1. **INTRODUCTION**
   1. Purpose of This Test Plan
2. **MAMNUAL TESTING**
   1. Scope
   2. Items to be Tested
   3. Deliverables
3. **AUTOMATION TESTING**
   1. Scope
   2. Test Approach
   3. Automation Requirements
   4. Tools
   5. Deliverables
4. **API TESTING**
   1. Scope
   2. Test Approach
5. **SECURITY TESTING**
   1. Scope
   2. Approach
6. **PROJECT CHALLENGES**
   1. Introduction
   2. Challenges/Constrains
   3. Mitigation

**APPENDIX A: REFERENCIES**

1. **INTRODUCTION**
   1. PURPOSE OF THE TEST PLAN DOCUMENT

This Test Plan Document documents the necessary information required to effectively define the requirements of the automation challenge and the approach I took to accomplish this challenge.

The purpose is that anyone that goes through this document will easily understand what requirements of the challenge is, the approach taken to accomplish this challenge, the expected deliverables, deliverables met, deliverables not met with reasons, the tools used, the challenges encountered and how they were handled and finally how to locate the deliverables.

1. **MANUAL TESTING**
   1. SCOPE

The scope of the manual testing is to write 5 Test Case Scenarios to test the Search Feature of the SUT. Each Test Case Scenario does an end to end of what a user would do while using the search feature.

The 5 TCS which is DLR1, with Test Cases, Test Steps, Expected Result, Actual Result …, can be found in the Test Case Document.

2.1 ITEM TO BE TESTED

The item to be tested is the Search functionality of the SUT. For each functional TCS, a negative test scenario is described (The Negative TC are included in the TCD), also, a brief security and API test descriptions is done (See the appropriate sections for details)

* 1. The Deliveries for the Manual Testing are:

1. 5 Functional TC DLR2 (See TCD)
2. 5 Negative TC DLR3 (See TCD)
3. 2 Security Testing description DLR4 (See Security section on this TPD)
4. API testing strategy for the search feature DLR5 (See API section on this TPD)
5. Test Plan Document
6. **AUTOMATION TESTING**
   1. SCOPE

Automated Test Scripts are created for the above 5 functional manual testing. The automation was done using WebDriverIO and Cucumber Framework (DLR6)

* 1. TEST APPROACH

To achieve the test automation, WebDriver IO tool was used as required with Cucumber framework. I used Page Object Model design pattern.

* 1. AUTOMATION REQUIREMENT

Listed below in bullet form are the automation requirement base on the challenge. I have also listed these requirements in a CHECK LIST with status to show if each requirement was achieved or not.

* All test inputs were data driven for easy maintenance and reusability (DLR7).
* Test execution can be triggered from Jenkins (DLR8)
* Automation run report generated (DLR9)
* Log to Console and Report number Item returned and top 3 reviews (DLR10)
* Report should clearly generate relevant metrics that shows the effectiveness of automation. (DLR11)
* Capturing execution time per test and Screenshot to view failed TC (DLR12)
* Report should include the preceding n runs where n 3. (DLR13)
* All Delivery should be done via Github, to include Readme, (Video capture) and test report (DLR14)
* Test should be architected such that they can be executed iteratively multiple times with different set of test imputes (DLR115)
* One of the 5 testcases should run with 3 different test set (DLR16)
* Use design pattern so that test can be run independently (DLR17)
* Use standard practice (DLR18)
  1. TOOLS

The following tools were used for this project

* WebDriverIO
* Cucumber
* Selenium Stand Alone
* Allure Reporting tool
* Jenkins CI/Reporting tool
* Github Version Control tool
* Visual Studio Code IDE
  1. DELIVERABLES

The following are the deliverables.

* Automated test script pushed to Github
* Readme file of the Github repository
* Video capturing of the automated test run
* Automated Run Report
* Report should show relevant metrics
* Delivery Requirement Checklist DLR

1. API TESTING
   1. SCOPE

The scope of the API Testing is to describe how I would perform the above functional testing using API. Listed below are the test I would take.

4.2 TEST APPROACH

* To perform this testing using API, I will use SOATest/SoaPUI or any other API tool that is available.
* I would need the Routes and Endpoints to the service call with necessary authentications.
* The API request will be parameterized
* I will pass the parameters to my Get API call end point request and use assertions to validate that I got back the expected response in terms of product, numbers of result and status code.
* Some of the json/xml step responses will be used as parameter input for the next step.
* This will be done until all steps are covered for each test case scenario
* Final Assertion/validation will be done for each scenario.

1. SECURITY TESTING
   1. SCOPE

The scope of the security testing is to describe how I will test to prevent security related attacks through the search feature

* 1. APPROACH

Basically, my security testing will be done to make sure users are not able to

1. Prevent user getting information they shouldn’t have access to, like personal or company information/Upcoming Product or Services that should not be made public yet. Test that the search features does not accept SQL query injection that could attack the database or input into the database.

* '; SHOW TABLES; -- (to verify that something like this does not output unnecessary information.)
* '; DROP TABLES; -- (to verify that the search feature does not accept something like this that could be destructive to the database)
* '; UPDATE TABLES; -- (to verify that users cannot do something like this to writ to the database through the search feature.

1. PROJECT CHALLENGES
   1. INTRUCTION

The purpose of this section is to state some of the challenges/constrains that affected me directly or indirectly in the execution/output of this project.

* 1. Challenges/Constrains
* Time: The major challenge was time, Due to the demands of my current job and family responsibility I was unable to perform the project and deliver all deliverable as soon as I would have love to.
* Software Hardware Issues: There seems to be some compatibility issue between the software on my laptop and the Allure plugins, this affected the rendering of screenshot. Also, I had some hardware space issues on my laptop.
* I had some session ID issues which affected Jenkins build