**POM**

**POM Design**

Locaters and Test Script stored Separately

Test Script

POM CLASS

**Dependencies:**

<dependency>

<groupId>io.appium</groupId>

<artifactId>java-client</artifactId>

<version>7.0.0</version>

</dependency>

**Desired Capabilities setup**

package android;

import java.net.URL;

import java.util.List;

import java.util.function.Function;

import java.net.MalformedURLException;

import io.appium.java\_client.MobileBy;

import io.appium.java\_client.android.AndroidDriver;

import io.appium.java\_client.android.AndroidElement;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

public class LoginAndroid {

public static void main(String[] args) throws MalformedURLException, InterruptedException {

DesiredCapabilities caps = new DesiredCapabilities();

     caps.setCapability(“test.user”, “your\_username”);

          caps.setCapability(“test.key”, “your\_password”);

     caps.setCapability(“app”, “hrpp://app\_url “);

        caps.setCapability(“device”, “OnePlus 10R”);

        caps.setCapability(“os\_version”, “9.0”);

     caps.setCapability(“project”, ” AppiumProject”);

            caps.setCapability(“build”, “build-1”);

            caps.setCapability(“name”, “test\_build”);

        assert(allProductsName.size() > 0);

              driver.quit();

}

**Driver Class**

This class instantiates the driver objects.

import io.appium.java\_client.android.AndroidDriver;

public class Driver extends DriverclassExample{

protected AndroidDriver driver;

public Driver() {

this.driver = super.getDriver();

}

}

**Page Class for Login Page**

**import org.openqa.selenium.WebElement;**

**import org.openqa.selenium.support.CacheLookup;**

**import org.openqa.selenium.support.FindBy;**

**import org.openqa.selenium.support.PageFactory;**

**import org.testng.Assert;**

**public class LoginPage extends Driver{**

**PageObjects loginPage;**

**String userName = “”;**

**String passWord = “”;**

**public LoginPage() {**

**super();**

**loginPage = new PageObjects();**

**PageFactory.initElements(driver, loginPage);**

**}**

**public boolean validateLoginpage(){**

**boolean elements = false;**

**if(loginPage.userNameFld.isDisplayed()){**

**if(loginPage.passwordField.isDisplayed()){**

**if(loginPage.checkBox.isDisplayed()){**

**if(loginPage.loginBtn.isDisplayed()){**

**elements = true;**

**}**

**}**

**}**

**}**

**else{**

**elements = false;**

**}**

**return elements;**

**}**

**public boolean testLoginWithoutCredentials() {**

**boolean loginStatus = false;**

**loginPage.loginBtn.click();**

**if (loginPage.inputError.getText().equalsIgnoreCase(“Username is mandatory”)) {**

**loginStatus = true;**

**}**

**loginPage.userNameFld.sendKeys(userName);**

**loginPage.loginBtn.click();**

**if (loginPage.inputError.getText().equalsIgnoreCase(“Password is mandatory”)) {**

**loginStatus = true;**

**}**

**return loginStatus;**

**}**

**class PageObjects {**

**@CacheLookup**

**@FindBy(id = “login\_username”)**

**public WebElement userNameFld;**

**@CacheLookup**

**@FindBy(id = “login\_password”)**

**public WebElement passwordField;**

**@CacheLookup**

**@FindBy(id = “btnSignin”)**

**public WebElement loginBtn;**

**@CacheLookup**

**@FindBy(name = “Invalid ID or password.”**

**public WebElement inputError;**

**@CacheLookup**

**@FindBy(id = “checkBox”)**

**public WebElement checkBox;**

**}**

**}**