**Device API Testing Report:**

This is automation report for execution of car API testing scenarios

This report consists of 3 parts:

1. High-level overview of feature tests execution
2. Detailed Scenarios execution of respective features.
3. Bug Report
4. Testing Technique
5. **High-level overview of feature tests execution**

**A picture containing chart

Description automatically generated**

Table

Description automatically generated

1. **Detailed Scenarios execution of respective features**

Please refer below table-

A picture containing text

Description automatically generated

1. **Bug Report**

During Testing found following bugs, please find below defect status:

|  |  |  |  |
| --- | --- | --- | --- |
| **Defect ID** | **Summary** | **Status** | **Severity** |
| 1 | Sometime API takes lot of time to respond, need to restart many times | Open | Critical |
| 2 | Disconnect API return true in message for device which is already disconnected | Open | Major |
| 3 | Set name API allows special character, number in name string | Open | Minor |
| 4 | Currently API is returning success true and false message is response .  Context specific success and failure message will be give more clarity of response | Open | Minor |

**Testing Technique:**

* In current implementation I have done rest API testing using **BDD cucumber and rest assured** .

I have covered following testing technique in current implementation

1. Health check of each API with valid status code.
2. Get response and validation.
3. Shema validation of get API to ensure structure and layout of response.
4. Data updation using post request.
5. Integration between API as /brightness, /color ,/name post API updation reflect to state API.

Other Technique that is not covered:

* Load Testing
* Contract Testing

**Techniques to improve your QA automation:**

**Test Pyramid:**

* Less end-to-end testing lower maintenance cost
* Less Bug
* quicker recovery time to identify and fix bugs

**A close up of text on a white background

Description automatically generated**