**TEST STRATEGY – CAR WASH SERVICES**

**HACKATHON PROJECT**

**Test Strategy**

**Table of Contents**

[1.0 Introduction 3](#_Toc88969293)

[1.1 Purpose 3](#_Toc1287491142)

[1.2 Target Audience 3](#_Toc1640273830)

[2.0 System Description 3](#_Toc1868589822)

[3.0 Test Deliverables 4](#_Toc1566829213)

[4.0 Test Types 4](#_Toc340382310)

[5.0 Multi-Browser Testing 4](#_Toc2114612709)

[6.0 Test Data 4](#_Toc2038676157)

[7.0 Resources Required 4](#_Toc1649909321)

[7.1 Hardware 4](#_Toc580139486)

[7.2 Software 4](#_Toc836185744)

**Test Strategy – CAR WASH SERVICES**

# **1.0 Introduction**

## **1.1 Purpose**

The Test Strategy document outlines the approach for testing the automation workflows implemented for the Car Wash Services project. It defines the types and levels of testing to be performed and ensures that all critical functionalities—both newly developed and existing—are validated effectively.

This document also establishes the scope of testing, which includes the development of test conditions, test cases, test data, automation scripts, and effort estimation. It serves as a communication tool to identify and address testing-related issues early in the project lifecycle.

## **1.2 Target Audience**

• Business Unit (BU) SME  
 • Academy Coach  
 • Technical Trainer

# **2.0 System Description**

The Car Wash Services Automation project is designed to streamline and validate user interactions with online service directories such as JustDial. The goal is to automate key workflows that simulate real-world user behavior, ensuring accuracy, usability, and robustness of the platform’s features. The automation is implemented using Selenium WebDriver with Java, supported by frameworks like TestNG and Apache POI for data handling.

Key Workflows Automated:

**1.Service Discovery Automation:**

* Search for car wash services near the user's location.
* Filter and display the top 5 services based on:
* Ratings above 4 stars
* More than 20 customer votes
* Extract and display:
* Service Name
* Phone Number
* Rating
* Vote Count
* Ensure the results are sorted in descending order of rating.

**2. Form Validation – Free Listing Registration:**

* + Navigate to the Free Listing section.
  + Fill out the registration form with one intentionally invalid input (e.g., incorrect phone number format).
  + Submit the form and capture the error message displayed.
  + Validate the presence and accuracy of the warning message using assertions.

**3. Menu Navigation and Extraction:**

* Navigate to the Fitness category.
* Select the Gym sub-category.
* Extract all sub-menu items listed under Gym.
  + Store the extracted items in a List collection and display them.
  + Validate that all expected sub-menu items are present.

**4. Alert Handling:**

* Detect and handle any JavaScript alerts or pop-ups during form submission or navigation.
* Use appropriate Selenium methods to accept, dismiss, or read alert messages.

**5. Multiple Navigation Paths:**

* + Validate that the same page (e.g., Gym services) can be reached via different navigation paths.
  + Ensure consistency in content and layout across paths.

**6. Back Navigation and Home Redirection:**

* Automate the process of navigating back to the home page from any sub-category.
* Validate that the home page loads correctly and all primary elements are visible.

**7. Search Option Testing:**

* Use the platform’s search bar to locate car wash services.
* Validate auto-suggestions, search results accuracy, and responsiveness.

# **3.0 Test Deliverables**

|  |  |
| --- | --- |
| Project Phase | Deliverables |
| Test Planning | Test Strategy Document |
| Test Analysis & Design | Test Conditions, Test Cases, Test Data, Test Environment, Automated Test Scripts |
| Test Execution | Test Logs, Screenshots of Results |
| Test Completion | Test Summary Report |

# **4.0 Test Types**

The following types of testing are applied to validate the CAR WASH SERVICES workflows:  
 • Functional Testing – To verify that each feature behaves as expected.  
 • Regression Testing – To ensure new changes do not affect existing functionality.  
 • Automation Testing – To execute repetitive and data-driven test cases efficiently using Java,

Selenium, TestNG, and Cucumber.

# **5.0 Multi-Browser Testing**

To ensure cross-browser compatibility, the automation scripts are executed on:  
 Google Chrome  
 Microsoft Edge  
   
This helps validate consistent behaviour across different browser environments.

# **6.0 Test Data**

* Test data is managed using Excel files and property files:
* Excel Files: Used for storing input data and expected results.
* Apache POI: Utilized for reading and writing Excel data.
* Property Files: Store configuration details such as URL, browser type, and credentials.
* XML File: For storing data and reading data purpose.
* Txt File: For unstructured data writing.
* Data-driven testing is implemented to support multiple test scenarios and improve coverage.

# **7.0 Resources Required**

## **7.1 Hardware**

* Personal Computer (PC)

## **7.2 Software**

* Windows 10 and above
* Google Chrome 60+
* Microsoft Edge
* Microsoft Excel
* Java Development Kit (JDK)
* Selenium WebDriver
* Apache POI
* TestNG
* Cucumber (for BDD implementation)
* Selenium Grid