TEST AUTOMATION FRAMEWORK

USING JAVA, SELENIUM AND TESTNG

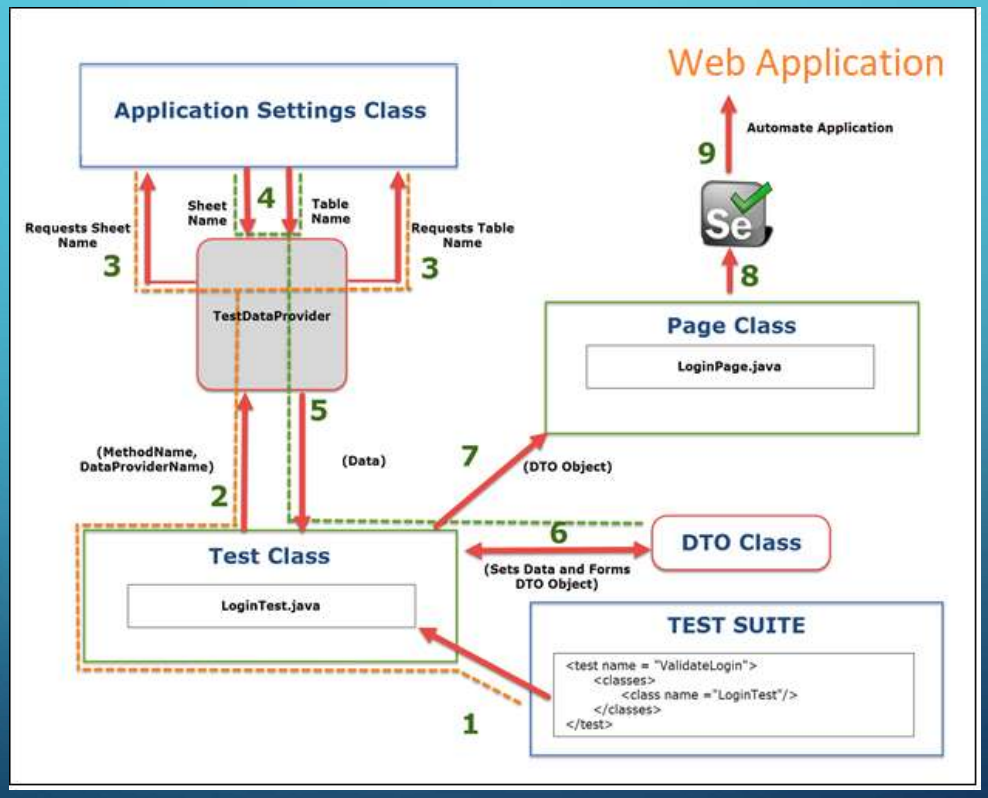
BY: IMTIAZ AHMED

ENOSIS SOLUTIONS

INTRODUCTION

- The Automation framework is a skeleton created as a generic Web UI automation framework. The framework primarily uses JAVA as the programming language. Selenium library is used for the web automation. TestNG framework is used for configuring tests and test result readability. This is a data driven framework where test data is collected from an external source (Excel sheet).
- This is a hybrid framework with multiple advantages:
- The framework allows data driven approach to testing. Data can be stored in a separate file and fetched as needed
- The testNG framework allows a standardized workflow for test case creation, maintenance and test result analysis
- Selenium provides pre-build necessary libraries as well as driver to perform actions on web applications

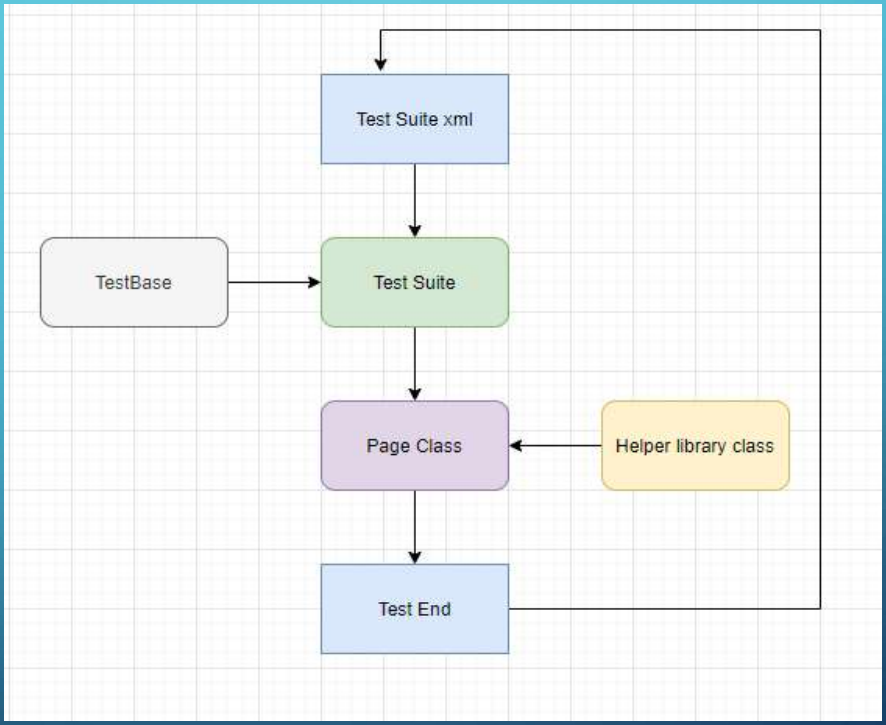
OVERALL ARCHITECTURE



OVERALL ARCHITECTURE(CONTINUED)

- The architecture uses POM (Page object model) for the framework. Maven POM file keeps all the dependencies and environment variables for the framework.
- All test classes are extended from the TestBase class. TestBase handles environment details for the tests to run. The environment details are fetched from Test suite xml (TestNG xml file). TestBase also handles initialization of driver as well as tear down after the test.
- Test class has each individual test written in java which are called from the Test suite xml file. Methods are annotated according to testNG framework. For example, test classes are annotated with @test annotation. @test methods get test data from the TestDataProvider class.
- Page classes contain all necessary tests for the specific page. Test methods in Test class call methods in Page class to execute the test.

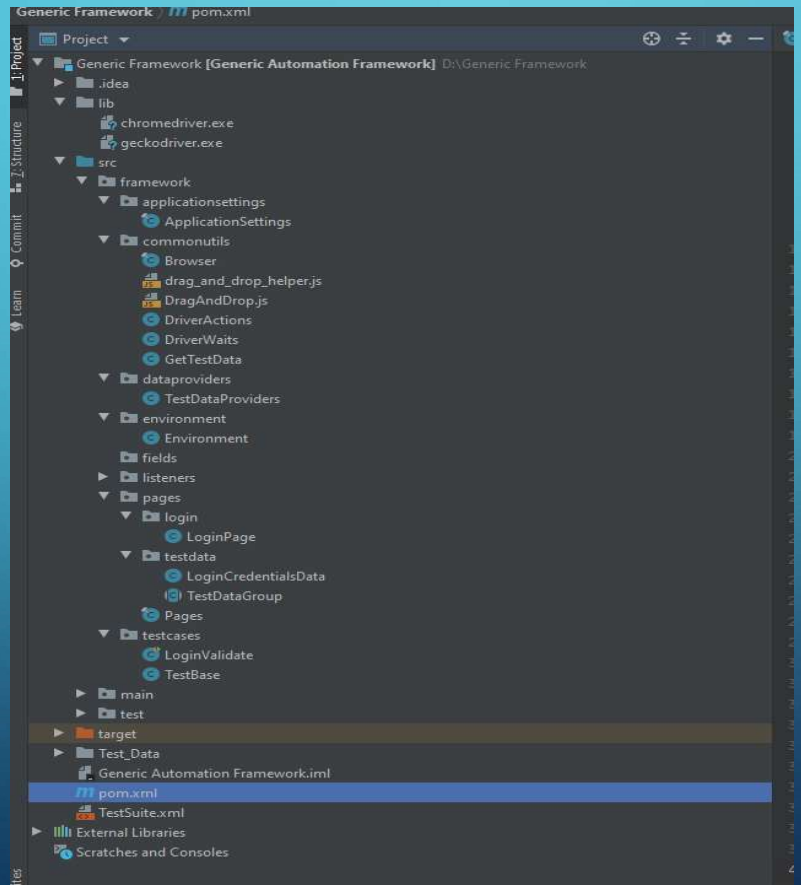
OVERALL ARCHITECTURE(CONTINUED)



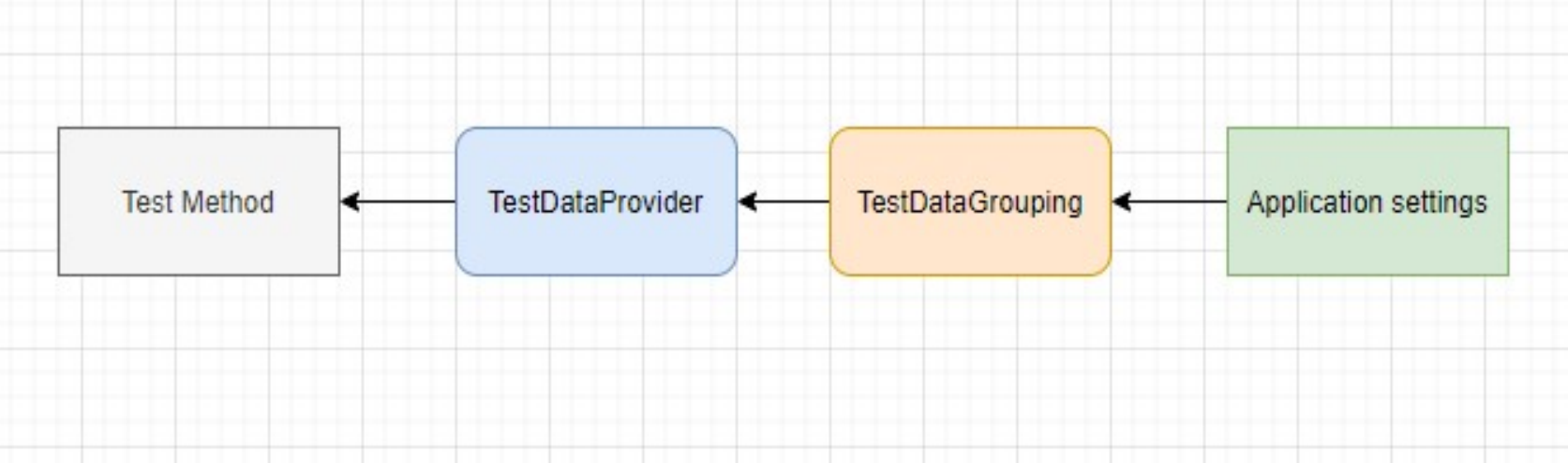
OVERALL ARCHITECTURE(CONTINUED)

- Each Type of classes are packaged separately to maintain ease of maintenance.
- Lib package contains external files, executables and tools as needed
- Commonutils package contain all necessary helper classes
- Listener package holds to listener for testNG
- Testcases package hold the test suite and Pages package contains the page classes

OVERALL ARCHITECTURE(CONTINUED)



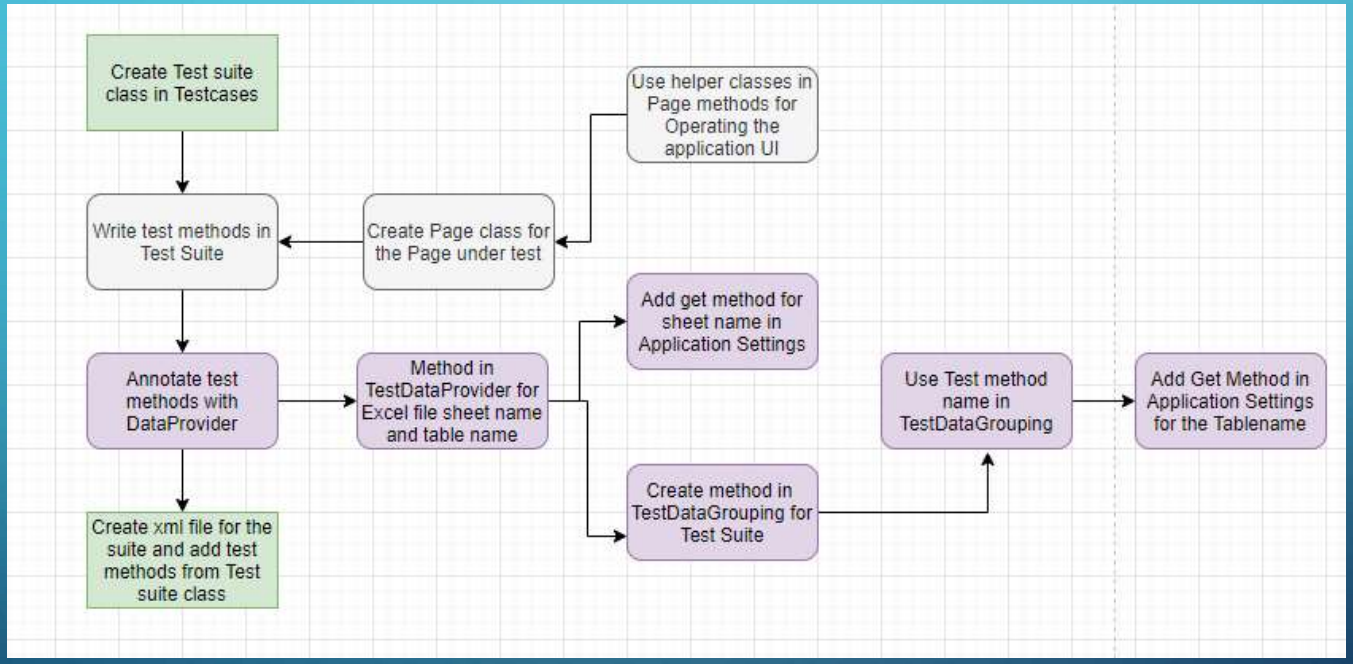
TEST DATA FLOW



TEST DATA FLOW(CONTINUED)

- Test method in Test suite class takes the test data from TestDataProvider.
- TestDataProvider uses the test method name in TestDataGrouping to find the table name from Application Settings class.

GENERAL WORKFLOW



GENERAL WORKFLOW(CONTINUED)

- Create the Test suite class/Skip for existing suites
- Write test methods in Test suite class
- Use Page class for Page under test to Operate the UI
- Setup DataProvider with Necessary Table and sheet name in Application Settings class
- Add the test cases from Test suite class to Xml file

KEY POINTERS

- Raw details such as Test data file location, table name, sheet name, and other environment details are kept in Application Settings class
- Running the test suite from terminal using Maven command will generate A emailable Report file in the folder named Target for report analysis
- TestBase and Browser class initializes the Test suite in runtime