Certification Project

for

Testing With Selenium Webdriver

1. Introduction:

This document enlists the architecture and functioning of Hybrid automation framework that is established to perform automation testing using Selenium Webdriver. This document also enlists the features of the framework, details about the automated scenarios, and limitations/risks. The framework is a hybrid one, i.e. it combines the features of both, keyword as well as data driven framework. Framework accepts inputs from the excel sheets, and after execution, the results are updated also in an excel sheet. The framework uses Page Factory and Page Object Model, both of the design pattern for ease of coding and maintenance.

1. Framework Architecture

The framework comprises of below packages and classes:

**Package appModules:** This package consits of the application module, i.e. the code releated to the AUT. It contains classes like setup, mercuryRegistration and mercuryFlightBooking. Here is the information about the classes in this package in breif:

1. Class **setup:**  Consists of code related to login and other validations.
2. Class **mercuryRegistration**: Consists of code for new user registration functionalty.
3. Class **mercuryFlightBooking**: Consists of coder for flight booking functionality.

Package commonLibs: As the name suggests, this package consists of all the methods that are common across the framework. This package addresses the problem of code duplication by maintaining all the frequently needed methods at single point. Here is the information about the classes in this package in brief

1. Class **dataProvider**: This class contains methods that acts as data supplier. These methods fetch data from input excel file, and pass it on to the calling method. The method usually can return data as a single value, arrays, and objects.
2. Class **excelDriver**: This class contains all the methods that are needed to interact with excel files. Excel file opening, reading, writing and saving can be performed using the methods of this class.
3. Class **keywordUtility**: This is the class where framework keywords are designed. These keywords are called from the main class of the framework. The methods are specified as test case steps in an excel sheet. This class also initialized WebDriver, that could be passed across different page object classes to perform that specific page related operations.
4. Class **Log**: This class contains methods to write the framework logs. This framework uses log4j for the logging purpose
5. Class **utils**: This class contains various methods which are needed across the timespan of automation testing. Those methods are accumulated in this single class, in order to avoid code duplication and easy maintenance.