**Digit 88 : Appium UI Automation Assignment**

1- Attached is the APK , Using Android studio/Eclipse/Atom or any other IDE . Analyse the APK for any performance issues.

2- Write a cross platform (android + iOS) automation framework of same APK. For iOS just add the code in framework need not to be running on simulator.

Assert that android application has launched on emulator or physical device and any of the element is displayed.

3-Commit framework to git or google drive and share the link with us.

**Submission:**

1. Analysis of .apk

Random mobile app does below:

1. Generates a random number. Inputs can be given as integer values only.
2. The generated random number must be between the min and max value given
3. If the min and max number is empty/0, it generated -1 as random number with an error “empty fields”/increase minimum or decrease maximum
4. The color keeps changing of an app elements
5. The app also can add persons and delete persons
6. When clicked on ‘Select Random’ without having any person added, an error message appears “Please, add persons”
7. When persons added, and clicked on ‘Select Random”, randomly a person is selected and displays on the screen
8. The app can generate a password of the length of give integer value as input. The password is generated as string having combination of a-z, A-Z & numbers
9. The TestNG framework has been created and the project is a maven project. The code does below:
10. The framework is integrated with all the required libraries – JRE System Libraries and Maven System Libraries 🡪Selenium-java libraries, java-client libraries, testng libraries, extent reports libraries, reporting libraries and many other important libraries which help the project to be built as per our requirement and makes it more strong. The libraries are given as dependencies in pom.xml which also has important plugins integrated, for example maven surefire plugin etc.
11. A runner testng.xml file has been generated which can run the test suites. Currently it is having one test suite which contains one test case.
12. Right click on testng.xml and run as TestNG Test, it will execute the test case and generate an Extent Report in Project folder named as ‘Reports’ with screenshots. Please click on the html file to see the test report status.
13. Extent Reports have been implemented for generating the test reports. ReportsConfig.xml is added.
14. ITestListeners has been implemented whose overridden methods give the result of tests. onTestSuccess and onTestFailure methods call the extent reports methods to generate reports.
15. JSONDataProvider utility class results in json object which then helps in providing the test data to the testcases.
16. Explicit Waits have been implemented in WaitConfig utility class.
17. TestBase.java file is responsible for instantiating AndroidDriver/iOSDriver
18. In configuraration.properties file, if your device already has the given .apk installed keep the *ANDROID\_APK=NoInstall* uncomment and *ANDROID\_APK=Random\_v8.0\_apkpure.com.apk* commented. If you want to install it through code, delete the .apk from your device and uncomment the second parameter and comment first one.
19. com.random.pages package contains the java files which implements the methods behind the test case execution
20. com.random.tests pakages contains the testcases
21. The framework and code for Android and iOS has been submitted in GoogleDrive. Please find the link below:

Link:

<https://github.com/laxmisinghal/Digit88_RandomMobileAppTest>