Westpac

**Test Approach Automation Exercise**

Compiled by – Radini Dasanayake

Date – 16-08-2022

Contents

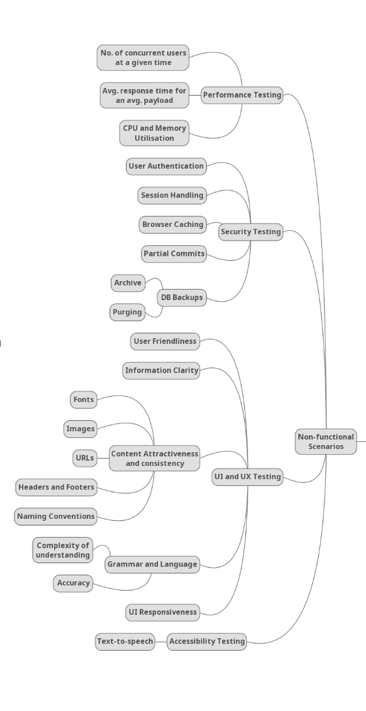
[Test Analysis 3](#_Toc111494983)

[Test Design 4](#_Toc111494984)

[Test Implementation 5](#_Toc111494985)

# Test Analysis

Below are the non-functional testing which we can consider for this application and I haven’t gone through all these at this stage.



I have considered only functional testing scenarios

* Functional Testing
* API Testing
* Positive Test Scenarios

-Login – Authentication-Successful Login

-Register

-Vote for cars

-Verify vote details

-Verify vote count

* Negative Test Scenarios

-Login – Authentication-unsuccessful Login

-Unsuccessful Register

-Unsuccessful Vote

-Incorrect vote count

-Incorrect vote details

# Test Design

|  |  |  |
| --- | --- | --- |
| **Function** | **Test Cases** | **Description** |
| Validate Register | TC\_01 | User gets successfully registered to the Buggy Cars Rating website |
| Validate Login | TC\_02 | User gets successfully logged to the Buggy Cars Rating website |
| Validate profile page | TC\_03 | Use edits additional information and gets them saved |
| Vote for Model | TC\_04 | User navigates to List of all registered models and votes |
| Vote for Model | TC\_05 | User verifies details of the Popular Model |

# Test Implementation

* To demonstrate automation exercise, used Selenium WebDriver with Java and automated positive flows only for the given scenario verification.
* Develop an automation framework that support easy maintenance, scalable, reuse class libraries.
* Used Page Object Model (POM)
* For how to run, follow the instructions in the **README file**.

I have added the same extract here for your reference.

*How To Run Pre-requisites*

1. *Install Java and*
2. *Install Maven*

*Steps*

1. *Clone git repository from https://github.com/radinisachindradasanayake/BuggyCarsRating.git*
2. *Import project to an IDE (Intellij or Eclipse)*
3. *Evaluator can either of two methods to run the project,*

*3.a. Right click on pom.xml --> Run As --> Maven test*

* Followed the Data Driven Testing Framework
* Folder Structure; please refer the **below image** for a brief description.

