|  |
| --- |
| DXC Technology |
| e-Commercial Shopping Automation Test |
| Version: 0.1 |

|  |
| --- |
| Anh Duong  7-11-2018 |

# Business Objectives

Automation of e-commerce Shopping web site <http://automationpractice.com/index.php>

# Scenarios

* Scenario1: Create account with valid data
* Scenario2: Create account with invalid data
* Scenario3: Sign-in with valid data
* Scenario4: Sign-in with invalid data
* Scenario5: Shopping Cart
* Scenario6: Basic search
* Scenario7: Advanced search

# Activities & Timeline

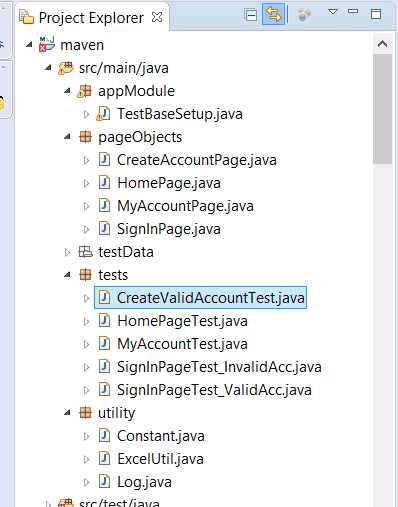
|  |  |  |
| --- | --- | --- |
| **Activities** | **Status** | **Due Date** |
| Develop automation test framework | Completed | 5-Jul-2018 |
| Enhance automation test framework | In-progress | TBD |
| Develop & execute scenario1 | Completed | 13-Jul-2018 |
| Develop & execute scenario2 | Completed | 13-Jul-2018 |
| Develop & execute scenario3 | Completed | 10-Jul-2018 |
| Develop & execute scenario4 | Completed | 10-Jul-2018 |
| Develop & execute scenario5 | In-progress | 26-Jul-2018 |
| Develop & execute scenario6 | Not-started | 02-Aug-2018 |
| Develop & execute scenario7 | Not-started | 02-Aug-2018 |

# Automation Test Framework Design

* Page Object Model
* Modular Driven Framework
* Functions Parameters
* Constant Variable
* Data Driven Framework
* Log4J
* Test NG Reporting
* User Defined Function
* *Object Repository – coming soon*
* *Exception Handling – coming soon*
* *CI – coming soon*

# Framework Introduction

* Test tools: Selenium web driver 3.x, TestNG, AutoIT, Apache POI APIs, JUNIT, JSON
* Programming language: JAVA
* appModule: Modular driven concept that stores all user defined functions, common and re-usable methods.
* pageObjects: Page Object Model that contains functions and objects of each page. Input is web drive and output is class object of next page.
* Tests: highest layer of POM where each test scenario having its class. These tests will be executed within TestNG test suit, testing.xml
* testData: Data Driven that allows users to separate data and test using excel and apache POI APIs. If data or functionality change, no big impacts on the others. Test cases could be executed with different data sets.
* Utility: contains classes that provide Constant Variable, Excel Util and Log4J.



# Source Code

* Git hub: <https://github.com/anhtduong/automation>
* Source Code: <https://github.com/anhtduong/automation/tree/master/maven/src/main/java>
* Log File: <https://github.com/anhtduong/automation/tree/master/maven/logfile.log>
* TestNG Report: <https://github.com/anhtduong/automation/tree/master/maven/test-output/E-commerce%20Shopping/Test.html>

# How to Run Test

* Clone or Download the copy from Git hub
* Configure Gecko driver path in Constant.java
* Setup Java, Selenium, TestNG, and APIs