# Test Approach For Buggy Rating Application

**Objective:** The objective of this document is to cover the testing details for the Application such as test cases covered and automation suite details

**Types of Testing:** Test case for the application will be coved under following categories.

Main categories:

**FUNCTIONAL TESTING:** This includes UI as well

1. **User Journey:**

* New user registration
* User login
* Profile verification / Profile update
* Logout

1. **Pages:** To verify that all the pages and details on it are working fine

* Popular Make
* Popular Model
* Overall Rating

1. **Page Details:** User Experience for details on each page such as: Should be clickable, correct (ascending descending) order, UI displayed.

* On Overall Page and Make Page:
* Make
* Model
* Rank
* Votes
* Engine
* Comments

**NON-FUNCTIONAL TESTING**

1. **Performance Testing:** Sauce lab (or Browser stack) can be used to run application on multiple browsers, OS, screen sizes and run parallel tests to increase load / or to use different configuration.JMeter can be use as automation testing tool.
   1. **Load Testing:** measure system performance when application will be used by so many concurrent user or transactions. The system is monitored to measure response time and system staying power as workload increases.
   2. **Stress Testing:** The goal of stress testing is to measure the software stability. At what point does software fail, and how does the software recover from failure?

Also, in case of environment failure, such as low memory or no space, how application is going to behave.

* 1. **Spike Testing:** When workload will be increased beyond normal expectation for a short period of time and how application is going to handle this.
  2. **Soak Testing:** How application going to behave in normal and keep running for a longer duration (Soak test results can be displayed in Kibana using with elastic search, if required).
* **Test Environment:**
  + **OS** : Windows XX, Mac XX, and XXX
  + **Browsers**: Google chrome XX, Firefox version XX and other browsers XX
* **Testing Tools:**
  + Tracking Tool (Jira or other tools), Chrome (or other browser), Sauce lab Or Browser Stack (for Cloud based Testing)
  + **Automation Testing Tools:** Visual Studio, Specflow, Nunit based framework using selenium, Chrome, Sauce lab Or Browser Stack (for Cloud based Testing)
  + **Performance Testing:** JMeter
  + **API Testing:** Soap UI or Rest sharp or Postman.
* **Continuous Integration and Release management:** Jenkins (or Bamboo or Teamcity)
* **Defect List and status:** Bug report can be found in another attached document

----------------------------------------------------------------------------------------------------------------------------------------

**AUTOMATION TESTING Framework Details:**

Testing Tools, Methodologies, and languages:

1. Selenium
2. Specflow
3. Nunit
4. C#
5. Gherkins for writing feature files

Features of the framework:

* Selenium: cost effective, supports range of multiple language and browses.
* Specflow: supports Behavior Driven Development (BDD). BDD increases and improve collaboration. High visibility.  Supports .NET Framework. Latest with Context Injector.
* Easy to understand: Defining Application behavior in plain meaning English text using Gherkin language, easy to understand for anyone.
* Flexible: Can be executed on any browser. Can be Integrate **Sauce lab** or **Browser Stack** for **cross browser Testing**.
* POM: Using Page object model wo keep code clean, easy to understand, more durable and comprehensive.
* Open Source: No extra cost.
* **Automation Framework Structure.**

Diagram

Description automatically generated

* **Workflow:**

Diagram

Description automatically generated

OutputReport HTML can be found with Failed Screenshots:

“{USER Repo}\BuggyCarsUiAutomation\bin\Debug\net461Output”

Graphical user interface, text

Description automatically generated

**Current Status:** Written test cases for Pages and User Journey:

* Login and Logout
* Registration of a new user
* Verify popular model.
* Verify popular make.
* Verify Overall Rating

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |