<<<FRAMEWORK CORE STARTS HERE>>>

1. Get the Application (Facebook or Gmail) and its version (V1.0 or V2.0) from the batch file.
   1. Read the “config.properties” file to get the appropriate
      1. Create an Entry in the Log file if the property file is loaded and able to read the application and versions then continue
      2. Create an Entry in the log file if Runner could not locate the Property file or Runner could not locate the appropriate Applications and its version then abort the Runner.
2. Load the appropriate application Environment file “testenv.properties”
   1. Read the content from the above property file and store all the values into global variable (Each Variables should be used till end process.
      1. Create an Entry in Log file if Property file is successfully loaded. Then Continue
      2. Create an Entry in Log file if Property file is not successfully loaded. Then abort the Runner.
3. Get the Test Suite Location
   1. Make sure that the system able to get this location and at least finds one Test Suite file (Excel File)
      1. Create an Entry in Log file If the Location is accessible
      2. Create an Entry in the Log file If the Location is not accessible Or If there is no Test Suite Excels Then abort the Runner.
4. Load Object Repository file.
   1. Load the Object Repository file “object-repo.xls”
      1. Create an Entry in the login file If Object Repository file is present and continue
      2. Create an Entry in the log file If Object Repository file is not present and ABORT the Runner.
5. Read the “object.repository” sheet of the Object Repository Excel file.
   1. Read the “object.repository” sheet and get the value from the columns (Object ID, Object Name, Locating By and Element) and iterate all the rows using “Object ID” and store all the objects in the global property file.
      1. Create an Entry in the Log file If file format is not supported or the Object Repository sheet does not have any objects, Then ABORT the Runner.
6. Get the First Test Suite (Excel file)
   1. Get the total Test Suite Count (# of the Excel file) and Iterate One by one
      1. System should know which test suite currently get executed
      2. Create an Entry in the Log file that Total # of Test Suite Identified
   2. Get the Excel type (.xls or .xlsx) from the “config.properties” ?? or “testenv.properties?? – TBD.
7. Read the Excel File (Test Suite)
   1. Read the first Test Suite Excel file using appropriate Excel Read method
      1. Create an Entry in the log file if system is able to read the first Test Suite Excel File and Continue
      2. Create an Entry in the log file If the File format is not supported (ex., .xls instead of .xlsx or ViceVersa) and Move on to Next Test Suite.
8. Read the First Sheet of the Test Suite file (Details)
   1. Get the (Test Suite ID, Test Suite Name, Location, Application Server Configuration and Database Server Configuration) and store this Value as Current Run Test Suite Parameters.
      1. Create an Entry in Log file If system able to load the Details sheet. Then continue
      2. Create an Entry in the log file and If could not locate the “Details” sheet and put the Current Run Test Suite Parameters as “Unknown” and continue.
9. Validate the Current Test Suite has appropriate test result sheet.
   1. To validate this file get on to the test results location.
      1. Create an Entry in the log file if system could not locate the specified location and terminate the TEST SUITE RUNNER
      2. Create an Entry in the log file if system is able to access the location and continue
   2. Validate the Current Test Suite result Excel file, the file name must be <TS ID>\_<Test\_Results>
      1. Create an Entry in the log file if system is able to locate the specified file and CONTINUE
      2. Create an Entry in the log file if system could not locate the specified file and ABORT the CURRENT TEST SUITE RUNNER and move on to Next Test Suite.
10. Read the Second Sheet of the Test Suite (Test Cases)
    1. Get the first test case details based on the “Execution Order” Set the first Test Case as Current Test Case Run and assign all the parameter (**CurrentRUNTestCaseParameter**) as Variables (Test Case/File Name, Execution Order, Test Case Status, OS Configuration, Browser Configuration, Host Machine and Host Machine Credentials). Iterate the Test Case based on the “Execution Order”
       1. Create an Entry in the Log file If Test Cases found in the sheet, for each Test Case with Test Case file name.
       2. Create an Entry in the Log file if system could not find any Test Cases to be Run. And move to Next Test Suite
       3. Pick the test Cases only which are in “Ready” status in the “Test Case Status” column, Ignore if Test Cases Status is either Blocked or Completed. (But the Completed or Blocked Test Cases should be listed in the Test Suite Results with appropriate status.
    2. Create a Method to store Test Suite Results.
       1. Create **Test Suite Run ID**, {Example: <TestSuiteID>\_<Current Data and Time>}
       2. This Test Suite Run ID is same for all the test cases within the Test Suite for the current Run
11. Get the Test Case Sheet from the Location as specified in the Test Suite
    1. Get the appropriate test case sheet from the location which is mentioned Test Suite Excel.
       1. Create an Entry in the Log file if system able to locate the Test Case Excel in the Specified location
       2. Create an Entry in the Log file if system could not find the Test Case sheet Then Continue
12. Read the Excel File (Test Case)
    1. Read the first Test Suite Excel file using appropriate Excel Read method
       1. Create an Entry in the log file if system id able to read the first Test Case Excel File and Continue
       2. Create an Entry in the log file If the File format is not supported (ex., .xls instead of .xlsx or ViceVersa) and Move on to Next Test Case Sheet (from Step 4)
13. Read the “Details” sheet of the Test Case Excel
    1. Read the “Details” sheet of the Test Case Excel and Get the Value from (“Test Case ID” and “Test Case Name” )and assign in to appropriate variable {Example: Test Case ID = currentrun.testcase.id and Test Case Name =currentrun.testcase.name}
       1. This above variables should store the appropriate Test Case ID and Test Case Name of the Test Case which being executed
       2. If Test Case ID and Test Case Name are not present in the sheet then put Test Case ID and Test Case Name as “Unknown”.
14. Read the “Manual” sheet of the Test Case Excel
    1. Read the “Manual” sheet of the Test Case Excel and Get the value from (Step ID, Design Step, Design Description and Expected Result) and assign the value in to appropriate variable {Example : Step ID =current.step.id, Design Step Description=current.designstep, and Expected Result=current.expected}
       1. Create an Entry in the log file If system could not find the sheet “Manual” or Could not find the above value, then assign “Unknown ” to the all current Step variables except the “current.step.id”, set the “current.stepid=1” (iterate each step +1)
15. Read the “Automation” sheet of the Test Case Excel.
    1. Read the “Automation” sheet of the Test Case Excel and Get the value from (Step ID) using current.step.id.
       1. Collet the automation parameters from appropriate columns (Object, Control, Action, Input Value, Expected Object and Expected Action) for the current.step.id {Example : (Object=current.object, Control=current.control, Action=current.action, Input Value=current.inputvalue, Expected Object=current.expectedobject and Expected Action=current.expectedaction}
       2. Create an Entry in the log file If system could not find the sheet “Automation” or Could not find any steps then ABORT the CURRENT TEST CASE RUN and move on to next Test Case.

<<<<<AUTOMATION CORE STARS HERE>>>

1. Create Test Case RUN ID as <Test Case ID>\_<Current Data and Time> and Start Date and Time.
   1. Store this value in to the variable, {Example : current.test.run.id and current.test.start.time}
2. Create a Method as “Generate\_Result”
   1. This Method should store the “current.testcase.resultcode”, “current.testcase.resultmsg”, “current.step.resultcode” and “current.step.resultmsg”
3. <Create a Method to Get the Object Property by Passing “current.objectname”>
   1. If “current.objectname” <> NA, then this method should get the “Element” value of the “current.objectname” from Step 5 Property file and store the value into “object\_id”
   2. This method returns “**object\_id**”
      1. If the Object is not present in the Object Repository global variable then invoke the method “Generate\_Result” and set the “current.step.resultcode = “Fail” and “current.step.result.message” = “ Object” + Object Name +”does not exist in the repository” then move on to next step.
4. Create Java file as “common.function.library “
   1. Create a Method that handle this values as Parameter (object\_id, current.contol,current.action,current.inputvalue)
   2. This method should return 3 values as (current.step.resultcode, current.step.resultmsg and current.step.continue<Boolean>)
5. Check the Conditions
   1. If current.step.continue = TRUE and current.step.expectedobject = NULL or NA then move on to next step.
   2. If current.step.continue= TRUE and current.step.expectedobject <> NULL or NA then
      1. Get the appropriate object from the object repository and assign the Element in to the object\_id.
      2. Get the “current.step.expectedaction” and pass the “object\_id” and “current.step.expectedaction” to “PostValidate” method in the commonLibrary class file
      3. The “PostValidate” method also should return 3 values as (current.step.resultcode, current.step.resultmsg and current.step.continue<Boolean>) Note : This time the “current.step.resultmsg” should concatenate the results.
   3. IF The Current Step is FAIL (or) WARNING then mark the current.testcase.runstatus with appropriate status as below
      1. IF current.step.runstatus = WARNING then MARK the curret.testcase.runstatus = WARNING only IF current.test.runstatus <> FAIL
      2. IF current.step.runstatus = FAIL then MARK the curret.testcase.runstatus = FAIL with NO CONTITION
      3. IF current.step.runstatus = PASS Then no action required
6. Get the appropriate Test Suite Result Excel file and do the following necessary steps (This sheet existence already validated during the Test Suite sheet Read steps)
   1. Get the appropriate Result Excel file and create a new sheet with the current.Testsuite.runid
      1. Create an Entry in the log file If the File format is not supported (ex., .xls instead of .xlsx or ViceVersa) and Move on to Next Test Suite.
      2. Create an Entry in the log file if system could not create a sheet with Test Suite Run ID and ABORT the CURRENT TEST SUITE RUNNER and move on to Next Test Suite
      3. Create an entry if system is able to create new sheet in the appropriate Test Suite result sheet
   2. Create a columns as in the first row from first column in the newly create sheet (Test Case ID, Test Case Name, Step ID, Step Name, Step Description, Expected Result, Actual Result, Stack Trace and Step Run Status)
      1. Create an Entry in the log file if system could not insert the specified column names and ABORT the CURRENT TEST SUITE RUNNER and move on to Next Test Suite
      2. Create an entry if system is able to create the appropriate column names in the excel sheet
7. Write the Test Steps Results in the Current Test Suite RUN ID sheet
   1. Write the Current step’s details in the second row of the created sheet (Test Case ID, Test Case Name, Step ID, Step Name, Step Description, Expected Result, Actual Result, Stack Trace and Step Run Status)
      1. Create an Entry in the log file if system could not insert the current steps details in the first row and ABORT the CURRENT TEST SUITE RUNNER and move on to Next Test Suite
8. CHECK the Step Continue Condition
   1. If current.step.continue = TRUE then continue to next Steps
      1. Clear all the Current Step Parameters to initial stage so that we can store the next Steps’ details in to the variables
   2. If current.step.continue=FALSE then stop the CURRENT STEP RUNNER and move on to next Test Case
      1. Follow the procedure for writing Current Test Case Results in the Results Excel sheet, steps explained in blow in this document.
9. Continue the Process until the STEPS ITERATTION IS COMPLETED.
   1. If all the Steps are completed in the Current TEST CASE then write the Current Test Case Results.
   2. Capture the Current Test Cases END date time.
10. Read the “Test Suite Result” sheet of the Current Test Suite Result Excel. (The Excel file Read validated already done during the step result writing process.
    1. Read the “Test Suite Result” sheet of the Current Test Suite Result Excel.
       1. Create an Entry in the log file if system could not read the specified sheet and ABORT the CURRENT TEST SUITE RUNNER
       2. Create an Entry in the log file if system is able read the specified sheet.
11. Write the current Test Case RUN details in the specified Excel sheet.
    1. Write the TEST CASE CURRENT RUN DETAILS in the sheet in next available Row with appropriate columns (Test Suite Run ID, Test Case Run ID, Test Case ID, Test Case Name, Run Status, Start Time, End Time, Duration, Host Machine, OS Configuration and Browser Configuration)
       1. Create an Entry in the log file if system could not find the specified column name in the sheet and ABORT THE CURRENT TEST SUITE RUNNER
    2. Clear the all the CURRENT TEST CASE DETAILS so that system could store the next run test case details
12. Continue this Process until all the test cases are completed in the Test Suite
13. Continue this process until all the Test Suites are completed in the location
14. Create an Entry in the log file that “The Test is Completed”
15. Send the Email to specified Email address <<Content of Email still needs to be decided.

<<<END>>>