TECHNICAL ASSESSMENT for IntelligentReach 23/02/2023
Isaac Michael Carini

Section 1: Your approach to work

- **What is QA and Testing to you, how do they differ?**

QA is a big umbrella, its purpose is to assure the highest possible quality by defining and implementing standards on a given product/process, testing is part of the QA process, its aim is to find faults and issues, it makes sure parts of the whole are doing what they're supposed to.

- **How do you approach testing a black box system?**

First step is to gather the requirements, what is expected of this 'object' that we're testing? what are the specifications of said system?
Once we have as much information on how it 'should' behave we can start investigating to tell whether or not an input yields the expected output, identify test scenarios (If I click here I m supposed to get there, if i login, the icon on the top should change etc) all the while documenting steps, expectations/actual output, and any further findings.

- **What steps would you expect to be involved in test development? Which stakeholders would you need to involve in relation to specific parts pf the process?**

Reviewing the objectives and requirement specifications, the product owner and requirements specialist are ideal when trying to understand the subject matter and make sure all is clear in regard to the client expectations, develop test plans and design test cases, execution of tests, reporting of results, signing off once there's confidence in the quality.

- **When supporting a wide range of different browsers + devices. How would you describe which to focus most effort towards?**

The focus should always be on the browsers/devices that are most commonly adopted, if the sample audience is predominantly (51% of total users in system) using Chrome as a browser and Android as device, those are the focus, with whatever comes after in importance (i.e. 39% for the second most used browser etc)

- **Under what circumstances is it useful to automate a test rather than performing it manual test-case runs? When would automation not be the best path?**

When we are expecting to be repeating similar tests over and over with barely if no difference between them, or due to their nature they're prone to encountering user error, the answer is to automate.
There are cases when the framework doesbt provide the best approach and the best way is through manual testing, in that case we can avoid automation, unless, that new change becomes a new

expectation, in which case it would be ideal to expand the framework and include the new 'change' into the tool.

- **What features would a great automation tool/framework have?**

Reliability (it needs to be actually saving time and effort while not wanting in quality), scalability (being able to add new tests to whatever new feature is being added without it being too difficult or time consuming), being clear and easy to read through, so that any person who approaches the tool can clearly understand what's happening and why, as well as reusability.

- **How would you know whether automation has been successful for our business?**

I f it acts as an early warning system for defects and delivers value, measurable in time saved in tests execution and actual productivity. To put it simply if its saving time and money, without losing in the actual quality, then automation is successful.

Section 2: Execution

Task 1

Below is link to Google Search we would like you to test it.

https://www.google.com/Methodologically demonstrate your ability using Black box testing or any other testing heuristics you are familiar with. Write down your thought process and any findings.

Test Plan: Google Search Engine [https://www.google.com/]

Purpose:
The purpose of this document is to report the results of UI functionality and usability tests on the Google search engine. The tests are intended to verify that functionality is working as intended, and to identify any possible defects that may impact the quality of the experience.

Test Environment:
- OS: Windows 10
- Browser: Google Chrome Version 110
- Test Automation framework: Selenium WebDriver
- Programming language: C#

Test Cases:

Test Case 1: Search for Valid Keyword
Preconditions: User is on the Google Homepage.
Test Steps:
- Enter a valid keyword in the search bar.
- Click the search button
Expected results:
- The search page is shown, containing relevant results related to the chosen input.
- The user has the ability to navigate to different pages as a result of the search.

Test Case 2: Search for Invalid Keyword
Preconditions: User is on the Google Homepage.
Test Steps:
- Enter an invalid keyword in the search bar.
- Click the search button
Expected results:
- The search results in a page showing that no results were found for the entered keyword.
- The user has the ability to navigate back and start with a new search.

Test Case 3: Search using voice input
Preconditions: User is on the Google Homepage.
Test Steps:
- Click the microphone icon in the search bar
- Speak using a valid keyword.
Expected results:
- The search page is shown, containing relevant results related to the chosen input.
- The user has the ability to navigate to different pages as a result of the search.

Test Case 4: Search using Image link input
Preconditions: User is on the Google Homepage.
Test Steps:
- Click the image icon in the search bar
- Insert an image path and click search.
Expected results:
- The search page is shown, containing relevant results related to the chosen image.
- The user has the ability to navigate to different pages as a result of the search.

Test Case 5: Verify auto-completion suggestions
Preconditions: User is on the Google Homepage.
Test Steps:
- Enter a valid keyword in the search bar.
- Wait for the auto completion suggestion tab to appear.
Expected results:

- The auto completion bar is displayed, containing relevant suggestions related to the chosen input.
- The user has the ability to choose from multiple suggestions by clicking on them.

Test Results:

The table below summarises the results of the testing applied to the Google Search engine:

Test Case	Result	Comment
1. Search with Valid Input	Pass	n/a
2. Search with Invalid Input	Pass	n/a
3. Search using voice Input	Pass	n/a
4. Search using image link as input	Pass	n/a
5. Verify autocompletion suggestions	Pass	n/a

Task 1

Setup a few Automated UI tests in C# and any desired automation framework for a few test cases for Googles Search Engine.

Document clearly each phase including setting up.

https://www.google.com/

Google Search Engine Automation Framework

Pre-Requisites:

- Visual Studio Installed
- Selenium WebDriver package installed
- Selenium Chrome Webdriver package Installed
- NUnit package Installed
- Nunit3TestAdapter package Installed

Setting up:

1. Right-click on the project in the solution explorer and select “Manage NuGet Packages”
2. Make sure to install all the packages mentioned above

Overview:

This automation framework is designed to automate the testing of the functionality of the google home page. This framework is built in C# and uses Selenium Webdriver and NUnit. The browser chose for these tests is Chrome Version 110

Components:

The framework is comprised of the following components:

- Test cases – collection of test cases.
- Actions class – simple library for common actions.
- Config class – Test data collection.

Test Cases:

The Framework contains the following test cases:

- Verify search bar functionality when using valid keyword as input.
- Verify search bar functionality when using invalid keyword as input.
- Verify correct functionality of Dark Theme button in settings.
- Verify correct functionality auto-completion suggestion tab becoming visible.

Test Data:

- Valid Keyword.
- Invalid Keyword.
- Invalid Keyword expected returned string.
- Auto-completion suggestion search keyword.

Test Execution:

The test script is used to execute the test cases using NUnit framework, which will report the result of the run tests. The tests can be located and run by selecting the window "Test Explorer".
The script acts as follows:

1. Launch the chosen browser and navigate to the main page.
2. Execute test cases.
3. Close the browser.

Conclusion:

The google search bar automation framework is a testing framework that covers some of the main functionalities of the main page.

Code available at:

https://github.com/NorthChild/TechnicalAssessment/tree/main/Framework_IntelligentReach/SetUpEnv