**Maven Project :**

Maven provides a comprehensive model for projects which is reusable, maintainable, and easier to comprehend plugins or tools that interact with this declarative model.

Maven project structure and contents are declared in a xml file, pom.xml, which is the fundamental unit of the entire Maven system.

**Src/main/java**

**com.upgrade.qa.base**

**TestBase.java** : This is a java class which contains the initialization of config .properties file and also a Initialization method which initializes the driver , set the property of the driver and does below

1. Maximizes the window
2. Delete All cookies
3. Page load timeout
4. Implicitly wait for all elements to load in a page
5. Go to the URL/landing Page/Start Page

Note : This is the parent class of all other classes declared below and will inherit the properties of Base class in the child classes using java OOP Inheritance feature.

**com.upgrade.qa.config**

**Config.Properties** : This is a property file that contains Environmental variables in a key value pair format(so that variables can be handled and maintained properly in a repository)

**com.upgrade.qa.pageobjects**

**StartPage.java**

**OfferPage.Java**

**InformationPage.java**

**LoginPage.java**

**LogoutPage.java**

Above classes contains all the Web Elements of different web pages which are located using Selenium locators (id/className/Css Selector/xpath/name/Linktext/Partiallinktext) and corresponding Web methods / actions that those web element does (Like Clicking a button / Sending a text too text field etc)

**com.upgrade.qa.testdata**

**testdata.xls :** This is an excel file that contains test data which selenium requires to input in a web page.

**com.upgrade.qa.util**

**TestUtil.java :**This java class contains all utility methods declared in static so that the test cases can easily access the most commonly used methods/variables/constants

**Src/test/java**

**com.upgrade.qa.tests**

**StartPageTest.java**

**LoginPageTest.java**

**InformationPageTest.java**

**OfferPageTest.java**

Above classes contains TestNG tests with setup and teardown methods.They use 3 TestNG Annotations mainly such as AfterMethod/BeforeMethod/Test. BeforeMethod annotation executes before execution of actual test case and AfterMethod Annotation executes After the Execution of every test case method. And this happens for all the test cases with “Test” Annotation in a class.

**Drivers**

This contains the driver executable file for different browsers(chrome in this case).

**Test-output**

TestNG generates report in this folder automatically, where we can verify the execution status in detailed HTML format.

**Application.log**

This is a log file generated using Apache log4j and gives the details execution log for debug purpose

**POM.xml**

A Project Object Model or **POM** is the fundamental unit of work in **Maven**. It is an **XML** file that contains information about the project and configuration details used by **Maven** to build the project in the form of dependencies.

Below are the list of dependencies used for this project

1. Selenium

2.TestNg

3.Log4j

4.Apache poi

**Testing.xml**

Maven can be used with any of the continuous integration systems. The plugin **maven-surefire-plugin** is used to **configure and execute the tests**. Here the said plugin is used to configure the testng.xml for the TestNG test and generate test reports.The plugin **maven-compiler-plugin** used to help in compiling the code and using the particular JDK version for compilation.

**Execution Process :**

**Using Eclipse:**

You can run all the test cases together using testing.xml file or can individually run the Test classes in eclipse

**Using run.bat :**

**Prerequisite : Install Apache Maven on your machine**

Go to project directory in terminal/command prompt —> **e.g.cd /Users/alekhya/eclipse-workspace/UpgradeLoanUiTestClient**

And then input —> **mvn clean install**

**Or**

Execute the run.bat file (it automatically launched the terminal and then runs the test cases)

Note : Failed test cases can be re executed by using testng-failed.xml file under test-output folder.

**NOTE \*\*\***

**Below are the local paths used for developing automation script , kindly update below paths while executing in your local.**

/Users/alekhya/eclipse-workspace/UpgradeLoanUiTestClient/src/main/java/com/upgrade/qa/config/config.properties

/Users/alekhya/eclipse-workspace/UpgradeLoanUiTestClient/src/main/java/com/upgrade/qa/testdata/testdata.xlsx