

# Appendix

## 1. Practical Part

This section contains all the practical work about manual test and automation test with test case sample and automation tool's screenshots with explained source code. Also, it includes the table of comparisons with tests execution and preparation with approximate cost.

### 1.1 Unit testing:

The unit test part contains the test case sample with the selenium automation tool screenshots. Also, it contains the source code from selenium, and it is commented for better understanding of the action execution by the code.

#### 1.1.1 Test case sample for Unit testing:

<b>Test Case ID</b>	01.
<b>Test Scenario</b>	Login functionality Of Amazon.in online shopping website.
<b>Test case title</b>	Verify the log in functionality of Amazon.in online shopping website with valid login ID and password.
<b>Pre-conditions</b>	The account is already created on the Amazon.in website.
<b>Dependencies</b>	No, there is not any dependencies for this test case.
<b>Test Designed By</b>	Ankit Vaniya
<b>Test Designed date</b>	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Open the website. Amazon.in in google chrome web browser.		Website is opened successfully.			
2	Click on sign in Button.		User is successfully navigated on the login page of Amzon.in website.			
3	Please enter the valid login id in the textbox.	e.g., ankit.xxxgmail .com	Login id successfully added.			
4	Click on “continue” button.		Password textbox is appeared.			
5	Enter valid password in textbox.	e.g., Password: xxxxxxxx	Password is inserted successfully.			
6	Please click on sign in button.		The user is successfully login in amazon.in website.			

## 1.1.2 Screenshots of Unit testing:

Figure-1.1.2.1

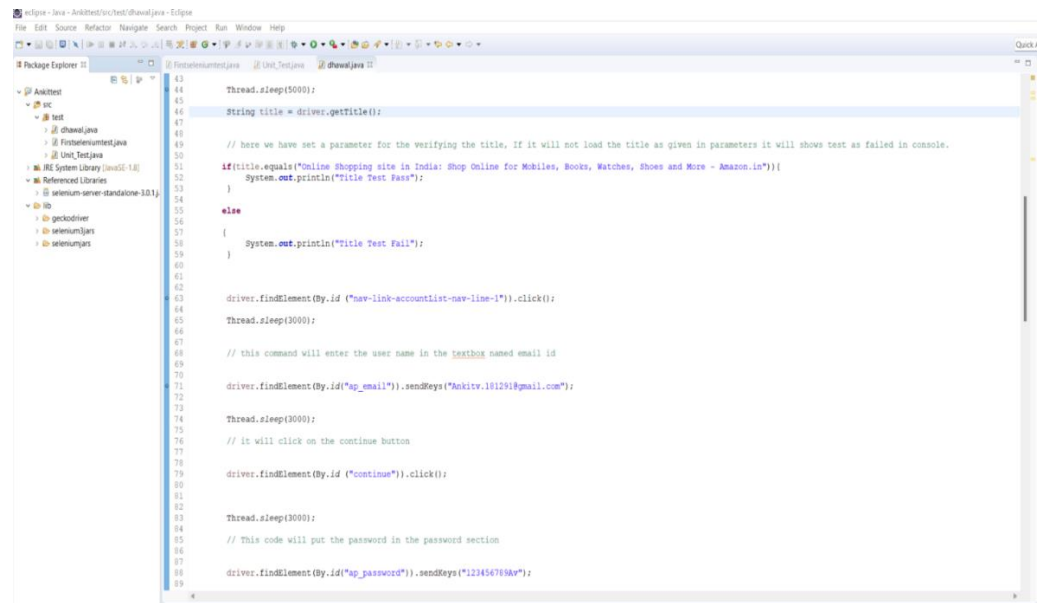
```

1 package test;
2 import java.util.ArrayList;
3 import java.util.List;
4 import org.openqa.selenium.support.ui.Select;
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.JavascriptExecutor;
7 import org.openqa.selenium.WebDriver;
8 import org.openqa.selenium.WebElement;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.support.ui.WebDriverWait;
11
12
13 public class dhaval {
14
15
16
17
18     // System Property for google Chrome driver
19     System.setProperty("webdriver.chrome.driver", "C:\\Users\\SIR14\\Downloads\\speco\\chromedriver_win32\\chromedriver.exe");
20
21
22     // Instantiate a ChromeDriver class.
23
24     WebDriver driver=new ChromeDriver();
25
26     WebDriverWait wait = new WebDriverWait(driver,5000);
27
28
29
30
31     // for load the window in maximum size.
32     driver.manage().window().maximize();
33
34
35     //searching for the amazon online shopping website and launching it
36     driver.get("https://amazon.in/");
37
38
39
40     // Thread . sleep is used as waiting time between executing two steps
41
42
43     Thread.sleep(5000);
44
45     String title = driver.getTitle();
46
47

```

Source: Author processing

Figure-1.1.2.2



```
43 Thread.sleep(5000);
44
45 String title = driver.getTitle();
46
47 // here we have set a parameter for the verifying the title, if it will not load the title as given in parameters it will shows test as failed in console.
48
49 if(title.equals("Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in")){
50     System.out.println("Title Test Pass");
51 }
52 else
53 {
54     System.out.println("Title Test Fail");
55 }
56
57 driver.findElement(By.id ("nav-link-accountlist-nav-line-1")).click();
58 Thread.sleep(3000);
59
60 // this command will enter the user name in the textbox named email id
61
62 driver.findElement(By.id("ap_email")).sendKeys("Ankitv.101291@gmail.com");
63
64 Thread.sleep(3000);
65
66 // it will click on the continue button
67
68 driver.findElement(By.id ("continue")).click();
69
70 Thread.sleep(3000);
71
72 // This code will put the password in the password section
73
74 driver.findElement(By.id("ap_password")).sendKeys("123456789a*");
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
```

Source: Author processing

Figure-1.1.2.3



```
77
78 driver.findElement(By.id ("continue")).click();
79
80
81 Thread.sleep(3000);
82
83 // This code will put the password in the password section
84
85 driver.findElement(By.id("ap_password")).sendKeys("123456789a*");
86
87
88 Thread.sleep(3000);
89
90 // it will click on the submit button
91
92 driver.findElement(By.id ("signInSubmit")).click();
93
94
95
96
97
98
99
```

Source: Author processing

### 1.1.3 Source codes for Unit testing:

```
package test;
import java.util.ArrayList;
import java.util.List;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class Ankit {

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

        // System Property for Chrome Driver

        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\91814\\Downloads\\gecko\\chromedriver_win32\\chromedriver.exe");

        // Instantiate a ChromeDriver class.

        WebDriver driver=new ChromeDriver();

        WebDriverWait wait = new WebDriverWait(driver,5000);

        // for load the window in maximum size.

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

        //Searching for the amazon online shopping website and launching it

        driver.get("https://amazon.in/");

        // Thread . sleep is used as waiting time between executing two steps

        Thread.sleep(5000);

        String title = driver.getTitle();

        // here we have set a parameter for the verifying the title,
        If it will not load the title as given in parameters, it
        will shows test as failed in console.

        if(title.equals("Online Shopping site in India: Shop Online for Mobiles,
Books, Watches, Shoes and More - Amazon.in")){

            System.out.println("Title Test Pass");
        }

        else

        {

            System.out.println("Title Test Fail");
        }

        driver.findElement(By.id ("nav-link-accountList-nav-line-1")).click();
    }
}
```

```
Thread.sleep(3000);

// this command will enter the username in the textbox named email id

driver.findElement(By.id("ap_email")).sendKeys("Ankitv.xxxxx@gmail.com");

Thread.sleep(3000);

// it will click on the continue button

driver.findElement(By.id ("continue")).click();

Thread.sleep(3000);

// This code will put the password in the password section

driver.findElement(By.id("ap_password")).sendKeys("xxxxxxxxxAv");

Thread.sleep(3000);

driver.findElement(By.id ("signInSubmit")).click();
```

## 1.2 Integration testing:

The integration test part contains the test case sample with the selenium automation tool screenshots. Also, it contains the source code from selenium, and it is commented for better understanding of the action execution by the code. Also, integration test is related to unit test practical part so please consider the test case and screenshot and source code from it.

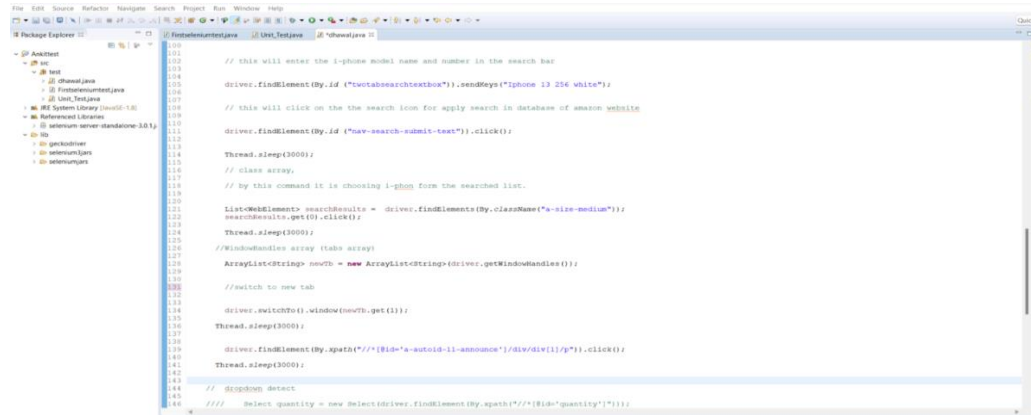
### 1.2.1 Test case sample for Integration testing:

<b>Test Case ID</b>	02.
<b>Test Scenario</b>	Search bar functionality of Amazon.in website.
<b>Test case title</b>	Verify the search button is working properly with database of Amazon.in online shopping website.
<b>Pre-conditions</b>	The account is already created on the Amazon.in website.
<b>Dependencies</b>	No, there is not any dependencies for this test case.
<b>Test Designed By</b>	Ankit Vaniya
<b>Test Designed date</b>	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Please perform all steps from test case 01.		All the steps from the test case 01 is successfully performed.			
2	Please check the user is successfully logged in the Amazon.in website.		The User is successfully logged on the website.			
3	Please Enter the data in search bar and click to search.	Data: I-phone13 256 white.	The Search result for the I-phone13 product is visible on the webpage.			
4	Click on I-phone as per your choice.		I-phone is selected and opened in new tab.			
5	Select the quality, color etc. of your choice.		Options are selected for the chosen I-phone product.			
6	Please check if "Add to cart" button visible.		"Add to cart button" is visible on screen.			
7	Please click on that button.		The product is successfully added to the cart.			

## 1.2.2 Screenshots of Integration testing:

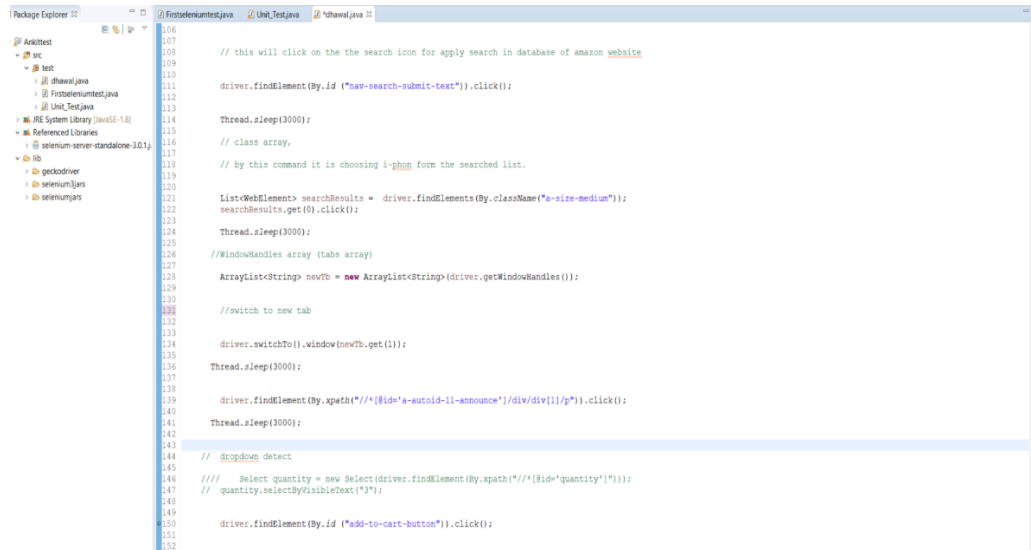
Figure-1.2.2.1



```
100
101
102 // this will enter the i-phon model name and number in the search bar
103
104 driver.findElement(By.id ("twotabsearchtextbox")).sendKeys("iphone 13 256 white");
105
106 // this will click on the the search icon for apply search in database of amazon website
107
108 driver.findElement(By.id ("nav-search-submit-text")).click();
109
110 Thread.sleep(3000);
111
112 // class array,
113
114 // by this command it is choosing i-phon form the searched list.
115
116 List<WebElement> searchResults = driver.findElements(By.className("s-size-medium"));
117 searchResults.get(0).click();
118 Thread.sleep(3000);
119
120 //WindowHandles array (tabs array)
121 ArrayList<String> newTab = new ArrayList<String>(driver.getWindowHandles());
122
123 //switch to new tab
124
125 driver.switchTo().window(newTab.get(1));
126 Thread.sleep(3000);
127
128 driver.findElement(By.xpath("//*[@id='a-autoid-l1-announce']/div/div[1]/p")).click();
129 Thread.sleep(3000);
130
131 // dropdown detect
132
133 // Select quantity = new Select(driver.findElement(By.xpath("//*[@id='quantity']")));
134
135 driver.findElement(By.id ("add-to-cart-button")).click();
136
```

Source: Author processing

Figure-1.2.2.2



```
104
105
106 // this will click on the the search icon for apply search in database of amazon website
107
108 driver.findElement(By.id ("nav-search-submit-text")).click();
109
110 Thread.sleep(3000);
111
112 // class array,
113
114 // by this command it is choosing i-phon form the searched list.
115
116 List<WebElement> searchResults = driver.findElements(By.className("s-size-medium"));
117 searchResults.get(0).click();
118 Thread.sleep(3000);
119
120 //WindowHandles array (tabs array)
121 ArrayList<String> newTab = new ArrayList<String>(driver.getWindowHandles());
122
123 //switch to new tab
124
125 driver.switchTo().window(newTab.get(1));
126 Thread.sleep(3000);
127
128 driver.findElement(By.xpath("//*[@id='a-autoid-l1-announce']/div/div[1]/p")).click();
129 Thread.sleep(3000);
130
131 // dropdown detect
132
133 // Select quantity = new Select(driver.findElement(By.xpath("//*[@id='quantity']")));
134 // quantity.selectByVisibleText("3");
135
136 driver.findElement(By.id ("add-to-cart-button")).click();
137
```

Source: Author processing

### 1.2.3 Source codes for Integration testing:

```
// this will enter the Iphone model name and number in the
search bar

driver.findElement(By.id ("twotabsearchtextbox")).sendKeys("Iphone
13 256 white");

// this will click on the the search icon for apply search in
database of amazon website

driver.findElement(By.id ("nav-search-submit-text")).click();

Thread.sleep(3000);

// class array,
// by this command it is choosing Iphone form the searched list.

List<WebElement> searchResults =
driver.findElements(By.className("a-size-medium"));
searchResults.get(0).click();

Thread.sleep(3000);

//WindowHandles array (tabs array)

ArrayList<String> newTb = new
ArrayList<String>(driver.getWindowHandles());

//switch to new tab

driver.switchTo().window(newTb.get(1));

Thread.sleep(3000);

driver.findElement(By.xpath("//*[@id='a-autoid-11-
announce']/div/div[1]/p")).click();

Thread.sleep(3000);

driver.findElement(By.id ("add-to-cart-button")).click();
```





### 1.3 System testing:

The system test part contains the test case sample with the selenium automation tool screenshots. Also, it contains the source code from selenium, and it is commented for better understanding of the action execution by the code. Also, system test is relate to unit test and integration test practical part so please consider the test case and screenshot and source code from it.

#### 1.3.1 Test case sample for System testing:

<b>Test Case ID</b>	03.
<b>Test Scenario</b>	Amazon.in system process from searching product to payment.
<b>Test case title</b>	Verify the system process flow of Amazon.in website from login, searching to payment process.
<b>Pre-conditions</b>	The account is already created on the Amazon.in website.
<b>Dependencies</b>	No, there is not any dependencies for this test case.
<b>Test Designed By</b>	Ankit Vaniya
<b>Test Designed date</b>	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Please perform all steps from test case 01.		All the steps from the test case 01 is successfully performed.			
2	Please check the user is successfully logged in the Amazon.in website		User is successfully logged in.			
3	Please Enter the data in search bar and click on search.	Data: I-phone13 256 white.	Search result for the I-phone product is visible in webpage.			

4	Click on I-phone as per your choice.		I-phone is selected and opened in new tab.			
5	Select the quality, color etc. of your choice.		Options are selected for the chosen product.			
6	Please check if “Add to cart” button visible.		“Add to cart button” is visible on screen.			
7	Please click on that button.		The product is successfully added to the cart.			
8	Please click on proceed to checkout button		User is navigated to shipping address page.			
9	Please select the address which is already exit there.		The shipping address is selected and continue button visible now.			
10	Please click on continue button		User is navigated to payment option page.			

### 1.3.2 Screenshots of System testing:

Figure-1.3.2.1

```

163
164
165 // this is clicking on the button go to cart
166
167
168 driver.findElement(By.linkText("Go to Cart")).click();
169
170
171 Thread.sleep(3000);
172
173
174 // this command will click for the proceed to buy
175
176
177 driver.findElement(By.xpath("//*[@id='sc-buy-ptc-button']/span/input")).click();
178
179
180 Thread.sleep(3000);
181
182
183 // here I click for the deliver to this address which is already added before
184
185
186 driver.findElement(By.className("ship-to-this-address a-button a-button-primary a-button-span12 a-spacing-medium")).click();
187
188
189 Thread.sleep(3000);
190
191 // here it will click on the continue button to proceed to payment page
192
193
194 driver.findElement(By.className("a-button-inner")).click();
195
196
197
198 driver.quit();
199
200

```

Source: Author processing

### 1.3.3 Source codes for System testing:

```
// this is clicking on the button go to cart

    driver.findElement(By.linkText("Go to Cart")).click();

    Thread.sleep(3000);

// this command will click for the proceed to buy

        driver.findElement(By.xpath("//*[@id='sc-buy-box-ptc-
button']/span/input")).click();

        Thread.sleep(3000);

// here I click for the deliver to this address which is
already added before

    driver.findElement(By.className("ship-to-this-address a-button
a-button-primary a-button-span12 a-spacing-medium")).click();

        Thread.sleep(3000);

// here it will click on the continue button to
proceed to payment page.

        driver.findElement(By.className("a-button-
inner")).click();

        driver.quit();

    }
}
```

### 1.4 Test case execution video link:

[https://docs.google.com/presentation/d/1Z9qA3BAP517iJjVuMn1U\\_gpWP-EEVRP/edit?usp=share\\_link&oid=102686019584461517141&rtpof=true&sd=true](https://docs.google.com/presentation/d/1Z9qA3BAP517iJjVuMn1U_gpWP-EEVRP/edit?usp=share_link&oid=102686019584461517141&rtpof=true&sd=true)

**1.5 Approximate data related the time/cost/factors of manual and automation testing:**

**1.5.1 Approximate time taken for Test case/ Script preparation:**

<b>Tester</b>	<b>Experience</b>	<b>Test case 1 Preparation time.</b>	<b>Test case 2 Preparation time.</b>	<b>Test case 3 Preparation time.</b>
Manual Tester 1	6m – 1 year	20 – 22 Minutes	18 – 20 Minutes	15 – 17 Minutes
Manual Tester 2	2.5 years	15 – 18 Minutes	13 – 15 Minutes	7 - 10 Minutes
Manual Tester 3	3 years +	10 – 12 Minutes	8 – 10 Minutes	5 - 7 Minutes
End User	-	15 – 30 Minutes	13 – 20 Minutes	10 – 15 Minutes
Automation Tester 1	6m – 1 year	40 – 1 hour	30 – 40 Minutes	20 -25 Minutes
Automation Tester 2	3.5 years	10- 15 Minutes	17 - 20 Minutes	22 – 25 Minutes

**1.5.2 Manual tester’s approximate test case execution time with hourly/monthly wages:**

<b>Tester</b>	<b>Experience</b>	<b>Test case 1 Execution time.</b>	<b>Test case 2 Execution time.</b>	<b>Test case 3 Execution Time.</b>	<b>Approximate Wages per Hour/month</b>
Manual Tester 1	6m – 1 year	1.10 Minutes	1.55 Minutes	2.35 Minutes	255.68 / 45000 CZK
Manual Tester 2	2.5 years	55 Seconds	1.40 Minutes	2.10 Minutes	304.97 / 53675 CZK

Manual Tester 3	3 years +	50 seconds	1.30 Minutes	1.50 Minutes	505.59 / 88984 CZK
End User	-	1.35 Minutes	2.10 Minutes	2.55 Minutes	-

**1.5.3 Automation tester's approximate test case execution time with hourly/monthly wages:**

Automation tester	Experience	Test case 1 Execution time.	Test case 2 Execution time.	Test case 3 Execution Time.	Approximate wages Hour/month
Automation Tester 1	6m – 1 year	31 sec	46 sec	52 sec	323.56 / 57000 CZK
Automation Tester 2	3.5 years	18 sec	27 sec	37 sec	547.23 / 96131 CZK

**1.5.4 Comparison of manual and automation testing as per the software testing factors:**

Factors	Manual testing	Automation testing
Human efforts	More	Less
Time	More	Less
Money	Cheap for small project	Cheap for long big project

Reusability	Less	More
Programing Knowledge	No	Yes
Accuracy	Less	High
Customer experience	Help in improving the customer experience	No guarantee for the positive customer experience
Compatibility	High	Less

### 1.5.5 Economic benefits of Software quality assurance, Manual testing and Automation testing:

Role	Responsibilities	Average salary in Prague
Test Manager	Manager all activity, Negotiation with management, development and customer, Preparation of test approach and strategy.	464.94 Czech Koruna per hour.
Test Leader	Reporting and negotiation with test manager, creation of test plan according to test strategy, creating task for team member according to requirement, fulfill the requirement for testing.	411.93 Czech Koruna per hour.
Test Analyst	Analysis and preparation for test documentation.	281.94 Czech Koruna per hour.
QA Engineer	Implementation of defined software quality assurance processes and standard.	330.13 Czech Koruna per hour.
Software Manual Tester	Testing software manually, creation test cases, test execution report, bug report, retesting.	276.70 Czech Koruna per hour.
Software Automation Tester	Managing and creating automation testing. Creation test report and documentation.	378.78 Czech Koruna per hour.

Country	Software Assurance	Quality	Software testing	Automation testing	Approx exchange rate
Czech Republic	58104 Czech Koruna / M		48700 Czech Koruna / M	66666 Czech Koruna / M	1 CZK = 0.045 USD and 3.5 INR.
India	45000 – 50000 Indian Rupee / M		24710 – 26129 Indian Rupee / M	33481 – 34742 Indian Rupee / M	1 INR = 0.012 USD, 0.26 CZK
USA	6406.91 USD / M		5719.16 USD / M	7155.75 USD / M	1 USD = 22 CZK, 82.10 INR.

### 1.5.6 Why Selenium tool is good for Automation testing?

Selenium is highly used by the people. It is open-source software available free of cost. The tutorials to learn and use are available on YouTube and Google. It is supported by the multiple language, platforms, and browsers.

Selenium	Automated test execution	43.2%
Jira	Test management, bug tracking	38%
Cucumber	Automated test execution	9.5%
SoapUI	Automated test execution, API testing	8.5%
Jmeter	Performance Testing	7.8%
Team Foundation Server	Test management	7.2%
HP QC / ALM	Test management	6.3%
LoadRunner	Performance Testing	6.2%
Confluence	Test management	5.8%
TestNG	Automated test execution, unit testing	5.8%
HP QTP	Automated test execution	5.5%
TestRail	Test management	5.5%
JUnit	Unit testing	4.4%
Katalon Studio	Automated test execution	2.65%
Bugzilla	Bug tracking	1.94%
TestComplete	Automated test execution	1.94%
Microsoft Test Manager	Test management	1.76%
Ranorex	Automated test execution	1.41%
Tosca	Automated test execution	1.23%

<b>Language</b>	<b>Platforms</b>	<b>Browser</b>
Java	Mac	Google Chrome
Java Script	Windows	Firefox
Python	Linux	Safari
PHP	Android (Selendroid, Appium, Robotium)	Opera
Ruby	iOS (with ios-driver or appium)	Microsoft Edge
C#		Internet Explorer
Objective-C		