**Tell me about your self**

Regarding my technical experience, I have total 16 years of experience in QA manual and automation testing

In my total 16 years of experience, I worked more on automation testing using tools like Selenium ,QTP API using Postmans and restassured.

I have worked for different clients like GE, UPS, Alley bank ,FANNIEMAE AND United health care

currently I am working for Carefirst at reston VA as automation developer

I have developed and implemented frameworks using Selenium and UFT for multiple applications.

In my current project I have developed BDD cucumber frame work for one of web application called Dx applciation.

As a part of framework development ,I have developed number of user defined methods in java using oops concepts for reusability, read and write the data in files like excel , browser automation ,data base testing .Worked on ci/cd environment and used git as version control system and Jenkins.

I have worked on git for code pushing and merging the scripts from master branches

I have developed and scheduled the jobs in Jenkins to run the automation scripts .

This is about my technical experience.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

regarding my current project at Carefirst is health care domain orgainziation for federal employees.

We work under DIVISION of FEPOC…Federal Employee program operation center.

There are multiple web application ..I am working on DX application

Dx application have multiple modules like SMC, Financial dash board, updte contact info member permssions ,NMWK ,EOB Explanation of benefits, 90days address verification etc.

We will get the requirements from DO office.

Currently I am working Autiaom..to automaete these applications we are using BDD frame work. Using selenium and Java

These automation scripts schedule through Jenkins and run the test suite every day.

**What is BDD? Cucumber frame work**

A cucumber is a tool based on Behavior Driven Development (BDD) framework which is used to write acceptance tests for the web application. It allows automation of functional validation in easily readable and understandable format (like plain English) to Business Analysts, Developers, Testers, etc.

It is helpful to involve business stakeholders who can’t easily read the code

Cucumber Testing enhances the end-user experience

Style of writing tests allow for easier reuse of code in the tests

Allows quick and easy setup and execution

Behavior Driven Development is an extension of Test Driven Development and it is used to test the system rather than testing the particular piece of code

Major three components are there Feature files, stepdefinations, runner class and HTML reports

WE developed feature files using gherkin kewords using Given,when,and then, background

Given – Pre condition, When --- action and ---- to combine same scenarios, Then –expected result

Background – used to repeat the same steps like login page etc.

We pass the tags per each scenario , based on tags we run the maven command and run the scripts

Step definaitions : Each feature we developed the methods using java called step definitions

Junit runner : to execute the the script we have a runner class its run based on feature name or tags.

import cucumber.api.junit.Cucumber;</pre>

import org.junit.runner.RunWith;

@RunWith(Cucumber.class)

@Cucumber.Options(format={"SimpleHtmlReport:report/smokeTest.html"},tags={"@smokeTest"})

Public class JUnitRunner {

}

At the end Cucumber generates its own HTML format

**@RunWith(),** which tells JUnit what is the test runner class.

**Cucumber supports hooks**, which are blocks of code that run before or after each scenario. ... Cucumber Hooks allows us to better manage the code workflow and helps us to reduce the code redundancy

**Glue property** is used to let the Cucumber framework identify the location of step definition files

**Cucumber Options** tag is used to provide a link between the feature files and step definition files. Each step of the feature file is mapped to a corresponding method on the step definition file

DryRun : It will check basic test whether all featuere files have corresponding step defs or not.

Each test is called a ***Scenario*** and is described using the Scenario: keyword.

In situations where one wants to execute the same ***Scenario*** with various combinations of values or arguments, one could use the ***Scenario Outline***.

But : logical conjunction like OR

Login should be successful but home page should not missing

Feature : High level description of a software feature

Scenario : title of the scenario

Hooks : code blocks run with annotations

Test harness : Framework of tests, test data test results and test report.

**POM:**

**POM : Page object Model :** Also called page chaining model

It is a design pattern.

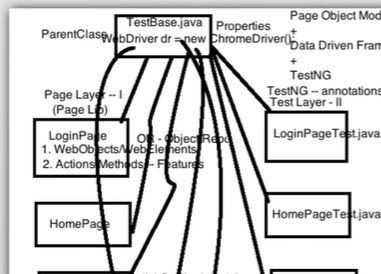
Create separate object repository to store the locators.

For each web page thre should be a separate page class.

**Benefits of POM** : Test objects and functions are separated which helps to more readable and easy maintenance.

1. Create page layer : where we create a class for each page in which we collect the all web elements and methods on that page (com.qa.pages is the package)
2. Test layer : we will create separate class for each page and writing test cases on each page using TestNG annotations like @Test --(com.qa.test is the package)
3. We need to create Base class -- It is the parent class of all pages. At base class we define common scenarios like

Driver initialization, Implicit wait, maximize the page , get(url) etc. (com.qa.base is the package)



Now use the inheretnce concept and access this base clasee (parent class) methods in all test layer classes.

1. Now create one more layer like config.properties (url, browser, username, pwd, db details) also called environment variables. ((com.qa.config is the package)
2. Test data we store in excel file. (com.qa.testData is the package)
3. We create utility class for generic fucntions (db connection, take screen shot, send an email ((com.qa.util is the package)
4. Testreport component : HTML report or TestNG or extent reports ((com.qa.reports is the package)

Technology wise we are using , Java,selenium,Maven,ApachePOI, Extent reports, Log4j,Jenkins, Git, browsers,Grid for parallel testing, VMs/clouds/Saucelabs or local machines

**Page factory Approach :**

The *Page Object Repository* is separated from the *Test Methods* using the *Page Factory* concept. Using it, you can initialize the *Page Objects* or directly instantiate them.

There are two main steps in Page object model which differentiate with regular approach

1.@FindBy annatoation : In normal approach we use findElement or FindElements to *Unlike the regular approach of initializing web page elements using*[**FindElement** or **FindElements**](https://www.toolsqa.com/selenium-webdriver/find-element-selenium/)**,***the Page Factory uses the***@FindBy***annotation*

*2***. Initializing the elements using initElements()***– This is a static method used to initialize the web elements that we locate using the***@FindBy***or other annotation(s), thereby instantiating the page class.*

@FindBy: It is used to locate web elements using different locators strategies.

@FindBys:To locate a web element with more than one search criteria, you can use @FindBys annotation.

@FindAll:The @FindAll annotation locates the web element using more than one criteria, given that at least one criteria match.

@CacheLookUp : using @CacheLookUp, we can store the web elements in cache memory right after reading for the first time.

|  |  |
| --- | --- |
| POM | Page object Factory |
| Finding web elements using By | Finding web elements using @FindBy |
| POM does not provide lazy initialization | Page Factory does provide lazy initialization |
| In POM, one needs to initialize every page object individually | In PageFactory, all page objects are initialized by using the initElements() method |

**What are the challenges in selenium and how to solve them**

As I am using selenium is automation tool in my current project , there should be more challenges compared to other vendor tools like UFT

As selenium is open source tool , there pro and corns in the open source tool

As selenium is browser automation tool, it is not non -web based applications like windows applications

To work on selenium we should have good coding background , where as in tools like UFT minimal coding knowledge is good enough ..for validation or excel read data you need to write the methods and classes in selenium where as in UFT tool we have readily available options available like chekpoints and to read the excel data we have data table concept etc.

Others challenges in selenium is it is not a good candiadate fro image validation , captche ,tool tips etc.

In selenium ,to upload the or download the files we need to use third party tools like Autoit , sikoli etc.

Selenuim doesn’t have any default reporting structure, we need to rely on third party plugins like extent reports,testNG

There is no dedicated people or organization or vendor to support the issues in selenium , we need to go through forums, blogs or we need to do R&D ,there is no direct support like HP for UFT tool.

Handling dynamic objects also one of the challenge , we need create xpaths or css locators accordingly as per element changes using regular expression concepts

**Agile methodology in Automation :**

At its simplest, Agile simply means continuous incremental improvement through small and frequent releases

In our current release we have 46 stories ( story means feature of application (one or multiple features are referred as story)

These stories breakdown in to 4 sprints sprint1 ,sprint 2 sprint3 and sprint 4 (sprint is short sdlc)

Every sprint has 2weeks span, we took 12 story points in first sprint.(3 developers and 3 QA)

Initially we execute the all the features manually make sure that are the application is stable.

Make sure that application is bug free to automate.

If any backlogs in first sprint will be added to next sprint, same thing will repeat in every sprint , in

In last sprint that is sprint 4 we are not doing any new development

Whatever we developed the automation scripts for stories in sprint1 ,2 and 3 are executed in sprint 4 (code freeze sprint) and make sure its bug free.

In some time they may ask interview question , like what is automation is not completed in particular sprint

You can say “ yes it will go to next sprint with proper justification”

After every release there is there will be retrospective meeting and scrum team go throough the release 1 summary like what went good and is there improvement needs we note down what would be the better way where we lacked and how to minimize the backlogs etc.

Scrum master will decide how many stories in each sprint

…

Agile is a process instead of mode

Agile team : Scrum master , dev team and QA team (size varies company to company 5 or 6 or 10 also)

Scrum master role : Whether the team following rules or not

Stoires assigned to Dev and QA team

Whether the deliverables released on time or not

After completing the sprint team , all team members retrospect

They will retrospect about what are the features we planned and what are we finished

In code freeze, we will not do any development we will do the regression test and we we make sure our is module is bug free by running the automation scripts.

In functional freez sprints we will do the development (dev team) and function testing (mandatory and do by the QA Team ) and automation development (QA team)

In code freeze sprint : no development and no functional testing only automation script execution.

In each sprint ,while developers doing the development QA team prepare prototypes like creating the methods creating data ,developing frame works and reusable functions etc

Once application is ready they will change the locators accordingly in their automation script.

Some time in last sprint (cf) during automation we found bug and that functionality is important it has to go for release (not in second release ),

In between two releases dev fix the bug and qa run their scripts and make sure no bug and now they

In grooming session : Business analyst will give headsup on functionality

Scrum teams usually have to work in iterations called sprints which usually last up to two weeks to one month long

**Epic:** a very large user story that is eventually broken down into smaller stories. Epics are sub-divided into stories

**User Stories:** From the client perspective user stories are prepared which defines project or business functions, and it is delivered in a particular sprint as expected.

**Task:** Further down user stories are broken down into different task

To track the project progress burnup and burn down, charts are used.

Burnup Chart: It shows the progress of stories done over time.

Burndown Chart: It shows how much work was left to do overtime.

An Agile retrospective is a meeting that's held at the end of an [iteration](https://www.techtarget.com/searchsoftwarequality/definition/iteration) in [Agile software development](https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development). During the retrospective, the team reflects on what happened in the iteration and identifies actions for improvement going forward

**Retrospective:** a session where the team and scrum master reflect on the process and make commitments to improve.

Acceptance criteria are a set of statements that describes the conditions a software product or a project deliverable must satisfy in order for the User Story to be accepted by a Product Owner, user, customer, or other stakeholder.

**Product backlog:** a prioritized list of stories that are waiting to be worked on.

**What are your strengths :**

As a automation developers, one of strength , I am good in coding with vb script and Java ,python

I can work with team or I can develop and implement the framework independently

I am good at adopt new technologies , in recent years , I enhance my skills in git ,Jenkins, python

**Diff between xpath and CSS**

Theoretically *cssSelector* taking less time then *XPath* as *XPath* need to traverse through HTML DOM. *XPath* we can search elements backward or forward in the DOM hierarchy while *CSS* works only in a forward direction.

UHC :

It is web based application ,called COMPAS Comprehensive policy Administration system. Application having multiple modules like enrollment, payment, billing and claims

In my current project I have implemented BDD cucumber framework

For multiple modules we have created multiple folder structures in framework. For each scenario , we have developed feature files and developed step definitions using java.Input data we are passing through feature files.

I have developed and scheduled Jenkins job, through which we can triggered the automation scripts which v

will be executed in saucelabs, its one of cloud environment.

After script execution it generates html reports and sends an email to user.

**TDD :**

TDD stands for Test Driven Development. In this software development technique,

we create the test cases first and then write the code underlying those test cases. Although TDD is a development technique,

it can also be used for automation testing development.

1) Write a test case:

2) Run all the test cases:

3) Develop the code for that test cases:

4) Run test cases again:

5) Refactor your code:

6) Repeat the steps 1- 5 for new test cases:

**BDD :**

BDD stands for Behavior Driven Development. BDD is an extension to TDD where instead of writing the test cases, we start by writing a behavior.

The scenario defined in the BDD approach makes it easy for the developers, testers and business users to collaborate.

1) Write the behavior of the application:

2) Write the automated scripts:

3) Implement the functional code:

4) Check if the behavior is successful:

5) Refactor or organize code:

6) Repeat the steps 1-5 for new behavior:

|  |  |
| --- | --- |
| **TDD** | **BDD** |
| The process starts by writing a test case. | The process starts by writing a scenario as per the expected behavior |
| Test cases are written in a programming language. | Scenarios are more readable when compared to TDD as they are written in simple English format. |
| Collaboration is required only between the developers. | Collaboration is required between all the stakeholders. |
|  |  |