# Limelight Android SDK Automation overall code flow information

## Source code directory details:

config – contains all the configuration files

config/system.cfg – contains configuration details of Appium, Device under test and App.

config/element-path.cfg – contains details like xpath, scroll-h(), scroll-v and tap-position for all the GUI elements accessed while running feature files.

consoleLog – contains all the output logs generated by the automation.

featureFiles – contains all the feature files.

lib – contains all 3rd party libraries.

lib/appium – appium module.

lib/behave – behave module.

lib/behave4cmd0 – support module for behave.

lib/bin – support module for behave.

lib/dateutil – support module for behave.

lib/enum – support module for behave.

lib/freetype – support module for behave.

lib/json2html – support module for appium.

lib/parse\_type – support module for appium.

lib/requests – support module for behave.

lib/selenium – support module for appium.

lib/parse.py – support module for appium.

lib/pexpect.py – support module for appium.

lib/pyparsing.py – support module for appium.

lib/six.py – support module for appium.

modules – modules written by Rebaca.

modules/access\_limelight\_cms.py – This module is written for accessing limelight CMS data.

modules/appium\_driver.py – This module is use to pass on the actions to be performed the

limelight app via appium client. The client then talks to to the

Appium server, which then talks to the android app.

modules/constant.py – Contains all the constants.

modules/exception\_mod.py – Does all the exception handling.

modules/limelight\_mod.py – Contains the methods to be called by the steps or glue code

from steps/ limelight\_glue\_code.py.

modules/logger.py – Responsible for generating console.logs.

modules/lvp\_auth\_util.py – URL authenticate module from Limelight.

modules/OpenCVLib.py – Module used image comparisons and to check image inside image.

modules/ordereddict.py – Helper module to create ordered dictionaries for comparison.

modules/report\_generator.py – Module responsible for generating html reports after run.

output – contains all the test reports

screenshots – contains all the screenshots for failure steps

screenshots/player\_btn – contains images for comparison.

steps – contains the glue code for the feature file

steps/ limelight\_glue\_code.py – This is glue code for the feature file where the feature steps

are parsed and actual methods are called.

behave.ini – configuration file for behave to switch on/off log and error capturing

environment.py – this is a module for behave, where the pre and post conditions for feature, scenarios and steps were written.

fullRun.sh – shell script for full run on Mac

main.py – automation initiator or main driver for automation

runOnWindows.bat – batch script for full run on Windows.

## Code flow:

Code flow diagram

1. The automation can be initiated via .bat file on windows, via .sh file on Mac or via console where we can run individual feature or set of features. All the methods will invoke the main.py file.
2. The main.py file will then calls behave to run the feature files.
3. Behave will read the feature file and start calling the glue code, which is written in step/limelight\_glue\_code file.
4. The limelight\_glue\_code file uses the following helper modules: modules/constant and modules/logger. Then the glue code module parses a particular feature steps and then calls the sequence of methods from the modules/limelight\_mod.
5. The limelight\_mod file uses the following helper modules: modules/constant and modules/logger. It also uses the modules/access\_limelight\_mod when data is required from limelight CMS. And also uses the modules/exeception\_mod to raise execeptions. It then calls the methods of modules/appium driver when it requires, getting data or performing action on the android app.
6. The modules/logger is responsible for writing to the log file in the consoleLog directory.
7. The modules/constant contains all the defined constants required by the automation.
8. The modules/access\_limelight\_mod uses the modules/lvp\_auth\_util for authorizing the REST url.
9. The modules/appium driver communicates with the lib/appium application, which again talks to the Appium server application and it talks to the attached device.

It also uses modules/openCVlib for validating screens. In cases of failure this module captures the screenshot of the app and stores it in screenshot directory.

It uses the following helper modules: modules/constant and modules/logger.

1. Once behave completes running all the feature file, the main.py initiates the modules/report\_generator for generating the reports.
2. The modules/report\_generator generates the report and put them in output directory.