Antonio Navas

Antonio\_navas40@hotmail.com

Abstract

Framework showing the possible implementation of UI & API testing based on the information provided in the assessment

Test Automation Framework

Table of Contents

[**Executive Summary** 2](#_Toc95777801)

[**Proposed Solution** 2](#_Toc95777802)

[**Framework Structure** 2](#_Toc95777803)

[**Configurations and Documentation** 2](#_Toc95777804)

[**Main Java** 3](#_Toc95777805)

[**Test Java** 4](#_Toc95777806)

# **Executive Summary**

# **Proposed Solution**

Framework with API and User Interface (UI) support.

## **Framework Structure**

### **Configurations and Documentation**

At the root level, the framework would include configuration files for each platform using it plus detailed documentation.

Text

Description automatically generated

### **Main Java**

Under documentations, the “src” directory which includes the “main” and “test” subfolders.

* Under main, users can find every package and class related to the frameworks.
* Under test, users can find every package and class related to the tests.

Text

Description automatically generated

As shown in the image above, under “\main\java\”, the user can find the framework substructures:

* The folder “apiEngine” will include interfaces, endpoints, routes, and classes used to support API automation.
* The folder “uiFramework” will include Page Objects and Selenium classes used to support the user interface automation.
* The folder “cucumber” will include Scenario Context and Test Context.
* The folder “dataProvider” will include Configuration and Jason Data Readers, Data Provider Helper, and other data related classes.
* The folder “enums” will include enumerators classes.
* The folder “managers” will include File Reader Manager and other manager classes.
* The folder “testDataHelpers” will include Constants and other test data helper classes.
* The folder “testDataType” will include pojo classes used to serialize and deserialize JSON, XML and other test data files.
* The folder “utilities” will include Settings and Test Utilities classes and other utility classes needed.
* The folder resources will include logs, logs properties and other resources.

### **Test Java**

The subfolder Test Java will include cucumber specific folders.

Graphical user interface, text

Description automatically generated

As shown in the image above, under “\test\java\”, the user can find the framework substructures:

* The folder “features” will include all the feature files with the test scenarios written in Gherkin.
* The folder “hooks” include the different “hooks” (or filters) used to navigate the automation as different users and/or different flows.
* The folder “rough” will include any java class needed to create end to end scripts to understand the way the functionality works. Those classes can be deleted after the scenario is migrated to the rest of the cucumber framework (step definitions, pages, etc.).
* The folder runner will include the test runner (or test runners) used to run the test automation.
* The folder steps definitions will include the classes linking the functionality as described in the feature file to the different API classes and or UI classes inside the framework.
* The folder resources include drivers, json files, xml files or any other file type used or created by the automation.