Test Automation Framework

1.1 Brief about the Framework

This framework consists of 8 modules :
1- Test Suits classes : 4 Test Classes which have the test cases scripts.
2- Pages Objects classes : Each page has its own class containing Functions that can be used in it to be used in achieving the scenarios in the test cases.
3- Generic Actions: which will be used by Page objects to apply actions on the website.
4- Data Driven Files :
Test Data -> will be used in each test case
Elements Locators -> which identify the website element to be used in page objects functions by requesting some Actions on them from the Generic Actions module.

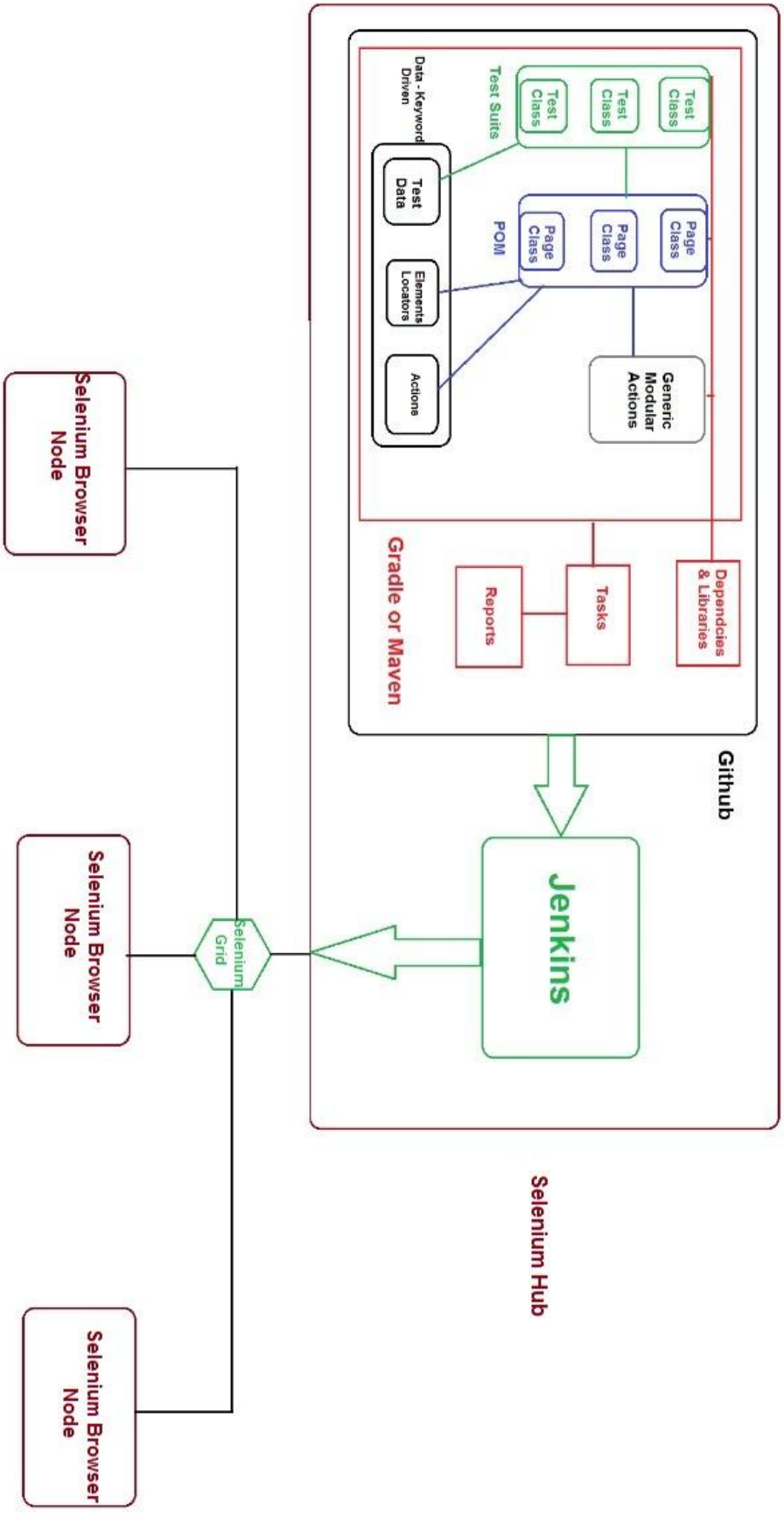
5- Gradle : will be used to organize the dependencies and tasks.
6- Dependencies files : Libraries which will be used in this project such :
Selenium WebDriver -> For web test automation.
Selenium Grid -> For parallel execution.
TestNg -> Unit testing and assertions.
Apache POI -> Excel handling .

7- Jenkins : Continuous Integration

8- Docker : Used to Set up multiple containers.

Check it out on Github

1.2 Framework Architecture



1.3 How to run the Project

Pre Conditions :

- Fill the required test data in the test data properties file named "test.properties" like the facebook email and password .. etc
- The project can run tests in parallel however in local browser instances or remote devices using selenium grid and docker , if you want to run it locally set the attribute "useSeleniumGrid" in the file "test.properties" to "false". For selenium grid change the attribute value to be "true".

The project can be run with different environments :

- Jenkins [Local running] :

1- Create a freestyle project.
2- In General Section : choose github project and paste the link of the project Repository.
3- Choose git as Source Control and paste the same link with ".git" at the end.
4- In the build Sections choose : Invoke Gradle Scripts.
5- In the Tasks TextField type "clean test"

- Jenkins [Selenium Grid + Docker] :

1- Open CMD or PowerShell and run these commands :

```
* Once :
$ docker network create grid

* Once :
$ docker run -d -p 4444:4444 --net grid --name selenium-hub
selenium/hub:3.141.59-bismuth

* 4 times to create 4 chrome containers
$ docker run -d -P --net grid -e HUB_HOST=selenium-hub -v
/dev/shm:/dev/shm selenium/node-chrome-debug:3.141.59-bismuth
```

2- Do the same as the previous Jenkins setup . P.S don't forget to change the flag of use selenium grid in the test.properties

- CMD / Terminal :

1- Open CMD and navigate to the project directory and type "gradlew clean test" for windows or "gradle clean test" for others.
2- For Docker : run the previous 3 docker commands before your run the clean test.

- Eclipse IDE or NetBeans or any IDE :

1- Right Click on the file "test.xml" Run as TestNG Suite
2- For Docker : run the previous 3 docker commands before running the tests.

1.4 Test Reports

After Running with Gradle:
- Open the page "index.com" in the folder "Build/reports/test…" to get a full tests results.

After Running with TestNG:
- Open the page "index.com" in the folder "test-output" to get a full tests results.

1.5 Recommended Enhancements

Due to time constrains some features couldn't be added to this milestone such as :
- Using Excel sheet for Data Driven and test case results using Apache POI .
- Using Allure or Jira For reporting bugs.
- Using Test Café instead of Selenium