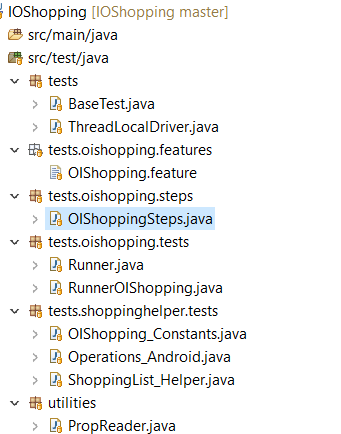
**Sivaranjani S - Technology Analyst**

**Infosys Limited**

**RBS – Mobile Automation Exercise [OI Shopping List]**

**Instruction Document**:



* **OIShopping.features** – Has two Scenarios covered
  + Creating list , adding items and deletion of an item from a list
  + Creating list , adding items and Sorting the values
* **OIShopping.Steps** – Has Steps defined corresponding to the Feature File.
* **OIShopping\_Constants** – All the page element attribute values for the above scenarios are declared here to generate the Xpath or to find elements with other locator strategy.
* By keeping the constants in separate Class would help when there is change in value. We need to update value only here, instead of disturbing the helper classes.
* **Operations\_Android** – This Class has all the android operations used in the project   
  (Find Element, Find Elements, Get Attribute, Get Attributes, Click, Send Keys,..).
* In project framework, creating this way helps to use the functions universal across project by keeping it in **Common Library folder**. We can create our own Permutation and combination of methods with the available Appium mobile commands for better Performance.
* **ShoppingList\_Helper** – This Class has all the implementation of the flows .The methods are partitioned in such a way the function could be used universal from any module, wherever the function needs to executed .Methods implemented here are: **Add List, Add Items, Delete List and Sort List.**
* **Some Common functions inside Helper Class** :
  + getItemsFromArrayList
  + StringListMatch
  + listWithStringComparison

Functions like these could be written common and kept in a common folder, so that it could be used universal in the project and can be used wherever required.

* **Prop Reader** – Has all the desired Capability Configurations .It’s also recommended to keep and pass the value from a file instead of Hardcoding, Changes could me made easily only here, not bothering about code validations.

**Further Improvements:**

* **TestNG**: Annotations, Parallel Execution – Can be used for Better Customized reporting we can use Listeners (iTest Listeners / iSuite Listener). Its implemented methods can be overridden in our own customized way upon or before the script execution.
* Creating Interfaces for Mobile Specific (functions related to drivers) in Library folder and we can implement and define them based on the tools used in the project.
* Following **Design Patterns** in Frame work would help to achieve high Code readability and maintenance.
* **Locators** **Strategy**- XPath should be generated on the GO, instead of hardcoding the values. By creating methods to generate Xpath by passing only the required paramets (Elemettype, Attribute, and Relation) – This helps a lot when we have use Xpath axes with multiple attributes.
* **Logger** – Can be used to print a Test Step, Actual Result and Expected Result. Instead of Syso statements.
* Download the project from GIT [https*://github.com/RanjaniRsk/IOShopping*](file:///C:\Users\HP\Documents\IOShopping)*.*Save the project and use RunnerOIShopping Class to execute the cases. Run-As – Junit.