# **Buggy Cars Rating -Test Approach**

## Version 0.1

Author: Paolo Dominic M. Mondelo

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Author | Reviewed By | Reason For Change |
| 12 May 2021 | 0.1 | Paolo Mondelo |  | Initial Version of Test Approach |

## References

|  |  |  |
| --- | --- | --- |
| Item Name | Item Version | Item Location |
| Github Repo |  | https://github.com/paolomondelo/Westpac\_Buggycars |
| Defect List |  | https://github.com/paolomondelo/Westpac\_Buggycars |
| Test Suite Execution Instructions |  | https://github.com/paolomondelo/Westpac\_Buggycars |
| Test Approach |  | https://github.com/paolomondelo/Westpac\_Buggycars |

## Terms & Abbreviations

|  |  |
| --- | --- |
| Term/Abbreviation | Description |
| TBD | TBD |

## Stakeholders

|  |  |
| --- | --- |
| Name | Title |
| Nikhil Hadwale | Product Owner/Stakeholder |
|  |  |

## Delivery Team

|  |  |
| --- | --- |
| Name | Title |
| Paolo Dominic M. Mondelo | Senior Test Automation Engineer |

# **Introduction**:

This document describes the approach, methodologies and objectives in testing the Buggy Cars Rating website. The main objective of UI test automation and manual test execution is to improve quality, to catch bugs, and to create a test automation suite that can be used for regression and smoke testing.

# **Test Objectives:**

The goals for this testing are divided into 3 parts, namely: Functional UI test automation, Functional manual test automation, and Exploratory testing.

**Functional UI Test Automation Goals:**

**Functional test automation is done in this project by automating test cases and by executing it using Java, Selenium, BDD, and Cucumber. Creating an automated UI testing test suite, can help the testers in executing test cases. This will also reduce the time for test execution allowing the QA testers to focus on more important tasks. Here are the objectives of our UI test automation efforts.**

* Create an automated test suite that can be used in regression and smoke testing.
* Create a script that covers the end-to-end scenarios for the Buggy Cars Rating website.
* Create a script to validate the issues found and potential fixes for the found defects/bugs.

**Functional Manual Test Execution Goals:**

**Functional manual testing is used to supplement our test automation efforts and to uncover any hidden bugs and defects.**

* To validate that the solution functions as expected under the expected system behavior.
* To supplement our test automation scripting.

**Exploratory Testing Goals:**

**Exploratory testing helps us in exploring side test cases. It helps us uncover hidden defects through our exploratory and Adhoc testing, with this, it will supplement our test automation, manual test execution, and exploratory testing. With exploratory testing, it will help us to know the basic functionalities of the system.**

* To supplement test automation and manual test execution.
* To conduct reconnaissance / knowledge about a system’s functionality.
* To validate the stability and robustness of the application under different load and scenarios

# **Process Flow:**

# **Diagram Description automatically generated**

# 

# **Test Types To Be Executed:**

## Exploratory Testing:

Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution. It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests.

## Smoke Testing:

**Smoke Testing** is a software testing technique performed post software build to verify that the critical functionalities of software are working fine. It is executed before any detailed functional or regression tests are executed.

## Adhoc Testing:

**Ad hoc Testing** is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application. Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the Software testing technique called **Error Guessing.** Error guessing can be done by the people having enough experience on the system to "guess" the most likely source of errors.

This testing requires no documentation/ planning /process to be followed. Since this testing aims at finding defects through random approach, without any documentation, defects will not be mapped to test cases. This means that, sometimes, it is very difficult to reproduce the defects as there are no test steps or requirements mapped to it.

# **Approach**:

Assuming we captured all required information and processes about the site to be tested, we will proceed to request test approach sign-off, test design, scripting, test execution, test reporting, sending of defects list, handover of artifacts ( repository,scripts, and test cases), and finally test closure.

Order of scripting efforts will be :

* 1st: Login Module
* 2nd: Registration Module
* 3rd: Popular Make Module
* 4th: Popular Model Module
* 5th : Overall Rating Module

Priorities for this work are the following:

* Scripting
* Test Execution
* Manual Test Execution
* Adhoc Testing/Exploratory Testing
* Test Reporting/ Sending Of Found Defects.

In the event we find an issue, we will log a defect and notify the project stakeholders. Found issue can be filed in list.

Once we test automation and testing is complete, test results will be analysed, and communicated to project stakeholders. On review by stakeholders, we will either request testing/test automation enhancements or sign-off.

# Test Preparation Steps:

|  |  |
| --- | --- |
| **Preparatory Tasks** | **Person In-Charge** |
| Get Test Environment | QA |
| Download Test Automation Tools | QA |
| Prepare Test Data | QA |
| Prepare Test Artifacts | QA |
| Ensure access to network, environment, and application are working/available | QA |

# 

# Test Schedule:

|  |  |  |
| --- | --- | --- |
| **Task** | **Week #** | **Duration(Days)** |
| **Install Test Automation Tools:**   * **Java** * **Selenium** * **Cucumber** * **IntelliJ** | **1** | **0.1** |
| **Exploratory Testing** | **1** | **0.1** |
| **Test Design** | **1** | **0.3** |
| **Scripting** | **1** | **1** |
| **Manual Testing & Adhoc Testing** | **1** | **0.3** |
| **Update Test Coverage** | **1** | **0.3** |
| **Test Execution** | **1** | **0.1** |
| **Filing of Defects** | **1** | **0.3** |
| **Summary Report/Sign-off** | **1** | **0.1** |

# Test Tools:

The tools that we will be using for the scripting of scenarios and execution for performance testing will be Java,Selenium,Cucumber, and IntelliJ.

# Entry Criteria

|  |
| --- |
| **Entrance Criteria** |
| Test environment is ready |
| Completion of test preparation tasks |
| Valid test accounts and test data |
| Access and permissions should be granted for applications, network, and environment |
| Planned test automation tools to be used is now decided |

# Test Scope and Data:

## Test Data Preparation

We’ve Identified the following activities for test data preparation:

* Gather and prepare valid users.
* Prepare lists of email address and credentials to be used as test data.

## Test Scenarios Identified :

**Covered Scenarios:**

1. Login
   1. Valid Login
   2. Invalid Login
2. Registration
   1. Profile Changes
      1. First Name
      2. Last Name
      3. Gender
      4. Age
      5. Address
      6. Phone Number
      7. Hobby
   2. New Password
   3. Language
3. Popular Make
   1. Viewing
   2. Voting
   3. Commenting
   4. Moving to the next page
   5. Checking of links
4. Popular Model
   1. Viewing
   2. Voting
   3. Commenting
   4. Moving to the next page
   5. Checking of links
5. Overall Rating
   1. Viewing
   2. Voting
   3. Commenting
   4. Moving to the next page
   5. Checking of links

**Not Covered Scenarios:**

1. Non Functional Testing
   1. Security
   2. Performance
   3. Usability

# Test Environment:

## SUT environment solely for Buggy Cars Rating test execution

Link: https://buggy.justtestit.org/

# Metrics:

1. # of created automated test scenarios
2. # of passed scenarios
3. # of failed scenarios
4. # of defects filed
5. # of executed test scenarios
6. # of blocked/in progress/ obsolete test cases

# Exit Criteria:

|  |
| --- |
| **Exit Criteria** |
| In-scope business processes and systems have been tested against for the test types identified |
| No major, severe, and show stopper issues should be present |
| Test sign-off from project stakeholders |

# Risk Identified

|  |  |
| --- | --- |
| Risk | Mitigation |
| System failure (crash) is possible due to multiple test executions done. Also, due to our running test automation scripts, users might see movements in the system or they may experience performance degradation. | We will be sending email communication to notify users that we are executing a series of tests. In that way we can prevent impact and lessen the likelihood of affecting other team members since it is a shared environment. |