# Advanced Software Quality Assurance
# IS4102
# Assignment - 02

**Prepared By**
**R.M.D.K.B. Rajakaruna**
**2018/IS/064**
**18020641**

## 1. Draft Test Cases

Test cases are drafted for the test the functionalities related to the **HR manager(admin)** to test the login, view employee profiles, review leaves, review attendance reports, and approve or reject timesheets.

### Test Scenario 1 - Login Functionality(Admin)

| TC Id | Test Objective | Steps | Expected Output |
|---|---|---|---|
| **1** | Admin should be able to login with valid credentials | 1.Enter valid email<br>2.Enter valid password<br>3.Click on login button | Navigate to the home page |
| **2** | Show error message when try to login with invalid passwords | 4.Enter invalid email<br>5.Enter invalid password<br>6.Click on login button | Show error message saying Username or password is incorrect |

### Test Scenario 2 - View Employee Profiles

| TC Id | Test Objective | Steps | Expected Output |
|---|---|---|---|
| **1** | Admin should be able to view a list of employees in the organization | 1.Click on "users" icon on the top nav bar | View list of users |
| **2** | Admin should be able to view the profile of an employee in the organization | 2.Click on the name of the user | View account information, settings and permissions |

**Test Scenario 3 - Review leaves and approve/reject them**

| TC Id | Test Objective | Steps | Expected Output |
|---|---|---|---|
| 1 | Admin should be able to view a list of leave requests from employees in the organization | 1.Click on “switch to actiPLAN”<br>2.Click on “Leave management” icon in top nav bar<br>3.Click “leave time requests” | View list of all the |
| 2 | Admin should be able to approve user leave request | 4.Click on the name of the user<br>5.Filter by status<br>6.Select requests to approve<br>7.Click on approve button | Status changed to “approved” status |
| 3 | Admin should be able to reject user leave request | 8.Click on the name of the user<br>9.Filter by status<br>10. Select requests to reject<br>11. Click on “reject “button | Status changed to “Rejected” status |

**Test Scenario 4 - Review attendance**

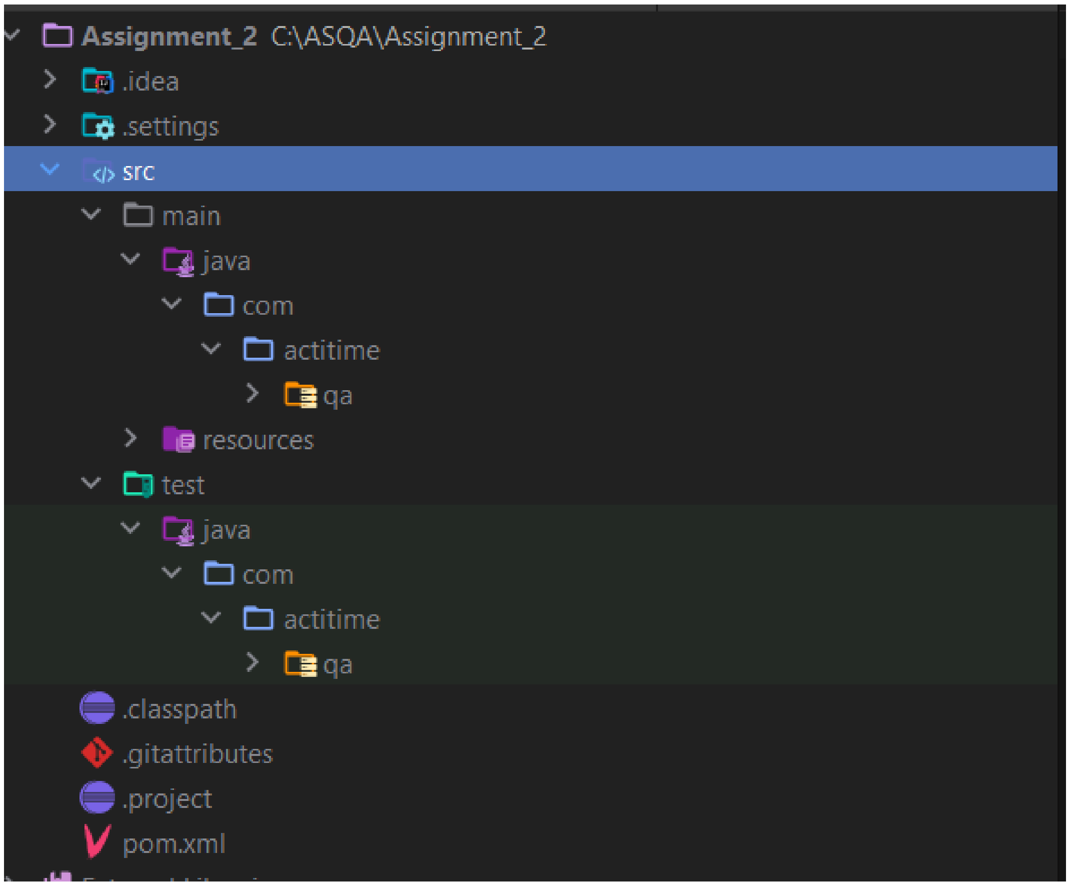
| TC Id | Test Objective | Steps | Expected Output |
|---|---|---|---|
| 1 | Admin should be able to view time track of a employee in the organization | 1.Click on “view time track”<br>2.Select employee name from dropdown “View time-track for” | View time entries and analytics of the selected employee |

## Test Scenario 5 - Approve/reject time-track

| TC Id | Test Objective | Steps | Expected Output |
|---|---|---|---|
| **1** | Admin should be able to view a list of time sheets submitted by employees in the organization | 1. Click on “Approve time track” | View list of all the time sheets submitted by all users. Groped by user’s name |
| **2** | Admin should be able to view detailed time track of a submitted time sheet by a user | 2. Click on the row | View detailed time tracks |
| **3** | Admin should be able to approve or reject time sheets | 3. Select the checkboxes to approve/reject<br>4. Click approve/reject button | Status changes as “Approved” or “Rejected” |

## 2. Build Hybrid Test Framework

I have initialized a maven project in Intelij Idea IDE and installed below dependencies in order to functioning of the Hybrid test framework.



```
Assignment_2  C:\ASQA\Assignment_2
  .idea
  .settings
  src
    main
      java
        com
          actitime
            qa
      resources
    test
      java
        com
          actitime
            qa
  .classpath
  .gitattributes
  .project
  pom.xml
```

**<u>Dependencies</u>**

1. Selenium-java
2. Apache POI API
3. Extend Reports
4. Commons IO
5. Log4j

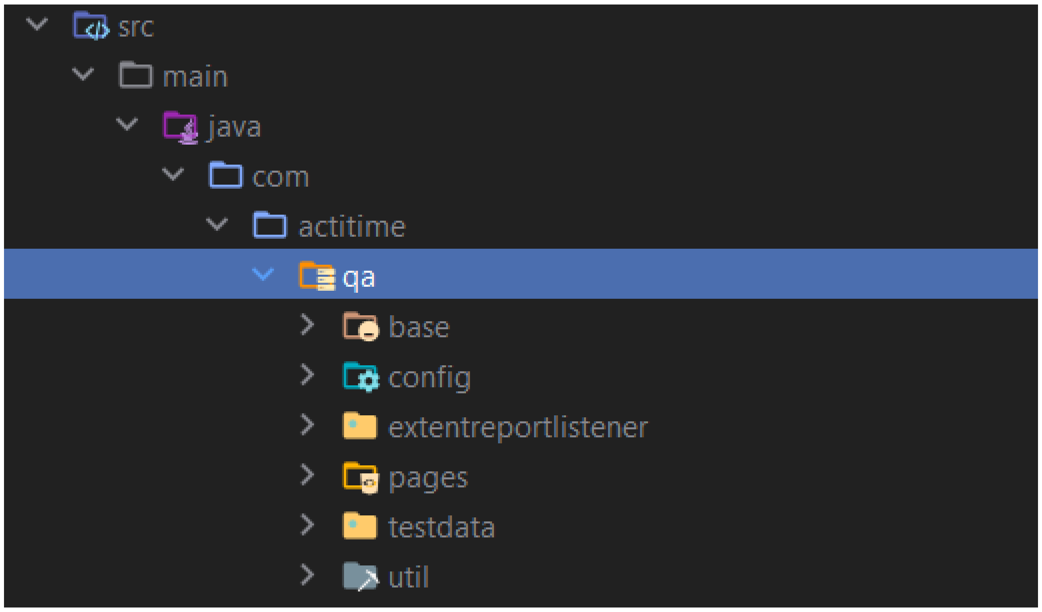
These dependencies are listed in the pom.xml file in the maven project folder.

Now it has the basic structure for the framework with Page Object Model + Data Driven framework + TestNG

## 3. Create the corresponding packages for Scripting

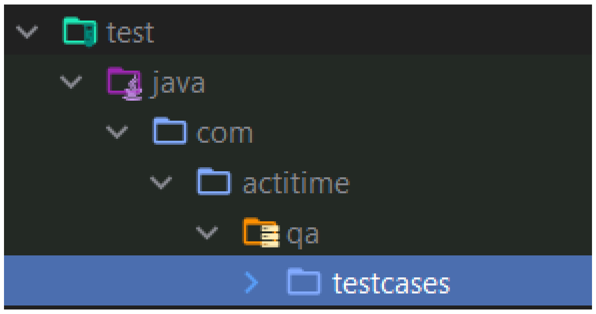
Then I have created packages required for (in src)

- Page Layer
- Test Base
- Configuration
- Utilities
- Test Data
- Listeners



```
src
  main
    java
      com
        actitime
          qa
            base
            config
            extentreportlistener
            pages
            testdata
            util
```

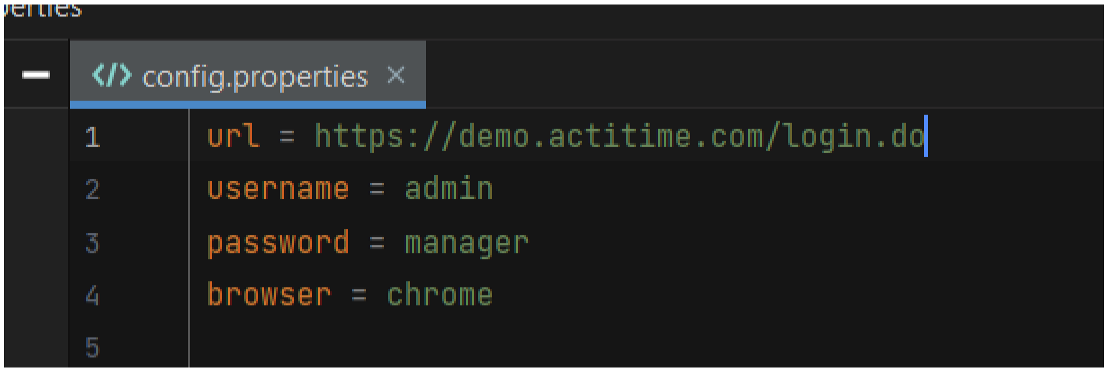
Also created a package as test layer in test folder



```
test
  java
    com
      actitime
        qa
          testcases
```

Now the project folder structure is done and next is to add corresponding files.

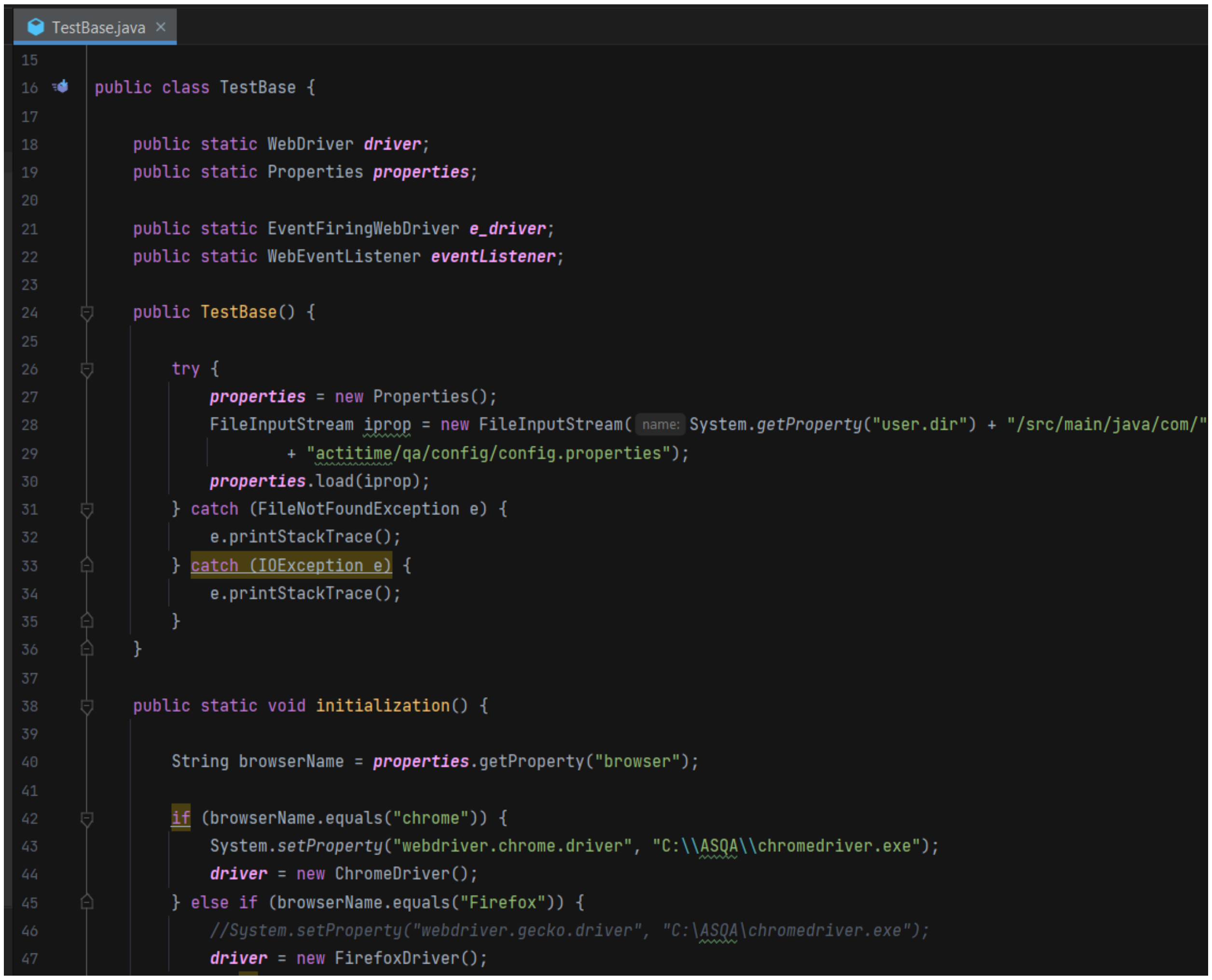
As next step, first I have created a configuration file ("config.properties") inside the config package and added required variables to it.



config.properties

```
url = https://demo.actitime.com/login.do
username = admin
password = manager
browser = chrome

```

Then initialized a test base by creating class named Testbase inside the package base.



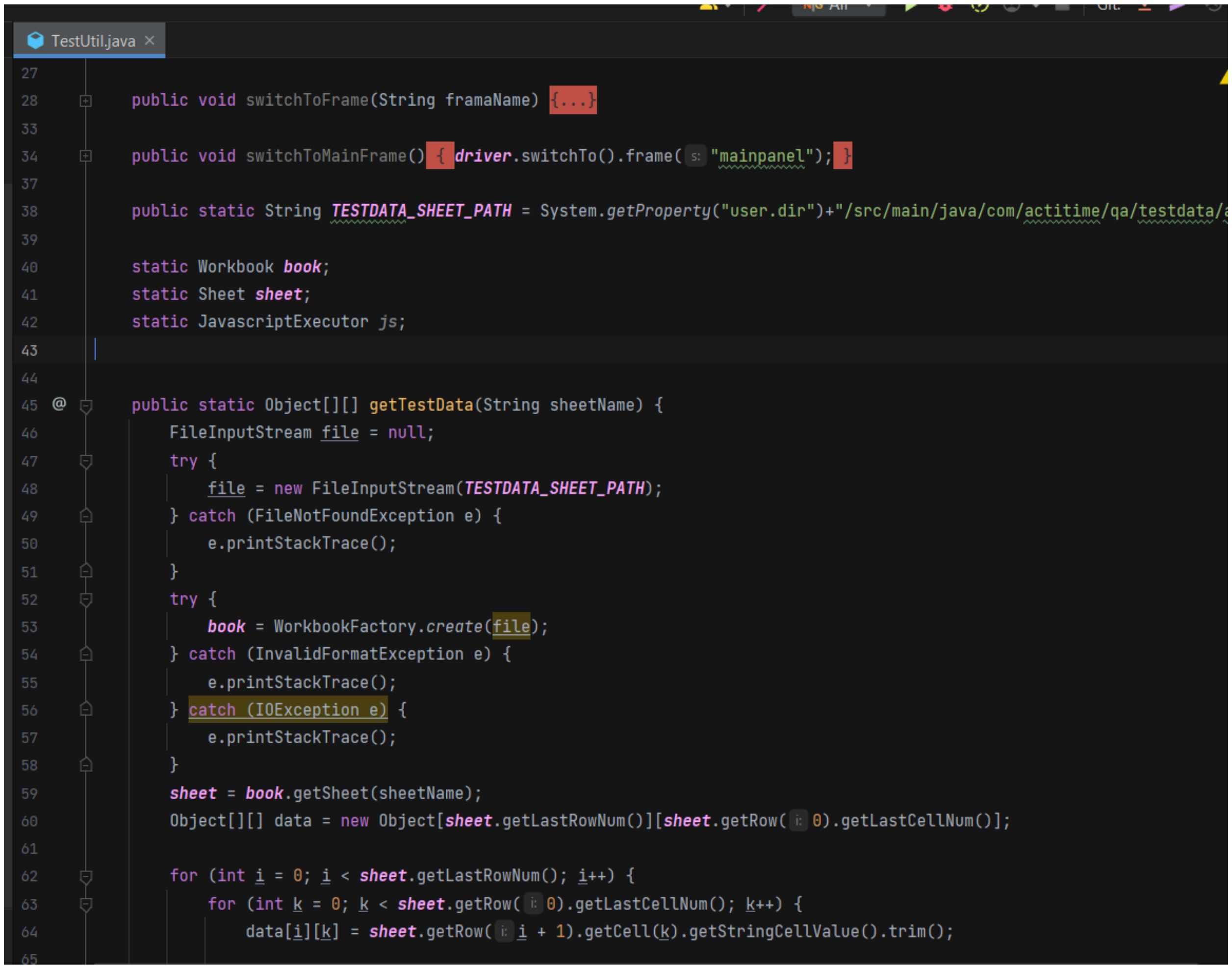
TestBase.java

```

public class TestBase {

    public static WebDriver driver;
    public static Properties properties;

    public static EventFiringWebDriver e_driver;
    public static WebEventListener eventListener;

    public TestBase() {

        try {
            properties = new Properties();
            FileInputStream iprop = new FileInputStream( name: System.getProperty("user.dir") + "/src/main/java/com/"
                    + "actitime/qa/config/config.properties");
            properties.load(iprop);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void initialization() {

        String browserName = properties.getProperty("browser");

        if (browserName.equals("chrome")) {
            System.setProperty("webdriver.chrome.driver", "C:\\ASQA\\chromedriver.exe");
            driver = new ChromeDriver();
        } else if (browserName.equals("Firefox")) {
            //System.setProperty("webdriver.gecko.driver", "C:\ASQA\chromedriver.exe");
            driver = new FirefoxDriver();
```

Here it contains the code fragment related to setting the properties required to run the test suite and initialization part by setting the chrome driver as web driver, setting the screen size, setting the event listeners and setting the timeouts.

The eventlisteners are implemented in the WebEventListener class inside the utils package.
It is implemented by WebDriverEventListener provided by selenium.

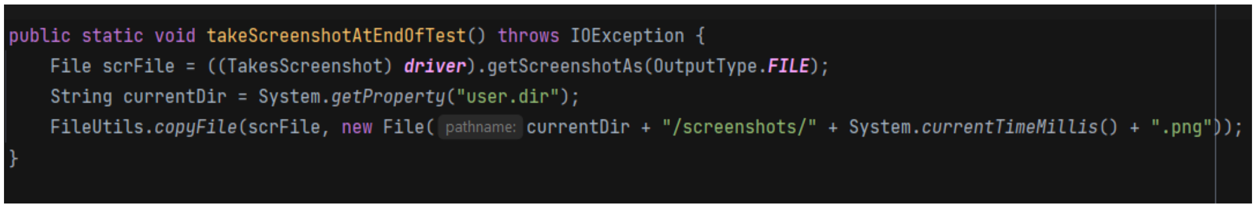
Then I have added some util functions which will important when using the data stored in Excel sheets.



TestUtil.java

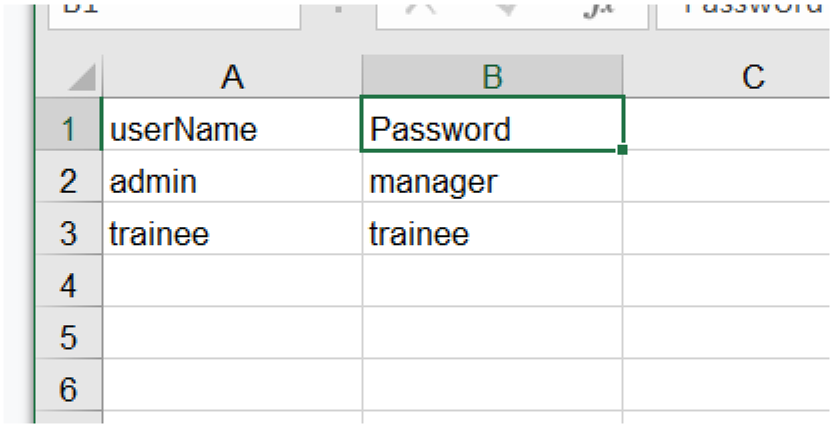
```java
    public void switchToFrame(String framaName) {...}

    public void switchToMainFrame() { driver.switchTo().frame( s: "mainpanel"); }

    public static String TESTDATA_SHEET_PATH = System.getProperty("user.dir")+"/src/main/java/com/actitime/qa/testdata/a

    static Workbook book;
    static Sheet sheet;
    static JavascriptExecutor js;


    public static Object[][] getTestData(String sheetName) {
        FileInputStream file = null;
        try {
            file = new FileInputStream(TESTDATA_SHEET_PATH);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        try {
            book = WorkbookFactory.create(file);
        } catch (InvalidFormatException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        sheet = book.getSheet(sheetName);
        Object[][] data = new Object[sheet.getLastRowNum()][sheet.getRow( i: 0).getLastCellNum()];

        for (int i = 0; i < sheet.getLastRowNum(); i++) {
            for (int k = 0; k < sheet.getRow( i: 0).getLastCellNum(); k++) {
                data[i][k] = sheet.getRow( i: i + 1).getCell(k).getStringCellValue().trim();
```

Also a util to take screenshot and save once a test case is finished.



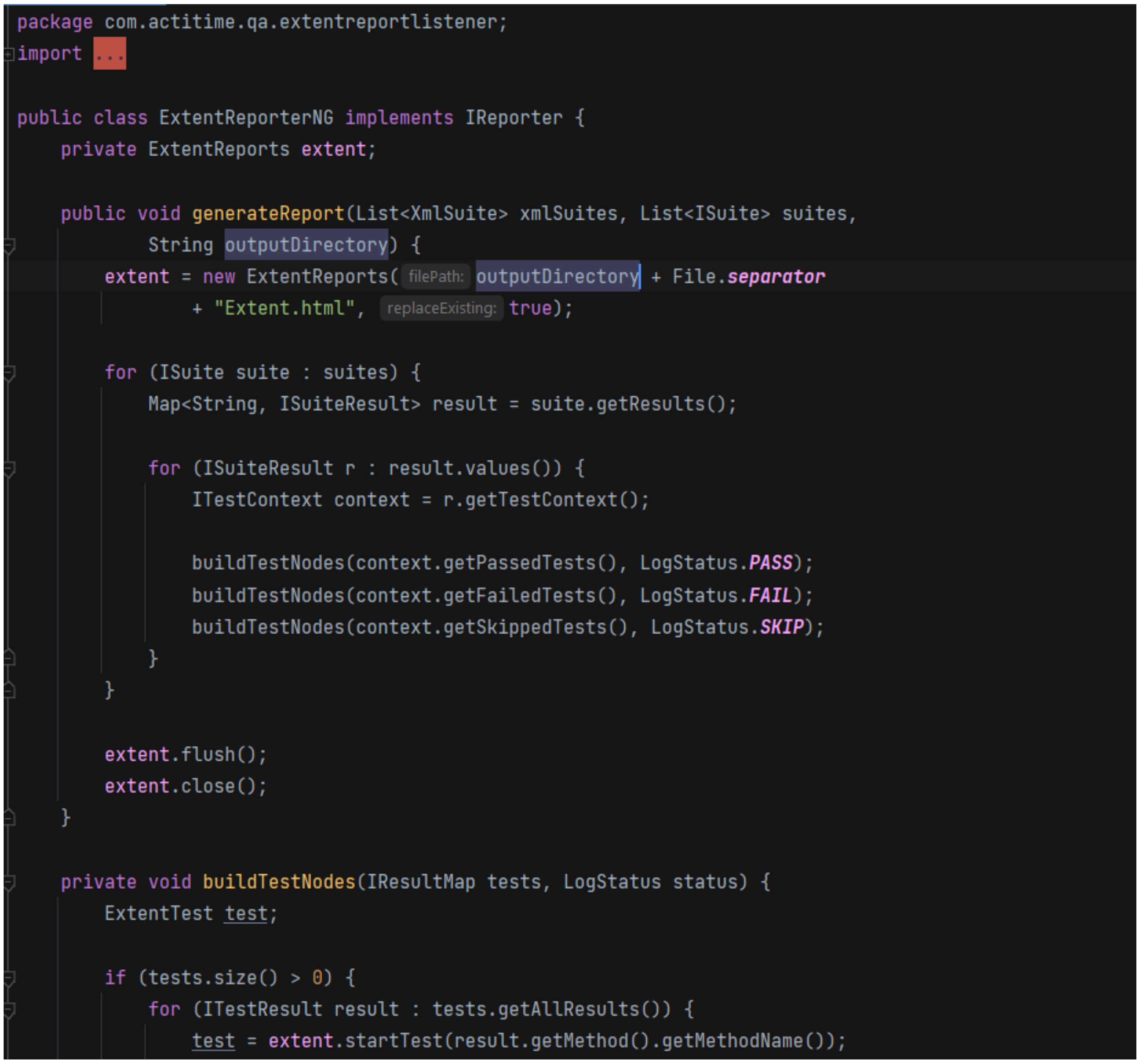
```
public static void takeScreenshotAtEndOfTest() throws IOException {
    File scrFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
    String currentDir = System.getProperty("user.dir");
    FileUtils.copyFile(scrFile, new File( pathname: currentDir + "/screenshots/" + System.currentTimeMillis() + ".png"));
}
```

Inside the testdata package, I have added a excel file which includes username and passwords required for login.



|  | A | B | C |
|---|---|---|---|
| 1 | userName | Password |  |
| 2 | admin | manager |  |
| 3 | trainee | trainee |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

For report generation, another listener is implemented and added to the package extentreportlistener.

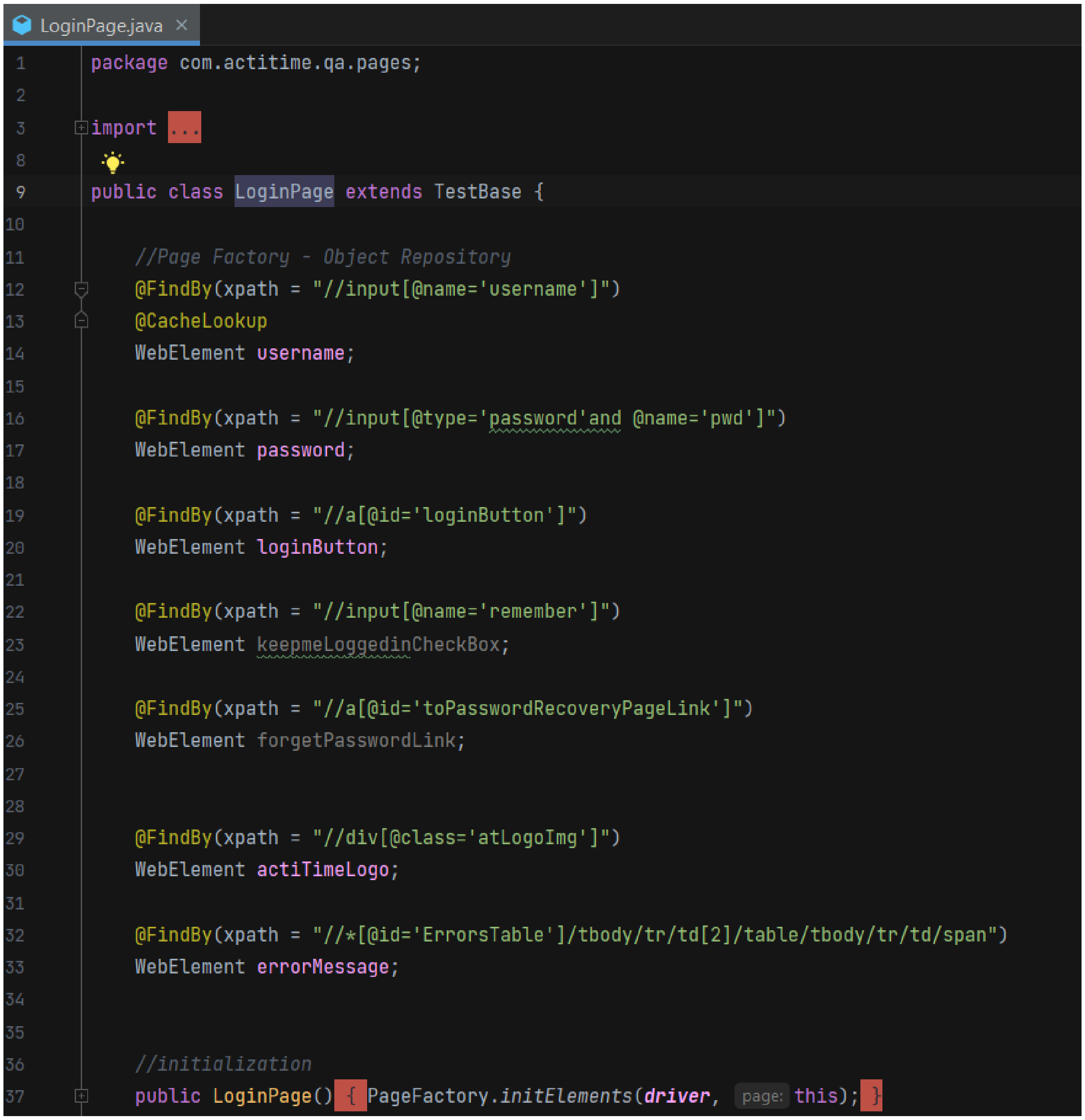


```java
package com.actitime.qa.extentreportlistener;
import ...

public class ExtentReporterNG implements IReporter {
    private ExtentReports extent;

    public void generateReport(List<XmlSuite> xmlSuites, List<ISuite> suites,
            String outputDirectory) {
        extent = new ExtentReports( filePath: outputDirectory + File.separator
                + "Extent.html",  replaceExisting: true);

        for (ISuite suite : suites) {
            Map<String, ISuiteResult> result = suite.getResults();

            for (ISuiteResult r : result.values()) {
                ITestContext context = r.getTestContext();

                buildTestNodes(context.getPassedTests(), LogStatus.PASS);
                buildTestNodes(context.getFailedTests(), LogStatus.FAIL);
                buildTestNodes(context.getSkippedTests(), LogStatus.SKIP);
            }
        }

        extent.flush();
        extent.close();
    }

    private void buildTestNodes(IResultMap tests, LogStatus status) {
        ExtentTest test;

        if (tests.size() > 0) {
            for (ITestResult result : tests.getAllResults()) {
                test = extent.startTest(result.getMethod().getMethodName());
```

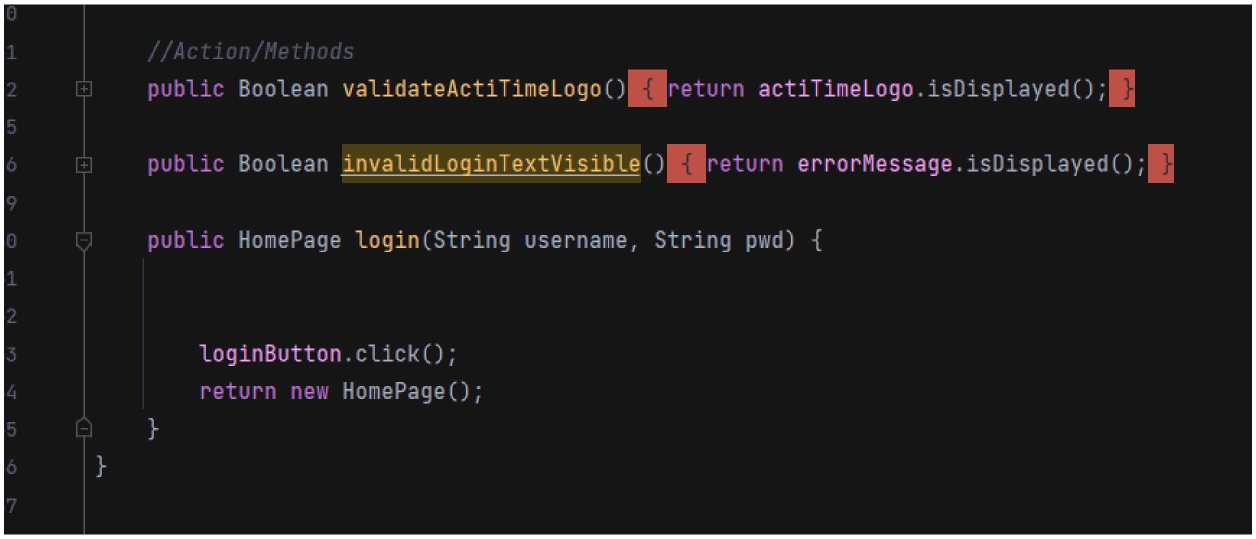
Now all the initializations are done and now it is the time to implement the page layer.

For the every page that we are going to automate the test, we need to create classes for each test objects using page factory. That class includes the html elements as attributes and they can be referred by xpath or name or id.

**Login page**



```java
package com.actitime.qa.pages;

import ...

public class LoginPage extends TestBase {

    //Page Factory - Object Repository
    @FindBy(xpath = "//input[@name='username']")
    @CacheLookup
    WebElement username;

    @FindBy(xpath = "//input[@type='password'and @name='pwd']")
    WebElement password;

    @FindBy(xpath = "//a[@id='loginButton']")
    WebElement loginButton;

    @FindBy(xpath = "//input[@name='remember']")
    WebElement keepmeLoggedinCheckBox;

    @FindBy(xpath = "//a[@id='toPasswordRecoveryPageLink']")
    WebElement forgetPasswordLink;


    @FindBy(xpath = "//div[@class='atLogoImg']")
    WebElement actiTimeLogo;

    @FindBy(xpath = "//*[@id='ErrorsTable']/tbody/tr/td[2]/table/tbody/tr/td/span")
    WebElement errorMessage;


    //initialization
    public LoginPage() { PageFactory.initElements(driver, page: this); }
```

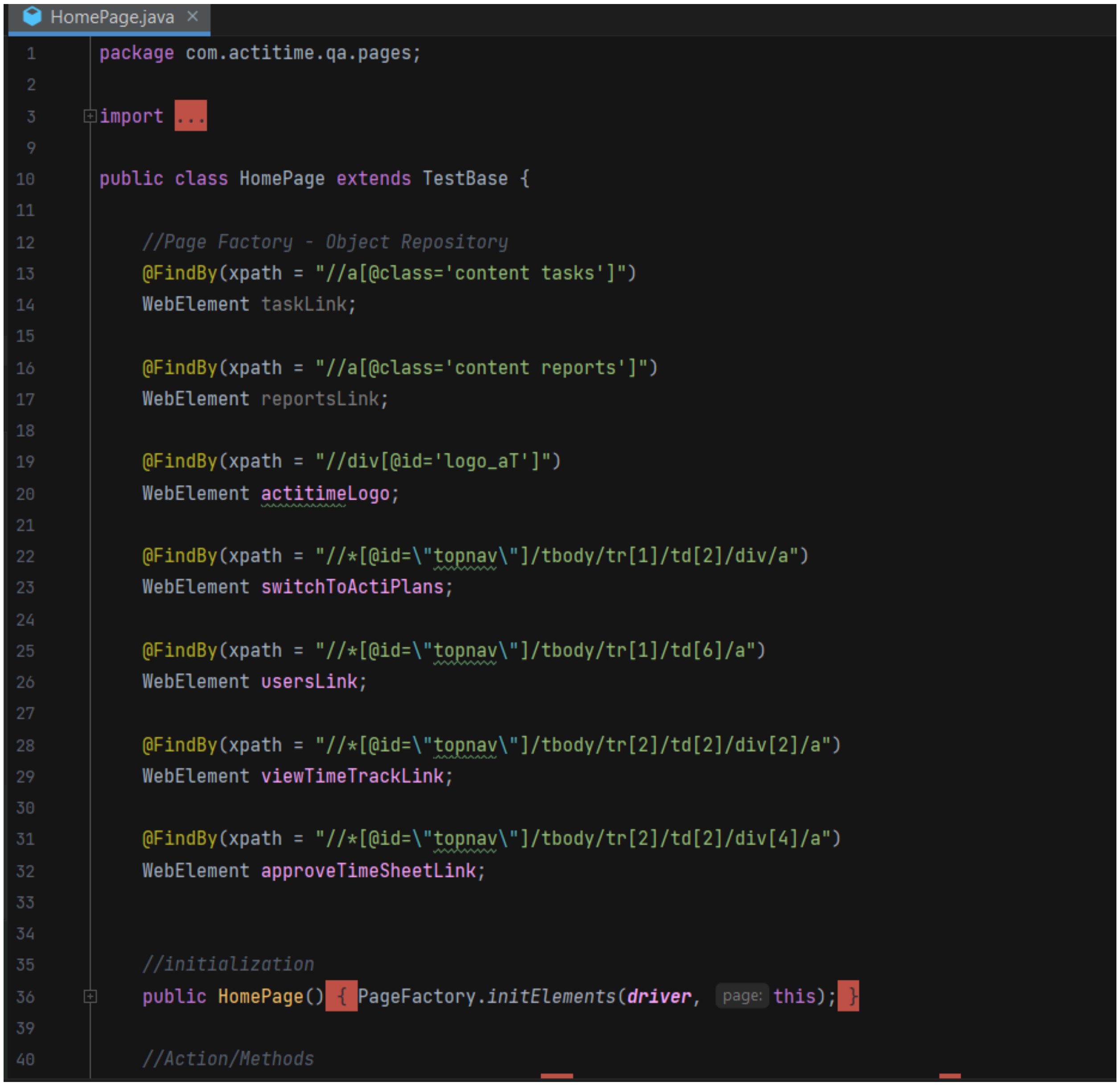

```
    //Action/Methods
    public Boolean validateActiTimeLogo() { return actiTimeLogo.isDisplayed(); }

    public Boolean invalidLoginTextVisible() { return errorMessage.isDisplayed(); }

    public HomePage login(String username, String pwd) {


        loginButton.click();
        return new HomePage();
    }
}
```

The loginPage.java file includes attributes related to the username field, password field and login button. It also includes forget password link and keep me logged in checkbox which are not related for the test scenario.

Methods in the class act as the actions. Contructor act as initializer. And I have implemented a methods to validate actitime logo to verify that page is loaded successfully and a method to login by providing username and password.

Likewise I have implemented the page model for home page.

## Home Page



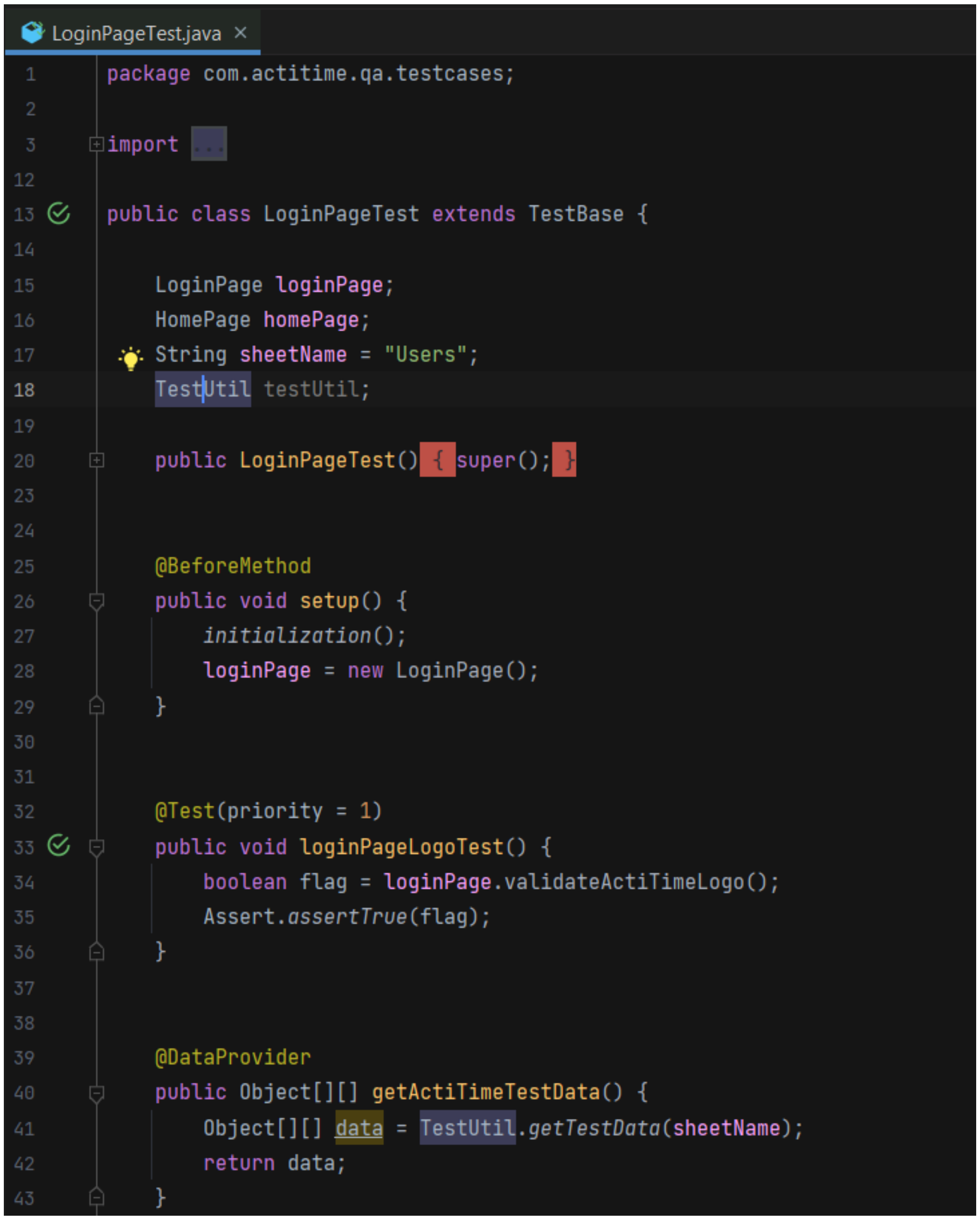
HomePage.java

```java
package com.actitime.qa.pages;

import ...

public class HomePage extends TestBase {

    //Page Factory - Object Repository
    @FindBy(xpath = "//a[@class='content tasks']")
    WebElement taskLink;

    @FindBy(xpath = "//a[@class='content reports']")
    WebElement reportsLink;

    @FindBy(xpath = "//div[@id='logo_aT']")
    WebElement actitimeLogo;

    @FindBy(xpath = "//*[@id=\"topnav\"]/tbody/tr[1]/td[2]/div/a")
    WebElement switchToActiPlans;

    @FindBy(xpath = "//*[@id=\"topnav\"]/tbody/tr[1]/td[6]/a")
    WebElement usersLink;

    @FindBy(xpath = "//*[@id=\"topnav\"]/tbody/tr[2]/td[2]/div[2]/a")
    WebElement viewTimeTrackLink;

    @FindBy(xpath = "//*[@id=\"topnav\"]/tbody/tr[2]/td[2]/div[4]/a")
    WebElement approveTimeSheetLink;


    //initialization
    public HomePage() { PageFactory.initElements(driver, page: this); }

    //Action/Methods
```

Like wise I have implemented page models for all the required pages that need be tested.

Then under the test folder, we can implement our test cases.

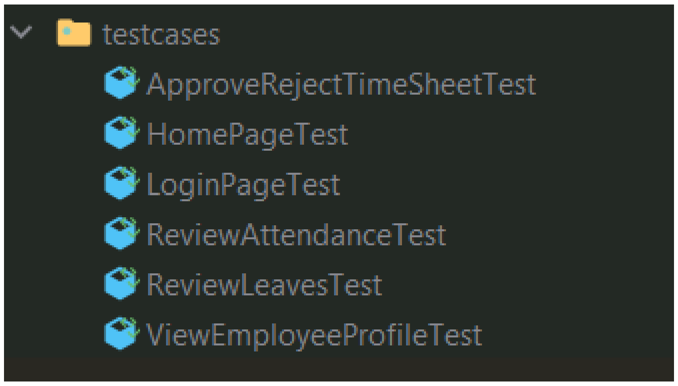
**Test case for login page**

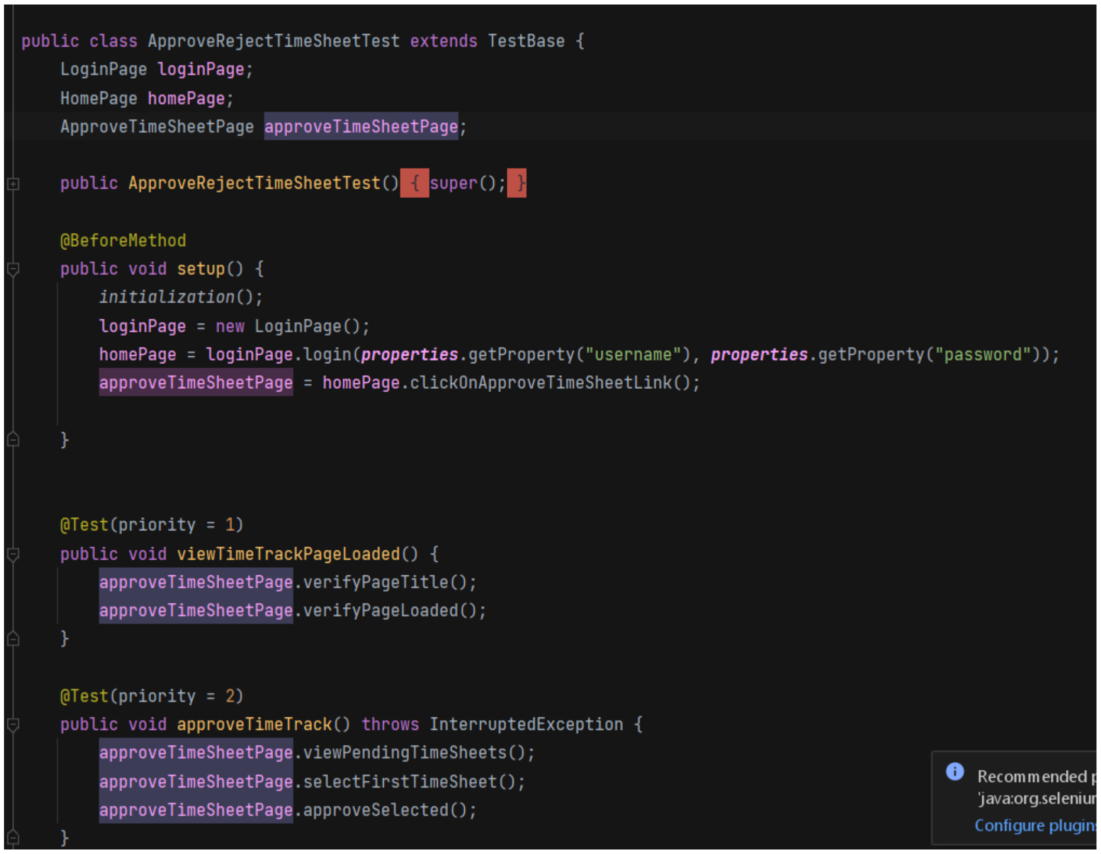


```java
package com.actitime.qa.testcases;

import ...

public class LoginPageTest extends TestBase {

    LoginPage loginPage;
    HomePage homePage;
    String sheetName = "Users";
    TestUtil testUtil;

    public LoginPageTest() { super(); }


    @BeforeMethod
    public void setup() {
        initialization();
        loginPage = new LoginPage();
    }


    @Test(priority = 1)
    public void loginPageLogoTest() {
        boolean flag = loginPage.validateActiTimeLogo();
        Assert.assertTrue(flag);
    }


    @DataProvider
    public Object[][] getActiTimeTestData() {
        Object[][] data = TestUtil.getTestData(sheetName);
        return data;
    }
```

The test case implementation class is a super class of Testbase class implemented earlier. constructor will call the super class constructor and that will provide access to driver.
It has a method named "setup" which which do the initialization. It is annotated with BeforeMethod so it will execute before executing the test case. Next methods with "Test" annotations execute as test steps. A DataProvider also added to get data from excel and execute test case. Finally the method with AfterMethod annotation will execute after the test steps execution.

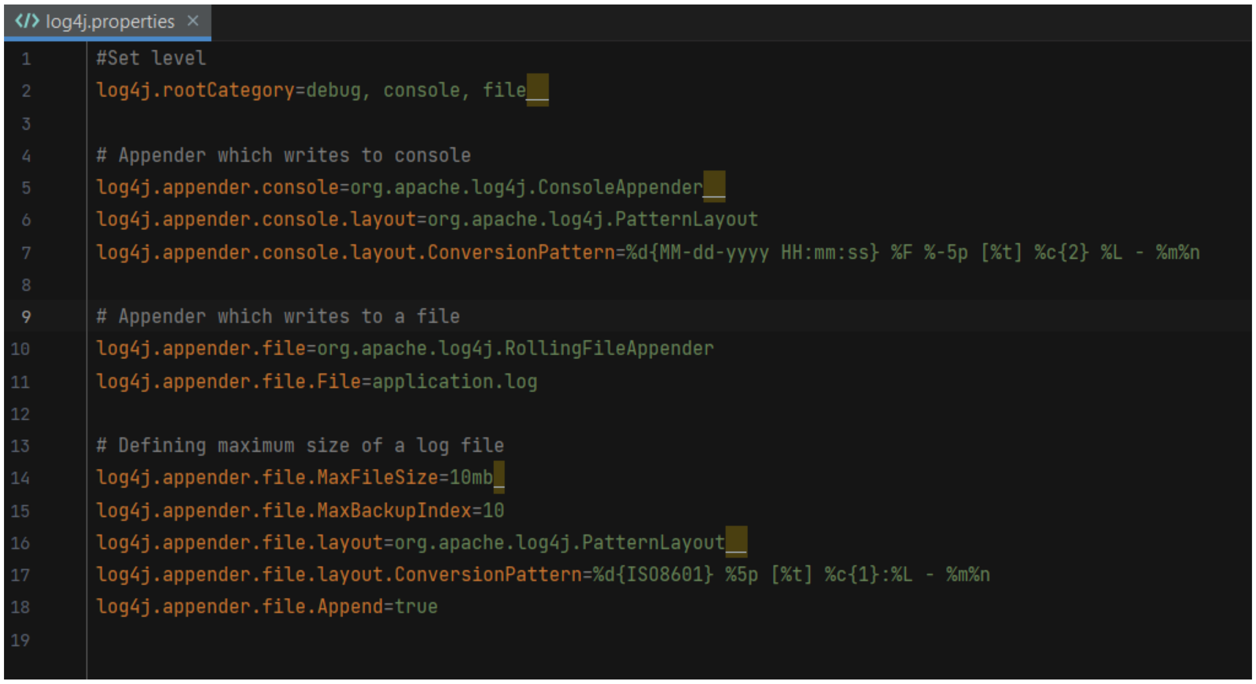
Likewise I have implemented classes for each draft test cases.





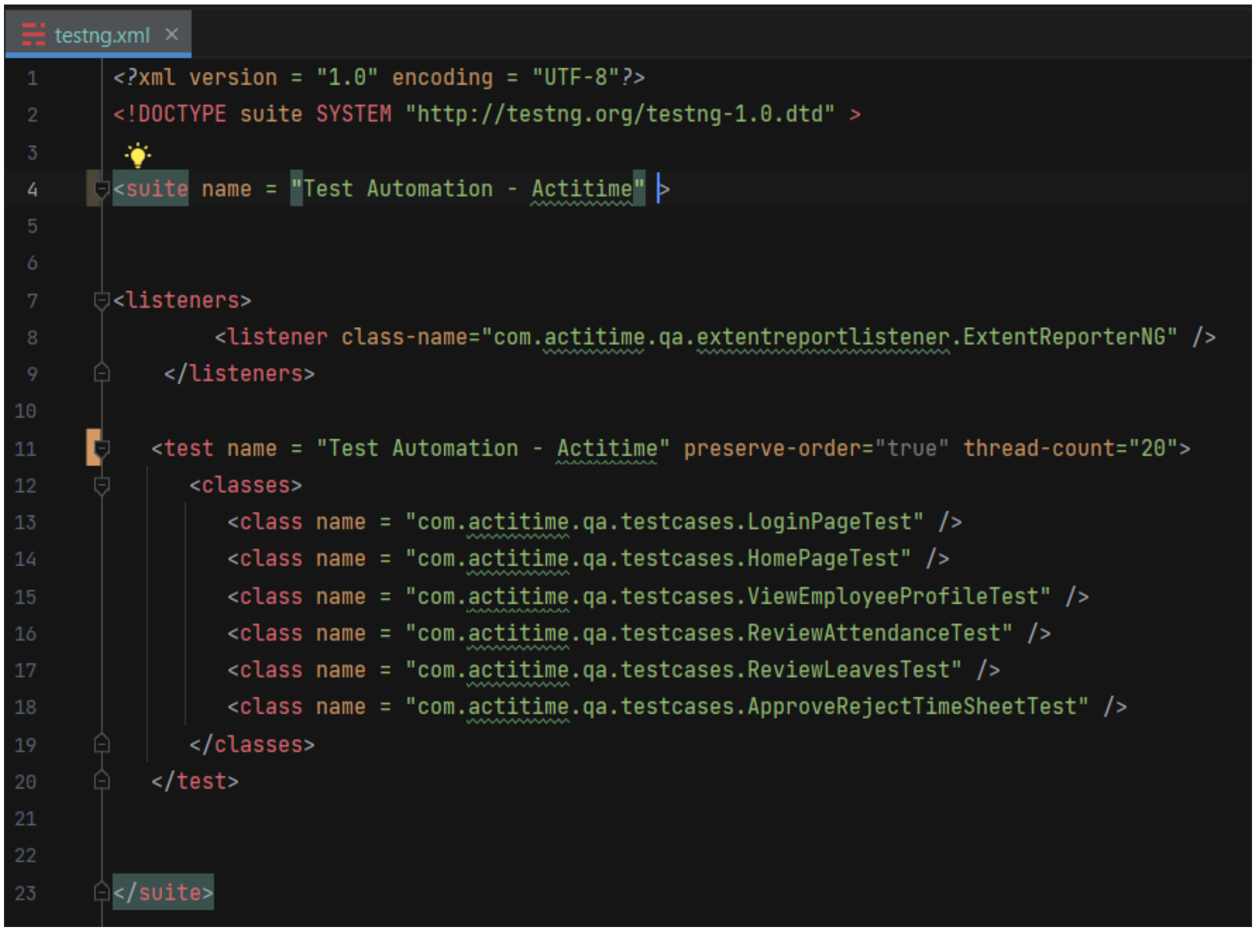
```
public class ApproveRejectTimeSheetTest extends TestBase {
    LoginPage loginPage;
    HomePage homePage;
    ApproveTimeSheetPage approveTimeSheetPage;

    public ApproveRejectTimeSheetTest() { super(); }

    @BeforeMethod
    public void setup() {
        initialization();
        loginPage = new LoginPage();
        homePage = loginPage.login(properties.getProperty("username"), properties.getProperty("password"));
        approveTimeSheetPage = homePage.clickOnApproveTimeSheetLink();

    }


    @Test(priority = 1)
    public void viewTimeTrackPageLoaded() {
        approveTimeSheetPage.verifyPageTitle();
        approveTimeSheetPage.verifyPageLoaded();
    }

    @Test(priority = 2)
    public void approveTimeTrack() throws InterruptedException {
        approveTimeSheetPage.viewPendingTimeSheets();
        approveTimeSheetPage.selectFirstTimeSheet();
        approveTimeSheetPage.approveSelected();
    }
```

log4j.properties file includes configurations required to logging output to the log files.



```
#Set level
log4j.rootCategory=debug, console, file

# Appender which writes to console
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern=%d{MM-dd-yyyy HH:mm:ss} %F %-5p [%t] %c{2} %L - %m%n

# Appender which writes to a file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=application.log

# Defining maximum size of a log file
log4j.appender.file.MaxFileSize=10mb
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ISO8601} %5p [%t] %c{1}:%L - %m%n
log4j.appender.file.Append=true
```

Inside the resource folder, it is required to define the test suite to TestNG with the testing.xml file.



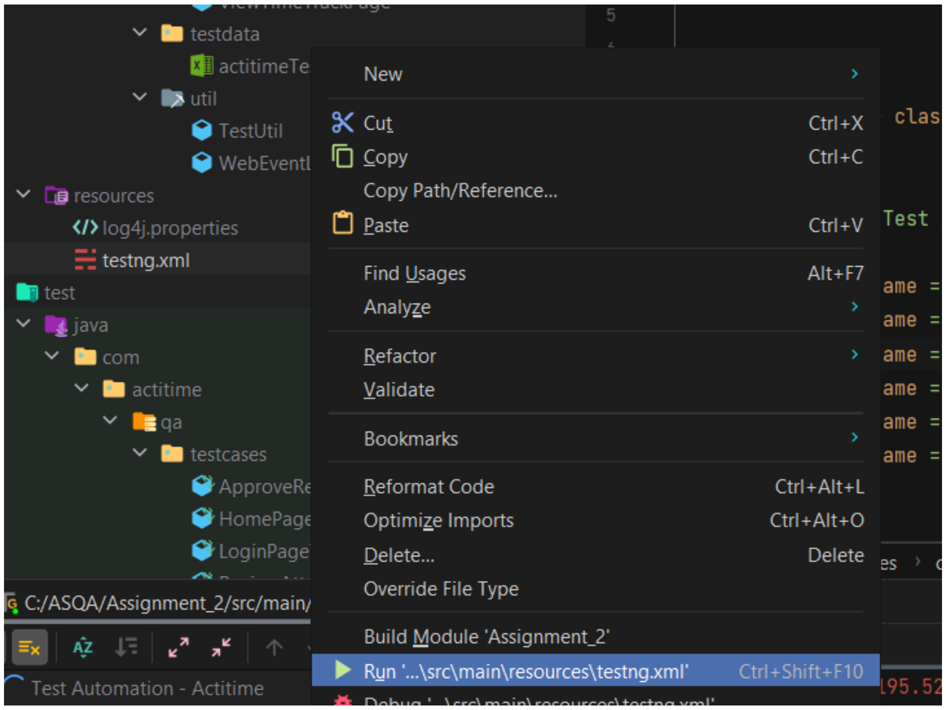
testng.xml

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name = "Test Automation - Actitime" >


<listeners>
        <listener class-name="com.actitime.qa.extentreportlistener.ExtentReporterNG" />
    </listeners>

   <test name = "Test Automation - Actitime" preserve-order="true" thread-count="20">
      <classes>
         <class name = "com.actitime.qa.testcases.LoginPageTest" />
         <class name = "com.actitime.qa.testcases.HomePageTest" />
         <class name = "com.actitime.qa.testcases.ViewEmployeeProfileTest" />
         <class name = "com.actitime.qa.testcases.ReviewAttendanceTest" />
         <class name = "com.actitime.qa.testcases.ReviewLeavesTest" />
         <class name = "com.actitime.qa.testcases.ApproveRejectTimeSheetTest" />
      </classes>
   </test>


</suite>
```

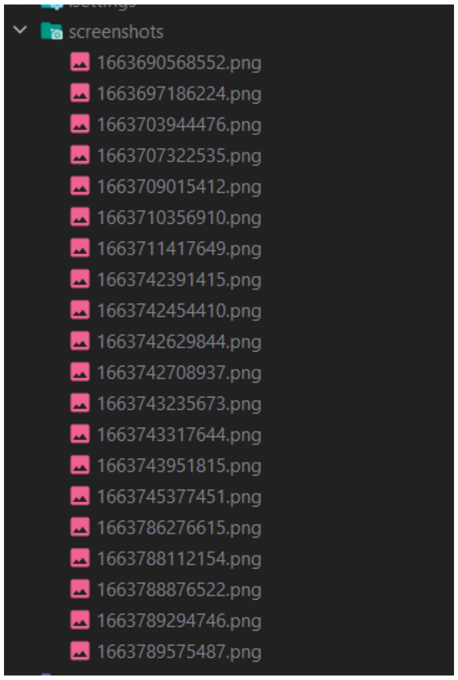
It contains the name of suite, listeners used and test classes used.

## 4. Execute Automation Scripts

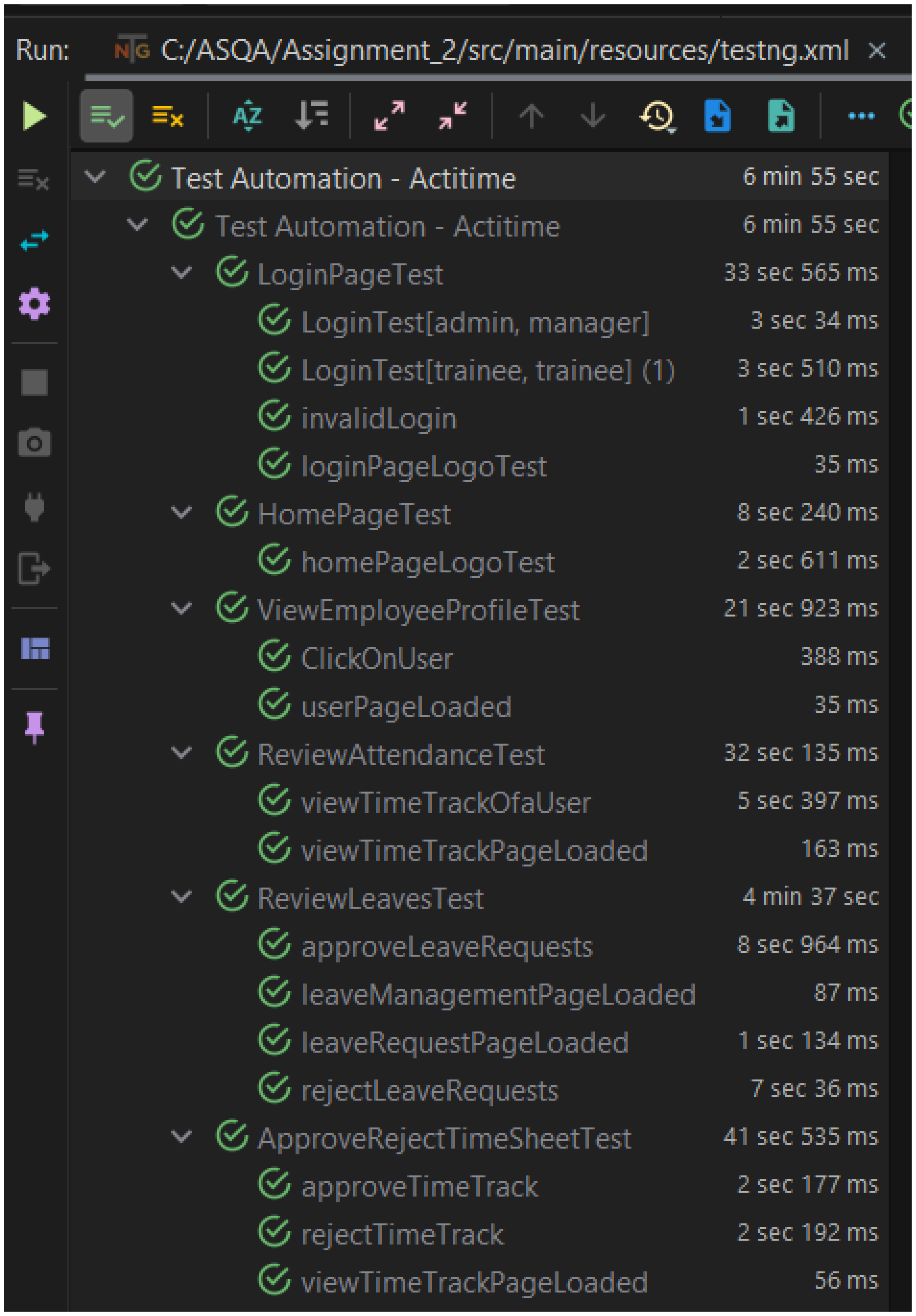
Finally tests can be executed by executing testing.xml file.



While running the tests, if errors are thrown, a screenshot related to that error is saved in “screenshots” folder.

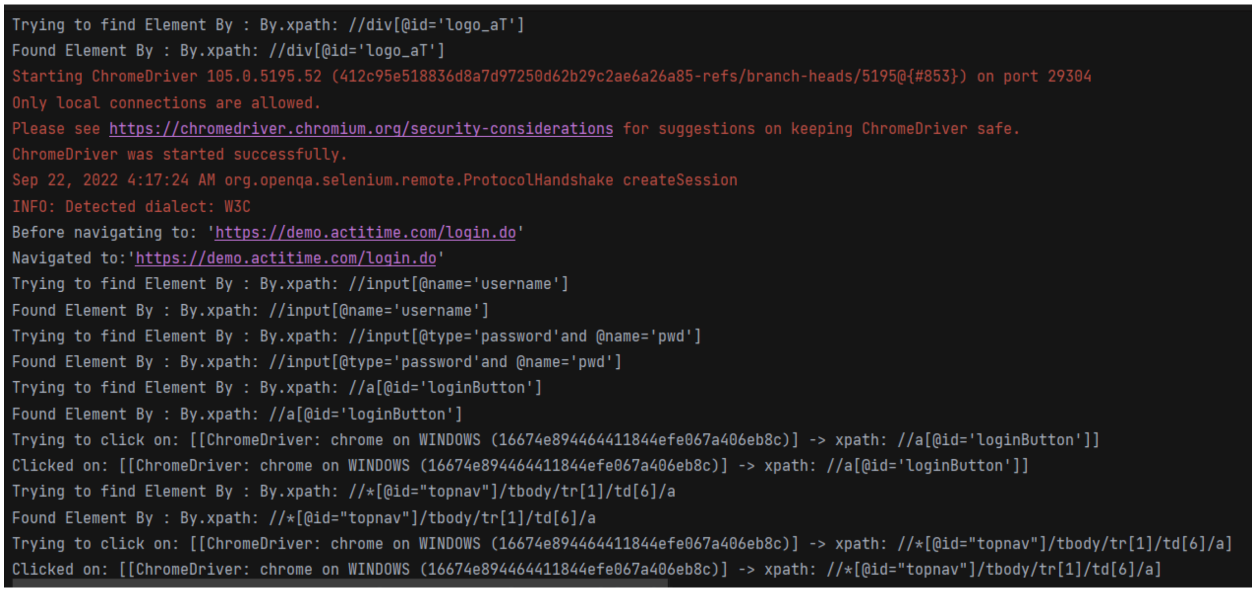


Test progress is visible in the test run window.

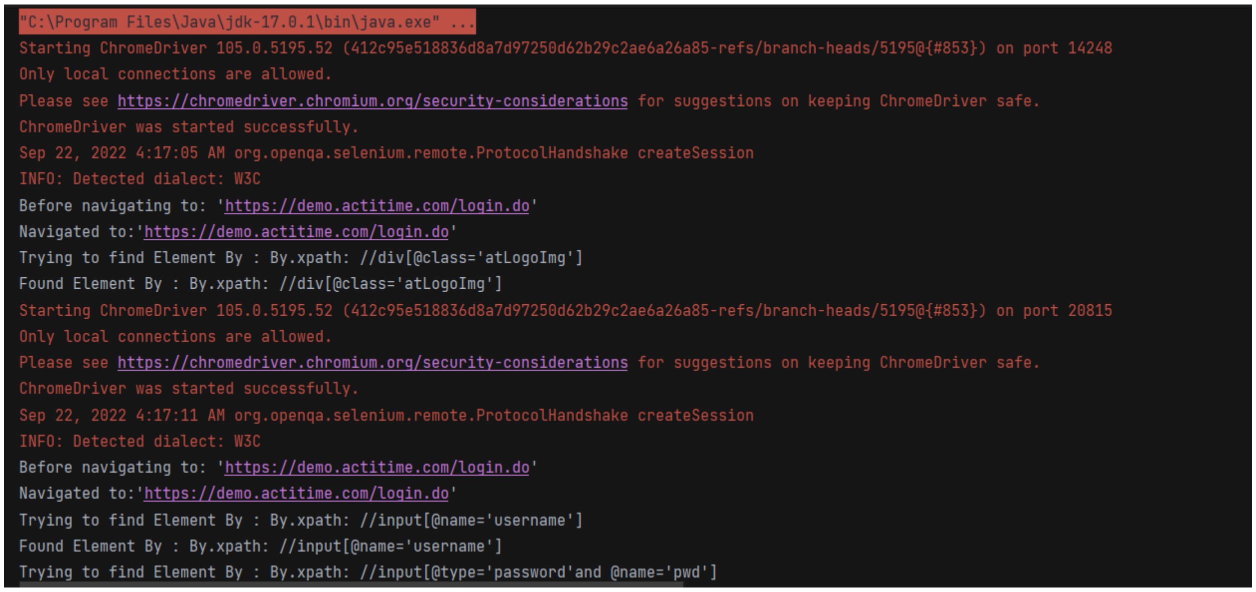


```
Run:  C:/ASQA/Assignment_2/src/main/resources/testng.xml

Test Automation - Actitime                        6 min 55 sec
  Test Automation - Actitime                      6 min 55 sec
    LoginPageTest                                 33 sec 565 ms
      LoginTest[admin, manager]                   3 sec 34 ms
      LoginTest[trainee, trainee] (1)             3 sec 510 ms
      invalidLogin                                1 sec 426 ms
      loginPageLogoTest                           35 ms
    HomePageTest                                  8 sec 240 ms
      homePageLogoTest                            2 sec 611 ms
    ViewEmployeeProfileTest                       21 sec 923 ms
      ClickOnUser                                 388 ms
      userPageLoaded                              35 ms
    ReviewAttendanceTest                          32 sec 135 ms
      viewTimeTrackOfaUser                        5 sec 397 ms
      viewTimeTrackPageLoaded                     163 ms
    ReviewLeavesTest                              4 min 37 sec
      approveLeaveRequests                        8 sec 964 ms
      leaveManagementPageLoaded                   87 ms
      leaveRequestPageLoaded                      1 sec 134 ms
      rejectLeaveRequests                         7 sec 36 ms
    ApproveRejectTimeSheetTest                    41 sec 535 ms
      approveTimeTrack                            2 sec 177 ms
      rejectTimeTrack                             2 sec 192 ms
      viewTimeTrackPageLoaded                     56 ms
```

Logs are logged with log4j

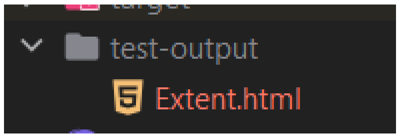


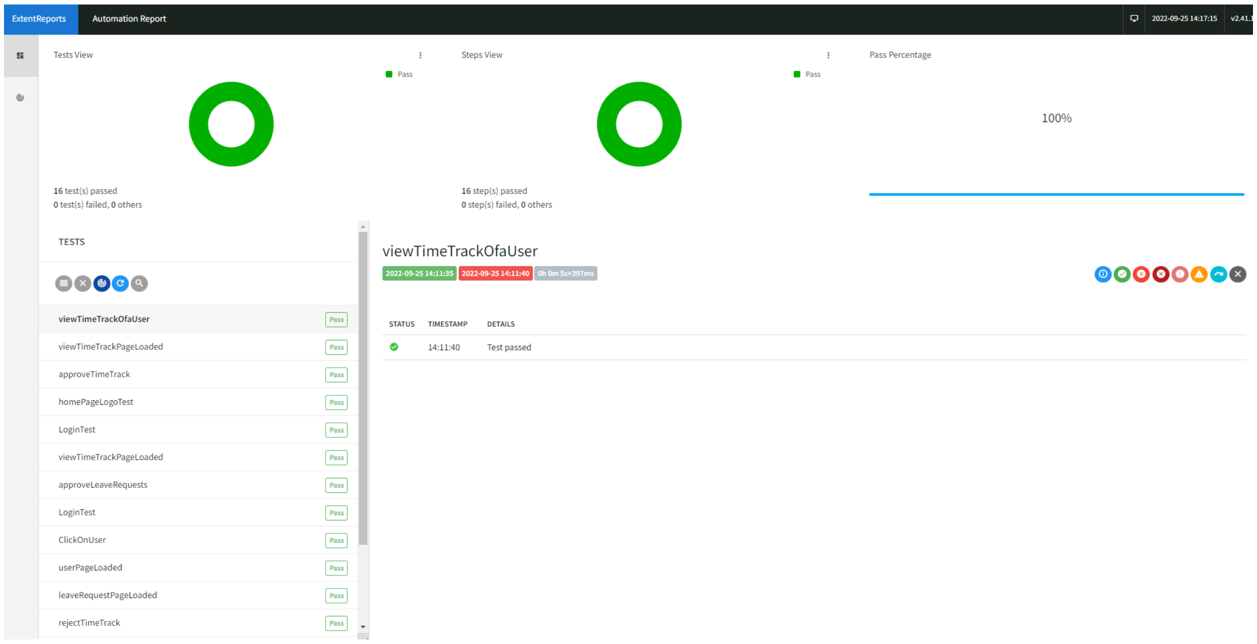
```
Trying to find Element By : By.xpath: //div[@id='logo_aT']
Found Element By : By.xpath: //div[@id='logo_aT']
Starting ChromeDriver 105.0.5195.52 (412c95e518836d8a7d97250d62b29c2ae6a26a85-refs/branch-heads/5195@{#853}) on port 29304
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Sep 22, 2022 4:17:24 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Before navigating to: 'https://demo.actitime.com/login.do'
Navigated to:'https://demo.actitime.com/login.do'
Trying to find Element By : By.xpath: //input[@name='username']
Found Element By : By.xpath: //input[@name='username']
Trying to find Element By : By.xpath: //input[@type='password'and @name='pwd']
Found Element By : By.xpath: //input[@type='password'and @name='pwd']
Trying to find Element By : By.xpath: //a[@id='loginButton']
Found Element By : By.xpath: //a[@id='loginButton']
Trying to click on: [[ChromeDriver: chrome on WINDOWS (16674e894464411844efe067a406eb8c)] -> xpath: //a[@id='loginButton']]
Clicked on: [[ChromeDriver: chrome on WINDOWS (16674e894464411844efe067a406eb8c)] -> xpath: //a[@id='loginButton']]
Trying to find Element By : By.xpath: //*[@id="topnav"]/tbody/tr[1]/td[6]/a
Found Element By : By.xpath: //*[@id="topnav"]/tbody/tr[1]/td[6]/a
Trying to click on: [[ChromeDriver: chrome on WINDOWS (16674e894464411844efe067a406eb8c)] -> xpath: //*[@id="topnav"]/tbody/tr[1]/td[6]/a]
Clicked on: [[ChromeDriver: chrome on WINDOWS (16674e894464411844efe067a406eb8c)] -> xpath: //*[@id="topnav"]/tbody/tr[1]/td[6]/a]
```



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" ...
Starting ChromeDriver 105.0.5195.52 (412c95e518836d8a7d97250d62b29c2ae6a26a85-refs/branch-heads/5195@{#853}) on port 14248
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Sep 22, 2022 4:17:05 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Before navigating to: 'https://demo.actitime.com/login.do'
Navigated to:'https://demo.actitime.com/login.do'
Trying to find Element By : By.xpath: //div[@class='atLogoImg']
Found Element By : By.xpath: //div[@class='atLogoImg']
Starting ChromeDriver 105.0.5195.52 (412c95e518836d8a7d97250d62b29c2ae6a26a85-refs/branch-heads/5195@{#853}) on port 20815
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Sep 22, 2022 4:17:11 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Before navigating to: 'https://demo.actitime.com/login.do'
Navigated to:'https://demo.actitime.com/login.do'
Trying to find Element By : By.xpath: //input[@name='username']
Found Element By : By.xpath: //input[@name='username']
Trying to find Element By : By.xpath: //input[@type='password'and @name='pwd']
```

Once done the folder named "test-output" is created and it includes the output of tests. The Extent report will show all the outputs.





16 test(s) passed
0 test(s) failed, 0 others

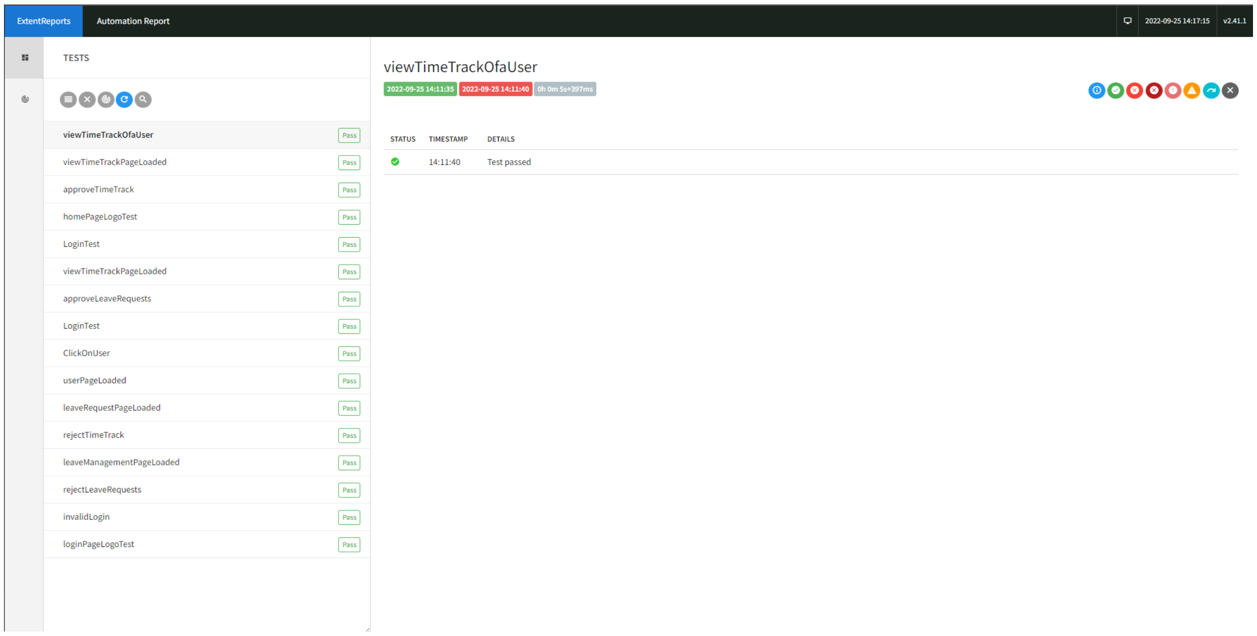
16 step(s) passed
0 step(s) failed, 0 others

| | |
|---|---|
| viewTimeTrackOfaUser | Pass |
| viewTimeTrackPageLoaded | Pass |
| approveTimeTrack | Pass |
| homePageLogoTest | Pass |
| LoginTest | Pass |
| viewTimeTrackPageLoaded | Pass |
| approveLeaveRequests | Pass |
| LoginTest | Pass |
| ClickOnUser | Pass |
| userPageLoaded | Pass |
| leaveRequestPageLoaded | Pass |
| rejectTimeTrack | Pass |

viewTimeTrackOfaUser

2022-09-25 14:11:35 2022-09-25 14:11:40 0h 0m 5s+397ms

| STATUS | TIMESTAMP | DETAILS |
|---|---|---|
| ✓ | 14:11:40 | Test passed |



| | |
|---|---|
| viewTimeTrackOfaUser | Pass |
| viewTimeTrackPageLoaded | Pass |
| approveTimeTrack | Pass |
| homePageLogoTest | Pass |
| LoginTest | Pass |
| viewTimeTrackPageLoaded | Pass |
| approveLeaveRequests | Pass |
| LoginTest | Pass |
| ClickOnUser | Pass |
| userPageLoaded | Pass |
| leaveRequestPageLoaded | Pass |
| rejectTimeTrack | Pass |
| leaveManagementPageLoaded | Pass |
| rejectLeaveRequests | Pass |
| invalidLogin | Pass |
| loginPageLogoTest | Pass |

viewTimeTrackOfaUser

2022-09-25 14:11:35 2022-09-25 14:11:40 0h 0m 5s+397ms

| STATUS | TIMESTAMP | DETAILS |
|---|---|---|
| ✓ | 14:11:40 | Test passed |

Github Repository URL - https://github.com/dumidu1998/hybrid-testing

Open in VS code editor - https://github.dev/dumidu1998/hybrid-testing

Extent Report public link - https://dumidu1998.github.io/hybrid-testing/

Path for the extent report in provided Zipped folder.
\Assignment_2\test-output\Extent.html