**Project Aim :**

Selenium is designed to automate web browsers, thus enabling software engineers to greatly speed up and automate testing. While test automation is its primary use, Selenium can also be used to automate certain repetitive actions, such as basic administration tasks.

Selenium supports Java. So, testers can leverage the active community of contributors and detailed documentation to write test cases. Programs written in Java are faster than other popular languages like Python.

**Projects:**

1. Automating Amazon
2. Automated Project Management

**Scope of Project :**

Selenium has a suite of software and each component of this suite has a different approach to support the test. Selenium is an open-source automation testing tool that is used to automate tests carried on various web- browsers. Selenium has a suite of tools namely, Selenium RC, Selenium IDE, Selenium Grid, Selenium WebDriver that caters to different organizational needs.

Selenium IDE simply has a very wide scope as of now. The software industry is its self a very dynamically self upgrading industry, which uses the latest technologies available. Automation and Manual testing has come at par with each other and testing has become dependent on both.

Tools & Technologies learned during Internship :

* JAVA
* Selenium Testing

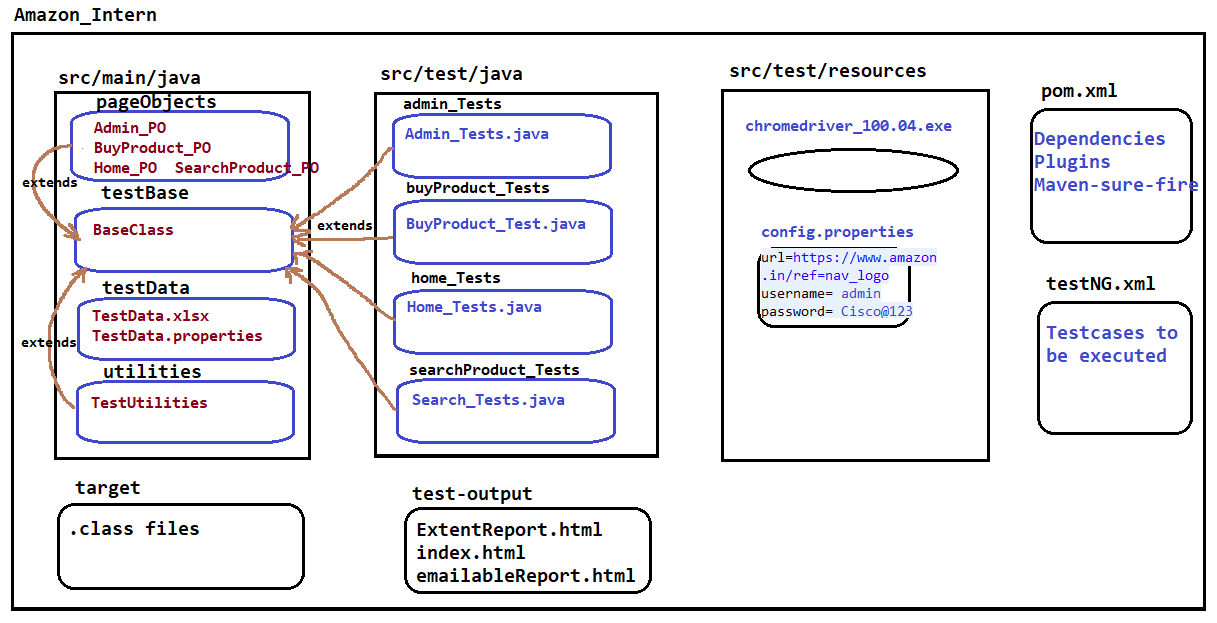
**Internship Work:**

|  |  |
| --- | --- |
| Core Java | Java & JDK installation |
|  | Eclipse installation |
|  | Java Program Execution process |
|  | Datatypes |
|  | Variables |
|  | Static and Nonstatic member |
|  | Functions |
|  | Functions Returntype |
|  | Method OverRiding |
|  | Method OverLoading |
|  | Constructor |
|  | Constructor OverLoading |
|  | Static and Nonstatic Block |
|  | If condition and Loops (While, DoWhile, For, ForEach) |
|  | Arrays |
|  | Logic Building programs |
|  | String Class |
|  | Inheritance |
|  | Super Keyword |
|  | This Keyword |
|  | This Variable |
|  | Collection Arrays |
|  | Array List |
|  | Linked List |
|  | Priority Queue |
|  | Iterator |
|  | Hash Map |
|  | Hash Table |
|  | Exception Handling |
|  | Package Access Modifier |
|  | Read and Write from Notepad |
|  | Read and Write from Properties file |

|  |  |
| --- | --- |
| Selenium | Understanding HTML Basics |
|  | Xpath |
|  | Automation Introduction |
|  | Handling TextBox |
|  | Handling CheckBox |
|  | Handling Buttons |
|  | Handling Radio Buttons |
|  | Handling Links |
|  | Getting Multiple Elements in a Browser using List |
|  | Thread.sleep |
|  | Handling implicitlyWait |
|  | Handling WebDriverWait |
|  | Handling FluentWait |
|  | Handling BrowserBack,Forward,Refresh |
|  | Take a ScreenShot |
|  | Browser Close |
|  | Browser Quit |
|  | Handling Window Scroll |
|  | Handling Multiple Browser/Windows |
|  | Handling Frames |
|  | Handling Drag and Drop |
|  | Handling Tool Tip |
|  | Handling MouseHover |
|  | Handling Alert |
|  | Toggle Button |
|  | Handling DropDown |
|  | Handling Notifications |

|  |  |
| --- | --- |
| Framework | TestNG |
|  | Maven |
|  | Extent Reports |
|  | FrameWork with Real time Application |
|  | GitHub |
|  | Jenkins |

**FrameWork Structure:**

****

**Automation:**

Simulation of any Human work by a System or a Tool is known as Automation.

**Advantages of Automation:**

1. **Reliable-** Accuracy on actions which is performed n number of times also. Consistency is achieved.
2. **Repeatable-** We can use it for n number of times to execute. So we can also test how our application reacts when same action performed for n number of times.
3. **Reusable-** You can reuse the automated tests on different versions of an application.
4. **Faster-** Compared to Manual work it’s 10 times faster.
5. **Final Cost is reduced-** Testing can be done in a faster manner. So employees strength can be reduced

**Disadvantages of Automation:**

1. Extra skill set required.
2. Initial cost is more.
3. Human errors will lead to major consequences if it is not identified properly.

**Different Automation Tools in Market:**

1. Selenium
2. QTP- Quick Test Professional
3. WinRunner
4. SilkTest
5. Watir
6. RFT- Rational Functional Tester
7. Test Complete

**Mobile Automation Tools in Market:**

1. Robotium
2. Appium
3. Selendroid

**Different Performance Tools in Market:**

1. Jmeter
2. Load Runner
3. Silk Performer
4. Rational performance tester

**Different Test Management Tools in Market:**

1. JIRA
2. Test Director
3. Quality center
4. Test Manager
5. Bugzilla
6. Mantis

**Test Cases that can be Automated**

* Test cases that we want to execute on different browsers. Smoke, Sanity, Regression Test Cases.
* Test cases that we want to execute with Multiple set of Datas.

**Test Cases that should not be Automated**

* Dynamically changing test cases.
* Performance testing related cases.
* UI(Images/Logo) related test cases.

**Selenium**

Selenium is an Open source automated testing tool which helps you to automate UI Functional Testing.

1. Selenium IDE (Integrated Development Environment)
2. Selenium RC(Remote Control)/ Selenium 1(GRID)
3. Selenium Webdriver/ Selenium 2(GRID)

**Selenium IDE**(Integrated Development Environment)

Selenium IDE was introduced as **Selenese** and later it is converted to Selenium IDE. It is developed by a company called Thought Works.

* Selenium IDE is a Record and Playback tool.
* IDE supports only Firefox Browser and Google Chrome.
* Its an Inbuilt Addon for Firefox Browser.

**Advantages of Selenium IDE**

* Open Source.
* Simple to use.
* Supports multiple OS. Eg- Windows, Ubuntu,MAC

**DisAdvantages of Selenium IDE**

* Password Encryption cannot be done.
* Only supports Firefox and Chrome Browser.
* FrameWork cannot be designed.
* By default Looping conditions not supported.
* Can’t be used Complex Applications.
* Can’t be used for Long term projects.

**Advantages of Selenium RC**(Remote Control)**:**

* Supports Firefox, Google Chrome, IE etc..
* Can be used for Complex Application.
* Can be used for Long term projects.
* Framework can be designed.
* Password Encryption can be done.
* Different Languages are supported. (Java,Ruby,Python,Csharp,Perl,javascript,php)

**Disadvantages of Selenium RC:**

* Does not support Android applications.
* Performance is Low.

**Advantages of Selenium WebDriver:**

* Supports Firefox, Google Chrome, IE etc..
* Can be used for Complex Application.
* Can be used for Long term projects.
* Framework can be designed.
* Supports Android, IOS applications.
* Performance is Good.

**Project Management Website (ActiTime)**

Processes and activities to Identify, Define, Combine, Unify and Coordinate the various processes and project management activities within the Project Management Process Groups

Throughout the project, integration includes characteristics of unification, consolidation, communication and interrelationship.

There are **seven (7)** Project Integration Management Processes. (I am be using PMBOK's notation for simplicity purpose.)

4.1) **Project Charter:** A document that formally authorizes the project. According to PMI, if there is no charter, there is no project.

4.2) **Develop Project Management Plan:** The document that defines and coordinates all plan components for entire project into an integrated plan

4.3) **Direct and Manage:** Leading the performance of project work and implementing approved changes

4.4) **Manage Project Knowledge:** Using existing knowledge and creating new knowledge to achieve the project goals and contribute to organization learning

4.5)**Monitor and Control Project Work:** Tracking, reviewing and reporting overall project progress

4.6) **Perform Integrated Change Control:**Reviewing all change requests and approving changes and managing changes to deliverables, organizational process assets, project documents, and project management plan as well as communicating the decisions to relevant stakeholders

4.7) **Closing:**Finalizing all activities the project, phase or contract

The 3 performance domains in *PMBOK 7* are:

1. People
2. Process
3. Business Environment

Within each domain, you’ll carry out a range of tasks as a project manager that fall into your responsibilities.

The big thing to note here is that these domains apply regardless of whether you are doing iterative, predictive, or hybrid projects, or any mishmash blend of them all. You still need to engage people, create effective processes and operate within your business environment: it’s all project management at the end of the day.

Let’s look at each of those domains next.

Diagram

Description automatically generated

## Domain I: People

The People domain covers the soft(er) skills required to be a good project manager. These are:

* Conflict management
* Team leadership
* Support team performance and development
* Empower team members and stakeholders so they can contribute
* Ensure people are adequately trained to participate in the project
* Build a team that can deliver on the needs of the project
* Remove roadblocks for the team
* Negotiation
* Collaboration and engagement (also across virtual teams)
* Build a shared understanding
* Create team ground rules
* Mentoring
* Apply emotional intelligence

When you look at the list, it looks like a lot, but this domain shouldn’t worry you. It’s all the stuff you know to do anyway, because it’s part of what it means to operate in a leadership role in a modern organization, whether you are working on iterative, predictive or hybrid projects.

It’s worth noting that the APM Body of Knowledge also covers interpersonal and softer skills, and people like me have been writing books about how important the people stuff is to project management for years.

## Domain II: Process

The Process domain covers the ‘technical’ skills of doing the work of being a project manager. These skills are:

* Deliver business value through smart project execution
* Communications
* Risk management
* Stakeholder analysis
* Budget and resource management
* Schedule management
* Quality management
* Scope management
* Project integration (readers may well have noticed that this list is starting to look a lot like the old list of Knowledge Areas from PMBOK 6)
* Project change management
* Procurement management
* Configuration management (the management of assets and [artifacts](https://rebelsguidetopm.com/9-types-of-artifacts-in-project-management/))
* Choosing the right methodology/approach
* Governance
* Issue management
* Knowledge transfer and project closure

If you were familiar with the PMBOK Sixth Edition, a lot of the themes and sections here will seem at home. Many of the same topics from the old Knowledge Area segments are called out, and that’s to be expected: we need to be able to do them, and do them well.

This domain represents the technical expertise of a project manager and the expert knowledge that you bring to a team about how to use project processes to get work done.

**Note for PMP exam takers:** 50% of your exam questions will be from this domain.

So far, we’ve seen an almost 50/50 split between technical and interpersonal skills, but there is a third domain, and that’s the business environment.

## Domain III: Business Environment

The Business Environment domain covers how to operate within the context of your organization. You’ll see what I mean when you read the list of skills that are covered by this domain:

* Compliance i.e. making sure the project meetings regulatory and compliance requirements
* Benefits management (although you might not take responsibility for tracking benefits long term, the project should set up a way to track benefits)
* Review external influences on project scope i.e. being aware of what is happening in the wider business environment so you can adjust the project accordingly. In the past, project sponsors have taken this responsibility, so I see the role of the PM as to make sure sponsors are passing on information to us where we don’t have access to it.
* Support organizational change.

This is only a short list but it’s very powerful. It’s everything to do with making a mark in the organization for the right reasons. It’s applying your business acumen. It’s realizing that there’s more to the organization than just your project and acting accordingly.

I think this domain is my favorite because I know from first-hand experience how much of a career difference it can make. It’s the difference between doing a reasonable job and being seen as a strategic influencer at work.

**Note for PMP exam takers:**8% of your exam questions will be from this domain. Yes, that’s right! Only 8%! Don’t lose sleep over it.

The 3 domains make up part of the PMBOK® Guide – Seventh Edition and the whole of the [Exam Content Outline](https://www.pmi.org/-/media/pmi/documents/public/pdf/certifications/pmp-examination-content-outline.pdf) for the PMP exam.

As a project manager, you are expected to do a lot and be a lot. You can see that in this structure from PMI, in the competency framework from APM and in other career frameworks for project managers.

Yes, it’s a lot.

But it’s also a great role, with great salaries, huge variety and the ability to make a tangible difference to the world.

All senior management roles require a similar blend of interpersonal skills and technical expertise, and we are no different. It’s great to see this acknowledged in such a clear way.

Diagram, schematic

Description automatically generated

* Database: A database is a collection of tables, with related data.
* Table: A table is a matrix with data. A table in a database looks like a simple spadsheet.
* Column: One column (data element) contains data of one and the same kind, for example the column postcode.
* Row: A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* Redundancy: Storing data twice, redundantly to make the system faster.
* Primary Key: A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
* Foreign Key: A foreign key is the linking pin between two tables.
* Compound Key: A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* Index: An index in a database resembles an index at the back of a book.
* Referential Integrity: Referential Integrity makes sure that a foreign key value always points to an existing row.

SOFTWARE VERIFICATION AND VALIDATION

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

* Validation : Are we building the right product?
* Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

* Fault – wrong or missing function in the code.
* Failure – the manifestation of a fault during execution.
* Malfunction – according to its specification the system does not meet its specified functionality

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

* M&S Verification is the process of determining that a ⦁ computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
* M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

Test Cases

* A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.
* Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.
* Levels
* **1 ) Unit testing :**White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
* **2 ) Integration testing :**White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
* **3 ) Regression testing :**White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

**Test Case Design Techniques**

* **Error Guessing.**
* **Equivalence Partitioning.**
* **Boundary Value Analysis.**

**Error Guessing:**

Error guessing has no explicit rules for testing; test cases can be designed depending on the situation, either drawing from functional documents or when an unexpected/undocumented error is found while testing operations.

In error guessing testers can think of situations where software will fail.

**For example:**

Division by zero

Pressing submit button on form without filling any entries.

Entering wrong data in the fields and checking software behavior.

**Inputs for Error Guessing on a Text Box:**

**1)** 0 **2)** 100Rs

**3)** 100.00 **4)** Blank

**5)** Special Char(100$ 100- 100!) **6)** Entering Characters (Hundred)

**7)** 90+10 **8)** 1,000

**Equivalence Partitioning:**

Equivalence partitioning (also called Equivalence Class Partitioning or ECP) is a software testing technique that divides the input data of a software into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once.

It is divided into two parts. They are.

* PressMan Concept (Roger Pressman)
* Practice Method.

**PressMan Concept:**

**1-** If the I/P is in range of values then design the TestCases for One valid value and Two Invalid values.

**Eg: Diagram, schematic

Description automatically generated**

**2-** If the I/P is in set of values then design the TestCases for One valid value and Two Invalid values.

**Eg: Online Shopping appln:**

**3-** I**Diagram

Description automatically generated**f the I/p is Boolean then design the Test Cases for both TRUE and FALSE values.

**Eg:** Checkboxes and Radio buttons are best example for boolean Values

Text

Description automatically generated

**Practice Method:**

Divide the range of values into Equivalent parts and test for all the values. Meanwhile ensure that you are checking for atleast 2 Invalid values.

Text

Description automatically generatedDiagram, schematic

Description automatically generated

If developers have coded as mentioned below. Practice method is the best way.

**Text

Description automatically generated**

**Boundary Value Analysis:**

If the I/P is in range of values b/w **A to B,** then  design the Testcases for

( A, A+1, A-1,    B, B+1, B-1 )

Diagram

Description automatically generated

**NOTE:**

* **Equivalence Class Partitioning**

**A picture containing diagram

Description automatically generated**

If there is a set of values and Boolean go for Equivalence partitioning and don’t go for BVA.

**Test Plan**

Test Plan is a document which derives all Testing activity. This is where all Future activities will be planned.

Generally it is prepared by Test Mgr or Test Lead.

**Sections on Test Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** |  | **Risks** |  | **Deliverables** |
| **Scope** |  | **Mitigation Plan/Contingency Plan/BackUp Plan** |  | **Defect Tracking** |
| **Approach** |  | **Test Automation** |  | **Schedule** |
| **Testing Methodologies** |  | **Roles and Responsibilities** |  | **Test Environment** |
| **Assumptions** |  | **Entry and Exit Criteria** |  | **Templates** |

**1. Objective:** It gives the aim of Test Plan.

**2. Scope:** It contains all the features to be tested & features not to be tested.

Eg of Features to be tested:

* Compose
* Inbox
* Sent Items
* Drafts.
* etc

Eg of Features not to be tested:

* Internal Ads.
* Help
* etc

Why we don’t need to Test Features

* Lack of time.
* Not a Business Critical.
* If it is a 3rd party component. We don’t do Functional Testing, But we do Integration Testing.
* We are not going to do Compatibility testing for the Features exposed internally. ie it is tested only for Base platform. Only features used by End Users will be done Compatibility Testing.

**3. Approach:**

Tells the way we go about testing the product. There are two types of Approach. They are

1. By Writing High Level scenarios.
2. By Writing Flow graphs.

High Level scenarios are written only for Critical features.

**4. Testing Methodologies:**

Types of testing conducted on product with Description for conducting it and an Eg.

* Smoke Testing.
* Functional Testing.
* Integration Testing.
* System Testing.
* Compatibility Testing.
* Performance Testing.
* Usability Testing.
* Adhoc Testing.
* Globalization Testing.
* Reliability Testing.
* Regression Testing.
* Recovery Testing.
* Security Testing.( If payments are there we should do Security Testing.)

**5. Assumptions:**

When we are writing Test Plan, certain assumptions are made like Technology, Resources etc.

**6. Risks:**

If Assumption is wrong. Risk is more. So we need to mention what type of Risk may come.

**For Eg:**

* Hardware/Software Problem.
* People quitting the Organization.
* Changes in Requirement.

**7. Mitigation Plan/Contingency Plan/BackUp Plan:**

Events that may or may not occur. If its occured what is the alternate solution to be handled.

**For Eg:**

* Needed Software to be purchased and installed prior time needed.
* Hardware/Software Problem should be handled by Admin team in a short span of time. Extra system will be there as a Backup.

**8. Test Automation:**

* Which tool we are planning to Use.
* Features to be automated and not to be automated.
* Automation framework to be used.

**9. Roles and Responsibilities:**

Each QA person involved in the Project will be termed with his Roles and Responsibility.

**For Eg:**

**9.1: Test Manager: <**Name**>**

* Write or Review Test Plan.
* Interact with Customer & Development team and Testing Team.
* Effort Estimation and Resourcing.
* Handle issues and Escalations.
* Sign off Release Note.

**9.2. Test Lead: <**Name**>**

* Write or Review Test Plan.
* Allocate Work to Test Engnnr.
* Ensure that the Test Enggrs are completing the Work before scheduled time.
* Consolidate Report sent by Test Engrr & communicate Dev team and Mgr.

**9.3. Test Engineer: <**Name**>**

* Review Test Plan.
* Write Test Cases and do a survey of the Business flow.
* Review Test Cases.
* Execute test Cases
* Write Traceability Matrix.
* Prepare Defect Report & communicate it to Dev Team.

**9.4 Test Engineer: <**Name**>**

* SetUp Test Environment & install the product.
* Identify Test Cases to be automated.
* Automate all the identified Test Cases using QTP/Selenium.
* Execute and maintain Automation Script.

**10. Entry and Exit Criteria:**

What are the criteria we should have met before starting the activity.

**For Eg:**

**Entry Criteria for Functional Testing**

* WBT should be Over.
* Test Cases should be ready.
* Application should be installed with proper Test Enviroment.
* Test Data should be ready.
* Resources should be ready.

**Exit Criteria for Functional Testing**

* % of Test Cases executed.
* Based on Defect Survey.
* Based on Pass %.

**11. Deliverables:**

**11.1-** Test Plan

**11.2-** Test Cases

**11.3-** Traceability Matrix

**11.4-** Defect Report

**11.5-** Test Execution Report

**11.6-** Graphs and Matrices

**11.7-** Release Notes

**12. Defect Tracking:**

**12.1-** Procedure to Track the Defects.

**12.2-** Severity

12.2.1- Blocker (or) Show Stopper.

12.2.2- Critical

12.2.2- Major

12.2.2- Minor

**12.3-** Priority

12.3.1- High

12.3.2- Medium

12.3.2- Low

**12.4-** Tool to be used for Defect Tracking

**13. Schedule:**

* Time to start System Study.
* Time to start Writing Test Cases.
* Time to start Executing Test Cases.
* Time to start Product Release.

**14. Test Environment:**

* Speaks about the Hardware required and software to be installed in system to test the product.

**14.1- Server**

**14.1.1- Hardware**

**14.1.2- Software**

OS- Ubuntu

WebServer-

Appln Server-

Database- MySql

**14.2- Client**

**14.2.1- Hardware**

* Windows system with 4gb ram
* 2.8 ghz Processor.
* 500 Gb Harddisk.

**14.2.1- Software**

* **OS-** XP, Win 7, Win 8, Win 8.1
* **Browser-** Mozilla, Google Chrome

**15. Templates:**

**15.1-** Test Case Template

15.2- Test Case Review Template

15.3- Traceability Matrix Template

15.4- Release Note

15.5- Defect Report

15.6- Test Execution Report

**CRS**

<http://www.anu.edu.au/sas/student_files/SampleScreens.pdf>

SRS

**Test Case Report**

|  |  |  |  |
| --- | --- | --- | --- |
| **GENERAL INFORMATION** |  |  |  |
| **Test Stage:** | Unit                  Functionality      Integration      System    Interface  Performance    Regression        Acceptance    Pilot  Specify the testing stage for this test case. |  |  |
| **Test Date:** | mm/dd/yy | **System Date, if applicable:** | mm/dd/yy |
| **Tester:** | Specify the name(s) of who is testing this case scenario. | **Test Case Number:** | Specify a unique test number assigned to the test case. |
| **Test Case Description:** | Provide a brief description of what functionality the case will test. |  |  |
| **Results:** | Pass Fail | **Incident Number, if applicable:** | Specify the unique identifier assigned to the incident. |
| **INTRODUCTION** |  |  |  |
| **Requirement(s) to be tested:** | Identify the requirements to be tested and include the requirement number and description from the Requirements Traceability Matrix. |  |  |
| **Roles and Responsibilities:** | Describe each project team member and stakeholder involved in the test, and identify their associated responsibility for ensuring the test is executed appropriately. |  |  |
| **Set Up Procedures:** | Describe the sequence of actions necessary to prepare for execution of the test. |  |  |
| **Stop Procedures:** | Describe the sequence of actions necessary to terminate the test. |  |  |
| **ENVIRONMENTAL NEEDS** |  |  |  |
| **Hardware:** | Identify the qualities and configurations of the hardware required to execute the test case. |  |  |
| **Software:** | Identify system and application software required to execute the test case. Specify any software that the test case will interact with. |  |  |
| **Procedural Requirements:** | Describe any constraints on the test procedures necessary to execute the test case. |  |  |
| **TEST** |  |  |  |
| **Test Items and Features:** | Identify and describe the items and features that will be exercised by the test case. Group the test cases into logically related scenarios that test related items and features. For each item or feature, a reference to its associated requirement source should be included. |  |  |
| **Input Specifications:** | Define each input required to execute the test case, and reference any required relationships between inputs. |  |  |
| **Procedural Steps:** | Describe the sequences of actions necessary to prepare and execute the test case. Provide detailed test procedures for each test case; explain precisely how each test case will be executed. |  |  |
| **Expected Results of Case:** | Describe the outcome anticipated from the test case. Specify the criteria to be used to determine whether the item has passed or failed. |  |  |
| **ACTUAL RESULTS** |  |  |  |
| **Output Specifications:** | Define all of the outputs and features required of the test case and provide expected values. While executing the test, record and describe the visually observable outputs as they occur. Produce tangible evidence of the output such as a screen print. At the conclusion, describe the actual outcome. Indicate whether the test passed or failed, and identify any discrepancies between the expected results and the actual results. |  |  |

JAVA Execution

Diagram

Description automatically generated

Procedure for Java installation

Steps to check if you have Java installed already

1. Goto--> My Computer--> Program Files--> Check the JAVA folder is present.
2. Open your cmd prompt and type java -version.  If it's installed you will get the version number. Orelse you will get an output as “java is not recognized”.

A screenshot of a computer

Description automatically generated with medium confidence

If JAVA is not installed in your system follow the below steps.

Procedure for JDK installation

Goto the link below <https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html>

Click on the download link under Java SE Development Kit 8u202 table

If your Machine is 64bit. You have to download the JDK.exe file for Windows x64.

If your Machine is 32bit. You have to download the JDK.exe file for Windows x86.

A screenshot of a computer

Description automatically generated with medium confidence

Now click on the downloaded file and click on RUN. and go with the installation process and install it.

After the installation is complete. Goto your c:/-->Program Files--> JAVA--> JDK-->bin folder and copy the location

For Eg:

C:\Program Files\Java\jdk1.7.0\_03\bin

Setting the PATH Variable

Right Click on MyComputer and click on "Properties".

Now click on "Advanced System Settings" and goto the “Advanced” tab and click on the "Environment Variables" button.

Graphical user interface, text, application, email

Description automatically generated

Now select "Path" from the System Variables section and double click on the PATH.

Graphical user interface, application

Description automatically generated

You will get a New window. In that window under "Variable Value". Goto the END of the line and check whether ";" is given, orelse give ";" and paste the Copied Path of JDK with bin folder and give a ";" and click on "OK" and again "OK"

Graphical user interface

Description automatically generated

Check whether JDK is installed in your System

Goto CMD prompt and type "java -version". It gives you the JDK Version.

Text

Description automatically generated

Check whether JDK path set in your System

Goto CMD prompt and type "javac -version". It gives you the JDK Version.

Text

Description automatically generated

Eclipse Installation

Browse Eclipse photo download in Google Search, and click on <http://www.eclipse.org/downloads/packages/release/Photon/M6> link. You will connect to the Eclipse main Website.



Under the “Package Solution” you will find a icon   which will have a heading  Click on the link “[Windows 32-bit](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-jee-kepler-SR2-win32.zip)” or “[Windows 64-bit](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-jee-kepler-SR2-win32-x86_64.zip)” according to your system configuration 32bit or 64bit. You will connect to a page for downloading the Eclipse.

Steps to Open Eclipse and Create a Project

After downloading the extract  the  Eclipse zip file. Inside the extracted folder, you will find an icon . Click on that icon your Eclipse will be opened.

When you open the Eclipse. You will get a window popup appeared on the Screen, which confirms you as where the Scripts written in your Eclipse to be saved. If you need to change the location, Click on the Browse button and change the location where you need to save the Code and click on OK button.

Graphical user interface, text, application

Description automatically generated

When you open the Eclipse for the First time. you will get a Welcome Screen. Close the Welcome Screen and create a Project.

Graphical user interface, text, application

Description automatically generated

* Click on File→ New→ Project. It will open a New window.

Graphical user interface, text, application, Word

Description automatically generated

* On the New window click on Java Project, orelse Java→ Java Project and click on  button.

Graphical user interface, text, application

Description automatically generated

* On the next window which is opened,give the Project name and click on  button.

Graphical user interface, application

Description automatically generated

* Now on you Eclipse window. You Project will be created under Package Explorer section which will have a Src folder and JRE System Library.

Graphical user interface, application

Description automatically generated

* src folder will hold all your Java files ie, the programs you write in the Eclipse window.
* JRE System Library will hold all your jar files which is related to Java usage.

Create a Package

Right click on your src folder and click on New→ Package. Which creates a folder section in your src folder, and the class files will be created on that package.

Graphical user interface, text, application, email

Description automatically generated

Give the Package Name

After clicking on the Package you will get a New window opened, which helps you give the Package name and where the Package to be created.

* Text box named Sourcefolder helps you to select under which Project the package to be created. By default it will be under the src folder of the Current Project selection.
* Give the name for the Project on the Text box named Name and click on the  button.

Graphical user interface, text, application

Description automatically generated

Create a Class

After creating the package if you open the src folder, the package will be created. Now Right click on the Package and click on New→ Class

Graphical user interface, text, application, email

Description automatically generated

Give the Class Name

After clicking on the Class you will get a New window opened, which helps you give the Class name and where the class to be created.

* Text box named Sourcefolder helps you to select under which Project the Class to be created. By default it will be under the src folder of the Current Project selection.
* Text box named Package helps you to select under which Package the Class to be created. By default it will be under the Current Package selection.
* Give the name for the Class on the Text box named Name and select the Modifier type of the Class either  from the Radio button. By default it will be in Public.
* Enable the checkbox  if you need the main() function. The JAVA Program can be executed only If we have the main() function .
* Now click on the  button. Your class will be created under the Selected Package.

Graphical user interface, application

Description automatically generated

* After creating the class you can see a Java file created under the Package section and the file will be opened in your Eclipse.

Graphical user interface, text, application

Description automatically generated

Hello World Program

Inside you main() function type System.out.println("Hello World"); This command will print you the output as Hello World in the Console window.

Graphical user interface, text, application

Description automatically generated



Boolean Data Type

The Boolean data type is used to store only two possible values: true and false. This data type is used for simple flags that track true/false conditions.

The Boolean data type specifies one bit of information, but its "size" can't be defined precisely.

Example:

Boolean one = false

Byte Data Type

The byte data type is an example of primitive data type. It isan 8-bit signed two's complement integer. Its value-range lies between -128 to 127 (inclusive). Its minimum value is -128 and maximum value is 127. Its default value is 0.

The byte data type is used to save memory in large arrays where the memory savings is most required. It saves space because a byte is 4 times smaller than an integer. It can also be used in place of "int" data type.

Example:

byte a = 10, byte b = -20

Short Data Type

The short data type is a 16-bit signed two's complement integer. Its value-range lies between -32,768 to 32,767 (inclusive). Its minimum value is -32,768 and maximum value is 32,767. Its default value is 0.

The short data type can also be used to save memory just like byte data type. A short data type is 2 times smaller than an integer.

Example:

short s = 10000, short r = -5000

Int Data Type

The int data type is a 32-bit signed two's complement integer. Its value-range lies between - 2,147,483,648 (-2^31) to 2,147,483,647 (2^31 -1) (inclusive). Its minimum value is - 2,147,483,648and maximum value is 2,147,483,647. Its default value is 0.

The int data type is generally used as a default data type for integral values unless if there is no problem about memory.

Example:

int a = 100000, int b = -200000

Long Data Type

The long data type is a 64-bit two's complement integer. Its value-range lies between -9,223,372,036,854,775,808(-2^63) to 9,223,372,036,854,775,807(2^63 -1)(inclusive). Its minimum value is - 9,223,372,036,854,775,808and maximum value is 9,223,372,036,854,775,807. Its default value is 0. The long data type is used when you need a range of values more than those provided by int.

Example:

long a = 100000L, long b = -200000L

Float Data Type

The float data type is a single-precision 32-bit IEEE 754 floating point.Its value range is unlimited. It is recommended to use a float (instead of double) if you need to save memory in large arrays of floating point numbers. The float data type should never be used for precise values, such as currency. Its default value is 0.0F.

Example:

float f1 = 234.5f

Double Data Type

The double data type is a double-precision 64-bit IEEE 754 floating point. Its value range is unlimited. The double data type is generally used for decimal values just like float. The double data type also should never be used for precise values, such as currency. Its default value is 0.0d.

Example:

double d1 = 12.3

Char Data Type

The char data type is a single 16-bit Unicode character. Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive).The char data type is used to store characters.

Example:

char letterA = 'A'

Why char uses 2 byte in java and what is \u0000 ?

It is because java uses Unicode system not ASCII code system. The \u0000 is the lowest range of Unicode system. To get detail explanation about Unicode visit next page.

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Sno** | **Arrays** | **Collection** |
| **1** | **Array Size is fixed can’t be modified** | **Size is not fixed can grow dynamically** |
| **2** | **Through index we can add values** | **No need of indexing** |
| **3** | **Only one type of data can be used at a time by default.** | **Different data types can be used** |

**List:**

1. List is a Collection of Objects.
2. List can store any kind of datatype.
3. List allows duplicate elements and Null values.
4. All the elements of the List collection are Index automatically.

**Egs of List:**

* Array List
* Vector
* Linked List

**Queue:**

1. Queue is a Collection of an Object.
2. Queue implements First In First Out.
3. Queue can store Duplicate values but not null.
4. It should be of same data type, because it is internally stored.
5. Queues are not indexed.
6. Queues are Auto Stored.

**Egs of Queue:**

* Priority Queue
* Linked List

**Note:** The Linked List class implements both List, Queue & Set. Hence Linked List class has List Feature, Queue Feature & Set Feature .

**Set:**

1. Set is a collection of Unique Objects.
2. Set can store null values but not Duplicate Values
3. Set is not an Indexed Collection.
4. Set can accept different data types.

**Egs of Set**

* Hash Set
* Linked List
* Tree Set

**Map:**

1. Map is a collection of Key-Value pair.
2. Key is a collection of Unique elements.
3. Value is a collection of Objects which can be duplicate and null.
4. Map is not an Indexed Collection.

**Egs of Map**

* Hash Map
* Linked Map
* Tree Map

**Difference b/w Array List, Linked List, Vector**

**ArrayList:**

It is implemented as a resizable array. As more elements are added to ArrayList, its size is increased dynamically. It’s elements can be accessed directly by using the get and set methods, since ArrayList is essentially an array.

**LinkedList:**

It is implemented as a double linked list. Its performance on add and remove is better than Arraylist, but worse on get and set methods.

**Vector:**

It is similar with ArrayList, but it is synchronized

**Difference b/w Collection Collections**

|  |  |  |
| --- | --- | --- |
| **Sno** | **Collection** | **Collections** |
| 1 | It is an Interface | It is an Class |
| 2 | Used to represent a group on Individual Objects as a Single entity | Used to define several Utility methods like (Sorting, Searching..) for Collection Objects |
| 3 | Java Collection simply means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet etc). | All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion etc. can be performed by Java Collections. |

**Inheritance**

* Extending the properties from Parent Class to Child class is called as Inheritance
* Inheritance can be done with a Keyword named as "extends"
* Parent class is also called as "Super Class". Child class is also called as "Sub Class"
* In Java Multi level Inheritance is supported, but directly we don't support Multiple Inheritance
* To achieve Multiple Inheritance we need **Interface**
* Method types should be same if it is Inherited (Static/Nonstatic)
* Variable types can be anything (Static/Nonstatic)
* When we create an Object for Parent class. (Only Parent class properties will be loaded)
* When we create an Object for Child class. (Both Parent and Child class properties will be loaded. If the same property Child class also has, then that will be overrided by the Child)
* When we an Object for Child class and reference is Parent Class (Only Parent class properties will be loaded. Nonstatic method alone will be overrided from the Child class if it is of same property name)

**Exception Handling:**

Diagram

Description automatically generated

**Throw and Throws:**

* Throws clause is used to declare an exception and throw keyword is used to throw an exception explicitly.
* If we see syntax wise, throw is followed by an instance variable and throws is followed by exception class names.
* Throw keyword is used inside method body to invoke an exception and throws clause is used in method declaration ( ).
* Throw keyword should be the last line in the method.
* Throw keyword is used to throw an exception explicitly. Only object of Throwable class or its subclasses can be thrown. Program execution stops on encountering

**Note:**

**try catch ==>** Checked(Compilation Time Java identifies) and Unchecked Exception(Execution Time Java identifies)

**throws ==>** Checked(Compilation Time Java identifies)

**throw ==>** Explicitly throws an Exception. (Helps to stop the program)

**throw Example**

**Eg-1**

throw new Exception("You have some exception")

throw new IOException("Connection failed!!")

**Eg-2**  
static{  
try {  
  throw new Exception("Something went wrong!!");  
} catch (Exception exp) {  
  System.out.println("Error: "+exp.getMessage());  
}  
}  
....

**Eg-3**

throw new ArithmeticException("An integer should not be divided by zero!!")  
throw new IOException("Connection failed!!")

**Throws:**

**Eg-1**

throws IOException, ArithmeticException, NullPointerException

**Eg-2**

public void sample() throws ArithmeticException{  
 //Statements  
  
.....  
  
 //if (Condition : There is an error)  
ArithmeticException exp = new ArithmeticException();  
 throw exp;  
...  
}

**Eg-3**

throws IOException, ArithmeticException, NullPointerException,   
ArrayIndexOutOfBoundsException

**Note:**

By using Throw keyword in java you cannot throw more than one exception but using throws you can declare multiple exceptions.

**Type Casting:**

**Assignment Statements**

 Two types of Assignment statements are present. They are

* Homogeneous Statement
* Heterogeneous Statement

**Homogeneous Statement**

If the reference Variable type and the value to saved in the memory location are same. Then they are called as Homogeneous Statement.

Text

Description automatically generated with medium confidence

**Eg1:**

int a = 10;

LHS = RHS

**Eg2:**

double b = 10.50;

LHS = RHS

**Heterogeneous Statement**

If the reference Variable type and the value to saved in the memory location are different. Then they are called as HeterogeneousStatement.

**Eg1:**

int a = 10.50;

LHS  RHS

**Eg2:**

double b = 10;

LHS  RHS

**Type Casting:**

       Converting a variable from one data type to another data type is called as Type Casting.

In JAVA there are two types of Casting. They are

·        Primitive Casting.

·        Derived Casting.

Diagram

Description automatically generated

**Primitive Casting:**

       Converting a Primitive type to another Primitive type can be termed as Primitive Casting.

It is classified into two types. They are

·        Auto Widening.

·        Explicit narrowing.

**Diagram

Description automatically generated**

**Auto Widening:**

       Converting a variable of lowest size data type to highest size data type is termed as Widening. Since compiler can perform Widening automatically, it is also termed as Auto Widening.

Icon

Description automatically generated

**Eg:**

int a = 5;

double b = a;

**Explicit Narrowing:**

       Converting a variable of highest size data type to lowest size data type is known as Explicit Narrowing. Since a programmer has to perform Narrowing explicitly is termed as Explicit Narrowing.

**A picture containing icon

Description automatically generated**

**Eg:**

Double i = 5.20;

int j = (int) i;

When we do Narrowing, we will have a loss of data in it.

In the above Eg when an double data type **“5.20”** is converted to an Integer data type, the value will be changed as **“5”.**

**Multi Casting:**

**A picture containing text

Description automatically generated**

**Derived Casting:**

Converting one Derived type to another Derived type is known as Derived Casting.

Derived Casting can be classified into Two types. They are

* Auto Upcasting.
* Downcasting.

**Auto Upcasting:**

Converting an Object of SubClass to any of its Superclass type can be termed as Upcasting.

       Since the Compiler can perform Upcasting automatically. It is also termed as Auto Upcasting.

       When a Superclass object is Upcasted to any of its Super class type, then the Sub class members cannot be accessible for the Upcasted Object.

Diagram

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

**Upcasting with Multi Level Inheritance:**

       If we Upcast a Class. The Upcasted object will have only the Upcasted member alone.

Diagram

Description automatically generated

**Down Casting:**

Only Upcasted members can be downcasted. Downcasted object will have both the members of Super class and Sub Class.

       Using the reference of Upcasted object only we can downcast.

Converting an Object of Super class to any of its Sub class type is termed as DownCasting.

       A Newly created object of a Super class cannot be Downcasted.

       The Compiler compiles the downcasting statement whereas the JVM throws an Exception with the name “Class Cast Exception” because the Newly created object of superclass will not have access to Subclass members.

       So only an Upcasted object can be Downcasted.

       A same Object can be Upcasted and Downcasted in every instance to increase the performance.

Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

**Configuration Steps to Work in Excel Sheet**

Browse for “apache poi Download” in Google Search and click on the URL <http://poi.apache.org/download.html>

#### Download the Apache POI

Under ”Binary Distribution” section click on the link ”[poi-bin-3.13-20150929.zip](http://www.apache.org/dyn/closer.lua/poi/release/bin/poi-bin-3.13-20150929.zip)”

which of 25MB. ie the Zip file which is of higher size.

After downloading if you extract the zip file. You will get a folder “poi-3.10-FINAL”.

**Jar files needed to add in Eclipse**

Open the extracted folder. Add these 3-Jar files to Eclipse

* poi-3.10-FINAL-20140208
* poi-ooxml-3.10-FINAL-20140208
* poi-ooxml-schemas-3.10-FINAL-20140208

Now open the folder “ooxml-lib” and add these 2 jar files.

* dom4j-1.6.1
* xmlbeans-2.3.0

**Working on Eclipse**

In your respective Project Add these 5 jar files to your Build Path. And now your code will support the Excel Files.

**Precautions to be done for Excel Work Book**

1. Format the cells to Text. So it will be considered as Active cells and gets the correct Row Count.
2. Save the Excel Sheet in .xlsx format.

**Note:**

If you are not having 2010 format . while saving the Excel file save it as filename.xlsx and with file type as “Excel WorkBook”

Graphical user interface, application, Word

Description automatically generated

**What we need for Getting the Row Count in Excel Sheet**

1. WorkBook name
2. WorkBook Location
3. Sheet Name/Sheet Index

**Eg:**

FileInputStream fis = new FileInputStream("G:\\Selenium\\tests.xlsx");

Workbook wb = WorkbookFactory.create(fis);

Sheet s = wb.getSheet(sheetName);

retVal = s.getLastRowNum();

**What we need for Reading in Excel Sheet**

1. WorkBook name
2. WorkBook Location
3. Sheet Name
4. Row no
5. Cell No/Column No

**Eg:**

FileInputStream fis = new FileInputStream("G:\\Selenium\\tests.xlsx");

Workbook wb = WorkbookFactory.create(fis);

Sheet s = wb.getSheet(sheetName);

Row r = s.getRow(rowNum);

Cell c = r.getCell(cellNum);

retVal=c.getStringCellValue();

**What we need for Writing in Excel Sheet**

1. WorkBook name
2. WorkBook Location
3. Sheet Name
4. Row no
5. Cell No/Column No
6. Value that need to Write

**Eg:**

FileInputStream fis = new FileInputStream("G:\\Selenium\\tests.xlsx");

Workbook wb = WorkbookFactory.create(fis);

Sheet s = wb.getSheet(sheetName);

Row r = s.getRow(rowNum);

Cell c = r.createCell(cellNum);

c.setCellType(c.CELL\_TYPE\_STRING);

c.setCellValue("Value to Write");

FileOutputStream fos = new FileOutputStream("G:\\Selenium\\tests.xlsx");

wb.write(fos);

fos.close();

**Xpath by Murali**

**Xpath info:**

Immediate Parent --- **[ ]**

Immediate Child --- **/**

Match any Element--- **\***

To Refer Attribute--- **@**

To Refer Text--- **text()**

For Approximate Match--- **contains()**

For Perfect Match--- **=**

For No Match--- **not(contains())**

To Select Last element--- **last()**

To Select Last Before element--- **last()-1**

To Select greater than some text value which is numeric--- **>**

To Select lesser than some text value which is numeric--- **<**

**preceding-sibling::\*-** Used to get the siblings before the respective node

Eg(//b[text()='Five']/preceding-sibling::\*) **⇒** Gets me the siblings before the b tag which has a text=’Five’

**following-sibling::\*-** Used to get the siblings after the respective node

**descendant::-** matches with any child node. For Eg-(//div[@class='search-container']/descendant::input)

**ancestor::-** matches with any Parent node.

**count(//b)-**  Gets me the count of the b tags for that Parent

position()

**position()-** Gets the element that matches the position for that tag.

For Eg- //b[position()=4] (Gets me the element that matches 4th position)

**Multiple Properties**

//input[@type='text' **and** @name='usrnm']

//input[@type='text' **or** @name='usrnm']

**Working with SVG maps**

**Xpath for SVG maps with name() convention**

//div[@id="canvas"]/\*[name()="svg"]/\*[name()="circle"][@cx=’272.34’]

**Xpath for SVG maps with name() convention and Location**

//div[@id="canvas"]/\*[name()="svg"]/\*[name()="circle"][1]

**Link for Eg of SVG maps**

http://stackoverflow.com/questions/14592213/selenium-webdriver-clicking-on-elements-within-an-svg-using-xpath

**Types of Xpath**

**Absolute Path-** Expression written to find the WebElement directly.

**Relative Path-**  Expression written to find the WebElement by help of another WebElement..

**Absolute Xpath Using Id Attribute**

//button[@id='u\_0\_i']

**Absolute Xpath  Using Multiple Properties (Both should Match)**

//button[@id='u\_0\_i' **and** @type='submit']

**Absolute Xpath  Using Multiple Properties (AnyOne should Match)**

//button[@id='u\_0\_i' **or** @type='submit']

**Absolute Xpath  Using Text Function**

//button[text()='Sign Up']

**Absolute Xpath  Using contains Function**

//button[contains(text(),'Sign')]

**Xpath usage for HTML5**

http://selenium2advanced.blogspot.in/2012/01/selenium2-and-html5.html

Graphical user interface, text, application, email

Description automatically generated

**Link for the Below Examples:**

<http://www.flipkart.com/ebooks/science-technology/pr?sid=ixq,y63&otracker=hp_nmenu_sub_books-media_0_Science%20%26%20Technology>

**Note:**

Change the Book name “A Brief History” to some other which is present in that page.

**Absolute Xpath in flipkart parent of BUYNOW**

//div[div[div[div[a[text()='Buy Now']]]]]

**Relative Path for BUY NOW button**

//div[div[div[div[a[contains(text(),'A Brief History')]]]]]/div/div/div/a[text()='Buy Now']

**Relative Path using contains for RS.149**

//div[div[div[div[a[contains(text(),'A Brief History')]]]]]/div[2]/div[1]/div/div[contains(text(),'Rs. 149')]

**Relative Path using Location for RS.149**

//div[div[div[div[a[contains(text(),'A Brief History')]]]]]/div[2]/div[1]/div[1]

**Relative Path using Location for BUY NOW**

//div[div[div[div[a[text()='A Brief History Of Time (From Big Bang To Black Holes)']]]]]/div[2]/div[1]/div[3]/a[text()='Buy Now']

**Relative Path for BUY NOW**

//div[div[div[div[a[text()='A Brief History Of Time (From Big Bang To Black Holes)']]]]]/div/div/div/a[text()='Buy Now']

http://www.flipkart.com/ebooks/science-technology/pr?sid=ixq,y63&otracker=hp\_nmenu\_sub\_books-media\_0\_Science%20%26%20Technology

//div[div[2][div[1][a[contains(text(),'Adidas Lite Pacer M')]]]]/div/div[2]/div[3]/div/div/span[text()='Rs. 4999']

**Detail Xpath info for flipkart.com withScreen Shot**

**Link for the below Example**

<http://www.flipkart.com/ebooks/science-technology/pr?sid=ixq,y63&otracker=hp_nmenu_sub_books-media_0_Science%20%26%20Technology>

Graphical user interface, text

Description automatically generated with medium confidence

Timeline

Description automatically generated with medium confidence

**Tree Structure for the FlipKart for the Evolution Book**

Timeline

Description automatically generated

**Find the Relative Xpath for the BuyNow button of Evolution Book**

* **Get the xpath for the Evolution Book**

//a[text()='Evolution (The Triumph of an Idea)']



* **Get the Parent for the Evolution Book**

//div[a[text()='Evolution (The Triumph of an Idea)']]



* **Get the Common Parent for the Evolution Book and Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]

Graphical user interface, text, application, email

Description automatically generated

* **Now travel through the Child of Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div

Graphical user interface, text, application, chat or text message

Description automatically generated

* **Come to the Next Child**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div/div

Graphical user interface, text, application

Description automatically generated

* **Come to the Next Child**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div/div/div

Graphical user interface, text, application

Description automatically generated

* **Come to the Next Child of Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div/div/div/a

Graphical user interface, application

Description automatically generated

* **Now select the Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div/div/div/a[text()='Buy Now']



**Find the Relative Xpath for the BuyNow button of Evolution Book through Location**

* **Get the xpath for the Evolution Book**

//a[text()='Evolution (The Triumph of an Idea)']



* **Get the Parent for the Evolution Book**

//div[a[text()='Evolution (The Triumph of an Idea)']]



* **Get the Common Parent for the Evolution Book and Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]

Graphical user interface, text, application, email

Description automatically generated

* **Now travel through the Child of Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div[2]

Graphical user interface, text, application

Description automatically generated

* **Come to the Next Child**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div[2]/div[1]

Graphical user interface, application

Description automatically generated

* **Come to the Next Child of Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div[2]/div[1]/div[3]



* **Now select the Buy Now button**

//div[div[div[div[a[text()='Evolution (The Triumph of an Idea)']]]]]/div[2]/div[1]/div[3]/a[1]



**XPATH tutorials**

[**http://www.toolsqa.com/selenium-webdriver/choosing-effective-xpath/**](http://www.toolsqa.com/selenium-webdriver/choosing-effective-xpath/)

[**http://seleniumeasy.com/selenium-tutorials/xpath-tutorial-for-selenium**](http://seleniumeasy.com/selenium-tutorials/xpath-tutorial-for-selenium)

[**http://software-testing-tutorials-automation.blogspot.in/2013/06/xpath-tutorials-identifying-xpath-for.html**](http://software-testing-tutorials-automation.blogspot.in/2013/06/xpath-tutorials-identifying-xpath-for.html)

**Types of Exceptions in Selenium**  
  
   
The complete list of exceptions in Selenium  
  
1. **ConnectionClosedException:** This exception takes place when there is a disconnection in the driver.  
  
  
2. **ElementClickInterceptedException:** The command could not be completed as the element receiving the events is concealing the element which was requested clicked.  
  
  
3. **ElementNotInteractableException:** This Selenium exception is thrown when an element is presented in the DOM but it is impossible to interact with such element.  
  
  
4. **ElementNotSelectableException:** This Selenium exception is thrown when an element is presented in the DOM but you can be able to select. Hence, it is impossible to interact with.  
  
  
5. **ElementNotVisibleException:** This type of Selenium exception takes place when existing element in DOM has a feature set as hidden. In this situation, elements are there, but you can not see and interact with the WebDriver.  
  
  
6.**ErrorHandler.UnknownServerException:** Exception is used as a placeholder if the server returns an error without a stack trace.  
  
  
7. **ErrorInResponseException:** This exception is thrown when a fault has occurred on the server side. You can see it happens when interacting with the Firefox extension or the remote driver server.  
  
  
8. **ImeActivationFailedException:** This exception occurs when IME engine activation has failed.  
  
  
9. **ImeNotAvailableException:** This exception takes place when IME support is unavailable.  
  
  
10. **InsecureCertificateException:** Navigation made the user agent to hit a certificate warning, which is caused by an invalid or expired TLS certificate.  
  
  
11. **InvalidArgumentException:** This Selenium exception is thrown if an argument does not belong to the expected type.  
  
  
12. **InvalidCookieDomainException:** This happens when you try to add a cookie under a different domain rather than the current URL.  
  
  
13. **InvalidCoordinatesException:** This happens if the coordinates offered to an interacting operation are not valid.  
  
  
14. **InvalidElementStateException:** This Selenium exception occurs if a command cannot be finished as the element is invalid.  
  
  
15. **InvalidSessionIdException:** Takes place when the given session ID is not included in the list of active sessions, which means the session does not exist or is inactive either.  
  
  
16. **InvalidSwitchToTargetException:** Happens if frame or window target to be switched does not exist.  
  
  
17. **JavascriptException:** This problem happens when executing JavaScript supplied by the user.  
  
  
18. **JsonException:** Happens when you afford to get the session capabilities where the session is not created.  
  
  
19.**MoveTargetOutOfBoundsException:** Takes place if the target provided to the ActionChains move() methodology is not valid. For example: out of document.  
  
  
20. **NoAlertPresentException:** Happens when you switch to no presented alert.  
  
  
21. **NoSuchAttributeException:** Occurs when the attribute of element could not be found.  
  
  
22. **NoSuchContextException:** Happens in mobile device testing and is thrown by ContextAware.  
  
  
23. **NoSuchCookieException:** This exception is thrown if there is no cookie matching with the given path name found amongst the associated cookies of the current browsing context’s active document.  
  
  
24. **NoSuchElementException:** Happens if an element could not be found.  
  
25. **NoSuchFrameException:** Takes place if frame target to be switch does not exist.  
  
  
26. **NoSuchWindowException:** Occurs if window target to be switch does not exist.  
  
  
27. **NotFoundException:** This exception is subclass of WebDriverException. It happens when an element on the DOM does not exist.  
  
  
28. **RemoteDriverServerException:** This Selenium exception is thrown when server do not respond due to the problem that the capabilities described are not proper.  
  
  
29. **ScreenshotException:** It is impossible to capture a screen.  
  
  
30. **ScriptTimeoutException:** Thrown when executeAsyncScript takes more time than the given time limit to return the value.  
  
  
31. **SessionNotCreatedException:** A new session could not be successfully created.  
  
  
32. **SessionNotFoundException:** The WebDriver is performing the action right after you quit the browser.  
  
   
  
33. **StaleElementReferenceException:** This Selenium exception happens if the web element is detached from the current DOM.  
  
  
34. **TimeoutException:** Thrown when there is not enough time for a command to be completed.  
  
  
35. **UnableToCreateProfileException:** You can open a browser with certain options using profiles, but sometimes a new version of Selenium driverserver or browser may not support the profiles.  
  
  
36. **UnableToSetCookieException:** Occurs if a driver is unable to set a cookie.  
  
  
37. **UnexpectedAlertPresentException:** This Selenium exception happens when there is the appearance of an unexpected alert.  
  
  
38. **UnexpectedTagNameException:** Happens if a support class did not get a web element as expected.  
  
  
39. **UnhandledAlertException:** It happens when there is an alert, but WebDriver is unable to perform Alert operation.  
  
  
40. **UnknownMethodException:** Thrown when the requested command matching with a known URL but not matching with a methodology for that URL.  
  
  
41. **UnreachableBrowserException:** This Selenium exception happens if the browser is unable to be opened or have crashed because of some reasons.  
  
  
42. **UnsupportedCommandException:** Occurs when remote WebDriver does not send valid command as expected.  
  
  
43. **WebDriverException:** This takes place when the WebDriver is performing the action right after you close the browser.

**TESTNG:**

**Get TestNG-**

Goto <http://testng.org/doc/download.html> and copy the link <http://beust.com/eclipse>/ which is on “For Eclipse 3.4 and above” version.

**Install TestNG in Eclipse**

Click on “HELP” in Eclipse and click on “Install New Software”. You will get a PopUp window. Copy the url <http://beust.com/eclipse> in “Work with” Text box.

When you paste the URL. You will get TestNG appeared in the below table box. Select the Checkbox and click on “Next” button and click on Finish. If you encounter any error. Ignore it.

**Confirm TestNG is installed Successfully**

Click on the Menu bar for Window → Show View-->Other. You will get a popup window SHOW VIEW. Now expand JAVA folder. You can see TestNG in the folder. So TestNG is installed successfully.

**Working on TestNG**

We don’t have “public static void main” in TestNG. We have only annotations.

Using annotations TestNG will run your Testcases.

To run the Script Right click on the respective Class file and goto RunAs-->TestNgTest.

**Annotations in TestNG**

**@Test- i**If you run the program all your tests will be executed.

**@BeforeMethod-** For each and every Test your BeforeMethod executes first

**@AfterMethod-** For each and every Test your BeforeMethod executes last

**@BeforeClass-** It executes before your class starts executing the respective class.

**@AfterClass-** It executes after your class ends executing the respective class.

**@BeforeTest-** It executes first before your Test and  BeforeMethod, but only once in the total program.

**@AfterTest-** It executes last after your Test  and BeforeMethod, but only once in the total program

**@BeforeSuite-** It executes before your total Batch file starts executing the classes once

**@AfterSuite-** It executes after your total Batch file starts executing the classes once

**Writing a Program using TestNg**

Create a New Class without public static void main function. Because TestNg is built in a way to execute the program without the help of main function.

**For Eg:**

@Test //TestCase

**public** **void** loginTest(){

System.*out*.println("Login to App");

  }

**Using Annotations in TestNg**

Write @Test before your method and Run the program with help of Junit.

**How to Run a Program with help of TestNg**

Right Click on your class file and Goto RunAs, and click on TestNg Test.  The methods which have @Test annotations will be executed. You can have n number of @Test methods. Make sure all your Methods have “Test” word in their method name. So it will execute in a particular order.

**Reports in TestNg**

Your Normal o/p will be in your Console window. You will have another window TestNg opened. That will give you clear information of the Test whether it’s executed successfully.

       You can also see a TestOutput Folder in your Project. If you expand it the index.html report file is automatically generated in TestNg.

**How to Run as a Batch File**

If you want to run the execution as Suite. You need TestNg.xml file. Copy the TestNg.xml file which is sent to you in your Project location. Open the xml file and do some changes.

**TestNG.xml**

In TestNG.xml file you have to specify your Suite name and your class file information as which are the class files need to Run.

**For Eg:**

**Give your Suite Name as**

<suite name="Yahoo">

**Give your TestName and class file info**

//TestName can be anything

<test name="Yahoo News Test">

    <classes>

//When you specify the class name give with PackageName.ClassName. Its Case Sensititve

        <class name="WebdriverBasic.YahooNewsTest" ></class>

    </classes>

</test>

**Run your Test in TestNG.xml file**

Right Click on your TestNG.xml file and RunAs TestNG Suite.

Now based on your Annotations in your Class file. All your Testcases will be executed as per Order.

**Skipping the Test Case**

If you want to Skip a Test Case. Goto to the respective Class file and on the @BeforeTest method give a condition to Skip the Total class file.

**For Eg:**

@BeforeTest

public void xyz(){

System.out.println("Executing Before Test from YahooNewsTest");

throw new SkipException("Skipping the Test");

}

**Generating the Reports**

Now goto the Project and open the test-output folder. You can see index.html file. Click on it you can see the Output of the Execution.

**Parameterizing Test Case**

If we want to Work with different datas. We can use the annotation @DataProvider.

Which will help us to work on 2dim Object Array.

**Xslt Report**

Gives you an User friendly Report after your Execution. If you need to generate the xslt Report. You need ANT tool to execute the execution.

**Jar Files need to work with xslt Report**

* saxon-8.7
* SaxonLiaison

**Steps to be Configured in build.xml**

In your build.xml file the reports folder value we need to give as test-output. The default folder test-output will save the reports.

On the Run target we give the information testng.xml. So the Testcases which is present in testng.xml alone will be executed.

You also have an another Target named “makexsltreports”. which will help you to prepare the xslt report.

**Target clean**

Deletes the Build folder

**Target compile**

Creates the Build folder, compiles all your Java files present on src folder based on the respective JDK version present on it and creates the class files on the Build folder.

**Target run**

Takes your testcases from testng.xml file and run the files with help of class files present in build folder.

**Target Makexsltreports**

Based on your Execution xslt reports will be prepared

**GITHUB:**

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

**A GitHub Use Case**

Let’s consider the case of Decathlon, the world’s largest sporting goods retail brand. The company has over 1600 stores in 57 countries, with more than 87,000 employees.

Every company, no matter how large or small, inevitably experiences challenges and obstacles. We could best summarize Decathlon’s problems as:

1. How would the company maintain workflow visibility and avoid redundancies in such a large workforce?
2. How would the company hire developers for so many diverse locations?

[GitHub](https://www.simplilearn.com/tutorials/git-tutorial/git-vs-github) to the rescue! GitHub is not only an affordable resource but also features a great open-source community. Since it is a cloud-based tool, the code is conveniently visible across the entire client organization, facilitating every participant's contributions.

GitHub allows collaboration with developers from all over the world. Open-source solutions like GitHub enable potential developers to contribute and share their knowledge to benefit the global community.

The version control system, or VCS, is the element in [Git](https://www.simplilearn.com/tutorials/git-tutorial/what-is-git) that is best suited for tackling Decathlon’s two problems. So, let’s expand our knowledge of GitHub by taking a closer look at the Git version control system and see why it’s such a game-changer.

**What is a Version Control System?**

The Git version control system, as the name suggests, is a system that records all the modifications made to a file or set of data so that a specific version may be called up later if needed. The system makes sure that all the team members are working on the file’s latest version, and everyone can work simultaneously on the same project.

Before we dig deeper into what GitHub is, we must examine first what the ‘Git’ part is all about.

## **What is Git?**

[Git](https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner) is a version control system used for tracking changes in computer files, making it a top-rated utility for programmers world-wide. Git can handle projects of any size.

Git is used to coordinate the workflow among project team members and track their progress over time. It also benefits both programmers and non-technical users by keeping track of their project files. Git allows multiple users to work together without disrupting each other’s work.

Now that you've been introduced to Git, you have the foundation needed to understand what is GitHub better.

## **What is GitHub?**

GitHub is a Git repository hosting service that provides a web-based graphical interface. It is the world’s largest coding community. Putting a code or a project into GitHub brings it increased, widespread exposure. Programmers can find source codes in many different languages and use the command-line interface, Git, to make and keep track of any changes.

GitHub helps every team member work together on a project from any location while facilitating collaboration. You can also review previous versions created at an earlier point in time.

So now we know what Git and GitHub are. Time to gain a better understanding of the importance and relevance of what is GitHub by exploring its features.

## **What are GitHub’s Features**

### **1. Easy Project Management**

GitHub is a place where project managers and developers come together to coordinate, track, and update their work so that projects are transparent and stay on schedule.

### **2. Increased Safety With Packages**

Packages can be published privately, within the team, or publicly to the open-source community. The packages can be used or reused by downloading them from GitHub.

### **3. Effective Team Management**

GitHub helps all the team members stay on the same page and organized. Moderation tools like Issue and Pull Request Locking help the team to focus on the code.

### **4. Improved Code Writing**

[Pull requests](https://www.simplilearn.com/tutorials/git-tutorial/git-pull-request) help the organizations to review, develop, and propose new code. Team members can discuss any implementations and proposals through these before changing the source code.

### **5. Increased Code Safety**

GitHub uses dedicated tools to identify and analyze vulnerabilities to the code that other tools tend to miss. Development teams everywhere work together to secure the software supply chain, from start to finish.

### **6. Easy Code Hosting**

All the code and documentation are in one place. There are millions of repositories on GitHub, and each repository has its own tools to help you host and release code.

Now that we’ve gained some familiarity with GitHub let’s check out the competition.

## **So How Do You Get Started With GitHub?**

It’s easy to get things going with GitHub. For starters, click onto the [GitHub site](https://github.com/) and create an account. Then, consider installing [Git](https://git-scm.com/downloads)on your system, especially if you plan on using your local computer. Then, go to your terminal and make yourself [known to Git](https://www.simplilearn.com/tutorials/git-tutorial/git-installation-on-windows) by setting up your user name in every repository. Use this command:

git config --global user.name "<your\_name\_here>"

Make sure the “your name here” parameter is your own name. Pick any name you’d like.

Next, share your email address with Git. It should be the same address you entered when you joined GitHub.

git config --global user.email "<[your\_email@email.com](mailto:your_email@email.com)>"

You’re now ready to use Git!

## **How Do You Use Git and GitHub?**

Here’s a very broad overview of the steps you need to use both Git and GitHub. You can find more details regarding the specific commands and syntax here on [opensource.com](https://opensource.com/article/18/1/step-step-guide-git).

1. Create your GitHub account, which you should have already done, thanks to the previous section!
2. Create a repository or “repo” for short. This is where you store your code.
3. Build a file.
4. Make a commit. Whenever you create a file or change it, you create a Git commit to store the new version.
5. Connect your repo with your computer system.

**GIT CHEAT SHEET:**

**INSTALLATION & GUIS:**

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

**GitHub for Windows:**

htps://windows.github.com

**Git for All Platforms:**

<http://git-scm.com>

**SETUP:**

Configuring user information used across all local repositories

**git config --global user.name “[firstname lastname]”**

set a name that is identifiable for credit when review version history

**git config --global user.email “[valid-email]”**

set an email address that will be associated with each history marker

**git config --global color.ui auto** set automatic command line coloring for Git for easy reviewing

**SETUP & INIT** Configuring user information, initializing and cloning repositories git init initialize an existing directory as a Git repository git clone [url] retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT Working with snapshots and the Git staging area git status show modified files in working directory, staged for your next commit git add [file] add a file as it looks now to your next commit (stage) git reset [file] unstage a file while retaining the changes in working directory git diff diff of what is changed but not staged git diff --staged diff of what is staged but not yet commited git commit -m “[descriptive message]” commit your staged content as a new commit snapshot

BRANCH & MERGE Isolating work in branches, changing context, and integrating changes git branch list your branches. a \* will appear next to the currently active branch git branch [branch-name] create a new branch at the current commit git checkout switch to another branch and check it out into your working directory git merge [branch] merge the specified branch’s history into the current one git log show all commits in the current branch’s history

**Code Files:**

**src/main/java**

**pageObjects**

package pageObjects;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import testBase.BaseClass;

public class Home\_PO extends BaseClass{

public Home\_PO(WebDriver driver) {

this.driver = driver;

PageFactory.initElements(driver, this);

}

@FindBy(xpath = "//a[@aria-label='Amazon']")

private WebElement amazon\_logo;

@FindBy(id = "nav-global-location-popover-link")

private WebElement location\_text;

@FindBy(id = "nav-search-dropdown-card")

private WebElement all\_dropdown;

@FindBy(name = "field-keywords")

private WebElement search\_textbox;

@FindBy(xpath = "//input[@id='nav-search-submit-button']")

private WebElement search\_button;

@FindBy(xpath = "//a[contains(@aria-label,'Choose a language')]")

private WebElement language\_dropdown;

@FindBy(xpath = "//a[@data-nav-role='signin' and @id='nav-link-accountList']")

private WebElement signIn\_dropdown;

@FindBy(id = "nav-orders")

private WebElement orders\_link;

@FindBy(id = "nav-cart")

private WebElement cart\_link;

@FindBy(xpath = "//a[@aria-label='Open Menu']")

private WebElement openMenu\_drop;

@FindBy(xpath = "//div[@id='nav-xshop-container']//div[@class='nav-progressive-content']//a")

private List<WebElement> allCategories\_link;

@FindBy(id = "nav-swmslot")

private WebElement ad\_on\_Category\_link;

public WebElement getAmazon\_logo() {

return amazon\_logo;

}

public WebElement getLocation\_text() {

return location\_text;

}

public WebElement getAll\_dropdown() {

return all\_dropdown;

}

public WebElement getSearch\_textbox() {

return search\_textbox;

}

public WebElement getSearch\_button() {

return search\_button;

}

public WebElement getLanguage\_dropdown() {

return language\_dropdown;

}

public WebElement getSignIn\_dropdown() {

return signIn\_dropdown;

}

public WebElement getOrders\_link() {

return orders\_link;

}

public WebElement getCart\_link() {

return cart\_link;

}

public WebElement getOpenMenu\_drop() {

return openMenu\_drop;

}

public List<WebElement> getAllCategories\_link() {

return allCategories\_link;

}

public WebElement getAd\_on\_Category\_link() {

return ad\_on\_Category\_link;

}

}

package pageObjects;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import testBase.BaseClass;

public class SearchProduct\_PO extends BaseClass{

public SearchProduct\_PO(WebDriver driver) {

this.driver = driver;

PageFactory.initElements(driver, this);

}

@FindBy(css = "input#twotabsearch textbox")

private WebElement search\_TextBox;

@FindBy(xpath = "(//div[@class='autocomplete-results-container']//div[contains(@class,'s-suggestion') and @role='button'])[1]")

private WebElement first\_AutoSuggest\_Link;

@FindBy(xpath = "//div[@class='autocomplete-results-container']//div[contains(@class,'s-suggestion') and @role='button']")

private List<WebElement> all\_AutoSuggest\_Link;

@FindBy(xpath = "//span[text()='RESULTS']")

private WebElement results\_Header;

@FindBy(xpath = "(//div[@data-component-type='s-search-result'])[1]//h2//a")

private WebElement firstProduct\_result\_Link;

@FindBy(xpath = "(//div[@data-component-type='s-search-result'])[1]//h2//span")

private WebElement firstProduct\_result\_Text;

@FindBy(xpath = "(//div[@data-component-type='s-search-result'])[1]//span[@class='a-price-whole']")

private WebElement firstProductPrice\_result\_Text;

public WebElement getSearch\_TextBox() {

return search\_TextBox;

}

public WebElement getFirst\_AutoSuggest\_Link() {

return first\_AutoSuggest\_Link;

}

public List<WebElement> getAll\_AutoSuggest\_Link() {

return all\_AutoSuggest\_Link;

}

public WebElement getResults\_Header() {

return results\_Header;

}

public WebElement getFirstProduct\_result\_Link() {

return firstProduct\_result\_Link;

}

public WebElement getFirstProduct\_result\_Text() {

return firstProduct\_result\_Text;

}

public WebElement getFirstProductPrice\_result\_Text() {

return firstProductPrice\_result\_Text;

}

}

**testBase**

package testBase;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.lang.reflect.Method;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import com.relevantcodes.extentreports.ExtentReports;

import com.relevantcodes.extentreports.ExtentTest;

import com.relevantcodes.extentreports.LogStatus;

public class BaseClass {

public static WebDriver driver;

public static ExtentReports extent;

public ExtentTest extentTest;

public static Properties prop;

public BaseClass() {

prop = new Properties();

try {

FileInputStream ip = new FileInputStream(System.getProperty("user.dir") + "\\src\\test\\resources\\config.properties");

prop.load(ip);

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

@BeforeSuite

public void init() {

System.setProperty("webdriver.chrome.driver", System.getProperty("user.dir")+"\\src\\test\\resources\\chromedriver\_100.04.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);

driver.get(prop.getProperty("url"));

extent = new ExtentReports(System.getProperty("user.dir") + "/test-output/ExtentReport.html", true);

// extent = new ExtentReports(System.getProperty("user.dir") + "/target/reports/{testng-report-dir}/ExtentReport.html", true);

extent.addSystemInfo("Host Name", "Windows");

extent.addSystemInfo("User Name", "Murali Sukumar");

extent.addSystemInfo("Environment", "QA");

}

@AfterSuite

public void EndTest() throws InterruptedException {

extent.endTest(extentTest); // ending test and ends the current test and prepare to create html report

Thread.sleep(5000);

driver.quit();

}

public void FailedTC(String testMethodName) {

File srcFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

try {

FileUtils.copyFile(srcFile, new File(System.getProperty("user.dir")+"\\src\\main\\resources\\screenshots\\"

+ testMethodName + "\_" + ".png"));

} catch (IOException e) {

e.printStackTrace();

}

}

public static void setExtent() {

}

public static ExtentReports getInstance() {

if (extent == null) {

setExtent();

}

return extent;

}

public void setup() {

extent = BaseClass.getInstance();

}

@AfterTest

public void endReport() {

extent.flush();

// extent.close();

}

public static String getScreenshot(WebDriver driver, String screenshotName) throws IOException {

String dateName = new SimpleDateFormat("yyyyMMddhhmmss").format(new Date());

TakesScreenshot ts = (TakesScreenshot) driver;

File source = ts.getScreenshotAs(OutputType.FILE);

// after execution, you could see a folder "FailedTestsScreenshots"

// under src folder

String destination = System.getProperty("user.dir") + "src\\main\\resources\\FailedTestsScreenshots\\" + screenshotName + dateName

+ ".png";

File finalDestination = new File(destination);

FileUtils.copyFile(source, finalDestination);

return destination;

}

@BeforeMethod

public void startTest(Method methodName) throws IOException {

System.out.println("Executing Before Method");

extentTest = extent.startTest(methodName.getName());

}

@AfterMethod

public void tearDown(ITestResult result) throws IOException {

if (result.getStatus() == ITestResult.FAILURE) {

extentTest.log(LogStatus.FAIL, "TEST CASE FAILED IS " + result.getName()); // to add name in extent report

extentTest.log(LogStatus.FAIL, "TEST CASE FAILED IS " + result.getThrowable()); // to add error/exception in extent report

String screenshotPath = BaseClass.getScreenshot(driver, result.getName());

extentTest.log(LogStatus.FAIL, extentTest.addScreenCapture(screenshotPath)); // to add screenshot in extent report

} else if (result.getStatus() == ITestResult.SKIP) {

extentTest.log(LogStatus.SKIP, "Test Case SKIPPED IS " + result.getName());

} else if (result.getStatus() == ITestResult.SUCCESS) {

extentTest.log(LogStatus.PASS, "Test Case PASSED IS " + result.getName());

}

}

}

**utilities**

package utilities;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import java.util.function.Function;

import org.apache.commons.io.FileUtils;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.StaleElementReferenceException;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

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

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

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

import org.testng.asserts.SoftAssert;

import testBase.BaseClass;

public class TestUtilities extends BaseClass{

public static WebDriver driver;

public WebDriverWait wait;

public Select s;

public int i;

public int count = 3;

public SoftAssert sa;

public static String filepath = "user.dir" + "/Screenshots";

public TestUtilities(WebDriver driver) {

this.driver=driver;

}

public boolean verifyElementDisplayed(WebElement element) {

boolean status= false;

if(element.isDisplayed()) {

status= true;

}

return status;

}

public boolean verifyMultipleElementsDisplayed(List<WebElement> elements, String att\_or\_Text, String attrType, String value) {

boolean status= false;

int size= elements.size();

int stat= 0;

if(att\_or\_Text.equalsIgnoreCase("getAttribute")) {

for(WebElement element:elements) {

if(element.getAttribute(attrType).contains(value)) {

stat++;

}

}

}

else if(att\_or\_Text.equalsIgnoreCase("getText")) {

for(WebElement element:elements) {

if(element.getText().contains(value)) {

stat++;

}

}

}

if(size==stat) {

status=true;

}

return status;

}

public List<String> getAllElements\_Text(List<WebElement> elements) {

ArrayList<String> myElements\_Text= new ArrayList<String>();

for(WebElement element:elements) {

myElements\_Text.add(element.getText());

}

return myElements\_Text;

}

public void printFromStringList(List<String> myDatas) {

for(String text:myDatas) {

System.out.println(text);

}

}

public void printFromLists(List<WebElement> elements) {

ArrayList<String> myElements\_Text= new ArrayList<String>();

for(WebElement element:elements) {

System.out.println(element.getText());

}

}

public void printFromList\_JS(List<WebElement> elements) {

ArrayList<String> myElements\_Text= new ArrayList<String>();

for(WebElement element:elements) {

JavascriptExecutor jsExecutor = (JavascriptExecutor) driver;

System.out.println(((JavascriptExecutor) driver).executeScript("return arguments[0].innerHTML;",element));

}

}

public void blindWait(int seconds) throws InterruptedException {

Thread.sleep(seconds\*1000);

}

public WebDriver switchtoFrameByWebElement(WebElement frameElement) {

driver.switchTo().frame(frameElement);

return driver;

}

public void switchtoFrameByIndex(int frameNumber) {

driver.switchTo().frame(frameNumber);

}

public void switchtoFrameByName(String frameName) {

driver.switchTo().frame(frameName);

}

public void explicitWaitinit(WebDriver driver) {

wait = new WebDriverWait(driver, 10);

}

public void visibilityofWebElement(WebElement element) {

wait.until(ExpectedConditions.visibilityOf(element));

}

public void elementToBeClickableAt(WebElement element) {

wait.until(ExpectedConditions.elementToBeClickable(element)).click();

}

public void SelectByValue(WebElement ele, String SelectValue) {

s = new Select(ele);

s.selectByValue(SelectValue);

}

public void SelectValue(WebDriver driver, WebElement element, String value, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.elementToBeClickable(element));

s = new Select(element);

s.selectByValue(value);

}

public void SelectVisibleText(WebDriver driver, WebElement element, String value, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.elementToBeClickable(element));

s = new Select(element);

s.selectByVisibleText(value);

}

public void SelectByVisibleText(WebElement ele, String StringText) {

s = new Select(ele);

s.selectByVisibleText(StringText);

}

public String CreateFolderForScreenshot() {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH.mm.ss");

Date date = new Date();

File file = new File(prop.getProperty("ScreenshotPath") + sdf.format(date));

file.mkdir();

String newScreenshotFolderPath = file.getAbsolutePath();

return newScreenshotFolderPath;

}

public void takeScreenshots(WebDriver wd, String newScreenshotFolderPath, String filename) throws Exception {

// SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH.mm.ss");

// Date date= new Date();

TakesScreenshot ts = (TakesScreenshot) wd;

File src = ts.getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src, new File(prop.getProperty("ScreenshotPath") + "\\" + filename + ".png"));

}

public void takeScreenshot(WebDriver driver, String fileName) throws Exception {

try {

File file = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(file, new File(prop.getProperty("ScreenshotPath") + fileName + ".png"));

// FileHandler.copy(file, new

// File("C:\\Users\\ddillibai\\eclipse-workspace\\Vodafone\_uk\\test-output"+fileName+".jpg"));

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

}

public void FailedTC() {

File srcFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

try {

FileUtils.copyFile(srcFile, new File(

System.getProperty("user.dir")+"/FailedShots"));

} catch (IOException e) {

e.printStackTrace();

}

}

public static void clickById(WebDriver driver, String id) {

Actions act = new Actions(driver);

act.moveToElement(driver.findElement(By.id(id))).click().build().perform();

}

public void clickByXpath(WebDriver driver, String xpath) throws InterruptedException {

Actions act = new Actions(driver);

act.moveToElement(driver.findElement(By.xpath(xpath))).perform();

act.moveToElement(driver.findElement(By.xpath(xpath))).click().build().perform();

}

public void selectDateByJS(WebDriver driver, WebElement element, String dateVal) {

JavascriptExecutor js = ((JavascriptExecutor) driver);

js.executeScript("arguments[0].setAttribute('value','" + dateVal + "');", element);

}

public static void selectCalenderDateJS(WebDriver driver, String id, String date) {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("document.getElementById('" + id + "').textContent='" + date + "';");

}

static String pageLoadStatus;

static JavascriptExecutor js;

public static void waitForPageLoad1() {

do {

js = (JavascriptExecutor) driver;

pageLoadStatus = (String) js.executeScript("return document.readyState");

} while (!pageLoadStatus.equals("complete"));

System.out.println("Page Loaded.");

}

public void waitForPageLoad(WebDriver driver, int timeout) {

wait = new WebDriverWait(driver, timeout);

wait.until(new Function<WebDriver, Object>() {

public Object apply(WebDriver webDriver) {

return ((JavascriptExecutor) webDriver).executeScript("return document.readyState").equals("complete");

}

});

}

/\*

\* public void waitForPageLoad(WebDriver driver, int timeout) { wait = new

\* WebDriverWait(driver, timeout); wait.until(webDriver -> ((JavascriptExecutor)

\* webDriver).executeScript("return document.readyState") .equals("complete"));

\* }

\*/

public static void selectDateByJSDate(WebDriver driver, WebElement element, String dateVal) {

JavascriptExecutor js = ((JavascriptExecutor) driver);

js.executeScript("arguments[0].setAttribute('value','" + dateVal + "');", element);

}

public void waitForPageLoad2(WebDriver driver, int timeout) {

// Wait<WebDriver> wait = new WebDriverWait(driver, 30);

wait = new WebDriverWait(driver, timeout);

wait.until(new Function<WebDriver, Boolean>() {

public Boolean apply(WebDriver driver) {

// System.out.println("Current Window State : "

// + String.valueOf(((JavascriptExecutor) driver).executeScript("return document.readyState")));

return String.valueOf(((JavascriptExecutor) driver).executeScript("return document.readyState"))

.equals("complete");

}

});

}

// Method to enter the values

public void sendKeys(WebDriver driver, WebElement element, String value, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.visibilityOf(element));

element.sendKeys(value);

}

// method to clear the text field

public void clear(WebDriver driver, WebElement element, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.visibilityOf(element));

element.clear();

}

// method to click on element using explicit wait

public void clickOn(WebDriver driver, WebElement element, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.elementToBeClickable(element));

element.click();

}

// method to check visibility of an element

public WebElement clickOn1(WebDriver driver, WebElement element, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.visibilityOf(element));

return element;

}

public void clickOnAfterPageLoad(WebDriver driver, WebElement element, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.stalenessOf(element));

new WebDriverWait(driver, timeout).until(ExpectedConditions.visibilityOf(element));

element.click();

}

// method to click using actions class

public void clickOnActions(WebDriver driver, WebElement element, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.elementToBeClickable(element));

Actions actions = new Actions(driver);

actions.moveToElement(element).click().build().perform();

}

// method to click using js

public void clickUsingJS(WebDriver driver, WebElement element, int timeout) {

JavascriptExecutor executor = (JavascriptExecutor) driver;

executor.executeScript("arguments[0].click();", element);

}

// method to get text

public String getText(WebElement element) {

new WebDriverWait(driver, 10).until(ExpectedConditions.elementToBeClickable(element));

return element.getText();

}

// method to select the value

public void selectValue(WebDriver driver, WebElement element, int timeout) {

JavascriptExecutor executor = (JavascriptExecutor) driver;

executor.executeScript("return document.readyState", "complete");

new WebDriverWait(driver, timeout).until(ExpectedConditions.visibilityOf(element));

executor.executeScript("arguments[0].click();", element);

}

public static void clickOnforStale(WebDriver driver, WebElement locator, int timeout) {

new WebDriverWait(driver, timeout).ignoring(StaleElementReferenceException.class)

.until(ExpectedConditions.elementToBeClickable(locator));

locator.click();

}

public void inVisibilityofEle(WebDriver driver, WebElement locator, int timeout) {

new WebDriverWait(driver, timeout).until(ExpectedConditions.invisibilityOf(locator));

}

public class ReadExcelConfig {

XSSFWorkbook wb;

XSSFSheet sheet1;

public ReadExcelConfig(String excelpath) {

try {

File src = new File(filepath);

FileInputStream fis = new FileInputStream(src);

wb = new XSSFWorkbook(fis);

} catch (Exception e) {

e.printStackTrace();

}

}

public String getData(int sheetNumber, int row, int column) {

sheet1 = wb.getSheetAt(sheetNumber);

String data = sheet1.getRow(row).getCell(column).getStringCellValue();

return data;

}

public int getRowCount(int SheetIndex) {

int row = wb.getSheetAt(SheetIndex).getLastRowNum();

row = row + 1;

return row;

}

}

// This method is used to get current date(yyyy-mm-dd)

public String getcurrentdate() {

SimpleDateFormat formater = new SimpleDateFormat("yyyy-MM-dd");

Date currentdate = new Date();

String Date = formater.format(currentdate);

System.out.println(Date);

return Date;

}

// This method is used to add days for current date

public String getaltereddate(int daystobeadded) {

Calendar cal = Calendar.getInstance();

cal.add(Calendar.DAY\_OF\_MONTH, daystobeadded);

SimpleDateFormat formater = new SimpleDateFormat("yyyy-MM-dd");

String Date = formater.format(cal.getTime());

System.out.println("Taking 5+ days from current day for api validation " + Date);

return Date;

}

}

**src/test/java**

**home\_Tests**

package home\_Tests;

import java.io.IOException;

import java.util.List;

import org.openqa.selenium.By;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import pageObjects.Home\_PO;

import pageObjects.SearchProduct\_PO;

import testBase.BaseClass;

import utilities.TestUtilities;

public class Home\_Tests extends BaseClass{

public Home\_PO home\_PO;

public TestUtilities util;

public Home\_Tests home\_TESTS;

@BeforeClass //#AM-001

public void beforeClass() {

home\_PO = new Home\_PO(driver);

util = new TestUtilities(driver);

util.explicitWaitinit(driver);

}

@Test(priority = 0)

public void verifyHeaders() throws SecurityException, IOException {

if(util.verifyElementDisplayed(home\_PO.getAmazon\_logo())) {

System.out.println("Amazon logo displayed");}

if(util.verifyElementDisplayed(home\_PO.getLocation\_text())) {

System.out.println("Location displayed");}

util.visibilityofWebElement(home\_PO.getAll\_dropdown());

if(util.verifyElementDisplayed(home\_PO.getAll\_dropdown())) {

System.out.println("Search Dropdown displayed");}

if(util.verifyElementDisplayed(home\_PO.getSearch\_textbox())) {

System.out.println("Search Box displayed");}

if(util.verifyElementDisplayed(home\_PO.getSearch\_button())) {

System.out.println("Search Key displayed");}

if(util.verifyElementDisplayed(home\_PO.getLanguage\_dropdown())) {

System.out.println("Language selection displayed");}

if(util.verifyElementDisplayed(home\_PO.getSignIn\_dropdown())) {

System.out.println("SignIn selection displayed");}

if(util.verifyElementDisplayed(home\_PO.getOrders\_link())) {

System.out.println("Returns& Orders displayed");}

if(util.verifyElementDisplayed(home\_PO.getCart\_link())) {

System.out.println("Cart displayed");}

}

@Test(priority = 1)

void verifyCategories\_AmazonHome() throws SecurityException, IOException {

// startTest(new Object(){}.getClass().getEnclosingMethod());

driver.navigate().to(prop.getProperty("url"));

if(util.verifyElementDisplayed(home\_PO.getAmazon\_logo())) {

System.out.println("Amazon logo displayed");}

if(util.verifyElementDisplayed(home\_PO.getOpenMenu\_drop())) {

System.out.println("ALL slider icon displayed");}

if(util.verifyMultipleElementsDisplayed(

home\_PO.getAllCategories\_link(),"getAttribute","class","nav-a")) {

System.out.println("All elements are displayed under Category section");}

List<String> categories= util.getAllElements\_Text(home\_PO.getAllCategories\_link());

util.printFromList\_JS(home\_PO.getAllCategories\_link());

if(util.verifyElementDisplayed(home\_PO.getAd\_on\_Category\_link())) {

System.out.println("Amazon Ad image displayed");}

}

// #AM-005 Verify Sign In Page

@Test(priority = 2)

void VerifySignInPage() {

driver.navigate().to(prop.getProperty("url"));

if(util.verifyElementDisplayed(home\_PO.getAmazon\_logo())) {

System.out.println("Amazon logo displayed");}

if(util.verifyElementDisplayed(home\_PO.getOpenMenu\_drop())) {

System.out.println("ALL slider icon displayed");}

if(util.verifyMultipleElementsDisplayed(

home\_PO.getAllCategories\_link(),"getAttribute","class","nav-a")) {

System.out.println("All elements are displayed under Category section");}

List<String> categories= util.getAllElements\_Text(home\_PO.getAllCategories\_link());

// amazonHomePage.printFromList(categories);

util.printFromList\_JS(home\_PO.getAllCategories\_link());

if(util.verifyElementDisplayed(driver.findElement(By.id("nav-swmslot")))) {

System.out.println("Amazon Ad image displayed");}

}

}

**Search\_ProductTests**

package searchProduct\_Tests;

import java.io.IOException;

import java.util.List;

import org.openqa.selenium.By;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import pageObjects.Home\_PO;

import pageObjects.SearchProduct\_PO;

import testBase.BaseClass;

import utilities.TestUtilities;

public class Search\_Tests extends BaseClass{

public Home\_PO home\_PO;

public TestUtilities util;

public Search\_Tests search\_TESTS;

public SearchProduct\_PO searchProduct\_PO;

@BeforeClass //#AM-001

public void beforeClass() {

home\_PO = new Home\_PO(driver);

searchProduct\_PO = new SearchProduct\_PO(driver);

util = new TestUtilities(driver);

util.explicitWaitinit(driver);

}

// #AM-008 Search for a Product in Search bar

@Test(priority = 3)

void searchProduct\_SearchTextbox() throws InterruptedException {

driver.navigate().to(prop.getProperty("url"));

if(util.verifyElementDisplayed(home\_PO.getAmazon\_logo())) {

System.out.println("Amazon logo displayed");}

if(util.verifyElementDisplayed(searchProduct\_PO.getSearch\_TextBox())) {

System.out.println("ALL slider icon displayed");}

util.elementToBeClickableAt(searchProduct\_PO.getSearch\_TextBox());

searchProduct\_PO.getSearch\_TextBox().sendKeys("WWE 2K22");

util.blindWait(3);

util.visibilityofWebElement(searchProduct\_PO.getFirst\_AutoSuggest\_Link());

util.printFromLists(searchProduct\_PO.getAll\_AutoSuggest\_Link());

util.elementToBeClickableAt(searchProduct\_PO.getFirst\_AutoSuggest\_Link());

util.visibilityofWebElement(searchProduct\_PO.getResults\_Header());

if(util.verifyElementDisplayed(searchProduct\_PO.getFirstProduct\_result\_Link())) {

System.out.println("Product displayed for Search result");}

System.out.println(util.getText(searchProduct\_PO.getFirstProduct\_result\_Text()));

System.out.println(util.getText(searchProduct\_PO.getFirstProductPrice\_result\_Text()));

util.elementToBeClickableAt(searchProduct\_PO.getFirstProduct\_result\_Link());

}

}

**pom.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.technologics</groupId>

<artifactId>Amazon\_Intern</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>framework</name>

<!-- FIXME change it to the project's website -->

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.14.3</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.13</version>

</dependency>

<!-- https://mvnrepository.com/artifact/log4j/log4j -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.13</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.relevantcodes/extentreports -->

<dependency>

<groupId>com.relevantcodes</groupId>

<artifactId>extentreports</artifactId>

<version>2.41.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.8</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.4</version>

</dependency>

</dependencies>

<build>

<pluginManagement><!-- lock down plugins versions to avoid using Maven

defaults (may be moved to parent pom) -->

<plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

<configuration>

<!-- Suite testng xml file to consider for test execution -->

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

<!-- Compiler plugin configures the java version to be usedfor compiling

the code -->

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

**testNG.xml**

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"Amazon Suite"*>

<test name=*"Home Test"*>

<classes>

<class name=*"home\_Tests.Home\_Tests"* ></class>

</classes>

</test>

<test name=*"Search Test"*>

<classes>

<class name=*"searchProduct\_Tests.Search\_Tests"* ></class>

</classes>

</test>

</suite>

**Jenkins**

Jenkin is the most popular, open source [Continuous Integration tool](https://www.guru99.com/top-20-continuous-integration-tools.html). It has tons of plugins that enhance its functionality. In this online Jenkins tutorial for beginners, you will learn Jenkins basics like how to install Jenkins, create jobs, pipelines, integrate with GitHub, Selenium, and other tools.

Pipelines automate testing and reporting on isolated changes in a larger code base in real time and facilitates the integration of disparate branches of the code into a main branch. They also rapidly detect defects in a code base, build the software, [automate testing](https://www.techtarget.com/searchsoftwarequality/definition/automated-software-testing) of their builds, prepare the code base for deployment (delivery), and ultimately deploy code to containers and virtual machines, as well as bare metal and cloud servers. There are several commercial versions of Jenkins. This definition only describes the upstream [open source](https://www.techtarget.com/whatis/definition/open-source) project.

### History

Jenkins is a fork of a project called Hudson, which was trademarked by Oracle. Hudson was eventually donated to the Eclipse Foundation and is no longer under development. Jenkins development is now managed as an open source project under the governance of the CD Foundation, an organization within the Linux Foundation.

Continuous integration has evolved since its conception. Originally, a daily [build](https://www.techtarget.com/searchsoftwarequality/definition/build) was the standard. Now, the usual rule is for each team member to submit work, called a commit, on a daily (or more frequent) basis and for a build to be conducted with each significant change. When used properly, continuous integration provides various benefits, such as constant feedback on the status of the software. Because CI detects deficiencies early on in development, defects are typically smaller, less complex and easier to resolve.

### Jenkins and CI/CD

Over time, [continuous delivery](https://www.techtarget.com/searchitoperations/definition/continuous-delivery-CD) and deployment features have been added to Jenkins. Continuous delivery is the process of automating the building and packaging of code for eventual deployment to test, production staging, and production environments. Continuous deployment automates the final step of deploying the code to its final destination.

In both cases, automation reduces the number of errors that occur because the correct steps and best practices are encoded into Jenkins. Jenkins describes a desired state and the automation server ensures that that state is achieved. In addition, the velocity of releases can be increased since deployments are no longer bounded by personnel limitations, such as operator availability. Finally, Jenkins reduces stress on the development and operations team, by removing the need for middle of the night and weekend rollouts.

### How Jenkins works

Jenkins runs as a server on a variety of platforms including Windows, MacOS, Unix variants and especially, [Linux](https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system). It requires a Java 8 VM and above and can be run on the Oracle JRE or [OpenJDK](https://www.theserverside.com/definition/OpenJDK). Usually, Jenkins runs as a Java servlet within a Jetty application server. It can be run on other Java application servers such as Apache Tomcat. More recently, Jenkins has been adapted to run in a [Docker container](https://www.techtarget.com/searchitoperations/definition/Docker). There are read-only Jenkins images available in the Docker Hub online repository.

To operate Jenkins, pipelines are created. A pipeline is a series of steps the Jenkins server will take to perform the required tasks of the CI/CD process. These are stored in a plain text Jenkinsfile. The Jenkinsfile uses a curly bracket syntax that looks similar to [JSON](https://theserverside.techtarget.com/definition/JSON-Javascript-Object-Notation). Steps in the pipeline are declared as commands with parameters and encapsulated in curly brackets. The Jenkins server then reads the Jenkinsfile and executes its commands, pushing the code down the pipeline from committed source code to production runtime. A Jenkinsfile can be created through a GUI or by writing code directly.

### Plugins

A [plugin](https://www.techtarget.com/whatis/definition/plug-in) is an enhancement to the Jenkins system. They help extend Jenkins capabilities and integrated Jenkins with other software. Plugins can be downloaded from the online Jenkins Plugin repository and loaded using the Jenkins Web UI or CLI. Currently, the Jenkins community claims over 1500 plugins available for a wide range of uses.

Plugins help to integrate other developer tools into the Jenkins environment, add new user interface elements to the Jenkins Web UI, help with administration of Jenkins, and enhance Jenkins for build and source code management. One of the more common uses of plugins is to provide integration points for CI/CD sources and destinations. These include software version control systems (SVCs) such as Git and Atlassian BitBucket, container runtime systems -- especially Docker, virtual machine [hypervisors](https://searchservervirtualization.techtarget.com/definition/hypervisor) such as VMware vSphere, public cloud instances including Google Cloud Platform and Amazon AWS, and private cloud systems such as OpenStack. There are also plugins that assist in communicating with operating systems over FTP, CIFS, and SSH.

A plugin is written in Java. Plugins use their own set of Java Annotations and design patterns that define how the plugin is instantiated, extension points, the function of the plugin and the UI representation in the Jenkins Web UI. Plugin development also makes use of Maven deployment to Jenkins.

### Advantages and disadvantages

As is the case with most software, there are pros and cons to Jenkins.  One of the advantages of Jenkins is that it can be extended using plugins. This makes Jenkins adaptable to changes in IT environments. Plugins also contribute to the flexibility of Jenkins, as does the rich scripting and [declarative](https://www.techtarget.com/searchitoperations/definition/declarative-programming) languages that allow for highly custom pipelines. Since Jenkins is highly unopinionated, it fits well into most environments, including complex hybrid and multi-cloud systems.

Jenkins has been around much longer than other solutions in this space. This, plus its flexibility, has led to it being widely deployed. For this reason, Jenkins is well understood, with a broad knowledge base, extensive documentation, and abundant community resources. These resources make it easier to install, manage and troubleshoot Jenkins installation.

Finally, Jenkins and its plugins are built on Java. Java is a proven enterprise development language with a broad ecosystem. This places Jenkins on a solid base that can be extended using common design patterns and frameworks.

Jenkins is, of course, not perfect. While it is easy to install (with simple to follow directions), production Jenkins can be difficult to implement.  Developing production pipelines using Jenkinsfiles requires coding in either its declarative or scripting language. Complex pipelines, especially, can be difficult to code, debug and maintain.

The open source system is also a single server architecture. This makes it easy to install but caps resources to those of a single computer, virtual machine or container. Jenkins does not allow for federation across servers resulting in performance issues. Lack of federation can also lead to a proliferation of independent Jenkins servers that are difficult to manage across a large enterprise.