

# **IS4102 Advanced Software Quality Assurance**

## **Assignment 02**

### **Test Automation Report**

#### **TecRoot LK**



**Name-: W.D Kumudika**  
**Index No-: 20020597**

# Test Automation Strategy

## Scope Description

- **User Interaction Scenarios:** Verification of user actions like login, sign up to newsletter etc.
- **End-to-End Flow Validation:** Ensuring users can navigate to the login page and login to the SUT using correct login credentials.
- **Data-Driven Testing:** Inputs such as username, password are dynamically read from external data sources like to maximise coverage without hard-coding credentials inside the automation framework.

## Test Automation Framework

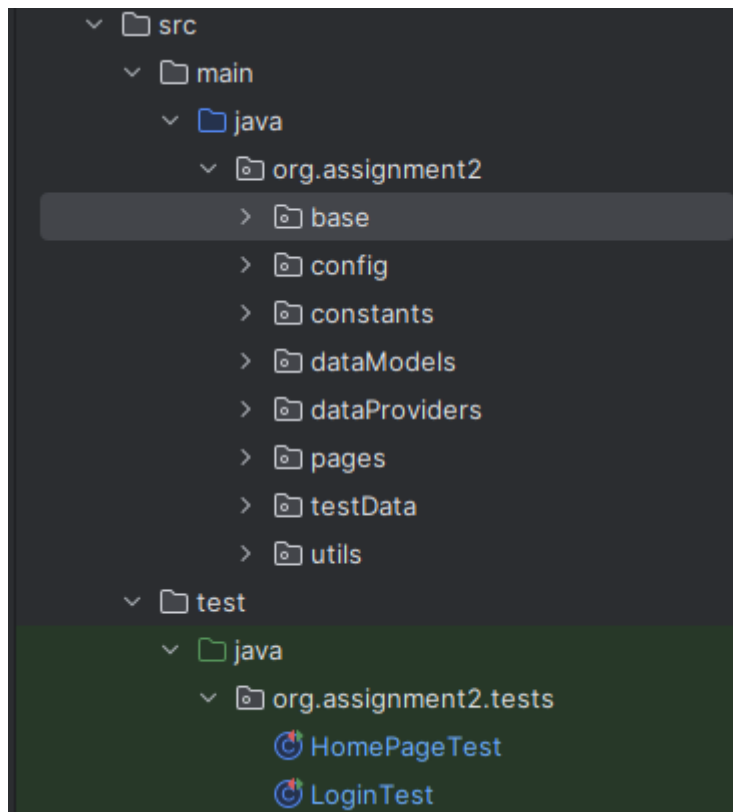
### Page Object Model Usage

- Page objects such as **HomePage** and **AccountPage** to represent the web elements inside respective pages and actions/interactions users can perform.
- Decouple tests for each web page from its web elements and actions.
- Utility classes to handle operations common to all tests, reading data from external data sources to facilitate data driven testing, parallel execution and manage configurations.

### Coding Standards

- **Exception Handling:** exceptions are handled and proper messages are logged when elements are not found or conditions aren't met, aiding test reliability and debugging.
- **Thread-Safe Execution:** By leveraging the ***ThreadLocal*** WebDriver instance in ***BrowserFactory***, parallel execution is achieved without interference between test threads.
- Comments and docstrings are used if necessary for better readability.
- Managed project structure.

## Project Structure



**Base Directory** - Contains BaseTest and BasePage classes.

**Config Directory** - contains configurations required for the project such as base url

**Data Providers** - data provide classes that feed test data from external data sources

**Pages** - Page classes that represent their web elements and actions users are performed

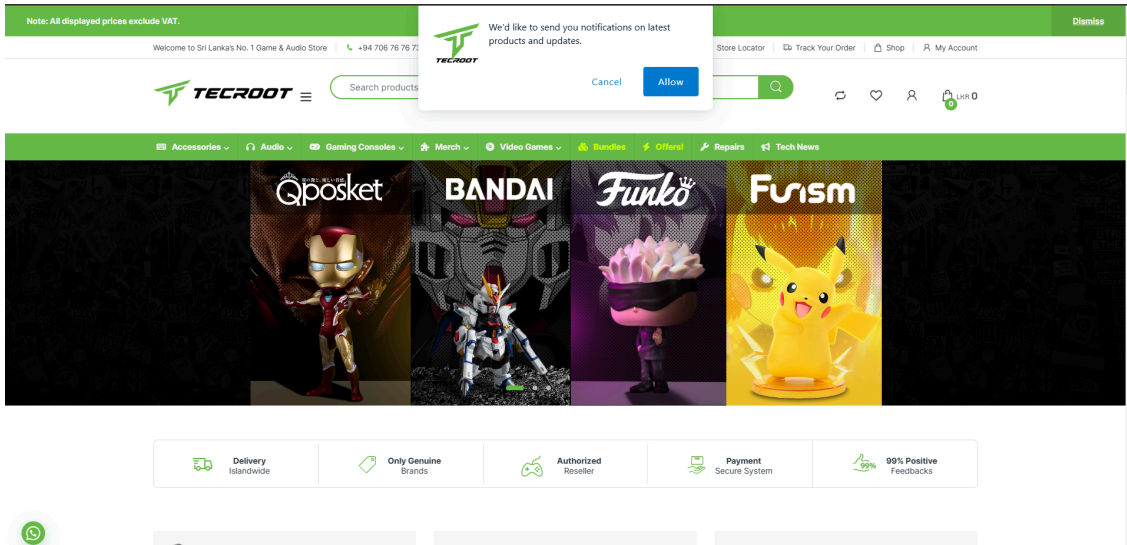
**Test Data** - contains external test data sources

**Utils** - utility functions

**Tests** - contains all the tests performed, one test class per web page

# Test Case Scenarios

## Home Page Tests



### Test 1:- Verify home page title

```
no usages DinithKumudika
@Test(priority=1)
public void verifyHomePageTitleTest(){
    String homePageTitle = homePage.verifyHomePageTitle();
    System.out.println(homePageTitle);
    Assert.assertEquals(homePageTitle, this.pageTitle, message: "Home page title not matched");
}
```

**Expected Result:-** TecRoot | Best Game & Audio Store In Sri Lanka

**Actual Result:-** TecRoot | Best Game & Audio Store In Sri Lanka

**Status:-** Passed



### Test 2:- Test Notification pop up

```
3 usages DinithKumudika
@Test(priority = 2)
public void allowNotificationPopUpTest() {
    homePage.clickAllowOnPopup();
    Assert.assertTrue(homePage.isNotificationPopupDisplayed(), message: "Notification dialog did not close.");
}

no usages DinithKumudika
@Test(priority = 2)
public void cancelNotificationPopUpTest() {
    homePage.clickCancelOnPopup();
    Assert.assertTrue(homePage.isNotificationPopupDisplayed(), message: "Notification dialog did not close.");
}
```

**Scenario-:** When the user first visits the SUT, the website asks the user whether they would like to receive notifications from the SUT. Web elements in the SUT remain inaccessible until users either accept or cancel the notification.

**Expected Result 1-:** When user clicks on `Allow` option, notification pop up disappears, Home page becomes accessible for interactions.

**Expected Result 2-:** When user clicks on `Cancel` option, notification pop up disappears, Home page becomes accessible for interactions.

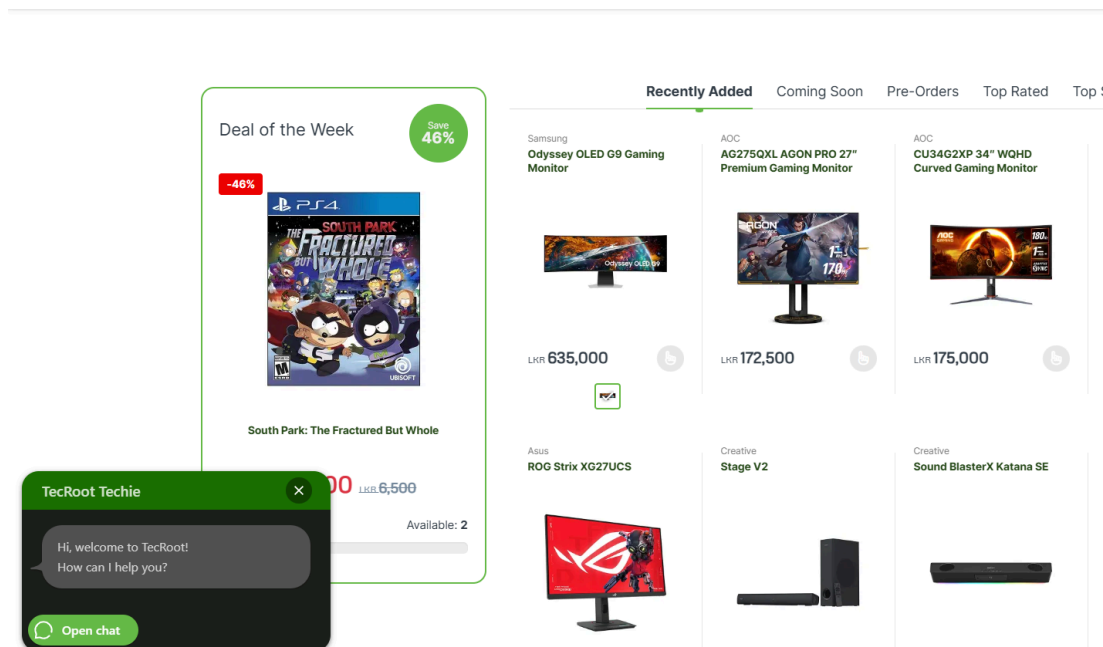
**Actual Result-:** When user clicks on either `Allow` option or `cancel` option, notification pop up disappears, Home page becomes accessible for interaction

**Limitations-:** when the user clicks on `allow`, the web browser asks permission to send notifications for the SUT, but selenium does not facilitate a way to test that.

**Status-:** Passed



### Test 3-: Redirect to whatsapp chat



```

@Test(priority = 2, dependsOnMethods = {"allowNotificationPopUpTest"})
public void openChatWindowAndRedirectToWhatsappTest() {
    homePage.clickChatButton();
    Assert.assertTrue(homePage.isChatWindowOpen(), message: "Chat window did not open.");

    homePage.clickInnerChatButton();

    // Switch to the new tab that opens (WhatsApp tab)
    ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());
    driver.switchTo().window(tabs.get(1));

    String currentUrl = driver.getCurrentUrl();
    Assert.assertTrue(currentUrl.contains("https://web.whatsapp.com"));

    driver.close();
    driver.switchTo().window(tabs.get(0));
}

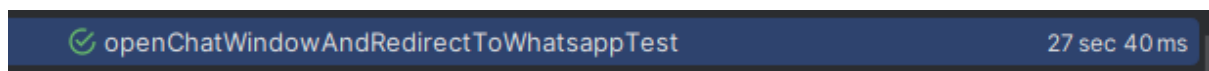
```

**Scenario-:** Home page of SUT has a chat button that pops up when a user clicks on it and redirects to the merchants whatsapp chat upon clicking on 'Open Chat' button.

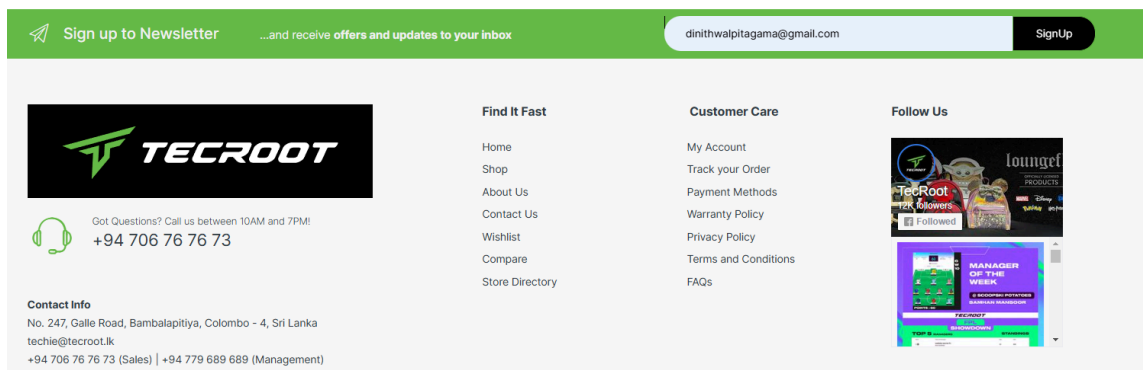
**Expected Result-:** click on 'Open Chat' button opens the web.whatsapp.com in new window leading to merchant's whatsapp chat

**Actual Result-:** click on 'Open Chat' button opens the web.whatsapp.com in new window leading to merchant's whatsapp chat

**Status-:** Passed



## Test 4-: Sign up to merchant's newsletter



TecRoot: Subscription Confirmed 

TecRoot tech@tecroot.uk  mandirapp.com  
to me

Wed, 6 Nov, 22:10 (23 hours ago)   

## TecRoot

Your subscription to our list has been confirmed.  
For your records, here is a copy of the information you submitted to us...

Email Address: [dinithwalitajana@gmail.com](mailto:dinithwalitajana@gmail.com)  
First Name:   
Last Name:   
Birthday:

If at any time you wish to stop receiving our emails, you can:

[unsubscribe here](#)

You may also contact us at:  
[tech@tecroot.uk](mailto:tech@tecroot.uk)



```
@Test(priority = 3, dependsOnMethods = {"allowNotificationPopUpTest"})
@Parameters("email")
public void signUpToNewsletterTest(String email) throws InterruptedException {
    homePage.enterEmailForNewsletter(email);
    homePage.clickSignUpNewsletterButton();

    // Switch to the new tab that opens for confirmation
    ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());
    driver.switchTo().window(tabs.get(1));

    String currentUrl = driver.getCurrentUrl();
    String expectedUrl = homePage.getBaseUrl() + "subscription-confirmation/";
    Assert.assertTrue(currentUrl.contains(expectedUrl), message: "The subscription confirmation page did not load as expected.");

    driver.close();
    driver.switchTo().window(tabs.get(0));

    Thread.sleep(2000);

    boolean emailReceived = EmailVerification.checkForConfirmationEmail(email, properties.getProperty("gmailAppPassword"), subject: "TecRoot: Subscription Confirmed");
    Assert.assertTrue(emailReceived, message: "Subscription email wasn't sent to the email: " + email);
}
```

**Scenario:-** A user can sign up to the merchant's newsletter to receive product updates and other details through email. When a user sign up to the newsletter by providing their email, an email from the tecroot will be sent to user's inbox with the subject `TecRoot: Subscription Confirmed`

**Expected Result:-** user receive a email to the email they provided with the subject of `TecRoot: Subscription Confirmed`

**Actual Result:-** user receive a email to the email they provided with the subject of `TecRoot: Subscription Confirmed`

**Status:-** Failed

**Limitations:-** Test failed due to the email client not being able to read emails from the inbox of the given email. Using a dedicated email testing service might resolve this error.

✖ Chrome Tests	1 sec 89 ms
▼ ✖ HomePageTest	1 sec 89 ms
✖ signUpToNewsletterTest[kumudikadinith@gmail.com]	665 ms

## Test 5:- Navigate to account the page

```
@Test(priority = 3, dependsOnMethods = {"allowNotificationPopUpTest", "signUpToNewsletterTest"})
public void goToAccountPageTest() throws InterruptedException {
    AccountPage accountPage = homePage.clickMyAccountLink();
    String accountPageTitle = accountPage.verifyAccountPageTitle();
    Assert.assertEquals(accountPageTitle, accountPage.getPageTitle(), message: "Account page title not matched");
}
```

**Scenario-:** When the user clicks on the `My Account` button, the user is directed to my account page which contains the Login and Register for the user.

**Expected Result-:** Upon clicking on `My Account` user is directed to the account page with endpoint <https://tecroot.lk/my-account/>.

**Actual Result-:** Upon clicking on `My Account` the user is directed to the account page with endpoint <https://tecroot.lk/my-account/>.

**Status-:** Passed

## Account Page Tests

Home > My Account

Login

Welcome back! Sign in to your account.

Username or email address \*

dinith123

Password \*

.....

☐ Remember me

Log in

Lost your password?

or

Register

Create new account today to reap the benefits of a personalized shopping experience.

Username \*

Email address \*

Password \*

☒ Subscribe to our newsletter

Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our privacy policy.

Register



## Test 1:- Verify account page title

```
1 usage  👤 Dinith Kumudika
@Test(priority=1)
public void verifyAccountPageTitleTest(){
    String accountPageTitle = accountPage.verifyAccountPageTitle();
    System.out.println(accountPageTitle);
//    Assert.assertEquals(homePageTitle, this.pageTitle,"Home page title not matched");
}
```

**Expected Result:-** My Account | TecRoot

**Actual Result:-** My Account | TecRoot

**Status:-** Passed

✔ AccountPageTest	1 min 23 sec
✔ verifyAccountPageTitleTest	22 ms

## Test 02:- User Login

### My Account

Hello **dinith123** (not **dinith123**? Log out)

From your account dashboard you can view your recent orders, manage your shipping and billing addresses, and edit your password and account details.

**ERROR:** The username or password you entered is incorrect. Lost your password?

#### Login

Welcome back! Sign in to your account.

Username or email address \*

Password \*

☐ Remember me

[Lost your password?](#)

or

#### Register

Create new account today to reap the benefits of a personalized shopping experience.

Username \*

Email address \*

Password \*

☒ Subscribe to our newsletter

```

no usages  Dinith Kumudika +1 *
@Test(dataProvider = "getLoginCredentials", dataProviderClass = TestDataProvider.class, priority=1, dependsOnMethods = {"verifyAccountPageTitleTest"})
public void loginTest(String testCaseId, String username, String password, String expectedResult) {
    // Implement login logic here
    System.out.println("Running test case ID: " + testCaseId);
    System.out.println("Username: " + username);
    System.out.println("Password: " + password);
    System.out.println("Expected Result: " + expectedResult);

    accountPage.setUsernameLogin(username);
    accountPage.setPasswordLogin(password);
    accountPage.clickOnLoginBtn();

    boolean usernameDisplayed = accountPage.isUsernameDisplayed(username);
    boolean errorDisplayed = accountPage.isErrorDisplayed();

    // Assert based on the expected result
    Assert.assertFalse(usernameDisplayed, message: "user login failed, invalid username or password");
    Assert.assertTrue(errorDisplayed, message: "user login successful");
}

```

**Scenario-:** after a user signup to the SUT, using their email, a username and a password they can login into the SUT using the given username or email and password. When the user successfully signed up their username is displayed on the web page alongside a option to logout

**Expected Result-:** If the username/email and password are valid, username is displayed in the same web page. If the invalid credentials are provided an error message is displayed

**Actual Result-:** when valid username/email and password are provided, username is displayed in the same web page. When the invalid login credentials are provided an error message is displayed with the text **'The username or password you entered is incorrect.'**

**Status-:** Passed

```

v  AccountPageTest 56 sec 529 ms
  verifyAccountPageTitleTest 15 ms
  loginTest[T001, johndoe, john@123, FAIL] 8 sec 884 ms
  loginTest[T002, kumudikadinith@gmail.com, john123, FAIL] (1) 6 sec 329 ms
  loginTest[T003, dinith123, Dinith@123, PASS] (2) 6 sec 760 ms
  loginTest[T004, dinith123, Dinith123, FAIL] (3) 6 sec 189 ms
  loginTest[T005, dinithwalpitagama@gmail.com, Dinith@123, PASS] (4) 6 sec 210 ms
  loginTest[T006, dinithwalpitagama@gmail.com, john123, FAIL] (5) 6 sec 10 ms

```

## Setup instructions of the Project

1. Clone the the project using the below command  
`git clone <https://github.com/DinithKumudika/SQA-POM.git>`
2. Install the project dependencies using the pom.xml
3. Set configurations of the project in config.properties file  
Ex-: base url of the SUT
4. Run testng.xml to execute the test suite

**Github Link-:** <https://github.com/DinithKumudika/SQA-POM.git>

**Presentation Video Link-:**

[https://drive.google.com/file/d/1\\_zd9BwAanaEazhKyPOQ9UUOfjljU3d1H/view?usp=sharing](https://drive.google.com/file/d/1_zd9BwAanaEazhKyPOQ9UUOfjljU3d1H/view?usp=sharing)