**Overview**

The task solution is implemented with a hybrid TDD framework design. The solution is currently designed for Android automation but also contains few packages and little code for IOS which can be used to achieve modularity and platform coverage

Below are the components of the framework –

·        Automation tool – Appium + Selenium

·        Framework + scripting language – Core Java

·        Base framework – TestNG

·        Build Automation tool – Maven

·        Reporting -  Extent Reports + Customized TestNG Report

·        Design pattern – Page Factory, Page object modelling

**Prerequisites –**

Below are the prerequisites that need to be present on the automation machine for the flawless execution.

·        Install Java Jdk 1.8

·        Install apache maven

·        Install Android studio (optional)

·        Add environment variables for JAVA\_HOME, M2\_HOME, ANDROID\_HOME(optional)

·        Ready android emulator or real android device connected to machine with Developer option -> USB Debugging enabled

·        Install Appium via npm command or install Appium server with GUI support

·        Eclipse IDE (optional – Needed only to update scripts or develop new scripts)

**Steps to Execute –**

·        Download the source from github repository to the automation machine <https://github.com/alhadpingle06/CNTaskSolution.git>

·        Import the project as existing maven project in eclipse

·        Go to command prompt and navigate to the project folder

·        Perform mvn clean and then mvn install

·        Launch the Appium server from the command prompt with the command ‘appium --allow-insecure chromedriver\_autodownload’. This is to enable Appium download appropriate browser driver on the fly to support webview test cases

·        Users can choose which test cases to execute from TestExecutor.xlsx which is present at src/test/resources/TestConfig folder. To enable the test case for execution, set the flag to Yes otherwise No. Make sure that you save and close the TestExecutor.xlsx

·        Execute the project with mvn test from command prompt or from eclipse with specifying test as a goal

**Reports –**

There are 2 reports that are being generated.

·        Customized TestNG Report – The customized TestNG report is visible after the execution at ‘target\reports\{timestamp folder}\custom-report.html’. This report contains the detailed execution report with screenshot on failure and other details

·        Extent Report – This is more fancy report with all the same details as present in Customized TestNG Report and is visible after execution at Reports\Reports\_Android.html

* One sample report each is uploaded for quick reference in the source. Sample reports’ name starts with ‘Sample\_’

**Important Points To Note –**

* There are 2 optional command line arguments / parameters that can be used during execution. They are ‘Platform’ and ‘suiteName’

·        Platform can have two values – Android and IOS. However for this solution we are supporting only Android. Code snippet for IOS is present in the source but is not supported because of security reasons

·        suiteName is currently regressionSuite. We can add more suites and can configure the execution. This can be discussed further during the discussion round

·        As of now these parameters are optional and are working with default values as explained above

* The current solution is implemented with few assumptions and can be discussed and updated further to provide the optimized solution

·        Any problems in the set up or any queries can be addressed over the discussion on this solution