_Toc180965900)

[**2.SCOPE 2**](#_Toc180965901)

[**3.CODING STANDARDS 2**](#_Toc180965902)

[**3.1 NAMING CONVENTIONS 2**](#_Toc180965903)

[**3.2 INDENTATION 4**](#_Toc180965904)

[**3.3 ERROR HANDLING 4**](#_Toc180965905)

[**3.4 GENERAL GUIDELINES FOR CODING 4**](#_Toc180965906)

[**4. TESTING STANDARDS 5**](#_Toc180965907)

[**4.1 TESTING STRATEGIES 5**](#_Toc180965908)

[**Positive testing: 5**](#_Toc180965909)

[**Negative testing: 5**](#_Toc180965910)

[**4.2 PROJECT ASSUMPTIONS 5**](#_Toc180965911)

[**4.3 INSPECTION RESULTS 6**](#_Toc180965912)

[**“PASS” 6**](#_Toc180965913)

[**“FAIL” 6**](#_Toc180965914)

[**“PENDING” 6**](#_Toc180965915)

[**4.4 GENERAL GUIDELINES FOR TESTING 6**](#_Toc180965916)

[**5. AUTOMATION FEASIBILITY 6**](#_Toc180965917)

[**6. TYPES OF TESTING 7**](#_Toc180965918)

[**6.1 SMOKE TESTING 7**](#_Toc180965919)

[**6.2 REGRESSION TESTING 7**](#_Toc180965920)

[**7. FRAMEWORK 7**](#_Toc180965921)

[**7.1 TestNG 7**](#_Toc180965922)

[**7.2 MAVEN 7**](#_Toc180965923)

[**7.3 APACHE POI 8**](#_Toc180965924)

[**7.4 PAGE OBJECT MODEL (POM) 8**](#_Toc180965925)

[**8. BROWSERS 8**](#_Toc180965926)

[**8.1 CHROME 8**](#_Toc180965927)

[**8.2 MICROSOFT EDGE 9**](#_Toc180965928)

[**9. CONCLUSION 9**](#_Toc180965929)

# 1.INTRODUCTION

This document will help to ensure consistency of the code which results in the efficiency of the developed code. This document describes a collection of standards, conventions, and the guidelines for designing and developing framework for java code in selenium automation and testing guidelines.

# 2.SCOPE

This document outlines all the work required to deliver a project. It includes the functions and features of the final deliverables, as well as any factors that defines project success. Users of this document must treat these guidelines as a reference and adopt these guidelines accordingly.

# 3.CODING STANDARDS

## 3.1 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.

The naming conventions used:

* It reduces the effort needed to read and understand source code.
* It enables code reviews to focus on more important issues than arguing over syntax and naming standards.
* If enable code quality review tools to focus their reporting mainly on significant issues other than syntax and style preferences.

Project Elements

|  |  |  |
| --- | --- | --- |
| S. NO | PROJECT ELEMENT | NAMING CONVENTIONS |
| 1 | Classes | Description name of entity. |
| Browser – For opening the browser page. |
| Read – Reading the data from the excel sheet. |
| Locators – To find the location of web elements. |
| Screenshot – Taking the screenshot of results. |
| Testcases – Checking the functional testcases |
| Smoke suite – Checking the smoke suite testcases. |
| Regression suite – Checking the regression suite testcases. |
| 2 | Interfaces | TakeScreenshot, WebDriver, findByXpath |
| 3 | Methods | Reading details. |
| Taking screenshot.  Providing the result |
| Reports (Extent Reports). |
| 4 | Property | Providing the application properties for selection of respective browsers of given application and the URL. |

## 3.2 INDENTATION

Indentation is one of the most important aspects while writing programs on any programming language. The purpose of code indentation and style is to make the program more readable and understandable. It saves lots of time when we are revisiting the code base.

**It makes code:**

* Easier to modify or enhance.
* Easier to maintain.
* Easier to read and understand Error Handling.
* Error Handling refers to the anticipation.

## 3.3 ERROR HANDLING

* Error handling refers to the anticipation, detection and resolution of programming, application, and communication errors.
* Specialized programs, called error handles are available for some applications.

A development error is one that can be prevented. Such an error can occur in syntax or logic:

* Syntax errors, which are typographical mistakes or improper us of special characters. Are handled by rigorous proofreading.
* Logic errors, also called bugs, occur when executed code does not produce the expected or desired result.

## 3.4 GENERAL GUIDELINES FOR CODING

* Keep codes concise.
* Keep codes stable.
* Make codes that are unique.
* Allow codes to be sortable.
* Avoid confusing codes.
* Keep codes uniform.
* Allow the modification of codes.
* Make codes meaningful.

# 4. TESTING STANDARDS

## 4.1 TESTING STRATEGIES

### Positive testing:

Positive testing determines that your application works as expected. If an error is encountered during positive testing, the test fails.

### Negative testing:

Negative testing ensures that your application can gracefully handle invalid input or unexpected user behavior.

## 4.2 PROJECT ASSUMPTIONS

* You’ll have access to all the resources you need to complete the project, both human and material.
* Project team members will have the resources they need to complete their individuals’ tasks on time, from specialized equipment and software down to electricity during working hours.
* Personnel costs will not change during the project cycle.
* Other material and resources costs will remain consistent throughout the project.
* The overall cost of day-to-day operations will not increase.
* All equipment will be in working condition through the project cycle.
* The scope of the project will not change throughout the life cycle.

## 4.3 INSPECTION RESULTS

Most product inspections yield one of three overall results.

“PASS” – product quality is within pre-determined AQLs (Assured Quality Levels), the product has passed any on-site tests required and it meets any other requirements or regulations.

“FAIL” – the quality of defects or non-conformities more than AQLs (Assured Quality Levels) the product has failed one or more important on-site test, or it has not met a critical requirement or regulation.

“PENDING” – quality is within pre-determined AQLs, the product has passed any on-site tests required and it meets any other requirements of regulations, but there’s some issue the inspector wants to bring to the importer’s attention that was not included in reference documents or specification.

## 4.4 GENERAL GUIDELINES FOR TESTING

* Testing should uncover software defects and improve software quality.
* Testing should be done throughout the SDLC.
* Testing should do both black-box and white-box testing.
* Testing should be preformed by using business logic.
* Testing should be performed effectively and reduce software risks.

# 5. AUTOMATION FEASIBILITY

Feasibility analysis in automation testing refers to a checklist on basis of which we can decide whether we should proceed with the automation of the test cases or not. This checklist consists of various factors upon which automation can be decided and performed.

Selenium is free (open source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc. to create Selenium Test Scripts.

# 6. TYPES OF TESTING

## 6.1 SMOKE TESTING

Smoke test checks if the software build is stable and can be used by the QA team.

## 6.2 REGRESSION TESTING

Regression test checks for any small changes in the code and whether any other existing functionalities are affected due to the newly developed code.

# 7. FRAMEWORK

## 7.1 TestNG

* TestNG is an automation testing framework in which NG stands of “Next Generation”.
* TestNG is a testing framework that can make Selenium tests easier to understand and of generating reports that are easy to understand.
* Using TestNG, you can generate a proper report, and you can easily come to know how many testcases are passed, failed, and skipped. You can execute the failed test cases separately.

## 7.2 MAVEN

* Maven is a build automation tool which is distributed under Apache Software Foundation.
* It is mainly used for Java Projects.
* It makes build, consistent with another project.
* Maven is also used to manage the dependencies.

## 7.3 APACHE POI

* Apache POI is the most used API for selenium data driven tests. POI is a set of library files that gives an API to manipulate Microsoft documents like .xls and .xlsx
* To read and write the excel file in Java, Apache provides a very famous library POI.
* This library is capable enough to read and write both XLS and XLSX file format of Excel.

## 7.4 PAGE OBJECT MODEL (POM)

* Page Object Model, also known as POM, is a design pattern in Selenium that creates an object repository for storing all web elements.
* Page Object Model is useful in reducing code duplication and improves test case maintenance.
* Page Object Model considers each web page of an application as a class file.
* In this case we will use Page Factory to initialize web elements that are defined in webpage classes or page objects.

# 8. BROWSERS

## 8.1 CHROME

* Chrome browser implements the web driver protocol using an executable called “chromedriver.exe”.
* This executable starts a server on your system which in turn is responsible for running your test scripts in Selenium.

## 8.2 MICROSOFT EDGE

* Microsoft Edge browser implements the web driver protocol using an executable called “msedgedriver.exe”.
* This executable starts a server on your system which in turn is responsible for running your test scripts in Selenium.

# 9. 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.