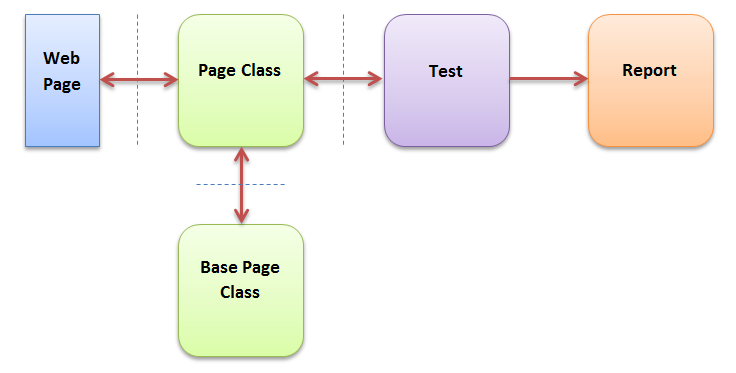
**Automation Framework UserGuide**

**1.Introduction**

A Selenium Maven project with Data Driven Framework designed by Page Object Model(POM) using Page Factory and TestNg to develop Test Cases

As per POM, a separate class is maintained for each web page and each class holds the functionality and members of that web page. Separate classes for every individual test



**2.Advantages**

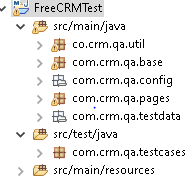
**1.Less Code Maintenance:** Since the test classes are separate from the pages (containing web elements and the actions), updating the code is very easy if any web element changes.

**2.Readaibilty of Code:** One can easily read through the project and test scripts due to a fine line between the test classes and the different web pages.

**3.Framework Structure**

**3.1Packages Overview**

Packages are used in Java to prevent naming conflicts, to control access, to make searching/locating and usage of classes easier.



Following are the packages used in Automation Framework:

|  |  |  |
| --- | --- | --- |
| **Package Names** | **Classes/Files Included** | **Packages Description** |
| com.crm.qa.base | TestBase.java | It contains functions to load Config properties, initialize webdriver |
| com.crm.qa.config | config.properties | This file is used to define global environment variables |

|  |  |  |
| --- | --- | --- |
| com.crm.qa.pages | Page1.java  Page2.java  Page3.java | It contains all the pages (contains web elements and methods) of Application Under Test(AUT) |
| com.crm.qa.testdata | TestData.xlsx/.xls | It contains all Test Data maintained in an excel |
| com.crm.qa.util | TestUtil.java | It contains all generic functions to be used in the framework |
| com.crm.qa.testcases | Test1.java  Test2.java  Test3.java  TestRunner.java | \*It contains all testcases to be executed for each pages.  \*A runner class can also be used to execute test classes (contains testcases) |
| com.crm.qa.testcases | Testng.xml | It is an xml file wherein test classes can be grouped together to execute as suite |

**4.Reporting**

Currently, this framework generates TestNG html reports. The scope of the reporting can be maximized by using extend Report.

**5.Technologies Used**

• Selenium WebDriver - v3.14 (Open Source)

• JDK 1.8 (Java Development Kit)

• TestNG (Test Unit Framework)

• Maven (Build Automation Tool)

• Apache POI API (Read-Write utilities for Excel - Test Data Handling) Eclipse/IntelliJ (Java Editor)

• Browser - Google Chrome/FF