The given high level framework is developed using the OOPs concepts of Java in a Page Object Model.

This is a Maven project where all dependencies are added in the **pom.xml**

The testNG framework is being used with **testng.xml** as the test runner

The properties like platform name, version, app name, package name etc can be set and modified according to need in the **testproperties.properties** file available under **src/test/resources**.

Test Data can be provided either in the form of a data provider class or excel file. Here, data provider class has been used for login credentials. This is available under **com.w2a.appium.testData**

The TestBase class serves as a base for all the test case classes where Appium server is started and drivers are initialized

Each Page has a related class associated for its locators and methods. The two screens available now are **Login.java** and **ProductScreen.java** under **com.w2a.appium.testcases.**  Elements are located by FindElement of PageFactory (using Android FindBy) and both these ways are shown in these files

Test cases using testNG are available under **com.w2a.appium.testcases**.

Extent reports are integrated for the report generation and the output can be found in **test-output/extentreport.html**

**Note**:

This automation file mainly focuses on automating some of the use cases in ‘Validate PDP’ flow using Android Appium Driver. The framework is robust and can be used with iOS Appium app automation as well by changing/adding the drivers for iOS , desired capabilities and iOS Locators (iOS FindBy)

Please note that as I do not have a Mac machine in hand, iOS scenarios have not been implemented