

Read me

#### System Design:

DataAccess: Contains one Singleton Class (XMLDataReader) that creates one Document object with all the needed data to be read across all Testcases

Report: Singleton Class that Creates one Test Report across all Testcases

#### TestCases:

- Class (TestCaseTemplate) that defines all test cases format and needs to be implemented by any test case
- Testcases Scenarios – Each class represents a test case

#### SeleniumInterface: Contains

- Class (Actions) that has all needed Selenium steps
- Class createWebDriver that opens the driver for the browser chosen by the user in the XML data file

#### XML Data Holds list of Data objects that

- First Data Object Holds the browser chosen by user
- Second Data Objects holds list of Web Elements to be used
- Third Data Objects holds Testing Data Tester will use

#### Input/Output Paths:

Inputs (TestAutomationData.xml), Drivers (CH,FF and IE), and output (HTML report) are all located under: \\src\\main\\resources

#### Notes:

1. One parameter in Step 3 in SampleTestcase 4 is wrong , to Test Failed testcases and Error Handling.
2. Tsester has to Enter the Text of Browser from these options Only as the same Text {Chrome,Firefox,IE}
3. Need to make sure that TestNG plug in is installed in your IDE

Evaluation Criteria	Evaluation Score (Done, Partially Done, Not Done)
Script is Modular	Done
All Dependencies are included (or on Maven)*	Done
Test Data are in External Files*	Done
Object Identifiers (locators) are in external file*	Done
Execution Browser is managed by the user*	Done
Browser Support – IE	
Browser Support – CH	Done
Browser Support – FF	Done
Junit/TestNG Tests	TestNG - Done
Execution Report	Done
Failure Recovery	Done
All Test Scenario Steps are covered	Done

Solution Description and JavaDoc	Done
Extra 1: Selenium Grid Support**	
Extra 2: Keyword-driven Framework**	