Challenge #1: **TEST PLAN**

**Objective**

The main objective of this Test plan is to outline what the new automation tester will be doing on automating the regression test for Travel Insurance page.

**Scope**

Verify the functionality of Travel Insurance page: Focus on testing the comparison result page, make sure all these functions work properly.

Specifically: In this test, we will check the function ‘find travel insurance for single trip’

**Out of Scope**

These are out of scope:

* The rest functions of the page
* Database logical
* Website Security & Performance

**Testing Strategy**

Build a simple Automation FW with Selenium, Page Object Model and Java

**Resource/Responsibilities**

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Hong Dinh | Automation QA Engineer | * Create manual test cases, evaluate the ability of automation for them * Building frameworks * Implement the test script |

**Risks**

* Not enough time
* Due to time limitation, so the test is written very simple, some items I still not added in the test for example: Report, common lib, CI/CD integration,…

**Environment**

* Browsers: Chrome, Firefox, Safari
* OS: Mac OS, Window 10

**The tool(s) you would use to test it and the reason**

* Selenium: Selenium is a very popular powerful tool to perform functional and regression testing and completely free to download and use; Selenium provides WebDriver API which enables tester to write their test in many programming languages: Java, C#, Python, PHP, …
* Java: Java is the safest choice to start the project from scratch for the new automation tester because there is a big community works with Java, so you can easily to find the code examples or troubleshooting the issue. Java is also the top priority for each new release of Selenium
* Page Object Model: Easy to use, easy to understand and easy to maintain – it reduces duplication of code thus reduces the test maintenance cost.

**The architecture of the test script in that tool**

* Page: There are 2 class pages(pageComparison.java, pageTravelResults.java) to store all web elements locator in the system
* Test: class assignment2.java to store all sequence of test steps
* Main.java: works like test scenario, to call test steps from assignment2.java

A picture containing indoor, white, photo, looking

Description automatically generated