Test Approach

Testing approach for Buggy Cars web application is bit unique. Usually, functional requirements are known upfront for the system/application at test. Whereas in here, the approach starts from reverse engineering the web application. At a high-level, following steps are performed to test the system.

1. Prepare a functional requirement summary by analysing various navigations in buggy cars website.
2. Outline the high-level business contexts and various business rules associated with them
3. Manually test the web application and validate against the expected behaviour.
4. Prepare bug report with varying criticality and provide as much detailed information as possible.
5. Implement automation tests in Java/Groovy with Selenium

Functional Requirement Summary:
As a product owner I need a web application where customers can view ratings of various cars and the most popular car model and manufacturer. The application should have Login/registration options so that only legitimate registered customer should be able to rate the cars and provide comments.

High Level Business Contexts:

- Anonymous Browsing
- Registration
- Login
- Voting
- Overall ranking summary

Critical functional flows of the application:

1. Any user (anonymous/registered-logged-in) should be able to able to view the most popular Car Make/Model and overall Summary reports
2. Application should allow registration of new customers and prevent duplicate registration
3. Registered customers should be able to Login and Vote with some review comments
4. Popularity ranking of the cars should be updated properly on every vote from a customer.