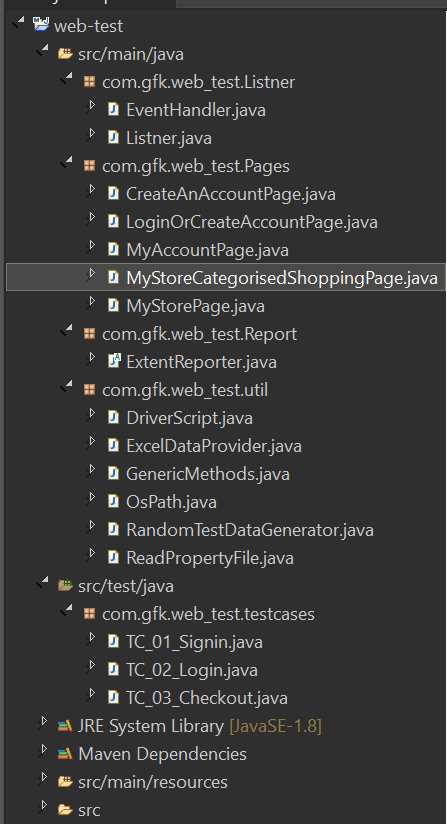
**GFK Coding Assignment 1**

Overview of the Framework

* Key features of this Framework
  + This Framework is designed based on Page Object Model which would help to keep the Page Objects and Test Cases separate for easy maintenance
  + Technology Stack
    - Java
    - Selenium API
    - TestNG
    - Extent Report
    - Log4j
  + Generate random meaningful user test data using JFairy API
  + Test data Driven – Read Test data from Excel using Apache POI and TestNG Data Provider annotation. All test data required is placed in "***TestData***" folder which is available in ***src***
  + Logs are generated from TestNG & WebDriverEvent Listners using log4j and all the logs are stored in TestResults Folder
  + Based on the OS, appropriate driver files are loaded [Window, MAC]. Path for browser is provided by OsPath.java class inside ***com.gfk.web\_test.util*** package.
  + Extent Reports are generated for every execution in ‘TestResults’ folder with the meaning steps details alongside screenshots
  + Browser and URLs are possible to configure in Config.Properties
  + Supports Parallel Execution
  + There are no static waits are used. Only dynamic waits are used
  + Test data driven dynamic Xpath is to handle the following functionalities
    - To click any one of three catagories [Women,Dresses,T-shirts]
    - To click a dress from the shopping list
    - To choose the color of the dress

**Project Structure**



src/main/java

com.gfk.web\_test.Listner

* EventHandler – This class implements all methods in WebDriverEventListner to create logs.
* Listner - This class implements all methods in ITestListener to create logs and reports

com.gfk.web\_test.Pages

* All the PageObjects and its actions are in this class

com.gfk.web\_test.Report

* ExtentReporter – This class is to generate the Extent report with each step description, test cases status and to attach screenshots against each step and close the extentreport.

com.gfk.web\_test.util

* RandomTestDataGenerator – This class will generate random user details using JFairy API and return that data in HashMap
* ReadPropertyFile – This class will read the property file and return that property value
* OsPath – returns the path of the driver files based on the operating system this script is executed
* ExcelDataProvider – returns the test data cell value based on the column name [Make sure the column name is the same what we pass it as parameter in the Test Methods]
* GenericMethods – This class contains ***initialize*** function to instantiate the driver objects based on the browser provided in config.properties. tearDown will close the browser and quit driver. Dynamic waits, getSnapshot, WebElement actions, Initialise EventHandler
* DriverScript - This class contains function to launch browser, also to initialize Eventwebdirver listener and quit browser

src/test/java

com.gfk.web\_test.testcases

* TC\_01 – Signin - To validate the sign-in process of the app
* TC\_02\_Login - To validate existing users are able to login to the app
* TC\_03\_Checkout - To validate the checkout process for an existing user

src/main/resources

* Drivers – This folder contains the browser drivers for MAC and Windows
* config.properties – This property file allow us to choose the browser and modify the URL for different environments like Dev, UAT
* extentreport.xml – To design the HTML report style
* log4j.properties – Contains the properties that required for log4j

Project Root Folder

* TestData – This contains the test data in an excel sheet
* TestResults – Contains the ExtentReport and two subfolders are created namely – Snapshots which stores all the snapshots taken by the webdriver and logs folder which generated by WebDriver listeners
* Pom.xml – contains all the required maven dependencies
* testng.xml – contains the list of test cases and parallel mode configuration

Setup Instructions:-

* Import it as a Maven Project and all the required jar files will be downloaded as per POM.xml
* Store all the driver files in src\main\resources\drivers

Test Execution:-

* Execute the scripts from testNG.xml
* I have executed all 3 test cases in Parallel mode and one report is generated. TC\_03 has 3 sets of test data with different shopping category, dress, color, size and quatity driven from test data.