**FRAMEWORK ARCHITECTURE**

**Framework** is a collection of predefined classes and functions working together in testing an application without any human intervention. It is an integrated structure of various functions like libraries, test data and reusable modules.

Frameworks can be used to handle the code in systematic way, which can be reviewed by a third person easily.

Components of FrameWork :

Have followed Hybrid driven framework in my project. I could say its Hybrid driven because its combination of

Data driven and also Modular driven

Its Data driven because – whenever we want to execute a particular Test script with huge set of data, we go for external resources

Its also Modular Driven since all the framework components is maintained module wise.

Components of FrameWork :

1)Generic Libraries

2)POM respository

3)Resource

4)TestData

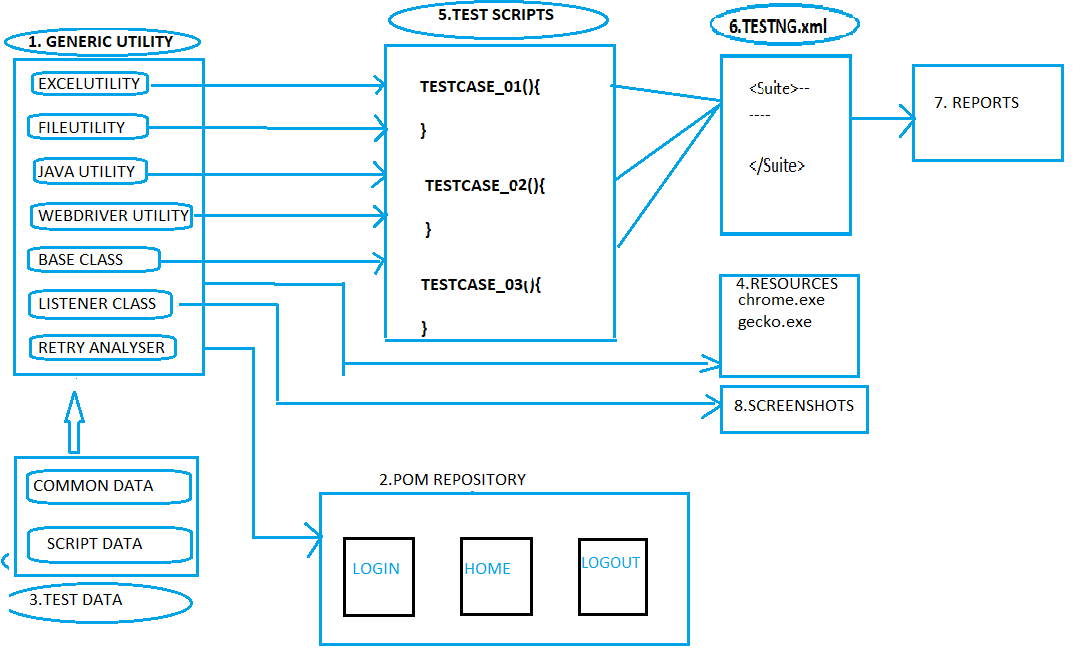
5)TestScript

6)Driver

7) Reports

8) Screenshots

**Framework Structure:**



GENERICUTILITY FILES – This file consists of all the utility files which consists and user defined data and codes which can be reused while scripting. It consists of

\* EXCELUTILITY - Excel utility contains access specific methods implemented by using Apache poi. As per rule of Automation data should not be hard coded in a test scripts. In order to achieve this we make use of Excel Utility.

\* FILEUTILITY – Contains reusable methods to interact with property file which stores common data . Again here since test script cannot be hardcoded we make use of property file to save data such as Browser, Url, Username, Password etc.. The property file stores information in key-value pair format.

\* JAVA UTILITY – Test scripts require data such as Random number , System date , such type of data is stored Java utility class.

\* WEBDRIVER UTILITY : Consists of WedDriver actions such as Implicitly wait, Explicitly wait, handling dropdown , switching to windows , mouse Hover actions , popups. The syntax of such action is little complex and time consuming. In order to overcome this, a frame work developer will develop these methods and save it in Webdriver utility. A test script writer will extend to Webdriver utlity class create objects and reuse all these methods and optimize the code.

\* BASE CLASS – Base class is implemented by using TestNG and it consists of Configuration annotation. Normally in most of the test scripts they are few actions that are common like, Connecting to database, launching the browser, Opening the application , login , logout , closing the browser and disconnecting from Database . Such preconditions are developed by using test annotations and are stored in Base class. All these preconditions are inherited into test script class by extending to Base class.

\*LISTENER CLASS– Listener implementation class is implemented by using TestNG. During test execution in runtime if the test scripts gets failed, Listener implementation class helps us to take a screen shot and also store in Screen shot folder. The Screen shot is used for debugging actions.

\*RETRY ANALYSER – When any test script fails in a suite due to performance issue or browser issue or server issues(say unexpected delay in response time). These failures needs to be investigated and TestNg provides Retry analyser feature which helps us to automatically re-run a test script .

2) OBJECT REPOSITORY

A repository is a centralized place to manage and store data.

An object repository is a common storage location for all the application objects and their properties. Here objects typically refer to locators used to uniquely identify the webelements and store in centralized place to avoid hardcoding within script.The major advantage of Object repository is segregation of objects from test cases. It is implemented using POM design pattern.

In Agile methodology since requirements/UI keep changing / updating and in such cases maintaining the element in xpath will be a tedious work.

In such case if we create a separate class for each page and store only elements pertaining to that particular page, it would be easier to identify the elements and modify them as required.

3) Test Data – In Test data we maintain all the data required in the Framework. There are two type of data

a) Common data ( consists of – Browser , URL , Username , password)

In case of we want to run the entire execution on a different browser it can be changed in the common data. If we want to execute the entire execution by changing the credentials we can change it in the common data.

b) Script data – In case we want to execute the complete execution by changing the complete set of data, we can change it in excel.

So maintenance and modification of test data is faster by using Testdata folder

4) Resource - Resources consists of driver specific executable files which is used to launch particular browser. It is used to achieve Cross Browser testing.

5) Test Scripts – It is collection of TestNG test scripts. We create specific classes for every test case, a by using TestNG annotations and all other components of the Framework Test scripts are developed.

6) Driver – It is made of TestNg xml files. By using driver component we achieve Group , Batch and Parallel execution.

7) Reports – The reports of Test script execution shows the stability of the application. We will be able to decide whether the software can be released based on the report.

8) Screenshot - The screen shot of failed Test cases during runtime will be save in this folder. It is used in debugging the application.