Test Automation Basics

First, I need to clearly understand the web application of an online shopping site. In here, I’m going to explore the functionality of the application manually. I need to understand what are the pages interacting in the web application, what are the elements in the web application, how the interactivity of these elements and I need to clearly identify what to automate and what we cannot automate in the web application.

Then I’m going to document the test cases and clearly identify the test cases for automation.

Now, I need to select the programming language and tool to do the automation. Since I’m familiar with Selenium with Java, I’m going to use it. Also, I need to select suitable tool to do this. I select Eclipse and I’m going to use TestNG to structure test cases. In here I can create a project as a maven project and I can add relevant dependencies to the pom.xml file and then can start implementing the framework.

Now I can implement the framework. We can utilize src/test/java folder to add test cases. To improve readability, re-usability, and maintainability I use page object model(POM) design pattern. According to POM, we write locators and actions in a separate class and we implement test cases in separate class. That means, we can create separate folders in src/test/java folder as ‘page_objects’ and ‘testcases’ and include those classess. Also, we can include common codings in each class into base class. Under the project, we can create a separate folder called ‘testdata’ and then we can add data use to automate the site in that folder. Also, in the framework, we can create logs, reports, screenshots folders to utilize in the project.

In this way, I can create an automation framework. Now, first I can implement login testcase. To do that, I need to identify navigation steps to login, and what are the elements in that login page and understand what actions we can done in that page. Then I need to create a class called ‘LoginPage’ in the ‘page_objects’ folder. Then I should add constructor, locators(web-elements) and actions in that class. Then I need to create Login_Test class in ‘testcases’folder and now I can do validations on that class. Likewise, we can implement other functions as well in this manner.

According to my knowledge, above steps, I’m going to use to create a test automation framework.